

| | | |
|-----------|--|-----------|
| 1. | ÚVOD DO MS ACCESS 2007 | 3 |
| 1.1. | RELAČNÁ DATABÁZA | 3 |
| 1.2. | NÁVRH DATABÁZY - TROCHA TEÓRIE..... | 4 |
| 1.2.1. | <i>Prvá normálna forma - 1NF</i> | 5 |
| 1.2.2. | <i>Druhá normálna forma - 2NF</i> | 6 |
| 1.2.3. | <i>Tretia normálna forma - 3NF</i> | 7 |
| 1.2.4. | <i>Vzťahy medzi tabuľkami</i> | 8 |
| 1.3. | DATABÁZA KNIŽNICA | 10 |
| 1.3.1. | <i>Evidencia kníh</i> | 10 |
| 1.3.2. | <i>Evidencia čitateľov</i> | 10 |
| 1.3.3. | <i>Evidencia výpožičiek</i> | 11 |
| 1.4. | VYTVORENIE DATABÁZY V ACCESS 2007 | 12 |
| 1.5. | OBJEKTY V ACCESS 2007..... | 13 |
| 1.6. | POUŽÍVATEĽSKÉ ROZHRANIE | 14 |
| 2. | TABUĽKY | 16 |
| 2.1. | VYTVORENIE PRVEJ TABUĽKY | 16 |
| 2.1.1. | <i>Údajové zobrazenie pri tvorbe tabuľky</i> | 17 |
| 2.1.2. | <i>Návrhové zobrazenie pri tvorbe tabuľky</i> | 18 |
| 2.2. | PREPOJENIE TABULIEK | 25 |
| 2.3. | PRÁCA S TABUĽKOU | 30 |
| 3. | DOTAZY | 33 |
| 3.1. | VÝBEROVÝ DOTAZ..... | 34 |
| 3.1.1. | <i>Parameter v dotaze</i> | 37 |
| 3.1.2. | <i>Vytvorenie nového stĺpca</i> | 38 |
| 3.1.3. | <i>Agregačné funkcie</i> | 39 |
| 3.1.4. | <i>Dotaz využívajúci viacero tabuliek</i> | 41 |
| 3.2. | VYTVÁRACÍ DOTAZ..... | 42 |
| 3.3. | PRIPÁJACÍ DOTAZ..... | 42 |
| 3.4. | ODSTRAŇOVACÍ PRÍKAZ | 43 |
| 3.5. | AKTUALIZAČNÝ DOTAZ..... | 44 |
| 3.6. | KRÍŽOVÝ DOTAZ | 45 |
| 3.7. | KONTINGENČNÁ TABUĽKA A GRAF..... | 45 |
| 4. | FORMULÁRE | 50 |
| 4.1. | VYTVORENIE JEDNODUCHÉHO FORMULÁRA..... | 50 |
| 4.2. | VYTVORENIE FORMULÁRA S PODFORMULÁROM..... | 52 |
| 4.2.1. | <i>Pridanie podformulára do hlavného formulára</i> | 54 |
| 4.2.2. | <i>Pridanie agregáčnych výpočtov do formulára</i> | 56 |
| 4.3. | FORMULÁR ZO SPOJENÝCH TABULIEK | 59 |
| 4.4. | SWITCHBOARD..... | 61 |
| 5. | MAKRÁ..... | 63 |
| 6. | ZOSTAVY | 69 |
| 6.1. | VYTVORENIE ZOSTAVY POMOCOU SPRIEVODCU..... | 69 |
| 6.2. | VYTVORENIE ZOSTAVY NA ZÁKLADE TABUĽKY PRÍKAZOM ZOSTAVA | 71 |

| | |
|---|-----------|
| 7. FUNKCIE | 74 |
| 7.1. FUNKCIE DÁTUMU A ČASU | 74 |
| 7.2. AGREGAČNÉ FUNKCIE | 76 |
| 7.3. MATEMATICKÉ FUNKCIE..... | 76 |
| 7.4. TEXTOVÉ FUNKCIE..... | 76 |
| 7.5. OSTATNÉ FUNKCIE | 77 |
| ZOZNAM OBRÁZKOV A TABULIEK | 80 |

1. Úvod do MS Access 2007

Microsoft Access je program na vytváranie a správu relačných databáz, t. j. umožňuje navrhnuť relačnú databázu, vkladať do nej údaje, poskytovať z nej požadované údaje alebo ich aktualizovať. Vznikol v roku 1992 a od roku 1993 je časťou balíka Microsoft Office Professional.

Existuje viacero výkonnejších relačných databázových systémov (Oracle, Microsoft SQL, Informix, MySQL, PostgreSQL, ...), ktoré sú ale optimalizované pre spustenie na samostatných serveroch. Access je určený pre bežných používateľov a na rozdiel od spomínaných systémov je určený pre desktopy.

1.1. Relačná databáza

Databáza je štruktúrovaná množina dát uložená na pamäťovom médiu. Program, umožňujúci prístup k uloženým dátam (k databáze) a manipuláciu s nimi sa nazýva databázový systém. Používa sa tiež názov systém riadenia báz dát (SRBD), čo je preklad z database management system (DBMS). Je zvykom, že ak je z kontextu jasné či ide o uložené dáta alebo softvér, tak sa používa iba pojem databáza.

Existuje viacero druhov databáz, ktoré sa líšia podľa toho ako sú v nich dáta usporiadané. Relačné databázy sú najrozšírenejšie. Okrem nich existujú:

- Hierarchické - údajové prvky sú prepojené do stromovitej štruktúry, podobne ako rodokmeň s práve jedným rodičom.
- Sieťové – vychádzajú z hierarchických ale každý dátový prvok môže mať viacerých rodičov.
- Objektové – informácie sú uložené vo forme objektov.

Model relačnej databázy navrhol E.F. Codd v roku 1969. Model je založený na teórii množín a logike. Hlavnou výhodou relačných databáz je prirodzená reprezentácia spracovávaných dát. Slovo *relačná* znamená, že údaje sa ukladajú do tabuliek (relácií). Databáza bežne obsahuje množstvo tabuliek, ktoré bývajú rôzne prepojené. Obr. 1.1. ukazuje príklad jednoduchkej tabuľky.

Riadok sa v takýchto tabuľkách nazýva záznam a napríklad pri evidencii kníh v knižnici by jeden riadok mohol predstavovať jednu knihu.

Stĺpec sa nazýva pole alebo atribút. V stĺpcoch sú uchovávané jednotlivé informácie. V prípade knižničnej databázy by to mohli byť informácie: názov, cena, autor a pod. Každý stĺpec má v rámci tabuľky jedinečný názov a obsahy buniek v ňom musia byť toho istého typu. Napríklad ak stanovíme rok vydania ako celé číslo, tak používateľ nebude môcť zadať číslo desatinné ani text (napríklad „145 pred našim letopočtom“).

| Identifikácia | Názov | Cena | Rok vydania | Autor | Žáner |
|---------------|--------------------|---------|-------------|-----------------|----------|
| 2 | V slepých uličkách | 11,50 € | 1999 | Kamil Peteraj | Poézia |
| 3 | Radost z poznání | 14,00 € | 1999 | Richard Feynman | Eseje |
| 4 | Umenie klamu | 20,00 € | 2002 | Kevin Mitnick | Učebnica |

Obr. 1.1. Jednoduchá tabuľka

Pre používateľa, ktorý sa stretne s databázovým systémom po prvýkrát je možno prekvapujúca potreba ďalšieho programu. Veď údaje by sa dali do tabuliek ukladať napríklad v MS Excel. Pre malý objem dát by to bolo možné. Musíme si ale uvedomiť, že Excel je určený hlavne na výpočty. Vkladanie, aktualizovanie alebo vyhľadávanie dát nie je činnosť, pre ktorú je vhodný a so zväčšujúcim sa objem údajov by sa to prejavilo negatívne na dobe spracovania. Databázové systémy musia spĺňať podmienky, ktoré by sa v programe Excel (alebo v inom nedatabázovom programe) nedali docieľiť. Sú to najmä:

- **Perzistencia** – trvalosť, dáta sa zachovávajú aj po skončení programu, ktorý s nimi pracuje.
- **Nezávislosť** – programy prístupujúce k údajom pracujú nezávisle na tom, kde a ako sú údaje uložené, t. j. zmena fyzického uloženia údajov neovplyvní funkčnosť databázového systému..

- **Konzistencia údajov** – ak sa údaj nachádza v databáze viackrát, musí mať všade rovnakú hodnotu.
- **Nízka redundancia** - údaje by sa nemali v databáze zbytočne opakovať. Ak sa nachádzajú v databáze tie isté údaje opakované, tak to okrem zaberania diskového priestoru môže zvyšovať pravdepodobnosť nekonzistencie údajov.
- **Zachovávať integritu údajov** – databázové systémy umožňujú pomocou rôznych podmienok a pravidiel zabrániť vykonať neželané operácie (napr. odstrániť autora ak máme od neho nejakú knihu) alebo zadanie chybných údajov (napr. 29. 2. 2010).
- **Zabezpečenie pred neoprávneným prístupom** – databázový systém umožňuje nastaviť pre používateľov pravidlá prístupu k databáze alebo jej jednotlivým častiam.
- **Konkurentnosť** – systém musí umožniť súčasnú prácu viacerým používateľom a riešiť prípadné kolízie ak sa pokúsia o zmenu tých istých dát.

Definícia relačnej databázy okrem toho vyžaduje aby bol každý riadok v tabuľke **jedinečný**. V tabuľke na obr. 1.1 sa ale teoreticky môže stať, že knihu s rovnakým názvom, cenou i žánrom napíšu autori, ktorí sa volajú rovnako. Preto sa v takomto prípade na zaručenie jedinečnosti (a teda rozlíšenia takýchto kníh) pridáva stĺpec navyše. Databázový systém môže v tomto stĺpci automaticky priradovať hodnoty a dbať, aby žiadne dve neboli rovnaké. Takýto stĺpec sa nazýva **klúč** (jedinečné označenie záznamu). Access vytvára tieto stĺpce automaticky pomenúva ich *Identifikácia*.

Kľúčom môže byť aj spojenie dvoch alebo viacerých stĺpcov ak takéto spojenie jednoznačne identifikuje každý riadok. Najmenší z kľúčov sa nazýva primárny (v Accesse hlavný) kľúč. Často sú tieto stĺpce pomenované aj *ID* alebo *ID knihy* a pod. a pri ich popisovaní sa podčiarkujú, aby sa zvýraznilo, že nejde o obyčajný stĺpec. Ak sa v tabuľke nachádza stĺpec s primárnym kľúčom inej tabuľky, tak sa nazýva cudzí kľúč.

1.2. Návrh databázy - trocha teórie

V tejto kapitole si objasníme niekoľko pojmov a postupov, ktoré budeme potrebovať pri návrhu štruktúry databázy (t. j. čo budeme evidovať, ako bude databáza vyzerat' apod.), s ktorou budeme ďalej pracovať v celej učebnici. Pracovať znamená, že na základe tohto návrhu v nasledujúcich kapitolách postupne databázu vytvoríme a ukážeme si ako ju možno využívať.

Učebnica je teda koncipovaná tak, aby zachytila životný cyklus jednej skoro reálnej databázy. Skoro reálnej preto, lebo učebnica je zameraná na oboznámenie sa s MS Access a nie na vytváranie profesionálnych databáz. Napríklad v reálnej databáze je množstvo tabuliek, ktorých vytváranie je opakujúca sa činnosť – v realite potrebná, ale pre pochopenie postupu nezaujímavá. Nám bude stačiť vysvetliť postup jedenkrát. Napríklad, ak by sa v reálnej tabuľke evidoval, okrem iného, plat a odmeny, v našej odmeny vynecháme. Kto chce, môže si ich pridať do tabuľky rovnako ako plat (postup je popísaný už v kapitole 2.1.1). Naša databáza bude mať preto menej tabuliek ako reálna a tieto tabuľky môžu mať menej stĺpcov, čo snád' uľahčí pochopenie.

Databáza, s ktorou budeme pracovať, bude slúžiť na evidenciu kníh v knižnici. Vychádzame z reálnych potrieb, ktoré by knižnica mohla mať – to je prvý bod každého návrhu databázy – zistiť čo konkrétne sa bude evidovať:

- Evidencia kníh,
- evidencia čitateľov,
- evidencia ~~aktuálne~~ (prečiarknutie je vysvetlené nižšie) požičaných kníh.

Druhým bodom je čo by sme chceli z databázy zisťovať:

- Či niekto nemešká s vrátením (zrejme sa to bude zisťovať na základe dátumu požičania, preto si poznačíme, že pri evidencii požičaných kníh je nutné zaznamenať dátum požičania).
- Nájsť a odmeniť čitateľa, ktorý si prečítal (požičal) najviac kníh za rok. To nás privedie na myšlienku, že musíme evidovať požičané knihy aj po ich vrátení. Pôvodný úmysel mohol byť

taký, že pri požičaní sa zapíše čitateľ do evidencie aktuálne požičaných kníh a pri vrátení knihy sa z evidencie vymaže. To by sme ale prišli históriu výpožičiek.

- Nájst' knihy, ktoré sa nikdy alebo dlho (napríklad 5 rokov) nepožičali. Alebo obdobne nájst' neaktívneho čitateľa. Vidíme, že naozaj sme slovo aktuálne prečiarkli vhodne. Je jasné, že musíme evidovať históriu čitateľov aj kníh, nielen aktuálne výpožičky. Už sa môže črtat' ako bude vyzerat' evidencia požičaných kníh – tabuľka, v ktorej každý riadok bude jedna výpožička a stĺpce (čo budeme o každej výpožičke evidovať) budú požičaná kniha, čitateľ a dátum výpožičky.
- Nájst' čitateľov, ktorý najčastejšie meškali s vrátením alebo nájst' knihy, ktoré boli často vracané s omeškaním. Táto požiadavka nám ukáže, že musíme do tabuľky požičané knihy pridať aj dátum vrátenia.
- Atd'. napríklad:
 - Kto všetko mal požičanú knižku XY?
 - Kto ju má požičanú teraz?
 - Ktorá knižka/autor/žáner sa najviac požičiaval?
 - Ktorý čitateľ má najviac požičaných knižiek?
 - Ktorý čitateľ si doteraz požičal najviac knižiek?
 - Ktorý čitateľ si od začiatku roka požičal najviac knižiek?
 - Ktorí čitatelia majú ktoré knižky požičané dlhšie ako mesiac?
 - Ktorá knižka bola najdlhšie vypožičaná? A pod.

Vidíme, že pochopenie toho na čo má databáza slúžiť vedie k celkom dobrej predstave o tom ako bude vyzerat' a táto predstava nás môže nasmerovať k dobrému návrhu. Bohužiaľ je pravidlo, ktoré potvrdí každý tvorca databáz, že používatelia často zabudnú na niektorú požiadavku alebo si po roku používania databázy vymyslia požiadavky nové.

Aby sa nové (samozrejme aj staré) požiadavky ľahšie implementovali, je potrebné pri návrhu databáz rešpektovať určité pravidlá. Aplikovanie týchto pravidiel sa nazýva normalizácia. Normalizované tabuľky majú optimalizovanú štruktúru údajov a tým umožňujú rýchlejšie zmeny údajov v tabuľkách, prípadne vyšší počet paralelných (t. j. súčasných) prístupov.

Popíšeme si prvé tri normalizačné formy, ktoré sa bežne používajú. Celkovo ich je päť. Platí, že čím sú tabuľky vo vyšších normálnych formách, tým je miera redundancie menšia. V nasledujúcich tabuľkách sú primárne kľúče podčiarknuté.

1.2.1. Prvá normálna forma - 1NF

Tabuľka spĺňa podmienku 1NF, ak každý stĺpec obsahuje iba atomické hodnoty, t. j. iba hodnoty, ktoré sa z pohľadu databázy už ďalej nedajú deliť na menšie. Inak povedané: každému atribútu je priradená vždy len jedna hodnota a nie množina hodnôt.

Tab. 1.1. Nenormalizovaná tabuľka

| <u>Čitateľ</u> | <u>Požičané knihy</u> |
|----------------|--------------------------------|
| Nováková | Nadácia, Zatmenie, Access 2007 |
| Labuda | Súmrak, Nov |

Na vysvetlenie 1NF zvolíme už spomínanú zjednodušenú tabuľku evidencie *aktuálne* požičaných kníh (bez dátumov). V tabuľke bude pri každom čitateľovi zoznam požičaných kníh – ak knihu čitateľ vráti, tak ju zmažeme a ak vráti všetky zmažeme riadok. Tab. 1.1 teda ukazuje vcelku amatérsky spôsob ako evidovať

požičané knihy. Problém je hlavne v tom, že keď čitateľ vráti knihu, tak je potrebné túto knihu v bunke nájsť, čo je pomalé. Prístup k jednotlivým knihám v bunke je totiž programovo náročnejší ako k celému riadku. Ak by sme do takejto tabuľky pridali ešte stĺpce dátum požičania a vrátenia, celá tabuľka by sa skomplikovala, pretože knihy môžu byť požičané a tiež vrátené v rôzne dni.

Mohli by sme tabuľku navrhnuť tak, že namiesto jedného stĺpca „požičané knihy“ dáme do tabuľky stĺpce kniha1, kniha2, kniha3. Museli by sme ale potom pridať aj stĺpce „dátum požičania1“, „dátum požičania2“, „dátum požičania3“. A obdobne aj pre dátumy vrátenia. Okrem toho pri akom počte kníh sa zastaví? Ak ich pridáme veľa bude to mrhanie miestom a ak málo nemusí to niekedy stačiť.

Tab. 1.2 už spĺňa 1NF. Navyše vieme rýchlo zistiť koľko kníh má knižnica požičaných - je to počet riadkov tabuľky. Do tejto tabuľky sme už pridali stĺpec *Dátum výpožičky* a aj primárny kľúč *ID výpožičky*, ktorý jednoznačne určuje každý riadok tabuľky.

Tab. 1.2. Tabuľka vyhovujúca 1NF

| ID výpožičky | Čitateľ | Požičaná kniha | Dátum výpožičky |
|---------------------|----------------|-----------------------|------------------------|
| 3 | Nováková | Nadácia | 12.3.2009 |
| 7 | Labuda | Súmrak | 17.5.2009 |
| 9 | Nováková | Zatmenie | 12.3.2009 |
| 11 | Nováková | Access 2007 | 7.8.2009 |
| 15 | Labuda | Nov | 17.5.2009 |

1.2.2. Druhá normálna forma - 2NF

Tabuľka spĺňa podmienku 2NF, ak je v 1NF a žiaden neklúčový stĺpec nie je závislý na časti primárneho kľúča. Inak povedané každý neklúčový stĺpec musí závisieť na celom kľúči, nielen na jeho časti.

Predstavme si tabuľku (tab. 1.3.) na evidenciu *aktuálne* požičaných kníh. Za predpokladu, že

1. stĺpec *Kniha* jednoznačne identifikuje požičanú knihu a stĺpec *Čitateľ* čitateľa (t. j. dokážeme rozlíšiť medzi dvoma knihami s rovnakým názvom alebo dvoch čitateľov s rovnakým menom) a
2. po vrátení knihy, riadok s danou výpožičkou zmažeme,

tak môžeme v takejto tabuľke spojením týchto stĺpcov *Kniha* a *Čitateľ* jednoznačne určiť každý riadok¹. Teda ich spojením vznikne primárny kľúč.

Tab. 1.3. zachytáva takýto návrh tabuľky. Táto tabuľka spĺňa kritéria 1NF ale nespĺňa kritéria 2NF. Ak je čitateľ jednoznačne určený stĺpcom *Čitateľ* (to bol predpoklad), tak je týmto stĺpcom určená aj adresa čitateľa. Adresa ale nezávisí (nemá nič spoločné) od stĺpca *Kniha*. Teda neklúčový stĺpec *Adresa čitateľa* závisí iba od časti primárneho kľúča.

Normalizácia do 2NF spočíva v tejto tabuľke v presunutí atribútu *Adresa čitateľa* do tabuľky, v ktorej sú údaje o čitateľovi. V tomto prípade je presun adresy logický (adresa sa týka čitateľa, preto bude v tabuľke, kde máme o ňom údaje) ale nie vždy je to na prvý pohľad tak viditeľné. Táto úprava je podobná ako úprava do tretej normálnej formy, ktorá je popísaná v nasledujúcej kapitole. V nej sú aj podrobnejšie popísané nevýhody nenormalizovanej tabuľky a znázornené rozdelenie a zhrnuté výhody takýchto úprav.

Tab. 1.3. Tabuľka evidencia aktuálne požičaných kníh nevyhovujúca 2NF

| Kniha | Čitateľ | Adresa čitateľa | Dátum výpožičky |
|--------------------|----------------|------------------------|------------------------|
| Čo bolí to prebolí | Novák | Žilina | 1.1.2010 |

¹ Prečo sú nutné tieto dve podmienky? Čo by sa mohlo stať ak jedna alebo druhá nie je splnená?

| | | | |
|-------------|---------|----------|----------|
| Súmrak | Novák | Žilina | 1.1.2010 |
| Access 2007 | Šottová | Martin | 3.1.2010 |
| Lipohrádok | Rusnák | Snežnica | 3.1.2010 |

1.2.3. Tretia normálna forma - 3NF

Tabuľka spĺňa podmienku 3NF, ak je v 2NF a všetky neklúčové stĺpce sú navzájom nezávislé.

Tab. 1.4 znázorňuje zoznam kníh evidujúci názov knihy, jej autora a štátnu príslušnosť autora. Lahko si všimneme, že štátna príslušnosť pri Peterajovi je „slovenská“ a pri Mayerovej USA. Vždy. Ak budeme mať od nejakého autora 100 kníh, tak v takejto tabuľke by sme stokrát k autorovmu menu pripísali rovnakú štátnu príslušnosť.

Tab. 1.4. Tabuľka nevyhovujúca 3NF

| <u>ID knihy</u> | Názov knihy | Autor | Štátna príslušnosť autora |
|-----------------|--------------------|----------|---------------------------|
| 2 | Čo bolí to prebolí | Peteraj | slovenská |
| 7 | Súmrak | Meyerová | USA |
| 8 | Zatmenie | Meyerová | USA |
| 17 | Lipohrádok | Peteraj | slovenská |
| 5 | Nov | Meyerová | USA |

Takýto spôsob organizovania údajov zvyšuje mieru redundancie a môže viesť k chybám. Knihy do databázy vkladajú viacerí ľudia a jeden Meyerovej uvedie USA, druhý „amer.“, ďalší US a podobne. Čo ale robiť ak by sme hľadali všetky jej knihy ale existovali by aj autorky s rovnakým menom z Nemecka a Izraela? Mohli by sme hľadať americkú Meyerovú na základe jej štátnej príslušnosti. Čo ale hľadať? US, USA alebo „amer.“? To by sa ešte dalo vyriešiť (s trochou práce navyše) postupným alebo spoločným hľadaním². Horšie by bolo, ak by sme nevedeli, že niekto zadal autorke pri niektorej knihe štátnu príslušnosť „am.“

Ďalší problém by mohol nastať, ak by sa niektorý štát (napr. USA) rozdelil³ alebo premenoval a my by sme museli pri každej Meyerovej knihe v našej tabuľke prepisovať USA na Arizona.

Tab. 1.5. Tabuľka autorov

| <u>ID Autora</u> | Autor | Štátna príslušnosť autora |
|------------------|----------|---------------------------|
| 5 | Peteraj | slovenská |
| 8 | Meyerová | USA |
| 17 | Dumas | francúzska |

Tab. 1.6. Tab. 1.4 po úprave spĺňa 3NF

| <u>ID Knihy</u> | Názov knihy | ID_Autora |
|-----------------|--------------------|-----------|
| 2 | Čo bolí to prebolí | 5 |
| 7 | Súmrak | 8 |
| 8 | Zatmenie | 8 |

² Naučíme sa to v kapitole 3. Dotazy

³ Áno, pri návrhu databáz sa od tvorcu čaká aby myslel na všetky možnosti. Pri USA asi takáto nepríjemnosť nehrozí ale v našej tabuľke sa to jednému z autorov prihodilo.

| | | |
|----|------------|---|
| 17 | Lipohrádok | 5 |
| 5 | Nov | 8 |

Našťastie sa dá zbytočnému zvyšovaniu redundancie a práce vyhnúť a tak sa zároveň vyhnúť aj potenciálnym problémom. Stačí ak tabuľky, kde neklúčový stĺpec (príslušnosť) závisí na inom/iných neklúčových stĺpcoch (autor) pri návrhu databázy rozdelíme tak ako je to znázornené na dvoch tabuľkách vyššie.

Tab. 1.6 obsahuje v treťom stĺpci namiesto mien čísla, netreba sa ale obávať neprehľadnosti. Databázový systém, vďaka prepojeniu tabuliek cez kľúč *ID_Autora*, môže zobrazit' namiesto čísel mená a prípadne aj štátnu príslušnosť.

Je dobré si všimnúť, že štátnu príslušnosť stačí každému autorovi zapísať (na rozdiel od pôvodnej tabuľky) iba raz – a to vtedy, keď autora zaraďujeme do tabuľky autorov.

Aj keď normalizáciou vytvárame optimálnu štruktúru tabuliek, v niektorých prípadoch sa tabuľky denormalizujú s cieľom zvýšenia výkonu databázy. Z rovnakého dôvodu sa bežne nepoužívajú 4. a 5. normalizačné pravidlá.

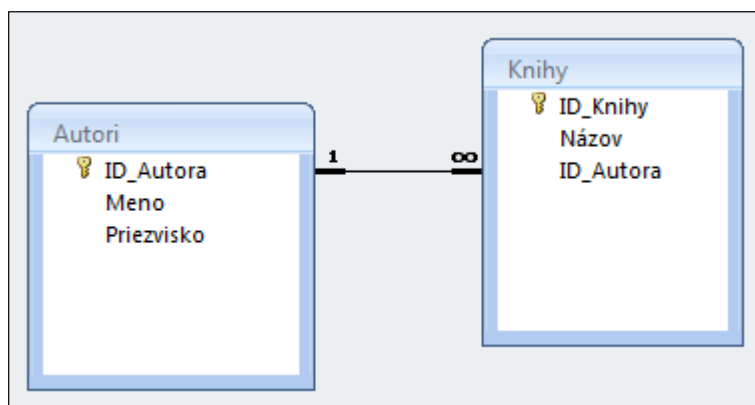
1.2.4. Vzťahy medzi tabuľkami

Tab. 1.6 a aj tab. 1.5 obsahujú stĺpec *ID_Autora*, resp. *ID_Autora*. Hodnoty v týchto stĺpcoch sú v niečom odlišné.

Hodnoty v stĺpci *ID_Autora* sú jedinečné, to by malo byť jasné - je to primárny kľúč (aj to by malo byť jasné – názov je podčiarknutý). A je to aj logické, každého autora evidujeme práve **jedenkrát**. Hodnoty v stĺpci *ID_Autora* v nižšej tabuľke sa môžu opakovať, pretože od niektorých autorov môžeme mať **viac** knižiek. Spomínané tabuľky sú prepojené kľúčom *ID_Autora* vzťahom, ktorý sa nazýva **1:N** (one to many). Access označuje tento vzťah **1-∞**. Zápis symbolizuje to, čo už je napísané v tomto odseku, zhrnutie:

- Prepájame dve tabuľky. Prepájame ich pomocou kľúča.
- V tabuľke, kde sa jedná o primárny kľúč sa jeho hodnoty (samozrejme) neopakujú.
- V druhej tabuľke sú iba hodnoty, ktoré sa nachádzajú v danom stĺpci v prvej tabuľke a žiadne iné. (Prečo?) Môžu sa opakovať 0 alebo viackrát.

Na obrázku nižšie je znázornený vzťah 1:N tabuliek *Autori* a *Knihy*. Trocha iný prípad by nastal, ak by sme z tabuľky *Knihy* chceli vybrať najviac jednu knihu od každého autora. Dostali by sme vzťah N:1, ktorý je otočením vzťahu 1:N.

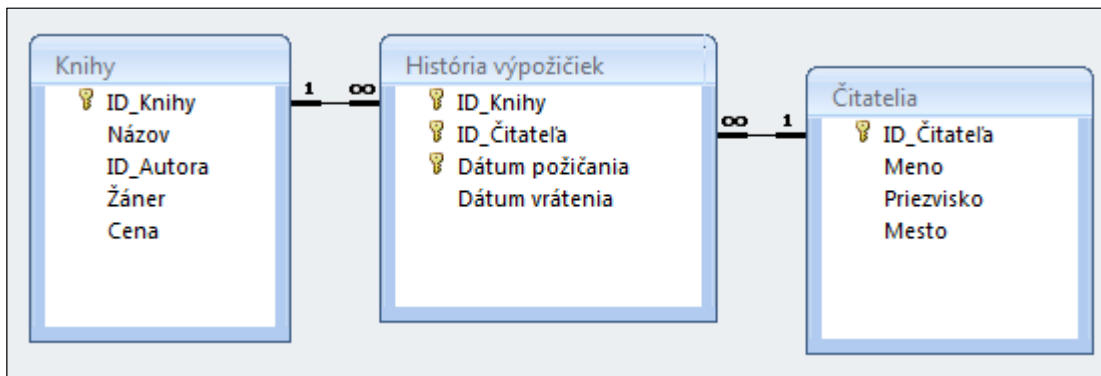


Obr. 1.2 Vzťah 1:N

Ak máme spoločný kľúč, pomocou ktorého môžeme niektoré riadky jednej tabuľky priradiť k viacerým riadkom druhej tabuľky a podobne z druhej tabuľky možno niektoré riadky priradiť k viacerým riadkom prvej tabuľky ide o vzťah **M:N** (many to many, ∞ - ∞).

Takýto vzťah sa nedá doceliť pri existencii len dvoch tabuliek. Je nutná tretia, tzv. spojovacia tabuľka (v príklade na obr. 1.3. je to tabuľka *História výpožičiek*), v ktorej sa vzťah M:N rozloží na dva vzťahy 1:N a N:1. Tento proces sa nazýva dekompozícia vzťahu M:N.

Obr. 1.3 znázorňuje vzťah M:N. V tabuľke *História výpožičiek* sa môžu nachádzať niektoré knihy a aj niektorí čitatelia viackrát. V tejto tabuľke sme použili zložený primárny kľúč – výpožička je jednoznačne určená požičiavanou knihou, čitateľom ktorý si požičiava a dátumom (a časom) výpožičky⁴.



Obr. 1.3 Vzťah M:N

Ak by sa v dvoch spájaných tabuľkách mohla vyskytovať hodnota spoločného kľúča v každej tabuľke najviac jedenkrát, dostali by sme vzťah **1:1** (one to one). Tento vzťah nie je v databázach príliš častý. V našom knižničnom príklade by mohol nastať ak by sa knižnica rozhodla každému čitateľovi darovať jednu knihu.

To môže dopadnúť rôzne:

1. Ak je viac kníh a každý čitateľ dostane knihu, tak je každý čitateľ jednoznačne určený darovanou knihou teda patrí do tabuľky knihy. Budeme používať iba kľúč ID_Knihy. Do tabuľky knihy pridáme stĺpce aké sú v tabuľke čitateľ, ku každej knihe pridáme jej majiteľa. Tabuľku *Čitatelia* môžeme zmazať.
2. Ak je viac čitateľov a každá kniha je darovaná, urobíme to naopak, pretože každá kniha je určená svojím vlastníkom. Budeme používať iba kľúč ID_Čitateľa. Zmažeme tabuľku *Knihy*.
3. Ak je ich rovnako, dostaneme sa do stavu, že každý čitateľ bude mať knihu a každá kniha bude darovaná. Vtedy môžeme urobiť ľubovoľnú z predchádzajúcich možností.
4. Posledný prípad by nastal, ak by niektorí čitatelia knihu nedostali a zároveň by nejaké knihy ostali nepodarované. Aj v tomto prípade by sme mohli tabuľky spojiť. Istá komplikácia by ale nastala: čo s chýbajúcim ID_Knihy v prípade čitateľa bez knihy resp. čo s chýbajúcim ID_Čitateľa v prípade knihy bez čitateľa? To sa našťastie dá vhodným prečíslovaním alebo priamo zvolením nového kľúča pomerne ľahko zvládnuť.

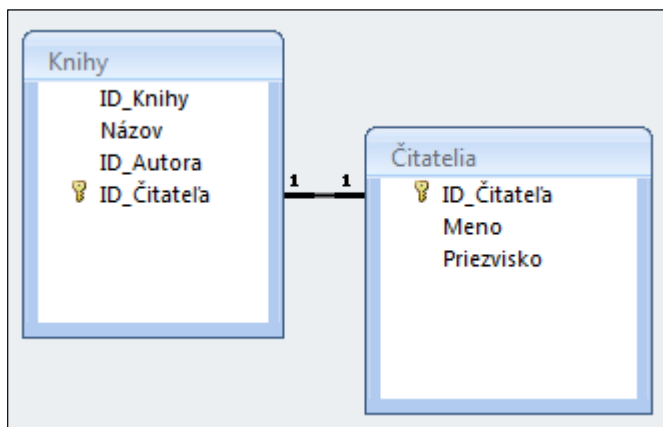
Vidíme, že dve tabuľky vo vzťahu 1:1 sa dajú nahradiť jednou novou tabuľkou. V prvom prípade bude v novej tabuľke toľko riadkov ako bolo v tabuľke *Knihy*. V druhom toľko ako v tabuľke *Čitatelia*. V treťom majú všetky tri rovnaký počet riadkov. V štvrtom prípade bude v novej tabuľke o niečo viac riadkov ako ich bolo vo väčšej z dvoch pôvodných.

Aj keď je iba vzťah 1:1 komplikovanejším vyjadrením jednej tabuľky, nejaké využitie predsa len tento vzťah má. Databázové systémy majú rôzne obmedzenia. Napríklad tabuľka v Access 2007 môže mať iba 255 stĺpcov. Ak potrebujeme stĺpcov v tabuľke napríklad 400, môžeme takúto tabuľku navrhnúť ako dve dvestostĺpcové (plus spoločný kľúč) tabuľky vo vzťahu 1:1. Bude sa to podobať na tretí prípad až na to, že ich nebudeme môcť spojiť.

Iným využitím je oddelenie niektorých stĺpcov do druhej tabuľky z bezpečnostných dôvodov (táto tabuľka má nastavené iné prístupové práva). Alebo preto, že sú iba na krátkodobé využitie a po ich využití ich možno naraz zmazať ako jednu tabuľku.

⁴ Prečo je dôležité evidovať dátum? A prečo aj čas? Kedy by sme čas nemuseli evidovať?

Na obr. 1.4 je znázornený vzťah 1:1 z bodu 3. Všimneme si, že v tabuľke *Knihy* už nie je primárny kľúč *ID_Knihy* ale *ID_Čitateľa*. Je to to isté ako keby sa všetky stĺpce z tabuľky *Knihy* presunuli do tabuľky *Čitatelia* (samozrejme aj so všetkými riadkami).



Obr. 1.4 Vzťah 1:1 – Ak by sme každému čitateľovi darovali práve jednu knihu a žiadna nezvyšila

1.3. Databáza knižnica

V tejto kapitole využijeme poznatky uvedené v predchádzajúcej kapitole a navrhne štruktúru našej pracovnej databázy, t. j. z akých tabuliek sa bude skladať a ako budú tieto tabuľky prepojené.

V učebnici budeme pracovať s databázou určenou na evidenciu výpožičiek kníh. Budeme preto potrebovať minimálne tri tabuľky: evidenciu kníh, čitateľov a samotných výpožičiek.

1.3.1. Evidencia kníh

Z predchádzajúcej kapitoly už vieme, že niektoré tabuľky je vhodné rozdeliť. V knižnici bude zrejme viacero kníh od toho istého autora, preto bude dobré vytvoriť zvlášť tabuľku *Autori*. V tejto tabuľke budeme evidovať meno a priezvisko autora. Pretože môžu existovať autori s rovnakým menom, každý autor bude jednoznačne identifikovaný primárnym kľúčom *ID_Autora*. Aby sme si ukázali možnosti MS Access budeme evidovať aj životopis a fotografiu autora. Nám bude takáto tabuľka stačiť, v skutočnej knižnici by mohli o každom autorovi evidovať ešte napríklad dátum narodenia a úmrtia, národnosť a pod.

Pri každej knihe budeme evidovať aj žáner. Podobne ako pri autoroch, aj žánre budú mať samostatnú tabuľku, v ktorej bude každý žáner určený primárnym kľúčom *ID_Žánra*.

Tabuľka *Knihy* bude obsahovať názov knihy, jej nákupnú cenu, jazyk a spomínané cudzie kľúče *ID_Autora* a *ID_Žánra*. Kvôli ukážke umožníme ukladať aj elektronickú verziu knihy. V reálnej knižnici by mohli požadovať napríklad aj ISBN, vydavateľa, rok vydania. Z nášho pohľadu sú to ale stĺpce, na ktorých sa nič nové nenaučíme a preto ich vynecháme.

1.3.2. Evidencia čitateľov

Pri čitateľoch budeme uvádzať meno a priezvisko. Podobne ako pri autoroch, *ID_Čitateľa* bude primárny kľúč potrebný na rozlíšenie čitateľov s rovnakým menom.

Okrem toho bude v tabuľke údaj o zaplatení ročného členského poplatku. Mohli by sme mať na to zvlášť tabuľku (a v skutočnej knižničnej databáze by to zrejme aj bolo tabuľkou zrealizované) ale podobné tabuľky budeme mať a radšej si ukážeme ďalšiu možnosť. Budeme evidovať iba hodnotu *Zaplatené* a *Nezaplatené*. Fungovať by to mohlo tak, že vždy na Nový rok zapíšeme všetkým *Nezaplatené* a keď príde čitateľ v novom roku prvýkrát do knižnice, tak mu po zaplatení zapíšeme *Zaplatené*. (Ak nepríde celý rok, tak služby knižnice nevyužíval a nemusí za ten rok platiť.)

V reálnej evidencii čitateľov by mohli byť ešte údaje ako adresa, email, dátum narodenia apod. V našej cvičnej databáze budú pre zachovanie jednoduchosti chýbať, pretože sa na nich aj tak nič nové nenaučíme.

1.3.3. Evidencia výpožičiek

Návrh evidencie kníh a čitateľov bol pomerne jednoduchý. Evidencia výpožičiek je o niečo zložitejšia. Hlavne preto lebo sa musíme rozhodnúť, ktorým zo spôsobom ju budeme vykonávať.

Niektoré spôsoby by mali byť na prvý pohľad nevhodné, napríklad:

- Mohli by sme požičané knihy zapisovať priamo do tabuľky *Čitatelia*. V takomto prípade by sme museli do tejto tabuľky pridať stĺpce *kniha1*, *kniha2*, až po maximálne povolený počet súčasne požičaných kníh. Okrem ohraničenia počtu požičaných kníh by to bolo nevhodné aj pri zisťovaní, ktoré knihy sú aktuálne požičané – museli by sme prezrieť záznam o každom čitateľovi.
- O čosi lepšie by bolo obrátiť predchádzajúci prípad a evidovať čitateľa v zázname knihy. Pridali by sme jeden stĺpec (*ID_Čitateľa*) a do neho by sme zapisovali aktuálneho „majiteľa“ knihy. Opätovne by bol problém pri zisťovaní, ktoré knihy sú požičané – museli by sme prezrieť záznam o každej knihe. Rovnaká situácia by nastala, ak by sme hľadali knižky, ktoré má požičané niektorý konkrétny čitateľ.

Oba spôsoby majú navyše spoločnú nevýhodu v tom, že nearchivujú kto si požičal ktorú knihu. Z dôvodu archivácie histórie výpožičiek vyplýva potreba novej tabuľky. Máme dve možnosti:

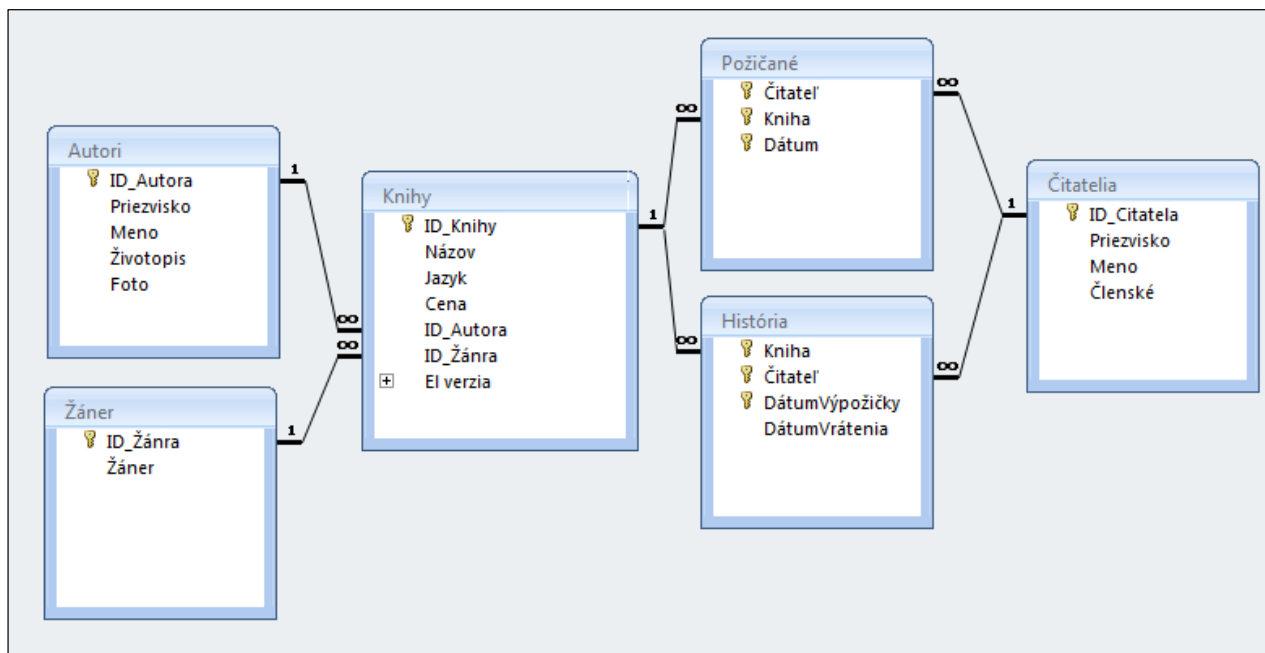
- Prvá je znázornená na obr. 1.3. Jeden riadok = jedna výpožička. Do tabuľky budeme zapisovať ID čitateľa a knihy a dátumy požičania a vrátenia. Ak bude dátum vrátenia chýbať (bude *Null*), bude to znamenať, že kniha je ešte požičaná.
- Druhý spôsob by bol podobný ale používali by sa dve tabuľky. Výpožička by sa zaznamenala do tabuľky ako znázorňuje tab. 1.2 – teda bez dátumu vrátenia. Po vrátení by sa zaznačilo do druhej tabuľky – rovnakej ako je v predchádzajúcom bode a z prvej by sa riadok zmazal. Prvá tabuľka by teda zaznamenávala aktuálne požičané (a zatiaľ nevrátené) knihy. Druhá zasa iba ukončené výpožičky.

Oba spôsoby majú svoje výhody. V prvom je to jasné - vystačíme s jednou tabuľkou. Druhý spôsob je ale výhodnejší z dlhodobého hľadiska. Pri prevádzke knižnice používame históriu výpožičiek pomerne málo, iba keď sa robia nejaké štatistiky, čo býva tak raz za rok. O mnoho častejšie pracujeme s aktuálne požičanými knihami.

Ak by sme použili prvý spôsob, tak by sme pracovali s čoraz väčšou tabuľkou, v ktorej by sme museli hľadať pri vrátení ten správny riadok aby sme do neho zapísali dátum vrátenia.

Pri druhom spôsobe by bola tabuľka *Aktuálne požičané knihy* viac menej stále rovnako veľká. Druhá by sa síce zväčšovala ale do nej by sme záznamy iba pridávali (+štatistiky raz za rok) a to nie je také náročné ako vyhľadávanie.

V učebnici budeme používať spôsob s dvoma tabuľkami, (aktuálne) *Požičané* a *História*. Celá štruktúra databázy je znázornená na obr. 1.5.



Obr. 1.5. Prepojenie tabuliek v databáze knižnica

Pre vytvorenie databázy s takto poprepájanými tabuľkami potrebujeme:

- Prázdnu databázu. Postup jej vytvorenia je popísaný v nasledujúcej kapitole 1.4.
- Základné informácie o prostredí MS Access, ktoré sú uvedené v kapitolách 1.5 a 1.6.
- Tabuľky. Informácie pre vytvorenie tabuliek sú v kapitole 2.1.
- Prepojenie tabuliek, o ktorom pojednáva kapitola 2.2.

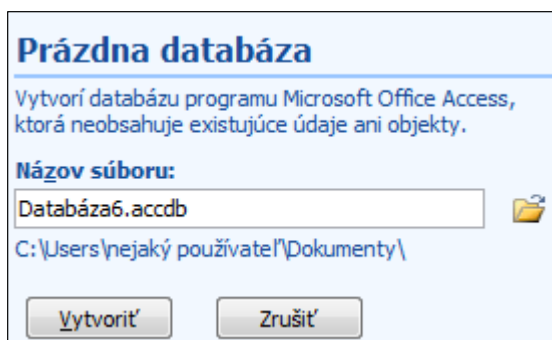
1.4. Vytvorenie databázy v Access 2007

Po spustení MS Access si môžeme vybrať z troch možností:

1. Vytvoriť novú (prázdnu) databázu.
2. Vytvoriť novú databázu z predefinovaných šablón.
3. Otvoriť existujúcu. Pre tento prípad je v pravej časti okna zoznam naposledy otvorených databáz. Ak chceme otvoriť novú alebo naopak dávnejšie nepoužívanú databázu klikneme na odkaz „ďalšie“.

Tieto možnosti sú priamo na úvodnej obrazovke alebo je možné kliknúť na tlačidlo Office nachádzajúce sa v hornom ľavom rohu okna a vybrať si z menu.





Obr. 1.6 Pomenovanie a umiestnenie databázy

Je **vhodné** novú databázu (tak ako pri každom novom súbore) pomenovať a prípadne určiť kliknutím na 📁, kde bude uložená⁵. V tomto prípade sa autor zahral na študenta, t. j. súbor nepomenoval a ani sa nepozrel, kde ho ukladá. Ale ak je to študent vnímavý a prečítal si poznámku pod čiarou, tak vie, kde bol súbor automaticky uložený a môže si ho odtiaľ skopírovať na kľúč alebo presunúť do správneho priečinka. V preddefinovanom priečinku bude síce súborov veľa (bohužiaľ) ale ten posledne uložený bude mať pravdepodobne najvyššie číslo, v našom prípade *Databáza6.accdb*⁶.

V prípade, že už máte otvorenú databázu a pokúsite sa otvoriť ďalšiu, aktuálne otvorená databáza sa zavrie. Access dovoľuje totiž pracovať iba s jednou databázou súčasne. Netreba sa ale báť – v jednej databáze môžu byť [stovky](#) tabuliek.

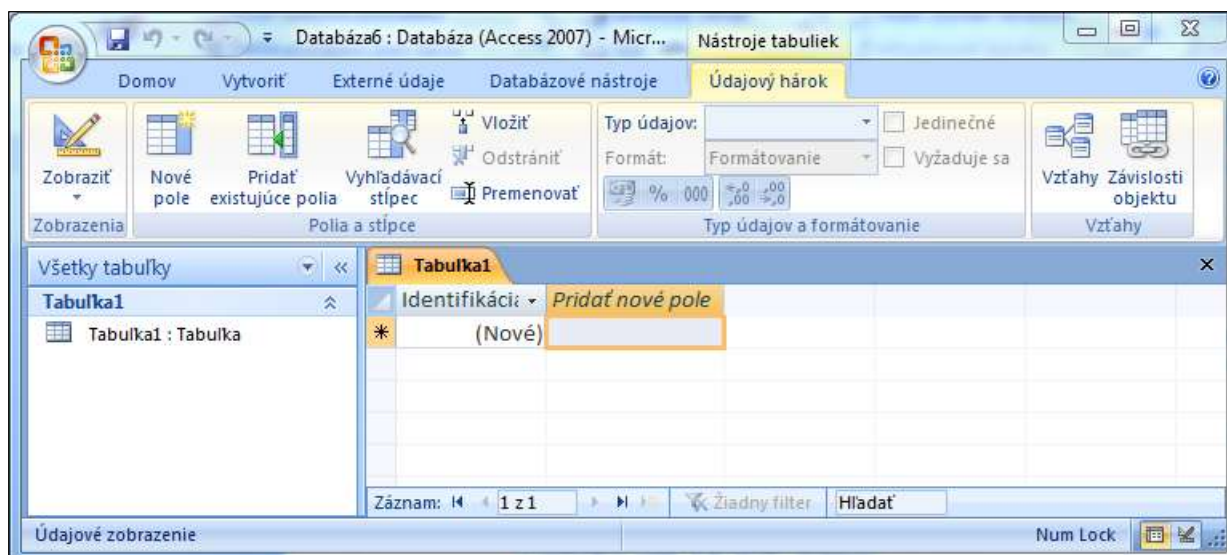
1.5. Objekty v Access 2007

Databáza v Access 2007 sa môže skladať (a väčšinou aj skladá) z viacerých typov objektov. Sú to:

- Už spomínané **tabuľky**, ktoré obsahujú samotné dáta. Je dobré zdôrazniť, že v databázovej tabuľke sa na rozdiel tabuliek tabuľkových kalkulátorov (napríklad v MS Excel) nedajú vykonávať výpočty, na to slúžia dotazy!
- **Dotazy**. Slúžia na vykonávanie takých činností ako sú filtrovanie, spájanie viacerých tabuliek, odstraňovanie údajov z tabuliek alebo ich aktualizovanie, triedenie, sumarizovanie a mnoho ďalších.
- **Formuláre**. Umožňujú pridávať a editovať dáta z jednej alebo viacerých tabuliek vo vlastnom používateľskom prostredí. Okrem zmeny vzhľadu môžeme určiť, ktoré polia povolíme editovať.
- **Zostavy**. Zostavy zobrazujú dáta (často získané z viacerých tabuliek pomocou dotazov) v podobe vhodnej na tlač.
- **Makrá**. Podobne ako vo Worde aj v programe Access možno automatizovať časté operácie pomocou makier. Ale nie zaznamenávaním postupnosti stlačených klávesov a kliknutí. V Accesse je makro postupnosť akcií, ktoré si možno vybrať zo zoznamu. Napríklad: otvorenie formulára alebo dotazu. Po výbere akcie je nutné určiť prípadné argumenty ako sú názov otváraného formulára alebo dotazu. Po uložení môžeme makro priradiť k tlačidlu a definovanú postupnosť krokov kedykoľvek zopakovať.
- **Moduly**. Podobne ako makrá aj moduly môžu vylepšiť používanie databáz. Balík Microsoft Office používa programovací jazyk VBA (Visual Basic for Applications). V moduloch sú uložené programy v tomto jazyku. Programovanie ale nie je náplňou tejto učebnice, preto sa modulom venovať nebudeme.

⁵ Po stlačení tlačidla Office, *Možnosti programu Access* je možné zmeniť preddefinovaný priečinok.

⁶ Na ďalšie cvičenie to platí už nemusí: Pribudnú súbory a čo je nepríjemnejšie, niekto určite zmení preddefinovaný priečinok.

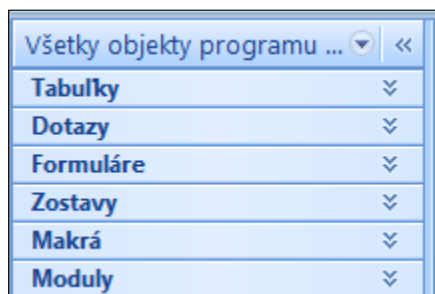


Obr. 1.7 Pracovná plocha po vytvorení novej databázy

1.6. Používateľské rozhranie



Používateľské rozhranie je rozdelené na tri hlavné časti:


Pás s nástrojmi. Nástroje na páse je možno prepínať klikaním na karty. Na obr. 1.7 je označená kontextová⁷ karta *Údajový hárok*. Aktuálne zobrazené nástroje sú rozdelené do skupín. Na obrázku sú to skupiny *Zobrazenia*, *Polia a stĺpce* atď. Zobrazenie jednotlivých nástrojov sa prispôsobuje šírke okna, preto sa Vám určite stane, že na notebooku s malým monitorom bude vyzeráť pás nástrojov inak ako na širokom 24 palcovom monitore. Pás obsahuje aj panel s nástrojmi *Rýchly prístup*, na ktorý je možné pridať často používané príkazy.



Obr. 1.8 Navigačná tabla

Pod pásom s nástrojmi vľavo je **Navigačná tabla**. Na obr. 1.7 je to časť s názvom *Všetky tabuľky*.

Možno ju ukryť alebo zobrazit' stlačením F11 alebo tlačidla  resp. . Detail tejto tably je na obr. 1.8. Pomocou navigačnej tably máme prístup ku všetkým objektom databázy, ktoré sme spomínali v kapitole 1.5. Objekty v Access 2007. Samozrejme musia sa v databáze nachádzať. V prípade, že sme práve teraz vytvorili novú databázu, tak je na nej iba jedna tabuľka ako je vidno na obr. 1.7. Ak by sme tabuľku zatvorili a neuložili, nebol by na table žiaden objekt.

Kliknutím na  v záhlaví navigačnej tably môžeme nastaviť ako sa budú objekty zobrazovať. Štandardne sa zoskupujú podľa typu objektu (*Typ objektu*), tak je to aj na obr. 1.8. Môžeme ale nastaviť

⁷ Kontextová karta znamená, že sa zobrazí alebo zmení závislosti od vykonávanej činnosti. Podobne existuje aj kontextové popup menu. Klikaním na rôzne miesta pravým tlačidlom myši sa zobrazuje rôzne menu. A každý snád' pozná kontextovú pomoc – ukážeme na niečo myšou a za chvíľu sa zobrazí informácia na čo to vlastne ukazujeme alebo čo sa stane ak na ten objekt klikneme.

zokupovanie podľa toho ako objekty spolu súvisia alebo podľa dátumu a možno si vybrať aj vlastný spôsob zokupovania. V takom prípade pri každom objekte musíme určiť (kliknutím pravým tlačidlom myši na daný objekt) do akej skupiny patrí. Okrem toho môžeme určiť či sa zobrazujú všetky objekty alebo len jeden typ. Kliknutím pravým tlačidlom myši na záhlavie alebo prázdne miesto môžeme navyše objekty triediť alebo hľadať.

V tretej časti sa zobrazujú objekty. Môžu sa zobrazovať ako maximalizované okná s kartami (tiež sa nazývajú ušká, záložky) klikaním, na ktoré môžeme medzi nimi prepínať. Alebo je možné nastaviť zobrazovanie prekrývajúcich sa okien: *Tlačidlo Office*, *Možnosti programu Access / Aktuálna databáza / Možnosti okna dokumentu*. Nastavenie sa prejaví až po zavretí a znovuotvorení databázy.

Okrem týchto troch častí sa v spodnej časti zobrazuje stavový riadok. V jeho ľavej časti sa často zobrazuje kontextová pomoc a v pravej si budeme často vyberať spôsob zobrazenia objektu.

2. Tabuľky

V tejto kapitole je popísaný postup ako vytvoriť jednoduchú tabuľku a naplniť ju údajmi. Tiež sa tu dozvieme ako nastaviť tabuľku tak, aby sme znížili pravdepodobnosť vloženia zlého údaja. Napríklad aby nebolo možné omylom vložiť namiesto požadovaného dátumu alebo čísla text. Prípadne ako používateľovi uľahčiť zadávanie, aby nemusel vypisovať napríklad žáner knihy, ale aby si ho mohol vybrať z ponuky.

Pretože tabuľka je (základnou) časťou databázy, musíme najskôr otvoriť existujúcu alebo vytvoriť novú databázu. To je popísané v kapitole 1.4.


2.1. Vytvorenie prvej tabuľky

Obr. 1.7. zobrazuje situáciu po vytvorení novej databázy. Jedna tabuľka už je pripravená na úpravu. Do rovnakého stavu sa dostaneme ak otvoríme prázdnu databázu a klikneme na karte *Vytvoriť* v skupine *Tabuľky* na položku *Tabuľka*. Takto môžeme pridať do databázy potrebný počet tabuliek.



Každá novovytvorená tabuľka má pole (stĺpec) *Identifikácia*. Je to automaticky vytvorený primárny kľúč - dôvod jeho existencie bol vysvetlený v kapitole 1.1.

Polia môžeme do tabuľky pridávať (ale aj uberať či premenovávať) v dvoch režimoch. V **údajovom** alebo **návrhovom zobrazení**. Medzi týmito zobrazeniami tabuľky môžeme prepínať:



- Tlačidlami  v stavovom riadku v pravej dolnej časti okna. Na obr. 1.7 vidno obe tlačidlá napravo od nápisu *Num Lock*. Neskôr tam pribudnú ďalšie tlačidlá.



- Na kontextovej karte *Údajový hárok* kliknutím na tlačidlo *Zobrazit*  alebo . Obrázok sa mení podľa toho, ktoré zobrazenie bude po kliknutí aktívne. Všimnite si, že tlačidlo je zložené z dvoch častí (hornej a dolnej). To má dobrý dôvod. Uvidíme, že niekedy bude možností zobrazenia viac. V takom prípade je pohodlnejšie kliknúť na spodnú polovicu tlačidla a vybrať si priamo požadované zobrazenie.
- Kliknutím pravým tlačidlom myši na záložku tabuľky (na obr. 1.7 je to žltohnedo podfarbené slovo *Tabuľka1*) a výberom z kontextového menu. Okrem iného sú tu aj často používané možnosti *Zavrieť* a *Zavrieť všetko*.
- V navigačnej table kliknutím pravým tlačidlom myši na názov tabuľky. Možnosť *Otvoriť* je totožná s dvojklikom a znamená údajový režim.

Pri tvorbe tabuľky musíme vedieť, aké údaje budeme v tabuľke do jednotlivých stĺpcov ukladať, t. j. aké polia je potrebné do tabuľky pridať. Polia môžu byť rôzneho typu. MS Access pozná nasledujúce typy údajov:

- **Text** - Polia typu text môžu obsahovať akýkoľvek znak (aj číslice). Do textových polí sa ukládajú aj PSČ alebo telefónne číslo. Tieto polia majú síce v názve slovo „číslo“ ale nie sú to v skutočnosti čísla. Nemá totiž význam s nimi robiť matematické operácie. (Aké je priemerné telefónne číslo v našej skupine?) Maximálna dĺžka textu môže byť 255 znakov. V našich tabuľkách nastavíme tento typ pre polia *Meno*, *Priezvisko*, *Názov knihy*, a pod.
- **Memo** - Typ údajov určený pre dlhé texty. Ručným vkladaním možno vložiť až 65 535 znakov. A iba toľko zobrazuje aj z dlhších textov. Programom možno vložiť do poľa typu memo až miliardu znakov. Pole *Životopis* v tabuľke *Autori* je typu memo.
- **Number** a **Currency** - Obsahom poľa môže byť iba číslo. Currency = mena, číselný údaj prispôsobený pre peniaze. V tabuľke *Knihy* je pole *Cena* typu currency.
- **Date/Time** - Dátum a čas. Budeme používať pre dátum požičania a vrátenia knihy.

- **AutoNumber** - Automatické číslo. Access automaticky vyplňa toto pole (používateľ nemôže), pričom dbá aby sa v celej tabuľke v danom stĺpci číslo neopakovalo. Tento typ je používaný pre primárne kľúče.
- **Yes/No** - Áno/Nie. Dvojpohový prepínač On/Off. Zobrazuje sa ako zaškrtnuté políčko (checkbox). Tento typ využívame pri poli *Členské* v tabuľke *Čitatelia*.
- **OLE Object** - Objekt OLE. Umožní vložiť do stĺpca objekt (napríklad dokument Wordu, zvuk, obrázok apod.), podporujúci metódu OLE. V tabuľke *Autori* uložíme pomocou tohto typu fotku autora do poľa *Foto*.
- **Hyperlink** - Hypertextový odkaz. Mohli by sme využiť na uloženie odkazu na web autora.
- **Attachment** - Príloha. Možno pripojiť ľubovoľný typ súboru, podobne ako príloha v emailoch. Tento typ údajov je pružnejší ako typ OLE Object. V starších verziách Accessu sa nenachádzal. Tento typ má v tabuľke *Knihy* pole *El verzia*, do ktorého môžeme ukladať napríklad danú knihu vo formáte pdf.
- **Lookup Wizard** - Sprievodca vyhľadávaním. Nie je to typ údajov ale umožňuje v poli zobrazovať údaje na základe prepojenia s inou tabuľkou a pod. Prepojenia sú znázornené na obr. 1.5 a postup ich vytvorenia je vysvetlený v kapitole 2.2.

Podrobnejší popis jednotlivých typov údajov aj s ich vlastnosťami je v kapitole 2.1.2.

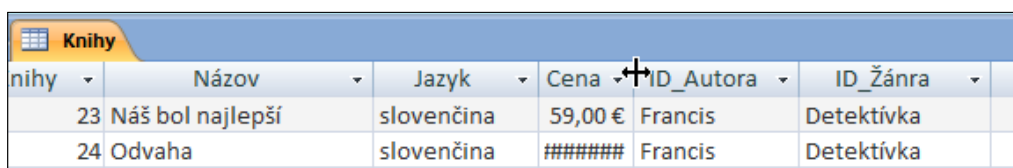
2.1.1. Údajové zobrazenie pri tvorbe tabuľky

Pre prvú tabuľku bude vzorom tabuľka pre evidenciu kníh na strane 3. Obr. 1.7 znázorňuje tabuľku v údajovom zobrazení. Budeme teda musieť pridať stĺpce *Názov*, *Cena*, *Rok vydania*, *Autor* a *Žáner*. Prvým krokom býva najčastejšie premenovanie stĺpca. Nový názov môže mať maximálne 64 znakov⁸. Môžeme to urobiť tromi spôsobmi:

- Po dvojkliku na záhlavie posledného stĺpca (na text *Pridať nové pole*) môžeme napísať názov poľa, v našom prípade *Názov*.
- Druhou možnosťou je na záhlavie kliknúť pravým tlačidlom myši a z kontextového menu vybrať voľbu *Premenovať*.
- Treťou možnosťou je kliknúť kdekoľvek do premenovávaného stĺpca a v karte *Údajový hárok*, skupine *Polia a stĺpce* kliknúť položku *Premenovať*.

Po pomenovaní stĺpca *Pridať nové pole* pribudne nový „bezmenný“ stĺpec, ktorý možno opäť pomenovať a tak vytvoriť ďalší stĺpec. Takto môžeme postupne vytvoriť všetky požadované stĺpce. Pre každý stĺpec je dobré určiť vhodný typ údajov pomocou voľby *Typ údajov* na v karte *Údajový hárok*, skupine *Typ údajov a formátovanie*. Zároveň môžeme nastaviť aj formátovanie, ktoré nájdeme v tej istej skupine príkazov.

Nové pole môžeme pridať aj voľbou *Nové pole* v skupine *Polia a stĺpce* na karte *Údajový hárok*. Access v takom prípade ponúkne najčastejšie používané polia aj s nastaveným typom údajov.



| Knihy | Názov | Jazyk | Cena | ID_Autora | ID_Žánra |
|-------|------------------|------------|---------|-----------|------------|
| 23 | Náš bol najlepší | slovenčina | 59,00 € | Francis | Detektívka |
| 24 | Odvaha | slovenčina | ##### | Francis | Detektívka |

Obr. 2.1 Automatické prispôsobenie šírky stĺpca

O mnoho dôležitejšie sú voľby napravo: *Pridať existujúce polia* a *Vyhľadávací stĺpec*. Po kliknutí na *Pridať existujúce polia* sa zobrazí *Zoznam polí*, z ktorého si môžeme vybrať polia z už vytvorených tabuliek v našej databáze. Po výbere poľa sa spustí sprievodca vyhľadávaním, ktorý je popísaný v kapitole 2.2. Sprievodca vyhľadávaním sa spustí aj po kliknutí na príkaz *Vyhľadávací stĺpec*.

⁸ V nepravdepodobnom prípade, ak potrebujeme dlhší názov, môžeme to vyriešiť pomocou vlastnosti popis, uvedenej v kapitole 2.1.2.

V tabuľke môžeme meniť poradie stĺpcov kliknutím na záhlavie, podržaním a presunutím doľava či doprava. Tabuľka v údajovom zobrazení umožňuje vkladať údaje. To môžeme urobiť pomocou klávesnice, ale môžeme napríklad vložiť (*Prilepiť*) aj celý stĺpec z programu MS Excel. Rovnako ak v MS Excel môžeme meniť šírku stĺpcov alebo výšku riadkov. Na obr. 2.1 je znázornená situácia, keď sa namiesto ceny zobrazuje „#####“. Je to preto lebo číslo je väčšie ako šírka stĺpca. Dvojklikom v záhlaví na hranicu stĺpcov *Cena* a *ID_Autora* (keď sa zmení kurzor tak ako je na obrázku) nastavíme šírku stĺpca *Cena* tak, aby sa do neho zmestila každá hodnota v stĺpci.

| Názov poľa | Typ údajov |
|---------------|------------|
| Identifikácia | AutoNumber |
| Názov | Text |
| Cena | Number |
| Rok vydania | Number |
| Autor | Text |
| Žáner | Text |
| datum | Text |

| Všeobecné | | Vyhľadávanie | |
|------------------------|--------------|--------------|--|
| Veľkosť poľa | 255 | | |
| Formát | | | |
| Vstupná maska | | | |
| Popis | | | |
| Predvolená hodnota | | | |
| Overovacie pravidlo | | | |
| Overovací text | | | |
| Povinné | Nie | | |
| Povoliť nulovú dĺžku | Áno | | |
| Indexované | Nie | | |
| Kompresia kódu Unicode | Áno | | |
| Režim IME | Bez kontroly | | |
| Režim vety IME | Žiadne | | |
| Inteligentné značky | | | |

Obr. 2.2 Návrhové zobrazenie pri tvorbe tabuľky

Po ukončení vkladania údajov môžeme tabuľku zatvoriť. Pri zatvorení tabuľky (napr.: kliknutím pravým tlačidlom myši na záložku tabuľky...) je ponúknutá možnosť uložiť tabuľku a v prípade, že sme ju ešte nepomenovali, tak ju môžeme premenovať. Opätovne ju môžeme otvoriť dvojklikom na jej názov v navigačnej table. Ďalšie možnosti pri práci s tabuľkou sú popísané v kapitole 2.3.

2.1.2. Návrhové zobrazenie pri tvorbe tabuľky

Obr. 2.2 znázorňuje tvorbu a úpravu tabuľky v návrhovom zobrazení. Na rozdiel od údajového zobrazenia pri návrhovom zobrazení nie je možné vkladať údaje. Na druhú stranu, máme väčšie možnosti pri výbere typu údajov a vlastností jednotlivých polí.

Návrhové zobrazenie pri úprave tabuľky sa skladá z dvoch častí. Horná časť slúži na pomenovanie stĺpcov, určenie ich typu. V treťom stĺpci môžeme určiť popis, ktorý sa bude zobrazovať pri úprave poľa v stavovom riadku pri formulárovom zobrazení. V dolnej časti, ktorá sa mení podľa typu údajov, môžeme presnejšie určiť, čo sa má v stĺpci nachádzať, koľko miesta pre údaj rezervujeme, ako sa bude údaj zobrazovať a mnoho ďalších vlastností. Táto časť sa skladá z dvoch kariet *Všeobecné*, ktorej obsah je predmetom tejto kapitoly a *Vyhľadávanie*, ktorá je popísaná v kapitole 2.2.

Na strane 16 sú základné informácie o typoch údajov. V nasledujúcich odsekoch sú popísané ich vlastnosti. Pretože niektoré vlastnosti majú rovnaké možnosti pre rôzne typy údajov, tak ich popíšeme na type text a pri ostatných typoch ich vynecháme.

Text

Text je predvoleným typom údajov, to znamená, že ak neurčíme typ údajov, bude stĺpec automaticky textový. To možno zmeniť v *Možnosti programu Access* v časti *Návrhári objektov*. Možnosti nastavenia textového údajov sú zobrazené v spodnej časti obr. 2.2.

Veľkosť poľa. Maximálna dĺžka textu, ktorú bude možno zadať a uložiť do tabuľky. Access rezervuje toto miesto pre každý údaj v danom stĺpci. Dĺžku preto treba nastaviť tak, aby sa zbytočne nemrhalo miestom, ale zároveň aby sa nestalo, že sa niektoré dáta nezместia. Predvolená veľkosť je 255 znakov, t. j. najväčšia povolená dĺžka textu. Predvolenú veľkosť poľa možno zmeniť na rovnakom mieste ako predvolený údaj.

Formát. Umožní vybrať ako sa budú dáta zobrazovať. Pri formátovaní textového poľa môžeme vo formátovacom reťazci použiť tieto formátovania:

- <, riadiaci znak, celý text bude zobrazený malými písmenami.
- >, riadiaci znak, celý text bude zobrazený veľkými písmenami.
- @, zástupný symbol. Namiesto jedného tohto znaku sa zobrazí jeden znak z obsahu poľa. Nahradzovanie znakov @ znakmi z poľa sa vykonáva odzadu (sprava doľava), t. j. namiesto posledného @ sa zobrazí posledný znak poľa, namiesto predposledného @ sa zobrazí predposledný znak poľa atď. Ak je vo formátovacom reťazci viac znakov @ ako je znakov v poli, tak sa neobsadené @ sa zobrazia ako medzery. Ak je viac znakov v poli, tak sa tieto zobrazia normálne. Ak zadáme @@@@ a údaj sa skladá iba z troch znakov, napríklad je to slovo „rok“, tak sa pred slovo vložia dve medzery – „, rok“.
- &, zástupný symbol. Podobný ako @, namiesto jedného tohto znaku sa zobrazí jeden znak z obsahu poľa. Ale ak je vo formátovacom reťazci viac znakov & ako je znakov v poli, tak sa neobsadené & vynechajú (nenahradia sa medzerou).
- Literál, je znak alebo skupina znakov, ktorá sa zobrazí presne tak ako je uvedená vo formátovacom reťazci. Pretože niekedy môže dôjsť problému či ide o literál alebo formátovací znak, tak sa literál dáva do úvodzoviek. Napríklad formát "@@" znamená, že pri zobrazení hodnoty poľa sa pred posledný znak vložia dva zavináče. Ak vkladáme len jeden konštantný znak môžeme ho uviesť spätnou lomkou (\). Ak by sme v predchádzajúcom príklade chceli zobraziť pred posledným znakom len jeden znak @, tak by sme použili formátovací reťazec \@@. Formát \@ zobrazí pred posledným znakom medzeru. Formát &"ABC" pridá na koniec každého údajov text „, ABC“.
- !, riadiaci znak, priradzuje znaky z obsahu poľa k znakom @ alebo & zľava doprava a nie sprava doľava. Výkričník sa umiestňuje na začiatok reťazca. Napríklad !@|@ umiestni zavináč za prvý znak v poli, formát !"ABC "& zobrazí text „,ABC “ pred obsah poľa. Pre porovnanie si vyskúšajte použiť výkričník v príkladoch z predchádzajúceho bodu.
- Čiarka a oddeľovač tisícov. Používajú sa nastavenia Windows. U nás čiarka a medzera, v USA bodka a čiarka. 1 234,56_{sk} = 1,234.56_{us}.
- *, riadiaci znak, opakuje znak, ktorý sa nachádza za znakom hviezdička tak, aby sa vyplnila celá šírka stĺpca. Skúste si formáty *-@, @*-, prípadne ich „výkričníkové“ ekvivalenty.
- [farba], je možné použiť iba farby biela, čierna, modrá, žltá, zelená, purpurová a azúrová. Napríklad formát !@" T"*r"nie "[Purpurová], vloží za druhé písmeno slovo Tíííínie (pridá toľko „í“ aby sa vyplnila celá šírka stĺpca) a zobrazí dáta v stĺpci purpurovou farbou. V hranatých zátvorkách nesmie byť medzera. Farba funguje iba s formátovacím znakom, t. j. [Modrá] nefunguje ale [Modrá]@ už áno.


Textový formát sa môže skladať z dvoch častí oddelených bodkočiarkou. Druhá časť je určená pre polia prázdne alebo obsahujúci prázdny reťazec. Formát @[Zelená];"Nič nie je zadané"[Červená] zobrazí texty

zelenou a v prípade *Null* alebo prázdneho reťazca vypíše červenou farbou text „Nič nie je zadané“. Situácia je znázornená na obr. 2.3.

Vstupná maska. Určuje v akom tvare bude používateľ vkladať údaje. Pomocou masky môžeme zobrazit' pri vkladaní dáta inak ako sú zobrazované v tabuľke (to je nastavené vlastnosťou *Formát*). Maska sa často používa na zoskupenie čísiel pri PSC, dátume alebo telefónnom čísle. Zároveň ju využívame na donútenie používateľa, aby údaje vložil v požadovanom tvare.

Podobne ako pri formáte sa maska skladá zo zástupných symbolov, riadiacich znakov a literálov:

- 0, zástupný znak. Namiesto jedného tohto znaku musí používateľ zadať číslicu. Napríklad maska 0000 vyžaduje zadať presne štyri číslice. Ani viac ani menej.
- 9, zástupný znak. Namiesto jedného tohto znaku môže používateľ zadať číslicu. Napríklad maska 9999 umožní zadať jedno až štvorciferné číslo (ktoré je ale brané ako text – pole je textové).
- #, zástupný znak. Namiesto jedného tohto znaku môže používateľ zadať číslicu, medzeru, znamienko plus alebo mínus. Napríklad maska #000 umožní zadať celé čísla -999 až 9999, číslo 3 je ale nutné zadať ako 0003. Tu vidno výhodu ukladania niektorých „čísel“ ako text – pri číslach by úvodné nuly zmizli a to by mohlo robiť problémy pri „číslach“ rodných, smerovacích a telefónnych.
- L, zástupný znak. Namiesto jedného tohto znaku musí používateľ zadať jedno písmeno. Maska LLL vyžaduje zadať tri písmená.
- ?, zástupný znak. Namiesto jedného tohto znaku môže používateľ zadať jedno písmeno. Maska L0??? umožní zadať dvoj až päť znakový text, v ktorom bude na prvom mieste písmeno, na druhom číslicu a ostatných miestach (ak nejaké budú) písmena.
- A, zástupný znak. Namiesto jedného tohto znaku musí používateľ zadať jedno písmeno alebo číslicu.
- a, zástupný znak. Namiesto jedného tohto znaku môže používateľ zadať jedno písmeno alebo číslicu.
- &, zástupný znak. Namiesto jedného tohto znaku musí používateľ zadať znak (aj medzera je znak).
- C, zástupný znak. Namiesto jedného tohto znaku môže používateľ zadať jeden ľubovoľný znak.
- !, riadiaci znak. Vstupnú masku sa bude vypínať zľava doprava namiesto sprava doľava.
- > resp. <, riadiace znaky. Všetky nasledujúce znaky budú napísané veľkými resp. malými písmenami. Maskou >L<???? docielime, že sa text „ahoj“ alebo „aHoJ“ uloží ako „Ahoj“.
- \, nasledujúci znak je literál. Dvojnakový a dlhší literál dávame do úvodzoviek. Napríklad pre májové dátumy (v textovej forme) má zmysel maska 90,,. mája “0000.
- Heslo, namiesto vkladaných a aj zobrazovaných znakov sa zobrazujú *.

Vstupné masky pre bežné situácie sú predefinované a možno si ich vybrať po kliknutí na *Spríevodcu vstupnou maskou* . Tlačidlo sa nachádza na konci riadka pre vstupnú masku. Okrem výberu existujúcej masky si používateľ môže vytvoriť, otestovať a uložiť vlastnú masku.

Ak je definovaná vstupná maska a nie je definovaný formát, použije sa pri zobrazení formát vstupnej masky.

Popis. Môžeme definovať text v záhlaví stĺpca. Požíva sa hlavne ak sme stĺpec pomenovali bez medzier a diakritiky ale pri zobrazení tabuľky ho chceme mať zobrazené pekne a gramaticky správne.

Predvolená hodnota. Môžeme definovať hodnotu, ktorá sa ponúkne v poli pri vytváraní nového záznamu. Napríklad v tabuľky *Knihy* v stĺpci *Jazyk*, by mohlo byť preddefinované „slovenčina“, pretože predpokladáme, že väčšina kníh bude v tomto jazyku.

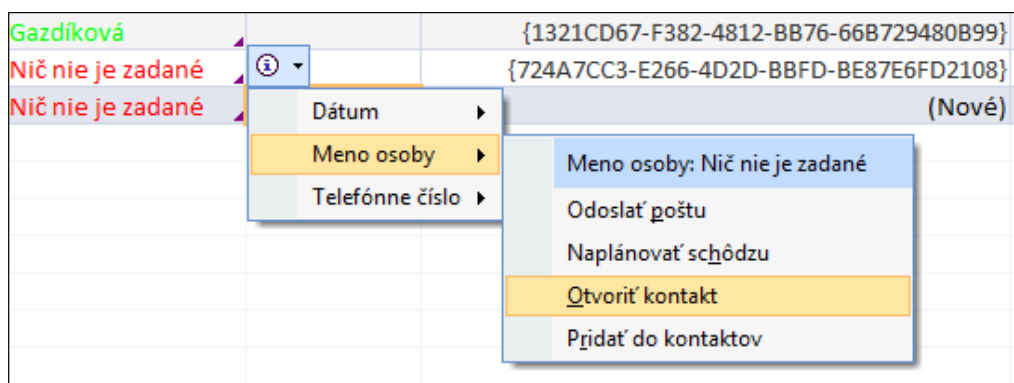
Overovacie pravidlo. Je veľmi dôležité aby sa do databázy nedostali údaje, ktoré tam nepatria. Preto sa vkladané údaje viac stupňovo kontrolujú. Základnou kontrolou je typ údajov, ktorý ľahko odhalí neexistujúci dátum (29.2.2011) alebo text namiesto čísla. Ďalšou kontrolou môže byť vstupná maska. Tretí stupeň kontroly môže tvoriť práve *Overovacie pravidlo*.

Existujú dva typy overovacích pravidiel. Vo *Vlastnostiach poľa* je možné použiť pravidlo na kontrolu daného poľa. Táto kontrola sa vykonáva vždy po zadaní obsahu každého poľa a nastavuje sa pre každé pole zvlášť. Druhý typ sa vykonáva pred opustením riadku a môže kontrolovať vzájomný vzťah viacerých polí (napríklad či dátum vrátenia nie je pred dátumom požičania). Toto overovacie pravidlo sa nastavuje v *Hárku vlastností* tabuľky.

Overovacie pravidlo je podmienka, ktorá musí byť splnená, aby bolo používateľovi umožnené opustiť pole pri editácii. Ak omylom vytvoríme podmienku, ktorá nie je splnená nikdy alebo nevieme, čo zadať aby bola splnená, môžeme editáciu ukončiť stlačením klávesu ESC. Napríklad podmienka „a*“ dovoľí zadať iba text začínajúci písmenom a, podmienka >“f“ iba texty, ktoré sú v abecede za písmenom f.

Pre číselné polia to môže byť podmienka >0 And <6, ktorá umožní zadať iba čísla väčšie ako 0 a menšie ako 6. Spolu s nastavenou veľkosťou *Byte* (t. j. celé čísla 0-255), tak môžeme obmedziť vkladanú hodnotu na čísla 1, 2, 3, 4 alebo 5.

Podrobnejšie sú podmienky aj s ďalšími príkladmi (v inej súvislosti) rozoberané v kapitole 3.1.



Obr. 2.3 Nastavenie niektorých vlastností poľa

Overovací text. Ak nie je splnená podmienka nastavená vlastnosťou *Overovacie pravidlo*, tak sa vypíše tu uvedený text.

Povinné. Niektoré údaje môžu byť nepovinné, napríklad *Cena*. *Priezvisko* čitateľa je ale zrejme podstatný údaj a preto by mal byť povinný – bez neho nie je možné pridať do tabuľky nového čitateľa.

Povolit' nulovú dĺžku. Táto vlastnosť určuje pri poliach typu *Text* a *Memo*, či je možné zadať prázdny textový reťazec, t. j. dve dvojité úvodzovky bez medzery medzi nimi: "". V prípadne zadania *Nie* musí byť vložený aspoň jeden znak (napr. medzera). Ak je pre túto vlastnosť nastavená hodnota *Áno*, do poľa je možné zadať prázdny reťazec. Pomocou týchto reťazcov je možné uviesť, že pre pole neexistuje žiadna hodnota. Ak by sme neuviedli nič (ani "") bola by v poli hodnota *Null*.

Indexované. Používa na zrýchlenie niektorých operácií s databázou. Hlavne operácie vyhľadávania a triedenia. Access si pre indexované stĺpce vytvára vlastné pracovné tabuľky. Indexy sa môžu použiť aj na zamedzenie zadávania duplicitných údajov. V prípade povolenia indexovania sa musíme rozhodnúť či sú duplicity povolené alebo nie. Primárne kľúče sú automaticky indexované a sú samozrejme bez duplicit.

Kompresia kódu Unicode. Access používa tabuľku znakov Unicode, ktorá kóduje jeden znak dvomi byte-ami. Znak používané v západných jazykoch sa dajú kódovať jedným bajtom a v tabuľke Unicode je pre tieto znaky prvý byte 0. Ak je táto vlastnosť nastavená na *Áno*, tak sa takéto znaky pred uložením komprimujú a pred výberom z databázy dekomprimujú.

Vlastnosti IME. Nastavujú podporu pre východné jazyky.

Inteligentné značky. V programoch MS Office 2007 môžeme použiť inteligentné značky na vykonanie rôznych akcií. Spoznáme ich podľa malého fialového trojuholníka v pravom dolnom rohu. Touto vlastnosťou môžeme poľu priradiť akcie ako odoslať emailom, pridať do kalendára a podobne. Akcie sú zobrazené na obr. 2.3.

Memo

Memo je textový typ určený pre ukladanie dlhých textov. Od typu *Text* sa odlišuje nielen veľkosťou obsahu, ale aj možnosťou formátovania obsahu navyše formátom RTF (rich text format) podobne ako v textovom editore. Oproti typu *Text* má dve vlastnosti navyše.

Formát textu. Umožňuje alebo zakazuje použitie formátovania RTF.

Len priložiť. Povolenie tejto vlastnosti má za následok, že používatelia môžu pridávať nové údaje, ale nemôžu zmeniť ani odstrániť už existujúce. Na prvý pohľad vypadá, že údaje sa menia a mazajú, ale zachováva sa história všetkých operácií (a teda aj pôvodný obsah).

Použitie typu memo je zobrazené na obr. 4.1 v poli *Životopis*.

Number a Currency

Veľkosť poľa. Možnosti sú určené počtom bytov, ktoré číslo môže zaberáť a tým či umožníme vkladať číslo celé alebo aj desatinné.

Celé čísla:

- *Byte.* Čísla od 0 do 255. Zaberá jeden byte.
- *Integer.* Čísla od - 32 768 do 32 767. Zaberá dva byty.
- *Long Integer.* Čísla od -2 147 483 648 až 2 147 483 647. Zaberá štyri byty.

Čísla s pohyblivou desatinnou čiarkou:

- *Single.* Čísla od $-3,4 \times 10^{38}$ do $+3,4 \times 10^{38}$. Zaberá štyri byty. Presnosť 7 platných číslic⁹.
- *Double.* Čísla od $-1,797 \times 10^{308}$ do $+1,797 \times 10^{308}$. Zaberá osem bytov. Presnosť 15 platných číslic¹⁰.
- *Decimal.* Čísla od -10^{28} do $+10^{28}$. Zaberá dvanásť bytov. Pri tomto type sa dá nastaviť presnosť až na 28 miest. Musíme určiť koľko z nich bude napravo od desatinnej čiarky.

Špeciálnym typom je *Replication ID*. Je to globálne (celosvetovo) jedinečný¹¹ identifikátor. Zaberá 16 bytov. Používajú sa napríklad, ak je databáza kvôli výkonu alebo veľkosti rozdelená na viacero serverov, napríklad pre objednávky svetových e-shopov. Objednávky sa v konečnom dôsledku sústreďujú na nejakom mieste v jednej tabuľke. Ak by sa používal obyčajný *AutoNumber* mohlo sa stať, že dve objednávky z rôznych serverov by mali rovnaký primárny kľúč. Takéto číslo sa zapisuje v šestnástkovej sústave v tvare {764A23C8-249B-496D-9024-9DAF17C2B7BB}. Situácia je znázornená na obr. 2.3.

Formát. Možno si vybrať z nasledujúcich predefinovaných formátov:

- *Všeobecné číslo.* Číslo sa zobrazí tak ako bolo zadané. Pre čísla majúce viac platných číslic ako je presnosť typu sa číslo zaokrúhli a použije sa vedecký zápis, ktorý je popísaný nižšie.
- *Currency (mena).* Použije sa nastavenie z Windows. Pre Slovensko to bude oddeľovanie tisícok, dve desatinné miesta a znak € za číslom.
- *Fixed (pevný).* Rovnaké ako formát *Všeobecné číslo* ale nepoužije sa vedecký zápis.

⁹ To znamená, že napríklad číslo 1 234 567 890 123 456 789 môže byť uložené rovnako ako číslo 1 234 567 333 333 333 333. Namiesto posledných trojok môžu byť iné číslice. Inak povedané číslo sa uloží v tvare $1,234\ 567 \times 10^{18}$.

¹⁰ Vyššie spomínané číslo by sauložilo rovnako ako číslo 1 234 567 890 123 453 333. Namiesto posledných trojok môžu byť iné číslice. Uloží sa v tvare $1,23456789012345 \times 10^{18}$.

¹¹ Jedinečný treba brať z istou rezervou. Číslo sa pseudonáhodne generuje a preto existuje nenulová pravdepodobnosť, že dve hodnoty typu *Replication ID*, ktoré sa stretnú v spoločnej tabuľke budú rovnaké. Táto pravdepodobnosť je ale taká malá, že sa neberie do úvahy.

- *Standard (štandardný)*. Rovnaké ako formát *Fixed*, navyše sa použije oddeľovač tisícov (medzera, ovplyvnené nastavením Windows).
- *Percent (percentá)*. Za číslo sa pridá automaticky znak %. Existujúce čísla v stĺpci sa vynásobia číslom sto. Ak pri tomto zobrazení vložíme číslo, uloží sa jeho stotina. (5% = 0,05).
- *Scientific (vedecký)*. Použije sa semilogaritmický tvar, t. j. číslo 26 358,25 sa zobrazí ako 2,635825E+04 (teda $2,635825 \times 10^4$) a číslo 0,00458 sa zobrazí ako 4,58E-03.

Okrem týchto predefinovaných formátov je možné použiť aj vlastné. Podobne ako pri formátovaní textu môžeme aj tu použiť vo formátovacom reťazci zástupné a riadiace znaky alebo literály.

- #, zástupný znak. Ak má číslo menej číslic pred desatinnou čiarkou ako je vo formátovacom reťazci znakov #, tak sa tieto zástupné znaky ignorujú. Napríklad pri (často používanom) formátovaní # ### sa číslo 53 zobrazí ako 53. Číslo 1234 resp. 1234567889 sa pri tomto formátovaní zobrazí 1 234 resp. 123 456 789.
- 0, zástupný znak. Ak má číslo menej číslic pred desatinnou čiarkou ako je vo formátovacom reťazci znakov #, tak sa namiesto tieto zástupné znaky nahradia 0. Napríklad pri formátovaní 0 000 sa číslo 53 zobrazí ako 0 053. Číslo 1234 resp. 1234567889 sa pri tomto formátovaní zobrazí 1 234 resp. 123 456 789.
- Znaky *, \, úvodzovky, [farba], čiarka, oddeľovač tisícov a literály sa používajú rovnako ako pri formáte textu.
- !, riadiaci znak. Zarovná čísla doľava. V tomto prípade sa nemôžu použiť zástupné znaky 0 a #. Možno ale použiť zástupné znaky pre text (@, &).
- %, riadiaci znak. Vynásobí obsah poľa číslo 100 a zobrazí na konci znak percenta.
- E- alebo E+, riadiaci znak. Rozširuje možnosti vedeckého formátu, ktorý čísla upravuje tak, že pred desatinnou čiarkou je práve jedna číslica. Napríklad číslo 1 234 567 by zobrazil (v závislosti na počte zobrazovaných desatinných miest) ako 1,234567E+06. Ak by sme chceli pred desatinnou čiarkou mať stovky (tri miesta) mohli by sme použiť formát ###,##E+.

Formátovať možno zvlášť čísla kladné, záporné, nulu a nezadané hodnoty. Jednotlivé formáty oddeľujeme bodkočiarkou a musia nasledovať v uvedenom poradí. Ak niektorý z formátov nechceme definovať ale nasledujúci áno, tak iba pridáme 0 a bodkočiarku. Napríklad formát 0;0;0;"Nič" zobrazí slovo „Nič“, ak je pole prázdne.

Desatinné miesta. Počet desatinných miest, ktoré sa zobrazujú. Zobrazujú sa aj v prípade celých čísel.

Ostatné vlastnosti sú zhodné s vlastnosťami textu.

Date/Time

Umožňuje uložiť dátum a čas. Aj keď sa hodnota môže (v závislosti na formátovaní) zobrazovať rôzne, je dobré povedať už na tomto mieste, že Access chápe tento typ ako číslo. Ako číslo znamenajúce počet dní od 30. decembra 1899. Teda 1. 1. 1900 je číslo 1, 2. 1. 1900 je číslo 2. Dátum 5. 5. 2000 je 36 651. Pre dátumy spreď 30. 12. 1899 sa používajú záporné čísla.

Čas sa vyjadruje ako časť dňa. Napríklad 24. 12. 2009 o šiestej večer je číslo 40 171,75, pretože 0,75 z 24 hodín je 18.

Výhodou takejto interpretácie je možnosť odčítania dvoch dátumov od seba a tak zistiť koľko dní (času) medzi nimi uplynulo. Alebo môžeme napríklad k dnešnému dátumu pripočítať číslo 100 a zistíme, koľkého bude o sto dní. Aj porovnanie dvoch čísel je rýchlejšie ako porovnanie rokov a prípadne mesiacov, hodín, minút a sekúnd.

Tento údajový typ povoľuje dátum od 1. januára 100 n. l. do 31. decembra 9999 n. l a časy od 0 do 23:59:59. Ak očakávame dátum alebo čas mimo intervalu je nutné použiť textové pole.

Formát. Môžeme si vybrať z predefinovaných spôsobov zobrazenia dátumu a času. Niektoré z možností sú ovplyvnené nastavením v *Miestnych a jazykových nastaveniach* MS Windows.

Pri dvojčifernom roku sa roky 00 až 29 považujú za roky 2000 až 2029. Roky 30 až 99 sa považujú za roky 1930 až 1999.

Vlastné formáty dátumového typu môžu obsahovať nasledujúce znaky:

- /, : oddeľovač (oddeľovač: znak, ktorý oddeľuje jednotky textu alebo čísel.) dátumu a času.
- c - rovnaký formát ako preddefinovaný formát *Všeobecný dátum*.
- d, resp. dd - deň v mesiaci, resp. vždy dvojmiestny (01) deň v mesiaci.
- ddd, resp. dddd - skratka dňa v týždni (po až ne) resp. celý názov.
- ddddd, resp. dddddd - rovnaký formát ako preddefinovaný krátky resp. dlhý dátum.
- w - číslo dňa v týždni (1 až 7).
- ww - týždeň v roku (1 až 53).
- m, resp. mm - číslo mesiaca v roku, resp. vždy dvojmiestne číslo (01).
- mmm, podľa pomocníka by sa mali zobrazovať prvé tri písmená názvu mesiaca (jan až dec) – v skutočnosti sa zobrazuje rímske číslo mesiaca.
- mmmm - celý názov mesiaca.
- q - dátum zobrazený ako štvrtrok roka (1 až 4).
- y - číslo dňa v roku (1 až 366).
- yy, resp. yyyy - posledné dve číslice roka (01 až 99), resp. štvorciferné zobrazenie roka.
- h, resp. hh - hodina, resp. vždy dvojmiestna hodina (00-23).
- n, resp. nn - minúta jednociferná, resp. vždy dvojčiferná (00 až 59).
- s, resp. ss - sekunda jednociferná, resp. vždy dvojčiferná (00 až 59).
- tttt – formát zhodný s preddefinovaným formátom dlhého času.
- AM/PM, resp. A/P - čas v dvanásťhodinovom formáte s označením AM alebo PM, resp. A alebo P. Analogicky možno použiť aj malé písmená.
- AMPM - čas v dvanásťhodinovom formáte. Dopoludnie/odopoludnie je označené podľa nastavenia definovaného vo Windows.
- *, \, úvodzovky, literály a [farba] sa používajú rovnako ako pri formáte textu.

Napríklad dátum 9.8.2010 16:15:33 sa pri formáte *ddd", "dd.\ mmmm\ yyyy\ h:nn:ss* zobrazí ako *pondelok, 09. august 2010 16:15:33*.

Vlastné formáty, ktoré nie sú konzistentné s nastaveniami zadanými v miestnych nastaveniach systému Windows sa ignorujú.

AutoNumber

Access pri tomto údajovom type vkladá hodnoty sám, pričom dbá, aby sa hodnoty neopakovali. Ide o uľahčenie práce pre používateľa, pretože tento typ sa používa pre primárny kľúč. Napríklad: Access pri vytvorení tabuľky automaticky vytvorí primárny kľúč s názvom *Identifikácia*, ktorý je typu *AutoNumber*.

Veľkosť poľa. Predvolená hodnota je *Long Integer*. Druhá možnosť je *Replication ID*. Vysvetlenie týchto pojmov je pri údajovom type *Number*.

Nové hodnoty. Určuje spôsob ako sa vytvorí ďalšie číslo. Prednastavený spôsob je postupné zvyšovanie o jedna. Hodnoty sú v takom prípade stále vyššie a vyššie a to aj keď sa uvoľní nejaké číslo odstránením niektorého záznamu. Druhý spôsob je náhodné generovanie čísel.

Indexované. Pokiaľ je stĺpec použitý ako primárny kľúč je možné iba indexovanie bez duplicity. V každom prípade sú ale hodnoty jedinečné.

Yes/No

Predvolená hodnota. V prípade 0, *Off*, *False*, *No* bude zaškrtnuté políčko bez fajky. Ľubovoľné iné číslo, *On*, *True*, *Yes* znamenajú políčko zaškrtnuté.

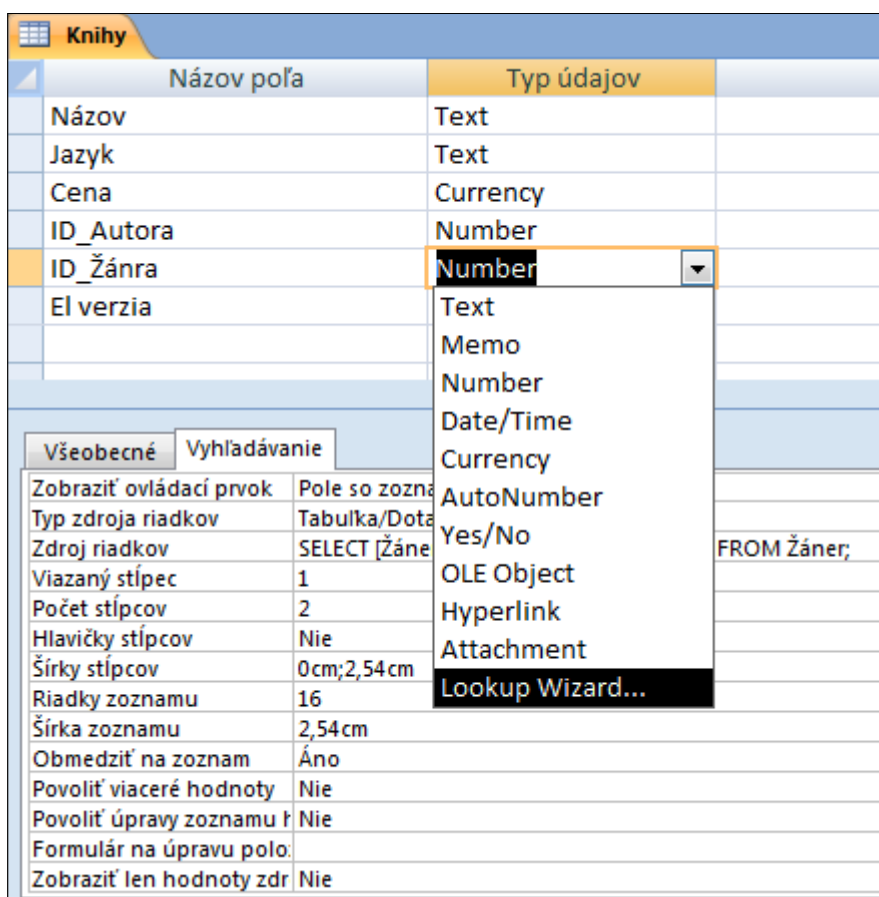
Overovacie pravidlo pre tento typ nemá veľké uplatnenie, pretože sú iba dve možné hodnoty a ak jednu nepovolíme, tak tú druhú môžeme priamo nastaviť ako predvolenú.

Z rovnakého dôvodu nemá význam *Indexovanie bez duplicit*. Fungovalo by iba v tabuľke s maximálne dvoma riadkami, z ktorých jeden by bol *Yes* druhý *No*.

Ostatné údajové typy majú rovnaké vlastnosti, preto sa im nebudeme venovať. *Lookup Wizard* (Sprievodca vyhľadávaním) je vysvetlený v kapitole 2.2.

2.2. Prepojenie tabuliek

Z prvej kapitoly už vieme, že niekedy je dobré tabuľku rozdeliť. Tabuľky, ktoré vznikli rozdelením ale nie sú úplne samostatné, zvyčajne sa odvolávajú jedna na druhú. Prepojenie tabuliek v našej databáze je znázornené na obr. 1.5. Napríklad v tabuľke *Knihy* je odkaz na *ID_Autora* v tabuľke *Autori* a odkaz *ID_Žánra* do tabuľky *Žánre*. Aby mohol Access s takýmito tabuľkami pracovať, musíme toto prepojenie v databáze vytvoriť.



Obr. 2.4. Vlastnosti prepojenia tabuliek

Využijeme pri tom v zobrazení v návrhovom režime a v stĺpci typ údajov zvolíme *Lookup Wizard* (Sprievodca vyhľadávaním), ktorý dovoľuje vyhľadať a použiť hodnotu z inej tabuľky alebo zoznamu.

Ak chceme tabuľku *Knihy* prepojiť s tabuľkami *Žánre* a *Autori*, tak musia tieto tabuľky existovať. Pri vytváraní tabuľky *Knihy* vyberieme možnosť *Lookup Wizard* tak ako je to znázornené na obr. 2.4.

Spustí sa sprievodca vyhľadávaním. Máme dve možnosti:

- Hodnoty pre vyhľadávací stĺpec načítať z tabuľky alebo dotazu.

- Hodnoty zadá používateľ.

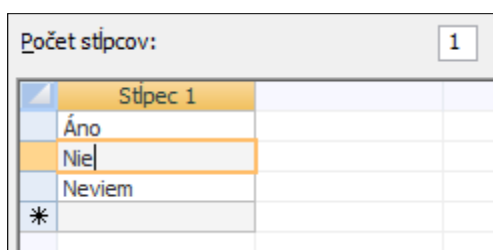
V našej databáze vyberieme prvú možnosť – dáta budeme brať z tabuľky. Je to rovnaký postup ako keby sme ich brali z dotazu (o dotazoch pojednáva kapitola 3).

Intermezzo

Pred tým ako budeme pokračovať, rozoberieme druhú možnosť. Tú by sme vybrali, ak by sme hodnoty nechceli brať z inej tabuľky (alebo dotazu), ale zadefinovali by sme ich spolu s typom údajov.

Napríklad, ak by sme mali tabuľku pre anketu. Každý riadok sú odpovede jedného respondenta a stĺpce sú jednotlivé otázky. Jedná z otázok by bola „Máte radi leto?“. Na výber by boli odpovede „Áno“, „Nie“ a „Neviem“ (Pri iba dvoch možnostiach by sa použil typ *Yes/No*).

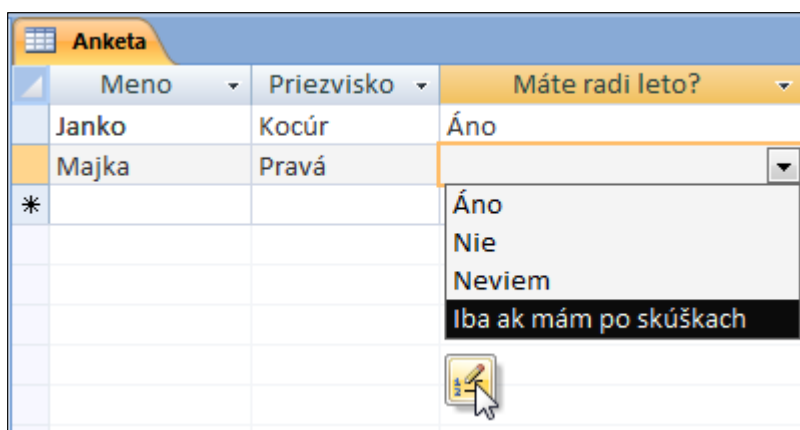
Po výbere možnosti *Hodnoty zadá používateľ* sa zobrazí okno, v ktorom môžeme určiť aké hodnoty budú používať. Situácia je znázornená na obr. 2.5. V tomto prípade nám stačí jeden stĺpec ale môžeme použiť aj viac.



Obr. 2.5 Zázpis povolených hodnôt pri sprievodcovi vyhľadáváním, po výbere možnosti *Hodnoty zadá používateľ*

V ďalšom kroku by sme mali pomenovať stĺpec a tým je určenie hodnôt používateľom ukončené. Použitie tabuľky je znázornené na obr. 2.6. Pri výbere z povolených hodnôt sa pod možnosťami zobrazí ikona pre *Úpravu položiek zoznamu*. Po kliknutí na ňu môžeme hodnoty doplniť alebo ubrať. V ponuke na obrázku vidieť, že niekto už jednu hodnotu pridal (*Iba ak mám po skúškach*).

V prípade, že sa v tabuľke namiesto textových hodnôt zobrazuje číslo, asi máte v stĺpci *Máte radi leto?* údajový typ *Number*. Stačí ho zmeniť na *Text*.



Obr. 2.6 Pridanie ďalšej povolenej hodnoty

Koniec intermezza.

Ak sme vybrali prvú možnosť, musíme vybrať tabuľku alebo dotaz, z ktorej budeme čerpať hodnoty. Vyberieme tabuľku *Žáner*. V ďalšom okne musíme vybrať stĺpec, podľa ktorého sa bude vyhľadávať (ID). Okrem neho vyberieme aj druhý, v ktorom sú pomenované žánre. V prípade tabuľky *Autori* by sme vybrali okrem primárneho kľúča aj stĺpce *Priezvisko* a *Meno*. Vyskúšajte si aj pridanie stĺpcov *Meno* a *Priezvisko*, teda

v inom poradí a otestuje to v tabuľke v údajovom zobrazení. Tento problém si vyriešime pomocou dotazu v kapitole 3.1.2.

V ďalšom okne môžeme určiť či majú byť hodnoty usporiadané. Pre lepšie vyhľadávanie v zozname je vhodné dať triediť podľa poľa *Žáner*. V nasledujúcom kroku môžeme nastaviť šírku stĺpca. Stĺpec s kľúčom sa automaticky skryje – to je v drvivej väčšine prípadov požadované. V tomto kroku je možné jeho zobrazenie nastaviť. V poslednom kroku môžeme stĺpec pomenovať.

Pri popise návrhového zobrazenia v kapitole 2.1.2 sme si popísali typy údajov v karte *Všeobecné*. Pri prepojení môžeme využiť nastavenia z karty *Vyhľadávanie* (obr. 2.4).

- *Zobraziť ovládací prvok*. Určuje ako sa bude zobrazovať toto pole vo formulári.
 - *Textové pole*. Používateľ nebude mať na výber z povolených možností. Ak chce vložiť novú hodnotu alebo zmeniť existujúcu musí ju napísať.
 - *Zoznam*. Viac riadkové okno, v ktorom možno rolovať a vyberať z povolených možností.
 - *Pole so zoznamom*. Výzor bude ako pri textovom poli ale po kliknutí do poľa sa rozbalí zoznam s hodnotami, v ktorých si používateľ môže vybrať. Táto voľba je väčšinou najpraktickejšia.
- *Typ zdroja riadkov*. Odkiaľ sa budú brať dáta. Sú tri možnosti:
 - *Tabuľka/Dotaz*. V *Zdroji riadkov* (nižšie) je názov tabuľky, dotazu alebo SQL výraz (ktorý sa v prípade použitia sprievodcu vyhľadávaním vytvorí automaticky).
 - *Zoznam hodnôt*. V *Zdroji riadkov* sú uvedené hodnoty oddelené bodkočiarkou.
 - *Zoznam polí*. V *Zdroji riadkov* je názov tabuľky, dotazu alebo SQL výraz.
- *Zdroj riadkov*. Najčastejšie SQL výraz, dotaz alebo zoznam hodnôt oddelených bodkočiarkou.
- *Viazaný stĺpec*. Číslo stĺpca (zľava), ktorým sa zoznam prepája s tabuľkou. Väčšinou 1.
- *Počet stĺpcov*. Koľko stĺpcov sa má zobraziť. Dá sa ovplyvniť zadaním šírky 0 vo vlastnosti *Šírky stĺpcov* (nižšie).
- *Hlavičky stĺpcov*. Nastavuje či sa majú zobrazovať záhlavia stĺpcov.
- *Šírky stĺpcov*. Šírka v centimetroch. Ak zdroj riadku vráti viacero stĺpcov môžete nastaviť šírku každého oddelením bodkočiarkou. Napríklad 0; 3; 3; 1 znamená, že prvý stĺpec sa nezobrazí.
- *Riadky zoznamu*. Určuje koľko riadkov má mať rolovacia ponuka, ak je zvolené *Pole so zoznamom*.
- *Šírka zoznamu*. Nastavuje šírku ponuky. V prípade, že je šírka stĺpca väčšia ako tu nastavená hodnota, bude šírka ponuky rovnako široká ako stĺpec.
- *Obmedziť na zoznam*. Určuje či sú prípustné iba hodnoty v ponuke.
- *Povolit' viaceré hodnoty*. Ak sa rozhodnete povoliť viaceré hodnoty v jednom poli, tak toto rozhodnutie už nebudete môcť zmeniť. (A pretože to odporuje [INF](#), tak to zrejme čoskoro oľutujete.)
- *Povolit' úpravy zoznamu hodnôt*. Ak je táto možnosť zakázaná, tak nie je možné pridať do zoznamu hodnôt ďalšiu položku. Pozri obr. 2.6.
- *Formulár na úpravu položiek zoznamu*. Názov formulára. Formuláre sú vysvetlené v kapitole 4.
- *Zobraziť len hodnoty zdroja riadka*.

Relácie medzi tabuľkami

Ak chceme porozumieť fungovaniu databázy, musíme poznať jej štruktúru, t. j. logické spojenia (vzťahy) medzi tabuľkami. Na zobrazenie vzťahov v databáze môžeme použiť kartu *Databázové nástroje* a v skupine *Zobraziť alebo skryť* voľbu *Vzťahy*. Ak chceme vzťahy aj upravovať, musíme uložiť a zavrieť všetky otvorené tabuľky. Vzťahy by sa mali zobrazovať podobne ako na obr. 1.5. Ak sa vzťahy nezobrazujú správne, skontrolujte:

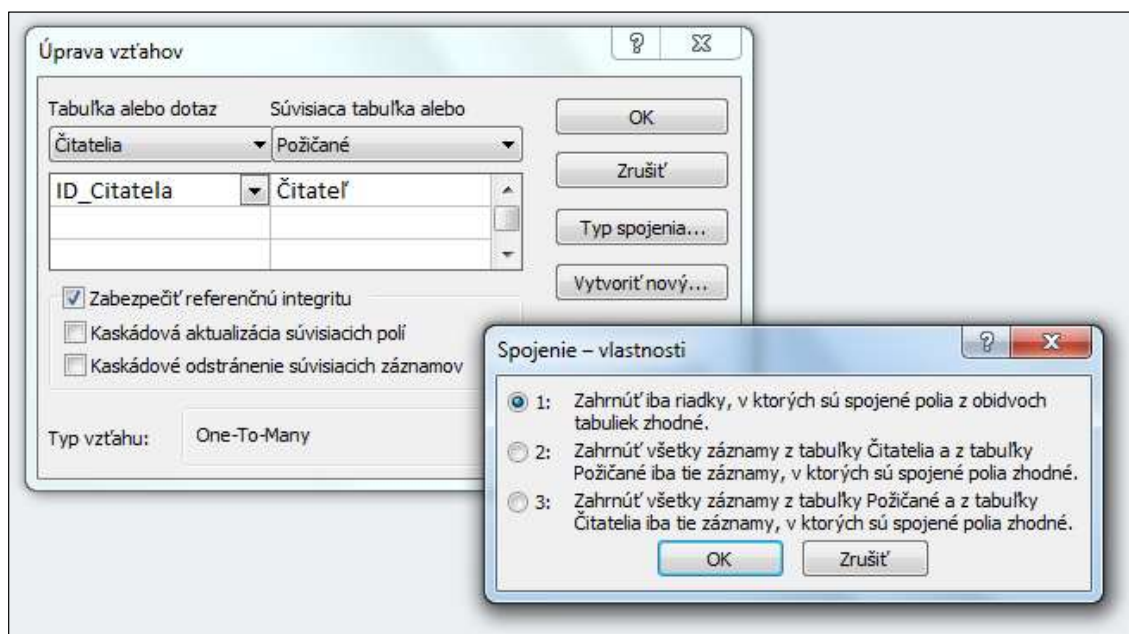
- Či sú všetky tabuľky uložené - ak ste pridali pole do tabuľky a tabuľku ste neuložili (pri zavretí sa uloží), tak sa vo vzťahoch nezobrazia.
- Ak vo vzťahoch chýba (uložená) tabuľka, tak ju môžeme zobraziť pomocou voľby *Zobraziť tabuľku* alebo *Všetky vzťahy* na karte *Návrh*, v skupine *Vzťahy*.
- Ak sú tabuľky poprehadzované a spojenia sa krížia, môžete tabuľky popresúvať po kliknutí na záhlavie.

Vzťahy vytvárame štandardne pri definovaní tabuliek pomocou sprievodcu vyhľadávaním. Je ich ale možné vytvoriť, upraviť a rušiť aj cez *Vzťahy*. Dokonca, ak by sme chceli odstrániť z niektorej tabuľky pole, ktoré je súčasťou vzťahu, Access to nedovolí. Najprv musíme vzťah zrušiť.

Vzťah odstránime kliknutím na spojnicu medzi poľami a stlačíme kláves *Delete* alebo po kliknutí pravým tlačidlom na spojnicu a výberom z kontextového menu.

Pre vytvorenie nového vzťahu klikneme na pole, podržíme ho a presunieme na pole v inej tabuľke. Pre úpravu existujúceho stačí dvojklik na spojnicu alebo kliknutie pravým tlačidlom na spojnicu a vybratie možnosti *Upraviť vzťah* Na obr. 2.7 je znázornené okno na úpravu vzťahu. Ak chceme zabezpečiť, aby sa nemohlo stať, že v tabuľke *Požičané* bude odkaz na čitateľa, ktorý neexistuje v tabuľke *Čitatelia*, tak označíme *Zabezpečiť referenčnú integritu*. To zároveň zobrazí aj značky 1 a ∞ na koncoch spojnic.

V prípade, že by sme zaškrtnuli aj voľbu *Kaskádové odstránenie súvisiacich záznamov*, tak by sa po odstránení čitateľa z tabuľky *Čitatelia*, automaticky zmazali v tabuľke *Požičané* všetky záznamy, v ktorých sa dotýčny čitateľ nachádzal.



Obr. 2.7. Úprava vzťahov v databáze

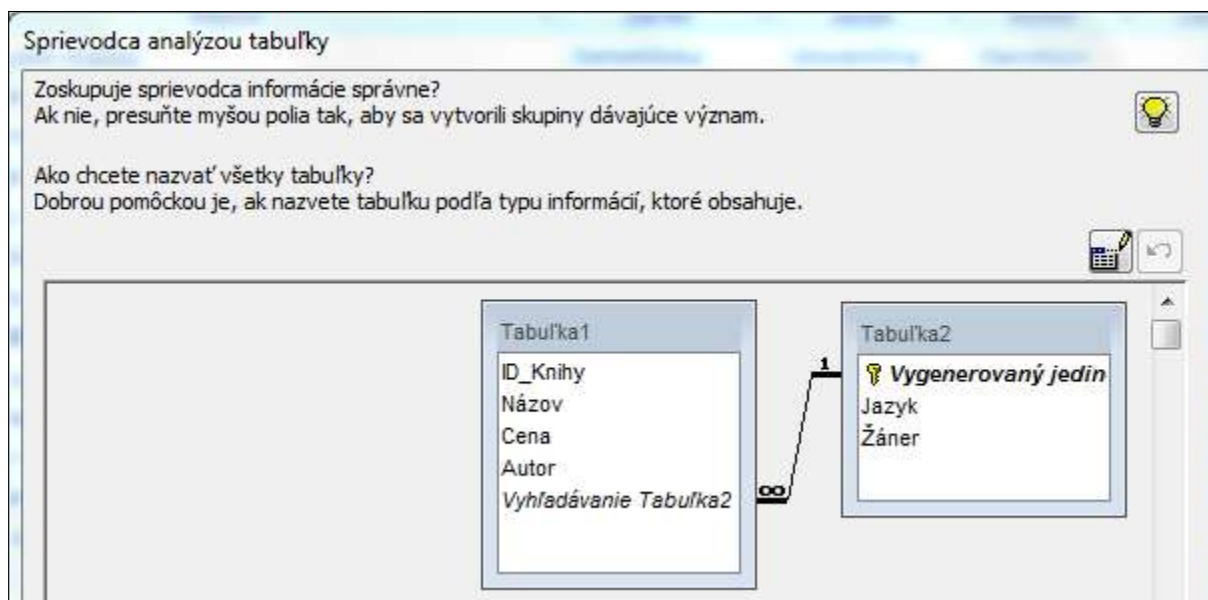
Pri úprave vzťahov môžeme vybrať z troch možností (sú znázornené na obr. 2.7) spojenia tabuliek. Najčastejšie je prvé prepojenie, spojenie typu dva je použité v kapitole 4.3. na obr. 4.10 a tiež v kapitole 7.5. na obr. 7.4. a obr. 7.5.

V našom návrhu na obr. 1.5. sme nechali pole *jazyk* v tabuľke *Knihy* ako pole typu text. Pretože sa hodnoty v tomto poli opakujú (v našej tabuľke sú iba tri – slovenčina, ruština a čeština), bolo by výhodnejšie vytvoriť tabuľku *Jazyky* a na ňu sa v tabuľke *Knihy* odkazovať. Teda urobiť to podobne ako s poľom *Žáner*. Neurobili sme to pri návrhu preto, aby sme si ukázali, že Access poskytuje nástroj na vyhľadanie a opravenie takýchto nedokonalostí.

Ide o to, že používatelia (firmy, organizácie) si často svojpomocne vytvorili databázu, najčastejšie ako jedinú tabuľku. Tú naplňajú a využívajú až do štádia, keď im prerastie cez hlavu a zistia, že takto to už ďalej nejde – tabuľku treba rozdeliť, optimalizovať. Jedným zo spôsobov je návrh novej štruktúry, do ktorej by sa „nejako“ preniesli údaje z pôvodnej databázy. Samotný prenos sa vykonáva ručne :-/ alebo pomocou dotazov. Dotazmi sa zaoberá kapitola 3.



Druhým spôsobom je využitie schopnosti programu Access analyzovať tabuľku, navrhnúť úpravy a prípadne ich aj zrealizovať. Na karte *Databázové nástroje*, v skupine *Analyzovať* je príkaz *Analyzovať tabuľku*. Po jeho spustení Access v prvých dvoch oknách stručne vysvetľuje približne to, čo je popísané v kapitolách 1.2. až 1.3. Je dobré si príklady prečítať. V treťom okne si vyberieme tabuľku, ktorá má byť analyzovaná. Vyberieme tabuľku *Knihy*.

Ďalej si môžeme vybrať či úpravu necháme na Access alebo ju navrhne sami. Pretože jeho návrh môžeme pozmeniť neskoršie (je pohodlnejšie opravovať ako tvoriť), vyberieme si prvú možnosť.



Obr. 2.8. Návrh na úpravu tabuľky po analýze programom Access

Obr. 2.8. ukazuje čo navrhuje Access. Vidíme, že správne identifikoval pole *Jazyk*. Z nejakého dôvodu ho ale spojil s poľom *Žáner*. Kliknutím na pole *Žáner* a jeho pretiahnutím naspäť do ľavej tabuľky chybu napravíme. Pri prvom použití si môžeme prečítať tipy, ktoré zobrazíme kliknutím na ikonu žiarovky v pravej hornej časti okna sprievodcu. Kliknutím na ikonu s ceruzkou môžeme označenú tabuľku premenovať.

V ďalšom kroku sa zmení ikona premenovania na ikony s kľúčmi. Pomocou nich môžeme označiť ktoré pole je kľúč  alebo vytvoriť nový . V našom obrázkovom príklade označíme pomocou prvej ikony *ID_Knihy* ako primárny kľúč. Môžeme si prečítať nové tipy.



V nasledujúcom okne, ktoré sa nemusí vždy zobrazovať, Access upozorní na možné chyby. V našom prípade, pretože z cca 3000 kníh je iba niekoľko českých a ruských, upozorňuje na možnosť, že by mohli byť všetky slovenské. Tu to vyznieva divne, ale je to podobný prípad ako v príklade z prvého okna sprievodcu analýzou tabuľky.

V poslednom kroku si môžeme vybrať, či chceme aby Access vytvoril dotaz (dotazy sú vysvetlené v kapitole 3.), ktorý by vytvoril z našich dvoch nových tabuliek efektívnejšiu imitáciu pôvodnej (tá zostane zachovaná, k jej názvu sa pridá „_OLD“).

2.3. Práca s tabuľkou

V tejto kapitole si popíšeme základy práce s tabuľkou. Na obr. 2.9. vidíme tabuľku *Knihy*. V druhom riadku je práve upravovaná hodnota v stĺpci *Názov*. Keď sa vykoná nejaká zmena, zobrazí sa naľavo od záznamu ikona ceruzky (vidno ju aj na obrázku). Tento symbol úpravy záznamu ostane zobrazený, kým kurzor neopustí záznam.

Ak je nastavené overovacie pravidlo pre záznam, tak sa vykoná predtým ako ikona zmizne. Ak po úprave záznamu hodnoty nespĺňajú toto pravidlo, Access nám nedovolí opustiť riadok a ukončiť editáciu. Ak nedokážeme vložiť údaje spĺňajúce overovacie pravidlo záznamu splniť, tak musíme editáciu ukončiť stlačením *Esc* a hodnoty sa vrátia do pôvodného stavu. (Obdobne to platí aj pre overovacie pravidlo poľa.)

V prípade pridávania nového záznamu je to rovnaké až na to, že sa zobrazí symbol . Nový záznam pridáme príkazom *Nové* z pásu s nástrojov v skupine *Záznamy* na karte *Domov*. Druhá možnosť je písať za posledný záznam. Treťou navigačné tlačidlo , na obrázku je ho vidieť pod posledným záznamom, pod slovom „svět“. Nový záznam môžeme pridať aj klávesovou skratkou *Ctrl + +* (Ctrl a znak plus).

Príkazom *Odstrániť* z pásu nástrojov alebo stlačením *Ctrl+-* riadok odstránime. Ak je riadok označený (a nie je v režime úprav) môžeme ho odstrániť aj klávesom *Delete*. Riadok označíme kliknutím na miesto, kde sa zobrazujú ikonky editovania a pridávania záznamu.

Tieto základné operácie môžeme vybrať aj z kontextového menu riadka po kliknutí pravého tlačidla na **označený** záznam alebo na stĺpček pre ikonky.

V kontextovom menu riadka sa nachádzajú aj príkazy kopírovať, vystrihnúť a prilepiť. Po označení riadku môžeme celý riadok skopírovať do schránky a prilepiť po do iného (aj nového) riadku, ten musíme najskôr celý označiť. V skupine *Schránka* môžeme príkazom *Priložiť* pridať jej obsah na koniec tabuľky, aj keď nový riadok nemáme označený. Obdobne to platí aj pre stĺpce.



| | Názov | Jazyk | Cena | Autor | Žáner |
|---|------------------------------|------------|---------|-----------|--------------------|
| + | Nie je čas ceremónií | slovenčina | 40,00 € | Valero | próza |
| | Tajomný ostrov | slovenčina | 40,00 € | Verne | próza |
| + | Afrika | slovenčina | 40,00 € | Votrubec | odborná literatúra |
| + | Procesy, ktoré vzrušili svet | čeština | 39,00 € | Borovička | liter. faktu |

Záznam: 517 z 3016 Žiadny filter svět

Obr. 2.9. Základy práce s tabuľkou

Kliknutím na záhlavie stĺpca označíme celý stĺpec. Po kliknutí pravým tlačidlom na záhlavie stĺpca alebo na označený stĺpec si môžeme z kontextového menu vybrať nasledujúce príkazy:

- Usporiadanie podľa daného stĺpca. Utriedenie ďalšieho stĺpca nezruší prípadné predchádzajúce triedenia úplne. Podľa nich sa bude triediť v prípade rovnosti hodnôt v novom stĺpci. To znamená, že ak chceme triediť podľa ceny a v rámci nej podľa autora, tak je najlepšie triediť najskôr podľa autora a potom podľa ceny.

Môžeme triediť aj po kliknutí na trojuholníček vedľa názvu stĺpca alebo pomocou príkazov v páse nástrojov. Tam sa nachádza aj príkaz na zrušenie aktuálnych usporiadaní. Na obr. 2.9. sú v záhlaviach stĺpcov autor a cena šípky znázorňujúce smer triedenia daného stĺpca. Cena je

triedená zostupne a autor vzostupne. Bez pohľadu na údaje nie je možné určiť, ktoré triedenie má prednosť.

- Skryť alebo znovu zobrazit' niektoré stĺpce.
- Zmeniť šírku stĺpca. To je možné aj posúvaním hranice stĺpca doľava, doprava alebo dvojklikom ako je to zobrazené na obr. 2.1.
- Ukotviť alebo uvoľniť stĺpce. V prípade, že máme veľa stĺpcov alebo sú široké a posunieme sa doprava na posledné, tak nám ľavé stĺpce odrolujú – zmiznú. To môže byť nepohodlné a aj nebezpečné. Ak vľavo máme meno a priezvisko osôb a pri úprave údajov sa dostaneme na koniec riadku, môžeme omylom opraviť údaj osobe o riadok vyššie či nižšie. Preto je dobré niektoré stĺpce zafixovať – ukotviť. V spomínanom prípade by sme ukotvili meno a priezvisko. Ukotvené stĺpce sa automaticky presunú doľava a pri rolovaní v tabuľke sa neposúvajú. Ukotvené stĺpce nemôžeme presúvať. Po uvoľnení stĺpce ostanú na novej pozícii a na pôvodné miesto ich musíme presunúť my.
- Pridať, odstrániť alebo premenovať stĺpec. Premenovať stĺpec môžeme aj po dvojkliku na jeho záhlavie alebo v návrhovom režime.
- Hľadať a nahrádzať. Tieto príkazy sa podobajú na hľadanie v programe Word. Môžeme si vybrať či hľadáme iba v aktuálnom stĺpci alebo v celej tabuľke. Či sa hľadá ľubovoľná časť slova, celé slovo alebo jeho začiatok. Môžeme určiť aj smer hľadania: od aktuálnej pozície (nahor alebo nadol) alebo od prvého záznamu. Pri vyhľadávaní sa štandardne nerozlišujú malé a veľké písmená. Pre nahradzovanie to platí rovnako. Hľadať môžeme aj cez navigačné tlačidlá. Takéto hľadanie je znázornené na obr. 2.9 pri hľadaní slova „svět“, na obrázku sa hľadaný text zapisuje do bunky pod poľom *Cena*.

Niektoré z príkazov kontextového menu stĺpca a riadka nachádzajú aj v skupine *Záznamy* v príkaze *Ďalšie*.

Kontextové menu tabuľky pri neoznačenom riadku či stĺpci ponúkne filtre. Filtre slúžia na zobrazenie iba tých záznamov, ktoré vyhovujú nami zadanej podmienke. Príkazom pre filtrovanie venovaná skupina *Zoradiť a filtrovať*. Filtre umožnia zobraziť len záznamy vyhovujúce nejakej podmienke, napríklad len knihy od konkrétneho autora.

Kontextové menu po kliknutí na bunku sa líši podľa typu poľa. V prípade textového typu nám kontextové menu pri mene autora „Francis“ ponúkne možnosti „rovná sa Francis“, „nerovná sa Francis“, „obsahuje Francis“ a „neobsahuje Francis“. Stačí si vybrať a v tabuľke sa zobrazia iba riadky vyhovujúce zvolenej podmienke. Pri číselnom poli budú ponúkané možnosti „rovná sa ...“, „nerovná sa ...“, „menšie ...“ a „väčšie ...“. Pri dátumoch sa ponuka zmení na „pred ...“ a „po ...“.

Môžeme postupne filtrovať aj na základe viacerých polí. V tom prípade platia podmienky zároveň.

Ak chceme filter zrušiť použijeme príkaz *Rozšírené/Vymazať všetky filtre* v skupine *Zoradiť a filtrovať*. Ak chceme zrušiť iba časť filtra, napríklad filtrovanie podľa ceny, tak pri neznačenom stĺpci klikneme pravým tlačidlom na niektorú cenu a z kontextového menu vyberieme *Vymazať filter z Cena*.

Ak filtre nechceme zrušiť iba ich potrebujeme dočasne vypnúť, použijeme príkaz *Prepnúť filter* v skupine *Zoradiť a filtrovať*. Druhou možnosťou je kliknúť v navigačných tlačidlách na tlačidlo za príkazom nového záznamu. Na obr. 2.9 sa nachádza pod stĺpcom *Jazyk*. Rovnako filtre opätovne zapneme.

V príkaze *Filter* v skupine *Zoradiť a filtrovať* alebo v kontextovom menu sa nachádza príkaz *Textové filtre*. V prípade iných typov to môžu byť číselné alebo dátumové filtre. Tieto filtre umožňujú zadať podmienku nezávisle od obsahu poľa, na ktoré sme klikli. Napríklad podmienku pre všetky knihy začínajúce písmenom A.

Obyčajný filter neumožňuje zadať dve podmienky pre jeden stĺpec. Nedá sa nastaviť ani podmienka „Všetky detektívky a ešte aj knihy od Dumasa“. Ak by sme totiž vybrali žáner = detektívky a autor = Dumas, tak by sme dostali filter „Všetky detektívky od Dumasa“. Tomuto filtru zrejme vyhovujú iné knihy ako sme očakávali. Medzi filtrami pri doteraz popísaných spôsoboch platí logická spojka *A zároveň*.

Pre tieto prípady existujú ďalšie možnosti filtrovania. Príkaz *Rozšírené/Rozšírený filter alebo zoradenie* umožní vytvorenie ľubovoľnej podmienky pomocou dotazu. Dotazmi sa zaoberá kapitola 3. Príkaz *Rozšírené/Filtrovanie podľa formulára* je prechodom medzi jednoduchým filtrom a dotazom. Umožňuje zostaviť viaceré podmienky pre jedno pole a tiež spájať podmienky pomocou logickej spojky *Alebo*. Ide o podmnožinu možností, ktoré poskytujú dotazy, preto sa na tomto mieste nebudeme týmto príkazom príliš zaoberať. V spomínanom príklade „Všetky detektívky a ešte aj knihy od Dumasa“ by sme podmienku pomocou formulára pre filtrovanie vytvorili nasledovne:

- V stĺpci *Žáner* vyberieme detektívka.
- V spodnej časti klikneme na kartu (uško) s názvom *Alebo* a tak sa prepneme do ďalšieho formulára. Kliknutím na karty môžeme medzi formulármi prepínať. Tu napíšeme do poľa Autor „dumas“. Access by nám mal postupne slovo ponúknuť.
- Všimneme si, že pribudla ďalšia karta *Alebo*. Tam by sme mohlo zadať ďalšiu podmienku.
- V rámci jedného formulára platia podmienky *A zároveň*. Medzi formulármi platí *Alebo*.
- Po nastavení podmienok klikneme na príkaz *Prepnúť filter* v skupine *Zoradiť a filtrovať*.

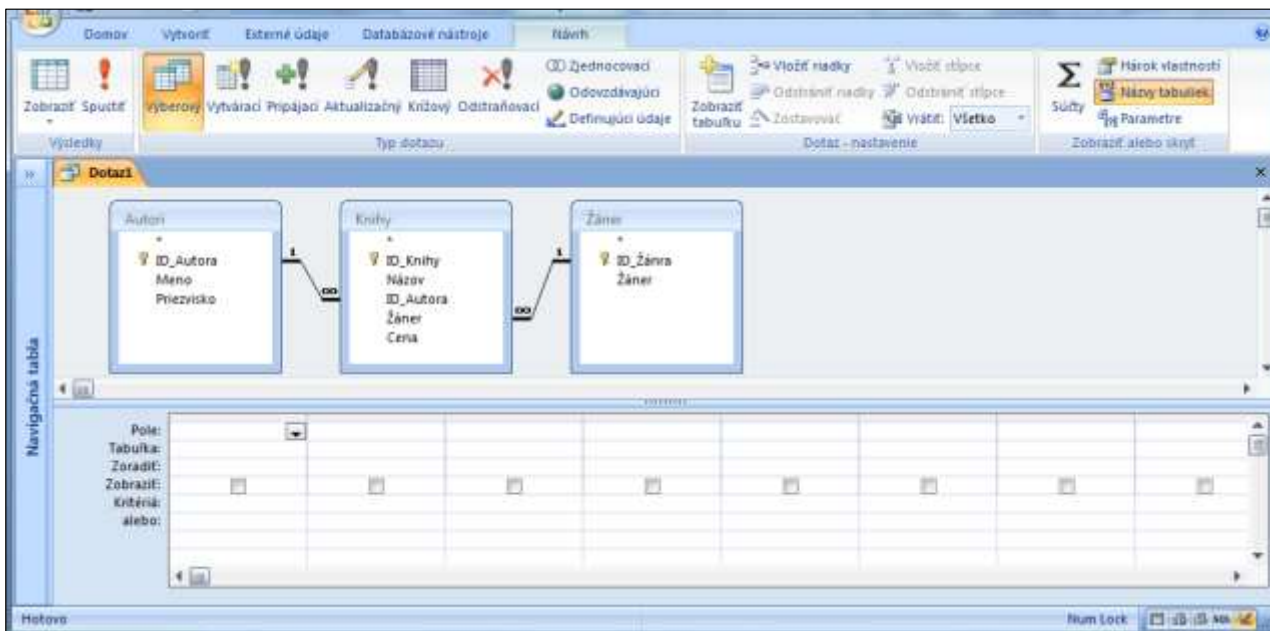
Ak pracujeme s filtrami a pri zatváraní tabuľky uložíme zmeny, tak sa po opätovnom otvorení tabuľky a použití filtra použije posledný filter pred uložením tabuľky. Filtre môžeme používať aj pri formulároch (formuláre sú vysvetlené v kapitole 4.).

V skupine *Záznamy* môžeme príkazom *Súčty* pridať na spodok tabuľky pridať riadok celkom, v ktorom môžeme pre každý stĺpec počítať súhrny. Pre textové polia je to iba počet. Pre dátumové počet, maximum, minimum a priemer. Číselný typ umožňuje oproti dátumovému navyše súčet a štatistické odchýlky.

Rôzne súhrny môžeme vytvoriť a zobrazit' aj pomocou *Zobrazenia kontingenčnej tabuľky* a *Zobrazenia kontingenčného grafu*. Tie sa ale často vytvárajú až po upravení tabuľky pomocou dotazu a preto sú popísané až v kapitole 3.7. na strane 45.

3. Dotazy

Na zisťovanie údajov z relačných databáz a na manipuláciu s nimi bol vytvorený jazyk SQL (Structured Query Language, štruktúrovaný dotazovací jazyk¹²). Tento jazyk má pomerne málo príkazov. Príkazy SQL sú jednoduché ale možno ich mnohonásobne kombinovať a tak nie je žiadnou zvláštnosťou vidieť výrazy zaberajúce mnoho riadkov. Takéto výrazy (skombinované príkazy) sa nazývajú dotazy – pomocou nich kladieme požiadavky databázovému serveru. Server nám (podľa typu dotazu) odpovie požadovanými údajmi alebo vykoná nejakú činnosť.



Obr. 3.1 Vytvorenie dotazu po výbere tabuliek

Výhodou dotazu je to, že po jeho vytvorení a uložení (pomenovanie a uloženie je rovnaké ako pri tabuľke) ho môžeme opakovane spúšťať. My alebo menej znalá obsluha databázy.

Je dôležité si uvedomiť, že údaje v databáze sa môžu meniť ale uložený dotaz môžeme spúšťať aj tak. Príkladom môže byť dotaz, ktorý po spustení zobrazí čitateľov meškajúcich s vrátením knihy – takýto dotaz by sa spúšťal zrejme každý deň na obsahovo rôznej databáze.

Aby bola práca s databázou pre používateľa ľahšia, MS Access umožňuje používať vizuálne rozhranie, ktoré dotaz zostaví a vykoná bez nutnosti znalostí SQL¹³. Na karte *Vytvoriť*, v skupine *Iné* sú možnosti *Sprievodca dotazom* a *Návrh dotazu*. My sa budeme zaoberať vytvorením dotazu len v návrhovom režime, kde možno vytvoriť všetky možnosti dotazu, dostupné aj pomocou sprievodcu. Sprievodcu by sme mohli výhodne použiť na vytvorenie dotazov pre nájdenie duplicitných alebo chýbajúcich záznamov.

Po kliknutí na tlačidlo *Návrh dotazu* sa zobrazí okno s názvom *Zobrazenie tabuľky*, v ktorom vyberieme tabuľku, s ktorou chceme pracovať. Môžeme pracovať aj s viacerými tabuľkami a dokonca aj s dotazmi, ktoré sme vytvorili predtým. Vyberáme si ich dvojklikom alebo označením a kliknutím na tlačidlo *pridať*.

Je veľmi dôležité aké tabuľky použijeme, či sú prepojené alebo nie alebo či použijeme niektorú tabuľku viackrát. Napríklad: Ak použijeme iba tabuľku *Knihy*, tak sa bude pracovať so všetkými knihami. Ak pridáme tabuľku *Požičané*, tak iba s požičanými (pretože sú tabuľky prepojené). Ak by sme použili tabuľku *Knihy* a s ňou nejakú nesúvisiacu (neprepojenú) tabuľku (nazvime ju X), tak by sa spracovala každá kniha (z tabuľky *Knihy*) toľkokrát, koľko je záznamov v tabuľke X. Preto je dôležité všimnúť si na príkladoch s dotazmi aj to aké tabuľky (alebo dotazy) sa používajú.

¹² Jazykový purista prekladajú query slovom dopyt. Tak tomu bolo aj v niekoľkých predchádzajúcich verziách MS Access. Vo verzii 2007 sa Microsoft vrátil k slovu dotaz.

¹³ Umožňuje používať aj SQL ale to by bolo na ďalšiu knihu – my sa SQL zaoberať v tejto knihe nebudeme.

Ak sme pridali všetky potrebné tabuľky a dotazy, okno *Zobrazenie tabuľky* zavrieme. Toto okno možno opätovne zobrazit' (a tak pridať zabudnutú tabuľku) pomocou karty *Návrh*, skupiny *Dotaz - nastavenia*, tlačidlom *Zobrazit' tabuľku*. Druhou možnosťou je kontextové menu po kliknutí pravým tlačidlom myši na plochu, kde sú zobrazené tabuľky. Ak omylom (alebo zámerne) pridáme tabuľku či dotaz navyše, môžeme ju z dotazu odstrániť klávesom *Delete* (po označení) alebo cez kontextové menu po stlačení pravého tlačidla myši na titulkovom pruhu tabuľky.

Na obr. 3.1 je znázornený stav po vybraní tabuliek. V hornej časti sú zobrazené tabuľky, ktoré v dotaze využívame. V spodnej časti sa zadávajú polia, s ktorými sa má pracovať a tiež podmienky, podľa ktorých sa údaje filtrujú. Na karte *Typ dotazu* - si musíme vybrať, aký dotaz ideme tvoriť. Na výber máme:

- **Výberový** - najpoužívanější typ, slúži na zobrazenie údajov na základe nami určených kritérií.
- **Křížový** - zobrazenie vo forme tabuľky so zoskupenými údajmi v riadkoch a stĺpcoch (v záhlaví tabuľky môžeme použiť iba zoskupenie podľa jedného poľa).
- **Akčné:**
 - **Vytvárací** - na základe kritérií uloží vyfiltrované údaje ako novú tabuľku.
 - **Aktualizačný** - upraví údaje podľa zadaných pravidiel.
 - **Pridávací** - pridá záznamy vyhovujúce kritériám z jednej tabuľky do inej tabuľky.
 - **Odstraňovací** - odstráni záznamy, ktoré vyhovujú daným podmienkam

Väčšina činností spojených s dotazmi môže byť vysvetlená na výberovom dotaze. Pri popise ostatných typov sú popísané iba rozdiely.

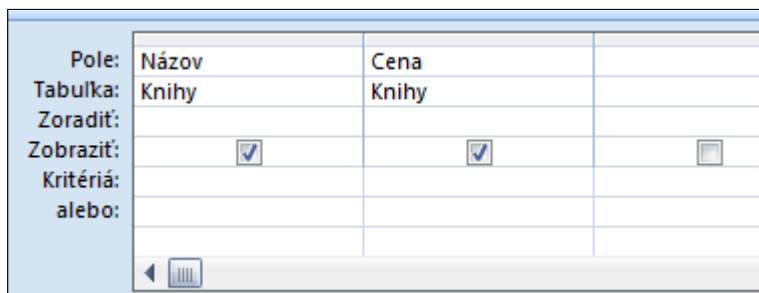
3.1. Výberový dotaz

Výberový dotaz umožňuje vybrať, ktoré stĺpce a ktoré riadky sa z tabuľky majú zobraziť. Vieme, že tabuľka *Knihy* obsahuje viacero stĺpcov. Môže sa stať, že budeme chcieť z tejto tabuľky zobraziť iba názov knihy a jej cenu. Možno to urobiť tromi spôsobmi:

1. Dvojklikom na *Názov* a *Cena* v tabuľke *Knihy* v hornej časti okna. Obe polia sa automaticky pridajú do riadku *Pole* v spodnej časti.
2. Metódou uchop a pusti (Drag and Drop) chytiť, potiahnuť a pustiť spomínané polia z tabuľky na niektorý stĺpec v dolnej časti.
3. Vybrať z ponuky v riadku *Pole* v spodnej časti.

Cieľ nášho snaženia je zobrazený na obr. 3.2. Ak sa pomýlime a pridáme iný stĺpec ako sme chceli môžeme ho samozrejme zmazať. To docielime kliknutím na záhlavie stĺpca (pri pohybe myši odspodu sa textový kurzor zmení na objektový – na šípku) a po označení stĺpca stlačíme kláves *Delete*. Druhou možnosťou je kliknúť hoci kde do stĺpca a na karte *Návrh* v skupine *Dotaz - nastavenie* kliknúť na *Odstrániť stĺpce*.

Okrem názvu polí v tabuľkách resp. v ponuke riadku *Pole* je možné vybrať aj „*“ resp. „Knihy.*“. Táto možnosť znamená výber všetkých stĺpcov danej tabuľky. Používa sa hlavne pri odstraňovanom dotaze.



Obr. 3.2 Výber stĺpcov v dotaze

Ak chceme zobraziť výsledok dotazu, tak na karte *Návrh* v skupine *Výsledky* tlačidlom *Zobrazit'* prepne do *Údajového zobrazenia*. Prepínať zobrazenia je možné aj tlačidlami v stavovom riadku alebo výberom z kontextového menu po kliknutí pravým tlačidlom myši na záložku dotazu. Pre akčné dotazy je v tejto skupine tlačidlo *Spustiť*, je to isté ako dvojklik na akčný dotaz v navigačnej table - **vykoná** sa dotaz. Ak chceme akčný dotaz upraviť otvoríme ho cez kontextové menu do návrhového režimu (Voľba *Otvoriť* v kontextovom menu spustí akčný dotaz tiež). Okrem zobrazení dotazu môžeme prepnúť do zobrazenia kontingenčnej tabuľky alebo grafu (sú popísané v 3.7) alebo zadať priamo príkazy v SQL.

Pomocou riadku *Zoradiť* môžeme výstup z dotazu utriediť podľa ľubovoľných stĺpcov. Triedenie sa vykonáva postupne zľava doprava. Ak by sme dali triediť v našom prípade na obrázku podľa oboch stĺpcov, tak by sa triedilo najskôr podľa názvu knihy a v prípade rovnakých názvov by sa takéto knihy utriedili podľa ceny.

Čo ale ak by sme chceli triediť hlavne podľa ceny a iba pri rovnosti cien triediť podľa názvu? V takom prípade máme dve možnosti:

- Mohli by sme vymeniť stĺpce. To docielime kliknutím na záhlavie stĺpca (pri pohybe myši odspodu sa textový kurzor zmení na objektový – na šípku) a metódou drag and drop stĺpec presunieme. Takto by bola cena v prvom stĺpci a preto by sa triedilo primárne podľa nej.
- Ak by sme ale požadovali aby sa názov nachádzal pri výpise na prvom mieste museli by sme použiť iný postup. Pridali by sme na prvú pozíciu ešte jeden stĺpec *Cena*. Podľa tohto stĺpca by sme triedili ale by sme ho nezobrazovali – v riadku *Zobrazit'* by sme tú možnosť odznačili. Na obr. 3.3 je takýto dotaz znázornený.

Vždy po zmene vo výberovom dotaze v návrhovom režime je dobré skontrolovať výsledok pomocou zobrazenia v údajovom režime.

| | | | |
|------------|--------------------------|-------------------------------------|-------------------------------------|
| Pole: | Cena | Názov | Cena |
| Tabuľka: | Knihy | Knihy | Knihy |
| Zoradiť: | Zostupne | Vzostupne | |
| Zobrazit': | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | | |
| alebo: | | | |

Obr. 3.3 Triedenie a zobrazovanie stĺpcov v dotaze

Pomocou podmienok v riadkoch *Kritéria* určujeme, ktoré záznamy budú vo výsledku dotazu zahrnuté. Podmienky môžeme písať do viacerých riadkov. Podmienky, ktoré sú v jednom riadku, musia platiť **zároveň**. Ak je riadkov viacero stačí, ak sú splnené podmienky v jednom riadku.

Začneme jednoduchou požiadavkou – zobraziť iba knihy lacnejšie ako 10€. Do stĺpca *Cena* riadku *Kritéria* napíšeme <10. Výsledok možno hneď skontrolovať v údajovom zobrazení. Ak by sme chceli knihy drahšie, podmienka by bola >10. Ak by nás zaujímali aj hraničné situácie použili by sme <=10 resp. >=10. Pre knihy v cene 10€ stačí napísať iba číslo. Pre všetky knihy okrem 10 eurových <>10 alebo Not 10.

| | | | |
|------------|-------------------------------------|-------------------------------------|--|
| Pole: | Názov | Cena | |
| Tabuľka: | Knihy | Knihy | |
| Zoradiť: | | | |
| Zobrazit': | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| Kritériá: | | >10 And <20 | |
| alebo: | | | |

Obr. 3.4 Dotaz filtrujúci knihy s cenou od 10€ po 20€

Čo ak by sme chceli knihy v rozpätí 10€ až 20€? Máme dve možnosti:

- Vložíme do dotazu dva stĺpce s cenou (druhý nezobrazíme) a v jednom dáme podmienku >10 a v druhom do toho istého riadku <20 . Povedali sme, že podmienky v jednom riadku musia platiť zároveň. Preto takejto zloženej podmienke vyhovujú všetky záznamy, v ktorých je cena väčšia ako 10€ a zároveň menšia ako 20€.
- Praktickejšie je vložiť celú túto podmienku do jedného stĺpca pomocou využitia logických spojok, tak ako je to znázornené na obr. 3.4. V našom prípade by teda podmienka vyzerala >10 And <20 .¹⁴ Iný zápis je *Between 10 and 20*.

Tab. 3.1 zobrazuje pravdivostné hodnoty výrazov *AND* a *OR*. Je dobré si všimnúť, že matematické *AND* je obdobou „ľudského“ „a zároveň“. *OR* sa ale od bežnej reči líši. Ak povieme, že do kina pôjdeme v pondelok alebo utorok, nepredpokladá sa, že pôjdeme v oboja dni. V bežnej reči sa totiž používa spojka *exclusive OR*, t. j. celý výraz platí iba ak platí práve jedno z A a B.

Pri *OR* teda stačí aby bola vo výraze splnená jedna podmienka. Pri *AND* musia byť splnené obe (všetky).

Tab. 3.1. Tabuľka pravdivostných hodnôt And a Or

| A | B | A or B | A and B |
|---|---|--------|---------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

Pri textových a memo poliach môžeme hľadať:

- Konkrétne hodnoty, napríklad ak chceme zobrazit' všetky knihy s názvom Máj, tak do riadku kritéria napíšeme "Máj" (môžeme napísať aj "máj"). V prípade, že úvodzovky nenapíšeme, tak ich (v takomto jednoduchom prípade) Access doplní sám. V zložitejších podmienkach to ale nemusí dokázať alebo ich doplní nesprávne.
- Ak hľadáme viacero kníh môžeme použiť príkaz In("Máj"; "Duna"; "Jano").
- Znak "*" nahradzuje pri hľadaní (nielen v Accesse) ľubovoľnú skupinu znakov. To môžeme využiť pri hľadaní kníh, ktorých názov nevieme presne, napríklad:
 - Ak poznáme iba začiatok názvu, tak zadáme ako podmienku "rok*" ¹⁵. Môžeme nájsť knihu „Rok aktívneho slnka“ ale aj „Roky v kine Úsmev“.
 - Ak vieme, že názov knihy končí na rok, zadáme "*rok". Môžeme tak nájsť knihu „Búrliový rok“ ale aj „Čas slnka a búrok“.
 - Ak sa rok nachádza v názve, použijeme "*rok*". Tejto podmienke vyhovujú knihy z predchádzajúcich dvoch odrážok, „Štyri roky vojny“ a tiež napríklad „Krokodíl Ham“.
- Podobne môžeme použiť znak "?", ktorý nahrádza práve jeden znak. Napríklad podmienku "??o??" spĺňajú knihy „Úroda“ a „Slovo“.
- Môžeme používať aj znaky nerovnosti. Podmienka $>"m"$ vyhovuje len knihám s názvom, ktorý je v abecede za písmenom „M“.
- Predstavme si situáciu, že chceme nájsť autorov, ktorí nemajú uvedené krstné meno (hodnota je *Null*). Pre takéto vyhľadávanie môžeme použiť podmienku *Is Null*. Negáciou - opakom je *Is Not Null*.

¹⁴ Pozor na častú chybu >10 Or <20 . To možno čítať: Všetky záznamy, v ktorých je cena väčšia ako 10€ a ešte aj tie v ktorých je menšia ako 20€. Táto podmienka je splnená vždy – pre každú knihu platí, že je drahšia ako 10€ alebo je lacnejšia ako 20€. Výraz >20 Or <10 je ale v poriadku. Čo znamená?

Podobne je častou chybou <10 And >20 , táto podmienka nie je splnená nikdy. Žiadna kniha nestojí menej ak 10€ a zároveň viac ako 20€.

¹⁵ Access upraví takto zadanú podmienku na Like „rok*“.

- Pretože ale môžu existovať autori, ktorí krstné meno nemajú, ale im nechýba (t.j. nemajú ho zadané, pretože ho nemajú), tak je potrebné to rozlišovať. Preto sa používa prázdny reťazec: "" = nič a okolo neho úvodzovky. Takýmto autorom by sme zadali namiesto mena prázdny reťazec a pri hľadaní všetkých s krstným menom by sme použili podmienku *Is Not Null And <> ""*.
- Podmienka "[d-f]*" nájde všetky knihy začínajúce písmenom D, E alebo F.

Podmienky môžeme kombinovať. Základné pravidlo pritom znie: Riadky sa spájajú s OR a podmienky v rámci jedného riadku s AND.

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Pole: | Názov | Cena |
| Tabuľka: | Knihy | Knihy |
| Zoradiť: | | |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | Like "[d-f]*" | >5 And <8 |
| alebo: | Like "Ja*" | |
| | | 9 |
| | Like "*rok*" | >20 |

Obr. 3.5. Príklad dotazu so zloženou podmienkou

Na obr. 3.5 je znázornený dotaz s podmienkou, ktorej vyhovujú knihy spĺňajúce **aspoň jednu** z podmienok:

- Začínajú písmenom D, E alebo F, a zároveň stoja viac ako 5€ ale menej ako 8€.
- Začínajú na „Ja“ (alebo „ja“). (Cena môže byť ľubovoľná.)
- Ich cena je 9€. (Názov môže byť ľubovoľný.)
- V ich názve je text „rok“ (nemusí to byť samostatné slovo) a zároveň sú drahšie ako 20€.

Podobne ako text označujeme úvodzovkami, pre dátumy používame znak #. V jednoduchých textových výrazoch úvodzovky môžeme vynechať (doplnia sa automaticky), ale pri dátumoch ich musíme použiť vždy. Napríklad:

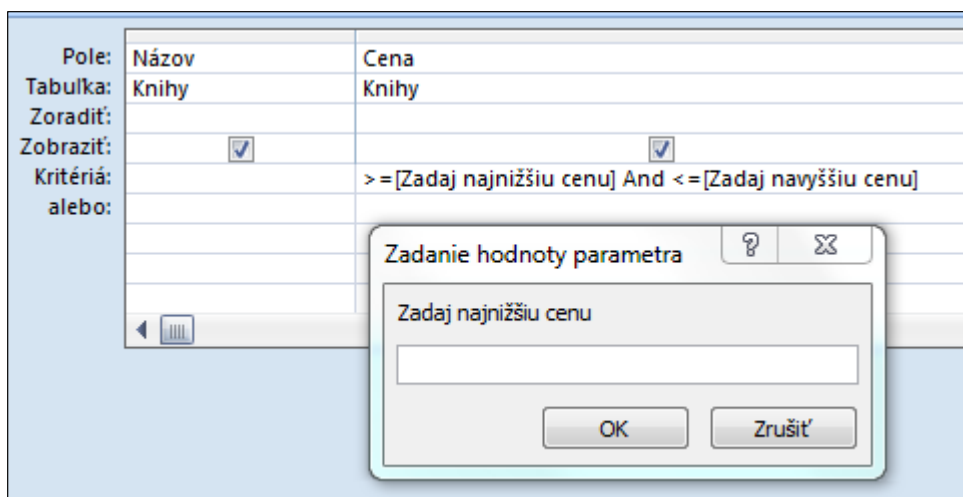
- <#4.7.2010# - tejto podmienke vyhovujú záznamy staršie 4. júlu 2010.
- Môžeme používať aj logické spojky >#15.5.2009# And <#21.5.2009#.
- Často sa používa funkcia Date(), ktorá vráti aktuálny dátum. Podmienke < Date() - 30 vyhovujú záznamy staršie ako 30 dní.

Pri hľadaní podľa stĺpca s logickou hodnotou, môžeme používať *True/On/Yes* resp. *False/Off/No*.

3.1.1. Parameter v dotaze

Predstavme si situáciu, že hľadáme knihy v cene od 5€ do 10€. Už vieme, že do stĺpca cena zadáme ≥ 5 And ≤ 10 . Takýto dotaz je ale príliš konkrétny, ak by sme hľadali knihy v inom rozpätí cien, tak by bol nepoužiteľný.

Preto sa používajú parametre. Namiesto konštánt 5 a 10 z vyššie uvedeného príkladu vytvoríme dotaz tak, aby sa na spodnú a hornú hranicu ceny knihy opýtal. Našťastie je to jednoduché, stačí ak v podmienke namiesto čísla 5 a 10 uvedieme v hranatých zátvorkách text. Po prepnutí do údajového zobrazenia (alebo spustení dotazu) sa text zobrazí a používateľ je vyzvaný, aby zadal hodnotu. Použitie aj otázka po spustení dotazu sú znázornené na obr. 3.6.



Obr. 3.6 Dotaz s parametrom

Ak by sme mali v podmienkach v hranatých zátvorkách viackrát ten istý text, otázka sa nebude opakovať a použije sa pre všetky výskyty tá istá hodnota. V prípade, že používateľ zadá iný typ údajov ako sa v podmienke očakáva, tak sa dotaz nevykoná a vypíše sa chyba.

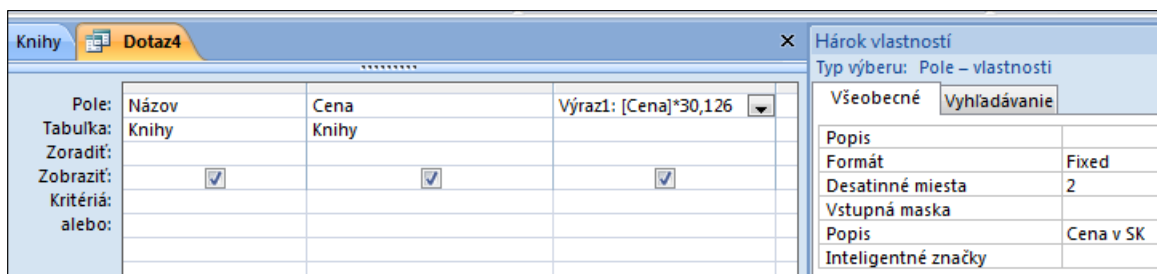
Predist' vloženiu iného typu možno zdefinovaním typu parametra v okne *Parametre dotazu*. Okno zobrazíme v karte *Návrh*, skupine *Zobraziť alebo skryť* stlačením *Parametre*. V ľavom stĺpci zadáme názov parametra a v pravom vyberieme typ údajov.

Text nemôže byť zhodný s názvom poľa. V našom prípade napríklad [Cena]. V takom prípade by sa Access nespýtal na hodnotu a namiesto toho by sa postupne dosadzovala cena knihy aktuálneho riadku. Využitie takéhoto postupu si ukážeme v nasledujúcom príklade.

3.1.2. Vytvorenie nového stĺpca

Predstavme si situáciu, že chceme zobraziť cenu kníh v bývalých slovenských korunách. Najlepšie tak, aby sme mali tri stĺpce *Názov*, *Cena v eurách* a *Cena v korunách*.

Pri vytváraní nového dotazu vyberieme tabuľku *Knihy*. Do dotazu pridáme stĺpce *Názov*, *Cena*. Do tretieho stĺpca do riadku pole napíšeme vzorec pre výpočet ceny v SK, t. j. „[Cena] * 30,126“. Pri zobrazení dotazu v údajovom režime vidíme, že v každom riadku sa cena vynásobila konverzným kurzom. Chybou na kráse je pomenovanie stĺpca *Výraz1* a rôzny počet desatinných miest. To ľahko upravíme pomocou *Hárku vlastností*¹⁶, ktorý nájdeme v karte *Návrh*, skupine *Zobraziť alebo skryť*. Využitie je zobrazené na obr. 3.7 - po kliknutí na tretí stĺpec. Takto môžeme nastaviť vlastnosti každého stĺpca. Samotné pomenovanie môžeme zmeniť aj prepísaním slova „Výraz1:“ v dotaze v riadku pole. Ak klikneme mimo využívaného stĺpca, môžeme nastavovať vlastnosti pre celý dotaz.



Obr. 3.7 Vytvorenie nového stĺpca pomocou dotazu

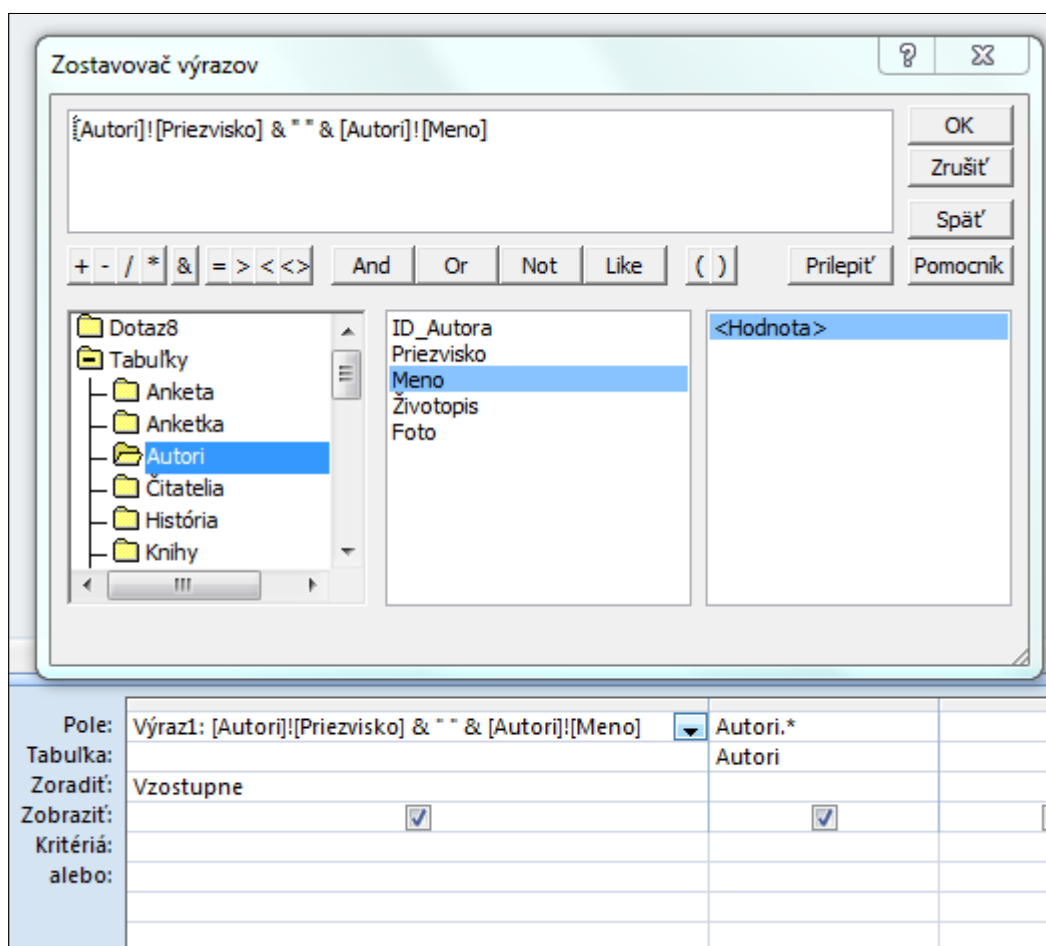
¹⁶ Vo vlastnostiach vidíme dvakrát *Popis*. Ide o nedokonalosť prekladu. Prvé (originálne description) je text, ktorý sa zobrazuje napríklad v údajovom zobrazení po kliknutí do stĺpca v stavovom riadku. Druhý popis (originálne caption) je titulok poľa.

Pri prepojení tabuliek *Autori* a *Knihy* resp. *Čitatelia* a *Požičané* sme sa stretli s problémom, že sa nám zobrazovalo iba priezvisko autora alebo čitateľa. Preto môžeme vytvoriť dotaz ktorý spojí meno a priezvisko a tento dotaz využiť namiesto tabuľky *Autori*, resp. *Čitatelia*. Na obr. 3.8 je tento dotaz znázornený.

Mohli sme do prvého stĺpca napísať *priezvisko & " " & meno* ale použili sme *Zostavovač výrazov*, ktorý je zobrazený v hornej časti obrázka. Možno ho zobrazíť po kliknutí pravým tlačidlom do riadku pole príkazom *Zostaviť* z kontextového menu alebo nájsť v skupine *Dotaz - nastavenia*.

V *Zostavovači výrazov* sa po dvojkliku na *Tabuľky* zobrazia dostupné tabuľky. Po kliknutí na tabuľku *Autori* sa zobrazia vedľa polia dostupné v tejto tabuľke. Po dvojkliku na *Priezvisko* sa do hornej časti vloží *[Autori]![Priezvisko]* čo je presné určenie poľa¹⁷. Potom dvojklikom na symbol *&* vložíme znak na spojenie textov. Ak by sme teraz vložili meno, tak by sa pridalo k priezvisku a vznikol by text bez medzery napríklad „PeterajKamil“. Preto za priezvisko vložíme medzeru (v úvodzovkách) - " " a ďalší *&*. Potom už môžeme dvojklikom pridať pole *Meno*.

Podobne ako v predchádzajúcom prípade môžeme pridať popis. A prípadne ďalšie polia. Na obrázku sme pridali všetky polia z tabuľky *Autori* - *Autori.**. Tento dotaz môžeme vybrať pri prepájaní tabuliek, keď spájame tabuľky *Autori* a *Knihy* namiesto tabuľky *Autori* a namiesto polí *Priezvisko* a *Meno* vybrať len jedno (v našom prípade) *Výraz1*. Týmto spôsobom sme vyriešili problém zobrazovania iba jedného poľa pri prepojených tabuľkách.



Obr. 3.8. Dotaz pre spojenie dvoch polí do jedného

3.1.3. Agregačné funkcie

Na karte *Návrh*, v skupine *Zobrazit' alebo skryť* sa nachádza dôležitý príkaz *Súčty*. Pridá do okna dotazu riadok *Celkom*. Pomocou neho môžeme v dotaze zoskupovať riadky do skupín a vytvárať rôzne skupinové

¹⁷ V tomto prípade je to zbytočne presné lebo používame len tabuľku *Autori*. Ak by sa ale v dotaze vyskytovala aj tabuľka *Čitatelia*, tak by pole s názvom *Priezvisko* nebolo jednoznačne určené. Preto *Zostavovač výrazov* používa vždy úplné určenie.

výpočty. Napríklad zisťovať, koľko kníh máme od toho ktorého autora, ktorý čitateľ si požičal najviac kníh alebo ktorá kniha bola najčastejšie požičaná.

Ak by sme teda chceli vedieť, koľko kníh máme od každého autora, vytvoríme výberový dotaz, pridáme tabuľku *Knihy*, stlačíme spomínané tlačidlo *Súčty* a do dotazu pridáme dvakrát stĺpec: *ID_Autora*. V prvom stĺpci v riadku *Celkom* vyberieme možnosť *Group By* (zoskupiť). V druhom stĺpci, v tom istom riadku vyberieme možnosť *Count* (počet). Access teda vie, že má vytvoriť skupiny kníh podľa autora a spočítať koľko ich je v každej skupine.

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Pole: | ID_Autora | Cena |
| Tabuľka: | Knihy | Knihy |
| Celkom: | Group By | Count |
| Zoradiť: | | Zostupne |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | |
| alebo: | | |

Obr. 3.9 Dotaz pre zistenie koľko kníh máme od toho ktorého autora

Dotaz (s mierne iným druhým stĺpcom) je znázornený na obr. 3.9. Pretože určujeme počet riadkov (*Count*) v tabuľke, tak je (skoro) jedno či spočítavame ID, ceny alebo názvy. **Pozor!!** Ak pri niektorej knihe nie je uvedená cena (je *Null*), tak dotaz tú knihu nezapočíta. Teda je jedno, ktorý stĺpec, ale musí mať vyplnené všetky riadky (ak chceme spočítať všetky). Preto používame pokiaľ možno stĺpec s nastavenou vlastnosťou *Povinný* = Áno. Najpohodlnejšie je používať stĺpec, ktorý je primárnym kľúčom.

Malou úpravou dotazu z obr. 3.9 môžeme pre každého autora vypočítať priemernú cenu jeho kníh, resp. celkovú cenu jeho kníh. Stačí ak v riadku *Celkom* vyberieme *Avg* (average, priemer), resp. *Sum* (summary, súčet). Obdobne by sme vybrali *Min* alebo *Max* pre najlacnejšiu alebo najdrahšiu knihu každého autora.

Vráťme sa k príkladu z obr. 3.9. Ak by sme chceli od každého autora zisťovať počet kníh drahších ako 30€. Prvý pokus by bol asi zlý a vyzeral by ako dotaz na obr. 3.10. Tento dotaz robí niečo iné – zisťuje, od ktorých autorov máme viac ako 30 kníh. V druhom stĺpci nie je totiž cena ale počet kníh.

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Pole: | ID_Autora | Cena |
| Tabuľka: | Knihy | Knihy |
| Celkom: | Group By | Count |
| Zoradiť: | | Zostupne |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | >30 |
| alebo: | | |

Obr. 3.10 Pozor na zlé použitie podmienky v súčtovom dotaze

Oprava je znázornená na obr. 3.11. Pridali sme stĺpec *Cena* s podmienkou >30. Pre podmienku musíme v riadku *Celkom* zvoliť *Where* (kde). Tento stĺpec ale nemožno zobraziť! Dôvodom je to, že každého autora uvádzame jedenkrát (ak má aspoň jednu knihu drahšiu ako 30€). Ak by mal viac kníh spĺňajúcich podmienku nebolo by jasné, ktorú cenu z nich uviesť.

| | | | | |
|-----------|-------------------------------------|-------------------------------------|---|--------------------------|
| Pole: | ID_Autora | Cena | ▼ | Cena |
| Tabuľka: | Knihy | Knihy | | Knihy |
| Celkom: | Group By | Count | | Where |
| Zoradiť: | | Zostupne | | |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | <input type="checkbox"/> |
| Kritériá: | | | | >30 |
| alebo: | | | | |

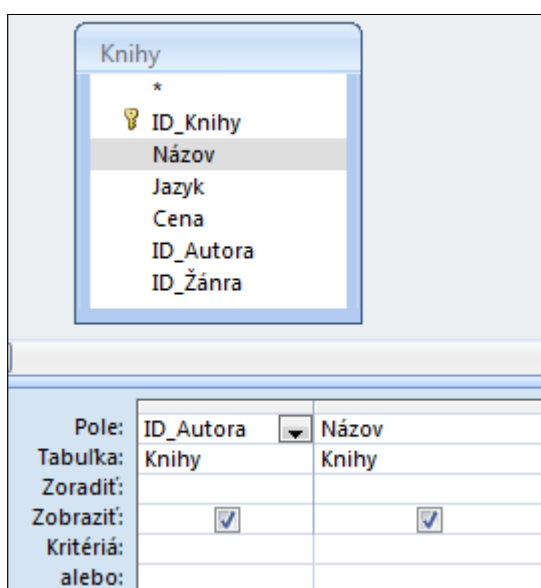
Obr. 3.11 Koľko kníh drahších ako 30€ máme od každého autora?

Ak by sme namiesto *Where* vybrali *Min*, *Max*, *First* (prvý), *Last* (posledný) *StDev* (smerodajná odchýlka) alebo *Var* (odchýlku) bola by hodnota v treťom stĺpci jednoznačne určená a stĺpec s cenou by sme mohli zobrazit'.

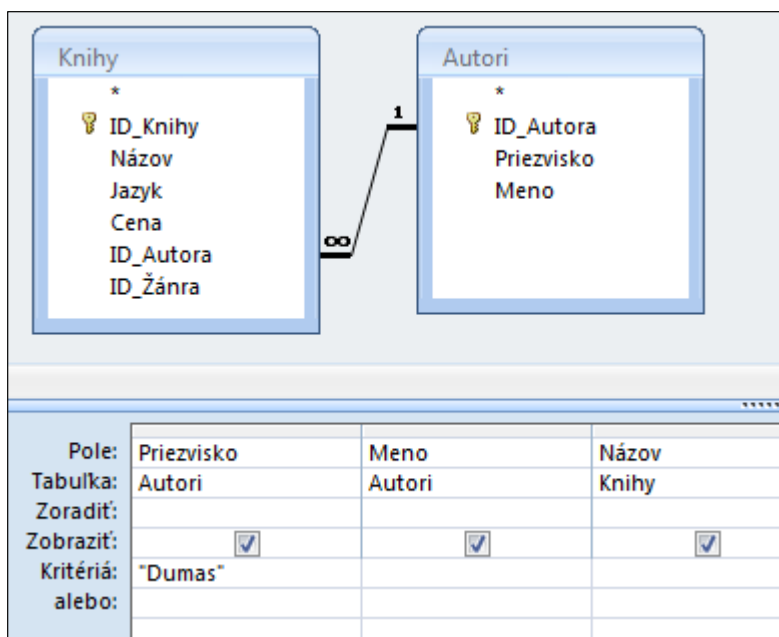
3.1.4. Dotaz využívajúci viacero tabuliek

Dotaz na obr. 3.12 zobrazí autora a názov knihy všetkých kníh v knižnici. V skutočnosti v prvom stĺpci nie je priezvisko autora, ale číslo – kľúč z tabuľky *Autori*. Access automaticky na základe tohto kľúča zobrazuje priezvisko. Tento fakt si je dôležité uvedomiť.

Ak by sme napríklad chceli zobrazit' iba knihy od Dumasa, tak by sa nám neznalosť tohto faktu mohla vypomstiť. Ak by sme v tomto dotaze do prvého stĺpca do riadka kritéria napísali „Dumas“, tak nielenže nám dotaz nič nenájde, ale ešte aj vypíše chybu. V stĺpci je totiž očakávané číslo a nie text. Preto musíme do dotazu pridať tabuľku *Autori* a hľadať v nej. Situácia je znázornená na obr. 3.13. Všimnite si, že meno a priezvisko sa berú z tabuľky *Autori* a názov knihy z tabuľky *Knihy*.



Obr. 3.12 V dotaze je *ID_Autora* a dotaz zobrazuje meno autora



Obr. 3.13 Dotaz využívajúci viacero tabuliek

3.2. Vytvárací dotaz

Výberový dotaz dáta neukladal. To bola výhoda, pretože vždy pracoval s aktuálnou tabuľkou. Niekedy ale potrebujeme výsledok dotazu v podobe tabuľky, napríklad ak sa obsah tabuľky časom mení a my si chceme uschovať jej terajší stav.

Na takéto účely možno použiť vytvárací dotaz. Pre jeho vytvorenie platí to čo pre výberový dotaz s dvoma odlišnosťami v postupe.

1. Po kliknutí na tlačidlo *Vytvárací* v skupine *Typ dotazu* musíme určiť do akej tabuľky a do ktorej databázy sa majú uložiť údaje (ktoré by sa pri výberovom dotaze iba zobrazili). Predpokladá sa, že určíme novú tabuľku. Ak zadáme názov existujúcej tabuľky, tak sa pôvodný obsah tejto tabuľky po spustení dotazu nahradí výsledkom nášho dotazu.
2. Pri prepnutí z návrhového zobrazenia do údajového sa výsledok dotazu iba zobrazí – tak ako pri výberovom dotaze. To môžeme využiť na kontrolu, či dotaz funguje podľa našich predstáv. Ak dotaz funguje správne stlačíme tlačidlo *Spustiť* v skupine *Výsledky*. Access sa pokúsi vytvoriť tabuľku a uložiť do nej výsledky dotazu. Ak sme pri vytváraní dotazu zadali názov existujúcej tabuľky objaví sa výstraha. Ak je cieľová tabuľka otvorená, musíme ju najskôr zatvoriť.

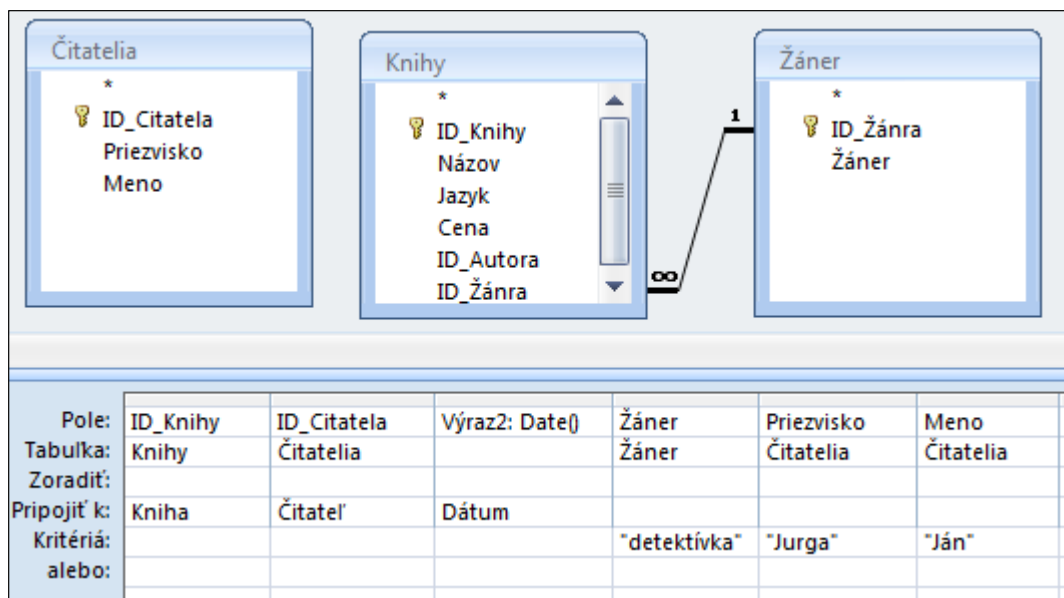
3.3. Pripájací dotaz

Pripájací dotaz funguje obdobne ako vytvárací ale s dvoma odlišnosťami v postupe:

1. Po kliknutí na tlačidlo *Pripájací* v skupine *Typ dotazu* musíme určiť k akej tabuľke a do ktorej databázy sa majú pripojiť údaje. Musíme určiť existujúcu tabuľku. Ak tabuľka neexistuje Access neskôr pri spustení dotazu oznámi chybu.
2. Riadok *Zobraziť* sa zmení na *Pripojiť k*. Existujúca tabuľka musí obsahovať nejaké stĺpce a pomocou riadku *Pripojiť k* určíme čo pridáme do ktorého stĺpca.

Napríklad, ak by si čitateľ Ján Jurga (jediný s takýmto menom v našej tabuľke čitateľa) požičal dnes všetky detektívky. Potrebujeme tieto výpožičky zapísať do tabuľky *Požičané*. Táto tabuľka má tri stĺpce, v ktorých evidujeme ID čitateľa a knihy a dátum výpožičky.

Na obrázku obr. 3.14 v prvých troch stĺpoch vidno, ako povieme Accessu, že má pridať *ID_Knihy* zapísať do stĺpca *Kniha*, *ID_Čitateľa* do stĺpca *Čitateľ* a dnešný dátum (použili sme funkciu *Date()*, funkcie sú popísané v kapitole 7.) do stĺpca *Dátum*. Štvrtým stĺpcom zabezpečíme, aby sa pridali všetky detektívky. Ak by sme poznali ID (číslo) Jána Jurgu, tak by sme ho mohli napísať do druhého stĺpca do riadku *Pole* (riadok *Tabuľka* by bol prázdny). Posledné dva stĺpce by boli zbytočné. A museli by sme odstrániť z dotazu aj tabuľku *Čitateľa*. Ak by sme tabuľku z dotazu neodstránili vykonal by sa dotaz toľkokrát, koľko je v tabuľke čitateľov. T. j. ak je detektívok 29 a čitateľov 92, pribudlo by v tabuľke *Požičané* 2668 riadkov.



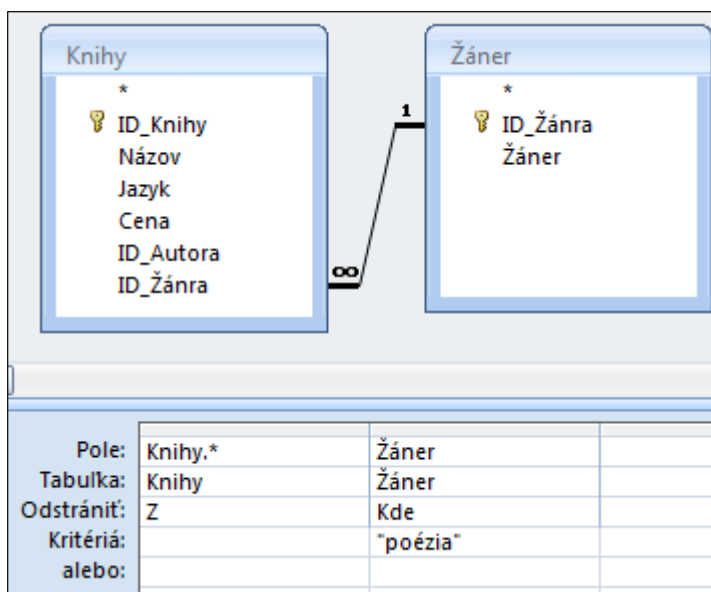
Obr. 3.14 Pridávací dotaz

3.4. Odstraňovací príkaz

Predpokladajme, že z tabuľky *Knihy* chceme odstrániť všetky knihy s poéziou. Napríklad preto, že ich chceme presunúť do špecializovanej pobočky našej knižnice.

Ak máme vo vzťahoch pri referenčnej integrite nastavené *Kaskádové odstránenie súvisiacich záznamov*, tak nám stačí zmazať v tabuľke *Žáner* riadok poézia. Druhým spôsobom je využitie odstraňovacieho dotazu.

Hlavným rozdielom oproti výberovému dotazu je zmena riadka *Zobraziť* na *Odstrániť*. V tomto riadku máme na výber z dvoch možností *Z* a *Kde*. Výber *Kde* slúži na určenie podmienky, t. j. ktoré záznamy sa majú zmazať. Inak sa zmažú všetky záznamy. Voľba *Z* určuje, v ktorej tabuľky budeme záznamy mazať. Pretože sa maže celý záznam musíme v riadku *Pole* vybrať hviezdičku. Dotaz pre zmazanie všetkých kníh s poéziou je znázornený na obr. 3.15.



Obr. 3.15 Dotaz pre odstránenie kníh s poéziou

3.5. Aktualizačný dotaz

Zrejme najčastejšie využívaným akčným dotazom je aktualizálny dotaz. V našej knižnici by sme ho mohli využiť pri úprave cien kníh. Napríklad by sme mohli cenu detektívok zvýšiť o 2€ alebo znížiť cenu kníh, ktoré boli požičané za posledný rok viac ako päťkrát o 5% (opotrebovanie). Alebo by sme mohli pridať na koniec každého názvu knihy bodku alebo ju odobrať.

Podobne ako v predchádzajúcich dotazoch aj v aktualizacom je riadok *Zobrazit'* nahradený iným. V tomto prípade je to riadok *Aktualizovať do*, do ktorého sa píše nová (aktualizovaná) hodnota.

Pozrime si dva dotazy na obr. 3.16 a obr. 3.17. Na prvom je častá chyba pri pokuse zvýšiť cenu o 2€. V skutočnosti sa do poľa *Cena* v každom riadku, kde je žánrom detektívka zapíše +2, t. j. 2. Ak by sme namiesto +2 napísali -2 (v snahe znížiť cenu o 2€) nastavili by sme cenu na -2€.

| | | |
|------------------|--------------|-------|
| Pole: | Žáner | Cena |
| Tabuľka: | Žáner | Knihy |
| Aktualizovať do: | | +2 |
| Kritériá: | "Detektívka" | |
| alebo: | | |

Obr. 3.16 Aktualizačný dotaz nastavujúci detektívkam cenu 2€

Správny postup pre zvýšenie ceny je použiť aktuálnu cenu, t. j. [*Cena*] a k nej pripočítať 2. Ak by sme chceli po zvýšení o 2€ cenu vrátiť naspäť stačí vytvoriť nový aktualizálny dotaz, kde by bolo [*Cena*] - 2. Je dobré si všimnúť, že po spustení dotazu z obr. 3.16 sa k pôvodným cenám už nedopracujeme (ak sme tabuľku nezalohovali).

| | | |
|------------------|--------------|-------------------|
| Pole: | Žáner | Cena |
| Tabuľka: | Žáner | Knihy |
| Aktualizovať do: | | [<i>cena</i>]+2 |
| Kritériá: | "Detektívka" | |
| alebo: | | |

Obr. 3.17 Aktualizačný dotaz zvyšujúci cenu detektívkam o 2€

Ak by sme do riadku *Aktualizovať do* napísali [*Cena*]+[*Zadaj hodnotu navýšenia*] dostali by sme parametrický aktualizálny dotaz. Vždy po spustení by sme zadaním kladného čísla určili, o koľko sa cena detektívok zvýši a zadaním záporného o koľko sa zníži.

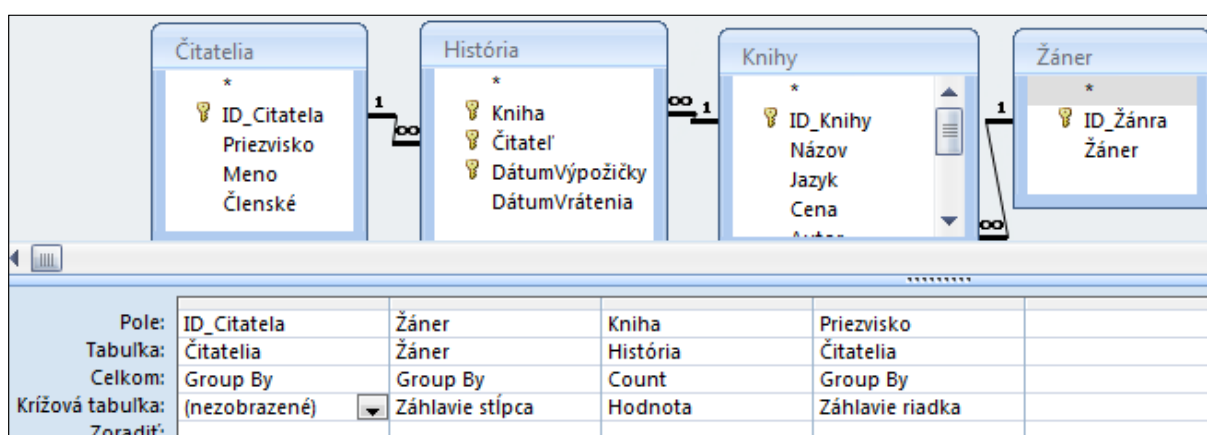
Ak by sme chceli zvýšiť cenu o 10%, tak by sme pôvodnú cenu vynásobili 1,1. Teda $[Cena] * 1,1$. Ak by sme chceli cenu o 10% znížiť, tak by sme pôvodnú cenu vynásobili 0,9.

Ak sme cenu zvýšili o 10%, a chceme ju vrátiť na pôvodnú tak ju vydělíme 1,1 ($[Cena] / 1,1$). Podobne po znížení o 10% ju na pôvodnú hodnotu dostaneme $[Cena] * 0,9$. V týchto prípadoch sa ale môže stať, že kvôli zaokrúhľovaniu nemusíme dostať presne pôvodný výsledok.

3.6. Krížový dotaz

Krížový dotaz sa troška podobá výberovému dotazu so súčtami. Krížový dotaz ale zoskupuje údaje nielen v riadkoch ale aj v stĺpci - vytvorí tabuľku. Oproti výberovému dotazu so súčtami pribudne v krížovom dotaze riadok *Krížová tabuľka*, v ktorom musíme určiť minimálne jedno záhlavie riadkov, jedno stĺpcov a jedno pole, ktoré bude obsahom tabuľky (*Hodnota*). Zatiaľ čo v záhlaví riadka sa zobrazí pri prepojených poliach (cudzíe kľúče) hodnota zobrazovaná cez sprievodcu prehládávaním, tak v záhlaví stĺpcov sa zobrazí vždy skutočná hodnota.

Ak by sme chceli zistiť, aké žánre si požičiavajú jednotliví čitatelia, mohli by sme vytvoriť krížový dotaz tak, ako je zachytený na obr. 3.18. Výsledok po spustení dotazu je na obr. 3.19.



Obr. 3.18. Krížový dotaz zisťujúci požičiavanie žánrov jednotlivými čitateľmi

Stojí za povšimnutie, že medzi tabuľkami v dotaze je aj tabuľka *Knihy*, ktorá sa v samotnej definícii dotazu (v druhom riadku) nenachádza. Služí na prepojenie žánra s knihou v tabuľke *História*. Bez tohto prepojenia by dotaz nemal zmysel a výsledky by boli zlé – zobrazili by každému čitateľovi v každom žánri rovnaké číslo – počet požičaných kníh daného čitateľa.

| Priezvisko | Detektívka | odborná literatúra | poézia | próza |
|------------|------------|--------------------|--------|-------|
| Tkáčik | 4 | | | 11 |
| Uličná | | | 1 | 3 |
| Valach | 2 | 1 | 1 | 4 |
| Verešová | | | 1 | |
| Vráblová | 1 | | | 3 |
| Adamková | 1 | | | |
| Bača | | | | 3 |
| Jurga | 22 | | | |

Obr. 3.19. Výsledok krížového dotazu zisťujúceho požičiavanie žánrov jednotlivými čitateľmi

3.7. Kontingenčná tabuľka a graf

Krížové tabuľky sú podobné kontingenčným tabuľkám. Kontingenčné tabuľky ponúkajú pohodlný spôsob zoskupovania údajov a vytváranie množstva rôznych súhrnov niekoľkými kliknutiami. Aj keď sa *Zobrazenie kontingenčnej tabuľky* dá zvoliť aj pri obyčajnej tabuľke, väčšinou je potrebné použiť dotaz, ktorý prepájané tabuľky spojí. Treba si totiž uvedomiť, že napríklad v tabuľke *História* nie je meno čitateľa ani názov

knihy či žánra. Sú tam iba ID – odkazy do patričných tabuliek. Tie musíme dotazom spojiť. Inak by sa v kontingenčnej tabuľke zobrazovali, namiesto mien a názvov, čísla. Ak by sa všetky skúmané údaje nachádzali v jednej tabuľke, tak by sme samozrejme mohli použiť kontingenčné zobrazenie priamo z danej tabuľky.

Kontingenčné zobrazenia sú na rozdiel od dotazov, väčšinou finálny produkt. Dotazy sú často určené na následné spracovanie v ďalších dotazoch, formulároch, zostavách alebo v kontingenčných tabuľkách – čo je aj prípad z nasledujúceho príkladu.

Chceme zistiť, aká je najčastejšia doba, po ktorej čitatelia vrátia knihu, pričom nás zaujíma, ktorí čitatelia sa na tej ktorej dobe podieľajú najviac. Navyše chceme, aby sme to mohli zisťovať zvlášť pre každý žáner a aj dokopy. Potrebujeme teda dobu výpožičky, žáner, priezvisko čitateľa. Pretože sa tieto polia nachádzajú v rôznych tabuľkách musíme ich spojiť dotazom. Dotaz je zobrazený na obr. 3.20.¹⁸ Používajú sa tie isté tabuľky ako v dotaze na obr. 3.18.

| | | | | |
|-----------|-------------------------------------|-------------------------------------|--|-------------------------------------|
| Pole: | Priezvisko | Žáner | doba: [DátumVrátenia]-[DátumVýpožičky] | Názov |
| Tabuľka: | Čitatelia | Žáner | | Knihy |
| Zoradiť: | | | | |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | | | |
| alebo: | | | | |

Obr. 3.20. Dotaz ku kontingenčnej tabuľke z obr. 3.21.

Po vytvorení dotazu vyberieme *Zobrazenie kontingenčnej tabuľky*. Zobrazí sa prázdna plocha reprezentujúca prázdnu tabuľku. Na nej sa nachádza oblasť pre riadky (*Sem presuňte polia riadkov*), stĺpce (*Sem presuňte polia stĺpcov*) a samotné údaje (*Sem presuňte súčty alebo podrobnosti*) - tie sa môžu v tabuľke zobrazovať podrobne alebo súhrnom. Nad tabuľkou je miesto pre polia, podľa ktorých sa bude filtrovať. Aby vznikla tabuľka musíme zadať údaje a aspoň jeden riadok alebo stĺpec.

Údaje priradíme na plochu z okna *Zoznam polí kontingenčnej tabuľky*. Ak okno nie je zobrazené, tak ho zobrazíme príkazom *Zoznam polí* v skupine *Zobraziť alebo skryť* v karte *Návrh*. Údaje môžeme priradiť na správne miesto pretiahnutím z tohto okna. Druhou možnosťou je po kliknutí na pole vybrať umiestnenie zo zoznamu spodnej časti okna tak, ako je to na obr. 3.21 a potom stlačiť tlačidlo *Pridať*.

¹⁸ Mohli by sme použiť aj spojenie poli meno a priezvisko, ako je to uvedené na strane 39.

| | | doba | | | | | | Celkový súčet |
|---------------|-------|-------|-------|-------|-------|-------|-------|---------------|
| | | 7 | 8 | 9 | 10 | 11 | 12 | |
| | | + - | + - | + - | + - | + - | + - | + - |
| Priezvisko | Počet | Počet | Počet | Počet | Počet | Počet | Počet | Počet |
| Klačko | | | | 1 | 1 | | | 2 |
| Komada | | | | | | 1 | | 2 |
| Rabajda | 1 | | | | | 5 | | 8 |
| Radvanská | | | | | | | | 3 |
| Strážovcová | | | | | | 1 | 2 | 5 |
| Súkeníková | | | | | | 2 | | 3 |
| Šalaga | | | | | | 10 | 1 | 11 |
| Tirschelová | | | | | | | 2 | 3 |
| Adamková | | | | | | | 1 | 1 |
| Tkáčik | | | | | | | 2 | 2 |
| Trnovec | 1 | | | | | | | 1 |
| Uličná | 1 | | | | | | | 3 |
| Valach | | | | | | | 2 | 4 |
| Vráblová | | 3 | | | | 1 | | 4 |
| Celkový súčet | 6 | 2 | | | | 20 | 10 | 52 |

Obr. 3.21. Kontingenčná tabuľka

Aby sme dostali tabuľku aká je na obr. 3.21., tak sme pole *Priezvisko* pridali do oblasti riadkov, *doba* do oblasti stĺpcov, *Žáner* do oblasti filtrov a *Názov* do údajovej oblasti. Ak by sme *Názov* vložili do oblasti podrobných údajov nevypisoval by sa ich počet, ale vypisovali by sa názvy kníh.

Názov môžeme pridať aj do oblasti údajov aj do oblasti podrobných údajov. V takom prípade by sme potom mohli klikaním + a – v záhlaví riadkov alebo stĺpcov zobrazovať aj podrobnosti alebo iba súhrny. Rovnaký výsledok dosiahneme kliknutím na názov poľa v záhlaví s príkazmi *Skryť podrobnosti* alebo *Zobraziť podrobnosti* v skupine *Zobraziť alebo skryť*.

| | | doba1 | | doba | | | |
|------------|-------|----------|------------------------|-------|-------|-------|-------|
| | | Skupina1 | | | | Iné | |
| | | 7 | 9 | Súčet | 8 | 10 | |
| | | + - | + - | + - | + - | + - | + - |
| Priezvisko | Počet | Počet | Názov | Počet | Počet | Počet | Počet |
| Klačko | | | Dom na streche | 1 | | | 1 |
| | | | | 1 | | | |
| Komada | | | Náš bol najlepší | 1 | | | |
| | | | | 1 | | | |
| Rabajda | 1 | | Tieň karavely | 2 | | | 1 |
| | | | | 1 | | | |
| Radvanská | | | Joseph Balsamo II. | 3 | | | |
| | | | Gróf Monte Cristo III. | | | | |
| | | | Ako sa zbaviť ženy | | | | |
| | | | | 3 | | | |

Obr. 3.22. Kontingenčná tabuľka so zoskupenými stĺpcami 7 a 9 a podrobnosťami v stĺpci 9

Po označení niektorých stĺpcov alebo riadkov (s klávesom Shift, s Ctrl aj nesusedných) ich môžeme zoskupiť príkazom *Zoskupiť* v skupine *Výbery*. Príkazom *Oddeliť* môžeme označenú skupinu zrušiť.

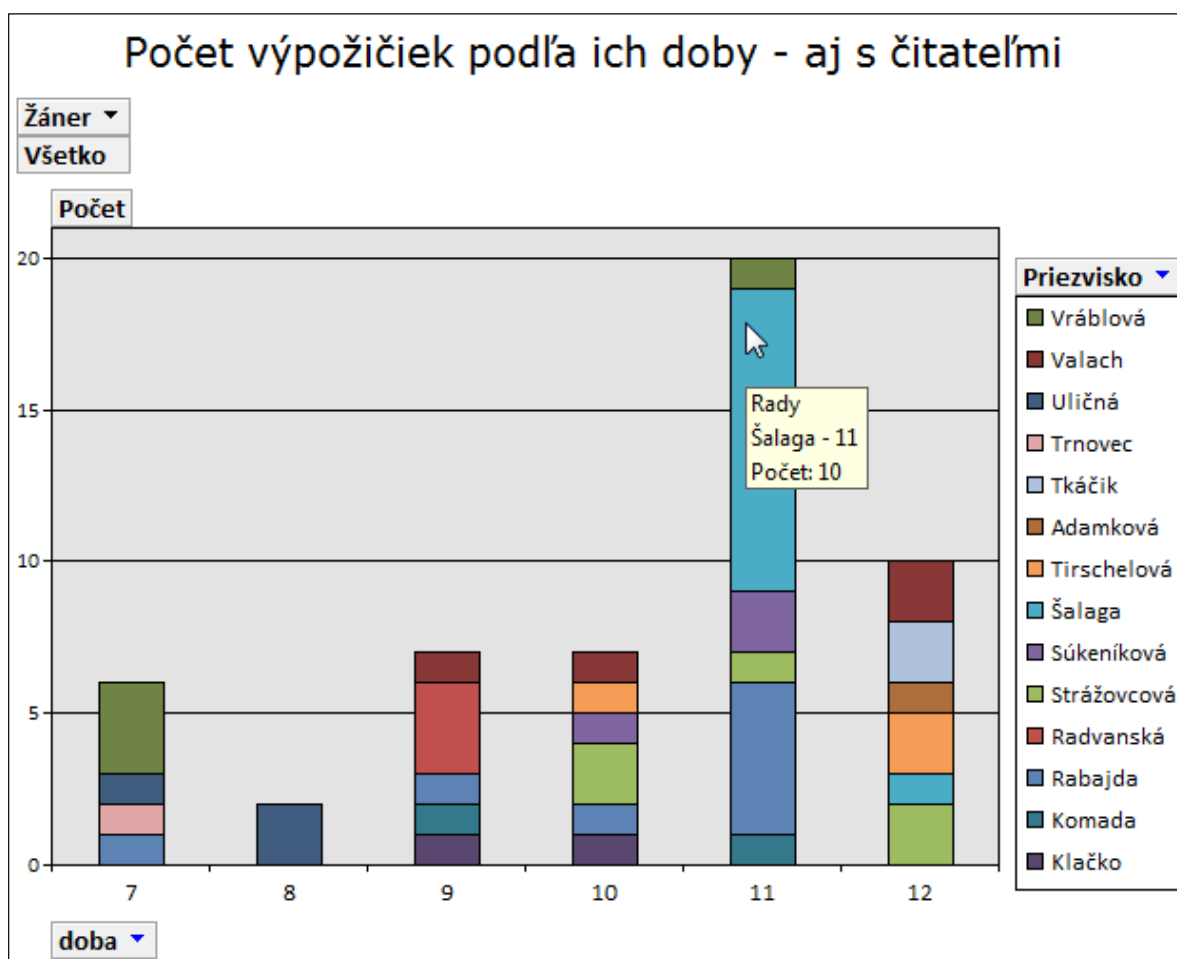
Na obr. 3.22 je tabuľka z obr. 3.21, v ktorej sú stĺpce (doby) 7 a 9 zoskupené (automaticky vznikol aj medzisúčet) a zároveň sú v stĺpci 9 zobrazené aj podrobnosti. Premenovať názov skupiny (ale aj skoro každého

objektu v kontingenčnej tabuľke) môžeme cez kontextové menu/vlastnosti/popisy alebo po označení skupiny cez *Hárak vlastností*.

Jednotlivé skupiny a polia môžeme po označení zbaliť, t. j. dočasne schovať. Napríklad skupinu Iné na obr. 3.22 by sme po jej označení (kliknutí na názov Iné) zbalili príkazom *Zbaliť pole* v skupine *Aktívne pole*. Tam sa nachádza aj príkaz *Rozbaliť pole*.

Príkazom *Automatický filter* v skupine *Filtrovať a zoradiť* môžeme prepínať zobrazenie s a bez filtrov. Ak sa podľa niektorého poľa filtruje, tak je vedľa jeho názvu modrý trojuholníček. Na obr. 3.21 vidno, že sú nastavené filtre pre polia *doba* a *Priezvisko* a pre pole *Žáner* nie.

Po kliknutí v záhlaví na názov poľa (v našom prípade na *Priezvisko* a *doba*) môžeme pre označené pole v skupine *Nástroje* nastaviť zobrazovanie medzisúčtov alebo vybrať príkazom *Automatický výpočet* vybrať typ agregačnej funkcie. Pre nečíselné polia je samozrejme k dispozícii iba počet.



Obr. 3.23. Kontingenčný graf (upravený) z tabuľky na obr. 3.21.

Na obr. 3.23 je graf, ktorý vznikne z kontingenčnej tabuľky (bez zoskupenia stĺpcov 7 a 9). Graf je upravený:

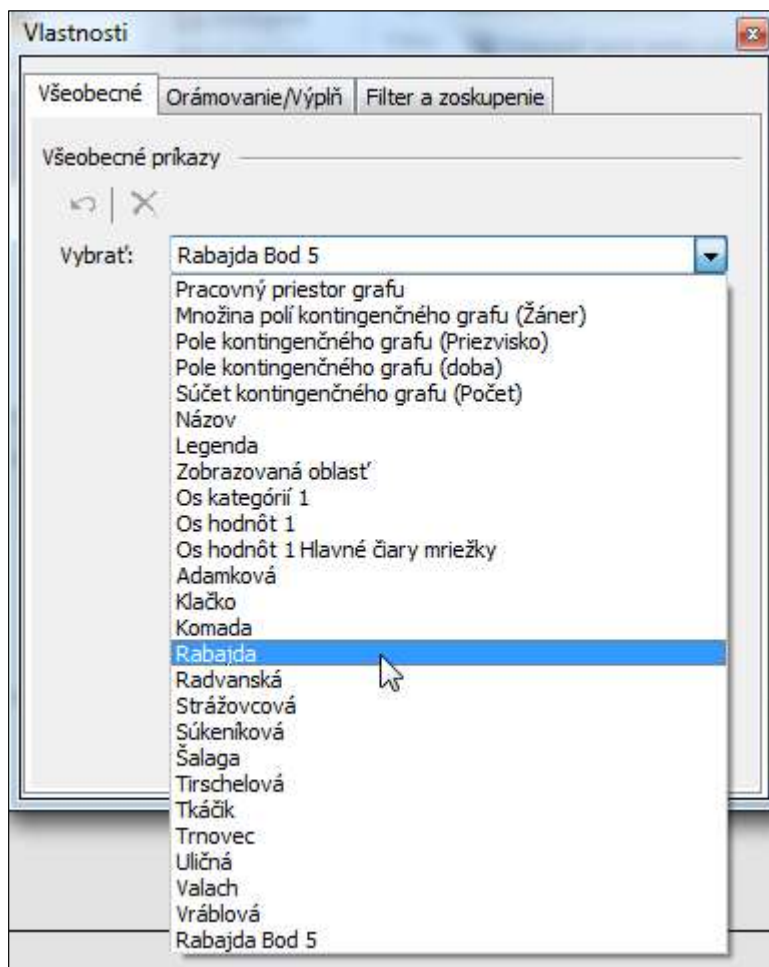
- Štandardne sa použije stĺpcový graf. Príkazom *Zmeniť typ grafu* sme ho zmenili na skladaný stĺpcový – stĺpce z jednej skupiny sa naukladajú na seba a tak vyjadrujú celkový počet.
- Zobrazili sme legendu. Príkaz sa nachádza v skupine *Zobraziť alebo skryť*.
- Pôvodne sa v legende zobrazovali doby a na osi kategórií (vodorovná os - tam, kde sú teraz na obrázku doby) boli priezviská. V skupine *Aktívne pole* sme polia prehodili príkazom *Prepnúť riadok a stĺpec*, to je to isté ako keby sme v tabuľke prehodili riadky a stĺpce.

Z grafu vieme vyčítať (v rámci filtrovaných dôb 7-12), že najčastejšie majú čitatelia knihu požičanú jedenásť dní – stalo sa tak dvadsaťkrát, čo je viac ako dvojnásobok bežnej doby. Ďalej vidíme, že „na svedomí“

to má hlavne čitateľ Šalaga, ktorý desaťkrát vrátil knihu na jedenásty deň. Ak nie on (a prípadne Rabajda s piatimi knihami), tak by jedenást'dňová doba vôbec nevybočovala z priemeru. V tabuľke by sme mohli týchto dvoch čitateľov premiestniť pretiahnutím na začiatok a tak by sme pekne videli, o koľko zvyšujú počet. Druhou možnosťou by bolo utriediť tabuľku podľa počtu kníh.

Pri porovnávaní čitateľov nám môžu vadieť podobné farby, ktoré im Access pridelil – u nás sa to týka čitateľov Komada a Rabajda. Preto je častou operáciou v grafe zmena farby. V hároku vlastností vyberieme v karte *Všeobecné* čitateľa a potom v karte *Orámovanie/Výplň* môžeme zmeniť formát stĺpca.

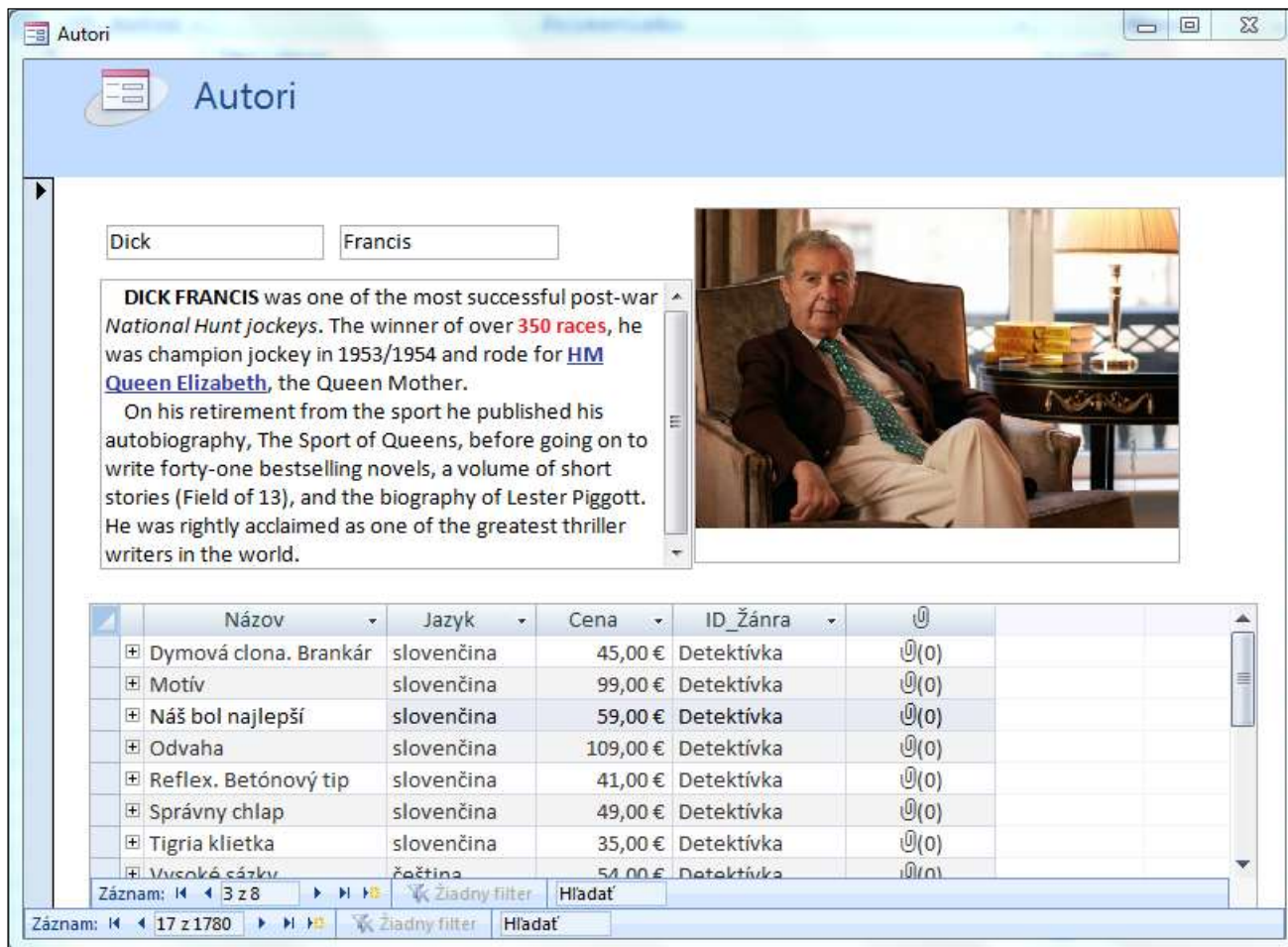
Ak by sme chceli z nejakého dôvodu zmeniť farbu čitateľa iba v jednom bode (stĺpci), stačilo by ho vybrať kliknutím a potom by sme mohli v hároku vlastností formátovať. Situáciu vidíme na obr. 3.24. Môžeme si vybrať čitateľa alebo čitateľa iba v bode 5 (posledné v ponuke). Bod 5 - klikli sme na piaty stĺpec. Do hároku vlastností sa dostaneme aj kliknutím pravého tlačidla na objekt príkazom *Vlastnosti* z kontextového menu.



Obr. 3.24. Hárok vlastností kontingenčného grafu

4. Formuláre

Slúžia na vytvorenie užívateľského rozhrania, pomocou ktorého užívateľ prístupuje k databáze. Na obr. 4.1 je zobrazený formulár pre tabuľku *Autori*. Okrem pridania nového autora, úpravy alebo zmazania existujúceho autora zároveň umožňuje pridávať, upravovať alebo mazať knihy každého autora.



Obr. 4.1. Ukážka formulára

4.1. Vytvorenie jednoduchého formulára

Vytvorenie formulára z obr. 4.1 je záležitosť niekoľkých kliknutí myšou:

1. Na navigačnej table si klikneme na tabuľku *Autori* - táto tabuľka bude zdrojom pre záznamy formulára¹⁹.
2. V karte *Vytvoriť* v skupine *Formuláre* klikneme na možnosť *Formulár*. Okrem toho sú v skupine aj možnosti:
 - *Rozdelené*. Podobný vzhľad ako na obr. 4.1. Namiesto tabuľky s knihami bude tabuľka autorov, pomocou ktorej môžeme prepínať, ktorého autora zobrazíme v hornej časti.
 - *Viaceré položky*. Podobné ako údajové zobrazenie, t. j. viac autorov na stranu ale je napríklad zobrazená fotka (OLE).
 - *Kontigenčný graf*.

¹⁹ Formulár môžeme vytvoriť aj z dotazu.

- *Prázdny formulár* a *Návrh formulára* umožňujú vytvoriť formulár od základov. Tieto možnosti sa líšia iba spôsobom zobrazenia.
- *Sprievodca formulárom*. Na základe jednoduchých otázok pomôže vytvoriť formulár.
- *Údajový hárok*. Formulár podobný ako údajové zobrazenie.
- *Modálny dialóg*. Okná (nielen) v MS Windows môžeme deliť na modálne a nemedálne. Ak je okno modálne, tak je na vrchu a nie je možné pracovať v ostatných oknách programu. Napríklad, ak dáte uložiť súbor v MS Word a máte vybrať názov a miesto, tak nemôžete pokračovať v úprave dokumentu MS Word (ani iného) pokiaľ toto okno nezavriete. Iným modálnym oknom je prihlásenie sa do MS Windows. Nemedálne okná umožňujú prepnutie do iného okna toho istého programu. A aj modálne okná jedného programu nebránia štandardne prepnutiu do okna iného programu, napríklad môžete pri ukladaní dokumentu prepnúť do prehliadača a surfovať.
- *Kontingenčná tabuľka*.

| | |
|-------------|--|
| ID_Autora: | 17 |
| Priezvisko: | Francis |
| Meno: | Dick |
| Životopis: | DICK FRANCIS was one of the most successful post-war <i>National Hunt jockeys</i> . The winner of over 350 races , he was champion jockey in 1953/1954 and rode for HM Queen Elizabeth , the Queen Mother. On his retirement from the sport he published his autobiography, <i>The Sport of Queens</i> , before going |
| Foto: | |

| | Názov | Jazyk | Cena | ID_Žánra | |
|---|-----------------------|------------|---------|------------|------|
| + | Dymová clona. Brankár | slovenčina | 45,00 € | Detektívka | 🔗(0) |
| + | Motív | slovenčina | 99,00 € | Detektívka | 🔗(0) |


Obr. 4.2. Formulár, ktorý vznikne po dvoch kliknutiach

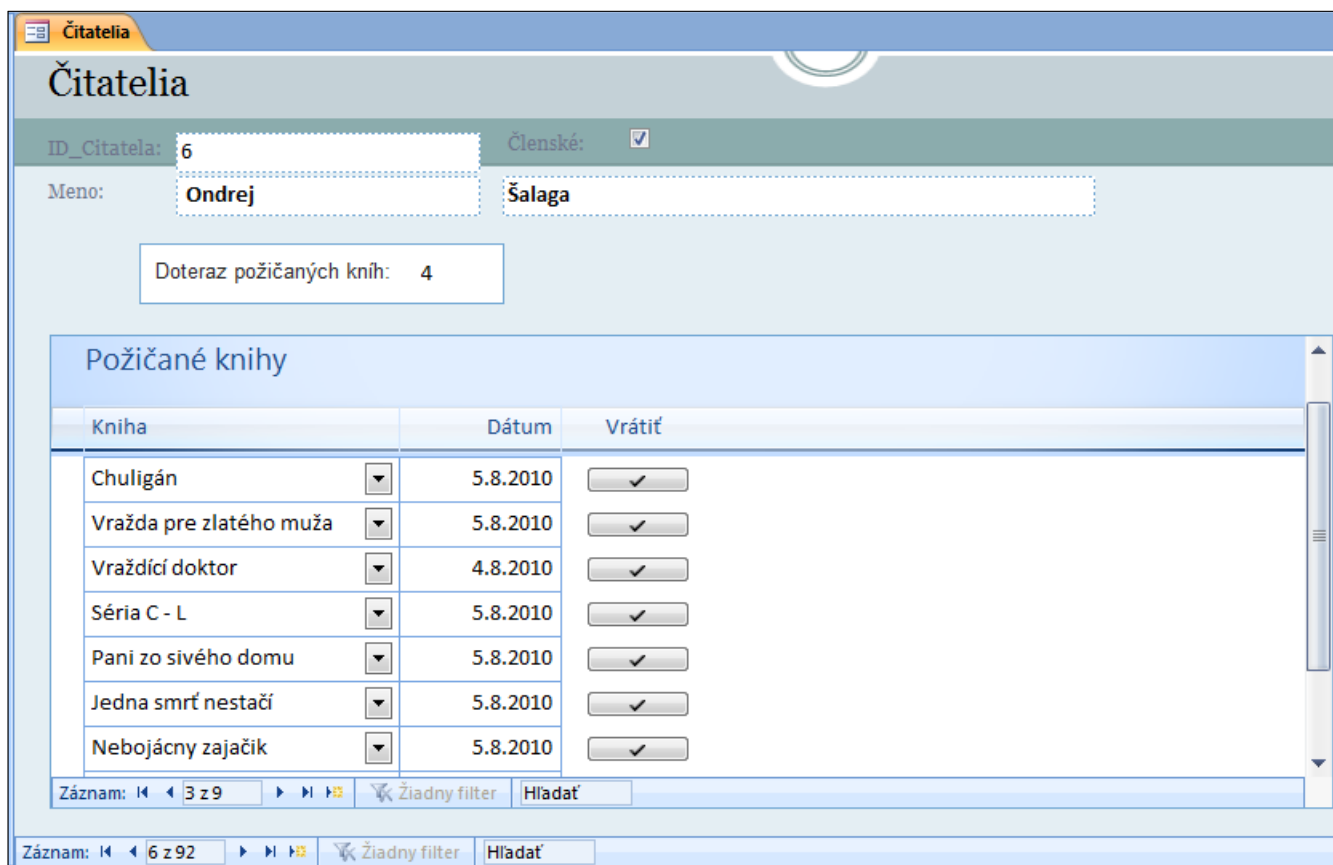
Formulár veľmi podobný formuláru na obr. 4.1 je hotový. Podobne ako má tabuľka návrhové a údajové zobrazenie má aj formulár rôzne spôsoby zobrazenia (prepína sa medzi nimi obdobne ako pri tabuľke alebo dotazoch):

- *Formulárové zobrazenie*. Zobrazenie, v ktorom sa formulár používa pri práci.
- *Zobrazenie rozloženia*. V tomto zobrazení sa ocitneme po vytvorení formulára. Formulár je zobrazený tak ako sa bude používať (*Formulárové zobrazenie*, t. j. aj s obsahom polí) ale navyše tu máme niektoré možnosti úpravy. V tomto zobrazení prídu karty *Formátovanie* a *Usporiadať*.
- *Návrhové zobrazenie*. V tomto zobrazení máme najväčšie možnosti úprav. Obsah polí sa nezobrazuje. V tomto zobrazení prídu karty *Návrh* a *Usporiadať* (jej obsah sa líši od rovnako pomenovanej karty pri *Zobrazení rozloženia*).

Ak by sme chceli premiestniť pole *Foto* napravo, ako je to na obr. 4.1, zistíme, že sa to (zatiaľ) **nedá**.

- Môžeme iba meniť poradie jednotlivých polí kliknutím a potiahnutím hore či dole, t. j. môžeme dať napríklad *Foto* nad *Životopis*.
- Dokonca nemôžeme zmenšiť šírku bez toho, aby sa nezmenili šírky ostatných polí.
- A ak by sme chceli odstrániť popis poľa, tak zmizne aj samotné pole.

Tieto polia sú zoskupené a pred úpravou samostatného poľa, musíme toto pole **oddeliť**. Ak chceme vedieť, s ktorými objektmi je pole zoskupené, môžeme po označení tohto poľa kliknúť na značku  , ktorá je zobrazená na obr. 4.2 v ľavom hornom rohu a označiť tak všetky spojené objekty. Po označení poľa²⁰ pomocou voľby *Odstrániť* zo skupiny *Rozloženie ovládacieho prvku* na karte *Usporiadať* pole oddelíme. Potom už možno toto pole premiestniť a upraviť jeho rozmery ľubovoľne a nezávisle na ostatných poliach. Podobne môžeme upraviť aj ostatné polia.



| Kniha | Dátum | Vrátiť |
|-------------------------|----------|--------------------------|
| Chuligán | 5.8.2010 | <input type="checkbox"/> |
| Vražda pre zlatého muža | 5.8.2010 | <input type="checkbox"/> |
| Vraždici doktor | 4.8.2010 | <input type="checkbox"/> |
| Séria C - L | 5.8.2010 | <input type="checkbox"/> |
| Pani zo sivého domu | 5.8.2010 | <input type="checkbox"/> |
| Jedna smrť nestačí | 5.8.2010 | <input type="checkbox"/> |
| Nebojácny zajačik | 5.8.2010 | <input type="checkbox"/> |

Obr. 4.3. Formulár pre zobrazenie čitateľov a ich kníh.

4.2. Vytvorenie formulára s podformulárom

Na obrázku obr. 4.3. je formulár, zobrazujúci čitateľov a ich aktuálne výpožičky. V ďalšom texte je ukázaný postup na jeho vytvorenie. Formulár obsahuje aj tlačidlá pre vrátenie požičaných kníh. Ich stlačením sa spustí makro, ktorého vytvorenie je popísané v kapitole 5.

Na formulári si možno všimnúť dva pásy navigačných tlačidiel. Spodným pásom sa posúvajú a vyhľadávajú čitatelia, vrchný slúži pre knihy aktuálne zobrazeného čitateľa.

Formulár pozostáva z troch rôznych častí.

- Hlavný formulár *Čitatelia*.
- Podformulár vytvorený zo súčtového dotazu, v ktorom zobrazujeme iba počet doteraz požičaných kníh. V kapitole 4.2.2 je ukázané, že sa použitiu podformulára v tomto prípade môžeme vyhnúť.
- Podformulár *Požičané knihy*.

Hlavný formulár začneme vytvárať z prázdneho formulára. Každý formulár, pracujúci s dátami **musí** mať definovaný *Zdroj záznamov*, t. j. odkiaľ sa budú brať dáta do polí vo formulári. Môže to byť tabuľka, dotaz

²⁰ Pri stlačení klávese Shift môžeme postupným kliknutím označiť viacero objektov.

alebo SQL príkaz. Ponuka sa zobrazí v pravej časti formulára. Ak by chýbala zobrazíme ju kliknutím na *Pridať existujúce polia* na karte *Formátovanie* v skupine *Ovládacie prvky*. Obdobne ju môžeme skryť. V našom prípade vyberieme tabuľku *Čitatelia* a pridáme postupne dvojklikom alebo potiahnutím všetky jej polia.

V tomto kroku je dobré si vyskúšať umiestňovanie jednotlivých polí a ich popisov, prípadne ich odstránenie a opätovné pridanie. Najlepšie v oboch zobrazeniach – v *Návrhovom zobrazení* aj v *Zobrazení rozloženia* a kontrolovať ako bude formulár vyzerat' *Formulárovým zobrazením*. Pri návrhovom zobrazení sa pri posúvaní poľa posúva aj jeho popis. Ak ale pole posúvame za štvorček v ľavom hornom rohu, tak sa popis neposúva.

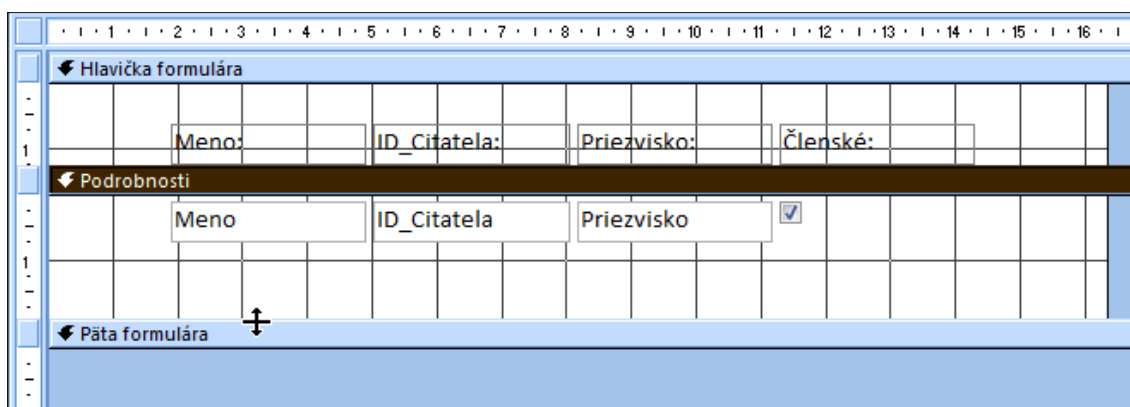
Ďalšia vlastnosť na odskúšanie sú možnosti z už spomínanej skupiny *Rozloženie ovládacieho prvku* na karte *Usporiadať* pole. Ak označíme všetky polia a vyberieme *Tabuľkové* alebo *Navrstvené* dostaneme rôzne automatické rozmiestnenie prvkov. Okrem toho sa tu nachádzajú možnosti *Prichytiť k mriežke*, ktorá pole umiestňuje do mriežky. Delenie nakreslenej mriežky definuje v *Hárok vlastností*, v časti *Formulár*, *Os x mriežky* a *Os y mriežky*. To znamená, že 1 je najhrubšie delenie – polia sa prichytia iba na mriežku, 2 – polia sa prichytia k mriežke alebo k jej polovici, 3 – polia sa prichytávajú k mriežke alebo k jej tretinám, atď. Ak túto možnosť vypneme budeme môcť umiestňovať objekty po bodoch a tak sa nám ich bude ťažšie zarovnávať. V tejto skupine sú aj nastavenia, ako ďaleko má byť obsah poľa od jeho okrajov (vnútorný okraj, *Okraje ovl. prvkov*) a ako ďaleko majú byť od seba jednotlivé polia (*Výplň ovl. prvku*, táto možnosť je aktívna iba pri tabuľkovom alebo navrstvenom rozložení).

Každý formulár sa môže skladať až z troch častí:

- Hlavička a päta formulára
- Hlavička a päta strany
- Podrobnosti – telo formulára.

Pri tabuľkovom rozmiestnení prvkov Access automaticky pridá hlavičku a päť formulára a do hlavičky premiestni popisy polí. V návrhovom režime môžeme hlavičky a päty pridať alebo ubrať po kliknutí pravým tlačidlom myši na formulár (tam kde je mriežka) a výberom z kontextového menu.

Hlavičky a päty majú rovnaké využitie ako iných dokumentoch. Umiešťa sa do nich grafika (logá), čísla strán, priemer, počet alebo súčet na stranu či skupinu, číslovanie apod. Veľkosť jednotlivých častí môžeme meniť tak ako je to znázornené na obr. 4.4. V zobrazovanom príklade sa zväčšuje alebo zmenšuje plocha častí *Podrobnosti*. Ak by sa kurzor posunul mierne dole, na spodnú časť lišty *Päť formulára*, tak by sa mohla zväčšiť päť, ktorá je momentálne nulová. Rozmery formulára sú štandardne zobrazené bielou plochou, na ktorej je zobrazená mriežka.



Obr. 4.4. Zmena veľkosti jednotlivých častí formulára

Po kliknutí na pole alebo jeho popis môžeme tieto objekty rôzne formátovať. Môžeme meniť typ, farbu, veľkosť písma, zarovnanie, farbu výplne, a pod., prípadne vybrať automatický formát.

Ak chceme, aby meno a priezvisko boli vedľa seba tak ako na obr. 4.3, môžeme ich presunúť ale zrejme budeme mať problém aby boli vyrovnané. Podobne ak chceme, aby boli polia pekne pod sebou. Na to použijeme možnosti na karte *Usporiadať*, v skupine *Zarovnanie ovládacieho prvku*. Na tej istej karte (iba pri návrhovom

zobrazení) v skupine *Velkosť* nájdeme možnosti na nastavenie šírky a výšky podľa najväčšieho alebo najmenšieho z označených prvkov. Zaujímavou možnosťou je *Ukotvenie*. Umožní prichytiť (ukotviť) objekt k jednému alebo dvom okrajom formulára a objekt si pri zmene veľkosti formulára zachová stále rovnakú vzdialenosť od okrajov, ku ktorým je prichytený. V skupine *Pozícia* sa nachádzajú možnosti *Dopredu* a *Dozadu*, ktoré umožnia pri prekrývajúcich sa objektoch určiť, ktorý objekt je na vrchu a ktorý na spodku. Nastavujeme tak viditeľnosť.

V návrhovom zobrazení sú na karte *Usporiadať*, v skupine *Zobraziť/Skryť* možnosti pre zobrazenie mriežky, pravickej a hlavičky s päťou formulára a strany.

4.2.1. Pridanie podformulára do hlavného formulára

Ak sme si vyskúšali všetky tieto možnosti, umiestnime všetky štyri polia aj s popismi do hornej časti formulára, tak ako je to na obr. 4.3.

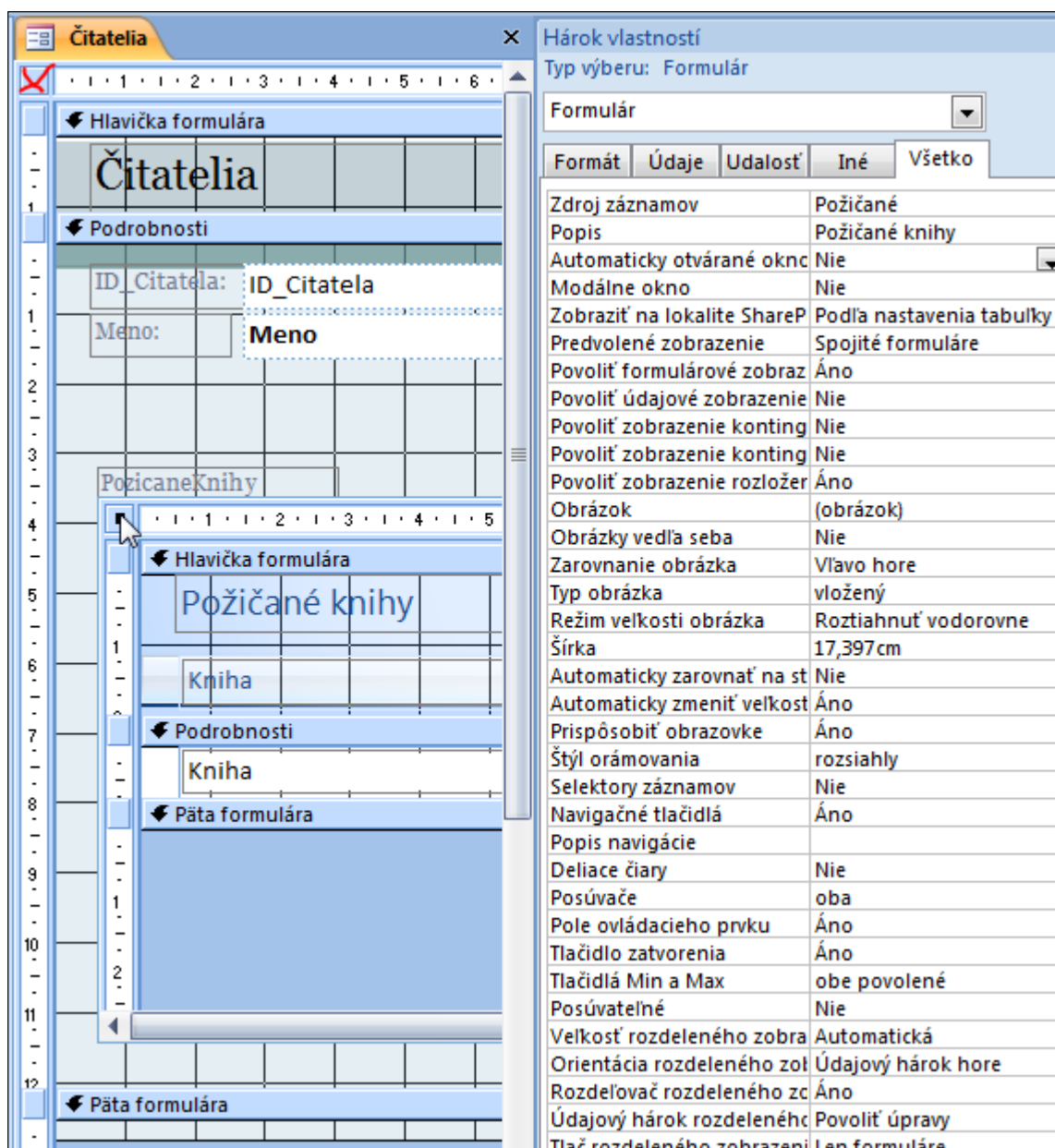
Podformulár *Požičané knihy* môžeme vytvoriť dvomi spôsobmi:

1. Vytvoríme ho ako nový formulár.
 - T. j. klikneme na tabuľku *Požičané*.
 - Klikneme na karte *Vytvoriť*, v skupine *Formuláre* na *Viacere položky*. Vytvorí sa formulár.
 - Formulár upravíme, zmažeme nepotrebné polia, pridáme tlačidlo (to je popísané nižšie).
 - Po uložení ho pretiahneme z navigačnej tably do hlavného formulára.
2. Vytvoríme ho priamo v hlavnom formulári.
 - V návrhovom režime klikneme v karte *Návrh* v skupine *Ovládacie prvky* na možnosť *Podformulár/Čiastková zostava*.
 - Vo formulári klikneme na prázdne miesto a potiahnutím vytvoríme obdĺžnik, v ktorom sa bude podformulár nachádzať. Spustí sa *Sprivodca podformulárom*. Ak sa nespustí zmažeme vytvorený objekt, označíme v tej istej skupine *Použiť sprivodcov ovládacími prvkami* a postup opakujeme.
 - V prvom kroku sprivodcu určíme odkiaľ budeme čerpať údaje pre podformulár. Ak by sme podformulár už vytvorili (napríklad ako samostatný formulár v prvom bode) mohli by sme ho použiť. V tomto prípade ale označíme *Použiť existujúce tabuľky a dotazy*.
 - V ďalšom kroku vyberieme tabuľku *Požičané* a polia *Knihy* a *Dátum*.
 - Access zistí, že tabuľky sú prepojené cez stĺpec *ID Čitateľa*. Prepojenie môžeme ale v niektorých prípadoch zadať ručne alebo nenastaviť žiadne.
 - V poslednom kroku podformulár pomenujeme.
 - Podformulár môžeme upravovať v rámci hlavného formulára ale to má význam iba pre jeho umiestnenie v rámci hlavného. Pohodlnejšie je upravovať podformulár samostatne, predtým ale musíme hlavný formulár zavrieť.
 - Podformulár sa zobrazuje štandardne ako tabuľka, ak ho chceme zobraziť tak ako na obr. 4.3. (aby sme mohli pridať tlačidlá) musíme zmeniť jeho predvolené zobrazenie pomocou hárika vlastností zo *Samostatný formulár* na *Spojité formuláre*. Pretože v kombinovanom formulári sú formuláre dva môže byť pre začiatočníka problém docieľiť aby *Hárok vlastností* zobrazoval vlastnosti toho správneho objektu.

Pokiaľ sa vlastnosti podformulára nastavujú samostatne ako bolo spomenuté v predchádzajúcom bode, tak je to jednoduchšie. V prípade kombinovaného formulára musíme pri zobrazenom hárike vlastností kliknúť raz/dvakrát do ľavého horného rohu

podformulára. Miesto (s kurzorom) je zobrazené na obr. 4.5. Červeným krížikom je označené miesto pre výber hlavného formulára.

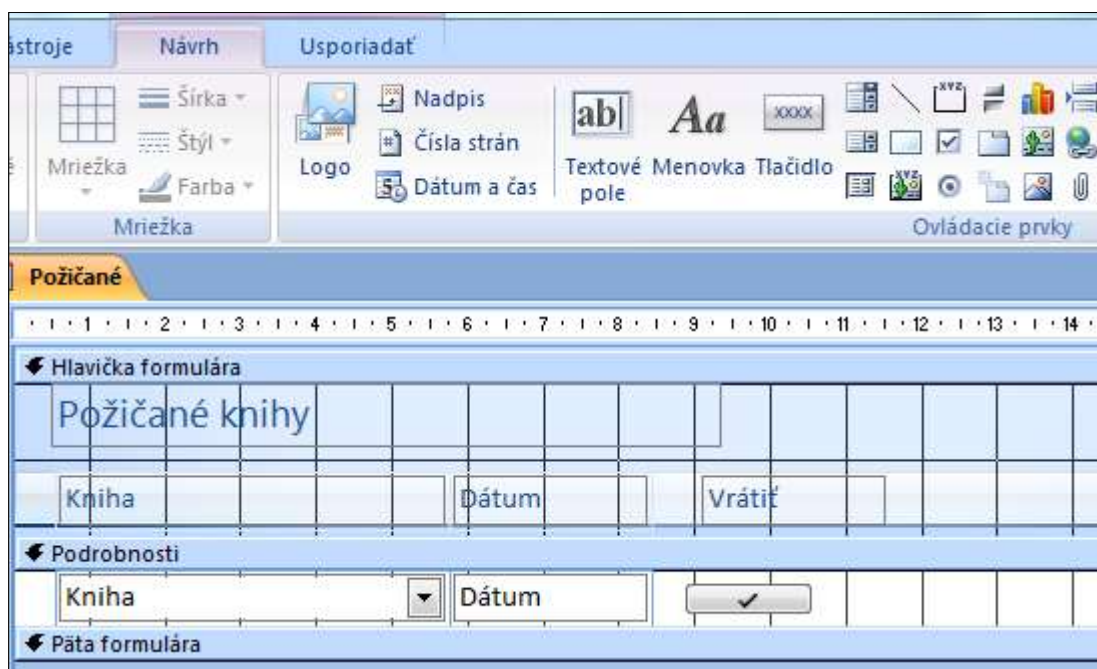
- o Po nastavení spojeného zobrazenia je vhodné presunúť popisy polí (Ctrl-X, Ctrl-V) do hlavičky formulára.



Obr. 4.5. Nastavenie vlastností podformulára

Z uvedeného je vidno, že prvý spôsob je o niečo pohodlnejší a rýchlejší - vytvorí spojený formulár jedným kliknutím a netreba presúvať popisy a polia.

Tlačidlo pridáme do podformulára v návrhovom zobrazení kliknutím v karte *Návrh*, v skupine *Ovládacie prvky* na *Tlačidlo*. Potom klikneme na miesto, kde ho chceme pridať. Spustí sa *Spríevodca príkazovým tlačidlom*. Môžeme si vybrať z množstva predefinovaných akcií. V našom prípade potrebujeme záznam presunúť z tabuľky *Požičané* do tabuľky *História*, také niečo ale neponúka žiadna akcia. Našťastie v poslednej skupine je možnosť spustiť makro. Preto si musíme najskôr vytvoriť makro, ktoré to urobí. Túto úlohu dokončíme v nasledujúcej kapitole, kde je ukázané vytvorenie (nielen) takéhoto makra.



Obr. 4.6. Pridanie tlačidla do formulára

4.2.2. Pridanie agregáčnych výpočtov do formulára

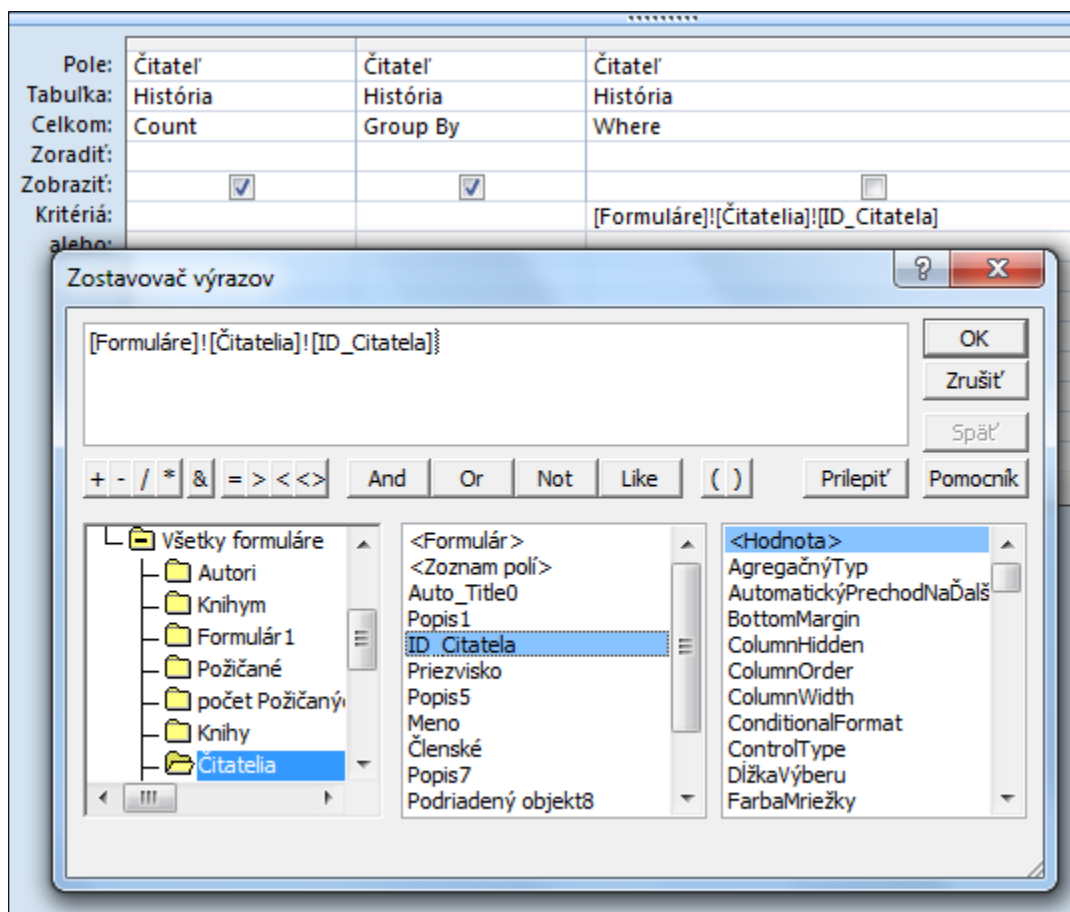
Aby sme dokončili náš formulár z obr. 4.3 musíme ešte pridať časť slúžiacu pre zobrazovanie počtu doteraz požičaných kníh. Pre jednoduchosť budeme počítať iba knihy z tabuľky *História*. Ukážeme si to tromi spôsobmi:

- Ako ďalší formulár:
 - pomocou dotazu s medzisúčtami a
 - s využitím databázových funkcií.
- Vytvorením dotazu spájajúceho tabuľku *Čitateľa* s počtom ich výskytov v tabuľke *História*. Tento dotaz použijeme pri tvorbe formulára namiesto tabuľky *Čitateľa*. Tento spôsob je popísaný v kapitole 4.3.

Podformulár vytvorený na základe dotazu

Najskôr musíme vytvoriť dotaz, ktorý spočíta koľkokrát sa každý čitateľ nachádza v tabuľke *História*. Na obr. 4.7 je dotaz veľmi podobný dotazu z obr. 3.11. V tabuľke zoskupíme riadky podľa čitateľa a zistíme ich počet. Ak by v dotaze nebol tretí stĺpec, tak by výsledkom dotazu bola tabuľka všetkých čitateľov v tabuľke *História* a počet ich výskytov v nej.

My nechceme všetkých čitateľov, ale iba toho jedného, aktuálne zobrazeného vo **formulári** *Čitateľa* (nie v tabuľke *Čitateľa*). Preto musíme pridať podmienku v treťom stĺpci. Aby sme sa nemuseli trápiť s presným zápisom (hranaté zátvorky, výkričníky, názvy polí a pod.) spustili sme pomocou kliknutia pravého tlačidla myši *Zostavovač výrazov*. Pre túto podmienku dotaz samostatne nefunguje – aby fungoval, musí byť spustený formulár *Čitateľa*.



Obr. 4.7. Dotaz pre výpočet počtu výpožičiek čitateľa zobrazeného vo formulári *Čitatelia*

Tento dotaz stačí vložiť do formulára *Čitatelia* (buď potiahnutím z *Navigačnej tably* alebo kliknutím v karte *Návrh* v skupine *Ovládacie prvky* na možnosť *Podformulár/Čiastková zostava*) a upraviť jeho vzhľad (Hlavne v *Hárku vlastností* nastaviť *Predvolené zobrazenie* na *Samostatný formulár*). Rovnako ako v predchádzajúcom prípade (v kapitole 4.2.1) je výhodnejšie upravovať obsah podformulára zvlášť a iba umiestnenie upraviť v hlavnom formulári.

Podformulár s využitím databázových funkcií

Tento podformulár začneme vytvorením formulára, ktorý vložíme do hlavného formulára. Na karte vytvoriť vyberieme *Návrh formulára*. Pretože budeme potrebovať iba jeden riadok, môžeme zmenšiť plochu *Podrobnosti*. V *Hárku vlastností*, v karte *Údaje* musíme určiť *Zdroj záznamov*, tým bude samozrejme tabuľka *História*. Ak by sme zdroj záznamov neurčili, tak by sme nemohli formulár prepojiť s hlavným formulárom. Takéto neprepojené formuláre a polia sa nazývajú *neviazané*.

Po pridaní sa môže automaticky nastaviť aj triedenie (*Zoradiť podľa* v karte *Údaje* v *Hárku vlastností*) podľa polí, ktoré my nepoužívame. To by robilo problémy, preto toto triedenie odstránime.

Na karte *Návrh*, v skupine *Ovládacie prvky* klikneme na *Textové pole*. Vo formulári kliknutím a potiahnutím toto textové pole nakreslíme. Pole bude zatiaľ *neviazané* – nie je s ničím prepojené, t. j. nečerpá údaje zo žiadneho zdroja. Ak je pole označené v *Hárku vlastností*, v karte *Údaje* môžeme určiť *Zdroj ovládacieho prvku*. Môžeme vybrať z ponuky alebo využiť *Zostavovač výrazov*, tak ako je to na obr. 4.8.

V našom prípade sme použili funkciu *DCount*, ktorá má tri argumenty:

- pole v úvodzovkách, ktoré spočítavame (je to jedno, ide nám o počet riadkov, napríklad *Čitateľ*),
- doménu v úvodzovkách (v našom prípade tabuľku *História*), v ktorej spočítavame a
- podmienku, kedy sa má spočítavať

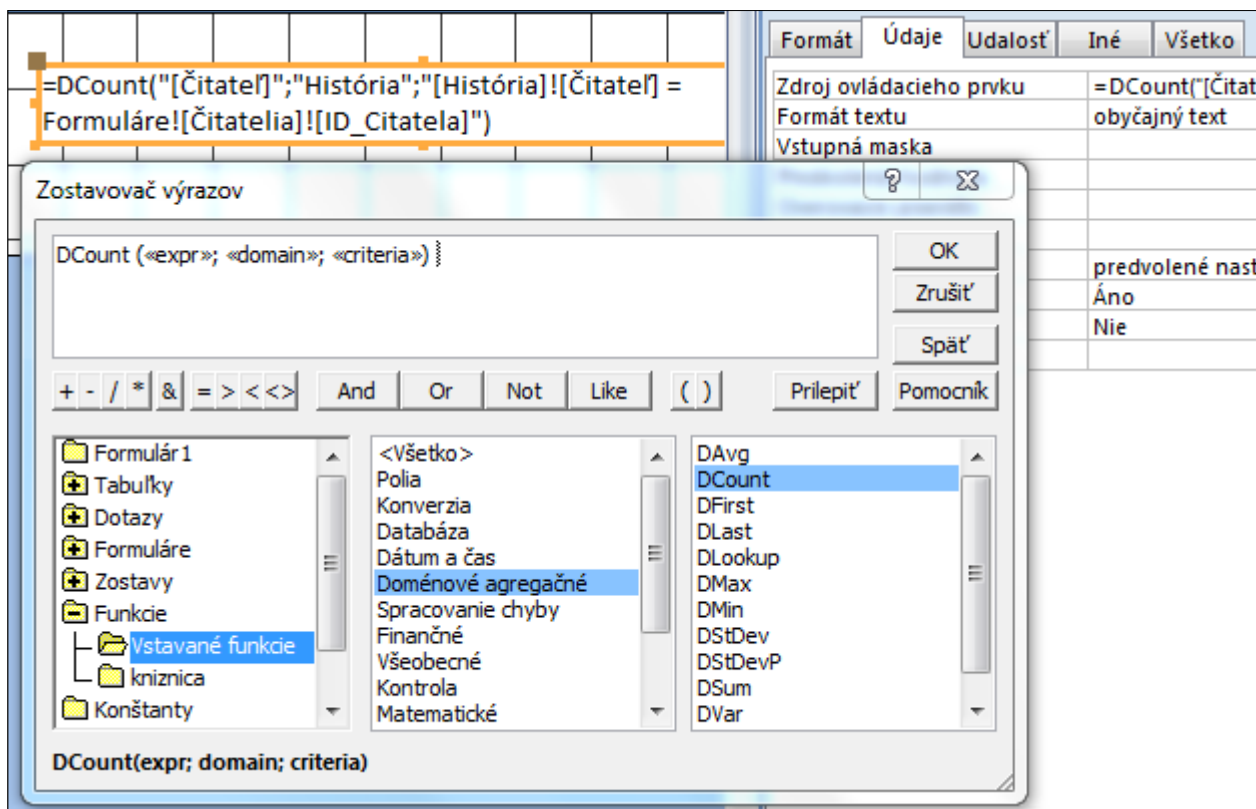
([História]![Čitateľ] = Formuláre![Čitatelia]![ID_Citela])

Celá funkcia:

=DCount("[Čitateľ]";"História";"[História]![Čitateľ] = Formuláre![Čitatelia]![ID_Citela]")

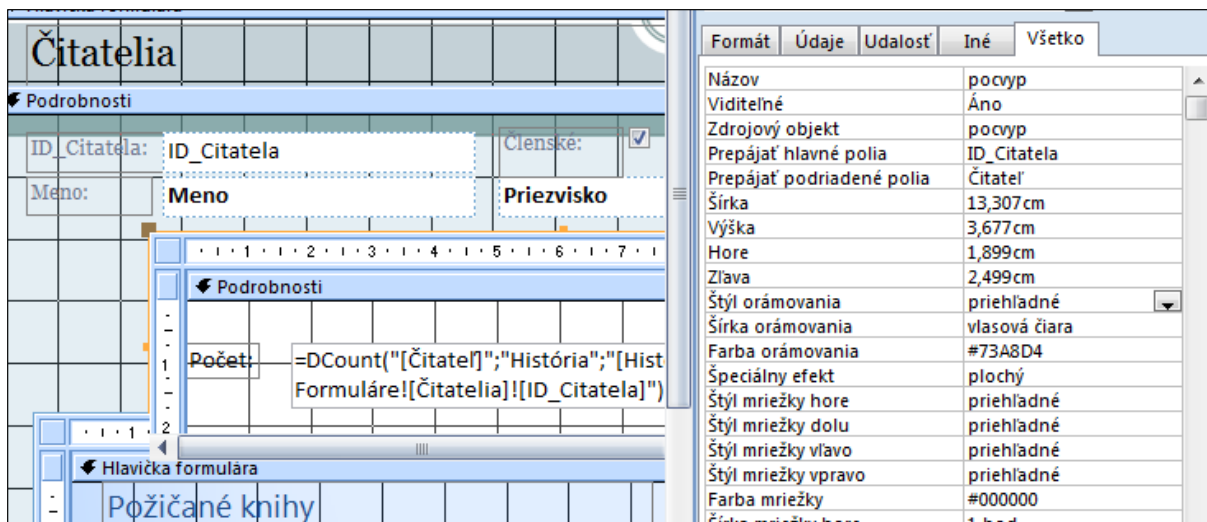
Zostavovač umožní vybrať funkciu, ukáže aké má argumenty a umožní ich vložiť.

Pri porovnávaní dotazu na obr. 4.7 a funkcie na obr. 4.8 by malo byť vidno veľa podobného.



Obr. 4.8. Nastavenie zdroja ovládacieho prvku

Formulár potom zavrieme a otvoríme hlavný formulár. Do neho môžeme vložiť čerstvo vytvorený podformulár rovnako ako pri dotaze vyššie. Na obr. 4.9 sú vlastnosti podformulára po jednom kliknutí do jeho ľavého horného rohu na obr. 4.5 po dvoch kliknutiach (brané zo stavu, keď nebol podformulár označený). Ak nechceme, aby bolo vidieť rám, navigačné tlačidlá a pod. podformulára (čo sa v tomto prípade nehodí) mali by sme vhodne nastaviť hlavne *Štýl orámovania*, *Navigačné tlačidlá*, *Selektory záznamov*, *Posúvače*. Je dobré poskúšať zmeniť aj ostatné vlastnosti. Vždy po kliknutí na vlastnosť sa jej popis zobrazuje vľavo v stavovom riadku.



Obr. 4.9. Označenie a vlastností podformulára

4.3. Formulár zo spojených tabuliek

V kapitole 4.2 je ukázané, ako možno vo formulári zobrazíť výsledky agregáčnych funkcií pomocou podformulára. Podstatou bolo, že v hlavnom formulári bola základom tabuľka *Čitatelia*. Môžeme ale postupovať tak, že základom bude tabuľka, ktorá vznikne, keď k tabuľke *Čitatelia* pripojíme výsledky agregáčnych funkcií. Takto sa vyhneme vytváraniu jedného podformulára.

Spojenie vytvoríme pomocou dotazu, ktorý je znázornený na obr. 4.10. Je to vcelku jednoduchý súčtový dotaz. Zoskupujeme (*Group by*) podľa ID čitateľa a počítame koľkokrát (*Count*) sa nachádza v tabuľke *História*. Jediný figeľ je v použití agregáčnej funkcie *First*, aby sme mohli vo výsledku použiť meno, priezvisko a členské. V každej skupine ide o jedného čitateľa, tak jeho meno zistíme ako meno prvého, atď.

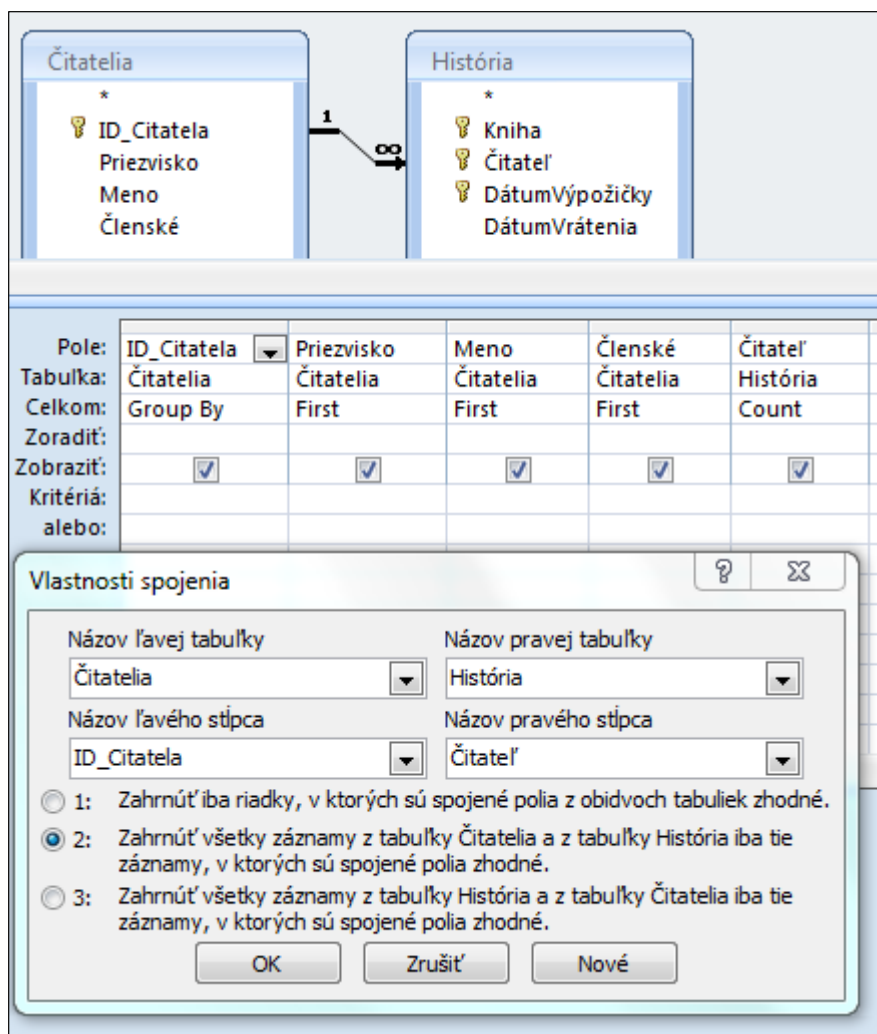
Vzťah medzi tabuľkami je pre tento dotaz upravený (všimnite si šípku). Ak by sme vo *Vlastnostiach spojenia* nechali prvý typ, tak by vo výsledku dotazu boli iba čitatelia, nachádzajúci sa v tabuľke *História*. Takto budú k dispozícii všetci čitatelia a tí, ktorí sa v tabuľke *História* nenachádzajú budú mať uvedenú nulu.²¹

Po vytvorení a uložení dotazu ho na navigačnej table označíme a klikneme na *Formulár* na karte *Vytvoriť*. Spolu s údajmi z tabuľky *Čitateľ* nám dotaz dodá aj počet výskytov daného čitateľa v tabuľke *História*.

Jedinou chybou na kráse je nezobrazovanie sa zaškrťavacieho políčka pri členskom. Namiesto neho sa zobrazuje 0 pre nezaškrtnuté a -1 pre zaškrtnuté políčko. To sa ale dá jednoducho vyriešiť. V návrhovom zobrazení na karte *Návrh* v skupine *Ovládacie prvky* klikneme na *Začiarkavacie políčko* a kliknutím na plochu formulára ho umiestnime. Toto políčko je zatiaľ neviazané. Musíme ho prepojiť so zdrojom. Po označení originálneho poľa vo formulári si v háрку vlastností môžeme na karte *Údaje* prečítať, že *Zdroj ovládacieho prvku* je *FirstzČlenské* (môže sa líšiť, napríklad, ak by sme si polia v dotaze popisovali). Označíme naše začiaravacie políčko a nastavíme zdroj na zistené *FirstzČlenské*. Pôvodné môžeme odstrániť (kláves Delete alebo kontextové menu).

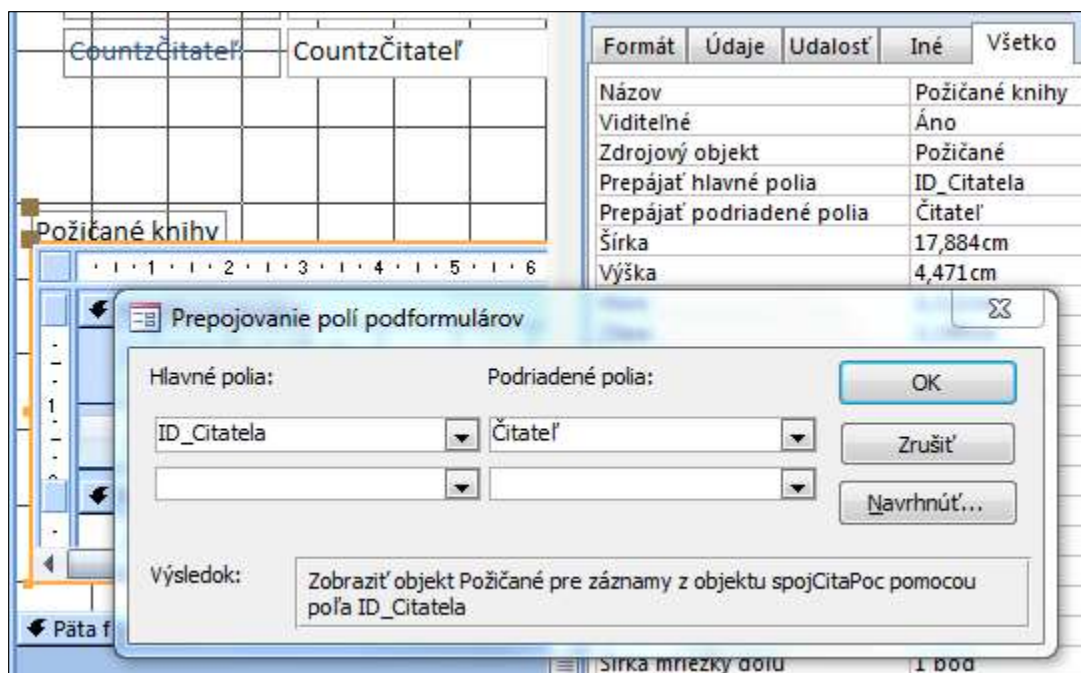
Oproti riešeniu s podformulárom sa tento postup líši aj v tom, že nemôžeme meniť údaje. Pri počte prečítaných kníh je to pochopiteľné – je to vypočítaná hodnota. Aj ostatné (meno, priezvisko, členské) nepochádzajú z tabuľky, ale z výberového dotazu, ktorý hodnoty nemení. To môže byť samozrejme niekedy nevýhoda. Ale bežne to problém nie je. A logicky by sa formulár pre zobrazovanie požičaných kníh ani nemal používať na zmenu údajov o čitateľoch. Na to by slúžil iný formulár, ktorý by pracoval s tabuľkou alebo využíval aktualizáčne dotazy.

²¹ Ako by sa prejavilo, ak by sme vybrali tretí typ spojenia?



Obr. 4.10. Dotaz pre spojenie počtu výskytov čitateľa v tabuľke *História* s tabuľkou *Čitatelia*

Do nášho nového formulára môžeme vložiť podformulár *Požičané*, ktorý sme vytvorili v kapitole 4.2.1. a je zobrazený na obr. 4.6. Aby mohol fungovať musíme urobiť malú opravu v jeho prepojení s hlavným formulárom. Táto oprava je zachytená na obr. 4.11. Po vložení alebo po jednom kliknutí do ľavého horného rohu podformulára vyberieme v háрку vlastností *Prepájať hlavné polia* alebo *Prepájať podriadené polia* a nastavíme ich (Access väčšinou ponúkne správne prepojenie). Okrem pozície na obrázku ich možno nájsť v háрку vlastností aj na karte *Údaje*.

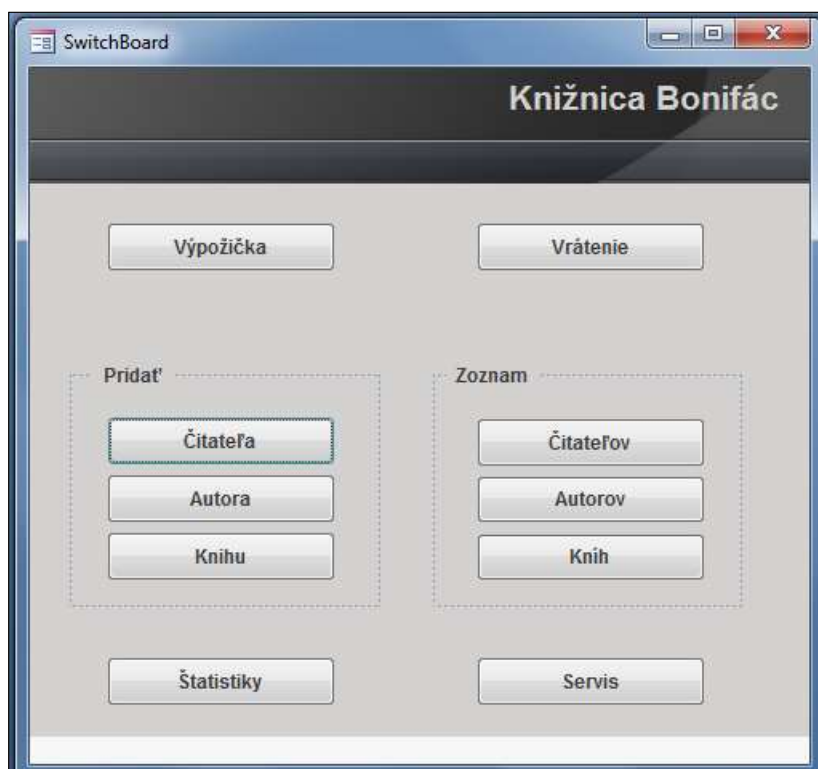


Obr. 4.11. Aby podformulár fungoval s novým formulárom, musíme upraviť prepojenie

4.4. Switchboard

Access môžeme nastaviť tak, aby sa po otvorení databázy automaticky spustil niektorý formulár. Na tomto formulári môžeme mať tlačidlá pre spúšťanie dotazov, formulárov, makier alebo zostáv. Je to výhodné ak máme veľa pomocných objektov, ktoré nevyužívame priamo a „pletú sa a zavadzajú“ na navigačnej table. Takto budeme mať prístup ku všetkým dôležitým objektom hneď po otvorení databázy.

Formulár, ktorý sa ma spustiť nastavíme cez tlačidlo *Office, Možnosti programu Access, Aktuálna databáza, Zobraziť formulár*. Pretože sa formulár vyberá z aktuálnej databázy, môžeme (musíme) si pre každú databázu vybrať formulár zvlášť.

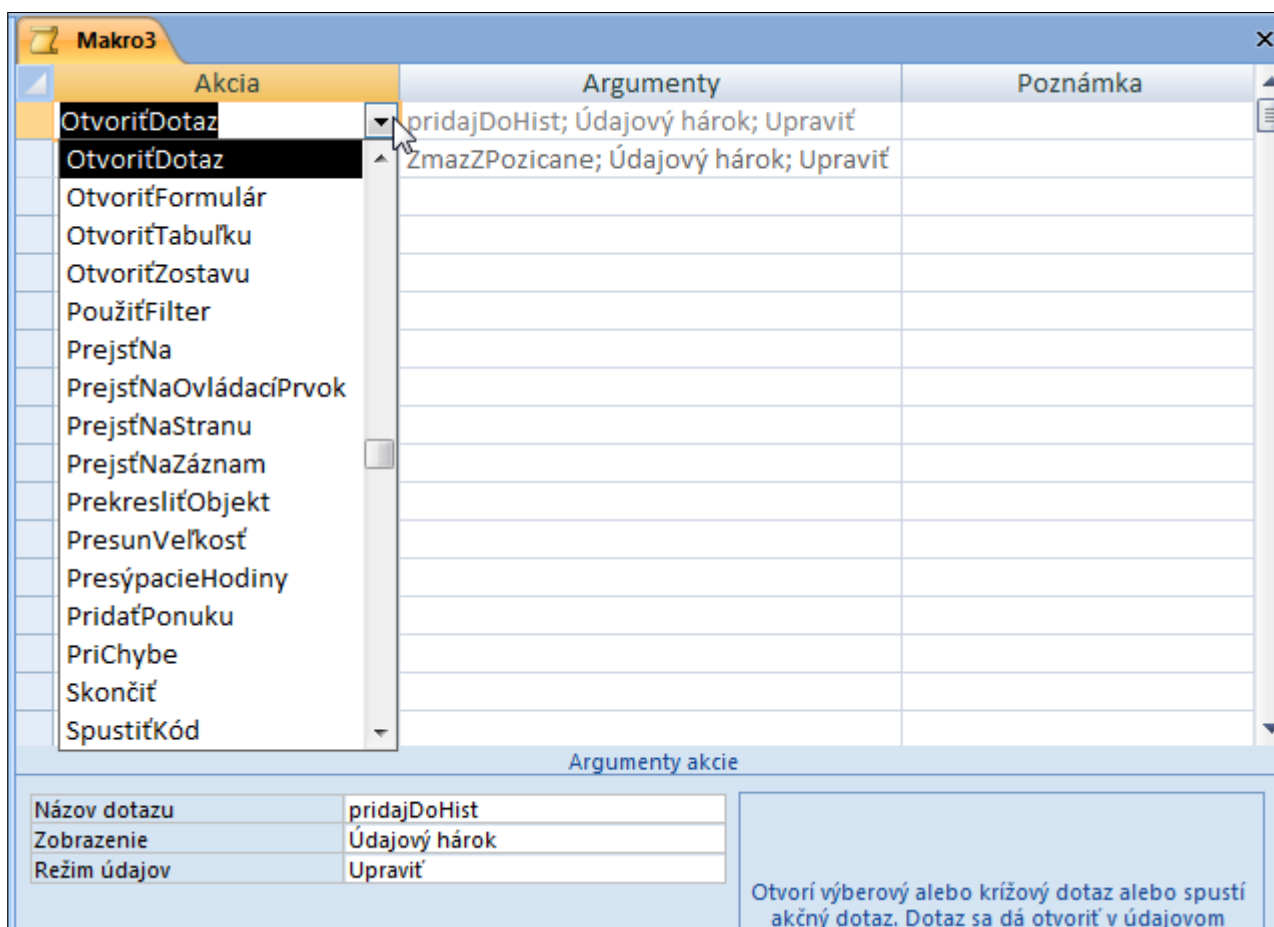


Obr. 4.12. Ukážka switchboardu

5. Makrá

Makrá sú vcelku úspešným spôsobom ako priblížiť obyčajným používateľom - neprogramátorom programovanie a jeho silu. Makrá umožňujú zautomatizovať často vykonávané (a stále rovnaké) činnosti. Ako už bolo spomenuté v kapitole 1.5., vytvorenie makra v Accessse pozostáva z činností - akcií, ktoré si vyberáme zo zoznamu.

Na obr. 5.1. je znázornený výber akcie z tohto zoznamu. Každá akcia môže mať nejaké argumenty. Argumenty zadávame v spodnej časti, často si stačí vybrať z ponuky. Na obrázku je vybraná akcia *OtvoriťDotaz*. Táto akcia má tri argumenty. Prvým z nich je názov otváraného dotazu. Access aj v príklade na obrázku ponúkol všetky dotazy v databáze a my sme si len z tejto ponuky vybrali. Ďalšími argumentmi sú v tomto prípade spôsob zobrazenia (návrhový alebo zobraziť/vykonať) a režim údajov (či možno dáta meniť, pridávať alebo iba čítať).



Obr. 5.1. Vytvorenie makra

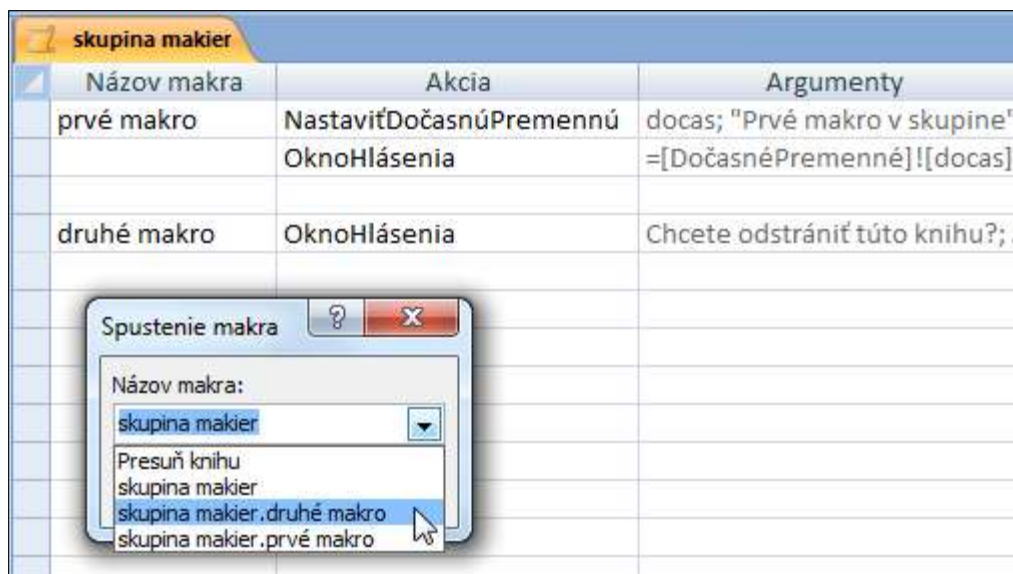
Microsoft považuje niektoré akcie za potenciálne nebezpečné, preto nie sú v štandardnej ponuke dostupné. Ak ich chceme používať, musíme ich povoliť cez kartu *Návrh* v skupine *Zobraziť alebo skryť* príkazom *Zobraziť všetky akcie*. Na obr. 5.3 na začiatku druhého riadku je žltý trojuholník znamenajúci, že ide o potenciálne nebezpečnú akciu.

Hotové makro môžeme spustiť piatimi spôsobmi:

1. Ak je otvorené a aktívne, tak z karty *Návrh* príkazom *Spustiť*.
2. Pri zavretom alebo neaktívnom z karty *Databázové nástroje*, v skupine *Makro* príkazom *Spustiť makro*.
3. Z navigačnej tably, z kontextového menu (po kliknutí pravým tlačidlo myši na názov makra) príkazom *Spustiť*.
4. Pomocou klávesovej skratky.

5. Z nejakého iného makra.

Pri vytváraní makra môžeme jednotlivé časti pomenovať. Stĺpec s pomenovaním zobrazíme kliknutím na *Názvy makier* na karte *Návrh* v skupine *Zobraziť alebo skryť*. Pomenovaním môžeme zdefinovať v jednom objekte makier tzv. skupinu makier. Pri spustení prvými dvomi spôsobmi máme možnosť vybrať, ktoré makro zo skupiny sa spustí. Situácia je znázornená na obr. 5.2. Pri spustení tretím spôsobom alebo pri výbere podľa názvu objektu (na obrázku *skupina makier*) sa spustí iba prvé makro. V ponuke sú makrá v rámci skupiny utriedené podľa abecedy a preto poradie v ponuke nemusí zodpovedať poradiu v návrhu (na obrázku je *prvé makro* uvedené v ponuke ako posledné).



Obr. 5.2. Skupiny makier

Na karte *Návrh* v skupine *Zobraziť alebo skryť* sa nachádza aj príkaz *Podmienky*. Tento príkaz pridá do návrhu makra stĺpec pre podmienky. Štandardne sa každá akcia v makre vykoná. V reálnom programe (podobne ako v živote) sa ale činnosti vykonávajú alebo nevykonávajú na základe iných skutočností najčastejšie na základe výsledku predchádzajúcich akcií (zmrzlinu iba ak zjeme obed).

Podmienka sa zadáva ako logický výraz, t. j. výraz, ktorý po vyhodnotení môže nadobudnúť iba logickú nulu alebo jednotku. Preto býva podmienkou často porovnanie. Napríklad hľadať autora budeme iba ak používateľ zadal aspoň jeden znak. V takom prípade by sa do podmienky zapísalo porovnanie s prázdny reťazcom, teda niečo takéto *HladaneMeno* <> "". Ak je hľadané meno hocičo iné ako prázdny reťazec, tak je podmienka pravdivá a akcia hľadať má význam. V opačnom prípade je výraz nepravdivý (nie je pravda, že sa nerovná "") a hľadanie sa nevykoná.

Veľmi časté využitie v podmienke má slovo *False* (Nepravda). V takom prípade sa akcia samozrejme nevykoná. Význam to má, ak máme v makre viacero akcií a makro nepracuje ako by (podľa nás) malo²². Aby sme nemuseli akcie mazať a opätovne ich aj s argumentmi pridávať, stačí ich pomocou nepravdivej podmienky dočasne vyradiť.

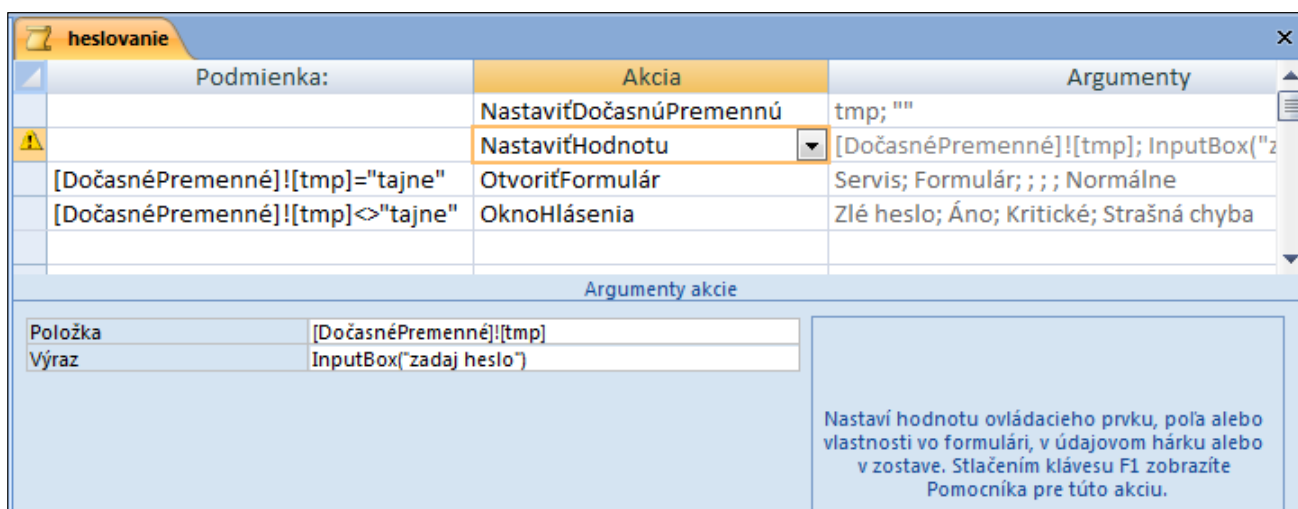
Pre hľadanie chýb v makre je určené aj spustenie po jednom kroku, tzv. krokovanie, ktoré sa nachádza na karte *Návrh* v skupine *Nástroje* pod názvom *Jeden krok*. Pokiaľ je aktívne, tak sa makro po spustení po každom kroku preruší.

Príkladom pre využitie podmienky je jednoduché makro, ktoré spustí nejaký formulár *Servis*, iba ak zadáme správne heslo.²³ Makro je zobrazené na obr. 5.3. Využíva v druhom riadku funkciu *InputBox*, ktorá zobrazí okno, do ktorého používateľ napíše heslo. To čo používateľ napíše sa uloží do dočasnej premennej

²² Tento jav je omnoho častejší ako si neprogramátori myslia.

²³ Toto makro, vôbec nie bezpečné, slúži iba ako ukážka niektorých akcií a podmienok. Bežné zabezpečovacie mechanizmy sa naň vôbec nepodobajú.

s názvom *tmp*²⁴. V prvom riadku sme ju vytvorili (makro by fungovalo aj bez toho, premenná by sa vytvorila automaticky pri použití v druhom riadku). V treťom a štvrtom riadku porovnávame, či je v premennej *tmp* slovo „tajne“, t. j. naše heslo. Ak áno, otvoríme formulár *Servis*. Ak je v premennej čokoľvek iné, použijeme akciu *OknoHlásenia* a vypíšeme oznam „Zlé heslo“.



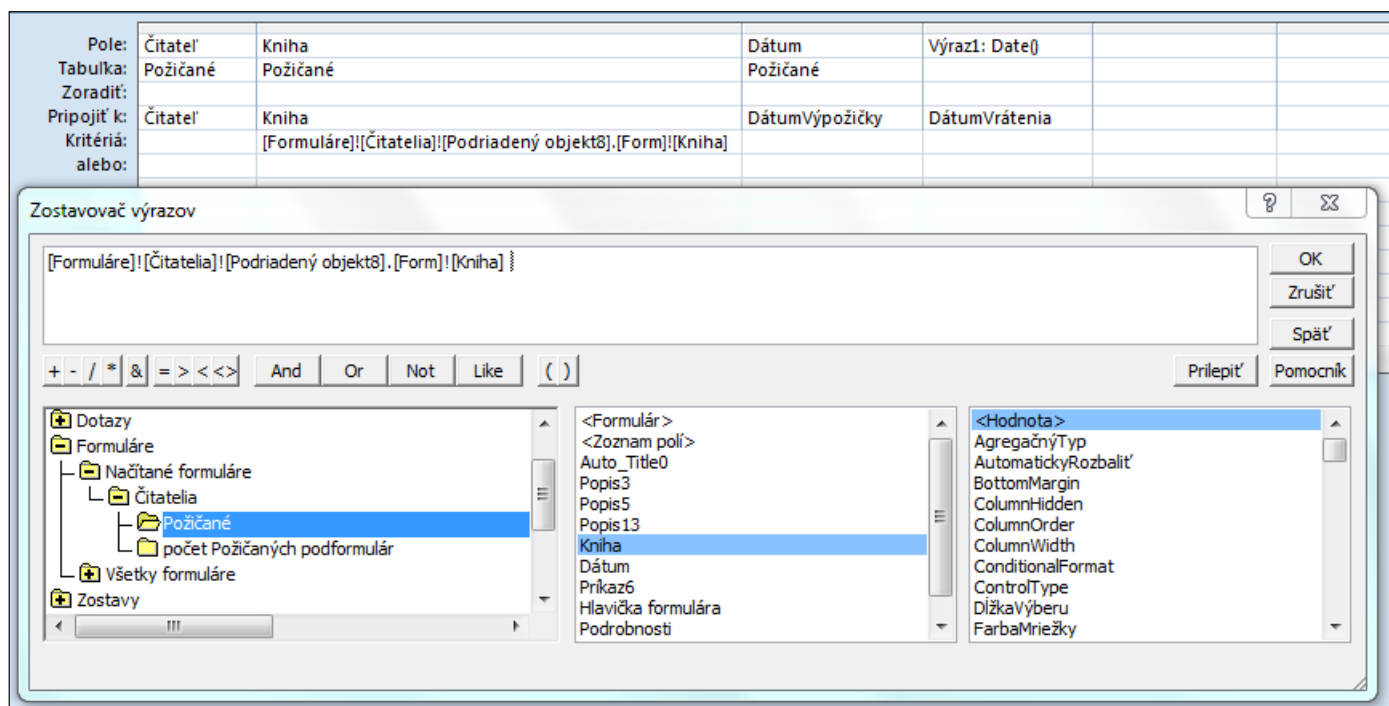
Obr. 5.3. Makro testujúce vstup od používateľa

V našom formulári *Čitateľa* chceme, aby sa po stlačení tlačidla pri knihe premiestnil záznam z tabuľky *Požičané* do tabuľky *História*. V *Accesse* to zrealizujeme tak, že najprv zapíšeme dáta do tabuľky *História* a potom ich z tabuľky *Požičané* odstránime.

Celé to zrealizujeme v dvoch krokoch, dotazoch:

- pridajDoHist, zistí aké údaje sa majú pridať a pridá ich. Je to teda dotaz pripájací. Dáta hľadáme v tabuľke (používame tabuľku) *Požičané* a pripájame samozrejme do tabuľky *História*. Dotaz je zobrazený (aj so zostavovačom výrazov) na obr. 5.4. Pri tomto dotaze potrebné si je uvedomiť dve skutočnosti:
 - Z tabuľky *Požičané* môžeme čerpať iba čitateľa, knihu a dátum výpožičky. Ale do tabuľky *História* pridávame aj dátum vrátenia knihy. V dotaze to riešime štvrtým stĺpcom funkciou *Date()*, ktorá vracia aktuálny dátum.
 - Musíme určiť, ktorý záznam sa má pridať do tabuľky *História*. Vo formulári *Čitateľa* sme v podformulári *Požičané* na zázname knihy, ktorú chceme vrátiť. Týmto kritériom vyberieme jediný záznam (knihu môže byť v jednom momente požičaná iba raz). Dôležité je, že musíme ísť v zostavovači výrazov cez *Načítané formuláre* a mať pri tom otvorený formulár *Čitateľa*. Nemôžeme použiť (ak by sme šli cez *Všetky formuláre*) *[Formuláre]![Požičané]![Kniha]*, pretože tento formulár nemusí byť v čase spustenia makra vôbec otvorený. Samotná podmienka môže u čitateľa vyzeráť mierne inak.
- ZmazZPozicane, odstráni správny záznam z tabuľky *Požičané*. Dotaz je odstraňovací a je veľmi podobný prvému dotazu. Je znázornený na obr. 5.5.

²⁴ Premenná v programovaní je trochu podobná schránke (Ctrl+C). Môžeme do nej odložiť nejaké dáta. Pretože pri programovaní potrebujeme premenných viac, tak ich pre rozlíšenie pomenovávame.

Obr. 5.4. Dotaz na pridanie záznamu z tabuľky *Požičané* do tabuľky *História*

| | | |
|------------|---|------------|
| Pole: | Kniha | Požičané.* |
| Tabuľka: | Požičané | Požičané |
| Odstrániť: | Kde | Z |
| Kritériá: | [Formulár]![Čitateľa]![Podriadený objekt8].[Form]![Kniha] | |
| alebo: | | |

Obr. 5.5. Dotaz na zmazanie záznamu z tabuľky *Požičané*

Ak po stlačení tlačidla na zmazanie knihy vo formulári *Čitateľa* (obr. 4.3)²⁵ naše dvojriadkové makro funguje (nezobrazí sa chybové hlásenie), tak sa záznam sa presunie do tabuľky *História*. Vo formulári ale ostane kniha zobrazená. Treba preto formuláru povedať, že v zdrojových tabuľkách sa niečo zmenilo. Napríklad kliknutím na zmazanú knihu alebo počkať nejaký čas²⁶. To ale určite nie je dobrá črta nášho makra. Navyše kniha nezmizne úplne, ale namiesto nej ostane prázdny záznam, tak ako je to zobrazené na obr. 5.6.

²⁵ Nepoužívať samostatne podformulár *Požičané*. Dotazy fungujú v rámci hlavného formulára a podformulára. Nie je ale problém prípadne ich prerobiť na fungovanie v podformulári.

²⁶ Čas obnovovania sa dá nastaviť v možnostiach programu Access.

| Požičané knihy | | |
|---------------------|-------------|--------------------------|
| Kniha | Dátum | Vrátiť |
| Pohár čistej vody | 5.8.2010 | <input type="checkbox"/> |
| | #Odstránené | <input type="checkbox"/> |
| Srdce blesk nespáli | 5.8.2010 | <input type="checkbox"/> |
| E. T. mimozemšťan | 5.8.2010 | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

Obr. 5.6. Zobrazenie odstráneného záznamu

Ak sa posunieme na nasledujúceho čitateľa a naspäť, tak sa **odstránené** záznamy stratia. To by bolo jedno z rýchlych riešení: Pridať do makra ešte dve akcie *PrejsťNaZáznam*. Táto akcia má 4 argumenty:

- Typ objektu: vyberieme formulár.
- Názov objektu: *Čitatelia*.
- Záznam: nasledujúci, v druhej akcii predchádzajúci.
- Počet záznamov: 1.

Toto riešenie funguje, hneď po zmazaní sa takto obnovia všetky záznamy. Ale je vhodný skôr na vysvetlenie tejto akcie²⁷. Reálne riešenie by bolo napríklad v akcii *Znova spustiť dotaz*, ktorá opätovne načíta (a prípadne prepočíta) hodnoty polí v aktívnom objekte. V našom prípade argument nepoužijeme.

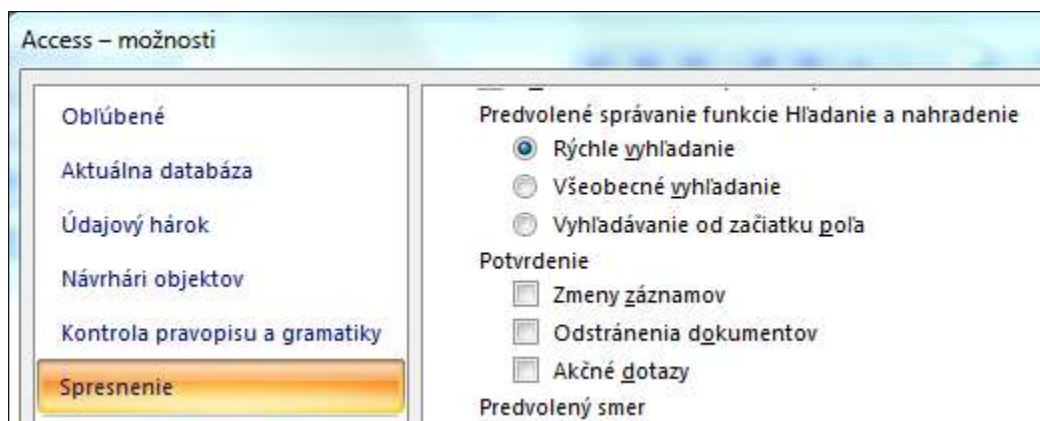
Pokiaľ sme na počet požičaných kníh použili podformulár a nie spojenie tabuliek, zistíme, že sa neobnovuje. Akcia *Znova spustiť dotaz* obnovuje iba aktuálny objekt a tým je podformulár *Požičané*. My však chceme aktualizovať iný objekt – podformulár s „krásnym“ a dlhým názvom *počet Požičaných podformulár*.²⁸

To vyriešime akciou *SpustiťKód*. Argumentom tejto akcie môže byť funkcia programovacieho jazyka Visual Basic. My využijeme funkciu *query*, ktorá je obdobou akcie *Znova spustiť dotaz*. Pretože sa nachádzame v (aktívny objekt je) podformulári *Požičané*, musíme akoby vyjsť o jednu úroveň vyššie pomocou príkazu *parent*. Celý príkaz (argument) je *parent.query*. Tak sa aktualizuje celý hlavný formulár aj s podformulármi. Ak by sme chceli aktualizovať len susedný podformulár *počet Požičaných podformulár* použili by sme: *parent.[počet Požičaných podformulár].query*. To sa podobá na situáciu, keď sme v pivnici a chceme ísť do druhej, ktorá je prepojená chodbou o poschodie vyššie. Musíme ísť hore (*parent*) a až potom zostúpiť do druhej pivnice *[počet Požičaných podformulár]*.

Pri každom stlačení tlačidla sa spustia dva akčné dotazy. Jeden pridáva záznam do tabuľky, druhý z inej tabuľky záznam odstraňuje. Access štandardne požaduje potvrdenie takýchto činností, čo vyžaduje niekoľko potvrdení. Jednou z možností ako sa nutnosti potvrdzovania zbaviť je vypnúť potvrdzovanie akčných dotazov v možnostiach Accessu tak ako je to na obr. 5.7.

²⁷ A vyskúšanie.

²⁸ Tak by to nepomenoval žiaden normálny človek - tak to nazval Access. A niekto bol lenivý to premenovať. A teraz netrpí len on ale aj iní. Berte si z toho poučenie.



Obr. 5.7. Vypnutie potvrdzovania akčných dotazov

Pre priradenie klávesovej skratky makru vytvoríme alebo otvoríme makro s názvom *AutoKeys*. Akcia bude *SpustiťMakro* a do stĺpca s názvom makra zapíšeme klávesovú skratku:

- ^ znamená (bežne v počítačoch, nielen teraz) kláves Ctrl, teda „^E“ znamená stlačenie Ctrl+E.
- + znamená kláves Shift, teda „+E“ znamená stlačenie Shift+E.
- Možno použiť aj {F1}, {F2}, ... , {INS}, {DEL}, šípky = {UP}, {RIGHT}, ... , {HOME}, {PGUP}, {ESC}, atď.

Znaky možno kombinovať. Napríklad zápis „^+E“ znamená stlačenie Ctrl+Shift+E a „^{DEL}“ Ctrl+Delete.

Ak použijete kombináciu, ktorú už Access používa, bude platiť nová skratka.

Ak chceme aby sa makro spustilo automaticky s otvorením databázy stačí ho pomenovať *AutoExec*.

<http://office.microsoft.com/sk-sk/access-help/zakladne-informacie-o-makrach-v-programe-access-2007-HA001214202.aspx>

<http://office.microsoft.com/sk-sk/access-help/akcia-makra-nastavitdocasnupremennu-HA010120216.aspx>

<http://office.microsoft.com/sk-sk/access-help/akcia-makra-najstzaznam-HA001226217.aspx>

6. Zostavy

Zostavy sú určené na tlač alebo prezentáciu údajov. Aj keď vytlačiť môžeme priamo aj tabuľku, dotaz či formulár, zostavy poskytujú pri tlačení širšie možnosti. Taktiež umožňujú zoskupiť údaje podobne ako pri dotazoch (*Group by*) a tak počítat počty a medzisúčty. Spôsob prezentovania výsledkov je veľmi dôležitý a často môžu prehľadné a pekne upravené výstupy pozitívne ovplyvniť rozhodnutie.

Vytvorenie zostavy v mnohom pripomína vytvorenie formulára. Pri formulári pribudli dve kontextové karty *Usporiadať* a *Návrh* v návrhovom zobrazení resp. *Formátovanie* v zobrazení rozloženia. Pri zostave sa tieto karty nachádzajú tiež a okrem nich ešte aj karta *Nastavenie strany*. Môžeme v nej nastaviť veľkosť, orientáciu, okraje strany, prípadne či sa majú tlačiť len údaje (v takom prípade sa netlačia napríklad záhlavia stĺpcov či nadpisy). Tieto nastavenia sa dajú meniť aj pomocou ďalšieho spôsobu zobrazenia – *Ukážka pred tlačou*. Okrem spomenutých nastavení tu môžeme aj exportovať údaje do súborov iných formátov. V prípade, že sa export bude opakovať pravidelne, môžeme si túto činnosť nechať zapísať do kalendára Outlooku.

Kontextové karty sú rovnaké len v karte *Návrh* resp. *Formátovanie* je jedna skupina navyše. Volá sa *Zoskupenie a súčty*. Obsahuje tri príkazy *Zoskupiť a zoradiť*, *Skryť podrobnosti* a *Súčty*.




Na karte *Vytvoriť* v skupine *Zostavy* je niekoľko spôsobov vytvorenia zostavy:

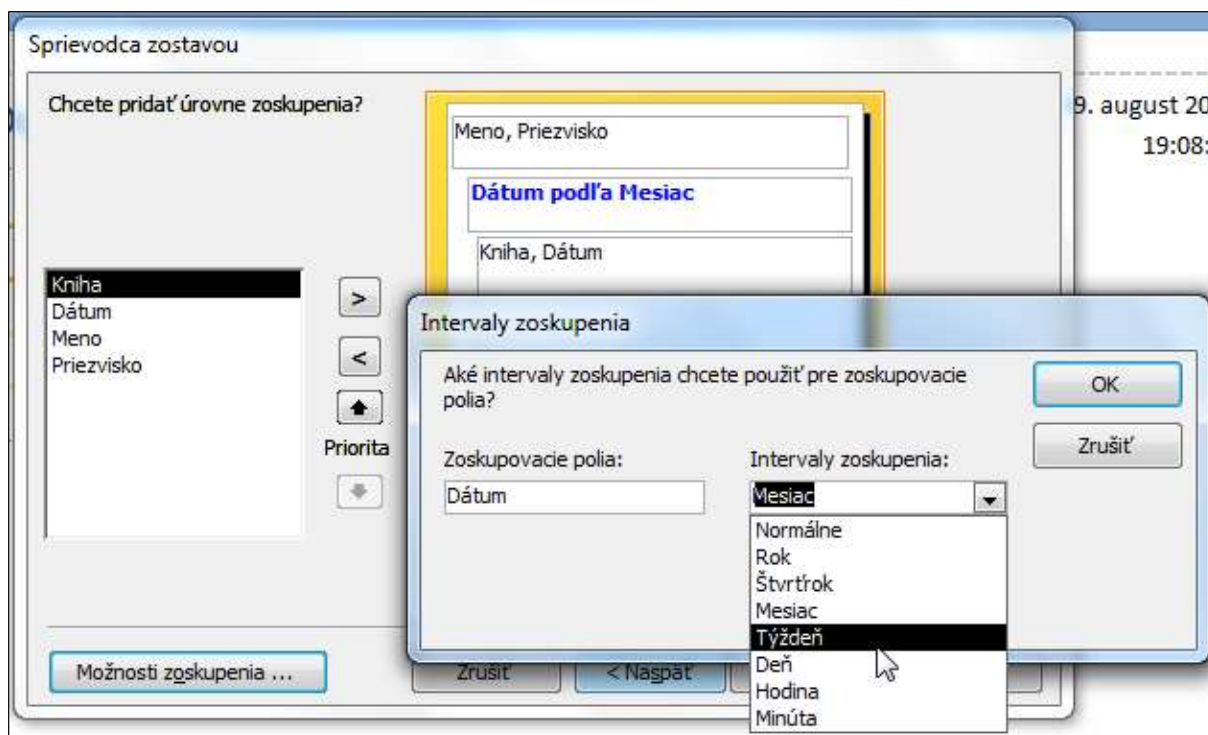
- Možnosť *Zostava* je aktívna iba ak je v navigačnej table označená tabuľka, neakčný dotaz alebo formulár.
- *Sprievodca zostavou*, ktorý na základe niekoľkých otázok pomôže vytvoriť zostavu.
- *Prázdna zostava* a *Návrh zostavy*. Obe možnosti umožňujú vytvoriť zostavu od základov. Líšia sa len úvodným zobrazením (zobrazenie rozloženia alebo návrhové zobrazenie). Častejšie sa používajú iné spôsoby vytvorenia, ktoré sa dodatočne upravujú v niektorom z týchto zobrazení.
- *Menovky*. Podobné ako menovky v hromadnej korešpondencii vo Word. Aj táto možnosť je aktívna iba ak je v navigačnej table označená tabuľka, neakčný dotaz alebo formulár.

V nasledujúcich riadkoch je popísané vytvorenie zostavy pomocou voľby *Zostava* a *Sprievodca zostavou*.

6.1. Vytvorenie zostavy pomocou sprievodcu

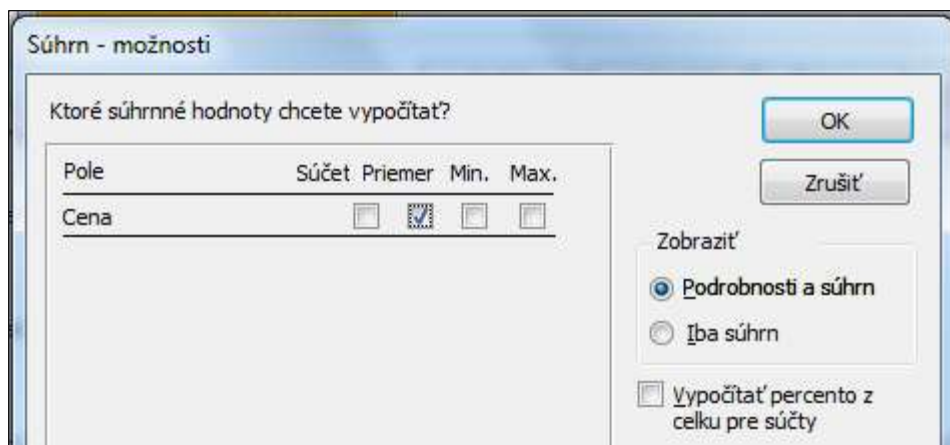
Vytvoríme si zostavu, v ktorej budú čitatelia, ktorí majú aktuálne požičané knihy.

1. V prvom kroku vyberieme tabuľky a dotazy, z ktorých chceme vytvoriť zostavu. Vyberieme tabuľku *Požičané* a z nej polia *Kniha* a *Dátum*. Potom vyberieme z tabuľky *Čitatelia* polia *Meno* a *Priezvisko*. Ak v zostave využívame viacero tabuliek, musia mať medzi sebou definované relácie.
2. V druhom kroku môžeme určiť prípadné zoskupenie údajov. V našom prípade by sa hodilo zoskupiť podľa čitateľa. Kliknutím na *podľa Čitatelia* sa na prvé miesto posunú polia *Meno* a *Priezvisko*.
3. Teraz môžeme pridávať ďalšie úrovne zoskupenia. V rámci každého čitateľa môžeme zoskupovať podľa dátumu. Toto zoskupenie vytvoríme dvojklikom na dátum v ľavej časti alebo označením a stlačením tlačidla . Analogicky môžeme z pravej strany pole odstrániť. Prioritu úrovni zoskupovania môžeme po ich označení na pravej strane meniť tlačidlami  a . Spôsob zoskupovania môžeme spresniť pomocou nastavenia *Možnosti zoskupenia* ... Tento krok je zobrazený na obr. 6.1.
4. Ďalej môžeme nastaviť triedenie, potom určíme rozloženie zostavy, štýl a názov zostavy. Takto vytvorenú zostavu možno ihneď zobrazit alebo ďalej upravovať.



Obr. 6.1. Sprievodca zostavou, tretí krok.

Je možné, že šírky stĺpcov budú nastavené nevhodne. To najľahšie vyriešime, podobne ako pri formulároch, v režime *Zobrazenie rozloženia*. Podobne ako pri formulároch môžeme meniť formát jednotlivých polí, prípadne v návrhovom režime polia pridať. Pri úprave v návrhovom zobrazení si všimneme, že jeden dôležitý rozdiel oproti formulárom tu predsa len je. Hlavičky sú viaceré – podľa úrovni zoskupovania.



Obr. 6.2. Možnosti súhrnu

V kroku, kde môžeme nastaviť triedenie sa ponúka pre každé nezoskupované číselné pole možnosť využiť agregáciu funkciu - *Možnosti súhrnu*. Napríklad, ak vyberieme z tabuľky *Knihy* polia *Názov*, *Cena*, z tabuľky *Žáner* pole *Žáner* a zoskupíme podľa neho, tak v štvrtom kroku môžeme určiť priemernú cenu jednotlivých žánrov. Situácia je znázornená na obr. 6.2.

Okrem toho si môžeme v *Možnostiach súhrnu* vybrať či chceme

- súhrn aj podrobnosti, výsledok je na obr. 6.3., alebo
- iba súhrny (v tom prípade sme vybrali pole *Názov* zbytočne) a výsledok je na obr. 6.4.

| Žáner | Názov | Cena |
|--|------------------------|---------|
| Detektívka | | |
| | Heeresma | 39,00 € |
| | Chuligán | 40,00 € |
| | | |
| | | |
| | Motív | 99,00 € |
| | Chelsejské vraždy | 30,00 € |
| Súhrn pre 'Žáner' = Detektívka (30 podrobné záznamy) | | |
| Avg | | 35,55 |
| Ieporelo | | |
| | Hádanky | 10,00 € |
| | Čo sa sníva zvieratkám | 10,00 € |
| | | |
| | | |

Obr. 6.3. Medzisúčty podľa žánra – priemerná cena. Aj s podrobnosťami

| Žáner | Názov | Cena |
|---|-------|--------|
| Detektívka | | |
| Súhrn pre 'Žáner' = Detektívka (30 podrobné záznamy) | | |
| Avg | | 35,55 |
| Ieporelo | | |
| Súhrn pre 'Žáner' = Ieporelo (15 podrobné záznamy) | | |
| Avg | | 10,467 |
| liter. faktú | | |
| Súhrn pre 'Žáner' = liter. faktú (3 podrobné záznamy) | | |
| Avg | | 43,333 |

Obr. 6.4. Medzisúčty podľa žánra – priemerná cena. Bez podrobností

6.2. Vytvorenie zostavy na základe tabuľky príkazom Zostava

Možnosť *Zostava* si ukážeme na zostave založenej na tabuľke *Knihy*. V navigačnej table označíme (klikneme) túto tabuľku. Na karte vytvoriť klikneme na príkaz *Zostava*. Vznikne zostava zobrazená na obr. 6.5.

Jednotlivé polia môžeme formátovať po ich označení. Po kliknutí na niektoré pole sa označí nielen pole ale aj celý stĺpec. Napríklad na obr. 6.5. je označený celý stĺpec po kliknutí na jeho popis. Po označení stĺpca ho môžeme potiahnutím posúvať do strán alebo meniť jeho šírku.

Pre zmenu výšky musíme v niektorých prípadoch použiť hárok vlastností a je pravdepodobné, že výšku budeme môcť iba zväčšiť. Access sa snaží zobrazíť celý obsah textového poľa a preto ho pri úzkom stĺpci zobrazí do viacerých riadkov, čoho výsledkom je vyšší riadok. Zníženie riadku možné v návrhovom zobrazení, zmenšením časti *Podrobnosti*. Pretože táto časť nemôže byť nižšia ako najvyššie pole, je možné, že pred jej zmenšením bude potrebné zmenšiť výšku polí.

Pri formátovaní obsahu poľa je jedno, ktorý záznam označíme. Napríklad pre úpravu obsahu poľa *Cena* môžeme označiť sumu 32 alebo 26. Zmena formátu by platila na všetky ceny. Popisy sa ale formátujú zvlášť. Po dvojkliku alebo v hároku vlastností môžeme text v popise meniť.

| Jazyk | Názov | Cena | ID_Autora | ID_Žánra | El verzia |
|------------|--------------------------|---------|-----------|--------------------|-----------|
| slovenčina | ...a poznali iba kameň | 32,00 € | Brinke | odborná literatúra | |
| slovenčina | ...a predsa sú dva svety | 26,00 € | Hulman | odborná literatúra | |

Zoskupenie, zoradenie a súčet

Obr. 6.5. Zostava – Zobrazenie rozloženia.

V spodnej časti môžeme určiť, podľa ktorých polí budeme triediť a tiež ako budeme zoskupovať. Táto časť sa zobrazuje a skrýva príkazom *Zoskupiť a zoradiť* v skupine *Zoskupenie a súčty* na karte *Návrh* resp. *Formátovanie*.

Po kliknutí na *Pridať zoradenie* môžeme vybrať, podľa ktorých polí sa má zostava utriediť. Môžeme zvoliť viacero polí a meniť ich poradie.

Príkazom *Pridať skupinu* zoskupíme podľa jedného či viacerých polí. Vyberieme zoskupenie podľa *ID_Žánra*. Podobne by sme mohli vybrať ďalšie zoskupovania v rámci žánra, napríklad podľa jazyka. Potom by sme kliknutím na šípky hore/dole v pravej časti okna mohli zoskupovanie zmeniť na zoskupovanie podľa jazyka a v rámci neho podľa žánra.

Po zoskupení môžeme okrem usporiadania zoskupujúceho poľa kliknutím na príkaz *Ďalšie*:

- Definovať spôsob zoskupovania (*podľa celej hodnoty*). Pri našom zoskupení podľa žánra nemá význam ale pri zoskupení podľa ceny by bol veľmi užitočný. Ak by sa malo zoskupovať podľa každej ceny asi by to nemalo význam, vznikali by totiž napríklad skupiny typu 3,20 a aj 3,25. Preto možno nastaviť, aby do jednej skupiny patrili knihy s cenou 0 do ďalších podľa intervalu cien napríklad (0,5), <5,10), <10, 15) a pod. Tento prípad je zachytený na obr. 6.6., kde vidíme, že je 25 kníh s cenou od 0 po 5, 90 kníh v cene 5 až 10 atď. Veľkosť intervalu sa dá meniť. Access ponúka toto číselné intervalové zoskupovanie aj pre náš príklad, pretože *ID_Žánra* je číslo. Ale ťažko vymyslieť aký by v tomto prípade malo zoskupovanie podľa intervalov zmysel.

V prípade, že zoskupujeme podľa textového poľa (napríklad názvu knihy), môžeme určiť podobným spôsobom na základe koľkých prvých písmen sa vytvárajú skupiny. Ak by sme vybrali iba jedno, vznikli by bežné skupiny podľa abecedy. V prípade inventárnych čísel (text typ) by mohlo byť zadané, že prvé dve písmena identifikujú budovu, tretie poschodie. Tu by sme mohli zoskupovať v rámci dvoch písmen (t. j. budov) a v rámci nich po troch – poschodiach. Obdobné je to pri dátumoch.

- Určiť popis zoskupujúceho stĺpca (príkaz *s názvom*, zmeniť popis *ID_Žánra* napríklad na *Žáner*).
- Nastaviť zobrazovanie hlavičky a päty.
- Či môže ostať na konci strany osamotený riadok (*neudržiavať skupinu na jednej strane*).
- Určiť súhrny (zobrazuje sa *Bez súčtov* alebo *so zhrnutým polom XY*). Môžeme vybrať aké agregáčne funkcie budeme vyžadovať. Po výbere poľa, určíme typ súhrnu a začiarukneme aspoň

jedno políčko. Potom to môžeme opakovať pre iné polia. Ak chceme súhrny z niektorého poľa odstrániť, tak to pole vyberieme a odstránime všetky začiarknutia.

Namiesto príkazu v poslednej odrážke môže byť pre vytvorenie súhrnov pohodlnejšie kliknúť do zostavy na pole a v skupine *Zoskupenie a súčty* vybrať požadované typy súhrnov pre toto pole. Postup uvedený v odrážke ale poskytuje viac možností.

| Cena | Počet kníh |
|--------------|------------|
| Cena - | 41 |
| Cena 0 - 4 | 25 |
| Cena 5 - 9 | 90 |
| Cena 10 - 14 | 416 |
| Cena 15 - 19 | 655 |
| Cena 20 - 24 | 538 |

Obr. 6.6. Zoskupovanie podľa intervalu ceny

Ak by sme chceli vytvoriť zostavu podobnú tej na obr. 6.4., tak by sme

- zoskupili podľa žánra,
- nastavili súhrn priemerná cena,
- prípadne zmazali nepotrebné stĺpce a
- klikli na *Skryť podrobnosti* v skupine *Zoskupenie a súčty*.

7. Funkcie

S funkciami sa môžeme stretnúť v databázach na každom kroku: v dotazoch, vo formulároch aj v zostavách. Niektoré sme už aj využili - funkciu *Dcount* pri formulároch v kapitole 4.2.2. a spojenie textových reťazcov operátorom *&* pri dotazoch na strane 39 a funkciu *Date* dokonca dvakrát na stranách 43 a 66.

Používali sme funkcie *priemer*, *počet*, *minimum* alebo *maximum*. Pri tabuľkách stačilo na karte *Domov*, v skupine *Záznamy* menu vybrať *Súčty* a Access počítal (pre číselné a dátumové polia) tieto funkcie bez nášho ďalšieho pričinenia. Podobne tomu bolo pri dotazoch a zostavách po zoskupení/súhrnoch.

Doteraz sme prišli do kontaktu s funkciami len letmo, dá sa povedať len tak mimochodom. Teraz sa na funkcie pozrieme troška podrobnejšie. Access ponúka desiatky funkcií, ktoré sa delia do skupín, podľa toho s akými argumentmi pracujú a čo počítajú. Nebudeme ich vysvetľovať všetky. Podrobný popis všetkých funkcií a ich argumentov sa nachádza na: <http://office.microsoft.com/sk-sk/access-help/CH010072891.aspx?CTT=97>

V tejto kapitole vyberieme z najčastejšie používaných typov niekoľko typických reprezentantov, tak aby sme ich mohli použiť na našu knižničnú databázu a odpovedať na otázky zo strany 5.

Ku funkciám sa v Accesse dostaneme cez zostavovač výrazov, ktorý sme v knižke už párkrát používali. Pri štúdiu funkcie si vždy musíme pozrieť koľko má argumentov a akého sú typu. Argumenty sa oddeľujú bodkočiarkou. Argumenty uvedené v hranatých zátvorkách sú nepovinné. Poradie argumentov sa musí dodržiavať.

7.1. Funkcie dátumu a času

Už sme používali funkciu *Date*, ktorej výsledkom (bežne sa hovorí - ktorá vráti) aktuálny dátum. Podobná je funkcia *Now*. Tá vráti aktuálny dátum a čas. Tieto funkcie nemajú argumenty. Tak ako všetky funkcie, ktoré nemajú argumenty sa píše s prázdnyimi zátvorkami: *Date()* a *Now()*.

V kapitole 2.1.2. sme sa dozvedeli, že dátum je číslo. Preto môžeme dva dátumy odčítať a zistiť tak, koľko dní je medzi nimi rozdiel. Na otázku kto mal najdlhšie požičanú knihu, nám stačí vypočítať v dotaze výraz *[DátumVrátenia]-[DátumVýpožičky]*, usporiadať zostupne a zobraziť prvý záznam.

Niekedy chceme vedieť, koľko uplynulo medzi dvoma dátumami týždňov, mesiacov, rokov a niektorí aj koľko štvrtrokov, hodín, minút alebo sekúnd. To by sme mohli zistiť vhodným vynásobením vyššie uvedeného výrazu ale napríklad pri mesiacoch by boli zrejme problémy (je 30 dní jeden mesiac?). V takom prípade môžeme použiť funkciu *Datediff*.

Funkcia má tvar *DateDiff(interval; dátum1; dátum2 [; prvýdeňtýždňa] [; prvýtýždeňroka])*. Má teda 5 argumentov, posledné dva sú nepovinné – to je znázornené pri popise hranatými zátvorkami. Popis argumentov:

- interval - môže byť
 - „yyyy“ rok, častý omyl ľudí, ktorí nedočítajú help do konca, neznamená počet rokov medzi dátumami ale počet Silvestrov.
 - „q“ - počet štvrtrokov medzi dátumami,
 - „m“ - počet mesiacov,
 - „y“ alebo „d“ počet dní,
 - „w“ počet (sedemdnových) týždňov, „ww“ počet nediel medzi dátumami (nedeľa v prvom dátume sa neráta),
 - „h“, „n“, „s“ počet hodín, minút a sekúnd medzi dátumami.
- dátum1, dátum2 – porovnávané dátumy.
- prvýdeňtýždňa – ktorým dňom začína týždeň. 1= nedeľa, 2=pondelok, ... 7= sobota, 0=podľa nastavenia systému.
- prvýtýždeňroka – kedy sa ráta týždeň ako prvý v roku. 1=1. januára, 2=prvý týždeň s aspoň 4 dňami v novom roku, 3= celý týždeň v novom roku, 0=podľa nastavenia systému.

Ak by sme chceli vypočítať koľko sekúnd nás delí od konca roka, môžeme skombinovať funkcie *Now* a *DateDiff*: *DateDiff("s"; Now(); "1.1.2011")*, namiesto zápisu dátumu v úvodzovkách môžeme použiť aj dátumový zápis *DateDiff("s"; Now(); #1.1.2011#)*. Dátum môžeme zapísať aj v tvare "1-Jan-2011"

Na obr. 7.1. vidíme ako si môžeme pochopenie funkcie a a jej argumentov otestovať pomocou dotazu. Použili sme dátumový zápis aj zápis pomocou úvodzoviek. Pri zápise s úvodzovkami sme pri druhom dátume vynechali rok – Access si pri spustení dotazu dosadí aktuálny²⁹. To znamená, že ak by sme dotaz uložili a spustili o rok resp. o dva, výsledok by bol 53 resp. 105.

| | | |
|-----------|--|--|
| Pole: | Výraz1: <i>DateDiff("w";#21.8.2010#;#22.8.2010#)</i> | Výraz2: <i>DateDiff("ww";"21.8.2010";"22.8")</i> |
| Tabuľka: | | |
| Zoradiť: | | |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | |
| alebo: | | |

Obr. 7.1. Porovnanie argumentov „w“ a „ww“ funkcie *Datediff* pomocou dotazu

Opačná funkcia k *DateDiff* je *DateAdd* (interval; číslo; dátum), ktorá k dátumu (tretiemu argumentu) pripočíta číslo (druhý argument). Prvý argument je totožný s argumentom s funkcie *Datediff* s jednou výnimkou - „w“ je totožné s „y“ a „d“. Pripočítavané číslo môže byť aj záporné.

Funkcie *Year*, *Month*, *Day*, *Hour*, *Minute*, *Second* potrebujú jeden argument – dátum a vrátia z neho príslušnú časť. To sa môže hodiť napríklad pri výskume, či sa v máji (bez ohľadu na roky) požíčiava viac poézie. Odpoveď na túto závažnú otázku poskytuje dotaz s funkciou *Month* na obr. 7.2.

; Kritériá: "poézia" (under Žáner), (under Kniha)." data-bbox="122 432 693 688"/>

| | | | |
|-----------|---|--------------------------|-------------------------------------|
| Pole: | Výraz1: <i>Month([História]![DátumVýpožičky])</i> | Žáner | Kniha |
| Tabuľka: | | Žáner | História |
| Celkom: | Group By | Where | Count |
| Zoradiť: | | | Zostupne |
| Zobraziť: | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | "poézia" | |

Obr. 7.2. Dotaz pre výskum zvislosti požíčavania poézie od mesiaca

V prípade, že by sme použili v prvom stĺpci pre zoskupenie ešte aj funkciu *MonthName*(*mesiac* [*skratka*]), vypisovali by sa namiesto čísel mesiacov ich názvy. Zápis by vyzeral *MonthName*(*Month*(*[História]![DátumVýpožičky]*)).

Funkcia má dva argumenty, prvým je číslo mesiaca – to sme zistili funkciou *Month*. Druhý je nepovinný, v prípade, že je uvedený musí to byť číslo alebo logická hodnota. Ak je rôzny od *nuly* a *False*, tak by sa mala podľa popisu funkcie zobrazit' skratka mesiaca, zobrazí sa ale číslo.

Obdobou funkcií *Month* a *MonthName* sú funkcie *Weekday* a *WeekdayName*.

Weekday(*date* [*prvýdeňtýždňa*]) vráti poradové číslo dňa v týždni a *WeekdayName*(*deňtýždni* [*skratka*] [*prvýdeňtýždňa*]) vráti jeho názov.

²⁹ V skutočnosti sme rok vynechali aj pri druhom dátume v prvom výraze ale Access ho hneď dosadil.

Funkcia *DatePart(interval; dátum [;prvýdeňtýždňa] [;prvýtýždeňroka])* slúži pre zistenie danej časti roka (určuje sa argumentom *interval*). Popis argumentov je obdobný ako pri funkcii *DateDiff*.

Funkcia *DateSerial(rok; mesiac; deň)* umožňuje poskladať (vytvoriť) dátum z jednotlivých častí. Ak by sme chceli zistiť, ktoré knihy sa požičali presne pred dvoma rokmi, mohli by sme použiť v podmienke dotazu výraz: *DateSerial(Year(Date())-2; Month(Date()); Day(Date()))*. Aktuálny dátum rozložíme na rok, mesiac, deň a po zmenšení roku dátum poskladáme. Mohli sme využiť aj funkciu *DateAdd("yyyy";-2;Date())*.

Obdobou funkcie *DateSerial* je funkcia *TimeSerial(hodina, minúta, sekunda)*.

7.2. Agregáčn  funkcie

Tieto funkcie s  využívané veľmi často. Poskytujú s hrnne v sledky typu s čet, po et, priemer, maximum, minimum, štatistick  odchýľky a rozptyly. Pri dotazoch a zostavách sa m žeme často vyhn ť ich pou itiu pou itím s hrnov. Pri formul roch sme si uk zali pou itie agregáčnej funkcie DCount v kapitole 4.2.2.

Na obr. 7.3. je dotaz pre zistenie priemernej doby, ktor  bola ka d  kniha po i an  v po i ky. V dotaze s  pou it  dva sp soby. V druhom st pci je pou it  agregáčn  funkcia Avg – priemer a v treťom rovnak  funkcia, ktor  je ale zabudovan  v ponuke s  tov ch dotazov. Ak by sme chceli vypo itať celkov  dobu po i ania ka dej knihy, zmenili by sme Avg na Sum. Obdobne pomocou funkcie Max by sme zistili maximálnu dobu v po i ky ka dej knihy. Keď je s  asťou v razu agregáčn  funkcia, tak do riadku *Celkom* sa mus  vybrať *Expression*, v niektor ch pr padoch si Access s m uprav  agregáčn  (sumariza n ) dotaz tak mto sp sobom.

| Pole: | Kniha | V raz1: Avg([D tumVr tenia]-[D tumV po i ky]) | V raz2: [D tumVr tenia]-[D tumV po i ky] |
|-----------|-------------------------------------|---|--|
| Tabuľka: | Hist ria | | |
| Celkom: | Group By | Expression | Avg |
| Zoradiť: | | Zostupne | |
| Zobrazit: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Krit ria: | | | |

Obr. 7.3. Priemern  doba po i ania ka dej knihy – dva sp soby

Pre v po et celkovej priemernej doby v po i ky, by sme pou ili oby ajn  v berov  dotaz a iba jeden st pec s v razom *Avg([D tumVr tenia]-[D tumV po i ky])*.

7.3. Matematick  funkcie

Okrem z kladn ch oper cií s tania (+), od tania (-), n sobenia (*) a delenia (/) Access poskytuje aj niektor  matematick  funkcie. Pre umocňovanie pou ijeme symbol strieška ^, teda 2^3 je z pis 2^3 . T  čo ovl daj  troš ka matematiku u  ved  aj odmocniť, napríklad $2^{1/3}$ je z pis $\sqrt[3]{2}$.³⁰ Pre druh  odmocninu existuje funkcia *Sqr( islo)* -  islo mus  byť nez porn , teda *Sqr(9) = 3*.

Funkcia *Log( islo)* vypo ita prirodzen  logaritmus ( islo mus  byť kladn ) a *Exp( islo)* zasa $e^{\text{ islo}}$. Access umozňuje po itať aj goniometrick  funkcie.

Funkcia *Fix( islo)* vr ti cel   asť z  isla. Napríklad: *Fix(3,14) = 3* a *Fix(-3,14) = -3*. Podobn  je funkcia *Int( islo)*, ktor  vr ti v pr pade cel ho  isla to ist  (rovnako ako *Fix*) a v pr pade desatinn ho najbli šie menšie cel   islo. Napríklad: *Int(3,14) = 3* ale *Int(-3,14) = -4*.

7.4. Textov  funkcie

Access poskytuje veľa textov ch funkci , ich vyu itie pre neprogram tora je dost obmedzen .

U  sme spom nali oper tor &, sl u iaci na spojenie textov ch reťazcov. Ďalšia jednoduch  funkcia je *Len(text)*, ktor  vr ti po et znakov, napríklad *Len(„Jaba daba d  2x“)* vr ti v sledok 15 (aj medzery a  isla s  znaky).

³⁰ Ale pozor! T  čo ved  matematiku naozaj iba troš ku, by mohli zapisať t to odmocninu $2^{1/3}$, umocňovanie m  ale prednosť pred delením a tento z pis by znamenal $\frac{2^1}{3}$.

Funkcia $Left(text; n)$ vráti n znakov zľava, podobná je funkcia $Right(text; n)$. Napríklad $Left(„nosorožec“; 3) = „nos“$ alebo príklad pre tvorbu iniciálok:

$Left([Čitateľa]![Meno];1) & „.“ & Left([Čitateľa]![Priezvisko];1) & „.“$

Funkcia $Mid(text; i; n)$ vráti n znakov od i -teho znaku. Napríklad $Mid(„abcdefg“;3; 2) = „cd“$.

$Asc(znak)$ vráti poradie znaku v tabuľke znakov (ASCII) a $Chr(číslo)$ zasa zodpovedajúci znak v tejto tabuľke. $Asc(„a“) = 97$, $Chr(32) = „ “$ (medzera).

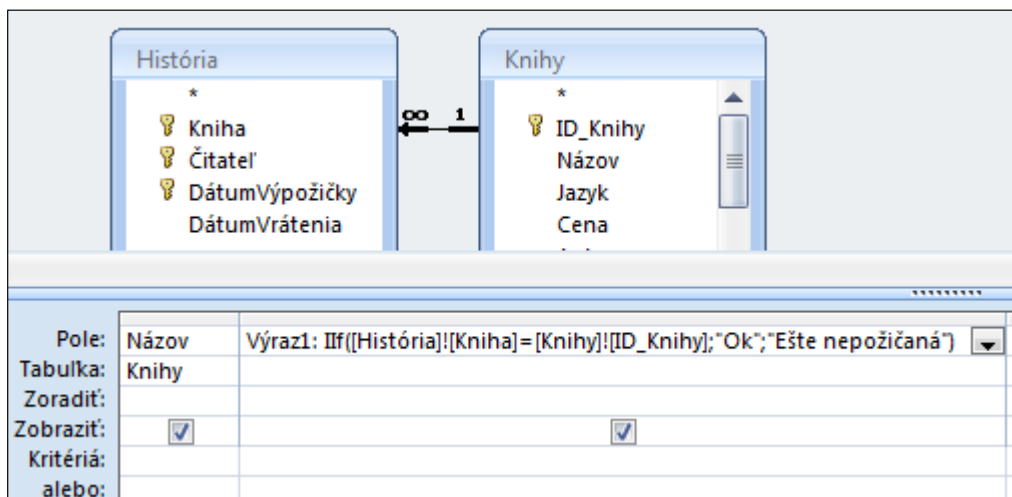
$LCase(text)$ resp. $UCase(text)$ vrátia argument $text$ napísaný malými resp. veľkými písmenami. Samotný reťazec $text$ sa nezmení.

7.5. Ostatné funkcie

Vo zvyšných skupinách je málo (pre nás) zaujímavých funkcií a preto sú popísané v tejto kapitole dokopy bez toho, aby sme ich členili podľa skupín, v ktorých sa nachádzajú.

IIf (logickývýraz; akpravda; aknepravda), táto funkcia (alebo jej variant) je určite jednou z najviac využívaných funkcií vo svete programátorov. A pre svoju jednoduchosť a silu ju využívajú radi aj skúsenejší používatelia. Prvý argument je logický výraz, ak je pravdivý, tak sa vyhodnotí druhý argument inak tretí.

Na obr. 7.4 je dotaz, ktorý ku každej knihe vypíše či už bolo požičaná (presnejšie či sa nachádza v tabuľke *História*). K pochopeniu dotazu si musíme všimnúť typ upravený typ spojenia tabuliek, stretli sme sa s ním už v kapitole 4.3. Pre každú knihu z tabuľky *Knihy* skúmame či sa jej ID nachádza v tabuľke *História*. Ak áno, rovnosť je pravdivá a za *Výraz1* sa dosadí „Ok“, ak sa kniha v tabuľke nenachádza je rovnosť je nepravdivá a dosadí sa tretí argument.



Obr. 7.4. Použitie funkcie IIf

Dotaz v príklade má ešte nedostatok – ak je kniha požičaná prvýkrát a ešte nevrátená, tak sa nachádza v tabuľke *Požičané*. Tento problém rieši použitím logickej spojky OR dotaz z nasledujúceho obrázka.

| | | |
|-----------|-------------------------------------|--|
| Pole: | Názov | Výraz1: IIf([História]![Kniha]=[Knihy]![ID_Knihy] Or [Požičané]![Kniha]=[Knihy]![ID_Knihy];"Ok";"Ešte nepožičaná") |
| Tabuľka: | Knihy | |
| Zoradiť: | | |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | |
| alebo: | | |

Obr. 7.5. Použitie funkcie IIf so zloženou podmienkou

Výraz1 by mohol byť zapísaný bez logickej spojky s dvojnásobným použitím *IIf*:

IIf([História]![Kniha]=[Knihy]![ID_Knihy]; "Ok";

IIf([Požičané]![Kniha]=[Knihy]![ID_Knihy]; "Ok"; "Ešte nepožičaná"))

Teda ak sa *[História]![Kniha]=[Knihy]![ID_Knihy]* tak *Výraz1* bude „Ok“. Ak sa nerovnejú, tak sa vyhodnotí tretí argument. Tretím argumentom je ale ďalší *IIf*. Ten porovnáva v tabuľke *Požičané* a s konečnou platnosťou rozhoduje, či bude *Výraz1* „Ok“ alebo „Ešte nepožičaná“.

Ak by sme chceli zobraziť iba knihy, ktoré majú stanovenú cenu mohli by sme mať malý problém. Niektoré knihy majú cenu určenú, iné nie a niektoré majú cenu nula. Ak by sme dali podmienku >0 , tak by sa nezobrazili knihy s cenou 0. V tomto prípade by sme mohli dať podmienku ≥ 0 ale v mnohých iných prípadoch by sme si nepomohli – napríklad pri poli evidujúcom hospodársky výsledok, ten by mohol byť kladný, záporný, nulový, neuvedený, mohol tam niekto zapísať medzeru, prázdny reťazec.

Na testovanie takýchto možností sa používajú kontrolné funkcie. Najpoužívanejšie sú

- *IsNumeric(výraz)* - je výraz číslo?
- *IsEmpty(výraz)* - je výraz prázdny?
- *IsError(výraz)* - je výraz chyba?
- *IsNull(výraz)* - je výraz Null?

Tieto funkcie testujú svoj argument a vrátia *Pravda (True)* alebo *Nepravda (False)*.

V kapitole 6.2. na obr. 6.6. sme si ukázali ako v zostave rozdeliť hodnoty do intervalov. Pomocou funkcie *Partition(číslo, štart, stop, interval)* to môžeme urobiť aj v dotaze. Na obr. 7.6. vidíme dotaz, ktorý rozdelí knihy z tabuľky *História* do skupín (intervalov) podľa dĺžky výpožičky. V prvom argumente vypočítame dĺžku výpožičky. Ďalšie dva argumenty určujú akú oblasť skúmame – v našom prípade od nula po sto. Posledný argument (5) určuje aké veľké budú jednotlivé skupiny.

| | | |
|-----------|---|-------------------------------------|
| Pole: | Výraz1: Partition([DátumVrátenia]-[DátumVýpožičky];0;100;5) | Kniha |
| Tabuľka: | | História |
| Celkom: | Group By | Count |
| Zoradiť: | Zostupne | |
| Zobraziť: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kritériá: | | |
| alebo: | | |

Obr. 7.6. Použitie funkcie *Partition*

Obr. 7.7. ukazuje výsledok predchádzajúceho dotazu. Vidíme napríklad, že 25 kníh bolo vrátených po 0 až 4 dňoch alebo 46 kníh po 10 až 14 dňoch. Nastavili sme oblasť od 0 po 100 ale jedna kniha bola požičaná

dlhšie. Žiadna kniha nebola požičaná 20 až 100 dní. Sedem kníh má evidentne dátum vrátenia, ktorý je pred dátumom požičania³¹.

| Výraz1 | CountzKniha |
|---------------|-------------|
| 101: | 1 |
| 15: 19 | 2 |
| 10: 14 | 46 |
| 5: 9 | 55 |
| 0: 4 | 25 |
| :-1 | 7 |
| Celkom | 136 |

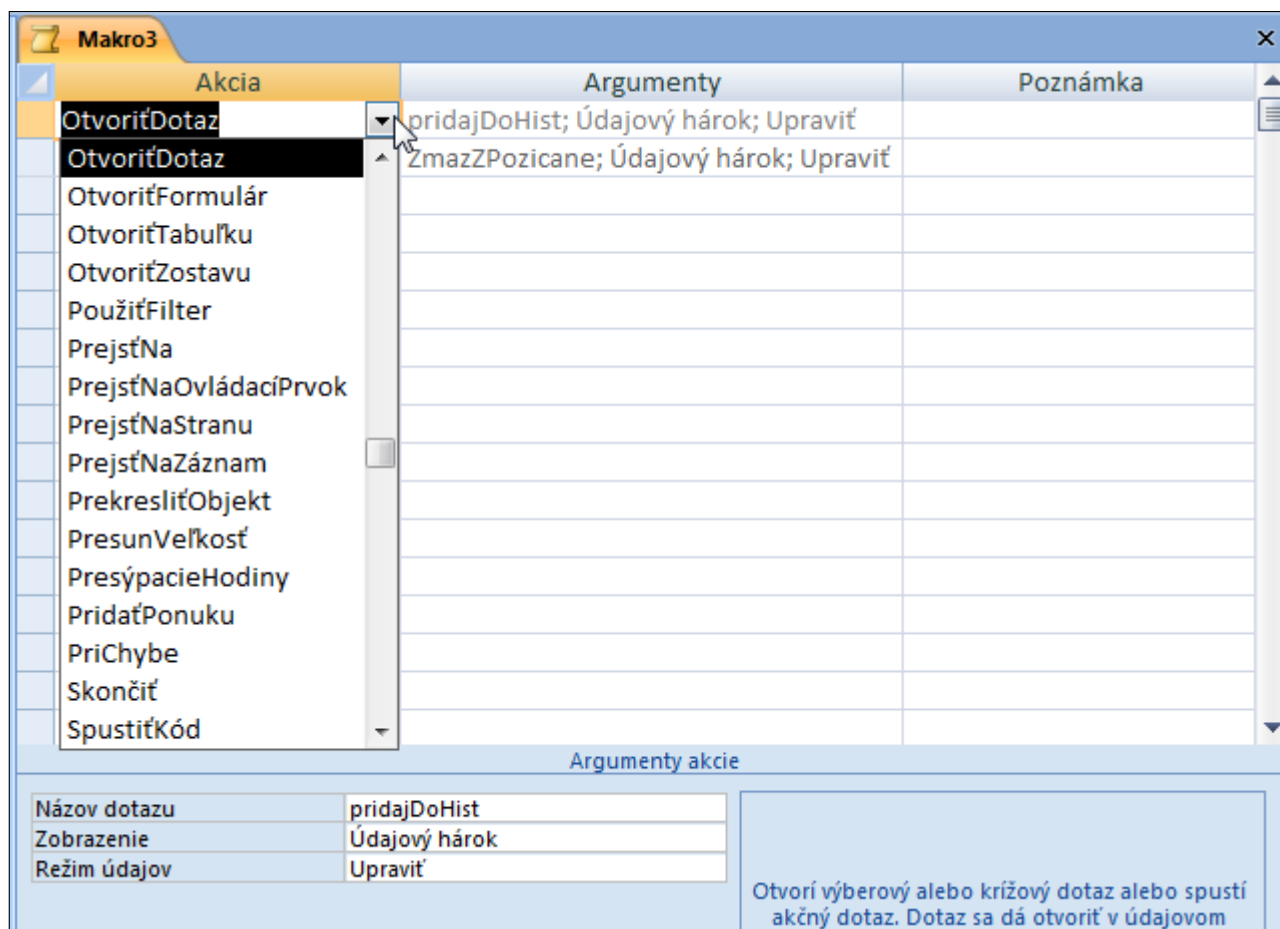
Obr. 7.7. Výsledok z dotazu na obr. 7.6.

³¹ Omyl sa určite stal v časoch, keď sa dátumy nahadzovali ručne a nie makrom funkciou Date().

Zoznam obrázkov a tabuliek

| | |
|---|----|
| Obr. 1.1. Jednoduchá tabuľka | 3 |
| Obr. 1.2 Vzťah 1:N | 8 |
| Obr. 1.3 Vzťah M:N..... | 9 |
| Obr. 1.4 Vzťah 1:1 – Ak by sme každému čitateľovi darovali práve jednu knihu a žiadna nezvýšila | 10 |
| Obr. 1.5. Prepojenie tabuliek v databáze knižnica | 12 |
| Obr. 1.6 Pomenovanie a umiestnenie databázy..... | 13 |
| Obr. 1.7 Pracovná plocha po vytvorení novej databázy | 14 |
| Obr. 1.8 Navigačná tabla..... | 14 |
| Obr. 2.1 Automatické prispôbenie šírky stĺpca..... | 17 |
| Obr. 2.2 Návrhové zobrazenie pri tvorbe tabuľky | 18 |
| Obr. 2.3 Nastavenie niektorých vlastností poľa | 21 |
| Obr. 2.4. Vlastnosti prepojenia tabuliek..... | 25 |
| Obr. 2.5 Zápis povolených hodnôt pri sprievodcovi vyhľadávaním, po výbere možnosti <i>Hodnoty zadá používateľ</i> | 26 |
| Obr. 2.6 Pridanie ďalšej povolenej hodnoty..... | 26 |
| Obr. 2.7. Úprava vzťahov v databáze..... | 28 |
| Obr. 2.8. Návrh na úpravu tabuľky po analýze programom Access | 29 |
| Obr. 2.9. Základy práce s tabuľkou..... | 30 |
| Obr. 3.1 Vytvorenie dotazu po výbere tabuliek | 33 |
| Obr. 3.2 Výber stĺpcov v dotaze..... | 34 |
| Obr. 3.3 Triedenie a zobrazovanie stĺpcov v dotaze | 35 |
| Obr. 3.4 Dotaz filtrujúci knihy s cenou od 10€ po 20€ | 35 |
| Obr. 3.5. Príklad dotazu so zloženou podmienkou..... | 37 |
| Obr. 3.6 Dotaz s parametrom | 38 |
| Obr. 3.7 Vytvorenie nového stĺpca pomocou dotazu | 38 |
| Obr. 3.8. Dotaz pre spojenie dvoch polí do jedného | 39 |
| Obr. 3.9 Dotaz pre zistenie koľko kníh máme od toho ktorého autora | 40 |
| Obr. 3.10 Pozor na zlé použitie podmienky v súčtovom dotaze | 40 |
| Obr. 3.11 Koľko kníh drahších ako 30€ máme od každého autora? | 40 |
| Obr. 3.12 V dotaze je ID_Autora a dotaz zobrazuje meno autora | 41 |
| Obr. 3.13 Dotaz využívajúci viacero tabuliek | 42 |
| Obr. 3.14 Pridávací dotaz..... | 43 |
| Obr. 3.15 Dotaz pre odstránenie kníh s poéziou | 44 |
| Obr. 3.16 Aktualizačný dotaz nastavujúci detektívkam cenu 2€ | 44 |
| Obr. 3.17 Aktualizačný dotaz zvyšujúci cenu detektívkam o 2€ | 44 |
| Obr. 3.18. Krížový dotaz zisťujúci požičiavanie žánrov jednotlivými čitateľmi | 45 |
| Obr. 3.19. Výsledok krížového dotazu zisťujúceho požičiavanie žánrov jednotlivými čitateľmi | 45 |
| Obr. 3.20. Dotaz ku kontingenčnej tabuľke z obr. 3.21. | 46 |
| Obr. 3.21. Kontingenčná tabuľka | 47 |
| Obr. 3.22. Kontingenčná tabuľka so zoskupenými stĺpcami 7 a 9 a podrobnosťami v stĺpci 9 | 47 |
| Obr. 3.23. Kontingenčný graf (upravený) z tabuľky na obr. 3.21. | 48 |
| Obr. 3.24. Hárok vlastností kontingenčného grafu | 49 |
| Obr. 4.1. Ukážka formulára | 50 |
| Obr. 4.2. Formulár, ktorý vznikne po dvoch kliknutiach | 51 |
| Obr. 4.3. Formulár pre zobrazenie čitateľov a ich kníh. | 52 |
| Obr. 4.4. Zmena veľkosti jednotlivých častí formulára | 53 |
| Obr. 4.5. Nastavenie vlastností podformulára | 55 |
| Obr. 4.6. Pridanie tlačidla do formulára..... | 56 |
| Obr. 4.7. Dotaz pre výpočet počtu výpožičiek čitateľa zobrazeného vo formulári <i>Čitatelia</i> | 57 |
| Obr. 4.8. Nastavenie zdroja ovládacieho prvku | 58 |
| Obr. 4.9. Označenie a vlastnosti podformulára..... | 59 |
| Obr. 4.10. Dotaz pre spojenie počtu výskytov čitateľa v tabuľke <i>História</i> s tabuľkou <i>Čitatelia</i> | 60 |
| Obr. 4.11. Aby podformulár fungoval s novým formulárom, musíme upraviť prepojenie..... | 61 |

Obr. 4.12. Ukážka switchboardu..... 62



| | |
|--|----|
| Obr. 5.1. Vytvorenie makra..... | 63 |
| Obr. 5.2. Skupiny makier | 64 |
| Obr. 5.3. Makro testujúce vstup od používateľa | 65 |
| Obr. 5.4. Dotaz na pridanie záznamu z tabuľky <i>Požičané</i> do tabuľky <i>História</i> | 66 |
| Obr. 5.5. Dotaz na zmazanie záznamu z tabuľky <i>Požičané</i> | 66 |
| Obr. 5.6. Zobrazenie odstráneného záznamu | 67 |
| Obr. 5.7. Vypnutie potvrdzovania akčných dotazov | 68 |
| Obr. 6.1. Sprievodca zostavou, tretí krok..... | 70 |
| Obr. 6.2. Možnosti súhrnu | 70 |
| Obr. 6.3. Medzisúčty podľa žánra – priemerná cena. Aj s podrobnosťami..... | 71 |
| Obr. 6.4. Medzisúčty podľa žánra – priemerná cena. Bez podrobností | 71 |
| Obr. 6.5. Zostava – <i>Zobrazenie rozloženia</i> | 72 |
| Obr. 6.6. Zoskupovanie podľa intervalu ceny..... | 73 |
| Obr. 7.1. Porovnanie argumentov „w“ a „ww“ funkcie Datediff pomocou dotazu | 75 |
| Obr. 7.2. Dotaz pre výskum zvislosti požičiavania poézie od mesiaca | 75 |
| Obr. 7.3. Priemerná doba požičania každej knihy – dva spôsoby | 76 |
| Obr. 7.4. Použitie funkcie Iif | 77 |
| Obr. 7.5. Použitie funkcie Iif so zloženou podmienkou..... | 78 |
| Obr. 7.6. Použitie funkcie <i>Partition</i> | 78 |
| Obr. 7.7. Výsledok z dotazu na obr. 7.6..... | 79 |

| | |
|---|----|
| Tab. 1.1. Nenormalizovaná tabuľka | 5 |
| Tab. 1.2. Tabuľka vyhovujúca 1NF | 6 |
| Tab. 1.3. Tabuľka evidencia aktuálne požičaných kníh nevyhovujúca 2NF | 6 |
| Tab. 1.4. Tabuľka nevyhovujúca 3NF | 7 |
| Tab. 1.5. Tabuľka autorov | 7 |
| Tab. 1.6. Tab. 1.4 po úprave spĺňa 3NF | 7 |
| Tab. 3.1. Tabuľka pravdivostných hodnôt And a Or | 36 |