

Uppgift:

- Skapa en Databas som består av Minst en **One-to-Many** relation och minst en **Many-to-Many** relation.
- Datan kan man använda olika tjänster för att skapa men det skall vara rätt typ av data och användning
- SQL Frågor som simulerar funktioner från en Applikation tex
 - Skapa användare/Kunder (Populera databasen)
 - Uppdatera data (Modifera)
 - Radera specifik data
 - Frågor som använder Limit, Offset
 - Använder sig av SQL Funktioner såsom DATE, COUNT etc
 - Fritext sökningar (LIKE)
- En JOIN som innehåller minst tre tabeller
- Skapa en View
- Rapport som använder HAVING (VG)
- En boolesk / CASE-etikett (VG)
- Stored Procedure (t.ex. registrera order) (VG)

Datamodell & Normalisering

- Motivera tabellindelning.
- Identifiera **Primary Keys**.
- Identifiera **Foreign Keys**.
- Beskriv **relationer** (1–1 / 1–M / M–M).
- Kort text om varför databasen är minst **3NF** (eller medvetna avsteg).

Vyer

Skapa minst **två vyer**:

1. En **rapportvy** (t.ex. försäljning / kursbeläggning / topplistor)
2. En **förenklad vy** för att underlätta SELECT-frågor

Relationer

Minst 1 kopplingstabell, exempelvis:

- Kunder ↔ Produkter (favoriter, beställningar)
- Studenter ↔ Kurser
- Användare ↔ Roller

En rapport med HAVING

Rapporter där **grupperade resultat filtreras**.

Exempel:

- Kunder som handlat mer än X.
- Studenter med fler än N kurser.

Indexering

- Minst ett **index** på en kolumn som ofta söks på.
- Kort motivering: "*Denna kolumn indexeras eftersom...*"

Teoretisk del – Databasutformning och motivering

I denna del ska du **redogöra för tankarna bakom din databasdesign**. Syftet är att visa att du förstår hur din lösning hänger ihop, och att du har gjort medvetna val när du skapade tabeller, relationer och datatyper.

Din text ska omfatta följande punkter:

1. Beskrivning av domänen / systemet

- Vad representerar databasen?
- Vilka användarroller finns (om relevant)?
- Vilken typ av information behöver systemet hantera?

2. Översikt av databasen

- Kort sammanfattning av vilka tabeller som finns och **vad varje tabell representerar**.
- Beskriv **vilka nycklar som används (Primary Keys)** samt **var Foreign Keys finns**.

3. Relationer och motivering

- Beskriv relationerna mellan tabellerna (t.ex. *en kund kan ha flera order*).
- Ange minst en **One-to-Many relation** och minst en **Many-to-Many relation**.
- Motivera varför du valt att separera tabeller som du gjort (ex. undvika duplicering av data, tydligare struktur eller framtida skalbarhet).

4. Normalisering

- Beskriv kort hur du säkerställt att databasen följer minst **3NF**.
- Om du har valt avsteg från normalisering, redogör för **varför** (exempel: enklare queries, prestandaskäl eller specifik funktionalitet).

5. Val av datatyper

- Motivera dina val av datatyper för centrala kolumner.
- Ge exempel på: `INT`, `VARCHAR`, `DATE`, etc., och varför dessa är lämpliga.

6. Säkerhetsaspekter

- Beskriv hur du säkerställer dataintegritet (exempel: begränsningar, `NOT NULL`, `UNIQUE`, kontroll via FK).

Slutrapport

Fil	Innehåll
schema.sql	CREATE TABLE, datatyper, Keys, Index och vyer
data.sql	Alla INSERT (Data)
queries.sql	Alla frågor enligt kravlistan med kommentar ovanför varje fråga
diagram.png	EER / ERD export
dokumentation.pdf	1–2 sidor text (datamodell + motiveringar)

Teoretiska uppgiften kan ni lämna in som PDF, MD eller docx