


Föreläsning 4

Dagens föreläsning går igenom

- E/R-modellen, fortsättning
- Frågor till flera tabeller samtidigt
- Många-till-många-relationer

 Läs om E/R-diagram i kapitel 2-3 och SQL i kapitel 7 i boken "Databasteknik"

SQL-frågor till flera tabeller

För att man ska ha nytta av förhållanden mellan tabeller måste man kunna ställa SQL-frågor till flera tabeller samtidigt. Om vi tex vill veta telefonnumret till en viss person måste vi fråga både tabellen person och abonnemang.

Vi kan börja med att ställa en fråga till båda tabellerna utan ytterligare villkor:

```
SELECT * FROM person, abonnemang
```

Resultat blir en *produkt* av tabellerna person och abonnemang

Med produkt i detta sammanhang menas att varje rad i en tabellen slås ihop med varje rad i andra tabellen. Om person har 2 rader och abonnemang har 5 rader kommer den resulterande tabellen att få $2 \times 5 = 10$ rader:

Din sql-fråga `select * from person,abonnemang` gav följande resultat:

namn	personnummer	nummer	operator	personnummer
Bo Peterson	6502019999	046-123456	Tele 2	6502019999
Sven Svensson	7312125555	046-123456	Tele 2	6502019999
Bo Peterson	6502019999	040-654321	Telia	7312125555
Sven Svensson	7312125555	040-654321	Telia	7312125555
Bo Peterson	6502019999	040-6657000	Telia	6502019999
Sven Svensson	7312125555	040-6657000	Telia	6502019999
Bo Peterson	6502019999	0739-654321	Comviq	7312125555
Sven Svensson	7312125555	0739-654321	Comviq	7312125555
Bo Peterson	6502019999	070-654321	Vodafone	7312125555
Sven Svensson	7312125555	070-654321	Vodafone	7312125555

Detta resultat är inte särskilt användbart, men blir det plötsligt om vi tillfogar villkor på nyckel-kolumnerna. Vi kan börjamed att titta på alla rader (gulmarkerade ovan) där primärnyckeln i person är lika med främmande nyckeln i abonnemang.

Bo Peterson

Vi kan välja ut de raderna med följande SQL-sats:

```
SELECT * FROM person INNER JOIN abonnemang  
ON person.personnummer=abonnemang.personnummer;
```

namn	personnummer	nummer	operator	personnummer
Bo Peterson	6502019999	046-123456	Tele 2	6502019999
Sven Svensson	7312125555	040-654321	Telia	7312125555
Bo Peterson	6502019999	040-6657000	Telia	6502019999
Sven Svensson	7312125555	0739-654321	Comviq	7312125555
Sven Svensson	7312125555	070-654321	Vodafone	7312125555

Observera att vi anger person.personnummer och abonnemang.personnummer för att särskilja kolumner i olika tabeller men med samma namn. Vi har nu fått en förteckning över alla telefonnummer. Denna operation brukar man kalla INNER JOIN

Om vi vill ha alla telefonnummer till Sven Svensson lägger vi till ett villkor:

```
SELECT * FROM person INNER JOIN abonnemang  
ON person.personnummer=abonnemang.personnummer  
WHERE namn='Sven Svensson'
```

namn	personnummer	nummer	operator	personnummer
Sven Svensson	7312125555	040-654321	Telia	7312125555
Sven Svensson	7312125555	0739-654321	Comviq	7312125555
Sven Svensson	7312125555	070-654321	Vodafone	7312125555

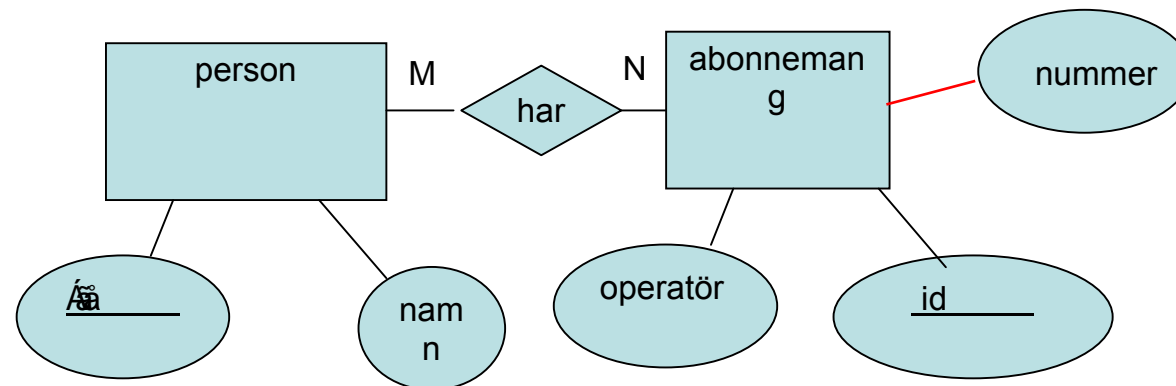
Om vi bara är intresserade av namn, nummer och operatör byter vi ut stjärnan mot de kolumner vi vill visa:

```
SELECT namn,nummer,operator FROM person  
INNER JOIN abonnemang  
ON person.personnummer=abonnemang.personnummer  
WHERE namn='Sven Svensson'
```

namn	nummer	operator
Sven Svensson	040-654321	Telia
Sven Svensson	0739-654321	Comviq
Sven Svensson	070-654321	Vodaphone

Mer om ER-diagram och många-till-många-relationer

Vi utgår från ett tidigare exempel med person och abonnemang men utökar nu till en många-till-många-relation så att varje person kan ha flera abonnemang och ett abonnemang kan delas av flera personer.



Hur översätter man detta till tabeller i en relationsdatabas?

Vi blir nu tvungna att införa en *mellantabell* som knyter samman person och abonnemang.

Vi inför en mellantabell *personabonnemang* med primärnycklarna från både person och abonnemang.

Tabellstrukturen blir:

person

<u>namn</u>	<u>id</u>
Bo Eriksson	67
Sven Svensson	73
Anna Eriksson	69

abonnemang

<u>id</u>	operatör	nummer
F	Tele 2	0800 123456789
A	Telia	0800 123456789
AH	Telia	0800 123456789

personabonnemang

<u>personid</u>	<u>abonnemangid</u>
67	1
67	2
73	3
69	1

I personabonnemang kan båda kolumnerna innehålla dubletter. Ingen kolumn kan då ensam vara primärnyckel. De två kolumnerna utgör tillsammans primärnyckeln!

SQL-exempel, många-till-många

För att välja få en lista över alla telefonnummer och namn kan vi skriva

```
SELECT person.namn,abonnemang.nummer FROM person
INNER JOIN personabonnemang
ON person.id=personid
INNER JOIN abonnemang
ON abonnemang.id=abonnemangid
```

För att få alla nummer till Bo Eriksson utökar vi frågan:

```
SELECT person.namn,abonnemang.nummer FROM person
INNER JOIN personabonnemang
ON person.id=personid
INNER JOIN abonnemang
ON abonnemang.id=abonnemangid
WHERE namn='Bo Eriksson'
```

Bo Peterson