

Databáze

LBF/ZUB22 Programové vybavení ordinace zubního lékaře

Mgr. Markéta Trnečková, Ph.D.

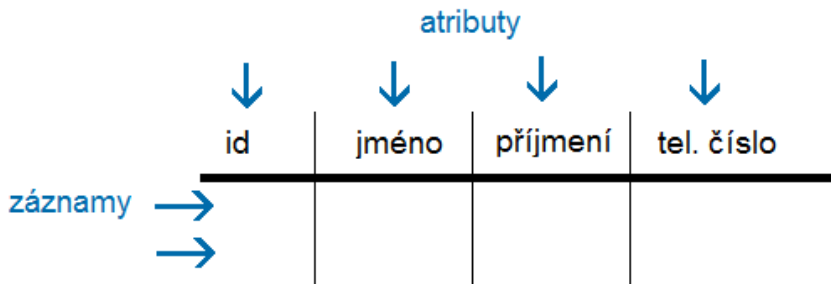
www.marketa-trneckova.cz



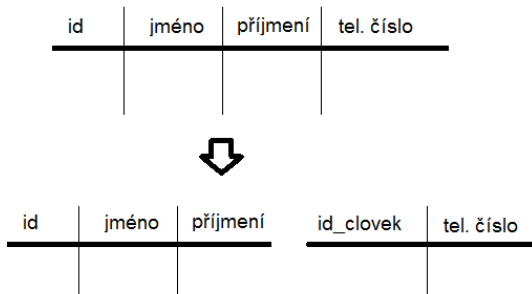
Palacký University, Olomouc

- **databáze** = uložení dat
- dříve členěny hierarchicky, nebo síťově
- **relační databáze**
- **báze dat** = samostatná data
- **DataBase Management System** (DBMS) = systém řízení báze dat - fyzické uložení dat, správa a operace s daty
- **Příklady DBMS:**
 - **mySQL** - webové projekty
 - **MS SQL** - od Microsoftu, drobnější projekty, produkty MS
 - **Oracle** - velké databázové projekty
 - **NoSQL** - není relační databáze
- **SQL** - Simple Query Language

- základ tvoří **tabulky**
- tabulky jsou propojeny předem definovanými vztahy
- **atributy** - sloupce tabulek, určují vlastnosti objektů, které se budou do tabulky vkládat
- **záznamy** - řádky tabulky
- objekty jsou stejného druhu, ale každý jen jednou (integrita dat)
- atributy při návrhu neobsahují konkrétní hodnoty, pouze určují jaké vlastnosti budou objekty mít



- sloupec v tabulce, vlastnost záznamů v tabulce
- **doména** - rozsah hodnot, kterých může nabývat
- **datový typ** - určuje se z domény atributu, např. celé číslo = integer
- DBMS se stará, aby do sloupce nebylo možné zapsat hodnotu, která nevyhovuje datovému typu
- **atomicita atributu** - atribut je atomický, pokud jej již nelze rozložit na dílčí atributy (jméno → jméno, příjmení, titul)
- atribut u jednoho záznamu - jedna hodnota (pokud potřebujeme víc - další tabulka)





Primární klíč

- tabulky nemají fixní pořadí záznamů
- potřeba identifikovat konkrétní záznam
- jeden atribut **unikátní** = primární klíč
- ! každý záznam musí mít jedinečnou hodnotu v rámci celé tabulky !
- každá tabulka **musí** mít primární klíč

Cizí klíč

- tabulky, které jsou propojeny vztahy musí být provázány
- provázání = vytvoření speciálního atributu (**cizí klíč**) v jedné z tabulek
- hodnota atributu = hodnota cizího klíče



- **Integer** - celé číslo
- **Varchar** - krátký text (do 255 znaků)
- **Text** - dlouhý text
- **Boolean** - logická hodnota (ano/ne, true/false,...)
- **Date/DateTime** - datum, případně datum a čas



- tabulky mají mezi sebou vztahy - návrh databáze
- např. v databázi knihovny - tabulky kniha, spisovatel
- v návrhu je nutné vztah popsat (spisovatel napsal knihu, multiplicita, ...)
- návrh fyzického vytvoření

- určuje kolik záznamů v první tabulce může být spojeno s jedním záznamem v tabulce druhé a naopak
- např. Kolik knih mohl autor napsat, kolik autorů může kniha mít

propojení 1:n

- jednoduché
- např. fotbalový klub (1 klub může mít více hráčů, každý hráč patří do 1 klubu)
- v tabulce s násobnou vazbou (hráči) - nový atribut (cizí klíč) identifikující k jakému klubu patří

propojení m:n

- např. kniha, autor
- tento vztah nelze jednoduše zapsat do tabulky - potřeba rozložit
- vytvoříme novou tabulku obsahující jen cizí klíče z obou tabulek (identifikace knihy a identifikace autora)

propojení 1:1

- málo časté
- např. zaměstnanec, služební telefon
- je jedno ve které tabulce je cizí klíč

- zefektivnění databáze - snížení velikosti, menší náchylnost k chybám
- několik stupňů zefektivnění - normálové formy (NF)

1. NF

- všechny atributy **atomické** = obsahují nedělitelné hodnoty
- např. adresa → ulice, číslo, město, psč
- **důvod**: neatomický atribut = omezená práce s tabulkou (např. vyhledávání lidí ze stejného města)

2. NF

- všechny atributy **závislé na primárním klíči**
- špatný příklad: tabulka student (id, jméno, příjmení, univerzita, adresa univerzity)
- adresa je nezávislá na id studenta
- chyba by mohla vzniknout při změně adresy univerzity (změnu je nutno provést u více studentů)
- **řešení**: nová tabulka s informacemi o univerzitě, student bude mít cizí klíč nastaven do této tabulky



3. NF

- všechny atributy **závislé přímo na primárním klíči**
- např. zaměstnanec, jeho pozice a plat
- pokud je plat odvozen od pozice, pak je závislý na tomto atributu ne na primárním klíči
- samostatná tabulka s ohodnocením jednotlivých pozic



- dobře navržená databáze = DBMS může zajistit mezi propojenými tabulkami **referenční integritu**
- např. neumožní odstranit autora, který je uveden jako autor nějaké knihy
- neumožní ke knize přiřadit autora, který neexistuje v tabulce autorů
- neumožní nám změnit primární klíč, který je jako cizí klíč v jiné tabulce



- atributy: A_1, A_2, \dots, A_n
- relační schéma: $R = (A_1, A_2, \dots, A_n)$
- např. kniha = (id, název, autor, počet stran)
- relace nad relačním schématem: $r(R)$
- např. název(kniha)
- **instance relace**: skutečné hodnoty - výčtem (tabulkou)
- prvek relace: n -tice (řádek v tabulce)
- např. 1 "Romeo a Julie" "William Shakespeare" 151



Selekce

- výběr řádků
- většinou se zadává podmínka → výběr řádků splňující podmínku
- `select * from kniha where početStran > 100`
- podmínky je možné spojovat logickými operacemi - or, and, ...

Projekce

- výběr sloupců, které chceme zobrazit
- `select autor from kniha`

Rozdíl

- $r - s$
- relace musí být **kompatibilní** (stejný počet atributů, domény atributů musí být po řadě shodné = stejný datový typ)

Table : r

A	B
α	1
α	2
β	1

Table : s

A	B
α	2
β	3

Table : $r - s$

A	B
α	1
β	1

Sjednocení

- $r \cup s$
- relace musí být **kompatibilní**

Table : r

A	B
α	1
α	2
β	1

Table : s

A	B
α	2
β	3

Table : $r \cup s$

A	B
α	1
α	2
β	1
β	3

Operace mezi tabulkami

Kartézský součin

- $r \times s$
- předpokládáme, že relace mají různé atributy
- pokud nějaké atributy mají stejné jméno - přejmenování
- sjednocení všech řádků r se všemi řádky s

Table : r

A	B
α	1
β	2

Table : s

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

Table : $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Další operace

Přejmenování

- není klasická operace
- umožňuje nově pojmenovat výsledek jiné operace

Skládání operací

- složitější operace = složení základních operací
- např. kartézský součin $r \times s$ a výběr takových řádků, kde $A = C$

Table : r

A	B
α	1
β	2

Table : s

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

- z praktických důvodů se zavádějí další odvozené operace - zjednodušení

Průnik

- $r \cap s$
- relace musí být kompatibilní
- vrací řádky, které jsou v obou relacích

Table : r

A	B
α	1
α	2
β	1

Table : s

A	B
α	2
β	3

Table : $r \cap s$

A	B
α	2

Přirozené spojení

- $r \bowtie s$
- relace r a s mají některé atributy společné
- výsledek je spojení takové, kde shodné sloupce jsou si rovny (hodnoty v těchto sloupcích)

Table : r

A	B
α	1
β	2

Table : s

A	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

Table : $r \bowtie s$

A	B	D	E
α	1	10	a
β	2	10	a
β	2	20	b



- $r \div s$
- určeno pro dotazy obsahující frázi "pro všechny"
- schéma: $r = (A_1, \dots, A_n, B_1, \dots, B_m)$, $s = (B_1, \dots, B_m)$
- výsledek má schéma: $r \div s = (A_1, \dots, A_n)$
- $r \div s = \{t \mid t \in \Pi_{R-S}(r) \wedge \forall u \in S : tu \in r\}$

Table : r

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

Table : s

D	E
a	1
b	1

Table : $r \bowtie s$

A	B	C
α	a	γ
γ	a	γ

Table : r

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

Table : s

D	E
a	1
b	1

Table : $r \bowtie s$

A	B	C
α	a	γ
γ	a	γ

$D E = a \ 1$

A	B	C
α	a	α
α	a	γ
β	a	γ
γ	a	γ

$D E = b \ 1$

A	B	C
α	a	γ
γ	a	γ
α	a	β

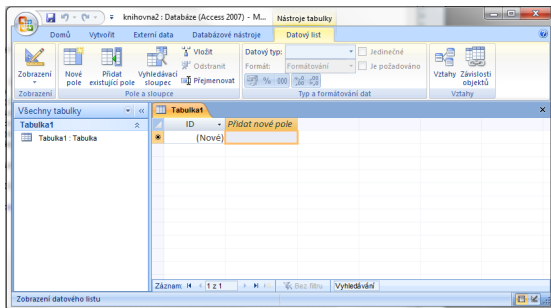
- nástroj na správu relačních databází
- vývojář: Microsoft
- součást Microsoft Office



Příklad knihovna - tabulky

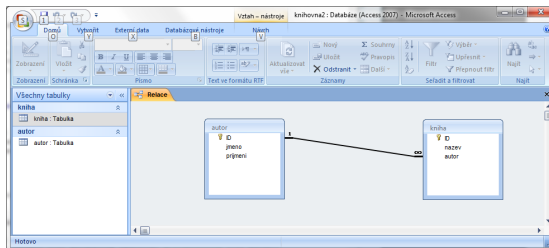


- vytvoříme prázdnou databázi (název, umístění)
- automaticky se vytvoří databáze s prázdnou tabulkou
- s tabulkou pracujeme obdobně, jako v excelu

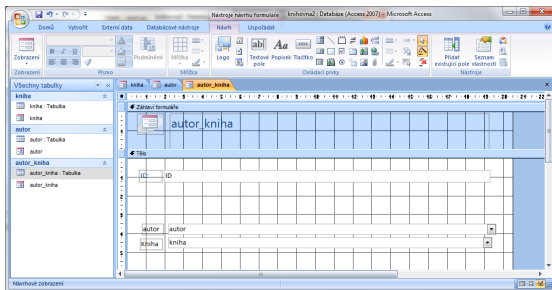


- přejmenujeme sloupce a nastavíme jim datový typ - záložka Datový list
- tabulku zavřeme a uložíme (přejmenovat tabulku)

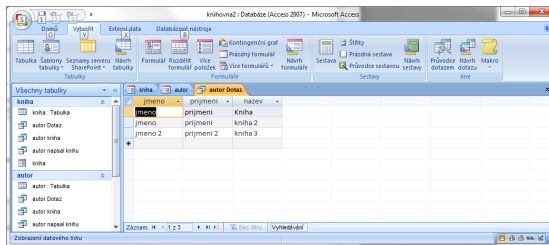
- vytvoříme vztahy: Databázové nástroje → Vztahy
 - přidáme tabulky
 - vytvořit novou
 - vybereme, co se má spojit (tabulku a sloupec)
 - zaškrtneme zajistit integritu



- formuláře slouží pro jednodušší zadávání položek do tabulky
- vytvoříme formuláře: Vytvořit → Formulář
 - automaticky se vytvoří - je možné ho editovat
 - je možné přidat například pole se seznamem (v tabulce autorKniha načítat autory a knihy z jiných tabulek)



- vytvoříme dotaz: Vytvořit → Průvodce dotazem





- navrhnete databázi pro lékaře, který si vede záznamy o pacientech (jméno, příjmení, r.č., telefon, adresa, ...)
- dále si vede záznamy o návštěvách jednotlivých pacientů (pacient, datum, diagnóza, předepsaná léčba, příznaky)