**1. Relační databázové systémy – popis, terminologie**

* Relační databáze je taková databáze, která ukládá data do tabulek, které mají mezi sebou "vztahy" (z angličtiny "**relations**")
* **DBMS**, přes který modifikujeme a přistupujeme k databázi zajišťuje:
  1. **Integritu databáze** (kontrola a správnost dat)
  2. **Víceuživatelský přístup** (transakce)
  3. **Přístup k datům pomocí jazyka SQL**
* **Základní názvy**:
* Tabulka = **Entita**
* Sloupec v tabulce = **Atribut**
* Řádek v tabulce = **Instance**
* Vztah mezi entitami = **Relace**

A picture containing text, screenshot, font, diagram

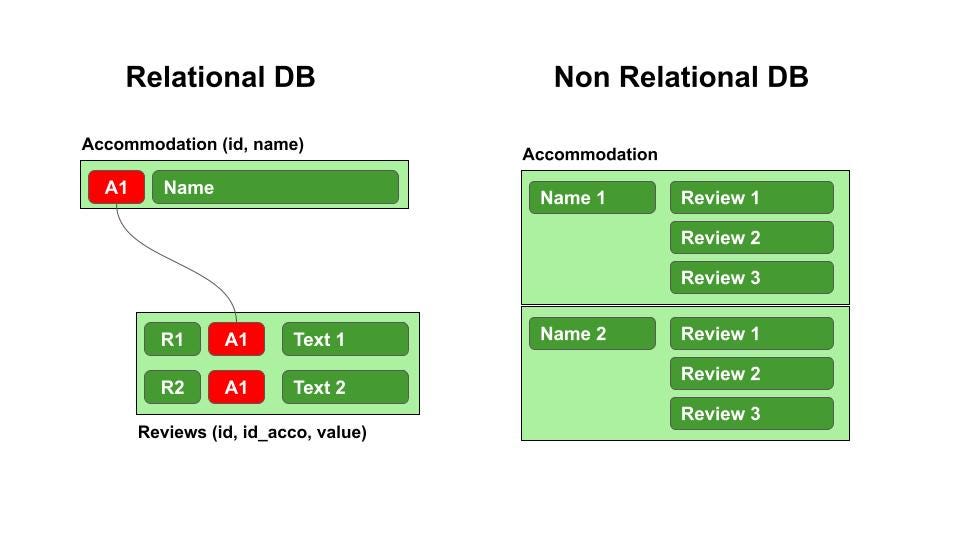
Description automatically generated

**Databázové technologie**

* “HIGH-END” systémy (Oracle DB, MSSQL server)
* “LOW-END” systémy (MySQL, postgres)
* Desktopové (Neposkytují transakční zpracování + jsou levné)

**Nerelační databáze**

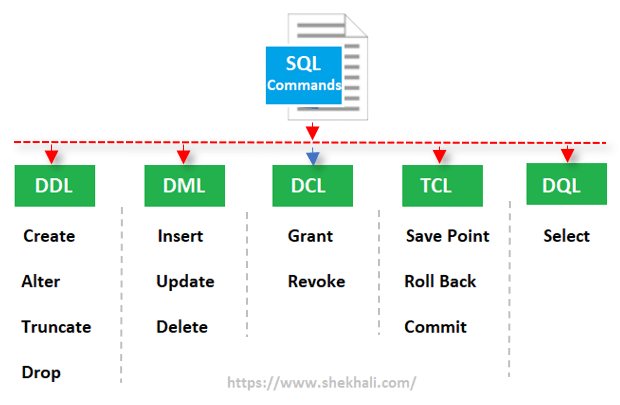
* Takové databáze, které nejsou relační (Mají PK, ale nemají vazby (RELACE))
* Například
  1. Ukládání do souboru
  2. Ukládání přihlašovacích údajů o uživateli
* **CSV, JSON** apod.



**SQL**

* Pro modifikaci databáze a dat v ni DBMS používá **SQL – Structured Query Language**, která funguje na **principu příkazů**
* Dělí se na:

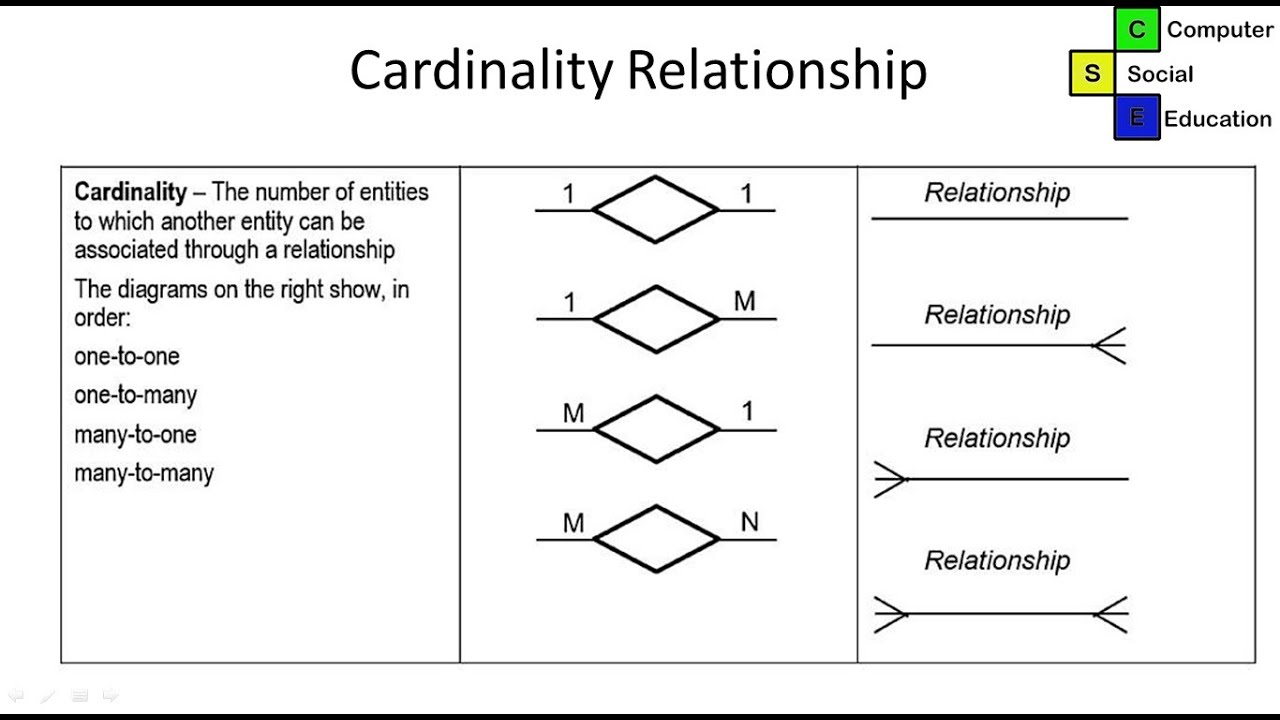
1. **DDL** – Data Definition Language  
   (Vytváří/modifikuje strukturu databáze)
2. **DML** – Data Manipulation Language  
   (Vytváří/modifikuje data v databázi)
3. **DCL** – Data Control Language  
   (Dává/ubírá příkazy uživatelům/rolí v databázi)
4. **TCL** – Transaction Control Language  
   (Stará se o vrácení kroků DML)
5. **DQL** – Data Query Language  
   (Získává data z databáze)



**Charakteristika vztahů**

* **Kardinalita**:
* Říká, kolikrát se může instance dané entity účastnit vztahu s instancemi entity jiné.

1. **1:1** - Právě jedna instance odkazuje na druhou a naopak
2. **1:N** - Tabulka může mít instance které mohou odkazovat na jednu nebo více instancí ze stejné či jiné tabulky
3. **M:N** - Tabulka může mít instance které mohou odkazovat na jednu nebo více instancí ze stejné či jiné tabulky a naopak



* **Parcialita**:
* Nastavuje povinnost nebo volitelnost vztahu

1. **Povinný**
2. **Nepovinný**

A picture containing text, font, line, screenshot

Description automatically generated

* **Stupeň vazby**:
* Říká, kolik tabulek se ve vztahu vystihuje

1. U**nární** – právě jedna entita (self-reference)

**b. Binární – dvě tabulky ve vztahu k sobě**

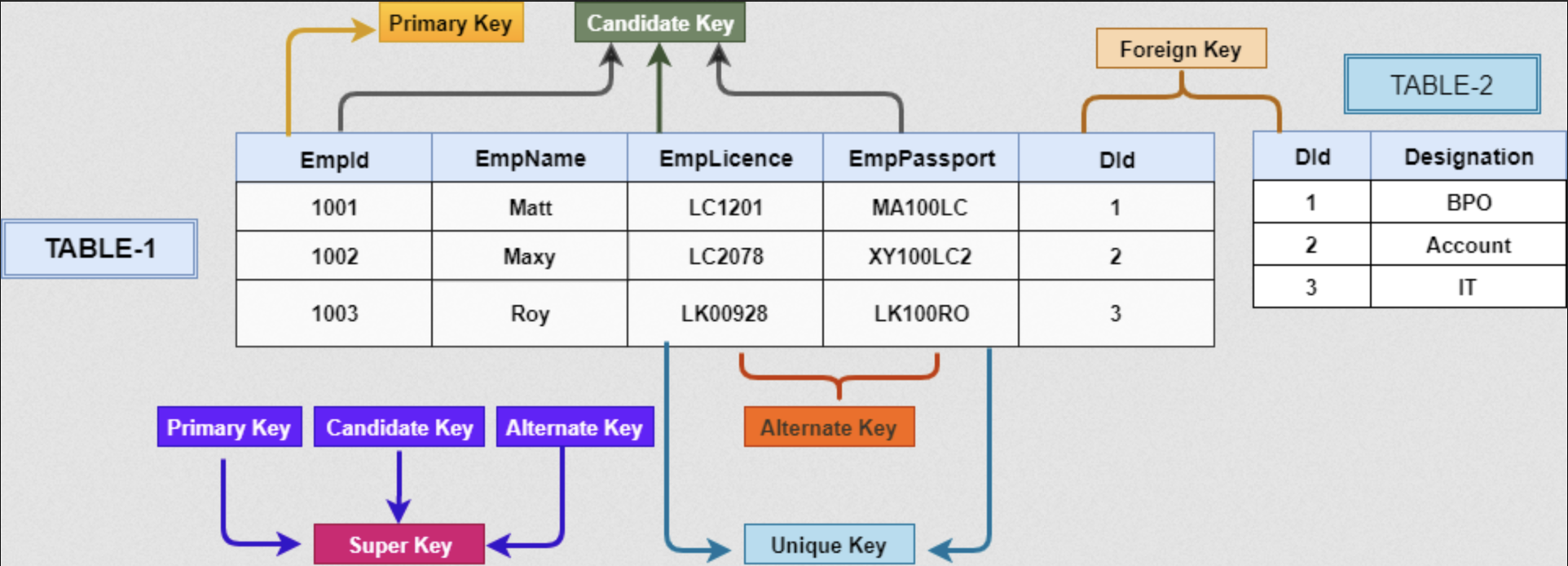
**c.N-nární** – více až *n* entit

**Klíče**

* **Kandidátní klíče**
* Unikátní klíče, které se mohou stát primárním
* **Primary klíče** **(Entitní integrita)**
* Unikátní klíč, který má každá Instance jiný
* Nemůže být prázdný => *musí existovat!*
* Díky primárnímu klíči může na jednotlivou instanci odkázat jiná instance ve stejné či jiné tabulce
* Primární klíče mohou být:

1. **Umělé** – uměle vytvořené = například ID apod.
2. **Přírodní** – atribut, který již VÍME, že je unikátní, tudíž nemusíme vytvářet umělý klíč (zápis v matrice)

* Primární klíč nemusí být složen z jediného atributu, ale i více atributů. Takový poté nazýváme **Složený**
* Alternativní klíče
* Kandidátní klíče, které se nestaly primárním
* **Cizí klíče** **(Referenční integrita)**
* Takový klíč, který v instanci odkazuje na primární klíč jiné instance stejné, či odlišné tabulky



**Ostatní databázové objekty**

* Pohledy
* Procedury
* Sestavuje složité příkazy do jedné procedury která se může repetetivně volat
* Může obsahovat i INPUTY
* Eventy
* "Procedury", které jsou časově automatizované
* Triggery

**ACID vlastnosti**

* **Atomicita** – Operace se provede celá, či vůbec
* **Konzistence** – Nesmí porušovat integritní omezení databáze
* **Izolovanost** – Vícero současně probíhajících transakcí se neovlivní
* **Trvalost** – Jakmile je transakce dokončena, je zaznamenána trvalým způsobem

