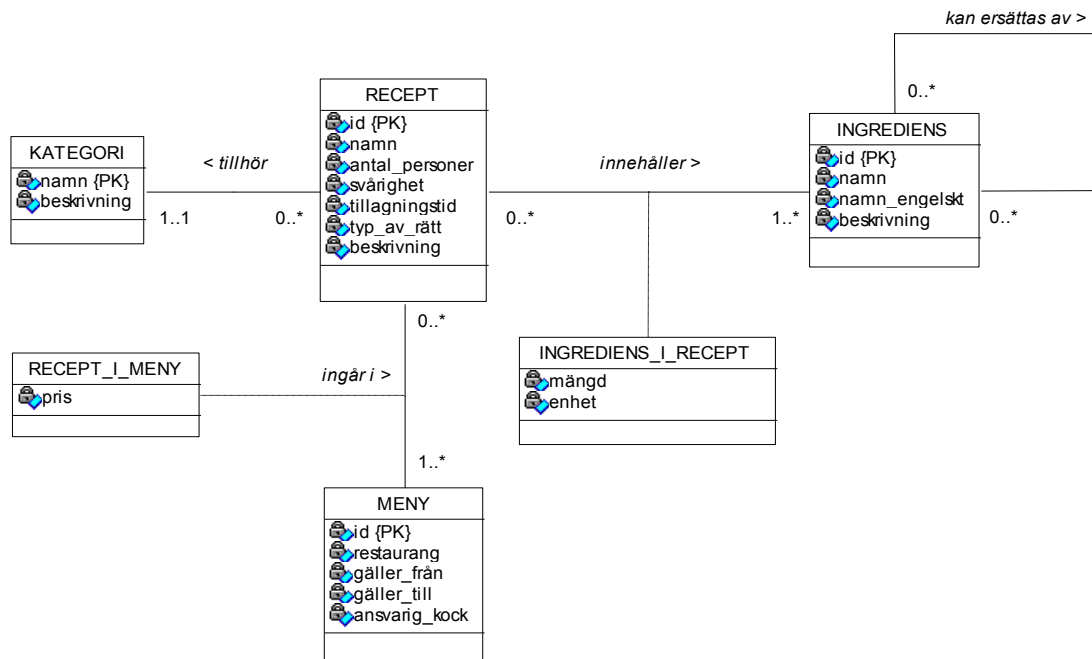


## Uppgift 1: Datamodellering (20 p)

### Receptdatabas

#### a) ER-modell



Det mest kluriga i denna uppgift är nog att ingredienser kan ersättas av andra ingredienser. Smidigaste lösningen är här ett rekursivt samband. En ingrediens kan ersättas av ingen, en eller flera andra ingredienser och en ingrediens kan i sin tur vara "ersättningsingrediens" för ingen, en eller flera ingredienser. I övrigt ska det inte ha varit några problem. Två många-till-många-samband och ett ett-till-många. En del av attributen jag har i min lösning kan vi istället låta vara egna entitetstyper. T.ex KOCK som är ansvarig för en meny. RESTAURANG som erbjuder meny. SVÅRIGHETS\_GRAD och TYP\_AV\_RÄTT kan även vara egna entitetstyper.

#### b) Relationsmodell

KATEGORI (namn, beskrivning)

RECEPT (id, namn, antal\_personer, svårighet, tillagningstid, typ\_av\_rätt, beskrivning, namn\*)

MENY (id, restaurang, gäller\_från, gäller\_till, ansvarig\_kock)

INGREDIENS (id, namn, namn\_engelskt, beskrivning)

INGREDIENS\_I\_RECEPT (recept\_id\*, ingrediens\_id\* mängd, enhet)

RECEPT\_I\_MENY (recept\_id\*, meny\_id\*, pris)

ERSÄTTTS\_AV (ingrediens\_id\*, ersätts\_av ingrediens\_id\*)

## Uppgift 2: SQL (15 p)

Utgå ifrån följande modell för en relationsdatabas (primärnycklar är understrukna) och svara på uppgifterna nedan.

```
FILM_STJÄRNA(s_namn, s_adress, s_födelsdatum, s_kön)
REGISSÖR(r_namn, r_adress, r_födelsdatum, r_kön)
FILM_STUDIO(studio_namn, studio_adress)
FILM_GENRE(genre_namn, beskrivning)
FILM(film_titel, film_år, längd, färg, omsättning, studio_namn, genre_namn)
STJÄRNA_I_FILM(film_titel, film_år, s_namn)
REGISSÖR_AV_FILM(film_titel, film_år, r_namn)
```

a)

```
FOREIGN KEY(film.studio) REFERENCES film_studio(studio_namn)
FOREIGN KEY(film.genre_namn) REFERENCES film_genre(genre_namn)
FOREIGN KEY(stjärna_i_film.film_titel) REFERENCES film(film_titel)
FOREIGN KEY(stjärna_i_film.film_år) REFERENCES film(film_år)
FOREIGN KEY(stjärna_i_film.s_namn) REFERENCES film_stjärna(s_namn)
FOREIGN KEY(regissör_av_film.film_titel) REFERENCES film(film_titel)
FOREIGN KEY(regissör_av_film.film_år) REFERENCES film(film_år)
FOREIGN KEY(regissör_av_film.s_namn) REFERENCES regissör(r_namn)
```

b)

```
STEG 1.
ALTER TABLE film_stjärna DROP CONSTRAINT fs_pk;

STEG 2.
ALTER TABLE film_stjärna ADD CONSTRAINT fs_pk PRIMARY KEY(s_namn);
```

c)

```
SELECT film_titel, film_år, omsättning
FROM FILM
WHERE film_år = 2004
AND omsättning BETWEEN 1000000 AND 10000000;
```

### Alternativ

```
SELECT film_titel, film_år, omsättning
FROM FILM
WHERE film_år = 2004
AND omsättning >= 1000000 AND omsättning <= 10000000;
```

d)

```
SELECT film.genre_namn, beskrivning, COUNT(*) AS antal
FROM film_genre, film
WHERE film_genre.genre_namn = film.genre_namn
GROUP BY film.genre_namn, beskrivning;
```

Observera att det inte enbart räcker att gruppera efter genre\_namn utan även beskrivning eftersom båda dessa ska synas i resultatet!

e)

```
SELECT r.r_namn, r.födelsedatum, f.film_titel, längd, genre_namn
FROM film AS f, regissör AS r, film_stjärna AS fs,
     stjärna_i_film AS sif, regissör_av_film AS raf
WHERE f.film_titel = sif.film_titel AND f.film_titel = raf.film_titel
     AND f.film_år = sif.film_år AND f.film_år = raf.film_år
     AND r.r_namn = raf.r_namn
     AND fs.s_namn = sif.s_namn
     AND r.r_födelsedatum < fs.s_födelsedatum
ORDER BY r.födelsedatum ASC;
```

Det svåra här är egentligen enbart att komma ihåg alla AND för att ”koppla” samman alla tabeller.

## Uppgift 3: Relationsalgebra (5 p)

När du skriver dina relationsalgebrauttryck använd följande symboler eller alternativa skrivsätt:

Operation	Symbol	Alternativ
Projektion	$\pi$	PROJECT
Selektion	$\sigma$	SELECT
Tilldelning	$\leftarrow$	=
Kartesisk produkt	$\times$	X

Operation	Symbol	Alternativ
Union	$\cup$	UNION
Snitt	$\cap$	INTERSECTION
Döpa om	$\rho$	RENAME
Join	$\bowtie$	JOIN

- a)  $\pi_{s\_namn}(\text{FILM\_STJÄRNA})$
- b)  $\pi_{\text{film\_titel}, \text{film\_år}, \text{omsättning}}(\sigma_{\text{omsättning} < 10000000 \text{ and } \text{omsättning} > 1000000 \text{ and } \text{film\_år} = '2004'}(\text{FILM}))$
- c)  $R \leftarrow \rho_R(\text{REGISSÖR\_AV\_FILM})$   
 $F \leftarrow \rho_F(\text{FILM})$   
 $RF \leftarrow \sigma_{r.\text{film\_titel} = f.\text{film\_titel} \text{ and } r.\text{film\_år} = f.\text{film\_år}}(R \times F)$   
 $\pi_{r\_namn}(\sigma_{f.\text{film\_år} = '2010' \text{ and } f.\text{film\_titel} = 'Inception'}(RF))$

## Uppgift 4: Normalisering (5 p)

Givet är följande tabell:

LOKALBOKNING (salsnr, start, slut, hus, beskrivning, platser, signatur, namn)

### Uppgifter:

a)

De funktionella beroenden som existerar är följande:

1. **salsnr, start, hus** → slut, beskrivning, platser, signatur, namn (*primär*)
2. **salsnr, hus** → platser (*partiell*)
3. **hus** → beskrivning (*partiell*)
4. **signatur** → namn (*transitiv*)

Första funktionella beroendet är given då primärnyckeln alltid determinerar övriga attribut i tabellen.

Andra är en partiell determinering. Vet vi vilket hus vi befinner oss i och i vilken sal vi sitter vi så vet vi antal platser den salen har. Med enbart salsnr kan vi inte peka ut antalet platser eftersom samma salsnr kan finnas i flera hus.

Även tredje är en partiellt determinering. Vet vi vilket huset är vet vi vilken typ av aktivitet som bedrivs där (beskrivning).

Den fjärde är en transitiv determinering. Signaturen pekar ut vilken person det är (namn). Observera att tvärtom inte gäller eftersom det finns flera personer med samma namn (men med olika signaturer).

b)

Den högsta normalform som tabellen LOKALBOKNING uppfyller är 1NF.

Anledningen är att vi har partiella determineringar som bryter mot 2NF. T.ex hus (hus är en del av kandidatnyckeln salsnr, start, hus) som determinerar beskrivning.

c)

Vi börjar med att åtgärda första partiella beroendet (salsnr, hus → platser) genom att skapa en ny tabell för salarna. Kvar i LOKALBOKNING lämnar vi salsnr, hus som främmandenycklar till den nya tabellen.

LOKALBOKNING (salsnr\*, start, slut, hus\*, beskrivning, signatur, namn)  
SALAR (salsnr, hus, platser)

Den nya tabellen SALAR är nu i BCNF eftersom enda determinanten är en kandidatnyckel. I LOKALBOKNING har vi fortfarande ett partiellt beroende (hus → beskrivning). Vi skapar ytterligare en tabell för husen och flyttar dessa data till den tabellen. Kvar i LOKALBOKNING lämnar vi hus som en främmandenyckel.

LOKALBOKNING (salsnr\*, start, slut, hus\*, signatur, namn)

---

SALAR            (salsnr, hus, platser)  
HUS             (hus, beskrivning)

Kvar återstår det transitiva beroendet (signatur → namn) i ORDER. Vi skapar en ny tabell för alla personer.

LOKALBOKNING (salsnr\*, start, slut, hus\*, signatur\*)  
SALAR            (salsnr, hus, platser)  
HUS             (hus, beskrivning)  
PERSONER       (signatur, namn)

Tabellerna LOKALBOKNING, SALAR, HUS och PERSONER är alla nu i BCNF eftersom de determineringar som existerar i dessa tabeller är kandidatnycklar.