



Vyšší odborná škola obalové techniky
a střední škola, Štětí

Digitální učební materiály

Programové vybavení - Databázový procesor

Ivan Pomykacz



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Licence



Digitální učební materiály, jejímž autorem je Ivan Pomykacz, podléhají licenci [Creative Commons: Uvedte autora - Nevyužívejte dílo komerčně - Zachovejte licenci 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Vytvořeno na základě tohoto díla: <http://dumy.odbornaskola.cz/pomykacz>

Práva nad rámec této licence jsou popsána zde: <http://dumy.odbornaskola.cz/pomykacz>.

Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

<http://dumy.odbornaskola.cz/pomykacz/>.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

Programové vybavení

Databázový procesor

Obsah

- Tabulky

Tabulky

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu			
Název tématické oblasti (sady)	Programové vybavení		
Název materiálu	Tabulky		
Anotace			
Autor		Jazyk	
Očekávaný výstup			
Klíčová slova			
Druh výukového zdroje		Věková skupina žáků	
Typ interakce		Ročník	
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)		Celková velikost	

Návrh tabulky

Obsah

- **Návrhové zobrazení**
 - **Návrhové zobrazení tabulky**
 - **Sloupce tabulky**
 - **Název tabulky**
 - **Datové typy**
 - **Příklad**

Návrhové zobrazení

Jak tabulky, tak dotazy, formuláře a sestavy se začínají tvořit v tzv. návrhovém zobrazení. (Sami si vyzkoušejte, jak funguje průvodce)

Návrhové zobrazení tabulky

Nejproblematictější na celé databázi je návrh toho, jak budou data uložena. Pokud se v návrhu objeví později chyba - když už databáze obsahuje tisíce záznamů, jsou vytvořené dotazy, formuláře a sestavy - může být velmi pracné, návrh změnit.

My začneme prozatím pouze tím, že navrhujeme strukturu jediné tabulky, která zatím nebude mít

žádné vazby do jiných tabulek. Tabulku tvoří její sloupce. Těmi je vymezeno, jaká data bude tabulka obsahovat. Jednotlivé řádky tabulky jsou už pak jen záznamy, které se přidávají dle potřeby.

Sloupce tabulky

Každý sloupec má svůj název a datový typ.

Název tabulky

Název tabulky se obvykle píše malými písmeny anglické abecedy a bez mezer. Chcete-li použít mezeru, nahradte ji podtržítkem "_" Databázový procesor vám ale umožní do názvu tabulky vložit ledacos. Až později zabřehnete do jazyka SQL, možná, že si vzpomenete na první větu tohoto odstavce.

Název tabulky je obvykle psán v množném čísle. Tzn. chceme-li vytvořit tabulku pro ukládání kontaktů, pojmenujeme ji "kontakty".

Datové typy

Datové typy, tak jako v programování zefektivňují ukládání dat, ale také umožňují počítači s nimi vhodně zacházet. Při návrhu databáze je třeba uvažovat ve velkém. Databáze bude patrně fungovat nějaký ten rok, a za tu dobu se může nastřádat hodně dat (třeba i milióny záznamů).

Např. v jednom ze sloupců tabulky budeme ukládat výšku člověka s přesností na centimetry. Jde tedy o číselný (celočíselný) datový typ. Jenomže my si můžeme vybrat mezi TINYINT, SMALLINT, INT a BIGINT. Každý z uvedených typů dokáže uložit různě velké číslo.

- TINYINT 0 .. 255
- SMALLINT -32768 .. 32767
- INT -2147483648 .. 2147483647
- BIGINT -2^{63} .. $2^{63}-1$

Který zvolit? S ohledem na efektivitu místa při ukládání dat ten nejmenší možný. Může to být TINYINT? Kolik měřil nejvyšší člověk na planetě zemi? Číslo 255 stačit nebude, takže druhý přípustný typ je SMALLINT.

Proč se babrat s datovými typy? Nu proto, že datový typ SMALLINT zabere 2B (ať jde o číslo 0 nebo 32000). Kdybychom použili INT nebo BIGINT, tak stejné číslo zabere 4B nebo dokonce 8B u BIGINT.

Kolik žije v ČR lidí? Deset miliónů? Vynásobíme-li deset miliónů krát 2B a převedeme na MB, vyjde nám číslo kolem 19 MB. U BIGINT by toto číslo bylo už 76 MB. A to bereme v potaz pouze jeden sloupec z jedné tabulky. Sloupců a tabulek může být desítky.

Další příklady datový typů:

typ	význam
INTEGER	celá čísla
DOUBLE	desetinná čísla
VARCHAR	alfanumerické znaky

MEMO	alfanumerické znaky (objemný text)
DATETIME	datum a čas

Příklad

Vytvořme tabulku "kontakty" pro ukládání několika informací o našich kontaktech.

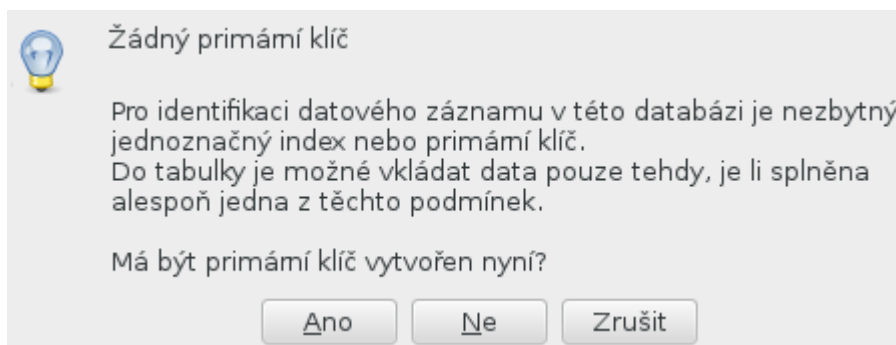
Chceme ukládat data jako:

- jméno
- příjmení
- telefon
- e-mail
- datum narození
- město, ve kterém žije

Pro většinu sloupců zvolíme patrně typ CHAR, resp. VARCHAR. Sloupci s datem narození se pak hodí typ DATE.

Název pole	Typ pole
jmeno	Text [VARCHAR]
prijmeni	Text [VARCHAR]
telefon	Text [VARCHAR]
email	Text [VARCHAR]
narozeni	Date [DATE]
mesto	Text [VARCHAR]

Při uložení tabulky se zobrazí varovný dialog, kde jsme upozorněni na to, že naše tabulka postrádá jednoznačný index nebo primární klíč. Prozatím zvolme možnost vytvoření primárního klíče. Později se k němu vrátíme.



Datové typy

Obsah

- Číselné datové typy
- Alfnumerické typy
- Binární typy
- Datum a čas

Číselné datové typy

Název	Datový typ	Počet bytů	Znaménko	Rozsah
BOOLEAN	pravda / nepravda	1	----	0,1
TINYINT	tiny integer	1	ne	0 to 255
SMALLINT	small integer	2	ano	-2^{15} to $2^{15}-1$
INTEGER	integer	4	ano	-2^{31} to $2^{31}-1$
BIGINT	big integer	8	ano	-2^{63} to $2^{63}-1$
NUMERIC	number	bez limitu	ano	(Max Scale, Max Precision) Max Scale = unlimited Max Precision = $e^{(+/-)2_{31}}$
DECIMAL	decimal	bez limitu	ano	(Max Scale, Max Precision) Max Scale = unlimited Max Precision = $e^{(+/-)2_{31}}$
REAL	real	4	ano	2^{-1074} to $(2-2^{-52}) * 2^{1023}$
FLOAT	float	4	ano	2^{-1074} to $(2-2^{-52}) * 2^{1023}$
DOUBLE	double	4	ano	2^{-1074} to $(2-2^{-52}) * 2^{1023}$

2^{-1074} to $(2-2^{-52}) * 2^{1023}$ could also be stated as $5e^{-324}$ to $1.7976931348623157e^{+308}$

Alfanumerické typy

Název	Datový typ	Maximální délka	Popis
LONGVARCHAR	memo	2GB (32 bit OS)	Uloží až maximální délku nebo počet nastavený uživatelem. Akceptuje jakýkoli UTF-8 znak.
CHAR	text (fix)	2GB (32 bit OS)	Uloží řetězec až do uživatelem definované délky. Při uložení kratšího řetězce, než je definovaná délka dojde k vyplnění mezerami. Akceptuje jakýkoli UTF-8 znak.
VARCHAR	text	2GB (32 bit OS)	Uloží řetězec až do uživatelem definované délky. Žádné výplně.
VARCHAR_IGNORECASE	text	2GB (32 bit OS)	Uloží řetězec až do uživatelem definované délky. Při porovnávání se ignoruje velikost písmen.

Binární typy

Název	Datový typ	Maximální délka	Popis
LONGVARBINARY	image	2GB (32 bit OS)	Uloží jakékoli pole bytů (obrázky, zvuky, atd.). Bez validace.
BINARY	binary (fix)	2GB (32 bit OS)	Uloží jakékoli pole bytů. Bez validace.
VARBINARY	binary	2GB (32 bit OS)	Uloží jakékoli pole bytů . Bez validace.

Datum a čas

Název	Popis	Formát
Date	Uloží rok, měsíc a den.	1/1/99 do 1/1/9999
Time	Uloží čas s přesností na hodinu, minutu a vteřinu	Počet vteřin od 1/1/1970

Timestamp	Uloží časovou značku od datu a času.	
-----------	--------------------------------------	--

Primární klíče

Obsah

- Zběžný úvod do primárních klíčů
 - ID?

Zběžný úvod do primárních klíčů

Definice říká, že primární klíč jednoznačně identifikuje každý záznam (tedy řádek) v tabulce. Už ale není blíže specifikováno, co to ten primární klíč tedy je.

Z pohledu tabulky to může být zvláštní sloupec, ve kterém se budou například jednotlivé řádky číslovat. Pokaždé, když se přidá nový řádek, tak se ve sloupci s primárním klíčem navýší hodnota o 1.

Pokud zvolíme dostatečně objemný datový typ, číslování nám vydrží dlouho. Tinyint se rozhodně nehodí na pozici primárního klíče.

Primárním klíčem může být ovšem i alfanumerický datový typ, tedy třeba i text. Zde ovšem není možné uplatnit automatické navyšování hodnoty, protože není definován způsob jak. Jedinečnost klíče pak musí zajistit ten, kdo do tabulky vkládá data. Pokud by byl vložen záznam se s primárním klíčem, který již byl použit, tak vložení záznamu selže.

Primární klíč může být navíc definován jako kombinace přes několik sloupců.

ID?

Jednotlivé záznamy je třeba umožnit identifikovat z různých důvodů. Jak byste si ukázali na záznam, který chcete smazat? Bez primárního klíče byste museli v dotazu použít jeden nebo spíše více atributů, které by tento záznam jednoznačně identifikovali, aby náhodou nedošlo ke smazání dalších záznamů se stejnými hodnotami (pokud to ovšem není záměr). Např. v tabulce kontakty, kdyby nebyl primární klíč, možná bychom mohli jednoznačně kontakt identifikovat podle příjmení. Jaká je šance, že mezi kontakty budou dvě osoby stejného příjmení? Dobře, můžeme použít kombinaci jména a příjmení? Pravděpodobnost je sice menší, ale stále není nepatrná, resp. nulová.

Procvičování návrhů

Obsah

- Audiotéka

Audiotéka

Vytvořte tabulku pro ukládání audioteky. Tabulka bude kromě primárního klíče obsahovat sloupce:

- název skladby
- délka skladby ve vteřinách
- rok vydání
- interpret
- žánr
- text písně
- album

Rozhodněte o vhodných datových typech.
