Databáze

* Určitá uspořádaná množina informací (dat) uložená na paměťovém médiu
* Kolekce vzájemně propojených informací, se kterou pracujeme jako s jediným celkem.
* **Database Management Systém (DBMS)**
  + Systém pro řízení báze dat (např. mySQL)
  + Je software poskytující veškeré služby pro správu a údržbu databáze
  + Databázový software
  + Velmi složité a náročné programy
* **Služby DBMS**
  + Ukládání dat do souborů (na disk, do paměti)
  + Správa souborů běžných přístupů k datům a zabránění konfliktním přístupům
  + Transakční zpracování dat
    - Pokud dojde uvnitř dávky příkazů k chybě je celá dávka odmítnuta
  + **Dotazovací jazyk**
  + Zálohování dat
  + Ochrana dat před neoprávněným přístupem
* **Datová abstrakce**
  + Vnější vrstva, logická vrstva a fyzická vrstva
* **SQL a NOSQL databáze**
  + Alternativy
    - SQL – tabulky, řádky, sloupečky
    - NOSQL – náhodně uložený
  + Lze je kombinovat
  + Tabulky a JSON dokumenty
  + Pevná struktura / flexibilita
  + Snadné připojování tabulek / nelze propojovat dokumenty
  + Integrita – soudržnost dat, navazují na sebe data (např. smažu tabulku třídy zároveň se studenty)
  + Transakce – nahraju všechna data nebo žádná data
* **Přehled SQL DBMS**
  + Microsoft Access
  + Microsoft SQL Server
  + Oracle
  + MySQL
  + Sybase
  + Postgres
  + Interbase
* **NOSQL DBMS**
  + mongoDB
  + CouchDB
  + Redis
* **Nezávislost databází**
  + Fyzická nezávislost dat
    - Schopnost přesunout své objekty, aniž by tím byli poškozeni uživatelé
  + …..
* **Flat file**
  + Uložení dat do souboru bez informací o struktuře dat
  + Organizaci dat zajišťuje samotná aplikace
  + Historický
  + Metadata
* **Relační model**
  + Dr. E. F. Codd
  + Data jsou uložena ve dvourozměrných tabulkách, propojených prostřednictvím samotných dat
  + Spojování tabulek
  + Jednoduchost dotazů a implementace
* **Objektový model**
  + Přidává k běžným statickým datům navíc také programovou logiku, která zajišťuje určitou funkcionalitu
  + Zbytečně komplexní a komplikovaný
* **Objektově relační model**
  + Kombinace relačního a objektového modelu
  + Jednoduchost + kód v databázi
  + Dnes v nejlepších DBMS

**Návrh databáze**

* Database Design
* ER Diagram je vizuální pomůcka
* **Konceptuální úroveň**
  + Vymezení informací ukládaných v systému, co je obsahem modelu, abstraktní model reality, ignorujeme technologická omezení
* **Technologická (logická) úroveň**
  + Jak budou data strukturována v reálné databázi, jak budou data ukládána
* **Implementační (fyzická) úroveň**
  + Určuje, čím je model realizován, konkrétní datové typy, programovací jazyk
* **Požadavky**
  + Pokrývá požadavky zadavatele
  + Reflektuje možná budoucí rozšíření
* **E-R Model**
  + Konceptuální model
  + Myšlenkový
  + Velmi abstraktní model situace
  + Definuje, co je obsahem modelu = koncept
  + Různé způsoby zápisu
  + **Peter Chen (1976)**
* **Entity**
  + Vyjadřuje jakoukoliv věc, osobu nebo třeba ideu, která je předmětem modelování
  + Entitní typ představuje množinu všech podobných Entit.
  + **Entitní typ** = Student
  + **Entita** = Jaromír Vopička
  + V popisu situace představují obvykle podstatná jména
* **Atribut**
  + Vlastnosti, které definují entitní typ
  + Informace, které nás na entitě zajímají
  + **Klíčový atribut** (klíč) = Atribut, který danou identitu identifikuje
  + **Superklíč =** (kolekce) atributů, které společně identifikují entitu
  + **Kandidátní klíč** = jeden z možných způsobů identifikace entity
  + **Primární klíč** = vybraný kandidátní klíč
  + **Atomický** nebo **složený atribut** (i klíč)
* **Relace**
  + Vyjadřují vztahy mezi entitními typy
  + Lze je pojmenovat
  + Při popisu situace se jedná o slovesa
* **Kardinalita**
  + Vyjadřuje, kolik entit daného typu se na relaci podílí nebo může podílet
  + Učitel vyučuje více studentů
  + Student může být vyučován více učiteli
* **Druhy relací dle kardinality**
  + **1 : 1** – one-to-one – člověk a jeho židlička
  + **1 : N** – one-to-many – student je ve třídě
  + **N : M** – many-to-many – knihy a autoři – v relačních databázích již není možné vytvořit bez tzv. asociativní entity
  + Všechny vazby mohou být realizovány i na jedné tabulce = **rekurentní relace** – člověk je potomkem člověka
* **Další vazby**
  + Vazba typu **ISA** je vztahem mezi supertypem a subtypem = dědičnost = subtyp dědí všechny atributy ze supertypu a může k nim přidávat své vlastní
  + Odpovídá dědičnosti v OOP
  + Vyjadřuje specializaci entit – např. student je speciálním případem člověka
  + V relačních databázích není čistě vyjádřitelná
* **Parcialita**
  + Vyjadřuje povinnost účasti v relaci
  + 0 – **nemandatorní relace**
  + 1 – **mandatorní relace**
  + Učitel nemusí vyučovat žádné studenty
  + Student musí být vyučován učitelem
* **Logická úroveň návrhu**
  + Způsob konceptuálního návrhu, který je možný realizovat do skutečné databáze
  + Datové typy
  + Vazba: Cizí klíče – primární klíče
  + Další omezení
  + **Cizí klíč** = Primární klíč z jiné tabulky realizující nějakou vazbu
* **Datové typy**
  + Velmi elementární
  + Int, char, varchar, enum, float

**Transformace relací**

* **1:1**
  + Člověk a lidský mozek
  + Občan a jeho občanský průkaz
  + Stát a jeho hlavní město
  + Je-li vazba mandatorní -> spojení entit do jedné tabulky
  + Pokud není -> spojení entit do jedné tabulky s prázdnými sloupci
    - Nebo se realizuje jako vazba 1:N
* **1:N**
  + Kniha a stránka
  + Hotel a pokoj
  + Třída a student
  + Realizuje se přidáním klíče z tabulky na straně 1 do tabulky na straně N jako cizího klíče
  + Pokud vazba není mandatorní, může být cizí klíč prázdný (null)
* **N:M**
  + Knihy a autoři
  + Filmy a herci
  + V relačních databázích je nelze realizovat přímo
  + Je nutné vytvořit asociativní tabulku