

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA PEDAGOGICKÁ  
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**PRÁCE S DATY V DATABÁZI MS ACCESS**  
BAKALÁŘSKÁ PRÁCE

**Jiří Ráž**

*Přírodovědná studia, obor Informatika se zaměřením na vzdělávání*

Vedoucí práce: Dr. Ing. Jiří Toman

**Plzeň, 2014**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni, 20. března 2014

.....  
vlastnoruční podpis

## OBSAH

1	ÚVOD.....	3
2	DATABÁZE .....	4
2.1	CO JE DATABÁZE? .....	4
2.2	POSTUP PŘI VYTVÁŘENÍ DOBŘE STRUKTUROVANÉ DATABÁZE.....	5
2.2.1	Příprava na papír.....	5
2.2.2	Sběr informací a podkladů.....	5
2.2.3	Tvorba tabulek .....	6
2.2.4	Definování datových typů.....	7
2.2.5	Správné propojení tabulek .....	7
2.2.6	Uživatelské rozhraní.....	8
3	TABULKY .....	9
3.1	VYTVOŘENÍ TABULKY .....	9
3.1.1	Vytvoření nové prázdné tabulky .....	9
3.1.2	Vytvoření tabulky z tabulkového procesoru MS Excel .....	11
3.2	TYPY POLÍ [5].....	11
3.2.1	Text.....	12
3.2.2	Memo .....	12
3.2.3	Číslo .....	12
3.2.4	Datum a čas .....	12
3.2.5	Měna .....	12
3.2.6	Automatické číslo.....	12
3.2.7	Ano/ne.....	13
3.2.8	Objekt OLE .....	13
3.2.9	Hypertextový odkaz .....	13
3.2.10	Příloha .....	13
3.2.11	Průvodce vyhledáváním .....	13
3.3	PRIMÁRNÍ A CIZÍ KLÍČ .....	14
3.3.1	Volba primárního klíče .....	15
3.3.2	Přidání primárního klíče .....	16
3.4	RELACE.....	16
3.4.1	Druhy relací.....	17
3.5	VSTUPNÍ MASKA.....	18
3.5.1	Vstupní maska definovaná, za použití průvodce .....	19
3.5.2	Vložení masky ručně.....	20
3.6	OVĚŘOVACÍ PRAVIDLO .....	21
4	DOTAZY .....	24
4.1	VÝBĚROVÝ.....	24
4.1.1	Vytvoření dotazu.....	24
4.2	KŘÍŽOVÝ .....	26
4.3	AKČNÍ .....	26
4.3.1	Aktualizační.....	27
4.3.2	Vytvářecí .....	27
4.3.3	Přidávací .....	27
4.3.4	Odstraňovací .....	27
4.4	PARAMETRICKÝ .....	28

---

5	FORMULÁŘE.....	30
5.1	VYTVORENÍ FORMULÁŘE.....	30
5.1.1	Vytvoření formuláře za pomoci průvodce.....	30
5.2	OVLÁDACÍ PRVKY FORMULÁŘE.....	32
5.1.2	Nejpoužívanější ovládací prvky [1].....	33
6	SESTAVY.....	34
6.1	VYTVORENÍ SESTAVY.....	34
6.1.1	Vytvoření sestavy s pomocí již vytvořeného objektu.....	34
6.1.2	Vytvoření sestavy pomocí Průvodce sestavou.....	34
7	UKÁZKOVÁ DATABÁZE.....	37
7.1	TABULKY.....	37
7.2	DOTAZY.....	39
7.3	FORMULÁŘE.....	41
7.4	SESTAVY.....	42
8	ZÁVĚR.....	43
	RESUMÉ.....	44
	SEZNAM LITERATURY A DALŠÍCH ZDROJŮ.....	45
	SEZNAM OBRÁZKŮ.....	46
	SEZNAM TABULEK.....	47
	PŘÍLOHY.....	I



## 1 ÚVOD

Cílem této bakalářské práce je vytvořit funkční ukázkovou databázi, v níž předvedu, jak správně vytvářet tabulky, jak do nich ukládat data. Dokáži, že jsou formuláře na ukládání dat přívětivější než přímé vkládání dat do tabulek. Ukáži jakým způsobem dát dohromady dotaz, kterým je možné vybrat specifická data a následně je uložit do nové tabulky.

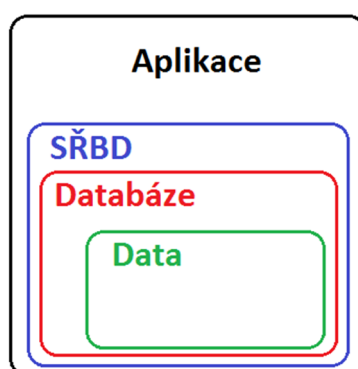
Databáze jsou dnes nedílnou součástí každodenního života. I když si to nemusíme uvědomovat, obklopují nás databáze skoro na každém kroku. Myslíte si, že se mýlím? Rád vás přesvědčím o opaku. Mluvili jste dnes s někým po telefonu? Máte jeho telefonní číslo uložené v mobilním telefonu? Pak je to právě databáze kontaktů, kterou denně využíváme, když potřebujeme volat kolegovi nebo napsat e-mail pro objednání nějaké služby. Dalším místem, kde se nevědomky setkáváte s databázovým systémem, je váš počítač. Souborový systém počítače obsahuje adresáře, podadresáře, soubory a programy, které jsou svojí strukturou databáze.

Tvorba databází může přinášet mnoho pozitivního – přece jen je to velmi efektivní nástroj pro práci s daty, ale pokud databázi špatně navrhne základ, může nám výsledná databáze přinést více starostí než užitku. Proto se hned v druhé kapitole budu věnovat právě správnému návrhu dobře strukturované databáze. Dostanu se od představy, jak má databáze vypadat, až po jednotlivé části, jako jsou tabulky, dotazy, formuláře a sestavy.

## 2 DATABÁZE

### 2.1 CO JE DATABÁZE?

Pod pojmem databáze se rozumí množina záznamů a souborů, které jsou určitým způsobem organizované a uložené. Databázím je hierarchicky nadřazený databázový systém označovaný jako SŘBD – Systém Řízení Bází Dat. Tento systém řeší veškeré operace nad databázemi, aplikace mohou pomocí tohoto systému čerpat, či jinak aktualizovat data v databázích. Princip systému řízeníází dat demonstruje obrázek 2.1. [4]



Obrázek 2.1: Princip řízeníází dat [4]

Programů, které pracují na zmíněném principu, je celá řada. Uvedu zde nejpoužívanější systémy, jsou to například programy Oracle, MySQL, Informix, Ingres, Microsoft SQL Server a Microsoft Access. V této práci se budu zabývat tvorbou databází v posledním zmíněném programu Microsoft Access.

Jako databázi si můžeme představit např. seznam s údaji o zaměstnancích, knižní kartotéku, souborový systém počítače (dokumenty, programy) apod. Zajisté vás napadne, proč používat právě databáze, když to samé přece zvládnou tabulkové procesory (MS Excel, LibreOffice Calc). To ovšem není zcela pravda. Tabulkové procesory nenabízejí tak širokou škálu možností jak pracovat s daty. Navíc databáze umožňuje lepší práci s daty ve smyslu změn, oprav a doplňování. To mohu demonstrovat na příkladu: [3]

Kód výpůjčky	Knihá	Datum vypůjčení	Čtenář	Adresa čtenáře
1	Neviditelný strážce	10.12.2013	Kamil Novák	Jateční 312, Plzeň
2	Inferno	15.12.2013	Petra Holubová	Komorní 11, Plzeň
3	Písečný muž	15.12.2013	Petra Holubová	Komorní 11, Plzeň
4	Eragon	16.12.2013	Ondřej Pecina	Olšová Alej 3, Plzeň
5	Pejsek a kočička	17.12.2013	Dominik Hašek	Komerční 5, Písek

Obrázek 2.2: Příklad kartotéky

Mám kartotéku vypůjčených knih. Kartotéka v MS Excel by vypadala následovně jako na obrázku obr. 2.2.

Pominu-li nutnost přepisovat každému záznamu v tabulce čtenáře a odkud čtenář pochází, nastane problém v okamžiku, když při zadávání údajů do tabulky dojde např. k přepsání, vizte následující obrázek obr. 2.3. Pozornost věnujte výpůjčce číslo 3.

Kód výpůjčky	Kniha	Datum vypůjčení	Čtenář	Adresa čtenáře
1	Neviditelný strážce	10.12.2013	Kamil Novák	Jateční 312, Plzeň
2	Inferno	15.12.2013	Petra Holubová	Komorní 11, Plzeň
3	Písečný muž	15.12.2013	Pwtra Holubová	Komorní 11, Plzeň
4	Eragon	16.12.2013	Ondřej Pecina	Olšová Alej 3, Plzeň
5	Pejsek a kočička	17.12.2013	Dominik Hašek	Komerční 5, Písek

Obrázek 2.3: Přepsání v kartotéce

Kdybych nyní nechal zobrazit vypůjčené knihy Petry Holubové, vyskočil by pouze jeden záznam. Pro počítač to jsou nyní dvě odlišné osoby, které si vypůjčily knihu. Tomuto problému se říká narušení referenční integrity a lze mu předcházet správným použitím databáze, což předvedu v následujících kapitolách.

## 2.2 POSTUP PŘI VYTVÁŘENÍ DOBRĚ STRUKTUROVANÉ DATABÁZE

Pokud jste se rozhodli, že si vyzkoušíte přejít na databázový systém MS Access, pokusím se Vám v následujících řádcích vysvětlit, co je důležité při sestavování nové databáze a na co byste si měli dát pozor.

### 2.2.1 PŘÍPRAVA NA PAPÍR

Než vůbec program MS Access spustíte, měli byste vzít do ruky papír a tužku a připravit si důkladný návrh databáze, neboť se vám může stát, že při nesprávném návrhu databáze budete muset později celou databázi předělávat. [2]

Nejprve je potřeba se zaměřit na určení účelu databáze, k čemu se bude používat a komu bude sloužit. Následně je vhodné si vypsát informace, kdy a jak budou uživatelé databázi používat a jaké jsou jejich nároky. [1]

### 2.2.2 SBĚR INFORMACÍ A PODKLADŮ

Dalším důležitým krokem je shromáždění veškerých podkladů pro naplnění tabulek. To znamená veškerou dokumentaci, jež je spojená s budoucí databází. Pokud budu dělat databázi stavební firmy, vypíši všechny položky potřebné k výkonu činnosti firmy. Vypíši

všechny firmy dodávající materiál, zaměstnance, kteří pro mě pracují, objednávky, které mám splnit, zkrátka vše, co se mé firmy týká. Při přípravě těchto podkladů, nemusí být seznam dokonalý, mnohem důležitější však je vypsát všechny položky, které se mně vybaví. O spolupráci mohu požádat i budoucí uživatele databáze. Jakmile dám dohromady všechny informace, mohu přejít na následující bod. [1]

### 2.2.3 TVORBA TABULEK

Než se podívám na následující část, měl bych vysvětlit několik pojmů vztahujících se k tabulkám. Tabulky se člení na sloupce a řádky. Sloupce se v databázích nazývají pole. Pole je tedy jedna popisovaná vlastnost. Například název společnosti, telefonní číslo, jméno zaměstnance. Každá tabulka obsahuje i řádky, ty se nazývají záznamy. Záznam je tedy celý řádek, který specifikuje jednu konkrétní položku. Například jednoho dodavatele, zaměstnance. [2]

Tabulky jsou základním kamenem databáze. Proto si na jejich vytvoření musím dát opravdu záležet. Veškeré informace při manipulaci s databází se totiž čerpají právě z tabulek.

Nyní projdu všechny sepsané informace a rozdělím je na hlavní témata, podle nichž pojmenuji tabulky. Poté už mohu pod hlavní téma vždy napsat, co bude daná tabulka obsahovat. Je nutno pamatovat na to, že každá tabulka by měla obsahovat jen položky, které odpovídají jednomu tématu.

Proč nelze umístit obsah tabulek dodavatelé, objednávky a zákazníci do jedné tabulky? Jak jsem ukázal v kapitole 2.1, každý řádek v tabulce by obsahoval data o zákazníkovi a jeho objednávce. Protože si zákazník může objednat více výrobků (služeb) od jednoho

<u>Dodavatelé</u>	<u>Zaměstnanci</u>
<ul style="list-style-type: none"> <li>• Název společnosti</li> <li>• Telefonní číslo</li> <li>• Adresa</li> <li>• Město</li> <li>• Stát</li> </ul>	<ul style="list-style-type: none"> <li>• Jméno</li> <li>• Příjmení</li> <li>• Adresa</li> <li>• Platová třída</li> </ul>
<u>Objednávky</u>	<u>Zákazníci</u>
<ul style="list-style-type: none"> <li>• Číslo objednávky</li> <li>• Prodejce</li> <li>• Datum objednávky</li> </ul>	<ul style="list-style-type: none"> <li>• Jméno</li> <li>• Příjmení</li> <li>• Telefonní číslo</li> <li>• Adresa</li> </ul>

Obrázek 2.4: Příklad rozdělení do tabulek [1]

či více dodavatelů, bylo by nutné všechny údaje (jméno, příjmení, ...) častokrát replikovat, a tím bych zapisoval redundantní data, kterými bych zbytečně databázi zvětšoval a zatěžoval. Navíc zde může dojít mnohem snáze k upsání a narušení referenční integrity naší databáze. Proto je mnohem efektivnější, když založím pro hlavní témata vždy speciální tabulku, jež následně propojím mezi sebou.

Uvedu na příkladu: Mám tabulku zaměstnanců. Každý zaměstnanec má svůj vlastní kód. Abych neuchovával informace o platech zaměstnanců v jedné tabulce s jejich běžnými daty, tyto informace uložím do další tabulky, kde bude pouze kód zaměstnance a výše jeho výplaty. Pokud budu chtít přiřadit kódu zaměstnance jméno, mohu tak jednoduše učinit pomocí dotazu a získat jméno z tabulky, v níž je uloženo. Proto je zbytečné stejná data uchovávat ve více tabulkách. [1]

#### **2.2.4 DEFINOVÁNÍ DATOVÝCH TYPŮ**

Již jsem si připravil všechny podklady pro zhotovení tabulek. Nyní musím vzít v potaz, že se každá tabulka skládá z jednotlivých polí. Z tohoto důvodu bych se taky měl věnovat určení datových typů. Proto se rozhodnu, jaká data budu do dílčích polí zapisovat a jak je budu ukládat pro následné zpracování. Jednotlivé datové typy budu rozebírat podrobněji v kapitole 3.2. [2]

Při tvoření databáze je vhodné, aby každá tabulka byla co nejvíce strukturovaná. Každá informace by se měla uvést do databáze v co nejmenším logickém celku. To znamená, že třeba adresu dodavatele neuložím v bloku do jednoho pole, ale údaje rozdělím do tří polí - na ulici, město a PSČ. Umožní mi to v budoucnu mnohem snazší manipulaci s daty, například když budu chtít vypsat dodavatele z určitých měst, nebude to činit sebemenší problém.[2]

#### **2.2.5 SPRÁVNÉ PROPOJENÍ TABULEK**

Jak jsem dříve napsal, důležitým faktorem pro vytvoření kvalitního návrhu databáze je odstranění redundancí. Toho jsem prozatím dosáhl tak, že jsem si veškerá data rozdělil do různých tabulek podle určitých témat. Následně musím rozdělené informace v tabulkách opět spojit. Toho docílím vložením společných polí do souvisejících tabulek. V hlavní tabulce je tomuto poli potřeba nastavit nejprve primární klíč, jinak by propojení tabulek nemohlo být nastavené správně. Problematice primárních a cizích klíčů se ale

budu podrobněji věnovat v kapitole 3.3. Následně je možné definování vzájemných relací mezi tabulkami. [1]

V databázích rozlišujeme tři typy relací. Zde je uvádím pouze v přehledu. Podrobněji se jim budu věnovat v kapitole o relacích.

- relace typu 1:N – v jednom státu se může provozovat mnoho hotelů
- relace typu M:N – hotel si může objednat více zákazníků a zákazník si může objednat více hotelů
- relace typu 1:1 – tato relace není obvyklá – jednomu záznamu v první tabulce odpovídá maximálně jeden záznam v tabulce druhé [1]

#### **2.2.6 UŽIVATELSKÉ ROZHRANÍ**

V neposlední řadě je pro dobře strukturovanou databázi potřebné rozhraní pro vkládání a získávání ucelených údajů a dat z tabulek. K tomuto účelu jsou v databázích formuláře a sestavy. Formulář slouží většinou pro příjemné uživatelské zadávání dat do tabulek a sestavy slouží k vytisknutí potřebných, většinou souhrnných, informací.

Doteď jsem vše připravoval pouze teoreticky. Je načase, abych ukázal, jak takovou databázi vytvořit přímo v programu MS Access.

### 3 TABULKY

Tabulky jsou alfou i omegou celé databáze. Je to primární zdroj informací, ale je nutné si dát pozor na duplicitu dat. Hodnota by měla být vždy uvedena pouze v jediné tabulce. Budu-li potřebovat rozdílná data, která jsou uvedena jinde, není problém v tomto systému data připojit. Mohu tak učinit pomocí dotazu, kterým mohu vybrat konkrétní data přesně dle požadavku. Dotazům se věnuji ve čtvrté kapitole.

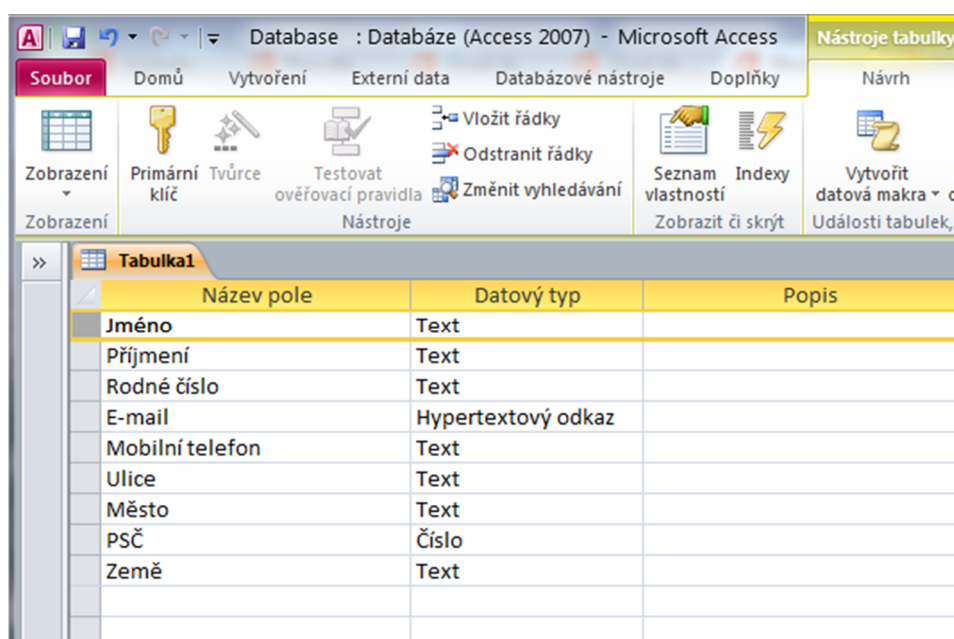
Jelikož jsem si všechny podklady pro vytvoření nové databáze nachystal, otevřu si novou prázdnou databázi, kterou pak postupně naplním daty. Pro následující práci s databází předpokládám alespoň základní orientaci v produktech od společnosti Microsoft minimálně z roku 2007.

#### 3.1 VYTVOŘENÍ TABULKY

Vytvořit tabulku lze několika způsoby. Zde ukáži dva, jež budu používat nejčastěji.

##### 3.1.1 VYTVOŘENÍ NOVÉ PRÁZDNÉ TABULKY

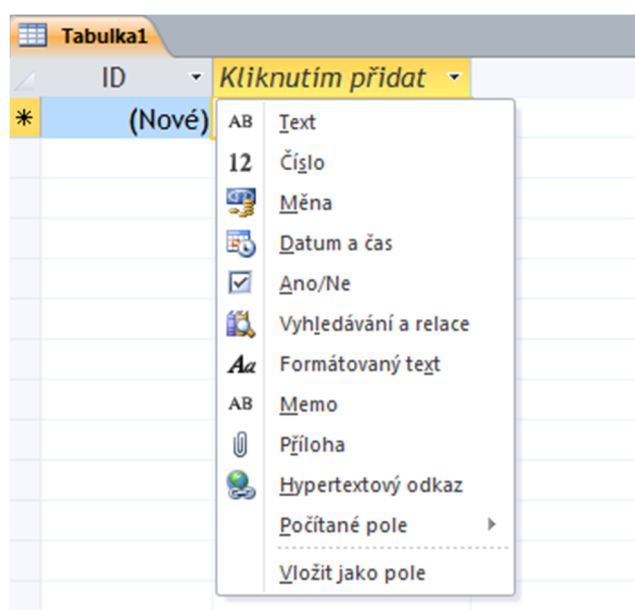
Již jsem vytvořil novou prázdnou databázi, do níž teď chci vytvořit novou tabulku. To provedu tak, že kliknu na kartu Vytvořit a zvolím ve skupině Tabulky položku Návrh tabulky. Otevře se nová prázdná tabulka v režimu návrhu. Uvidím zde Název pole, Datový typ a Popis. Název pole představuje jednotlivé sloupce v tabulce. Pro ukázkovou databázi potřebuji vytvořit tabulku Zákazníci. Bude obsahovat několik sloupců, které budou uchovávat informace jednotlivých zákazníků jako například jméno, příjmení, rodné číslo,



Obrázek 3.5: Připravená tabulka Zákazníci

adresu, atd. Datový typ udává, jaké hodnoty bude daný sloupec nabývat. Já se datovými typy budu zabývat v následující kapitole, ale ilustrativně zde uvedu výčet, jaké datové typy mohou pole nabývat – text, číslo, datum a čas, měna, ano/ne, apod. Popis je volitelná položka, jež umožňuje zapsat například nějakou nápovědu, co do daného pole napsat. Když potom tabulku zobrazím a kurzorem budu v poli, ve kterém mám vyplněný popis, ve stavovém řádku se mi tento popis zobrazí.

Další možností, jak vytvořit prázdnou tabulku je kliknout na tlačítko Tabulka v sekci Tabulky. Tato volba také vytvoří novou prázdnou tabulku s jediným rozdílem, že tabulka se neotevře v návrhovém zobrazení, nýbrž v zobrazení datového listu. Automaticky se v nové tabulce vytvoří pole s názvem ID, které je datového typu automatické číslo, což se většinou hodí pro unikátní pole, které budu označovat primárním klíčem. Další pole, která budu chtít vytvořit, přidám kliknutím na záhlaví nové tabulky. Po kliknutí na záhlaví se objeví seznam datových typů, vybráním jednoho druhu zvolím pro nové pole datový typ. Nabídka je zobrazená na obrázku 3.6.



Obrázek 3.6: Nabídka pro přidání nového pole



### 3.1.2 VYTVORENÍ TABULKY Z TABULKOVÉHO PROCESORU MS EXCEL

Databáze obecně umí importovat velké množství dat, různých formátů i z všelijakých služeb. Tato možnost se bude hodit, zejména pokud využíváte nějaký jiný program pro správu dat a nechce se vám je zdlouhavě přepisovat.

Import dat je poměrně intuitivní, ale může přinášet i různá úskalí. Pro importování připravených dat je potřeba postupovat následujícím způsobem. Na kartě Externí data ve skupině Importovat a propojit, vybrat typ dat pro importování do databáze. Pokud mám tabulku uloženou ve formátu .xlsx, kliknu na položku Excel. Následně se spustí průvodce Načíst externí data – Tabulka aplikace Excel. Vyberu zdroj dat (určím místo, kde se soubor na disku nachází), nechám zaškrtnutou možnost Importovat zdrojová data do nové tabulky v aktuální databázi a stisknu Ok. Pokud první řádek v tabulce obsahuje názvy jednotlivých sloupců, mohu zaškrtnout možnost První řádek obsahuje záhlaví sloupců. V dalším kroku lze upravit názvy jednotlivých sloupců i jejich datový typ. Je potřeba dát pozor na to, aby vkládaná data odpovídala zvolenému datovému typu. Mohlo by to způsobit problémy při převodu do databáze. V následujícím okně se nastavuje primární klíč. To je jedinečné identifikační číslo, které slouží k následnému propojování tabulek. O tento pojem se budu více zajímat v následujících podkapitolách. Nyní mám na výběr z položek Primární klíč přidá aplikace Access, Vlastní primární klíč a Neurčovat primární klíč. První možnost v databázi vytvoří nový sloupec pojmenovaný ID a v něm jsou standardně obsaženy položky od 1 do počtu záznamů, které se nacházejí v tabulce. V importované tabulce již byl sloupec s jedinečným identifikačním číslem, proto jsem zvolil Vlastní primární klíč a v rozbalovacím seznamu vybral jméno záhlaví sloupce. Poslední volba do naší importované tabulky žádný primární klíč nepřidá. Zvolím-li tuto možnost, mohu pak klíč do tabulky přidat ručně. V posledním kroku se průvodce ještě zeptá, do jaké tabulky chci data importovat. Zvolím vhodný název nové tabulky a stisknu tlačítko dokončit.

## 3.2 TYPY POLÍ [5]

Právě jsem ukázal, jak vytvořit tabulku, ať již importovanou z externího souboru nebo zcela novou. Abych mohl tabulku naplnit daty, musím zvolit pro jednotlivá pole vhodné datové typy. To je pro tabulku klíčové.

Uvedu příklad, proč je důležité zvolit správný datový typ. Budu-li požadovat od zákazníků, aby vyplnili rodné číslo, musím se rozhodnout, v jakém formátu jej budu ukládat. Pokud zvolím pole s rodným číslem jako datový typ číslo, nesmí se stát, že se některý ze zákazníků pokusí uložit rodné číslo včetně lomítka. To by totiž nastal datový konflikt a záznam by se neuložil. Pokud bych chtěl ukládat rodné číslo i s lomítkem, musel bych zvolit datový typ text. Ten umožňuje ukládat i čísla. Proto je vhodné si nejprve projít všechna pole, která budu do tabulky zadávat a rozmyslet se, jak budu daná data do tabulky ukládat.

### **3.2.1 TEXT**

Do pole typu text lze ukládat krátké alfanumerické hodnoty s omezenou interpunkční sadou znamének. Maximální hodnota textového pole je 255 znaků. [1]

### **3.2.2 MEMO**

Datový typ memo se používá pro souvislé bloky textu. Memo se uplatní při popisu produktu. Může nabývat alfanumerické hodnoty. Maximální velikost je 1 GB znaků. [1]

### **3.2.3 ČÍSLO**

Tento datový typ zaznamenává číselné údaje. Hodnota pole činí 16 bajtů.

### **3.2.4 DATUM A ČAS**

Do tohoto pole se ukládají hodnoty pro datum a čas. Velikost je 8 bajtů.

### **3.2.5 MĚNA**

Používá se pro uložení peněžní hodnoty s přesností na čtyři desetinná místa. Velikost měny je 8 bajtů.

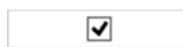
### **3.2.6 AUTOMATICKÉ ČÍSLO**

Datový typ, který generuje jedinečné automatické číslo pro každý záznam, vhodný zejména pro zvolení primárního klíče. Velikost pole je 4 bajty, nebo 16 bajtů – záleží na zvolených vlastnostech v položce Velikost pole.

### **3.2.7 ANO/NE**

Datový typ je definovaný pro ukládání logické hodnoty ano/ne. Access do databáze ukládá jednobitové hodnoty. A to pro logickou hodnotu pravda uloží -1 a pro logickou hodnotu

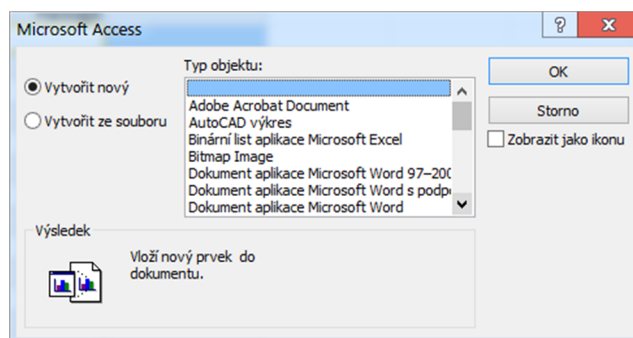
nepravda uloží hodnotu 0. Jak se v tabulce zobrazují hodnoty typu Ano/ne si můžete prohlédnout na obrázku.



Obrázek 3.7: Pravda zobrazená v tabulce

### 3.2.8 OBJEKT OLE

Slouží pro ukládání obrázků, grafů a objektů OLE z mnoha dalších programů. Maximální velikost souboru může být až 2 GB.



Obrázek 3.8: Vložení objektu OLE do databáze

### 3.2.9 HYPERTEXTOVÝ ODKAZ

Tento datový typ umožňuje ukládat různé hypertextové odkazy (cesty i adresy), případně i e-mailové adresy. Celková maximální délka odkazu je až 1 gigabajt znaků.[1]

<\\192.168.1.1\usbs\secured\script.php>

Obrázek 3.9: Příklad uložené cesty v hypertextovém odkazu

### 3.2.10 PŘÍLOHA

Příloha se používá pro vkládání (více) souborů – obrázků, souborů se sešity, různých dokumentů nebo grafů. Do tohoto datového typu je možné nahrát přílohy do velikosti až dva gigabajty. [1]

### 3.2.11 PRŮVODCE VYHLEDÁVÁNÍM

Byť je uveden mezi ostatními datovými typy, nejedná se přímo o datový typ. Výběrem této možnosti se spustí průvodce, který pomůže vytvořit vyhledávací pole. Vyhledávací pole slouží k testování jedné hodnoty v záznamu za využití jiné tabulky nebo dotazu. Velikost pole se váže na datový typ vyhledávacího pole.

### 3.3 PRIMÁRNÍ A CIZÍ KLÍČ

Primární a cizí klíč jsou pro tvorbu tabulek velice důležité. Již jsem zmínil, jak budu nové tabulky rozdělovat podle témat, která mají společná. Teď přijde na řadu postup, jak využít plný potenciál relačních databází. A právě proto je primární a cizí klíč tak podstatný, abych mohl jednotlivé tabulky opětovně propojit a využít tak správně spojených a ucelených dat.

Co to vlastně ten primární klíč je? Jedná se o samostatné pole, které je unikátním identifikátorem pro jednotlivé záznamy v tabulce. Tento identifikátor lze pak použít i v jiné tabulce, s níž budu chtít data sloučit. Když pak tento primární klíč použiju pro propojení s původní tabulkou, označuje se jako klíč cizí případně vedlejší. [1]

Státy			
KódStátu	NázevStátu		Kliknutím přidat
1	Bulharsko	🔑(1)	
2	Itálie	🔑(1)	
3	Madeira	🔑(1)	
4	Řecko	🔑(1)	
5	Španělsko	🔑(1)	

Hotely			
KódHotelu	KódStátu	KódDestinace	Jméno hotelu
1	1	Albena	Hotel Slavuna
2	1	Albena	Torre Ruffa
13	1	Slunečné pobřeží	Hotel Sineva
16	1	Slunečné pobřeží	Hotel Royal Park
41	2	Ischia	Hotel Castello
43	2	Ischia	Hotel Princess
52	2	Kalábrie	Gelina Village
64	2	Sicílie	Hotel Magnolias
72	3	Canico do Baixo	Hotel Promenade

Obrázek 3.10: Propojené tabulky klíčem [1]

Na obrázku 3.10 se nacházejí dvě propojené tabulky. V tabulce Státy mám primární klíč KódStátu (1), jenž je propojený s druhou tabulkou Hotely (2). V tabulce Hotely jsou dva klíče. Pole KódHotelu je pro tuto tabulku primární a pole KódStátu klíč cizí umožňující propojit data právě s tabulkou Státy. Kdybych nyní vytvořil dotaz, kterým bych chtěl vypsát všechny hotely v Bulharsku, nečinilo by to databázi sebemenší problém. [1]

Dotaz1					
KódHotelu	NázevStátu	Jméno hotelu	Počet hvě	Pláž	Stravování
1	Bulharsko	Hotel Slavuna	*****	Písčítá	Polopenze
2	Bulharsko	Torre Ruffa	*****	Písčítá	Polopenze
3	Bulharsko	Laguna Beach	***	Písčítá	Polopenze
4	Bulharsko	Hotel Orchidea	****	Písčítá	ALL INCLUSIVE
5	Bulharsko	Hotel Raluca	***	Písčítá	Polopenze
6	Bulharsko	Hotel Park Vita	****	Písčítá	Polopenze

Obrázek 3.11: Dotaz, který vypsal všechny hotely ve státě Bulharsko

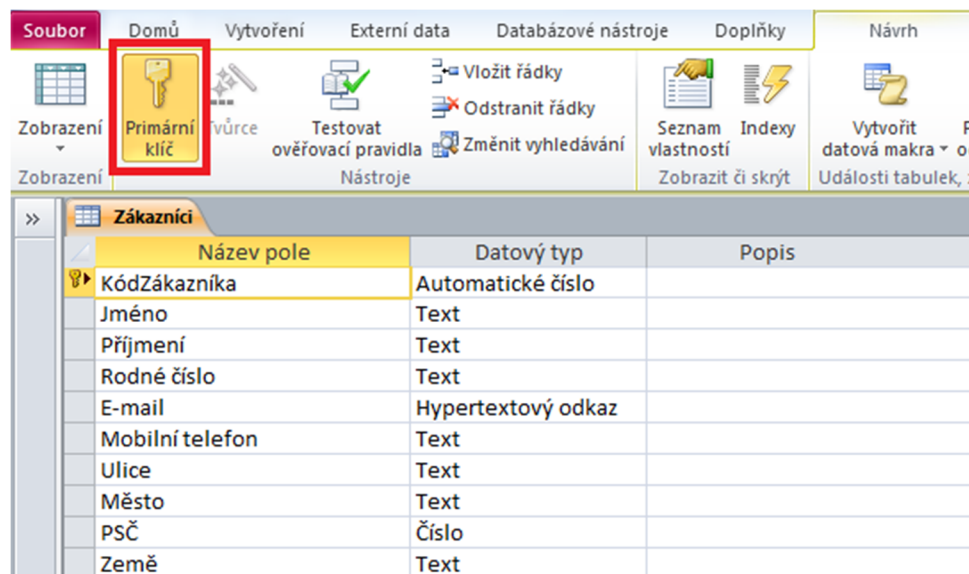
### 3.3.1 VOLBA PRIMÁRNÍHO KLÍČE

Pro pole s primárním klíčem nelze vybrat, co mě zrovna napadne. Toto pole musí být jedinečné pro každý záznam. Pro žádný záznam nesmí mít pole prázdnou hodnotu, v databázích označovanou jako Null, pole by neměla ani nabývat hodnot typu telefonní číslo, e-mail apod. z důvodu možné změny v budoucnu. To by znamenalo změnit všechny navazující záznamy, což není zrovna nic příjemného a může to zkomplikovat chod celé databáze. Adresa, jméno nebo příjmení nejsou vhodná jako primární klíč. Na jedné adrese může sídlit více firem, eventuálně lidí a jména dokonce i příjmení mají skoro vždy mnohočetný výskyt. [1]

Jako primární klíč pro tabulku lidí lze použít rodné číslo. V České republice je toto číslo unikátní pro každou osobu, tím pádem se nestane, že bych narazil na dvě osoby se stejným rodným číslem. V USA lze pro tyto účely použít například číslo sociálního zabezpečení, což je de facto rodné číslo, které se vydává pro pracující lidi. Další možností pro primární klíč je použití kódu. Jen je potřeba dát si záležet na tom, aby kódové označení každého záznamu neslo jinou hodnotu. Asi nejjednodušší variantou je použít pro primární klíč automatické číslo, generující se automaticky při vložení nového záznamu a nedovolující uživateli použít stejné číslo více než jednou.

### 3.3.2 PŘIDÁNÍ PRIMÁRNÍHO KLÍČE

Při vytvoření nové prázdné tabulky vznikne i automaticky nové pole nazvané ID, které má nastavený datový typ na automatické číslo. Toto pole lze následně odstranit, případně upravit v návrhovém zobrazení. [1]



Obrázek 3.12: Hotová tabulka zákazníci

Pokud tabulku vytvářím v návrhovém zobrazení od samého počátku, je nutné vybrat správné pole a označit jej znakem primárního klíče. To předvedu na již vytvořené tabulce Zákazníci, kterou si jistě pamatujete z obrázku 3.5. Do tabulky je potřeba vložit nové pole, což se provede tak, že kliknu pravým tlačítkem do prvního řádku a zvolím možnost vložit řádky. Vznikne nové pole, které nazvu KódZákazníka. Následně daný řádek označím – stačí klepnout buď do názvu pole, nebo do datového typu – a stisknu tlačítko se zlatým klíčem. Tato možnost doplní ikonku malého klíče před název zvoleného pole. Na obrázku 3.12 se podívejte na nově připravenou tabulku, kterou lze začít plnit daty.

Primární klíč lze odstranit tak, že vyberu pole s klíčem a poté kliknu na tlačítko Primární klíč, kterým ikonku klíče odeberu.

### 3.4 RELACE

Nyní vím, jak vytvořit správně tabulku, jak v ní vytvořit pole, které bude sloužit jako primární klíč a připraví ji na propojení dat. Dále ukážu, jak docílit toho, aby se databáze chovala přesně tak, k čemu byl MS Access navržený – jako relační databáze.[1]

Název kapitoly napovídá, že půjde o takzvané relace. Relace jsou vztahy mezi tabulkami porovnávající data v polích, jenž jsou označeny primárním klíčem v tabulce první a cizím klíčem v tabulce druhé. Jinými slovy přidružuje záznamy z jedné tabulky do tabulky druhé. Zpravidla jsou klíčové hodnoty ve vztahem provázaných tabulkách totožné. Toto jsem již ukazoval na obrázku 3.10, kde byla tabulka Státy s primárním klíčem KódStátu a tabulku

Hotely s cizím klíčem KódStátu. Když se nad tím zamyslím, jsou relace zcela jasným vyústěním faktu, že jsem tabulky rozdělil do tematických celků, a aby databáze uměla zpět spojit data jednotlivých tabulek, musím jim nadefinovat vztahy. Vztahy pro propojení tabulek jsou celkem tři – 1:1, 1:N a M:N. [1]

#### 3.4.1 DRUHY RELACÍ

- Relace 1:1

Když je nastavena relace mezi tabulkami 1:1 znamená to, že jeden záznam v první tabulce může odpovídat jen a pouze jednomu záznamu v tabulce druhé. Tohoto vztahu se moc nevyužívá. Snad jen v situacích, kdy není žádoucí, aby tabulka obsahovala moc sloupců, a pro její přehlednost některá pole uchováváme v druhé tabulce. Případně toto spojení mohu použít, když k hlavní tabulce mám doplňkové informace, které jsou platné jen po určitý čas. Poté lze smazat celou doplňkovou tabulku. Pokud chci nastavit toto propojení, nastavuje se primární klíč na obou stranách spojení. [1]

- Relace 1:N

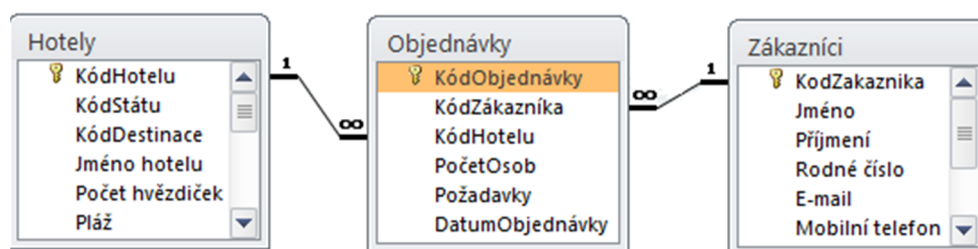
Tento vztah se pro spojování tabulek používá nejčastěji. Je nejobvyklejší z toho důvodu, protože vyhovuje spojení, kdy v první tabulce odpovídá jednomu záznamu několik záznamů z tabulky druhé. Opět se mohu vrátit k obrázku 3.10, kde se nachází tabulka Hotely a Státy. Tyto dvě tabulky jsou propojeny právě relací 1:N. Když se zakoukám do jednotlivých záznamů tabulky, vyčtu z nich, že mnoho hotelů může být postaveno v jediném státě. [1]

- Relace M:N

Relaci M:N si lze představit tak, že v jedné tabulce mnoho záznamů odpovídá i mnoha záznamům v tabulce druhé.

Pokud budete chtít použít mezi tabulkami právě tuto relaci, je nutné nejprve ještě vytvořit propojovací tabulku. Nově vytvořená tabulka bude obsahovat dvě pole. Obě pole mohou být nastavena jako primární klíč – takzvaný složený primární klíč. První hodnota pole bude odpovídat hodnotě pole primárního klíče v první tabulce, druhá hodnota pole bude odpovídat primárnímu klíči v druhé tabulce. Nyní mohu vytvořit i relace mezi těmito tabulkami a to tak, že tabulky budou v relacích 1:N a N:1. Výslednou relaci pak vytvořím

přetažením položek na sebe, například KódHotelu v tabulce Hotely na KódHotelu v tabulce Objednávky. Jak to bude vypadat v databázi, znázorňuje obrázek 3.13.



Obrázek 3.13: Relace M:N

### 3.5 VSTUPNÍ MASKA

Databáze je připravena pro zadávání dat. Vystává však jeden problém, budou-li zapisovat data do databáze jiné osoby než já, budou do ní vkládat data správně? Ve správném formátu? Budou u rodného čísla lomítko psát nebo ho vynechávat? Budou u telefonního čísla psát předvolbu státu? Budou oddělovat číslice po třech znacích? Všechny tyto obavy jsou oprávněné, protože právě stejný formát dat v databázi je důležitý pro zajištění její konzistentnosti, což je důležitý faktor při následném porovnávání a vyhodnocování dat.

Z tohoto důvodu aplikace MS Access umožňuje nastavit u sporných vstupních dat takzvanou vstupní masku. Vstupní maska dovolí uživateli předem nadefinovat pole pro vstupní data. Například v poli s rodným číslem si mohu vynutit lomítko, případně že ukládané telefonní číslo bude moci začínat jen předvolbou pro Českou republiku. Zkrátka vytvořím v poli šablonu, kterou bude muset uživatel zadávající data dodržovat.

Ačkoli se může zdát nastavení vstupní masky komplikované, není tomu tak. Masku lze nastavit dvěma způsoby. První možnost je nechat si pomoci od Průvodce vstupní maskou, případně převzít plnou kontrolu a masku určit ručně. Oba dva postupy ukážu a každý si může vybrat, který je pro něj vhodnější.

#### 3.5.1 VSTUPNÍ MASKA DEFINOVANÁ, ZA POUŽITÍ PRŮVODCE

Než vůbec spustím průvodce, je potřeba zmínit fakt, že vstupní maska lze nastavit jen u určitých datových typů. Těmi jsou text, číslo a datum a čas. Pokud chci masku nastavit pomocí průvodce, lze použít jen datové typy text a datum a čas. Jiné datové typy vstupní masku nepodporují, ale umožňují nastavit ověřovací pravidlo, které ve výsledku splní podobnou funkci. Jen uživatel zadávající data, neuvidí šablonu, podle níž by data vyplnil.



Zadaná data se zkontrolují až při pokusu o uložení. MS Access ověří, že ukládaná data odpovídají ověřovacímu pravidlu, když vyhovují, data uloží. Tímto problémem se ale podrobněji budu zabývat v jiné kapitole.

Průvodce vstupní maskou se spustí tak, že otevřu tabulku v návrhovém zobrazení. Vyberu pole, kterému chci předdefinovat vstupní masku a ve spodní části, jež se věnuje vlastnostem pole, klepnu do buňky Vstupní maska. Na konci řádku se objeví tlačítko se třemi tečkami, kliknutím na tlačítko se spustí Průvodce vstupní maskou.

Otevře se nové okno, ve kterém vidím již předvyplněné masky. Pokud bych chtěl vyzkoušet, jak bude vstupní maska vypadat, až bude uživatel vyplňovat data, navolím požadovanou masku a kliknutím do pole Vyzkoušet se zobrazí zvolená maska. Nyní si můžu zkusit napsat data tak, jak je chci ukládat. Pakliže maska nevyhovuje, lze ji upravit nebo si vytvořit zcela novou. Přizpůsobení provedu vybráním vhodné masky, kterou je potřeba upravit a stiskne se tlačítko Další.

Abych uměl samostatně upravit vstupní masku, je nutné znát, co které klíčové znaky znamenají. To ukazuje následující tabulka.

Symbol	Co znamená
0	Povinná číslice 0 – 9.
9	Nepovinné číslice nebo mezera 0 – 9.
#	Nepovinné číslice 0 – 9, znaky +, - nebo mezera. Nevyplněním se automaticky vypíše mezera.
L	Povinné písmeno.
?	Nepovinné písmeno.
A	Povinný znak. Číslice nebo písmeno.
a	Nepovinný znak. Číslice nebo písmeno.
&	Povinný znak nebo mezera.
C	Nepovinný znak nebo mezera.
., : ; - /	Oddělovače znaků (desetinná čárka, data a času,...). Vybraný znak závisí na lokálním nastavení vašeho operačního systému Windows.
>	Převede následující znaky na velká písmena.
<	Převede následující znaky na malá písmena.
!	Zobrazí vstupní masku zprava doleva – zadávání znaků probíhá stále zleva doprava.

\	Vypíše do pole znaky tak, jak byly bezprostředně zapsány za lomítko.
""	Do pole vypíše znaky tak, jak je uvedeme v uvozovkách.

Tabulka 3.1 – Symboly pro vstupní masku [1]

Já právě upravuji pole Rodné číslo, proto jsem si vybral vstupní masku Rodné číslo. V následujícím okně se zobrazí rozepsaná maska do symbolů. V rozevíracím seznamu mohu navolit, který znak se má zobrazit jako zástupný symbol. Opět mohu vyzkoušet, jak bude pole vypadat při zadávání dat. Pokud vše vyhovuje, přeju tlačítkem Další k poslednímu nastavení. Nyní průvodce dává na výběr ze dvou možností, jak bych chtěl data uložit. První možnost nabízí uchování dat včetně symbolů, které byly použity v masce nebo druhá možnost, že symboly z ukládaných dat odstraní. Zde záleží na použití výsledných sebraných dat, zkrátka co uživateli více vyhovuje. Tlačítkem dokončit mám vstupní masku pro rodné číslo hotové.

### 3.5.2 VLOŽENÍ MASKY RUČNĚ

Vstupní masku lze vytvořit i bez použití průvodce. Jen je potřeba porozumět částem vstupní masky. Celkem má maska tři části. Jedna je povinná a zbylé dvě nejsou nutné vyplňovat. Všechny tři části se oddělují středníkem a mají následující úlohy.

- První úsek definuje, jak bude vstupní maska vypadat – tato část je povinná.
- Druhý úsek může nabývat hodnot 0 nebo 1. Pokud bude hodnota rovna 0, budou se do tabulky ukládat i pomocné symboly obsažené ve vstupní masce. Když bude hodnota pole rovna 1, budou se do databáze ukládat pouze znaky, zadané uživatelem – to znamená bez pomocných symbolů v masce. Když tento úsek nebude vyplněn, doplní se tam automaticky výchozí hodnota 1.
- Třetí část znázorňuje, jaký bude zástupný znak – ten, který uvidíme při zadávání dat.

Výsledná vstupní maska pro rodné číslo může vypadat například takto – 000000\0009;0;\_.

### 3.6 OVĚŘOVACÍ PRAVIDLO

Jak už jsem naznačil v předchozí kapitole, jedná se o obdobnou vlastnost, jako je vstupní maska, pouze umožňuje širší kontrolu nad zadávanými daty. Ověřovací pravidlo mohou použít buď pro samotné ukládané pole, případně pro celý ukládaný záznam. Tato pravidla lze použít jak pro ověření polí tabulek, tak i pro řídicí prvky formulářů.

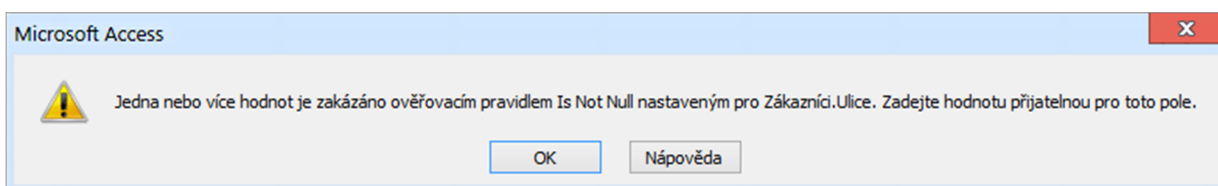
Pokud chci nastavit ověřovací pravidlo pro jednotlivá pole tabulky, přepnu tabulku do návrhového zobrazení, vyberu pole, kterému chci přiřadit ověřovací pravidlo a ve vlastnostech pole mohu začít psát do prázdného pole Ověřovací pravidlo. Pro přehled je zde uvedena tabulka 3.2, kde si lze prohlédnout příklady běžných operátorů ověřovacích pravidel.

Operátor	Funkce
NOT	Ověřuje, jestli jsou hodnoty zadány opačně. Je možné užít před každým operátorem mimo IS NOT NULL.
IN	Ověřuje, jestli jsou hodnoty rovny položkám v seznamu. Porovnávací hodnoty musí být v seznamu oddělené čárkou, seznam je sepsán v závorkách.
BETWEEN	Ověřuje, jestli je uvedená hodnota v rozsahu zadaných hodnot. Minimum a maximum musí být odděleno pomocí výrazu AND.
LIKE	Porovná řetězce se vzorem.
IS NOT NULL	Použitím tohoto výrazu se vynucuje zadání hodnoty. Nežadá-li se hodnota, vyskočí varovná zpráva, lze ošetřit uživatelskou zprávou ve vlastnostech Ověřovací text.
AND	Všechny zadané výrazy oddělené AND musí být pravdivé.
OR	Některé, nikoli však všechny části ověřovacího pravidla musí být pravdivé.
<	je menší než
<=	menší nebo rovno
>	je větší než
>=	větší nebo rovno
=	rovná se
<>	nerovná se

Tabulka 3.2: Přehled operátorů [1]

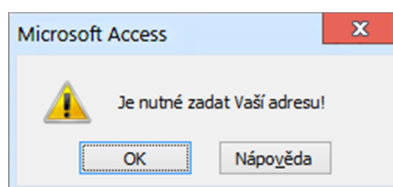
Operátory se používají pro tvorbu výrazu, kterým mohou regulovat zápis dat. Pokud například potřebuji, aby uživatel vyplnil kolonku Adresa, mohu do ověřovacího pravidla vepsat IS NOT NULL – což znamená, že pole nesmí zůstat prázdné. Když se ho uživatel pokusí přeskočit nebo uložit bez vyplnění, zobrazí se mu varovná hláška, jež bude

sdělovat, že zadané hodnoty jsou zakázané. Jak varovná hláška bude vypadat, zobrazuje obrázek 3.14.



Obrázek 3.14: Varovná hláška ověřovacího textu

Tato zpráva může být pro běžné uživatele velice nesrozumitelná, proto existuje další vlastnost pole a tou je Ověřovací text. Aby se uživatel nefrustraval pro něj nerozumnou hláškou, lze do ověřovacího textu napsat přesně taková hláška, aby ji uživatel, který pouze vkládá informace do databáze, porozuměl.



Obrázek 3.15: Ověřovací text

Na obrázku 3.15 je patrné, že tomuto hlášení bude uživatel věnovat více pozornosti a pochopení, než předchozí zprávě.

Aby bylo názornější, jak má vypadat ověřovací pravidlo, předvedu, jak jej zhotovit. Nejprve vyberu sporné pole, do kterého by mohl uživatel zadávat hodnoty špatně. Já vytvořím pravidlo v tabulce Zákazníci pro pole e-mail. Ověřovacím pravidlem překontroluji zadanou hodnotu, zdali zákazníkům e-mail obsahuje všechny náležitosti, to znamená @ a tečku před doménou. Abych zjistil, jestli v zadaném textu je @ musím vepsaný text porovnat, proto se podívám do tabulky 3.2 a zjistím, že pro porovnání řetězců slouží výraz LIKE. Jelikož vypsát všechny e-mailové adresy by byl nadlidský úkon, je možné použít takzvané zástupné znaky. Ty znázorňuje tabulka 3.3.

Znak	Popis
*	Zástupný znak pro libovolný počet jakýchkoliv znaků.
?	Zástupný znak pro libovolný abecední znak.
[ ]	Odpovídá jednomu ze znaků uvedených v závorkách.
!	Odpovídá libovolnému znaku, který není uveden v závorkách.

-	Odpovídá jakémukoli znaku v zadaném rozsahu znaků. Rozsah musí být zadán ve vzestupném pořadí (A až Z, nikoli Z až A).
#	Zástupný znak jakékoli jednotlivé číslice.

Tabulka 3.3: Přehled zástupných znaků [1]

Výsledný výraz by pak mohl vypadat následovně – LIKE "\*@\*.\*".

## 4 DOTAZY

Pokud používáte tabulkový procesor, například MS Excel, jistě dobře znáte možnosti filtrování dat v tabulkách. Dotazy jsou podobné filtrům, jen je možné je více přizpůsobit požadavkům a hlavně se dají uložit a použít kdykoliv znovu, což představuje jednu z jejich výhod. Pokud si vzpomínáte, zmiňoval jsem, že tabulky jsou alfou i omegou celé databáze. Dotazy jsou na tom velice podobně, s tím rozdílem že neslouží pro uchování dat, nýbrž pro komplexní práci s nimi. Pomocí dotazů můžeme data třídit, vytvářet souhrny, nové tabulky, přidávat záznamy, mazat tabulky, zkrátka skoro vše, co se člověku při práci s daty vybaví. Jelikož je tvorba dotazů provázána různými kritérii, budou se hodit tabulky z předchozí kapitoly. [3]

### 4.1 VÝBĚROVÝ

Tento typ dotazu je jeden z nejužívanějších. Pomáhá vybírat data, se kterými právě potřebuji manipulovat nebo kalkulovat. Data mohou být i z různých provázaných tabulek. Umožňuje zobrazit buď všechna zvolená pole s přidruženými záznamy, nebo je schopen připravit pro zvolená data souhrn, průměr i mezisoučty. [3]

Navrhnout výběrový dotaz lze dvěma způsoby. Buď pomocí Průvodce dotazem, nebo tak, že se celý dotaz sestaví podle potřeb pomocí Návrh dotazu na kartě Vytvoření v sekci Dotazy.

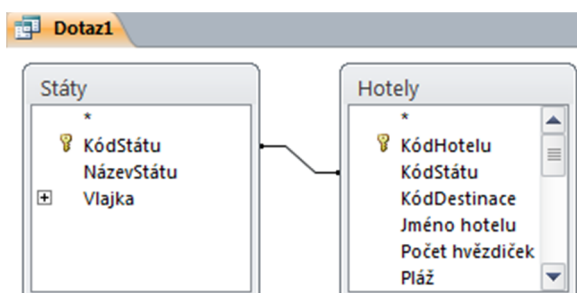
Ukáži teď univerzálnější způsob vytváření dotazů, který bude fungovat i pro další zmíněné dotazy.

#### 4.1.1 VYTVOŘENÍ DOTAZU

Nejprve je potřeba si ujasnit, co budu od vytvořeného dotazu požadovat. Když mám jasnou představu, můžu si udělat pár poznámek na papír, například z jaké tabulky chci data zobrazit, které sloupce budu chtít ve výsledné tabulce vidět a zdali budu chtít data zobrazit v nějakém souhrnu. Toto mně umožní snadněji dotaz sestavit.

Samotný návrh dotazu spustím výběrem Návrhu dotazu na kartě Vytvoření. Nyní se mě program ptá, se kterými tabulkami případně dotazy budu chtít pracovat. Já potřebuji udělat dotaz, jenž mi vypíše z tabulky Hotely počet hotelů v jednotlivých státech. To znamená, že potřebuji zobrazit tabulky Státy a Hotely. Vybrané tabulky se mně

zobrazí včetně názvů polí. A protože již mám nastavené i relace, objeví se v návrhu i nadefinované relace. Vše vypořádá obrázek 4.16.



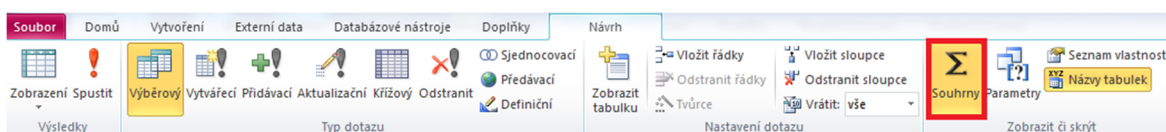
Obrázek 4.16: Zobrazení tabulek při návrhu dotazu

Když mám vybrány všechny tabulky, s nimiž potřebuji pracovat, vyberu požadovaná pole, pro zobrazení ve výsledném uspořádání. Pro moji potřebu vyberu dvojklikem z tabulky Státy položku NázevStátu a z tabulky Hotely KódHotelu. Vyplní se spodní tabulka, což ukazuje obrázek 4.17.

Pole:	NázevStátu	KódStátu
Tabulka:	Státy	Hotely
Řadit:		
Zobrazit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kritéria:		
Nebo:		

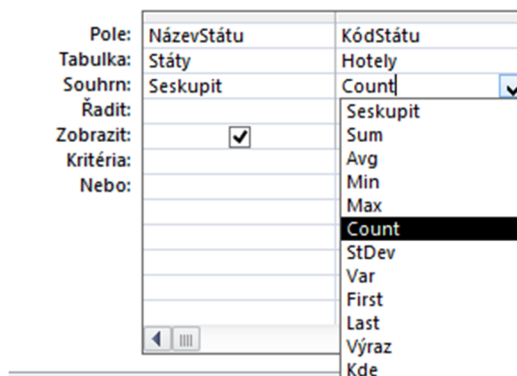
Obrázek 4.17: Návrh dotazu

Jelikož požaduji zobrazení počtu hotelů, musím si ještě nechat zobrazit položku Souhrn. Tu zobrazím kliknutím na ikonku  $\Sigma$  Souhrny v menu návrhu dotazu. Menu ukazuje obrázek 4.18.



Obrázek 4.18: Menu návrhu dotazu

Nyní ve sloupci KódStátu v poli Souhrn vyberu z rozevřacího seznamu položku Count. Jak si můžete všimnout, v seznamu je na výběr opravdu velké množství souhrnných operací, které lze na data aplikovat. Novou položku Souhrn a rozevřací seznam zobrazuje obrázek 4.19



Obrázek 4.19: Rozevírací seznam souhrnu

Dotaz je hotový. Nyní ho můžu uložit, případně před uložením vyzkoušet, jestli splňuje veškeré požadavky zadané před vytvořením dotazu. Dotaz ještě můžu vylepšit tím, že pole přejmenuji na výstižnější název. Do položky Pole napíši chtěný název, za který připiš dvojtečku a původní název pole. Pro ilustraci zde uvedu název pro pole `NázevStátu`, ten by vypadal například takto – `Název státu: NázevStátu`. Tím se výsledný dotaz zpřehlední a bude více srozumitelný pro uživatele, kteří jej nevytvářeli. Spuštěný dotaz si můžete prohlédnout na obrázku 4.20.

Název státu	Počet hotelů
Bulharsko	40
Itálie	29
Madeira	16
Řecko	48
Španělsko	46

Obrázek 4.20: Spuštěný dotaz

## 4.2 KŘÍŽOVÝ

Tento typ dotazů ulehčuje rozbor dat. Většinou se aplikuje, chceme-li získat z tabulek souhrnná data. Křížový dotaz umožňuje nahrazovat záhlaví sloupců za jednotlivé hodnoty, jež mám uložené ve sloupcích tabulky. Použitím tohoto dotazu vznikne takzvaná kontingenční tabulka. Kontingenční tabulka slouží pro přehledné zpracování velkého objemu dat v tabulkách. Platí pro ni podobné vlastnosti jako pro křížový dotaz, proto lze například prohazovat řádky a sloupce i mnoho dalšího.[2][1]

## 4.3 AKČNÍ

Akční dotazy představují komplexní označení pro dotazy, které provádějí nějakou akci s tabulkou. Dělí se do čtyř podskupin a nazývají se podle akce, kterou mají vykonávat.



Pokud budu chtít používat tyto dotazy, musí být databáze uložena v důvěryhodném umístění<sup>1</sup>.

#### 4.3.1 AKTUALIZAČNÍ

Aktualizační dotaz se používá, potřebujeme-li aktualizovat nebo odstranit data v jedné tabulce. Tento dotaz bych mohl například použít, když bych měl tabulku zboží a v ní pole cena. Kdyby například stát změnil daň, případně by se změnila měna, nebylo by rozumné přepisovat každou položku zvlášť, ale mnohem výhodnější by bylo použít aktualizační dotaz, který by změnil všechny požadované hodnoty podle zadání. [1]

Tento dotaz mohu použít, i když vyžadují odstranění některých polí. V tomto případě se hodnota aktualizovaného pole nastaví na hodnotu NULL, čímž se položka odstraní. [1]

#### 4.3.2 VYTVÁŘECÍ

Použitím vytvářecího dotazu vznikne zcela nová tabulka, která musí být založená na datech alespoň z jedné tabulky. Výsledná tabulka reprezentuje pouze sebraná data ze zdrojových tabulek, případně z dotazů. Neobsahuje však relace mezi ostatními tabulkami. Proto je nelze používat stejně jako tabulku, z níž jsme data zkopírovali.[1]

#### 4.3.3 PŘIDÁVACÍ

Pomocí tohoto dotazu je možné zvolit požadované záznamy a následně je rozmnožovat do dalších tabulek. Tento typ dotazu se hodí, pokud například koupím databázi zákazníků a budu ji chtít začlenit do již vytvořené databáze. [1]

Přidávací dotaz se dá zhotovit tak, že nejprve připravím výběrový dotaz, kterým navolím požadovaná data. Pak tento dotaz předělám na dotaz přidávací kliknutím na tlačítko Přidávací v menu Dotazy – nástroje. Toto menu je k vidění na obrázku 4.18.

#### 4.3.4 ODSTRAŇOVACÍ

Odstraňovací dotaz slouží k odstraňování jednotlivých i kompletních záznamů. Kdybych aplikoval tento dotaz na některou z tabulek v databázi bez náležitých úprav, odstranil by dotaz veškeré záznamy v tabulce včetně identifikátoru primárního klíče. Použití

---

<sup>1</sup> „Důvěryhodné umístění souborů je obvykle tvořeno složkou na pevném disku nebo síťovou sdílenou složkou. Všechny soubory, které jsou umístěny do důvěryhodného umístění, lze otevřít bez kontroly Centrem zabezpečení.“ [1]

odstraňovacího dotazu nelze vzít zpět, proto je více než vhodné před spuštěním dotazu zálohovat aktualizovanou tabulku. [1]

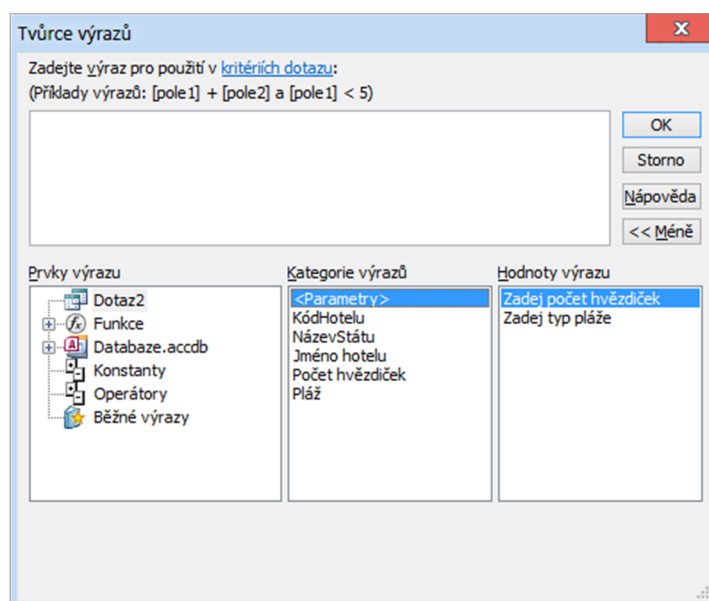
#### 4.4 PARAMETRICKÝ

Parametrické dotazy mohou být v podstatě jakéhokoliv dříve zmíněného druhu. Nejsou tedy zvláštním typem, pouze rozšiřují možnosti výběru a práci s daty. Při jejich spuštění se objeví okno vyžadující po uživateli zadání určitého parametru, kterým potom přizpůsobí výsledná data. To ulehčuje používání stejného dotazu, jemuž při každém jeho spuštění zadám jiný parametr a tím můžu konkretizovat finální požadovaná data.

Efektivní použití parametrického dotazu demonstřuji na následujícím příkladu. Vytvořím nový výběrový dotaz postavený nad tabulkami Státy a Hotely. Relace bude vypadat úplně stejně jako na obrázku 4.16.

Do spodní tabulky zvolím taková pole, která budou vypovídající pro výslednou tabulku. Tímto parametrickým dotazem vyfiltruji konkrétní hotely z celé obsáhlé tabulky. V mém případě jako vstupní parametr poslouží dvě hodnoty. První bude kvalita hotelu ve formě hvězdiček a jako druhý parametr jsem zvolil typ pláže, nacházející se v blízkosti hotelu.

Parametr se zadává tak, že v menu Dotazy – nástroje vyberu tlačítko Parametry, do nichž vyplním srozumitelnou otázku nebo heslo, kterým uživatel rozpozná, co má do parametru zadat. Následně zvolím datový typ, aby odpovídal filtrovanému poli. Poté do spodní tabulky do řádku kritéria vyberu sloupec, který chci vyfiltrovat, a stiskem pravého tlačítka myši navolím v menu Sestavit. Otevře se Tvůrce výrazů, jak vidět je na obrázku 4.21.



Obrázek 4.21: Tvůrce výrazů

V otevřeném okně vyberu v Kategorii výrazů <Parametry> a v Hodnotách výrazu zvolím požadovaný parametr pro aktuálně vybrané pole. Stiskem na tlačítko OK se parametr přidá do spodní tabulky. Jak bude tabulka vypadat si lze všimnout na obrázku 4.22.

Pole:	KódHotelu	NázevStátu	Jméno hotelu	Počet hvězdiček	Pláž
Tabulka:	Hotely	Státy	Hotely	Hotely	Hotely
Řadit:					
Zobrazit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kritéria:				[Zadej počet hvězdiček]	[Zadej typ pláže]
Nebo:					

Obrázek 4.22: Návrh parametrického dotazu

Abych vyzkoušel, jestli dotaz plní to, pro co byl navrhnout, spustím jej. Než se samotná tabulka načte, zeptá se dotaz na právě vytvořené parametry. Okno pro zadání parametru je k vidění na obrázku 4.23.

Obrázek 4.23: Výzva k zadání parametru

Do parametru zadám hodnoty, které bych chtěl nechat zobrazit v tabulce. Proto napíši \*\*\*\*\* pro hotel, a jelikož upřednostňuji písčitou pláž, v druhém parametru zadám písčítá. Dotaz teď projde tabulku a na základě těchto dvou parametrů přizpůsobí výsledná data. Jak lze zpozorovat na obrázku 4.24, ze 179 záznamů mi jich dotaz zobrazil pouze 19, což je pro vyhledání správné položky přehlednější i pohodlnější.

KódHotelu	NázevStátu	Jméno hotelu	Počet hvě	Pláž
1	Bulharsko	Hotel Slavuna	*****	Písčítá
2	Bulharsko	Torre Ruffa	*****	Písčítá
13	Bulharsko	Hotel Sineva	*****	Písčítá
16	Bulharsko	Hotel Royal Park	*****	Písčítá
22	Bulharsko	Hotel Ruffa	*****	Písčítá
30	Bulharsko	Hotel Golden Star	*****	Písčítá
35	Bulharsko	Hotel Lotos	*****	Písčítá
39	Bulharsko	Hotel Arabella	*****	Písčítá
43	Itálie	Hotel Princess	*****	Písčítá
52	Itálie	Gelina Village	*****	Písčítá
64	Itálie	Hotel Magnolias	*****	Písčítá
65	Itálie	Hotel Princes	*****	Písčítá
66	Itálie	Hotel Marina Beach	*****	Písčítá
137	Španělsko	Hotel Mar Ski	*****	Písčítá
160	Španělsko	Hotel Beatriz	*****	Písčítá
167	Španělsko	Hotel Villason	*****	Písčítá
173	Španělsko	Hotel Royal	*****	Písčítá
177	Španělsko	Hotel Confortel	*****	Písčítá
179	Španělsko	Hotel Valentin	*****	Písčítá

Obrázek 4.24: Výběrový dotaz

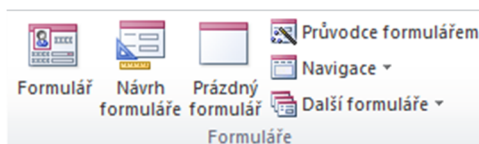
## 5 FORMULÁŘE

Formuláře jsou pro databáze velice mocný nástroj usnadňující uživatelům čtení, ale především také zápis dat do tabulek. Mohu si je představit jako šablonu, která skvěle poslouží pro zadávání i pro získávání dat. Jejich předností je možnost libovolného rozvržení jednotlivých prvků. Dalším faktorem zaručující přehlednost je zobrazování objektů OLE. Ty se ve formulářích dají buď otevřít – například dokument v aplikaci MS Word, nebo se dají soubory přímo zobrazit, jako kupříkladu obrázky. Formulář si lze zkrátka přizpůsobit zcela tak, jak bude pro uživatele, kteří jej budou používat, nejúčinnější. [1]

Formuláře neumožňují pouze přehlednější zápis a čtení dat. Dalším nastavením jednotlivých prvků lze omezit přístup k datům databáze. Pravidla jsou dvě – Zpřístupnit a Uzamknout. Pokud je vlastnost zpřístupnit na hodnotě ano, mohu do ovládacího prvku bez problémů kliknout. Je-li hodnota ne, je prvek zakázaný. Uzamknutím prvku chráním data před zápisem, ale samotná data se v poli zobrazí. [1]

### 5.1 VYTVOŘENÍ FORMULÁŘE

Formulář lze navrhnout několika způsoby, některé umožňují sestavit formulář jedním kliknutím. Takovýto formulář vytvořím tak, že v levém navigačním podokně vyberu tabulku, nad níž chci vytvořit formulář. Poté stisknu na kartě Vytvoření v sekci Formuláře tlačítko Formulář. Tyto úkony vygenerují plně funkční formulář z vybrané tabulky. Pochopitelně ho mohu dále upravovat, přidat si obrázek do záhlaví a mnoho dalšího. Sekce pro vytváření formulářů je k vidění na obrázku 5.25.



Obrázek 5.25: Sekce vytváření formulářů

#### 5.1.1 VYTVOŘENÍ FORMULÁŘE ZA POMOCI PRŮVODCE

Pokud by mi nevyhovovala možnost vytvořit formulář jen z jedné tabulky, mohu použít Průvodce formulářem. Průvodce umožňuje sestavit formulář nad dotazem, tabulkou, případně dovolí použít i více polí z různých provázaných tabulek. Vytvoření formuláře předvedu na příkladu.

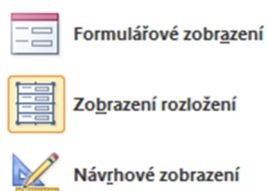
Potřebuji zobrazit formulář, který bude vypisovat kód hotelu, jméno hotelu, v jakém státě se hotel nachází a objednávky pro vybraný hotel, kde bude uvedený kód objednávky, kód zákazníka, příjmení zákazníka a město, odkud zákazník pochází. To znamená, že potřebuji sestavit formulář ze čtyř tabulek.

Spustím tedy Průvodce formulářem a z tabulek vyberu potřebná pole. Až vyberu všechna, která vyžadují ve formuláři, přejdou stiskem tlačítka další na následující nastavení umožňující různé prohlížení dat. Zobrazovat data lze podle jednotlivých zvolených tabulek. Pro můj formulář se hodí zobrazení podle tabulky Hotely. Zvolím tedy možnost Formulář s podformuláři, to znamená, že se mi doplňkové údaje zobrazí v podformuláři pod vybraným záznamem. Hotový formulář může vypadat tak, jak je zobrazen na obrázku 5.26.

KódObjednáv	KódZákaz	Příjmení	Město
7	5	Novák	Kokořín
8	3	Kokos	Kostelec
*	(Nové)		

Obrázek 5.26: Ukázka formuláře

Takto vytvořený formulář mohou i nadále upravovat. Lze měnit barevné schéma, písmo, zobrazit logo firmy nebo také přidat další pole. Pro úpravu formuláře je možné přepínat mezi jednotlivými režimy zobrazení.



Obrázek 5.27: Režim zobrazení

Režimy lze přepínat stisknutím tlačítka zobrazení na kartě Domů nebo na kartě Návrh. Tyto tři režimy umožňují různé činnosti s formuláři.

### Formulářové zobrazení

V tomto režimu je vidět zkompletovaný formulář. Editovat jdou jen záznamy. Formulář je pro editaci zablokovaný.

### Zobrazení rozložení

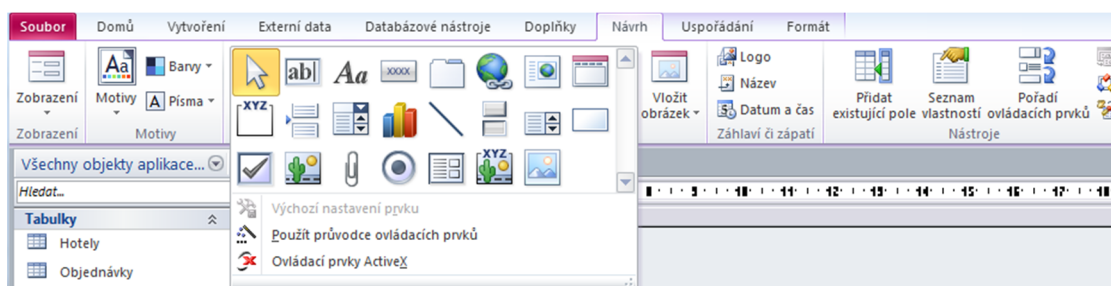
Tímto režimem je aktivní možnost přidávání různých polí a je umožněná omezená editace formuláře. Data z tabulek jsou přístupná.

### Návrhové zobrazení

V návrhovém zobrazení je plně umožněná editace formuláře, data z tabulek nejsou zobrazována.

## 5.2 OVLÁDACÍ PRVKY FORMULÁŘE

Abych mohl pro práci s daty navrhovat dobré formuláře, je potřeba mít alespoň základní přehled o ovládacích prvcích formulářů.



Obrázek 5.28: Nástroje návrhového menu

Většina prvků se standardně používá i v jiných programech, proto nebude problém s jejich užitím i MS Access pro zpřehlednění formulářů. Ovládací prvky se rozdělují na tři skupiny podle provázanosti s databází.

### Vázané

Vázané prvky vycházejí přímo z polí databáze – z tabulek nebo dotazů. Vázaný prvek slouží k zobrazování nebo k ukládání hodnot do polí v databázi. [1]

### Nevázané

Tyto prvky nepoužívají k zobrazování dat pole z tabulek nebo dotazů. Slouží k vypsání údajů, obrázků apod. Tímto prvkem může být například obrázek na pozadí formuláře. [1]

## Počítané

Zdrojem dat pro počítané ovládací prvky nejsou pole v tabulkách nebo dotazech, ale výrazy. Uvedu příklad: V tabulce produkty mám uvedenou cenu s DPH, ale ve formuláři bych rád uvedl i cenu bez DPH. Do počítaného pole proto napíšu výraz, kterým z koncové ceny výrobku spočítám cenu bez DPH. Ta se mi zobrazí na požadovaném místě. [1]

### 5.1.2 NEJPOUŽÍVANĚJŠÍ OVLÁDACÍ PRVKY [1]

Pokud použiji k sestavení formuláře průvodce, budu potřebovat většinou jen drobné úpravy. Proto představím pouze nepoužívanější ovládací prvky v aplikaci MS Access.

#### Textové pole

Běžně užívaný ovládací prvek. Uplatňuje se k zobrazování a editaci dat. Může být použito i pro počítané pole.

Texto

Obrázek 5.29: Ukázka textového pole

#### Popisek

Používá se pro upřesnění zadávaných dat. Může to být nápověda pro uživatele, kteří budou s formulářem pracovat.

popisek

Obrázek 5.30: Ukázka popisku

#### Tlačítko

Jedná se o ovládací prvek, jenž umožňuje spustit nějakou předdefinovanou akci nebo umožňuje vykonat připravené makro.

Tlačítko

Obrázek 5.31: Ukázka tlačítka

#### Pole se seznamem

Dovolí uživatelům zadat vlastní hodnoty do zvoleného seznamu, které lze následně kdykoliv vyvolat.

PoleSeSeznamem1

Obrázek 5.32: Ukázka pole se seznamem

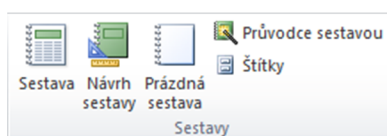
## 6 SESTAVY

V předchozích kapitolách jsem ukázal efektivní práci s daty, jejich filtrování, třídění, zpětné získávání i výběr přesně toho, co zrovna potřebuji. Ale co dělat, když ona data vyžadují vytisknout a uložit do archivu? Právě této záležitosti se budu věnovat v kapitole o sestavách.

Sestavy nabízejí mnoho možností. Lze je využívat pro vytištění seznamu zákazníků, přehledu zboží, faktury pro odběratele. Sestavy umožňují použití podmíněného formátování a zvýraznění položek překračujících povolené hodnoty. Usnadňují vytváření přehledů a trendů, v nichž mohou například seskupovat různé parametry a dosáhnout tím úplných přehledů o své firmě.

### 6.1 VYTVOŘENÍ SESTAVY

Sestavu lze vytvořit pro MS Access typicky několika způsoby. Na kartě Vytvoření v sekci Sestavy se nachází několik možností, kterými je možné vytvořit nová sestava. Sekce Sestavy si lze prohlédnout na obrázku 6.33.



Obrázek 6.33: Sekce vytváření sestavy

#### 6.1.1 VYTVOŘENÍ SESTAVY S POMOCÍ JIŽ VYTVOŘENÉHO OBJEKTU

Asi nejjednodušší varianta pro tvorbu sestavy je tlačítko Sestava. Tato volba vytvoří novou základní sestavu připravenou pro tisk z aktuálně vybrané tabulky, dotazu nebo formuláře. Uvedu příklad: již mám vytvořený parametrický dotaz, kterým dokáži vyfiltrovat hotely stejné kategorie a pláže. Kdybych chtěl tento seznam nyní vytisknout, stačí vybrat uložený dotaz a stisknout tlačítko Sestava. Vygeneruje se přehledná sestava z dotazu, která lze ještě snadno dále upravovat a rozšiřovat o další funkce.

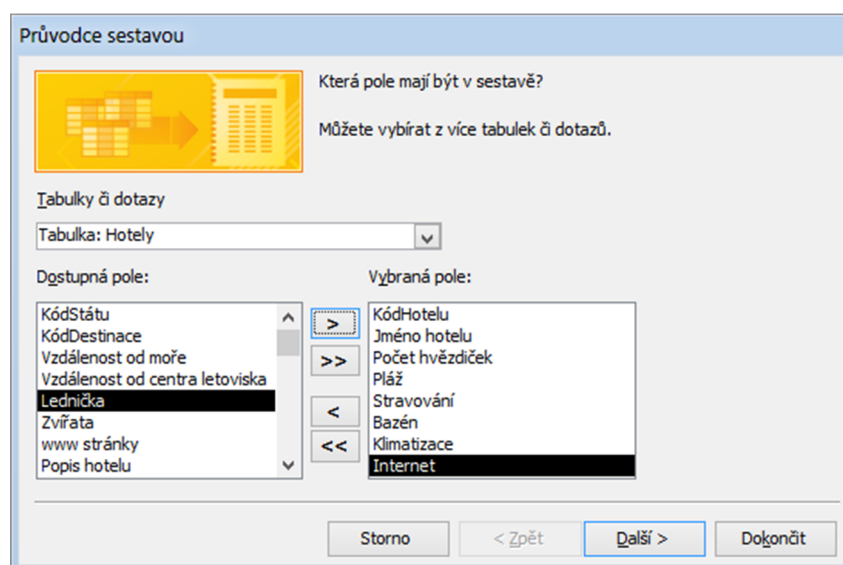
#### 6.1.2 VYTVOŘENÍ SESTAVY POMOCÍ PRŮVODCE SESTAVOU

Sestavu lze vytvořit i pomocí průvodce. Zvolením této možnosti se otevře nové okno, vybízející k výběru polí, která budu chtít zobrazit v nově vytvořené sestavě. Tato pole mohou pochopitelně být z různých zdrojů. Pole lze vybrat z vhodného dotazu, nebo



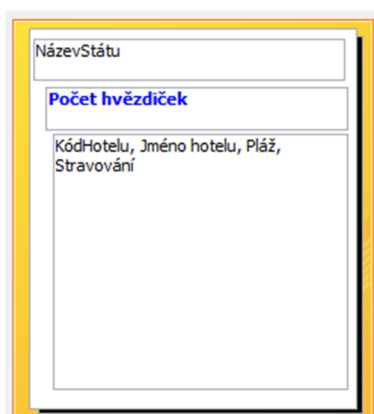
z různých tabulek. To umožní zkompletovat sestavu přesně odpovídající mým požadavkům.

Mohu si například vybrat tabulku Hotely a vytisknout tak komplexní přehled o hotelech. Případně nejprve aplikuji nějaký dotaz, čímž zredukuji výběr hotelů, které právě potřebuji vytisknout. Z této tabulky Hotely mohu navíc navolit jen ta pole mající pro danou sestavu nějaký význam. Na obrázku 6.34 je vidět, jak je možné vybrat pouze některá pole. Volbu provedu pomocí označení položky a stisknutí šipky. Dvojitě šipky přesunou všechny položky.



Obrázek 6.34: Průvodce sestavou

Po výběru vhodných polí musím nastavit prohlížení dat. Já bych rád vytisknul sestavu hotelů seřazených podle států a navíc vypsanych podle počtu hvězdiček. Proto jsem nejprve zvolil prohlížet podle Státy. Následně se průvodce zeptá, jestli nechci přidat další úroveň seskupení. Protože chci hotely řadit navíc podle hvězdiček, zvolím položku Počet hvězdiček a stisknu šipku vpravo. V dalším kroku je možné nastavit řazení polí. Chci, aby data v sestavě byla řazena podle KódHotelu. V seznamu polí vyberu toto pole a dám řadit



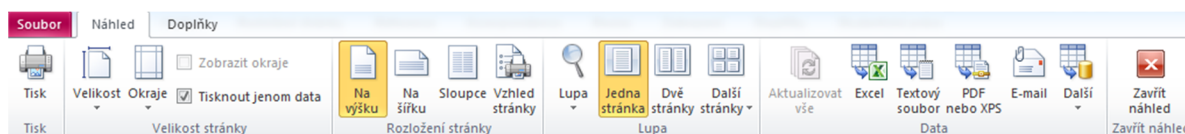
Obrázek 6.35: Úroveň seskupení

vzestupně. V následujícím kroku se nastavuje rozložení sestavy. Já zvolím blokové rozložení. V posledním kroku zvolíme vhodný název a tlačítkem dokončit spustím náhled sestavy.

Hotely podle států TISK				
NázevStátu	Počet hvězdiček	KódHotelu	Jméno hotelu	Pláž
Bulharsko	***	3	Laguna Beach	Písčítá
		5	Hotel Raluca	Písčítá
		9	Rocca Netuno	Písčítá
		17	Hotel Andalusia	Písčítá
		19	Hotel Royal Bay	Písčítá
		21	Hotel Ceasar	Písčítá
		24	Hotel Grifid	Písčítá
Bulharsko	****	40	Hotel Ljuljak	Písčítá
		4	Hotel Orchidea	Písčítá

Obrázek 6.36: Náhled sestavy

Hlavním účelem sestav je jejich tisk, proto nabízejí širokou škálu možností, kterými lze upravit sestava pro finální výstup. Možnosti tisku lze zobrazit výběrem položky Náhled v možnostech zobrazení.



Obrázek 6.37: Menu náhledu sestavy

Výstup nemusí být vždy jen na papír, sestavy se dají uložit i do digitálních podob, jako například do sešitu MS Excel, do PDF souboru nebo rovnou odeslat elektronickou poštou.

## 7 UKÁZKOVÁ DATABÁZE

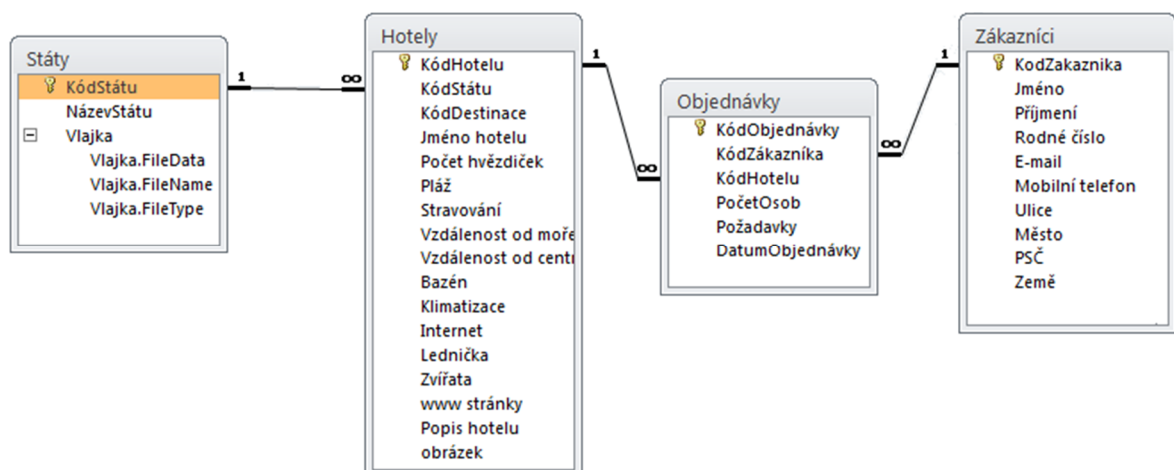
V této části bakalářské práce se budu věnovat praktické ukázce, již jsem pro tyto účely sestavil. Databáze obsahuje několik tabulek s daty. Dále jsem sestavil několik ukázkových dotazů, kterými se snažím jasně demonstrovat jejich využití, funkčnost a nastavení. Vytvořil jsem i několik rozdílných formulářů a sestav.

Tato kapitola bude členěna do jednotlivých podkapitol, v nichž se budu vždy věnovat jednomu tématu.

Rád bych upozornil na jeden častý problém, kterým je nadbytečnost dat. Tento problém je spojený už se samotným návrhem databáze. Jde o pole, jež bude do databáze několikrát opakováno. Pokud navíc databázi bude plnit více lidí, může se stát, že každý bude pole vyplňovat jinak. Uvedu na konkrétním příkladu. Do tabulky Hotely bych mohl rovnou zařadit pole Stát, ale vím, že toto pole se mi bude opakovat mnohokrát. Pro mě je důležité, aby v tomto poli nebyl překlep, nebo stát zapsaný s malým počátečním písmenem. Tím bych při vyhledávání dat nenašel relevantní výsledky. Proto je vhodné vytvořit pro toto pole novou tabulku, propojit ji s tabulkou Hotely a zajistit referenční integritu.

### 7.1 TABULKY

Ukázková databáze obsahuje celkem čtyři tabulky – Hotely, Státy, Zákazníci a Objednávky. Tabulky, včetně jejich polí a relací, jsou zobrazené na obrázku 7.38.



Obrázek 7.38: Tabulky včetně relací

## Datové typy

Snažil jsem se používat takové datové typy, abych jich obsáhl pokud možno co nejvíce a předvedl na příkladu práci s nimi. V každé tabulce jsem aplikoval automatické číslo, používané jako jedinečný identifikátor pro jednotlivé záznamy. Nejčastějším datovým typem je text. V tabulce státy jsem použil datový typ příloha, jež v tomto případě obsahuje vlajky států. Vlajky jsou uloženy obrázky ve formátu png. Vytvořím-li například nový formulář, mohu v něm zobrazovat i tyto obrázky.

V tabulce Hotely je nejvíce polí, tím pádem i nejvíce datových typů. Nejzajímavější typ je Objekt OLE. Tento slouží k uchování nejrůznějších typů dat. Objekt může být buď vložený, to znamená, že se uloží přímo do databáze, nebo může být propojený, tím se do tabulky uloží jen odkaz na daný soubor. To má pochopitelně své výhody i nevýhody – nezvětšuje velikost celkové databáze, ale pokud soubor někdo odstraní, nebude z databáze také přístupný. Platí to i o online zdrojích. Nebude-li k dispozici internet, soubor nemůže být načten. Já tohoto pole využívám pro ukládání obrázků k jednotlivým hotelům ve formátu bmp. Opět lze takto uložené obrázky, případně soubory, zobrazovat ve formulářích nebo sestavách.

Dalším typem je Ano/ne. Lze jej použít pro zobrazení dvou stavů. Je zobrazen jako zaškrtačací políčko, což databázi zpřehledňuje. Jediné úskalí, na které je třeba myslet, je při importu dat z jiných zdrojů. Importovaná tabulka musí obsahovat hodnoty 0 pro nepravdu a hodnotu -1 pro pravdu, jinak se data do databáze správně nenačtou.

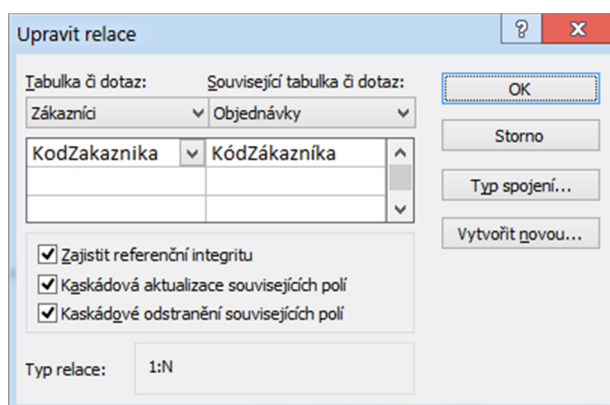
Hypertextový odkaz se uplatní na různé odkazy. Já ho v tabulce Hotely využívám pro ukládání webových odkazů na jednotlivé hotely nebo v tabulce Zákazníci pro uložení odkazu na elektronickou poštu.

Pole s datovým typem Memo v tabulce využívám pro rozsáhlé texty popisující dílčí hotely. Maximálně lze do pole Memo uložit až 2 GB dat.

## Relace

Po volbě primárního klíče je velice důležité správně nastavit relace. Pokud se relace nenastaví nebo se nastaví špatně, není možné využívat ucelenosti dat z více tabulek, což úplně potlačí potřebu používání databází. Uvedu zde příklad špatně nastavených relací.

Tímto problémem je narušení referenční integrity a s tím spojený tak zvaný slepý odkaz. Když vytvářím relace mezi tabulkami, přetáhnu společná pole souvisejících tabulek na sebe, čímž vznikne propojení tabulek. Než však propojení dokončím, zeptá se mi aplikace MS Access ještě na doplňující informace ohledně vznikajícího vztahu. Jsou tam zaškrtačací políčka, která musím zatrhnout pro správnou funkci databáze. Je potřeba, aby mezi propojenými tabulkami byla zajištěna referenční integrita. Kdyby nebyla, mohl bych například v tabulce Objednávky vytvořit novou objednávku, kterou by si mohl objednat neexistující zákazník, a to rozhodně není žádoucí. Zatrhnou-li pole Zajistit referenční integritu, databáze tyto výjimky sama ošetří a použít kód neexistujícího zákazníka mi nedovolí. Aby nevzniknul slepý odkaz, je potřeba při vytváření relace zaškrtnout i zbylá dvě pole – Kaskádová aktualizace souvisejících polí a Kaskádové odstranění souvisejících polí. Tím zajistím, že pokud odstráním záznam například v tabulce Zákazníci, automaticky se odstraní i navazující záznamy v tabulce Objednávky. Kdybych tato dvě pole neoznačil, stalo by se mi, že kdybych odstranil zákazníka, pořád by tomuto zákazníkovi zůstaly vytvořené objednávky.



Obrázek 7.39: Nastavení relace

## 7.2 DOTAZY

Jak jsem již poznamenal ve čtvrté kapitole, dotazy jsou velice mocným nástrojem pro manipulaci s daty. Každá databáze bude obsahovat nějaké dotazy. Je to prostředek usnadňující výběr dat. Velké množství dat potřebuji třídit a vybírat z nich jen ty informace, které jsou pro mě potřebné, proto předvedu v ukázkové databázi několik typických dotazů.

Nejpoužívanějším typem jsou dotazy výběrové. Jak název napovídá, pomáhají vybírat konkrétní data. Ve čtvrté kapitole jsem popsal, jak výběrový dotaz sestavit, proto se těmito dotazy více zabývat nebudu.

Naopak jsem malou část věnoval křížovému dotazu, který pomáhá sumarizovat hodnoty pro konkrétní data. Uvedu na příkladu.

### Křížový dotaz

Potřebuji souhrnně vypsát počet hotelů v každé hvězdičkové kategorii pro všechny státy. Spustím návrh dotazu, vyberu tabulky, z nichž budu čerpat data, a teď si musím říct, jak chci data zobrazit. Chtěl bych, aby jednotlivé státy byly vypsány na řádcích a jednotlivé kategorie hotelu ve sloupcích. Tedy vím, že potřebuji pole *NázevStátu* a *Počet hvězdiček*. Dále potřebuji jedno pole z tabulky *Hotely*, o kterém jsem si jistý, že je určitě vyplněné pro každý záznam. Pro tento účel použiji pole *KódHotelu*. Když mám vybraná všechna pole, změním typ dotazu z výběrového na křížový. Učiním tak kliknutím na tlačítko *Křížový* v menu *Dotazy – nástroje*. Tím se přidá do spodní tabulky nová možnost *Křížová tabulka*. V té pak vyberu, jak chci dané pole zobrazit. Pro pole *KódHotelu* musím vybrat možnost *Souhrn* a v něm položku *Count*, aby výsledná tabulka ukázala souhrnné informace. Jak bude vypadat hotová tabulka v návrhovém zobrazení, ukazuje obrázek 7.40. Křížovou tabulku si pak lze prohlédnout na obrázku 7.41.

Pole:	NázevStátu	Počet hvězdiček	KódHotelu
Tabulka:	Státy	Hotely	Hotely
Souhrn:	Seskupit	Seskupit	Count
Křížová tabulka:	Záhlaví řádku	Záhlaví sloupce	hodnota
Řadit:	vzestupně	vzestupně	
Kritéria:			
Nebo:			

Obrázek 7.40: Křížová tabulka v návrhovém zobrazení

NázevStátu	***	****	*****
Bulharsko	8	24	8
Itálie	10	13	6
Madeira	2	9	5
Řecko	13	28	7
Španělsko	18	20	8

Obrázek 7.41: Křížová tabulka

Pochopitelně lze vytvořit i výběrový dotaz, jenž bude zobrazovat stejné hodnoty jako křížový dotaz, jen nebude tak přehledný. Pro každý stát zobrazí tolik řádků, kolik

je kategorií hotelů, proto je vhodnější použít pro takovéto přehledy křížový dotaz. Výběrový dotaz můžete s křížovým porovnat na obrázku 7.42.

NázevStátu	Počet hvě	počet hotelů
Bulharsko	***	8
Bulharsko	****	24
Bulharsko	*****	8
Itálie	***	10
Itálie	****	13
Itálie	*****	6
Madeira	***	2
Madeira	****	9
Madeira	*****	5
Řecko	***	13
Řecko	****	28
Řecko	*****	7
Španělsko	***	18
Španělsko	****	20
Španělsko	*****	8

Obrázek 7.42: Výběrový dotaz

### Vytvářecí dotaz

Zakčních dotazů předvedu pouze sestavení vytvářecího dotazu. Tento druh dotazu se může hodit při potřebě vytvořit například pomocnou tabulku. Jak jsem dříve zmínil, takto nově vytvořená tabulka nebude obsahovat vazby na tabulky, ze kterých byla vytvořena. Na to je potřeba myslet, budu-li chtít data z této tabulky opět čerpat. Jelikož mohu nově vytvořenou tabulku uložit do jiné existující databáze, používá se tento dotaz převážně pro kopírování a zálohování dat. [1]

Při sestavování vytvářecího dotazu budu postupovat úplně stejně, jako kdybych chtěl vytvořit výběrový dotaz. Vyberu pole k uložení a následně tento výběrový dotaz předělám na dotaz vytvářecí kliknutím na tlačítko Vytvářecí v menu Dotazy – nástroje. Tím je dotaz hotový.


### 7.3 FORMULÁŘE

Formuláře slouží k přehlednějšímu čtení a zápisu dat, proto jsem připravil formuláře pro tabulky nejvíce používané k plnění a prezentování dat. Většina formulářů jde pohodlně připravit pomocí průvodce, nebo ještě jednodušeji pomocí tlačítka Formulář, jež ale zobrazí všechny položky z vybrané tabulky nebo dotazu.

Do ukázkové databáze jsem vytvořil několik formulářů vycházejících přímo z tabulek. Rozdíl v čitelnosti a přehlednosti dat je nejvíce pozorovatelný ve formuláři Hotely. Srovnám-li zobrazený záznam v tabulce Hotely s jedním načteným záznamem ve formuláři, je na první pohled vidět, proč jsou formuláře tak užitečné.

KódStátu	Jméno hotelu	Počet hv	Pláž	Stravování	Vzdálenost	Vzdálenost	Bazén	Klimatizace	Inte
1	Hotel Slavuna	*****	Písčítá	Polopenze	0	1200	<input checked="" type="checkbox"/>	ano	ano
Lednička	Zvířata	www stránky	Popis hotelu		ano		obrázek		
ano	<input checked="" type="checkbox"/>	<a href="http://www.slavuna.bg">www.slavuna.bg</a>	Nový hotel určený pro nejnáročnější klienty leží přímo u pláže.		Bitmap Image				

### Hotely

KódHotelu	<input type="text" value="1"/>
KódStátu	<input type="text" value="1"/>
Jméno hotelu	<input type="text" value="Hotel Slavuna"/>
Počet hvězdiček	<input type="text" value="*****"/>
Pláž	<input type="text" value="Písčítá"/>
Stravování	<input type="text" value="Polopenze"/>
Vzdálenost od moře	<input type="text" value="0"/>
Vzdálenost od centra letoviska	<input type="text" value="1200"/>
Bazén	<input checked="" type="checkbox"/>
Klimatizace	<input type="text" value="ano"/>
Internet	<input type="text" value="ano"/>
Lednička	<input type="text" value="ano"/>
Zvířata	<input checked="" type="checkbox"/>
www stránky	<input type="text" value="www.slavuna.bg"/>
Popis hotelu	<input type="text" value="Nový hotel určený pro nejnáročnější klienty leží přímo u pláže."/>
obrázek	

Obrázek 7.43: Porovnání zobrazení záznamu v tabulce a ve formuláři

## 7.4 SESTAVY

Sestavy, stejně jako formuláře, pomáhají zpřehledňovat výstup dat. V ukázkové databázi jsem vytvořil několik sestav, které mi pomohou zobrazovat přehledná data a informace o zákaznících nebo o jejich objednávkách.



## 8 ZÁVĚR

Hlavním cílem mé bakalářské práce bylo vytvořit obecný postup pro návrh dobře strukturované databáze a zhotovit praktickou ukázkou databáze v programu Microsoft Access. Snažil jsem se co nejlépe popsat, jak samotný teoretický postup, tak i sestavení samotné databáze. Práce je zaměřena pro začátečníky, kteří se chtějí naučit databázi navrhnout a sestavit. Využít ji mohou zajisté i středně pokročilí uživatelé, kteří využijí zejména navržené dotazy a práci s nimi.

V druhé kapitole jsem zjistil, že pro optimální navržení databáze poslouží nejlépe obyčejné věci, jako jsou papír, tužka, řádná příprava a zamyšlení se nad daným tématem. Bude-li čtenář postupovat podle vytvořeného návodu, měl by sestavit kvalitní návrh pro novou databázi, což je stěžejní při tvorbě kvalitní a funkční databáze.

V dalších kapitolách jsem se věnoval základním úkonům s databází – tvorbou tabulek, jak je sestavit a následně naplnit daty; práci s dotazy, filtrováním a tříděním dat. Pojednává se zde, jak efektivněji využívat či ukládat data za pomoci vytvořených formulářů i sestav.

Celá poslední kapitola se zaměřuje na praktickou ukázkou databáze v MS Accessu. Snažil jsem se do této části zahrnout důležité problémy, které se v předchozích částech práce buď přeskakovaly, nebo na ně nebyl prostor. Usiloval jsem o to, aby jednotlivé příklady byly co možná nejsrozumitelnější a zároveň obsahovaly co nejvíce běžně používaných postupů.

Rád bych zde uvedl, že právě proto, že jsou databáze nezbytnou součástí našeho každodenního života, by bylo jim potřeba v hodinách informatiky vyčlenit mnohem více prostoru. Navíc mě udivuje fakt, že na středních školách, až na opravdové výjimky, není práce s databázemi zařazena vůbec. Osobně to považuji za naprostou chybu. Proto jsem rád, že jsem mohl pracovat na tomto tématu a sepsat podpůrný materiál pro lidi, kteří budou mít zájem se naučit pracovat právě s databázemi MS Access.

Věřím, že tato bakalářská práce splňuje všechny zadané požadavky a může být přínosem nejen pro studenty, ale i pro běžné uživatele, kteří se o tuto problematiku budou zajímat.

**RESUMÉ**

The aim of this bachelor thesis is to describe how to create the well-structured database in MS Access. It includes information of making tables, queries, forms and reports which are divided into several chapters. The topics are supplemented by examples to make clear. I think this thesis should be useful and it contributes to students and other users who want to know how to create well-structured database.

Another part of bachelor thesis consists of practical sample database. In that part I focus on major problems of creation the database. In this sample database there are mentioned the most common problems related with creation of databases.

I believe that this thesis satisfies all the specified requirements and could be beneficial not only for students but also for common users, who are interested in this issue.

**SEZNAM LITERATURY A DALŠÍCH ZDROJŮ**

- [1] Domovská stránka podpory. MICROSOFT CORPORATION. Podpora Office: Access [online]. 2007 [cit. 2014-04-06]. Dostupné z: <http://office.microsoft.com/cs-CZ/>
- [2] KRUCZEK, Aleš. Microsoft Office Access 2007: podrobná uživatelská příručka. Vyd. 1. Brno: Computer Press, 2007, 364 s. ISBN 978-802-5116-081.
- [3] PÍSEK, Slavoj. Microsoft Access 2007: podrobný průvodce. 1. vyd. Praha: Grada, 2007, 197 s. Průvodce. ISBN 978-802-4719-672.
- [4] Michal Valenta. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE: Katedra softwarového inženýrství FIT [online]. 2010, 2012/09/24 [cit. 2014-04-06]. Dostupné z: <https://users.fit.cvut.cz/valenta/>
- [5] VIESCAS, John L. Mistrovství v Microsoft Office Access 2003. 1. vyd. Brno: CP Books, 2005, 960 s. ISBN 80-251-0537-7.

## SEZNAM OBRÁZKŮ

Obrázek 2.1: Princip řazení bází dat [4].....	4
Obrázek 2.2: Příklad kartotéky.....	4
Obrázek 2.3: Přepsání v kartotéce.....	5
Obrázek 2.4: Příklad rozdělení do tabulek [1].....	6
Obrázek 3.5: Připravená tabulka Zákazníci.....	9
Obrázek 3.6: Nabídka pro přidání nového pole.....	10
Obrázek 3.7: Pravda zobrazená v tabulce.....	13
Obrázek 3.8: Vložení objektu OLE do databáze.....	13
Obrázek 3.9: Příklad uložené cesty v hypertextovém odkazu.....	13
Obrázek 3.10: Propojené tabulky klíčem [1].....	14
Obrázek 3.11: Dotaz, který vypsal všechny hotely ve státě Bulharsko.....	15
Obrázek 3.12: Hotová tabulka zákazníci.....	16
Obrázek 3.13: Relace M:N.....	18
Obrázek 3.14: Varovná hláška ověřovacího textu.....	22
Obrázek 3.15: Ověřovací text.....	22
Obrázek 4.16: Zobrazení tabulek při návrhu dotazu.....	25
Obrázek 4.17: Návrh dotazu.....	25
Obrázek 4.18: Menu návrhu dotazu.....	25
Obrázek 4.19: Rozevírací seznam souhrnu.....	26
Obrázek 4.20: Spuštěný dotaz.....	26
Obrázek 4.21: Tvůrce výrazů.....	28
Obrázek 4.22: Návrh parametrického dotazu.....	29
Obrázek 4.23: Výzva k zadání parametru.....	29
Obrázek 4.24: Výběrový dotaz.....	29
Obrázek 5.25: Sekce vytváření formulářů.....	30
Obrázek 5.26: Ukázka formuláře.....	31
Obrázek 5.27: Režim zobrazení.....	31
Obrázek 5.28: Nástroje návrhového menu.....	32
Obrázek 5.29: Ukázka textového pole.....	33
Obrázek 5.30: Ukázka popisku.....	33
Obrázek 5.31: Ukázka tlačítka.....	33
Obrázek 5.32: Ukázka pole se seznamem.....	33
Obrázek 6.33: Sekce vytváření sestavy.....	34
Obrázek 6.34: Průvodce sestavou.....	35
Obrázek 6.35: Úroveň seskupení.....	35
Obrázek 6.36: Náhled sestavy.....	36
Obrázek 6.37: Menu náhledu sestavy.....	36
Obrázek 7.38: Tabulky včetně relací.....	37
Obrázek 7.39: Nastavení relace.....	39
Obrázek 7.40: Křížová tabulka v návrhovém zobrazení.....	40
Obrázek 7.41: Křížová tabulka.....	40
Obrázek 7.42: Výběrový dotaz.....	41
Obrázek 7.43: Porovnání zobrazení záznamu v tabulce a ve formuláři.....	42

**SEZNAM TABULEK**

Tabulka 3.1 – Symboly pro vstupní masku [1] .....	20
Tabulka 3.2: Přehled operátorů [1] .....	21
Tabulka 3.3: Přehled zástupných znaků [1] .....	23

## **PŘÍLOHY**

Přiložené DVD obsahuje elektronickou podobu bakalářské práce ve formátu pdf a ukázkovou databázi, která je spustitelná v programu MS Access od verze 2007.