

Databasteknik

Agenda

- Repetition
- Idag
 - JOINS
 - GROUP BY
 - CRUD

Hierarkin i en relationsdatabas

- Programmet som hanterar relationsdatabasen kallas RDBMS
 - Relational database management system
 - Microsofts variant kallas SQL Server
 - MySQL är en open-source-variant
- RDBMS har databaser
 - Databaser har tabeller
 - Tabeller har rader/kolumner

Min lokala RDBMS

RDBMS SQL Server

Databasen sakila

Tabellen city

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'sakila' database is expanded, showing the 'city' table. The 'city' table is further expanded to show its columns: 'city_id' (PK, int, not null), 'city' (varchar(50), not null), 'country_id' (FK, smallint, not null), and 'last_update' (datetime, not null). On the right, a data table is displayed with the same columns. The table has 16 rows of data. A red line is drawn across the top of the data table, and another red line is drawn vertically along the right side of the data table.

	city_id	city	country_id	last_update
1	1	A Corua (La Corua)	87	2006-02-15 04:...
2	2	Abha	82	2006-02-15 04:...
3	3	Abu Dhabi	101	2006-02-15 04:...
4	4	Acua	60	2006-02-15 04:...
5	5	Adana	97	2006-02-15 04:...
6	6	Addis Abeba	31	2006-02-15 04:...
7	7	Aden	107	2006-02-15 04:...
8	8	Adoni	44	2006-02-15 04:...
9	9	Ahmadnagar	44	2006-02-15 04:...
10	10	Akishima	50	2006-02-15 04:...
11	11	Akron	103	2006-02-15 04:...
12	12	al-Ayn	101	2006-02-15 04:...
13	13	al-Hawiya	82	2006-02-15 04:...
14	14	al-Manama	11	2006-02-15 04:...
15	15	al-Qadarif	89	2006-02-15 04:...
16	16	al-Qatif	82	2006-02-15 04:...

Kolumner

Rader

Varför lagrar vi inte all data i en stor tabell?

Exempel: databas med användare

FirstName	LastName	PhoneNumber	Address
Fredrik	Lindroth	070-2208244	Nånstansvägen 42 42421 Sundbyberg
Patrik	Lindroth	073-2208255	Ingenstansgatan 42 42421 Örebro
Henrik	Lindroth	079-2208277	Ingenstansgatan 42 42421 Grängesberg

Varför lagrar vi inte all data i en stor tabell?

Exempel: databas med användare

- Förnamn, efternamn, telefonnummer och adress
- Hur gör vi ifall man vill att en användare ska kunna ha flera adresser eller telefonnummer?

FirstName	LastName	PhoneNumber	Address
Fredrik	Lindroth	070-2208244	Nånstansvägen 42 42421 Sundbyberg
Patrik	Lindroth	073-2208255	Ingenstansgatan 42 42421 Örebro
Henrik	Lindroth	079-2208277	Ingenstansgatan 42 42421 Grängesberg

Varför lagrar vi inte all data i en stor tabell?

Exempel: databas med användare

- Hur gör vi ifall en användare ska kunna ha flera telefonnummer?

FirstName	LastName	PhoneNumber	Address
Fredrik	Lindroth	070-2208244, 070-2208245	Nånstansvägen 42 42421 Sundbyberg
Patrik	Lindroth	073-2208255	Ingenstansgatan 42 42421 Örebro
Henrik	Lindroth	079-2208277	Ingenstansgatan 42 42421 Grängesberg

FirstName	LastName	PhoneNumber	Address
Fredrik	Lindroth	070-2208244	Nånstansvägen 42 42421 Sundbyberg
Fredrik	Lindroth	070-2208245	Nånstansvägen 42 42421 Sundbyberg
Patrik	Lindroth	073-2208255	Ingenstansgatan 42 42421 Örebro
Henrik	Lindroth	079-2208277	Ingenstansgatan 42 42421 Grängesberg

Varför lagrar vi inte all data i en stor tabell?


Exempel: databas med användare

- Det första exemplet har en lista/array på en rad
 - Går emot relationsdatabasprincipen
 - Gör det svårt att läsa ut datan
- Det andra exemplet upprepar data
 - Kallas **redundans/redundacy**
- Att förhindra upprepning av data kallas **normalisering**
 - Mer om detta i kommande föreläsningar

Istället delar vi vår data i flera tabeller

- Tabellen PhoneNumber
- Varje **rad** har ett unikt id, kallas PrimaryKey
- Varje rad har också en **relation till en annan tabell**
 - I detta fall MemberId, kallas även ForeignKey

Id	PhoneNumber	MemberId
1	070-2208244	1
2	070-2208245	1
3	073-2208255	2
4	079-2208277	3



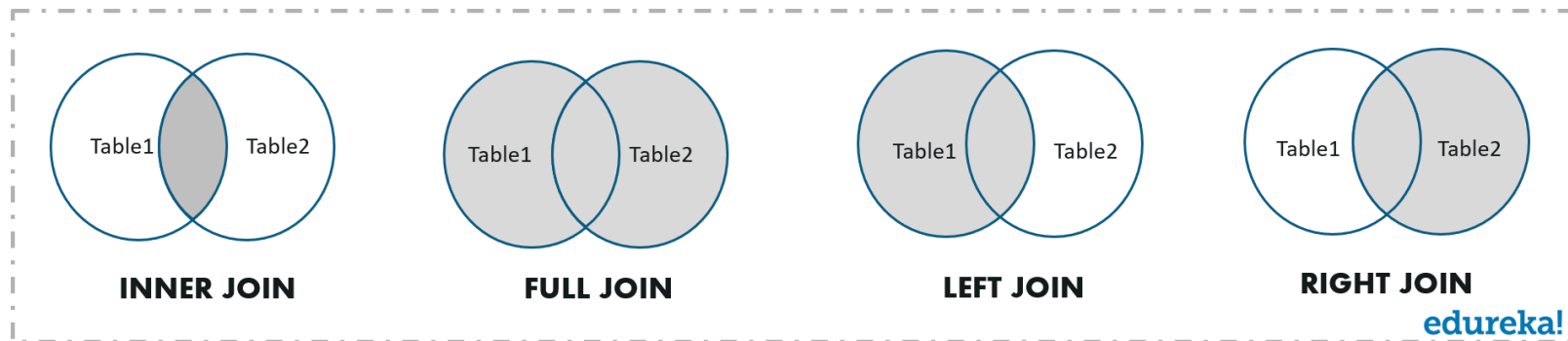
Id	FirstName	LastName	Address
1	Fredrik	Lindroth	Nånstansvägen 42 42421 Sundbyberg
2	Patrik	Lindroth	Ingenstansgatan 42 42421 Örebro
3	Henrik	Lindroth	Ingenstansgatan 42 42421 Grängesberg

Det leder oss in på JOINS

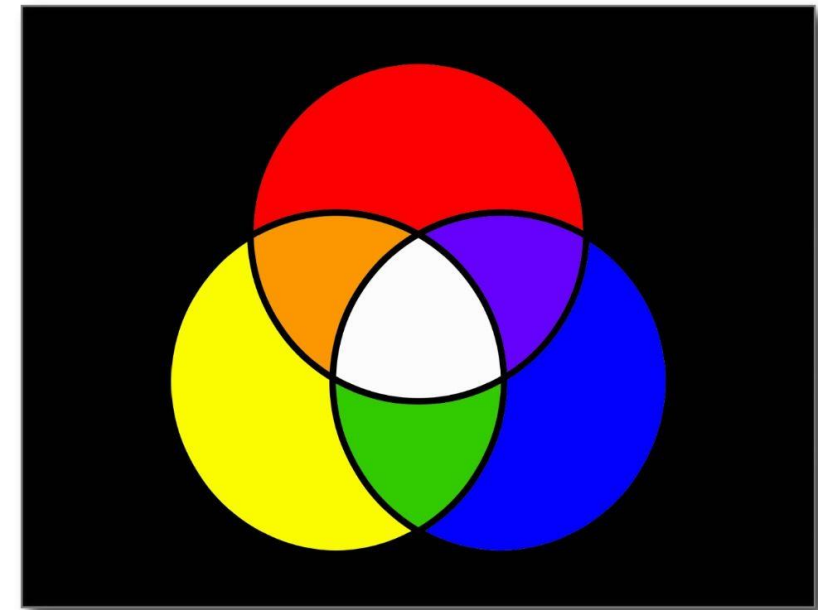
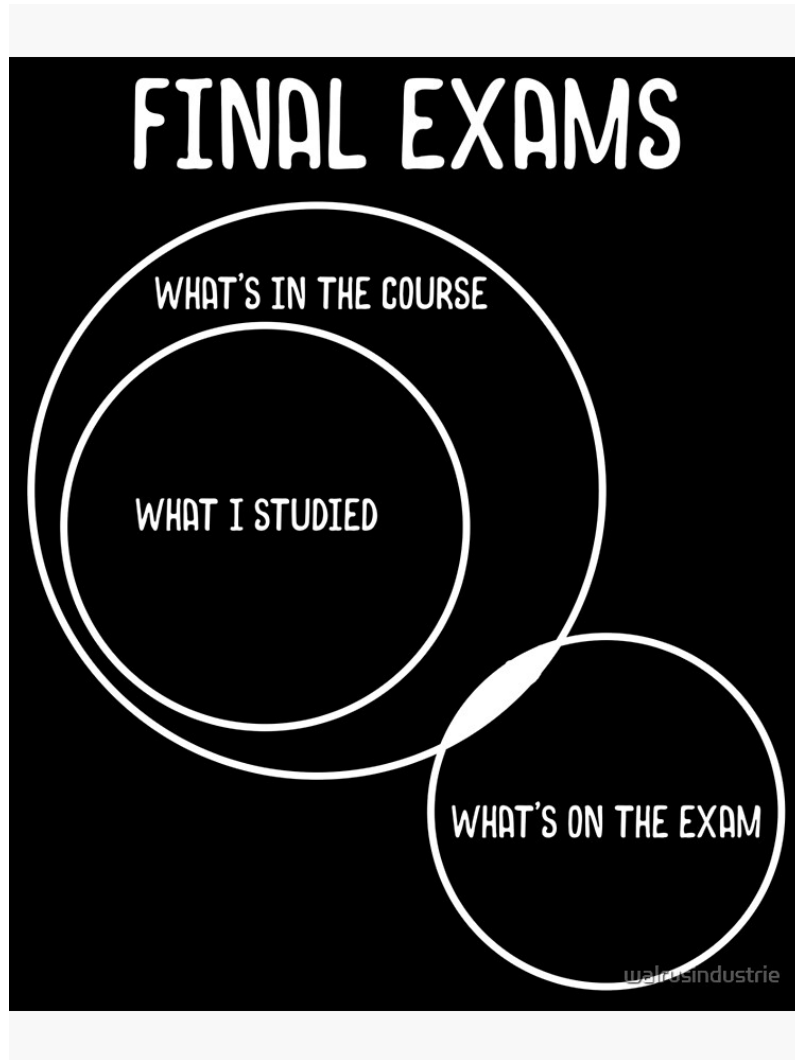
- **JOINS** är nyckelordet i SQL för att kombinera tabeller med relationer
- Två SQL-statements som ger oss två svar
 - Visa `SELECT * FROM PhoneNumber` och tabell
 - Visa `SELECT * FROM Member` och tabell

Vanliga JOINS

- Vanliga JOINS
 - JOIN (eller INNER JOIN)
 - FULL JOIN
 - LEFT JOIN
 - RIGHT JOIN
- Diagrammet kallas Venn-diagram



Venn-diagram



color venn diagram by asyrum

Zazzle

LEFT JOIN

- SQL-statement som kombinerar de två tabellerna och ger ett svar
- Här gör jag en LEFT JOIN då jag vill se alla telefonnummer

```
SELECT Member.FirstName + ' ' + Member.LastName as  
FullName, PhoneNumber.PhoneNumber FROM PhoneNumber  
LEFT JOIN Member ON PhoneNumber.MemberId = Member.MemberId
```

FirstName	PhoneNumber
Fredrik Lindroth	070-2208244
Fredrik Lindroth	070-2208245
Patrik Lindroth	073-2208255
Henrik Lindroth	079-2208277

Viktiga ord!

- RDBMS, databas, tabell, kolumn, rad
- Redundans
- Normalisering
- JOIN / INNER JOIN
- FULL JOIN
- LEFT / RIGHT JOIN

Att göra

- SQLBolt – lär dig SQL i browsern
 - Gör klart
- Övningar med Chinookdatabasen