

Databázové systémy

2. relačný model dát, normálové formy relácie, dátové jazyky, DDL, DML, integritné obmedzenia

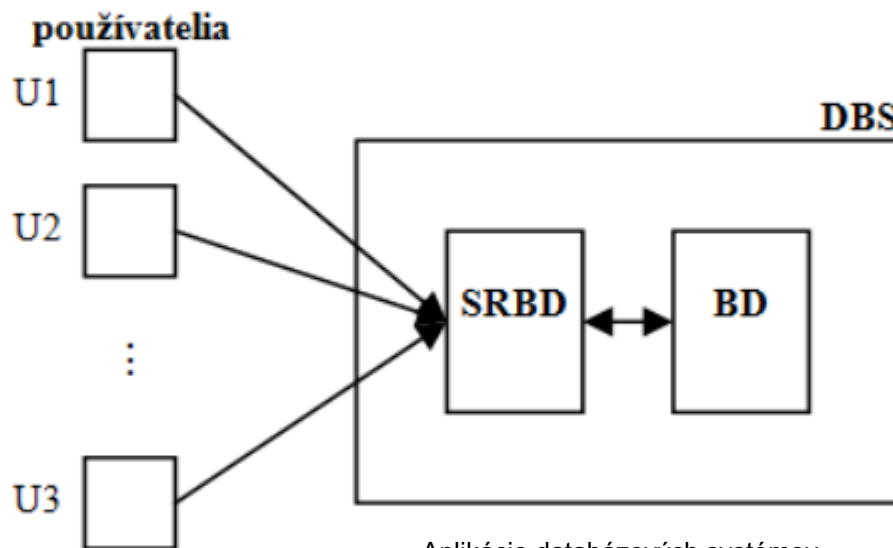


KKUI
Katedra kybernetiky
a umelej inteligencie

Opakovanie

Databázový systém pozostáva z:

- *báza dát (BD) resp. databáza* – organizovaná množina údajov
- *system riadenia bázy dát (SRBD)* – systém programov pre prístup a prácu s BD



Opakovanie

- **Model dát** je súhrn prostriedkov pre popis dát na konceptuálnej úrovni

Vytvorenie modelu dát na konceptuálnej úrovni spočíva v tom, že z reálneho sveta si vyberiem tie typy entít, ich vzťahy a atribúty, ktoré nás zaujímajú a sú pre aplikáciu dôležité.

- Modely dát:
 - **objektovo orientované** (E-R model) – zobrazuje množiny entít a vzťahy medzi nimi
 - **záznamovo orientované** (relačný model) –základnými prvkami sú tabuľky
 - **fyzické** (modely implementované na najnižšej úrovni)

Relačný model dát

- záznamovo orientovaný model
- zabezpečuje vysoký stupeň nezávislosti a minimálnu redundanciu dát spolu s ich konzistenciou
- reprezentovaný množinou dvojrozmerných tabuliek
- založený na teórií relácií, množinovom pojme relácií a relačnej algebre

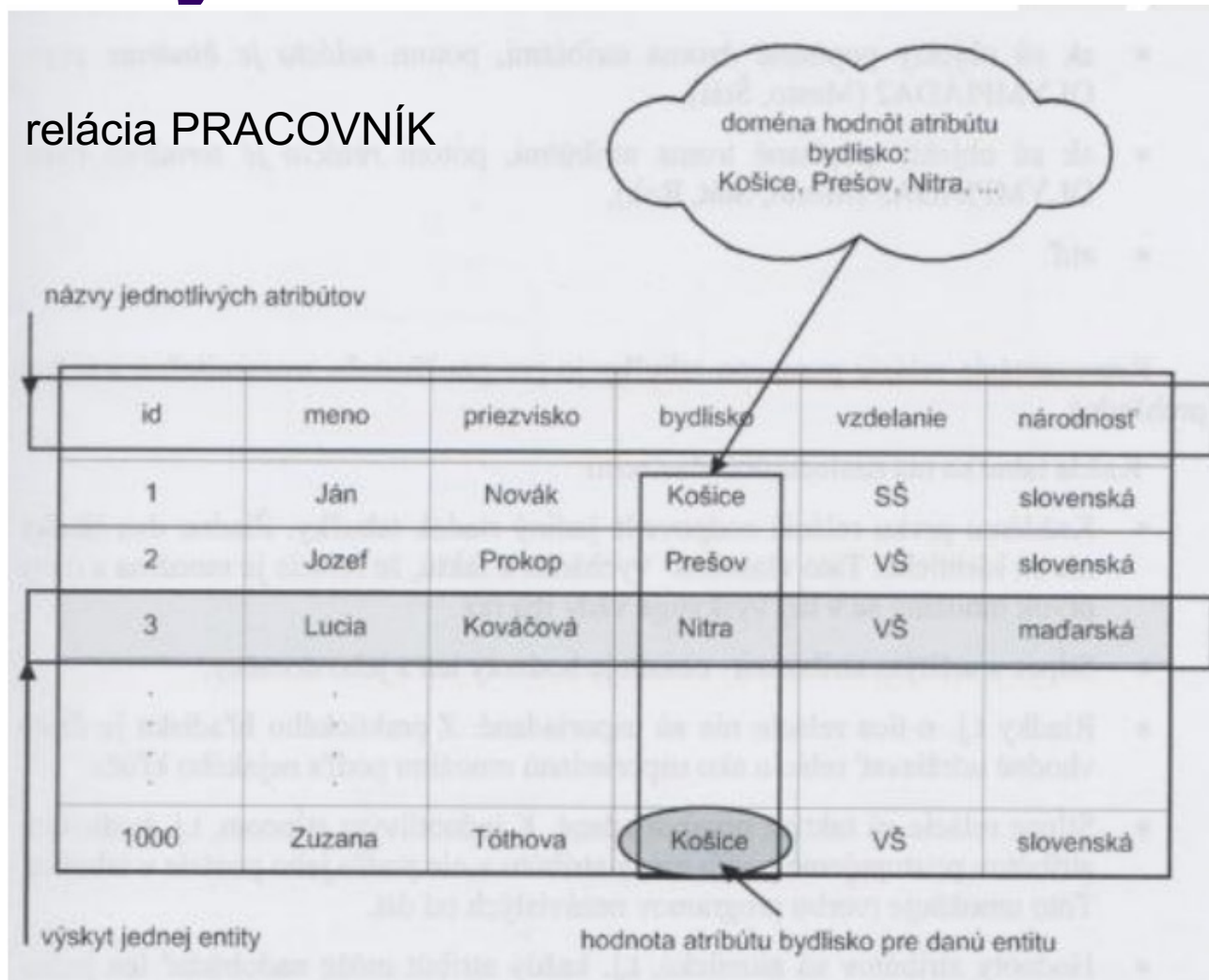
Relačný model dát

Relácia R je množina usporiadaných n -tíc. Je to podmnožina karteziánskeho súčinu $D_1 \times D_2 \dots \times D_n$, kde D_1, D_2, \dots, D_n sú domény atribútov.

Pre vytvorenie relácie R je potrebné zadať:

- konečnú množinu atribútov A , kde atribúty $a \in A$
- doménu $D(a)$ pre každý atribút $a \in A$
- podmnožinu karteziánskeho súčinu domén t.j. vlastné n -tice

Relačný model dát



Relačný model dát

Základné pojmy pre prácu s relačným modelom dát

- relácia = tabuľka
- n-tica = riadok tabuľky
- atribút = stĺpec tabuľky
- doména = množina všetkých hodnôt, ktoré môže nadobúdať atribút
- identifikačný kľúč = jednoznačný identifikátor riadkov tabuľky
- báza dát - zodpovedá množine tabuliek
- árnosť (stupeň) relácie = počet atribútov objektu udáva (ak sú objekty popísané jediným atribútom, relácia je unárna, dvomi - binárna, tromi - ternárna, atd.)

Relačný model dát

Každá tabuľka(relácia) má nasledujúce vlastnosti:

- každému prvku relácie zodpovedá jediný riadok tabuľky
- stĺpec s určitým atribútom obsahuje hodnoty len z jeho domény
- riadky t.j. n-tice relácie nie sú usporiadané
- stĺpce relácie sú taktiež neusporiadané, k hodnotám atribútov prístupujeme podľa mena atribútu a nie podľa jeho pozície v tabuľke
- hodnoty atribútov sú atomické
- schéma relačnej databázy je daná popisom štruktúry všetkých tabuliek a definíciou domén ich atribútov, ich integritných obmedzení, ako aj ostatných typov objektov databázového systému

Relačný model dát

Základné operácie nad reláciami:

- selekcia: výber riadkov tabuľky
- projekcia: výber stĺpcov tabuľky
- spájanie: karteziánsky súčin(JOIN)
- zjednotenie relácií (UNION)
- prienik relácií (INTERSECTION)
- rozdiel relácií (MINUS)

Návrh štruktúry relačnej databázy

Normalizácia:

- formálny proces na určenie, ktorý údaj prináleží ktorej tabuľke v relačnej databáze
- eliminovať nadbytočné polia, vyhnúť sa zlučovaniu tabuliek

Benefits normalizácie:

- eliminácia redundancie dát
- model reálneho sveta - reflektuje pravé závislosti medzi sledovanými položkami
- štruktúrovanosť dát, flexibilita modelu - zmenu údajov bez rizika vzniku nekonzistencie

Normálne formy relácie

Definícia 1.NF:

- Relácia je v prvej normálnej forme, ak hodnoty všetkých jej atribútov sú atomické t.j. prvkom žiadnej domény nie je pole.

1.NF

Meno	Priezvisko	Adresa	Telefon
Katarína	Biela	Hemerikova 2, Košice	0904 123 456, 0905 654 321
Zuzana	Malá	Hlavná 35, Poprad	0902 123 456
Peter	Mrkvička	Košická 10, Prešov	0910 123 456, 0918 654 321



Meno	Priezvisko	Adresa	ID
Katarína	Biela	Hemerikova 2, Košice	101
Zuzana	Malá	Hlavná 35, Poprad	102
Peter	Mrkvička	Košická 10, Prešov	103

ID	Telefon
101	0904 123 456
101	0905 654 321
102	0902 123 456
103	0910 123 456
103	0918 654 321

- atribút Telefon nie je atomický, teda relácia nie je v 1.NF
- riešením je dekompozícia na ďalšie dve relácie, atribútom zaisťujúcim identifikáciu údajov je atribút ID

Normálne formy relácie

Definícia 2NF

- Relácia je v druhej normálnej forme, ak je v prvej normálnej forme, v ktorej všetky sekundárne atribúty úplne funkčne závisia od primárnych atribútov.
 - ak atribút relácie závisí na celom kľúči a nezávisí na žiadnej jeho podmnožine hovoríme o úplnej funkčnej závislosti
 - atribút je primárny, ak sa vyskytuje v identifikačnom kľúči, ostatné atribúty sú sekundárne

2.NF

Vyrobok	Vyrobca	Kontakt	Mnozstvo	Cena
Horká čokoláda	Orion	0800 135 134	100	0,65
Horká čokoláda	Figaro	033 5915 412	150	0,7
Horalky	Sedita	031 7880 204	200	0,39
Mliečna čokoláda	Orion	0800 135 134	150	0,69
Mliečna čokoláda	Figaro	033 5915 412	200	0,65

Vyrobca	Kontakt
Orion	0800 135 134
Figaro	033 5915 412
Sedita	031 7880 204

Vyrobok	Vyrobca	Mnozstvo	Cena
Horká čokoláda	Orion	100	0,65
Horká čokoláda	Figaro	150	0,7
Horalky	Sedita	200	0,39
Mliečna čokoláda	Orion	150	0,69
Mliečna čokoláda	Figaro	200	0,65

- atribút Kontakt nespĺňa 2.NF, pretože nezávisí na celom PK(vyrobok+vyrobca), ale iba na časti
- problém nastáva v prípade, ak napr. Sedita prestane vyrábať Horalky. Zmazaním tohto záznamu stratíme aj informáciu o kontakte na firmu Sedita (firma Sedita stále existuje). Riešením je opäť dekompozícia

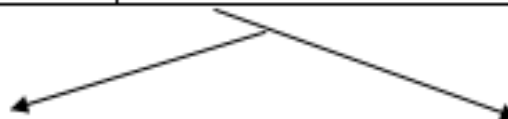
Normálne formy relácie

Definícia 3.NF

- Relácia je v tretej normálnej forme, ak je v druhej normálnej forme a navyiac platí, že množina všetkých sekundárnych atribútov je nezávislá t.j. že žiadny sekundárny atribút nezávisí na niektorom z ostatných sekundárnych atribútov
 - To znamená, že žiaden zo sekundárnych atribútov nie je tranzitívne závislý na kľúči.

3.NF

Vyrobca	Kontakt	Mesto	PSC
Orion	0800 135 134	Bratislava	811 01
Figaro	033 5915 412	Košice	040 01
Sedita	031 7880 204	Trnava	917 01



Vyrobca	Kontakt	Mesto
Orion	0800 135 134	Bratislava
Figaro	033 5915 412	Košice
Sedita	031 7880 204	Trnava

Mesto	PSC
Bratislava	811 01
Košice	040 01
Trnava	917 01

- atribút PSC závisí na atribúte Mesto, nie na atribúte Vyrobca(PK), teda nie je splnená 3.NF; PSC závisí na Mesto, Mesto závisí na Vyrobca, teda PSC tranzitívne závisí na Vyrobca
- teda ak zmažeme záznam o výrobcovi, prídem aj o informáciu o tom, aké PSC má mesto, v ktorom sídli (pri predpoklade, že v danom meste sídlil iba tento výrobca), riešením je opäť dekompozícia

Dátové jazyky

Dátový jazyk je množina operácií, ktoré umožňujú prístup k dátam v báze dát organizovaným pomocou dátového modelu.

Operácie dátového jazyka je možné rozdeliť do dvoch skupín a to na:

1. operácie výberu - týkajú sa vyhľadávania, získavania hodnôt dát alebo vzťahov medzi dátami uložených v BD
2. operácie modifikácie - týkajú sa zmeny hodnôt dát alebo vzťahov medzi dátami v BD

Dátové jazyky

Klasifikácia dátových jazykov

SRBD je možné rozdeliť podľa toho aké dátové jazyky poskytujú. Doterajší vývoj v tomto smere bol zameraný na:

- a) hostiteľský dátový jazyk
- b) samostatný dátový jazyk

Podľa výrazových prostriedkov používaných v dátových jazykoch môžeme tieto rozdeliť na:

- a) všeobecné
- b) aplikačne orientované dátové jazyky

Podľa miery procedurálnosti, ktorú jazyky poskytujú používateľovi môžeme dátové jazyky rozdeliť na:

- a) procedurálne
- b) neprocedulárne

Databázový jazyk SQL

- Databázový jazyk **SQL** (**Structured Query Language**) je počítačový jazyk primárne určený na manipuláciu a definíciu dát.
- vznikol na základe projektu spoločnosti IBM (70-te roky 20.stor.)
- 1979 - komerčne uvedenie do praxe spoločnosťou SDL(Oracle Corporation)
- pracuje s relačnými DB (IBM DB2, Oracle, Informix, MySQL, MS Access, ...)
- deklaratívny jazyk

SQL

Základné syntaktické konštrukcie jazyka SQL sa rozdeľujú do 4 zákl. skupín:

1. DML (Data Manipulation Language)

- príkazy určené na manipuláciu s dátami
- SELECT, INSERT, DELETE, UPDATE

2. DDL (Data Definition Language)

- príkazy určené na definíciu dát
- CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME

3. DCL (Data Control Language)

- určené na riadenie prístupu k objektom a dátam
- GRANT, REVOKE

4. TCL (Transaction Control Language)

- určené na riadenie zmien zrealizovaných DML príkazmi
- COMMIT, SAVEPOINT, ROLLBACK, SET TRANSACTION

Dátové typy databázy Oracle

- **znakové dátové typy**

- CHAR(veľkosť): reťazec znakov s pevnou dĺžkou, min.dĺžka 1, max 2000 bajtov
- VARCHAR(veľkosť): reťazec znakov s premenlivou dĺžkou o maximálnej dĺžke 4000 bajtov

- **číselné dátové typy**

- NUMBER (p,d): číslo s počtom všetkých číslic p, z čoho d číslic je za desatinnou čiarkou
- NUMBER(p,0) – celé číslo

- **dátové typy pre kalendárne a časové hodnoty**

- DATE: kalendárne dáta od dátumu 1.1.4712 p.n.l. do dátumu 31.12.9999 n.l.
- TIMESTAMP: presnejší dátový typ obsahuje rok, mesiac, deň, hodina, minúta, sekunda a zlomky sekúnd

- **dátové typy pre objemné dáta**

- BLOB: typ pre ukladanie binárnych dát max. veľkosť $(4GB-1) * DB_BLOCK_SIZE$,
- CLOB: veľké objekty v podobe jedno alebo viacbajtových znakov, podporuje pevnú aj premenlivú dĺžku a max. veľkosť objektu je $(4GB-1) * DB_BLOCK_SIZE$

- **rowid**

- hexadecimálny reťazec reprezentujúci jedinečnú adresu riadku v tabuľke; pseudostĺpec ROWID

SQL - vytvorenie tabuľky

```
CREATE TABLE <meno tabuľky>  
(<meno stĺpca1> <typ>(<dĺžka>),  
  <meno stĺpca2> <typ>(<dĺžka>),  
  ...);
```

- mená objektov databázy môžu obsahovať písmená, číslice a podčiarkovník(musia začínať písmenom); max. dĺžka je 30 znakov a ako identifikátory sa nesmú používať kľúčové slová
- meno objektu pozostáva z mena vlastníka a mena objektu
 <vlastník>.<meno tabuľky>

Integritné obmedzenia

- môže ich zahŕňať aj definícia tabuľky
- cieľom je zabrániť zadávaniu nesprávnych hodnôt do stĺpcov
- delíme ich na:
 - **Entitná integrita**
 - znamená, že každá tabuľka musí mať primárny kľúč, čo znamená, že stĺpec/stĺpce sú unikátne a majú zadané hodnoty (nemôžu byť NULL);
 - PRIMARY KEY, UNIQUE
 - **Doménova integrita**
 - atribúty nadobúdajú len prípustné hodnoty;
 - DEFAULT, NOT NULL, CHECK
 - **Referenčná integrita**
 - obmedzenia dát, pri ktorom každá hodnota konkrétneho atribútu tabuľky existovala ako hodnota iného atribútu i inej resp. tej istej tabuľke;
 - FOREIGN KEY

Obmedzenie PRIMARY KEY

- tabuľka môže mať len jeden primárny kľúč, a to jednoduchý alebo zložený
- zabezpečuje jedinečnosť primárneho kľúča a aj to, že žiadny atribút nebude obsahovať hodnoty NULL
- všeobecná syntax:

[CONSTRAINT <meno>] PRIMARY KEY

Obmedzenie PRIMARY KEY

-- jednoduchý primárny kľúč

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15),  
    objednavka_datum DATE,  
    dodavka_datum DATE,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY,  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5));
```

-- nepomenovaný zložený primárny kľúč

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15),  
    objednavka_datum DATE,  
    dodavka_datum DATE,  
    interne_cislo_obj NUMBER(12),  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5)  
    PRIMARY KEY (zakaznicke_cislo_obj, objednavka_datum) ;
```

Obmedzenie PRIMARY KEY

```
-- pomenovaný zložený primárny kľúč
CREATE TABLE objednavka (
    zakaznicke_cislo_obj NUMBER(15),
    objednavka_datum DATE,
    dodavka_datum DATE,
    interne_cislo_obj NUMBER(12),
    zak_cislo NUMBER(10),
    vybavuje NUMBER(5)
    CONSTRAINT pk_objednavka PRIMARY KEY (zakaznicke_cislo_obj,
    objednavka_datum)) ;
```

Obmedzenie UNIQUE

- zabezpečuje jedinečnosť hodnoty v prisluchajúcom stĺpci
- všeobecná syntax:

[CONSTRAINT <meno>] UNIQUE

Príklad:

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE,  
    dodavka_datum DATE,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY,  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5));
```

Implicitná hodnota DEFAULT

- zabezpečuje, že ak užívateľ pri vkladaní záznamu nezadá hodnotu pre atribút, ktorý ma v definícii klauzulu DEFAULT, bude do tabuľky vložená implicitná hodnota atribútu

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE DEFAULT SYSDATE,  
    dodavka_datum DATE,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY,  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5));
```

Obmedzenie NOT NULL

- obmedzenie NOT NULL vyžaduje zadanie hodnoty do príslušného stĺpca
- všeobecná syntax:
`[CONSTRAINT <meno>] NOT NULL`

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE DEFAULT SYSDATE,  
    dodavka_datum DATE NOT NULL,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY,  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5));
```

Obmedzenie CHECK

- definuje podmienky, ktoré musia byť splnené pre každý záznam, napr. hodnota atribútu v danom rozsahu alebo splnenie určitej podmienky
- môže byť špecifikované nad konkrétnym stĺpcom alebo ako obmedzenie nad tabuľkou (pri obmedzení nad tabuľkou podmienka CHECK kontroluje hodnotu daného atribútu vo vzťahu k hodnotám iného atribútu danej tabuľky)
- všeobecná syntax:
`[CONSTRAINT <meno>] CHECK(<podmienka>)`

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE DEFAULT SYSDATE,  
    dodavka_datum DATE NOT NULL,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY  
        CHECK (LENGHT(interne_cislo_obj)=12  
        AND SUBSTR(interne_cislo_obj, 1,4)='OBJ-'),  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5));
```

Obmedzenie CHECK

Ak je obmedzenie CHECK definované ako obmedzenie nad tabuľkou je možné realizovať len jednoduché porovnania atribútov a logické operátory AND, OR a NOT.

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE DEFAULT SYSDATE,  
    dodavka_datum DATE NOT NULL,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY  
        CHECK (LENGTH(interne_cislo_obj)=12  
            AND SUBSTR(interne_cislo_obj, 1,4)='OBJ- '),  
    zak_cislo NUMBER(10),  
    vybavuje NUMBER(5))  
CONSTRAINT datумы_por  
    CHECK(dodavka_datum > objednavka_datum));
```

- definuje vzťah k primárnemu alebo unikátnemu kľúču inej alebo tej istej tabuľky
- môže byť definované ako obmedzenie stĺpca aj ako obmedzenie nad tabuľkou
- referujúca tabuľka sa nazýva podradená tabuľka resp. child-table
- referovaná tabuľka sa nazýva nadradená resp. parent-table
- syntax

```
[CONSTRAINT meno] [FOREIGN KEY (<stĺpec|stĺpce>)]  
REFERENCES <tabuľka>[(<stĺpec|stĺpce>)]  
[ON DELETE CASCADE]
```


Obmedzenie FOREIGN KEY

```
CREATE TABLE objednavka (  
    zakaznicke_cislo_obj NUMBER(15) UNIQUE,  
    objednavka_datum DATE DEFAULT SYSDATE,  
    dodavka_datum DATE NOT NULL,  
    interne_cislo_obj NUMBER(12) PRIMARY KEY  
        CHECK (LENGTH(interne_cislo_obj)=12  
        AND SUBSTR(interne_cislo_obj, 1,4)='OBJ-'),  
    zak_cislo NUMBER(10) CONSTRAINT fk_zak_cislo REFERENCES  
zakaznik, (keďže sme nezadali názov stĺpca, odkazuje na PK parent-table)  
    vybavuje NUMBER(5)  
    CONSTRAINT datумы_por  
        CHECK(dodavka_datum > objednavka_datum)  
    CONSTRAINT fk_zam_cislo  
        FOREIGN KEY (vybavuje) REFERENCES  
zamestnanec(ev_cislo));
```

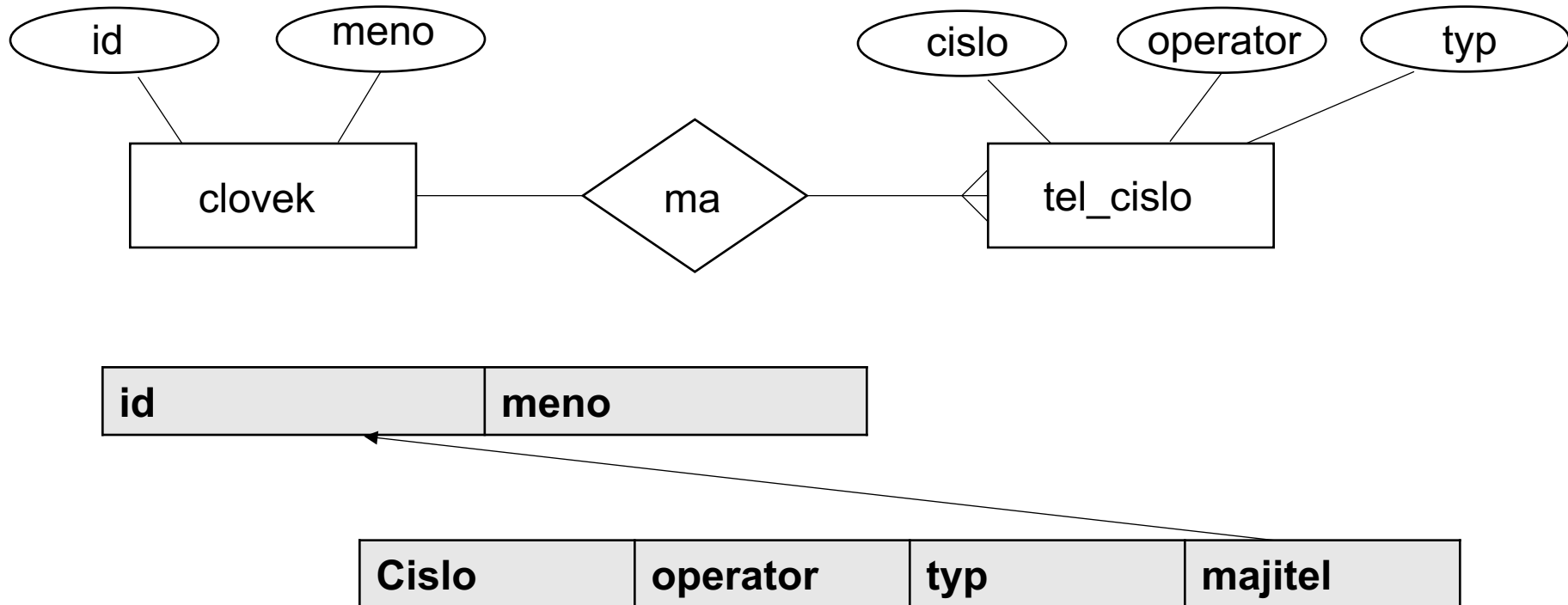
Transformacia E-R modelu na relačný model

- meno množiny entít → názov tabuľky
- entita → riadok tabuľky
- atribút → stĺpec
- vzťah → kombinacia primarny a cudzi kluc
- pravidlá pre doménovú a entitnú integritu → pomocou integritných obmedzení

Transformacia E-R modelu na relačný model

vzťah 1:N

jeden vzor má viacero obrazov,
napr. jeden človek má viacero telefonnych cisel



Transformacia E-R modelu na relačný model

vzťah M:N

viac vzorov sa zobrazí do viacerých obrazov

napr. viacero učiteľov môže vyučovať viacero predmetov;

