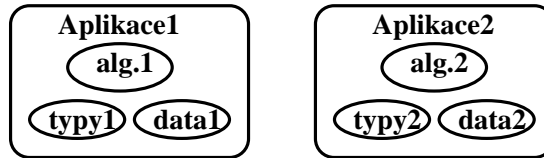


Databázové systémy – trocha teorie

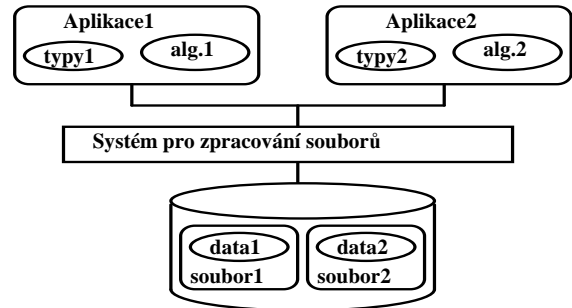
Základní pojmy

Historie vývoje zpracování dat:

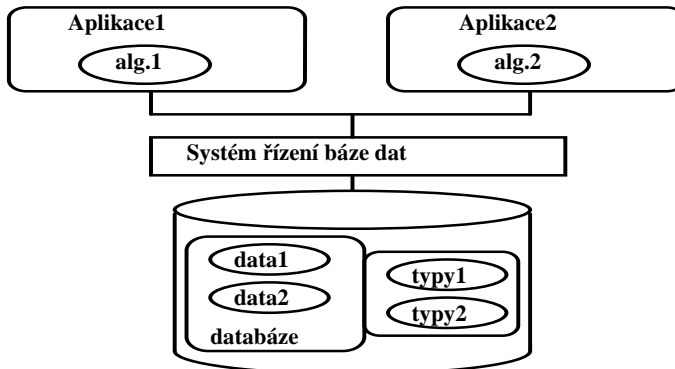
- 50. Léta vše v programu
nevýhody poměrně jasné



- vytvoření systémů pro zpracování souborů
nevýhody systému pro zpracování souborů:
 - redundance dat a nekonzistence
 - problémy s přístupem k datům (neplánované dotazy)
 - izolace dat (sbírání dat z jednotlivých souborů)
 - problémy s bezpečností dat (omezený přístup)
 - problémy integrity (implementace integritních omezení)
- ⇒ systémy řízení báze dat (SRBD)



- 2.polovina 60.let – systémy řízení báze dat



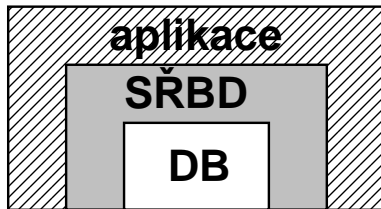
Databáze (DB) - perzistentní strukturovaná data, využívaná aplikačními systémy dané instituce.

Další vlastnosti databáze:

- *Integrovaná* - lze chápat jako sjednocení několika datových souborů s odstraněním redundance (úplným nebo částečným)
- *Sdílená* – typicky víceuživatelský přístup s případným omezením pohledu
- *Bezpečná* - snadněji se realizuje omezení práv přístupu k datům

SRBD – systém řízení báze dat, programová vrstva řešící operace nad DB. Tento poněkud krkolomný název vznikl přeložením původního anglického termínu DBMS -- DataBase Management System. Mezi SRBD patří také programy jako Oracle, MS SQL Server, Sybase, Informix, Progress, (MS Access).

Převážná většina dnes používaných SŘBD při uspořádání údajů v databázi vychází z **relačního modelu dat**. Název tohoto modelu vychází z relační algebry, což je matematický aparát, na kterém relační model dat staví. V tomto modelu jsou údaje uspořádány do **tabulek**. Tabulka zpravidla shromažďuje údaje o jednom druhu objektů. Můžeme tak mít například tabulku s osobními údaji zaměstnanců. Jednotlivé řádky odpovídají jednotlivým zaměstnancům. Sloupce pak obsahují informace o pracovnících například by to mohly být následující údaje: osobní číslo, jméno, rodné číslo, adresa a výše platu. Sloupcům tabulky obvykle říkáme v databázové terminologii *položky* nebo *atributy*. Jednotlivé řádky se pak nazývají *záznamy*.



- odstínění uživatele (aplikace) od technických detailů
- operace typu vytvoření DB souboru, ...

Databázový systém (DBS) – systém, který zahrnuje :

- technické prostředky
- data DB
- programové vybavení – SŘBD, vývojové prostředky, generátory sestav, aplikace, utility,...
- uživatelé DB (administrátor databáze – manipuluje se schématem databáze, aplikační programátoři – tvorba aplikací, zralí uživatelé – formulují dotazy , naivní uživatelé – pracují s aplikací)

Když to celé zjednodušíme tak se dá ve výsledku napsat : **DBS = DB+SŘBD**

Další pojmy:

Instance databáze – kolekce informací uložených v DB v konkrétním časovém okamžiku

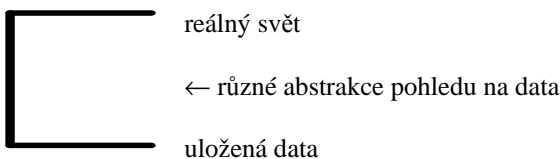
Schéma databáze – metainformace popisující data v databázi (v systémovém katalogu nebo v slovníku dat)

Logické schéma, fyzické schéma, externí schéma(subschéma) –uroveň pohledů, konceptuální schéma, interní schéma.

Analogie datového typu a hodnoty proměnné(v daném okamžiku). Logické schéma databáze odráží použitý datový model.

Datové modelování

Datová abstrakce – jedním z důležitých úkolů DBS je poskytnout uživatelům abstraktní pohled na data (jsou skryty detaily uložení a správy dat).

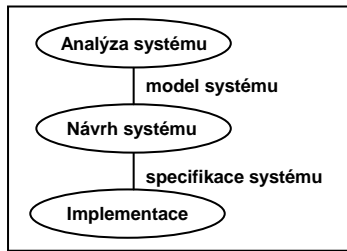


Základní úrovně abstrakce zahrnují:

- *fyzická (interní) úroveň* - popisuje data, *jak* jsou skutečně uložena
- *konceptuální úroveň* - popisuje *jaká* data jsou skutečně uložena v databázi a jaké vztahy mezi nimi existují.
- *úroveň pohledů* - popisuje pouze část databáze, která představuje data viditelná jednotlivými uživateli.

ER diagram a jeho transformace, konceptuální modelování

Konceptuální modelování: Fáze datové analýzy využívající modelů založených na objektech. Cíl - konceptuální schéma.



E-R modely:

- založeny na chápání světa jako množiny základních objektů (*entity*) a vztahů mezi nimi (*relationship*) popisuje data “v klidu”

entita - rozlišitelný, jednoznačně identifikovatelný objekt, o němž chceme mít informace v DB
Př) Jan Novák, r.č. 100650/0467; účet číslo 7892346-631 u KB

entitní množina - množina entit téhož typu
Př) Zaměstnanec, zákazník, účet

atribut - vlastnost entity, jejíž hodnotu chceme mít v DB
jednoduchý nebo složený (např. adresa:ulice, číslo, město)

vztah - asociace mezi několika entitami
Př) zákazník - má_účet – účet,

- obecně n-nární, nejčastěji binární, obecné lze převést na binární
- sémantika vztahu určuje "roli" entity ve vztahu (pracuje pro)
- vztah může mít atributy (má_účet = { datum })
- členství entity ve vztahu může být povinné nebo volitelné

kardinalita zobrazení - vyjadřuje počet vztahů dané vztahové množiny, ve kterých se entita může vyskytovat

pro binární vztahovou množinu R mezi entitními množinami A a B může kardinalita být:
1:1 1:M M:1 M:N

Odlišení entity a vztahu:

klíč - atribut (případně složený), jednoznačně identifikující entitu (vztah) v množině

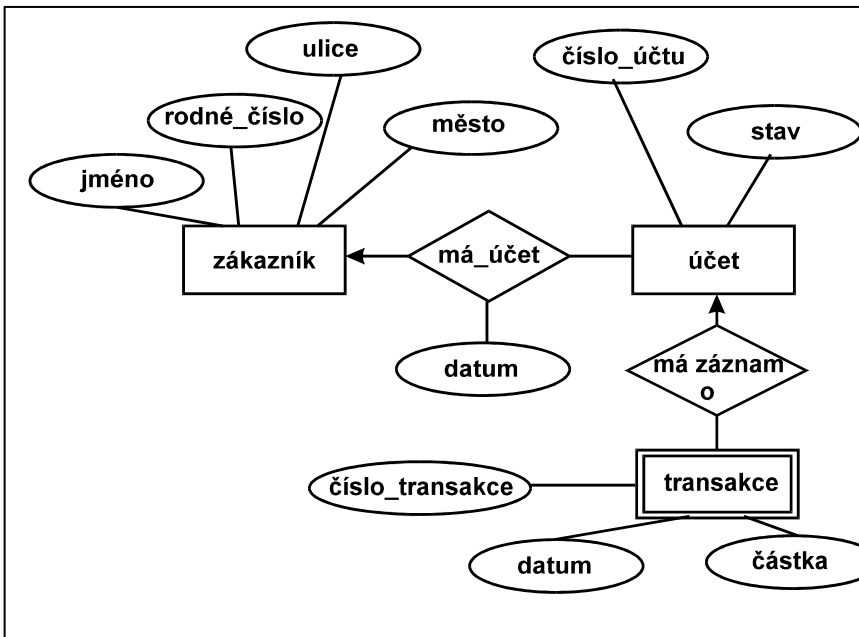
kandidátní klíč - minimální, *primární klíč*

Př) rodné číslo, (jméno, ulice, město)

E-R diagram

Vytváříme ho, když potřebujeme popsat entitní množiny, jejich atributy a vztahy.

Transformace ERD na tabulky



Ma zaznam – vztah

Ucet-entitni množina

Transakce-slabá ent. množina

Cislo uctu –atribut, prim. klíč

zakaznik – silná ent. množina

zakazník :

Rodne cislo	Jmeno	Ulice	mesto
-------------	-------	-------	-------

Učet:

Cislo_uctu	stav
------------	------

transakce:

Cislo_uctu	Cislo_transakce	Datum	Castka
------------	-----------------	-------	--------

- reprezentace vztahů

Cislo uctu	Stav	Rodne cislo	Datum
------------	------	-------------	-------

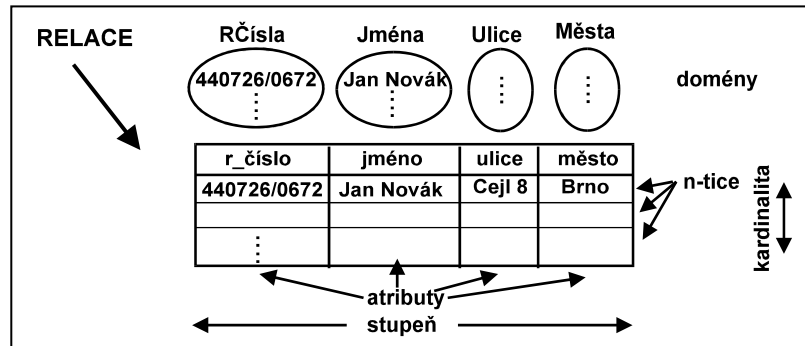
Cislo uctu	Rodne cislo	Jmeno pobocky
------------	-------------	---------------

Relační model dat: struktura dat, obecná integr. Omezení, relační algebra

Relační databáze je vnímána uživatelem jako kolekce tabulek.

Definice dat: definice tabulky = Bázová tabulka – autonomní, pojmenovaná tabulka, záhlaví sloupců + řádky hodnot.

Relační struktura dat:



Doména - pojmenovaná množina skalárních hodnot téhož typu.

Skalární hodnota - nejmenší sémantická jednotka dat, atomická (vnitřně nestrukturovaná)

- každý atribut je definován na nějaké doméně
- domény omezují porovnávání
- většina SŘBD pojem domén nepodporuje vůbec nebo jen částečně

Relace R na doménách D_1, D_2, \dots, D_n je dvojice $R = \langle H, B \rangle$, kde H značí záhlaví relace a B tělo relace.

Záhlaví relace je množina: $H = \{(A_1:D_1), (A_2:D_2), \dots, (A_n:D_n)\}$ $A_i \neq A_j$ pro $i \neq j$,

A_i ($i = 1, \dots, n$) značí atributy a D_i ($i=1, \dots, n$) jsou odpovídající domény.

Tělo relace je tvořeno časově proměnnou množinou n-tic:

$B(t) = \{r_1, r_2, \dots, r_m(t)\}$, kde $r_i = \{(A_1:vi_1), (A_2:vi_2), \dots, (A_n:vi_n)\}$

$i = 1, 2, \dots, m(t)$, n - stupeň (řád) relace, m - kardinalita relace.

Vlastnosti relace:

1. neexistují duplicitní n-tice,
2. n-tice jsou neuspořádané,
3. atributy jsou neuspořádané (nemusí vždy být)
4. hodnoty jednoduchých atributů jsou atomické

Relační databáze:

- vnímána uživatelem jako kolekce časově proměnných normalizovaných relací - tabulek
- veškerá data v relační databázi jsou reprezentována explicitní hodnotou

Integritní pravidla v relačním modelu

Primární klíč (superklíč) - atribut, který jednoznačně identifikuje n-tici v relaci

Kandidátní klíč - Atribut k relace R se nazývá *kandidátním klíčem*, když má tyto dvě časově nezávislé vlastnosti:

- jednoznačnost
- minimalita

každá relace má alespoň jeden kandidátní klíč, atribut, který je součástí kandidátního klíče se nazývá *klíčový*

Primárním klíčem je jeden z kandidátních klíčů (vybraný), zbývající kandidátní klíče se nazývají *alternativní* (někdy také *sekundární*).

- způsob výběru primárního klíče není v relačním modelu specifikován
- primární klíč je základním prostředkem adresace n-tic v relačním modelu

Cizí klíč (Foreign key, FK)

Atribut se nazývá *cizí klíč*, právě když splňuje tyto časově nezávislé vlastnosti:

Každá hodnota *FK* je buď plně zadaná nebo plně nezadaná.

Existuje bázová relace *RI* s primárním klíčem *PK* takovým, že každá zadaná hodnota *FK* je identická s hodnotou *PK* nějaké *n*-tice relace *RI*.

Pravidlo referenční integrity:

DB nesmí obsahovat žádnou nesouhlasnou hodnotu cizího klíče.

V praxi systém přímo podporuje cizí klíče nebo nepodporuje a aplikace si data musí kontrolovat sama.

Jazyk SQL (Structured Query Language)

- 1975 - Sequel v System R

- 1986 - standard ANSI (ANSI-86 SQL)- tzv.úroveň 1,v IBM vlastní standard SQL (SAA-SQL Systems Application Achitecture Database Interface)

- 1989 - ANSI-89 SQL, též ISO - tzv.úroveň 2, integritní dodatek (Integrity Addendum)

- dominantní úloha DB2

- řada dialektů SQL - základ norma (úroveň 1, 2, případně tzv. integritní dodatek)

Příkazy jazyka pokrývají oblasti:

- jazyka pro definici dat, (DDL data definition language, def. celé struktury db, def. atributů, zakládání, modifikace a indexace relací)

- jazyka pro manipulaci s daty, (DML data manipulation language, jazyk pro zápis, změny a mazání dat z db.

- jazyk na definování dotazů na databázi (DQL data query language)

– manipulačního jazyka pro hostitelské prostředí (embedded SQL) ,

- definice pohledů,

- řízení přístupových práv (authorization),

- integrity dat,

- řízení transakcí

Definice dat:

CREATE TABLE, ALTER TABLE, DROP TABLE práce s tabulkama

CREATE INDEX, DROP INDEX

CREATE SYNONYM, DROP SYNONYM

Manipulace s daty:

Dotaz: SELECT s jeho všema možnostma

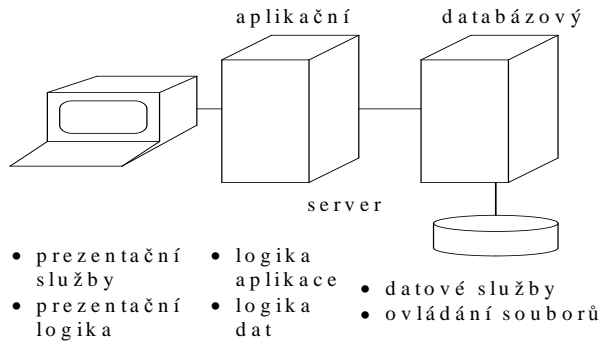
Agregační funkce: COUNT, SUM, AVG, MAX, MIN

Plus všechny možné další příkaz jazyka SQL, vytváření pohledů, sestav atd.

Architektury:

Klient/Server

Klient/server se třemi vrstvami



⊕ • nejvhodnější architektura

Př. MySQL / MS SQL databáze + PHP + (X)HTML

Lokální

Předpokládá se, že databáze je provozována na stejném stroji jako klient, který ji využívá. Ideálním případem je, pokud se vývoj, ladění i vlastní práce odehrávají přímo v integrovaném prostředí.

Př. Prostředí aplikace MS Access