



Försättsblad till skriftlig
tentamen vid Linköpings
universitet



Datum för tentamen	2016-01-19
Sal (1)	<u>TER3</u>
Tid	14-18
Kurskod	729G28
Provkod	TEN1
Kursnamn/benämning Provnamn/benämning	Webprogrammering och databaser Tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	6
Jour/Kursansvarig Ange vem som besöker salen	Erik Prytz
Telefon under skrivtiden	070-2762748
Besöker salen ca klockan	ca kl. 14:45
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Annelie Almquist 013-28 29 annelie.almquist@liu.se
Tillåtna hjälpmedel	Inga hjälpmedel tillåtna.
Övrigt	Instruktioner: Skriv läsligt. Visa ditt arbete när så krävs av uppgiften. Läs igenom tentan innan du börjar, så att du kan fråga jourhavande lärare om det finns oklarheter. Använd ett papper per fråga. Totalt antal poäng: 23 Poäng för G: 11,5 Poäng för VG: 17,5 Lycka till!
Antal exemplar i påsen	

1. **Begrepp:**

- a. **(1p) Flervalsfråga, välj endast ett alternativ:** I databassammanhang skiljer man på *instans* och *schema*. Vad är en instans av en databas?
- A: En beskrivning av de tabeller som kan finnas i databasen
 - B: En tabell bestående av en mängd tupler i databasen
 - C: Den ordnade listan av attributvärden i databasen
 - D: Den samling data som just nu finns i databasen
- b. **(1p) Flervalsfråga, välj endast ett alternativ:** I ER-modellen, vad är attribut?
- A: Kopplingar mellan olika koncept (typer av saker) i modellen
 - B: De koncept (typer av saker) som finns i modellen
 - C: Specifika instanser av koncept (typer av saker) som finns i modellen
 - D: De egenskaper som koncept (typer av saker) i modellen har
- c. **(2p) Kort svar:** Vilka tre nivåer ingår i den så kallade tre-schema-arkitekturen, och vad innebär varje nivå? (Svara med 1-2 meningar per nivå)

2. (4p) **ER-Design:** Nystartade företaget FilmNät, som har den unika idén att hyra ut digitala filmer till användare *över internet(!)*, vill ha en databas för att hantera sin verksamhet. De har anlitat dig för att ta fram ett designförslag på hur den nya databasen ska se ut. I detta skede vill de alltså ha ett ER-diagram som visar en lämplig lösning utifrån deras kravspecifikation.

-Det första FilmNät vill kunna lagra i sin databas är förstås de filmer som de har tillgång till och kan hyra ut. De vill spara titeln på filmen, hur många minuter filmen är, utgivningsår, samt vilket land filmen är producerad i. De vill även lagra skådespelare, med namn, ålder och skådis-id (ett unikt nummer för alla skådespelare i databasen). Eftersom samma skådis kan ha varit med i flera olika filmer, och de flesta filmer har mer än en skådis vill de kunna spara den informationen separat från filmernas information. Alla filmer har förstås minst en skådespelare.

-FilmNät vill också spara vem som regisserat filmen. De vill ha namn och unikt id på regissörerna. Varje film måste ha en och endast en regissör. Fast samma regissör kan ju ha regisserat flera filmer, förstås.

-Till sist vill FilmNät även spara data om användarna. Användarna måste ha unika användarnamn samt ett lösenord för att logga in på FilmNät. Användarens ort ska också sparas, samt användarens telefonnummer. En användare kan förstås hyra filmer. När en användare hyr en film ska information om vilket datum som uthyrningen sker (uthyrningen gäller bara i 7 dagar) samt till vilket pris (samma film kan ibland ha olika pris i samband med specialerbjudanden och så vidare). Varje användare är också automatiskt med i FilmNäts bonusprogram, PlusFilm. Varje gång en användare hyr en film får de ett antal PlusPoäng och det ska också sparas i databasen. FilmNät vill kunna söka fram det totala antalet poäng för varje användare också, för att ge specialerbjudanden. En användare kan hyra så många filmer de vill, och eftersom filmerna är digitala har inte FilmNät några fysiska kopior av filmerna och kan således hyra ut samma film till hur många som helst samtidigt.

Rita ett ER-diagram baserat på denna kravspecifikation. Markera nycklar, deltagande, samt kardinalitet där det behövs. Du behöver **inte** översätta ER-diagrammet till relationstabeller. Om information saknas eller kravspecifikationen är för vag i något avseende (ex. rörande unika värden för nycklar, deltagande, eller kardinalitet) när du bestämmer din design måste du specificera de antaganden du gör.

3. (4p) **ER till Relation:** Ett nytt rapporteringssystem för folkhälsa planeras i Sverige. Socialstyrelsen (SOS) har redan designat ett ER-diagram över hur de vill att databasen ska vara strukturerad, men har inte lyckats översätta ER-diagrammet till en relationsdatabas. Hjälp SOS genom att översätta diagrammet till en lämplig relationsdatabas i enlighet med översättningsstegen beskrivna i kursboken och/eller föreläsningarna.

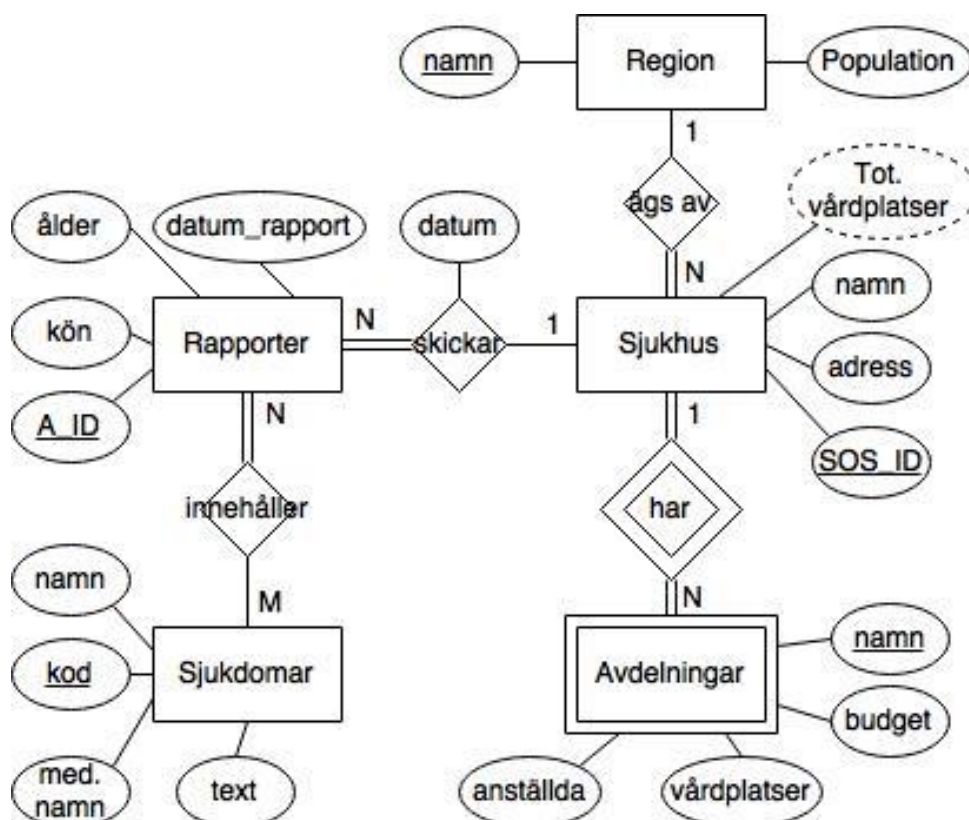
För denna fråga behöver du **inte** formellt normalisera relationerna, men bör ändå sträva mot en enkel och icke-redundant lösning. Skriv ut relationerna enligt exempelformatet nedan och stryk under alla primärnycklar. Skriv även separat alla främmande nycklar med pilar till de primärnycklar dessa refererar till. Skriv ut de antaganden du gör, om några.

Exempelformat för relationerna:

Relationsnamn(Attribut 1, Attribut 2, Attribut 3)

Exempelformat för främmande nycklar:

RelationEtt(FrämmandeNyckel) → RelationTvå(Attribut).



4. (4p) **Normalisering:** En klandig databasdesigner har implementerat en dåligt designad relationsdatabas åt affärskedjan Freudian Slippers, som säljer tofflor med roliga motivtryck. Nu är det upp till dig att få lite fason och ordning på denna databas. Den dåliga databasen består av en enda stor tabell:

BUTIK(AnställningsID, TransaktionsID, VaruID, Förnamn, Färg, Efternamn, Timmar, Adress, Pris, Märke, Datum, Storlek, Butiksnamn, Lagerantal, Timlön, Antal, Tryck)

I den ligger alltså all information, från information om de anställda (som deras namn och lön), till butikerna med sina namn och adresser, lagerstatus, till alla transaktioner (försäljningar) och så vidare. Du vet att följande fullt funktionella beroenden (FFB) finns:

{AnställningsID} → {Förnamn, Efternamn, Timlön}
 {AnställningsID, Butiksnamn} → {Timmar}
 {Butiksnamn} → {Adress}
 {Butiksnamn, VaruID} → {Pris, Lagerantal}
 {VaruID, TransaktionsID} → {Antal}
 {TransaktionsID} → {Datum, Butiksnamn}
 {VaruID} → {Märke, Färg, Tryck, Storlek}

Normalisera tabellen BUTIK så att den uppfyller Boyce-Codd Normalform (BCNF). Skriv ut relationer inklusive attribut. Lista alla främmande nycklar (se fråga 7 för exempelformat). Motivera varför din lösning uppfyller varje normalform. Ange eventuella antaganden du gör.

5. (3p) **Integritet:** En klantig databasdesigner har problem med en databas för ett rederi och kan inte förstå varför det är fel. Hen försöker köra följande antal kommandon, men vissa lyckas inte:

Rad	Kommando
1	Insert into Rutt values (1, 'Skutskär', 'Öland', 'FALSE');
2	Insert into Rutt values (4, 'Skutskär', 'Borneo', 'TRUE');
3	Insert into Rutt values (2, 'Öland', 'Borneo', 'TRUE');
4	Insert into Färja values (1, 'Eken', 200, 3);
5	Insert into Färja values (2, 'Lången', 1000, 4);
6	Insert into Färja values (4, 'Eken', 50, 4);
7	Insert into Färja values (2, 'Öland', 0, 2);

Du ber att få se SQL-koden som användes för att skapa de relationer som kommandona arbetar mot i databasen, och får då detta:

```
create table Rutt(
    Rutt_ID int not null primary key,
    Start varchar(255) not null,
    Mål varchar(255) not null,
    Taxfree boolean default FALSE);

create table Färja(
    Skepp_ID int not null,
    Namn varchar(255),
    Max_Antal_Passagerare int,
    Rutt int,
    primary key (Skepp_ID),
    foreign key (Rutt) references Rutt (Rutt_ID));
```

Några av de kommandon som databasdesignern försöker köra bryter mot något av de två olika integritetsvillkor. Vad heter de två integritetsvillkoren, vad innebär de, och vilka kommandon (ange rad) bryter mot vilket kommando på vilket sätt?

6. (4p) **SQL:**

- a. SQL är både ett DDL och ett DML. Vad står dessa förkortningar för, och vad innebär de två begreppen?
- b. Den frustrerade databasdesignern från fråga 5 har äntligen fått kläm på sina tidigare problem och har laddat in en hiskelig massa (korrekt) data i databasen. Nu försöker hen fråga databasen hur många färjor som trafikerar rutt 8 och där färjans namn börjar på "H" och kan ta mer än 150 passagerare...men det vill sig inte. Ta en titt på hens SQL-fråga och rätta till den så att den fungerar. Det är samma databas som i fråga 5, så titta på create table-koden från den frågan för tabeller, kolumner, nycklar, etc. Hint: det finns flera fel i nedanstående kod...

```
select Max(Skepp_ID) as Antal_Skepp from Färja
join Rutt as R on Färja.Skepp_ID = R.Rutt_ID
where Rutt_ID = 9 and Färja.Namn = "H%" and
Färja.Max_Antal_Passagerare>150;
```

- c. Vilka av följande SQL-satser manipulerar databasinstansen och vilka manipulerar databasschemat?
 1. Insert into Färja values(9, 'Oxe', 20, 5);
 2. Alter table Färja drop Namn;
 3. Create table Kapten(id int primary key, namn varchar(255));
 4. Delete from Färja where Skepp_ID = 10;