

# Informační systémy

27.2.2015

# Základní pojmy DTB zpracování

## ENTITA - OBJEKT

celá posloupnost položek popisuje objekt. Taková struktura položek, která má ucelený význam (zachycuje všechny potřebné údaje o sledovaném objektu) se nazývá **záznamem** (větou, recordem). Je to obvykle skupinová položka.

## MNOŽINA ENTIT – MNOŽINA OBJEKTŮ – DATOVÝ SOUBOR – OBSAH TABULKY

množinu záznamu stejného typu, zaznamenávající ucelenou informaci o množině sledovaných objektů a uloženou na paměťovém médiu, nazýváme **datovým souborem**. Množiny záznamů si můžeme snadno představit ve tvaru **tabulky**, kde každý objekt je popsán jedním řádkem a každý atribut objektu je v jednom sloupci.

## DATABÁZE

Množinu datových souborů, uchovávajících data o nějakém uceleném úseku reality, nazýváme **databází**.

# Základní pojmy DTB zpracování

## **SYSTEM ŘÍZENÍ BÁZE DAT – SŘBD**

programový systém (prázdný, bez datových souborů a bez aplikačních programů), umožňující definování datových struktur a datových souborů, řešící fyzické uložení dat ve vnější paměti počítače, umožňující manipulaci s daty a formátování vstupních i výstupních informací, nazýváme **systemem řízení báze dat**.

## **APLIKAČNÍ ÚLOHA**

**Aplikační úlohou** nad SŘBD nazýváme konkrétní program napsaný pomocí programových prostředků použitého SŘBD nad konkrétní databází, pro tuto úlohu vytvořenou.

# Základní pojmy DTB zpracování

## INFORMAČNÍ SYSTÉM

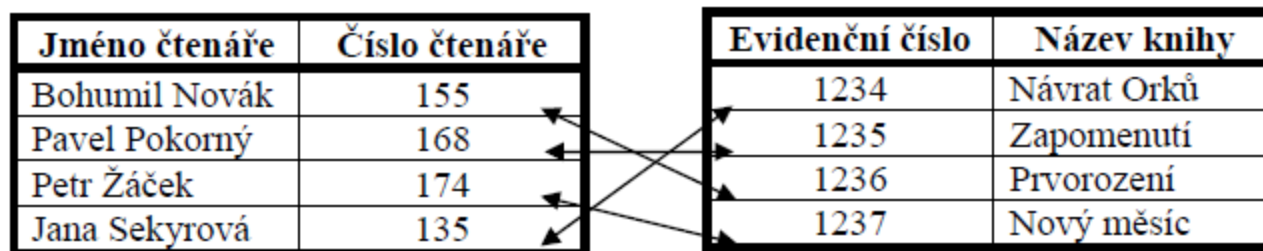
Aplikační úlohy nad společnou databází tvoří ucelený systém, nazývaný **databázovým nebo informačním systémem** (dále jen IS) nad použitým SŘBD.

V tomto pojetí tedy IS rozumíme celek, řešící rozsáhlejší oblast aplikační, naprogramovaný v jednom SŘBD s vhodně navrženými datovými strukturami tak, aby všechny aplikační úlohy k nim měly optimální přístup. Řeší **uložení, uchování, zpracování a vyhledávání informací a umožňuje jejich formátování** do uživatelsky přívětivého tvaru.

# Vztahy mezi entitami (relace)

1:1

V relaci 1:1 odpovídá jednomu záznamu v první tabulce maximálně jeden záznam v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce.



# Vztahy mezi entitami (relace)

## 1:N

V relaci 1:N odpovídá jednomu záznamu v první tabulce žádný, jeden nebo více záznamů v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce.

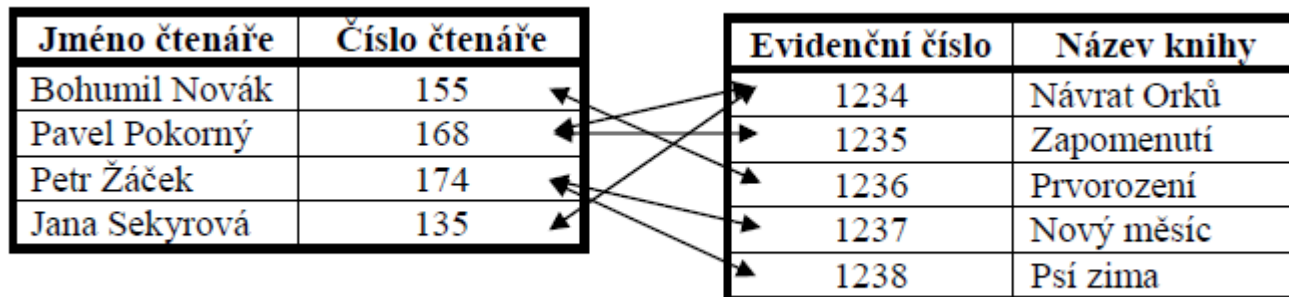
# Vztahy mezi entitami (relace)

## N:N

- V relaci N:N odpovídá jednomu záznamu v první tabulce žádný, jeden nebo více záznamů v druhé tabulce a naopak jednomu záznamu v druhé tabulce žádný, jeden nebo více záznamů v první tabulce.
- Chcete-li vyjádřit relaci typu N:N, musíte vytvořit třetí tabulku, která se často nazývá spojená tabulka, jež rozdělí relaci typu N:N na dvě relace typu 1:N. Primární klíč z těchto dvou tabulek vložíte do třetí tabulky. Výsledkem je, že třetí tabulka zaznamená každý výskyt nebo instanci relace

# Vztahy mezi entitami (relace)

N:N





# Úrovně DBS

Způsoby pohledu na data v databázi

- FYZICKÝ (INTERNÍ) – způsob fyzického uložení dat na disk
- KONCEPTUÁLNÍ – popis struktury databáze (tabulek), popis vztahů mezi uloženými daty
- EXTERNÍ – popis dat z pohledu uživatele, tj. uživatelské prostředí (formuláře, sestavy,...)

# Datové modely

Způsob uložení dat v databázi

- Hierarchický
- Síťový
- Relační
- Objektový

# Hierarchický DM

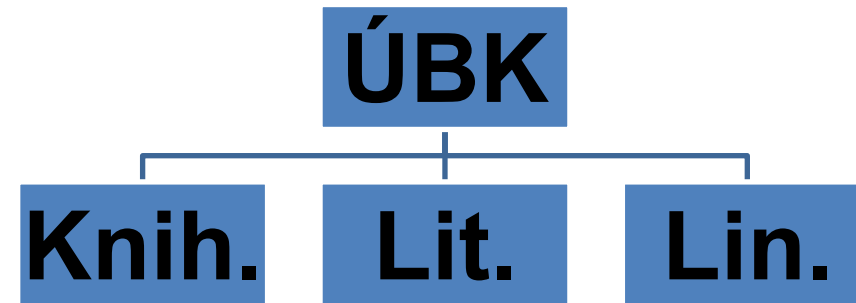
Záznamy jsou organizovány ve stromové struktuře

## VÝHODY:

- Řeší snadno a rychle vztahy 1:N
- Nezáleží na fyzické struktuře dat

## NEVÝHODY:

- Problémy při řešení vztahů M:N
- Problémy při změně struktury dat



# Síťový DM

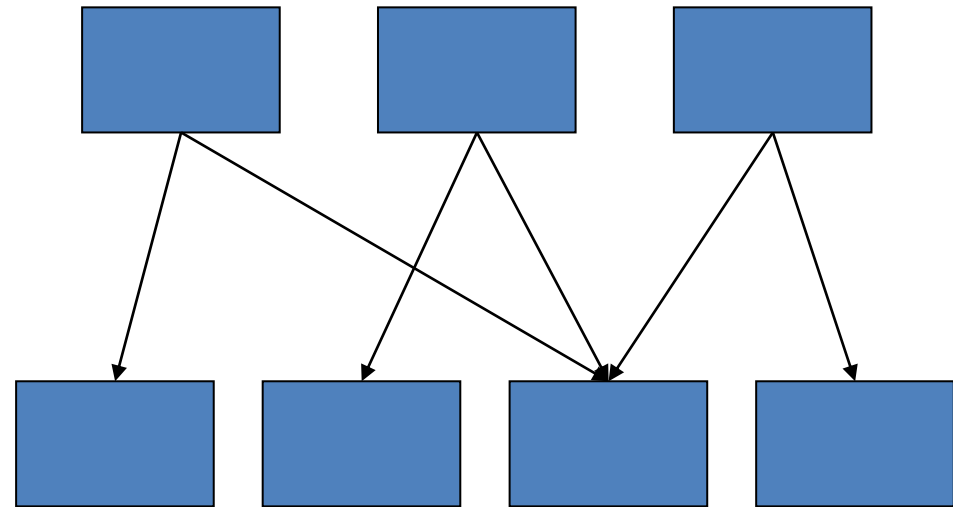
Data jsou reprezentována kolekcemi záznamů a vztahů mezi nimi.

## VÝHODY:

- Řeší snadno a rychle vztahy 1:N i M:N
- Nezáleží na fyzické struktuře dat
- Rychlé vyhledávání

## NEVÝHODY:

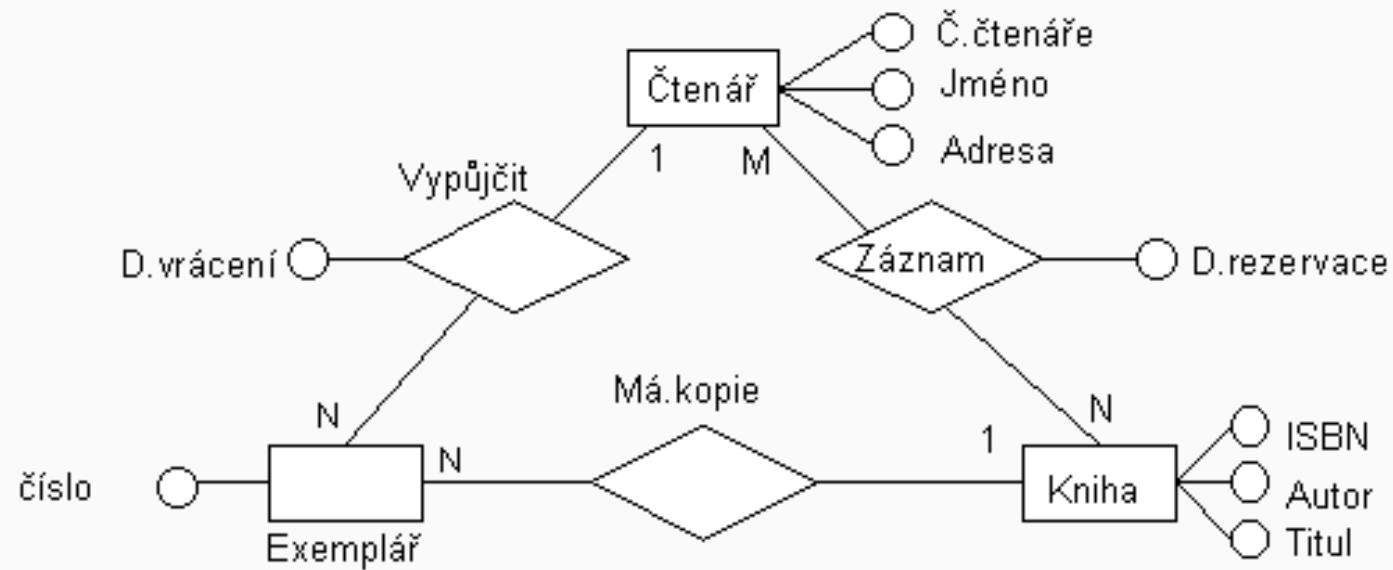
- Problémy při změně struktury dat



# Relační DM

- **RDBS** – relační databázový systém
  - Informace uchovávané v jednom typu objektu jsou uchovávané v tabulkách s určitou strukturou
  - Tabulky jsou navzájem provázány relacemi
  - Relace usnadňují vyhledávání různých informací uložených v těchto tabulkách
- **V SOUČASNOSTI NEJPOUŽÍVANĚJŠÍ MODEL**

# Schéma entitně-relačního modelu



# Relační DM

Číslo objednávky	Název výrobku
10456	003
10456	004
10457	101
10457	002
10457	001
10457	102
10458	006
10459	001
10459	004

Id zaměstnance	Jména zaměstnanců	Zaměstnán od
01	Jan Novák	1.2.2005
02	Emil Král	15.2.2001
03	Václav Nový	24.8.2002

Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

Číslo objednávky	Datum objednávky	Id zaměstnance
10456	12.8.2005	01
10457	12.8.2005	02
10458	13.8.2005	02
10459	13.8.2005	03

# Relační tabulka

atributy

záznamy

Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128




# Objektový DM

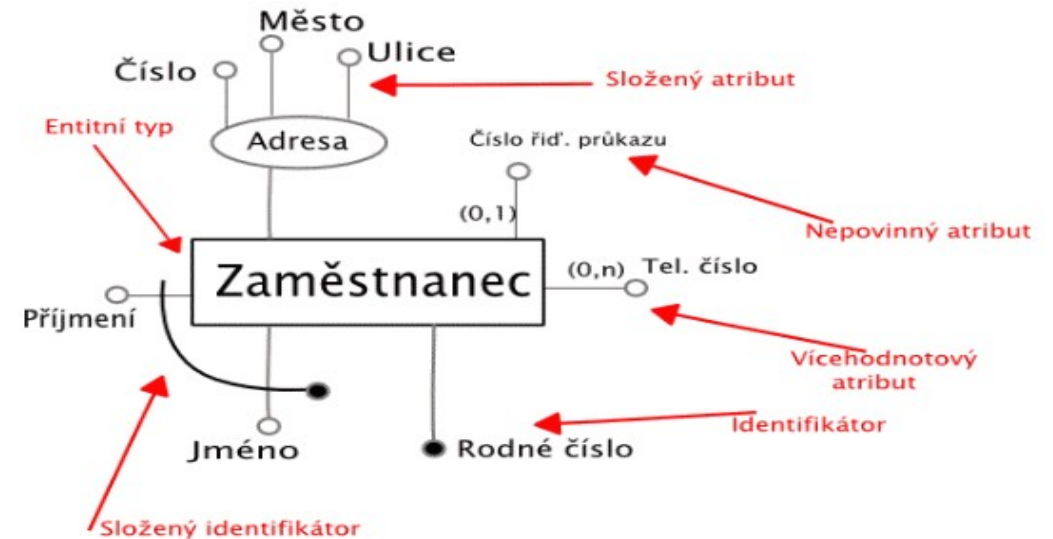
- Vychází z principů objektově orientovaného přístupu
- Objekt – datová struktura definovaná jako třída s určitými vlastnostmi a metodami
- Komunikace mezi objekty probíhá pomocí zpráv
- Výhody
  - Nejen statické, ale i dynamické chování objektů
  - Možné vytváření složitějších objektů
  - Snadnější zadávání dotazů

# Konceptuální model


- Slouží k popisu dat v databázi nezávisle na jejich fyzickém uložení
- Umožňuje zobrazit a popsat objekty v databázi a vztahy mezi nimi z hlediska jejich významu a chování
- Výsledkem je implementačně nezávislé schéma obecně aplikovatelné v jakémkoli prostředí
- Znázorňuje se v podobě **ER diagramu**, který definuje entity (třídy prvků), jejich atributy a relace (vztahy) mezi nimi

# Pojmy v ER-modelu

- ENTITNÍ TYP (v diagramu ) reprezentuje třídu entit (např. Zaměstnanec).
- Každý ENTITNÍ TYP má nějaké atributy (např. Jméno), z nichž některé mohou být identifikátory (jednoznačně určují instanci entity).
- Pokud ET nemá žádné identifikátory explicitně označené, jsou jimi všechny atributy dohromady (tzv. složený indikátor). Identifikátory mohou být více atributové.



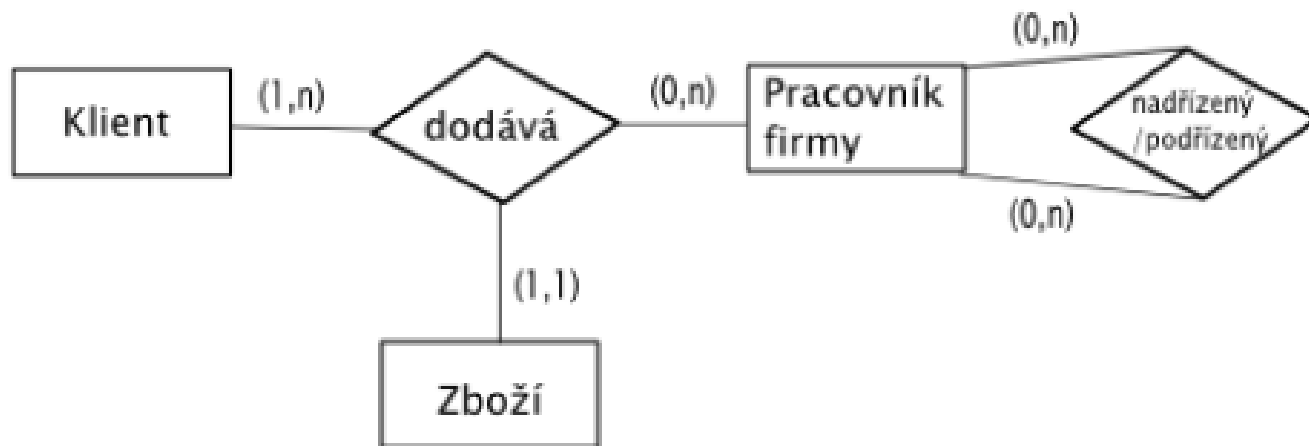
# Vztahový typ

- VT (v diagramu ) popisuje vztahy mezi jednotlivými entitami – s těmi entitami, se kterými je v nějakém vztahu, je spojen čarou.
- Vztah může mít danou i KARDINALITU (kolik entit z každé strany do vztahu vstupuje), která může být typu 1:1, 1:n, n:n a je značena vedle čáry spojující vztahový typ s entitou.
- Entity ve vztahu mohou mít navíc povinné či nepovinné členství (vstupovat do něj vždy či někdy)

# Vztahový typ

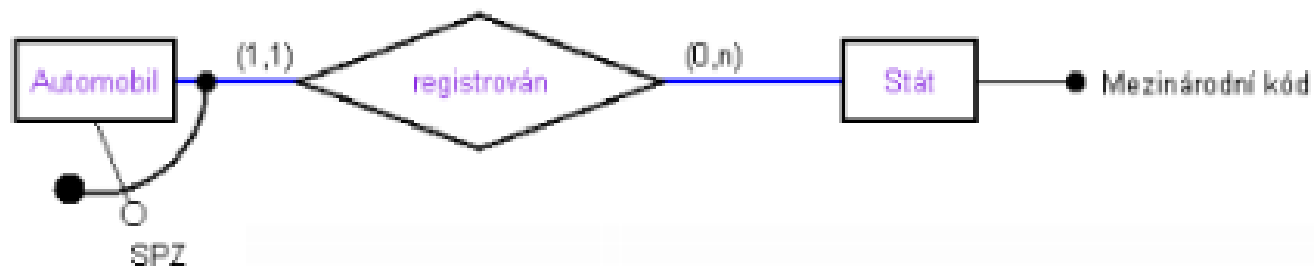
- Vztahy mohou být buď binární nebo **n-ární**, ale více než ternární vztahy se většinou neobjevují. Vztahy mohou být i **REKURZIVNÍ**, tj. do vztahů vstupují entity stejného typu.
- Instance vztahového typu je jednoznačně určena identifikátory instancí ve vztahu.
- Některé entitní typy mohou být spolu identifikovány (nebo přímo identifikovány) vztahem – pak se nazývají slabé entitní typy.

# Vztahové typy



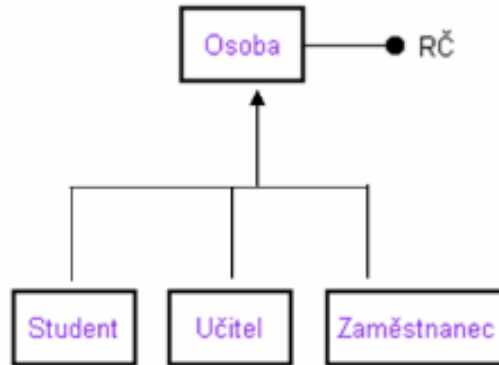
Obr. Ukazuje ternární vťah s různými kardinalitami – klientovi někdo dodává zboží jednou až n-krát, pracovník dodává nula až n-krát zboží (tj. jde o nepovinné členství ve vztahu, můžou existovat pracovníci, kteří nic nedodávají) a zboží je vždy dodáváno právě jednou. Na zaměstnancích je zároveň ukázán rekurzivní binární vztah.

# Slabý entitní typ



Obr. Ukazuje slabý entitní typ – automobil je identifikován SPZ a zároveň státem, ve kterém je registrován.

# ISA hierarchie



ISA hierarchie je rozšíření ER diagramů o dědičnost entit - tj. rozdělení entitních typů na subtypy (a přidání dalších vztahů nebo atributů pro subtypy). V ISA hierarchii se povoluje pouze jednonásobná dědičnost, navíc potomci nějakého entitního typu musí být jednoznačně identifikováni předkem (tj. všechny entity v hierarchii sdílí identifikátor).



# Základní pojmy

- **Klíč (key)**
  - Jeden nebo několik atributů tabulky určený pro setřídění záznamů podle hodnot v tomto poli (numerický, textový)
- **Unikátní klíč**
  - Klíč tabulky, ve kterém se každá hodnota atributu vyskytuje nejvýše jedenkrát
- **Duplicitní klíč**
  - Klíč tabulky, ve kterém se každá hodnota atributu může vyskytovat vícekrát (u více různých záznamů)

# Základní pojmy

- **Jednoduchý klíč**
  - Klíč tabulky, který je tvořen pouze jediným atributem
  - Nejčastěji se vyskytující typ klíče
  - I uměle vytvořený (id)
- **Složený klíč**
  - Klíč tabulky, který je tvořen alespoň dvěma atributy
  - Často součást tzv. spojovací tabulky

# Základní pojmy

- **Primární klíč (primary key)**
  - Klíč tabulky, který slouží k jednoznačné identifikaci záznamu
  - Musí být unikátní
  - Označuje se PK
  - V jedné tabulce může být nanejvýše jeden primární klíč
- **Cizí klíč (foreign key)**
  - Klíč tabulky, který slouží k propojení (vytvoření relace) s primárním klíčem jiné tabulky
  - Často obsahuje duplicitní hodnoty
  - Označuje se FK
  - V jedné tabulce může být i více cizích klíčů

# ER model (postup vytváření)

## 1. Určení typu entit

- zvolení množiny objektů stejného typu
- např. Objednávka, Zaměstnanec, Výrobek

## 2. Určení typů relací

- vztahů, do kterých mohou příslušné entity vstupovat
- např. objednávka obsahuje výrobek

## 3. Určení atributů

- přiřazení jednotlivým entitám a vztahům
- např. Objednávka (číslo, datum, ...)

## 4. Určení integritních omezení

- zpřesnění navrženého modelu
- např. atribut datum je datového typu Datum a čas

# ER model

Relační tabulka

atributy

záznamy

Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

**FK** **FK**

Číslo objednávky	Název výrobku
10456	003
10456	004
10457	101
10457	002
10457	001
10457	102
10458	006
10459	001
10459	004

**PK**

Id zaměstnance	Jména zaměstnanců	Zaměstnán od
01	Jan Novák	1.2.2005
02	Emil Král	15.2.2001
03	Václav Nový	24.8.2002

**PK** **FK**

Číslo objednávky	Datum objednávky	Id zaměstnance
10456	12.8.2005	01
10457	12.8.2005	02
10458	13.8.2005	02
10459	13.8.2005	03

**PK**

Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

# QBE (Query By Example)

- QBE = dotaz podle vzoru, je rozhraní pro zadávání dotazů grafickou formou. Znalost konkrétního dotazovacího jazyka není vyžadováno.
- Lze provádět základní relační operace (selekce, projekce, spojení), ale také další operace jako třídění, součty, apod. (v závislosti na konkrétním SŘBD).

Č.Studenta	Příjmení	Jméno	Adresa	Věk
	√ No..	√	√	√ >18, <=25

# Transakce

- TRANSAKCE = posloupnost operací nad objekty BÁZE DAT, které realizují jednu nebo více ucelených operací z hlediska uživatele.
- Transakce začíná vykonáním prvního příkazu nebo speciálním příkazem <např. BeginTrans>.
- Transakce může skončit úspěšně nebo neúspěšně.
- Neúspěch může vzniknout: poruchou hardware, chybou programového vybavení, chybnými daty, apod.



# Transakce

- Bod, od kterého lze považovat transakci za úspěšnou se nazývá **bodem potvrzení**.
- Informace o změnách během transakce se ukládají do tzv. **žurnálového (transakčního) souboru**, teprve po dosažení bodu potvrzení se promítnou do báze dat.
- Při vlastní transakci se data v bázi dat nemění, tzn. při chybě nebo poruše nedojde k porušení konzistentního stavu báze dat.

# SQL (Structured Query Language)

- SQL = strukturovaný dotazovací jazyk.
- Standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.
- Vznik v 70. letech 20. století (IBM – výzkum relačních databází) jako sada příkazů pro ovládání RD – vznik jazyka SEQUEL (Structured English Query Language).
- Cílem bylo vytvořit jazyk, ve kterém se příkazy tvoří syntakticky a v návaznosti na přirozený jazyk (angličtina).

# SQL

- Jazyk zahrnuje nástroje pro tvorbu databází (tabulek) a dále nástroje pro manipulaci s daty (vkládání dat, mazání, vyhledávání, atd.).
- SQL patří mezi tzv. DEKLARATIVNÍ PROGRAMOVACÍ JAZYKY, což znamená, že kód jazyka SQL nepíšeme v samostatném programu, ale vkládáme jej do jiného programovacího jazyka, který je již procedurální.
- Se samotným SQL můžeme pracovat pouze v případě terminálového přístupu na SQL server (skrze příkazový řádek).

# Příkazy jazyka SQL

SQL příkazy se dělí do 4 zákl. skupin:

- Příkazy pro MANIPULACI S DATY

(např.: SELECT, INSERT, UPDATE,...)

- Příkazy pro DEFINICI DAT

(např.: CREATE, DROP,...)

- Příkazy pro ŘÍZENÍ PŘÍSTUPOVÝCH PRÁV

(např.: GRANT, REVOKE,...)

- Příkazy pro ŘÍZENÍ TRANSAKČÍ

(např.: START TRANSACTION, COMMIT,...)

# SQL

SQL se skládá z několika částí:

- **DDL – Data Definition Language** – jazyk pro vytváření databázových schémat a katalogů,
- **SDL – Storage Definition Language** – jazyk pro definici způsobu ukládání tabulek,
- **VDL – View Definition Language** – jazyk pro návrháře a správce, určuje vytváření pohledů na tabulky,
- **DML – Data Manipulation Language** – jazyk obsahující základní příkazy INSERT, UPDATE, DELETE, SELECT.

S jazykem DML pracují nejvíce koncoví uživatelé a programátoři databázových aplikací.

# SQL - uplatnění

SQL plní v současnosti celou řadu úloh, např.:

- Interaktivní dotazovací jazyk - uživatel zapíše příkazy SQL do interaktivního SQL programu, který najde potřebná data a zobrazí je na obrazovce.
- Databázový programovací jazyk - k použití při tvorbě databázové aplikace.
- Jazyk pro správu databází - správci databází využívají SQL pro definici datových struktur a kontrolu uložených dat.

# SQL - uplatnění

- Klient/server jazyk - programy pro počítače používají SQL pro komunikaci v lokální počítačové síti (LAN) s databázovými stanicemi (database servers), kde jsou uložena sdílená data.
- Jazyk distribuované báze dat - systémy spravující distribuované databáze, používají SQL jako doplněk pro distribuci dat mezi několika spojenými počítačovými systémy.
- Komunikační jazyk SŘBD v LAN - v síti s několika různými SRBD je SQL jedinou cestou pro jejich vzájemnou komunikaci.