**Databáze**  je systém souborů s pevnou strukturou záznamů. Tyto soubory jsou mezi sebou navzájem propojeny pomocí klíčů. V širším smyslu jsou součástí databáze i [softwarové](https://cs.wikipedia.org/wiki/Software) prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software se  nazývá [**systém řízení báze dat**](https://cs.wikipedia.org/wiki/Syst%C3%A9m_%C5%99%C3%ADzen%C3%AD_b%C3%A1ze_dat)**(SŘBD**). Běžně se označením *databáze* – v závislosti na kontextu – myslí jak uložená data, tak i software (SŘBD).

Databázové modely nám souhrnně říkají, jakým způsobem jsou data organizována a jakým způsobem s nimi můžeme pracovat, resp. jak jsou uložena a reprezentována.

**Hierarchický model**

Datový model, ve kterém jsou data uspořádána ve stromové datové struktuře. Je to první z datových modelů, který byl v minulosti hojně využíván v praxi. Nejznámějším hierarchickým systémem řízení báze dat byl od společnosti [IBM](https://cs.wikipedia.org/wiki/IBM). Data jsou uložena postupně v hierarchii, např. kartotéka u lékaře. Data hledáme v postupu: Ročník > Písmeno příjmení > Jméno a příjmení > Konkrétní datum narození v případě shody jména.

**Základní konstruktory**:

**Věta**– obdoba relace v relačním datovém modelu. Každá věta je definována svými atributy. Oproti relačnímu modelu neexistuje omezení při tvorbě atributu – atributy nemusí být atomické (není zde zaveden pojem [normalizace](https://cs.wikipedia.org/wiki/Normalizace_datab%C3%A1ze) datové základny), atributy mohou být i vícerozměrnými strukturami jako jsou například [pole](https://cs.wikipedia.org/wiki/Pole_(datov%C3%A1_struktura)).

**Vlastnicko-členský vztah** (také vztah rodič–dítě) – vztah kardinality 1 : N mezi dvěma větami; věta na straně jedna se nazývá *vlastník*, věta na straně N se nazývá *člen*.

·        hierarchické schéma má jeden kořen, který není člen v žádném vztahu

·        každá věta, kromě kořene, je členem právě v jednom vztahu

·        každá věta může být vlastníkem 1 až n počtu vět

·        věta, která není vlastníkem v žádném vztahu, se nazývá list

Z výše uvedených vlastností vyplývá, že hierarchický model je schopen bezproblémově modelovat pouze vztahy kardinality 1 : N. Vztahy kardinality M : N je možné modelovat za využití dvou vztahů 1 : N za pomocí tzv. *směrníkových vět*, jedná se o speciální případ „virtuální věty“, která obsahuje klíče z obou vztahů 1 : N a binární ukazatele na umístění vztahů v hierarchickém schématu.

**Síťový (Stromový) model**

Stejná struktura jako jsou složky v PC. Data mohou být rovněž organizována heirarchicky, ale zároveň složky *Program File*s a *Users*nemají nadřazenou složku a jsou si postavením rovny, proto jde o strukturu síťovou. Důležitým aspektem tohoto modelu je rovněž fakt, že jeden rodič může mít více potomků.

Tento model je podobný hierarchickému databázovému modelu, který byl po dlouhou dobu takřka výhradně užívaným databázovým modelem. Na rozdíl od hierarchického databázového modelu poskytuje navíc vztahy více ku více, tedy jedna entita mohla mít více otců. Tato datová koncepce však byla překonána [relační koncepcí databáze](https://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_datab%C3%A1ze). Také navíc umožňuje [rekurzi](https://cs.wikipedia.org/wiki/Rekurze), tedy entita může být otcem svému otci.

Nevýhodou využívání síťové databáze je její nepružnost a z toho vyplývající obtížná změna její struktury.

·        zdokonalení hierarchické koncepce

·        přímá návaznost na hierarchickou koncepci

·        síťová struktura se modeluje pomocí tzv. Bachmanových diagramů (C. Bachmann tvůrce síťové koncepce)

**Základní konstruktory:**

**Věta**je souhrn vzájemně souvisejících datových položek.

**Set**je vztah mezi dvěma větami kardinality 1:N.

·        je definován jménem, vlastnickou a členskou větou.

·        ta na straně 1 se jmenuje věta vlastnická a na straně N věta členská.

·        set má své výskyty.

·        výskyt setu je dán výskytem konkrétního vlastníka a výskytem jeho členů

**Typy členství v setu:**

1.     *Automatické*- výskyt členské věty nově vstupující do databáze je automaticky připojen k odpovídajícímu výskytu v setu.

1.1.    *povinné*- výskyt členské věty nemůže existovat v databází aniž je přiřazen k výskytu určité vlastnické věty

1.2.    *volitelné*- výskyt členské věty může existovat v databází aniž je přiřazen k výskytu určité vlastnické věty

2.     *Manuální* - manuálně můžeme volně přesouvat výskyt věty do příslušného výskytu setu.

2.1.     *povinné* (viz výše)

2.2.     *volitelné* (viz výše)

**Objektový model**

Objektová [databáze](https://cs.wikipedia.org/wiki/Datab%C3%A1ze)je databázový řízený systém, ve kterém je informace reprezentována ve formě objektu a používá se v objektově orientovaném programování. Tento model sdružuje v sobě objekty, které mají vlastnosti a metody. Objektové databáze se odlišují od [relačních databází](https://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_datab%C3%A1ze), které jsou tabulkové. Také existují objektově relační databáze, které jsou hybridem dvou předchozích přístupů.

Většina objektových databází navíc nabízí různé typy dotazovacích jazyků, které dovolují objektům používat způsob deklarativního programování. To je oblast objektových dotazovacích jazyků, integrací dotazů a navigačních rozhraní.

Přístup k datům může být rychlejší, protože objekty můžou být vyhledány přímo bez hledání přes ukazatel.

Jinou možností rozlišování mezi produkty je způsob zadání diagramu, podle kterého je databáze je definovaná. Základní podmínkou je, že programovací jazyk a databázový diagram mají stejnou definici.

Mnoho objektových databází podporuje systém správy verzí. Nějaký objekt může být představen jako sada všech verzí. Kromě toho se různé verze objektů mohou chovat jako samy objekty.

Efektivita takové databází je dobrá tam, kde je potřeba obsahovat velkou částku informaci v jednom elementu. Například banka může poskytnou klientovi informaci o transakcích, záznamy uživatele, …

Porovnání s RDBMS

Objektové databáze uchovávají komplexní data a vztahy mezi daty, bez mapování vztahů řádků a sloupců, což je činí velmi pohodlnými pro aplikaci, pracující s komplexními daty. Objekty mají relaci N:N (vztah many-to-many) a jsou přístupné přes ukazatele. Ukazatele jsou připojeny k objektům pro provozní vztahů. Dalším významným rozdílem oproti RDBMS je, že můžou být programované s malými změnami bez vlivu na celý systém.

Nejužívanější objektovou databází je např. ActiveDirectory v doménách Windows. Kde objekty mohou představovat *Uživatelské účty* nebo *Počítače aj*. Každý z objektů pak má své vlastnosti a metody.

* Uživatelské účty
  + Vlastnosti: název, login, heslo.
  + Metody: ověření hesla, změna hesla.
* Počítače
  + Vlastnosti: název, ip adresa.
  + Metody: přihlášení uživatele.
* Tiskárny
  + Vlastnosti:: název, typ, stav.
  + Metody: tisk.

**Entitně-Relační (ER)**

Model vyplývá ze vzájemných vztahů (relací) mezi entitami. Na rozdíl od síťového modelu lze rodiče rekurzivně přiřadit k totožnému rodiči.

viz následující přednáška

**XML/JSON**

Tento databázový model je textovým blobem, který má strukturovaný zápis a využívá se především pro import/export dat a nebo pro sdílení dat skrze aplikační rozhranní (API).

**HIT (Hierarchický klasifikační model)**

Datový model HIT (Hierarchický klasifikační model) je [konceptuální model](https://moodle-stary.skolakrizik.cz/mod/lesson/view.php?id=6323) pro reprezentaci a organizaci dat, který kombinuje hierarchickou a klasifikační strukturu. Byl vyvinut na Ústavu teoretické a aplikované informatiky Akademie věd České republiky (ÚTIA AV ČR) v Praze (rok 1986 Marie Duží , Zlatuška). Tento model byl navržen pro klasifikaci a organizaci biologických druhů v taxonomických systémech. Jeho vývoj probíhal v rámci úsilí o efektivní a strukturované zachycení vztahů mezi biologickými druhy. Datový model HIT se snaží reflektovat reálné vztahy mezi objekty a zohledňuje hierarchii a klasifikaci entit.

**Základní charakteristiky a principy modelu HIT:**

1. **Hierarchie:** Model HIT vytváří hierarchickou strukturu, kde objekty (např. organismy) jsou uspořádány do různých úrovní. Každá úroveň představuje určitý druh klasifikace.
2. **Klasifikace:** Objekty jsou klasifikovány do různých kategorií podle jejich vlastností, charakteristik a vztahů. Klasifikace může být založena na různých kritériích, jako jsou morfologické rysy, genetická podobnost, ekologické faktory atd.
3. **Unikátní identifikace:** Každý objekt má unikátní identifikátor, který umožňuje jeho jednoznačnou identifikaci v rámci celého systému.
4. **Vzájemné vztahy:** Model umožňuje definovat vztahy mezi objekty, např. nadřazený-podřazený vztah mezi různými úrovněmi klasifikace.