



# Databasmodeller

Relational

# Modell = HUR datan organiseras

Data lagras i tabeller med rader och kolumner

Använder SQL som frågespråk

Stark struktur med fördefinierade relationer

Exempel: MySQL, PostgreSQL, Oracle, SQLite

Bra för strukturerad data med tydliga kopplingar

# Modell : Relationell

Data organiseras i strukturerade tabeller

Varje tabell representerar en typ av objekt/information

Information kopplas samman genom relationer

# Styrkor : Relationell

Garanterar dataintegritet

Förhindrar dubletter och felaktig data

Effektiv för komplexa sökningar

Standardiserat frågespråk (SQL)

Lätt att uppdatera data på ett ställe

# Användning : Relationell

Passar företagssystem och ekonomisystem

Bra när data har tydliga kopplingar

Idealisk för transaktioner (t.ex. bank)

När data behöver vara konsistent

När struktur är viktigare än flexibilitet

# Liknelser: Relationell

Som ett Excel-ark med smarta kopplingar

Som pusselbitar som passar perfekt ihop

Som en välorganiserad katalog

Som ett system där allt har sin plats

Som en stad med tydliga adresser

# Begränsningar: Relationell

Kräver välplanerad struktur

Mindre flexibel för ändringar

Kan vara komplex för enkel data

Prestanda kan påverkas vid mycket data

# Modell : Document

NoSQL

Lagrar data i flexibla dokument (ofta JSON)

Bra för komplex och varierande data

Exempel: MongoDB, CouchDB



# Table (tabell)

Består av rader (rows) och kolumner (columns)

Varje tabell representerar en typ av objekt

Måste ha ett unikt namn i databasen

Innehåller alltid en primärnyckel

Kan kopplas till andra tabeller

# Column (kolumn)

Beskriver en specifik egenskap hos data

Måste ha en bestämd datatyp (text, nummer, datum)

Kan ha regler för vad som får lagras (constraints)

Varje kolumn måste ha unikt namn inom tabellen

# Row (rad)

Kallas även "post"

Representerar en unik instans av data

Måste följa tabellens struktur

Varje rad identifieras av primärnyckel

# Table

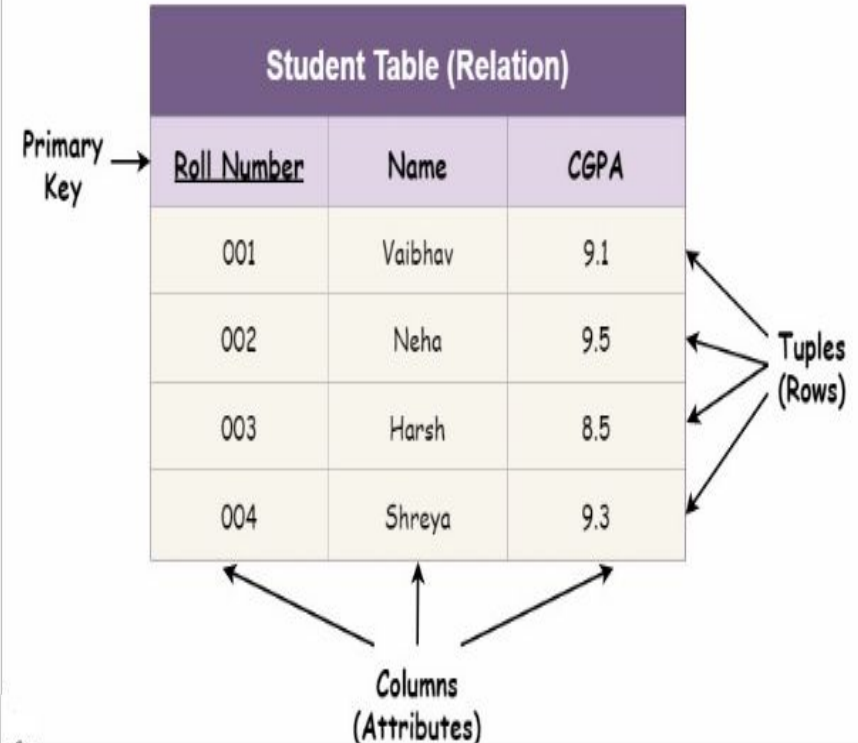
Tabellen bestämmer formen

Sätter regler för datan

Kolumnerna avgör vilken data som ska vara med

En rad i detta exempel är all data om en viss student

## Relational Model in DBMS



# Constraints

Säkerställer datakvalitet

Förhindrar felaktig data

Upprätthåller regler för verksamheten

Skyddar databasens integritet

Automatiserar datakontroll

# Constraints

NOT NULL - kräver ett värde

UNIQUE - värdet måste vara unikt

PRIMARY KEY - unik identifierare

FOREIGN KEY - koppling till andra tabeller

# Constraints

Kan kombineras för starkare regler

Aktiveras automatiskt vid dataändring

Ger felmeddelande vid regelbrott

Kan inte kringgå av användare

# Constraints

Förhindrar dubblettvärden

Garanterar dataintegritet

Förenklar felsökning

Minskar behovet av programkontroller

Skapar tydlig datastruktur



# Primary Key

Måste vara unik för varje rad

Kan inte vara null (tom)

Används för att identifiera specifika poster

Ofta ett autogenerated nummer

Används för att skapa relationer

# Foreign Key

Skapar kopplingar(relationer) mellan tabeller

Flera främmande nycklar kan finnas i samma tabell

Upprätthåller dataintegritet

Kunder		
KundID (PK)	Namn	Email
1	Anna	anna@mail.se
2	Erik	erik@mail.se
3	Maria	maria@mail.se

Ordar		
OrderID (PK)	KundID (FK)	Summa
1	1	299
2	1	499
3	2	799

### Förklaringar:

PK = Primärnyckel (Unikt ID för varje rad)

FK = Främmande nyckel (Refererar till en primärnyckel i en annan tabell)

Blå linje = Relation mellan tabeller

# Sammanfattning

**Modell** - Relational är vårt fokus, Document kan vi jämföra med

**DBMS** - SQLite ska vi använda, MySQL är ett annat. Mindre skillnader

**SQL** - Språket som används för att kommunicera med relational db

**Tabell**

**Kolumn**

**Rad**

**Primärnyckel (Primary Key)**

**Främmande nyckel (Foreign Key)**

**Datatyp** - nummer, text, osv

**Query** - **SQL** står för Structured Query Language. Queries är hur vi kommunicerar med vår databas.

# Sammanfattning

Wordgap-uppgiften på Omniway - Kurstillbehör

<https://www.w3schools.com/sql/default.asp>

[https://www.w3schools.com/sql/sql\\_exercises.asp](https://www.w3schools.com/sql/sql_exercises.asp)

<https://www.sql-practice.com/>

Vårt verktyg: <https://sqlitestudio.pl/>