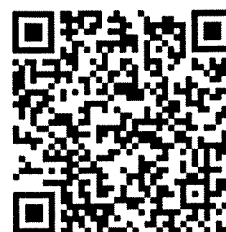




Försättsblad till skriftlig tentamen vid Linköpings Universitet



Datum för tentamen	2015-01-08
Sal (2)	<u>TER3</u> TER4
Tid	8-12
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. 09:00
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, anna.grabska.eklund@liu.se, ankn. 2362
Tillåtna hjälpmedel	inga
Övrigt	
Antal exemplar i påsen	

1 (6)

Linköpings Universitet
Institutionen för Datavetenskap
Erik Prytz

TENTAMEN

729G28

Datum: 2015-01-08
Jourhavande lärare: Erik Prytz
Telefon: 070-2762748
Besöker tentamenslokalen c:a kl. 09:00.

Poängfördelning

Fråga	Poäng
1	4
2	8
3	8
4	4
5	2
6	4

Totalt antal poäng: 30
Poäng för G: 15
Poäng för VG: 22

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.

Lycka till!

2 (6)

1. (4p) **Begrepp:**

- a. **Flervalsfråga, välj endast ett alternativ:** Vilket av följande alternativ är ett exempel på en implementationsmodell (även kallat representationsmodell)?
- A: ER-modellen
 - B: Konceptuella modellen
 - C: Relationsmodellen
 - D: Rollmodellen
- b. **Flervalsfråga, välj endast ett alternativ:** I ER-modellen, vad är entiteter?
- 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. **Begrepp:** I databas-sammanhang brukar man skilja på databas-instanser och databas-scheman. Förklara kort vad dessa två begrepp innebär och vad som skiljer dem åt.

2. (8p) **ER-Design:** Nystartade företaget FilmNät, som har den unika idén att hyra ut film 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 förstas 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 förstas 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, förstas.

-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 förstas 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.

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.

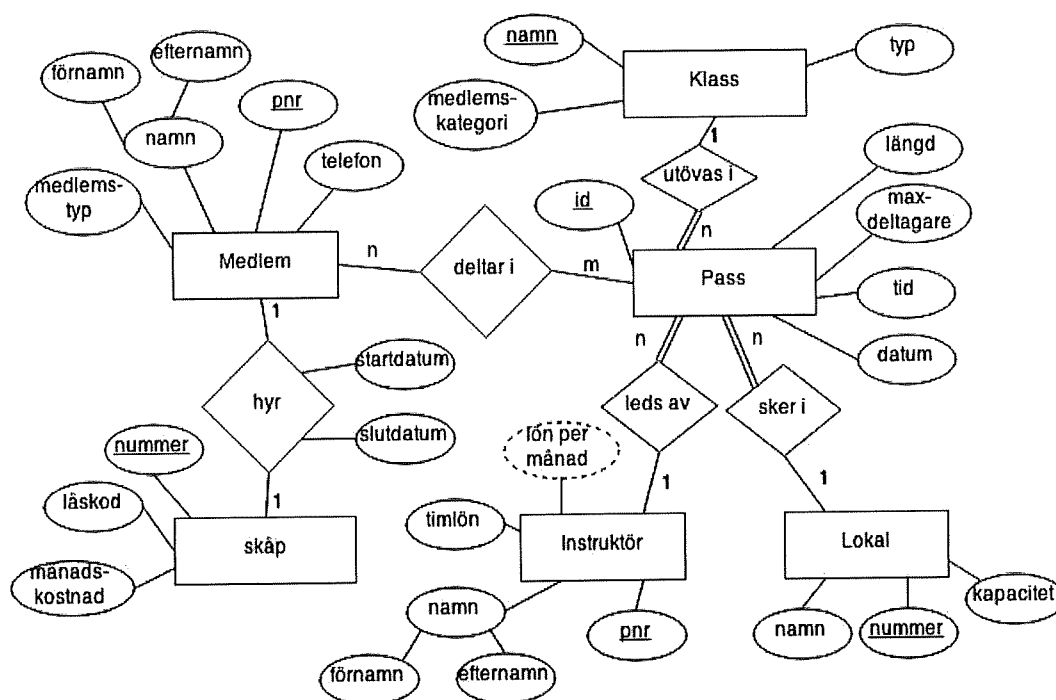
3. (8p) **ER till Relation:** Ett lokalt gym som drivs av Hampus och Calle, HampusCallen, vill ha en ny databas. De har redan designat ett ER-diagram över hur de vill att databasen ska vara strukturerad, men har inte lyckats översätta ER-diagrammet till en relationsdatabas. Hjälp Hampus och Calle genom 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 företaget ACS för att hantera deras anställda och lönebonusprogram (framförallt tjänstefordon). I dagsläget ser databasen ut så här:

HELAFÖRETAGET(personnummer, avdnr, fordons-id, förnamn, efternamn, stad, fordonstyp, lönekategori, avdelningsnamn, tillverkningsår, modell, tjänstefordonsklass, leasingdatum, semesterdagar)

Du vet att följande fullt funktionella beroenden (FFB) finns:

{personnummer} → {förnamn, efternamn, lönekategori, avdnr}
{avdnr} → {avdelningsnamn, stad}
{Fordonstyp} → {Tjänstefordonsklass}
{personnummer, fordons-id} → {leasingdatum}
{fordons-id} → {fordonstyp, modell, tillverkningsår}
{lönekategori} → {semesterdagar, tjänstefordonsklass}

Normalisera tabellen HELAFÖRETAGET så att den uppfyller Boyce-Codd Normalform (BCNF). Ange eventuella antaganden du gör.

5. (2p) **Integritet:** Man brukar tala om två typer av villkor som man måste ta hänsyn till för databasens integritet. Vad kallas dessa två villkorstyper, och vad innebär de?
6. (4p) **SQL:** En databas över en skobutiksfirma ser ut som följer:

Anställd(anstID, förnamn, efternamn, telefon, timlön)
 Butik(namn, adress, chef, startdatum)
 Vara(varuID, märke, modell)
 Kund(bonusprogramID, skostorlek)
 JobbarI(anställd, butik, timmar)
 Lager(butik, vara, antal, pris)
 Köp(kund, butik, vara, datum, antal)

Primärnycklarna är understrukna, och det finns även följande främmande nycklar:

JobbarI(anställd) → Anställd(AnstID)
 JobbarI(butik) → Butik(namn)
 Butik(chef) → Anställd(AnstID)
 Lager(butik) → Butik(namn)
 Lager(vara) → Vara(varuID)
 Köp(kund) → Kund(BonusprogramID)
 Köp(butik) → Butik(namn)
 Köp(vara) → Vara(varuID)

Skriv ned de SQL-frågor som behövs för att besvara de tre nedanstående frågorna. Om du tycker det saknas information så att du behöver göra antaganden ange då samtliga sådana antaganden du gör.

- Vilka kunder har en skostorlek större än 30? I den tabell som är resultatet vill jag dessutom att kolumnen med dessa kunder heter StorKund.
- Vad är den totala lönekostnaden per vecka för de som är butikschefer? (antag att jobbari.timmar är antal timmar per vecka)
- Skriv ut alla butiker och deras genomsnittliga vinst per köp. (Antag för enkelhetens skull att varje rad i tabellen köp räknas som ett köp, så om en kund till exempel köpt 5 stycken vara A och 3 stycken vara B vid samma tillfälle räknas det som två separata köp i databasen. Antag också att alla butiker har sålt produkter. Tänk på att olika butiker kan ha olika priser för samma vara)