

Linköpings Universitet
Institutionen för datavetenskap
Eva Ragnemalm

T E N T A M E N

732G16 Databaser Design och programmering

Datum: 2018-06-07 kl 8-12

Jourhavande lärare: Eva Ragnemalm

Tel: 070-1907391

Besöker tentamenslokalen: c:a kl 9:30

Hjälpmedel: Svensk-kinesisk, svensk-engelsk, svensk-norsk ordbok (vid behov).

Följ instruktionerna på tentaomslaget. Svara inte på flera uppgifter på samma papper.

Uppgift	Poäng
1	6
2	8
3	14
4	4
5	7
6	4
7	7
8	8

Betyg: G: 30p VG: 45p Max: 58p

1. **Begrepp:** För varje fråga ange vilket (endast ett) av alternativen som är rätt. (6p)

- a. Schemat för en databas:
 - i. beskriver när en viss användare får tillgång till en viss tabell
 - ii. beskriver vilka kolumner som ingår i en viss tabell
 - iii. är datat som ingår i en tabell
 - iv. alla ovanstående
 - v. ingen av ovanstående

- b. Att det finns ett funktionellt beroende $a \rightarrow b$ betyder att:
 - i. alltid när a har ett visst värde har b samma värde som a
 - ii. ibland när a har ett visst värde har b samma värde som a
 - iii. ibland när a har ett visst värde har b ett visst (annat) värde
 - iv. alla ovanstående
 - v. ingen av ovanstående

- c. Vi har arbetat med datamodeller på tre olika abstraktionsnivåer i kursen. Den mest abstrakta är:
 - i. ER-modell
 - ii. konceptuell modell
 - iii. begreppsmodell
 - iv. alla ovanstående
 - v. ingen av ovanstående

- d. Vi har arbetat med datamodeller på tre olika abstraktionsnivåer i kursen. Den mest konkreta är:
 - i. implementationsmodell
 - ii. fysisk datamodell
 - iii. representationsmodell
 - iv. alla ovanstående
 - v. ingen av ovanstående

- e. En loggfil används för att uppnå vissa av egenskaperna för en transaktion: vilka?
 - i. I, D
 - ii. A, C, D
 - iii. A, C, I, D
 - iv. alla ovanstående
 - v. ingen av ovanstående

- f. I en fil som lagras i en hashtabell kan man söka:
 - i. på primärnyckeln eller sekvensiellt genom filen
 - ii. med primärindex eller sekundärindex
 - iii. med binärsökning
 - iv. alla ovanstående
 - v. ingen av ovanstående

2. **SQL, Relationsmodell:** Antag att du har de tabeller som definieras av nedanstående kommandon, som beskriver passagerare som har biljetter till en viss tågresor (antag att kommandona fungerar - bortse från ev. stavfel/slarvfel): (8p)

```
CREATE TABLE Passenger (pnr CHAR(10) PRIMARY KEY, name (VARCHAR
(50), phone VARCHAR(18))
Engine InnoDB;
```

```
CREATE TABLE Ticket (id INTEGER PRIMARY KEY, trainNo INTEGER,
start (VARCHAR(20), destination VARCHAR(20), carNo INTEGER,
seatNo INTEGER, passenger CHAR(10),
FOREIGN KEY passenger REFERENCES Passenger (pnr))
Engine InnoDB;
```

Antag att tabellerna Passenger och Ticket har följande innehåll:

Passenger

<u>pnr</u>	name	phone
6707070707	Anna Andersson	070-1234567
9001010101	Lisa Andersson	0706-1234567
9505050505	Petter Nilsson	+46(0)70-1234567

Ticket

<u>id</u>	trainNo	start	destination	carNo	seatNo	passenger
2	12	Linköping	Stockholm	3	45	6707070707
3	2	Motala	Linköping	-	-	9001010101
5	12	Norrköping	Södertälje	2	23	9001010101

- Skriv ett SQLkommando som listar namn och telefonnummer (Passenger.name, Passenger.phone) för alla passagerare som har biljett till tåg 123 (Ticket.trainNo=123).
- Skriv ett SQLkommando som listar tågnummret (Ticket.trainNo) och antal biljetter för alla tåg som har mer än 100 biljetter sålda (såld=finns i tabellen).
- Givet ovanstående innehåll i Ticket och Passenger, vad blir resultatet av nedanstående kommando:

```
INSERT INTO Ticket VALUES (6, 5, "Norrköping", "Gävle", 4, 23,
"6707070707");
```

- Givet ovanstående innehåll i Ticket och Passenger, vad blir resultatet av nedanstående kommando:

```
INSERT INTO Ticket VALUES (2, 6, "Linköping", "Gävle", 4, 18,
"6707070707");
```

- e. Givet ovanstående innehåll i Ticket och Passenger, vad blir resultatet av nedanstående kommando:

```
DELETE FROM Passenger where pnr="6707070707";
```

- f. Givet ovanstående innehåll i Ticket och Passenger, vad blir resultatet av nedanstående kommando:

```
DELETE FROM Passenger where pnr="9505050505";
```

Kombinationsuppgift: följande är bakgrundsinformation till uppgift 3, 4, 5, 6 och 7.

Vi vill konstruera en app som förmedlar försäljning av överbliven mat, som annars skulle kastats bort, från restauranger. Den ska heta RRM - Rädda RestaurangMaten. Tanken är att restauranger ska kunna lägga ut överbliven mat (till reducerat pris), som folk ska kunna köpa via denna app och sedan hämta på restaurangen. Kunderna ska få notiser om erbjudanden och också kunna favorit-markera restauranger (peka ut restauranger som de gärna vill ha erbjudanden ifrån).

För restaurangerna behöver man hålla reda på namn och kontaktinformation (adress och ett eller flera telefonnummer), samt när maten kan hämtas (oftast mellan två klockslag men ibland bara en sista tid). Restaurangerna ska också ha en presentationstext som visas när man klickar på restaurangen.

För kunderna behöver man ett användar-id och lösenord (som krypteras), samt namn och vilket kreditkort de betalar med (kortnummer, giltighetstid och cvv-kod). Som kund ska man också kunna välja hur man vill ha notiser skickade till sin telefon (aldrig, varje timme, kl 14 eller bara från favorit-restauranger). Man ska också kunna välja ifall man har speciella krav på maten på grund av kosthållning eller allergier, som t.ex. vegetarian, vegan eller gluten-, laktos- eller nötallergiker, och då bara få erbjudanden som stämmer med detta. Observera att man kan vara både vegetarian och allergisk mot gluten (eller ha andra kombinationer av krav). Man ska också kunna markera speciella restauranger som man vill få notis om när de lägger ut erbjudanden (favorit-markering). Man ska också lagra vilka maträtter kunden köpt, och hur många portioner.

För att det ska bli enkelt för restaurangerna har man begränsad information om maträtterna. Maten säljs i färdiga portioner, och varje maträtt som läggs ut har en rubrik, en bild och finns i ett visst antal (minst 1). Pris anges både som ordinarie pris (vad maten kostade ursprungligen) och ett erbjudande-pris (vad det kostar nu via denna app). Varje maträtt uppfyller olika krav och kombinationer av krav - till exempel kan den passa både vegetarianer och glutenallergiker. Det är viktigt att de krav användarna kan ställa är samma som restaurangerna använder för att klassificera sina maträtter. Det finns inget som garanterar att den text restaurangen använder som rubrik på en maträtt inte används av en annan restaurang, däremot undviker restaurangen själv att använda samma rubrik på två rätter.

3. **Design:** Utforma en ER-modell (rita ett ER-diagram) för den databas som behövs för att lagra nödvändig data. Om du anser att det saknas information får du göra rimliga antaganden om du skriver ned dem. Glöm inte markera nycklar, kardinalitet och deltagande samt eventuella semantiska villkor.

4. **Design, Relationsmodell:** En klanlig programmerare utformade nedanstående tabell för att hantera restauranger och maträtter. Denna tabell misslyckas med att hantera en del av kraven på information om restaurang och maträtter som beskrivs ovan (tabellen klarar inte att lagra vissa saker). Nämn minst 2. (Observera att vi bortser från kraven på information om kund/användare här.) (4p)

<u>Restaurang</u>	Adress	Telefon	<u>Mat</u>	Matbild	Antal	OrdPris	Pris	Passar
Zodiaken	Mäster mattias väg	070-1234567	Sushimix, 10 bitar	url://www.hörs.se/sushibild.png	3	100	50	Glutenallergiker
Java	Hans Meijers väg	013-1234567	Ostfrallam paprika	url://www.java.se/ost.png	5	26	13	Vegetarian
Java	Hans Meijers väg	013-1234567	Skinkfralla	url://www.java.se/skinka.png	3	30	15	-

5. **Normalisering:** Tabellen ovan har problem med dubbellagring. De två markerade kolumnerna är tillsammans en fungerande primärnyckel. (7p)
- Identifiera de funktionella beroenden som rimligen finns och rita upp dem med pilar som vi gjort i kursen.
 - Normalisera tabellen så att den uppfyller BCNF. Du kan anta att den uppfyller första normalformen motivera övriga normalformer och eventuella uppdelningar.
6. **Transaktioner:** Programmeraren skriver programkod för att genomföra ett köp enligt följande 6 steg i pseudokod (pseudokoden beskriver vad programkoden gör):
- Läs den valda raden i tabellen.
 - Om Antal > 1 så: presentera maträtten med rubrik, bild, OrdPris och Pris.
Annars skriv: "Oops, den tog slut."
 - (vänta på användarens inmatning (knapptryckning). Andra användare får möjlighet att använda databasen här.)
 - Om användaren klickar på "Bekräfta köp" så: Skicka betalningsbegäran till banken med kortinformation och Pris. Ändra Antal på den inlästa raden till Antal-valt antal. Skriv ut den ändrade raden i tabellen.
- Om användaren klickar bakåtknappen, gå till listan igen (inget köp görs).

Ibland visar det sig att kunder råkat köpa mat fastän den var slut. Var i pseudokoden ska programmeraren lägga in de viktiga kommandon som löser problemet? Du får lägga till annan kod i pseudokoden om du motiverar varför den behövs. (4p)

7. **Fysiska databasen:** När man löst problemet i föregående uppgift blir appen blir en framgång och filen som ovanstående tabell lagras i blir stor och tar tid att söka i. Filen är sorterad på primärnyckeln och tar upp 15 000 block. Man skapar ett primärindex till den. En post i indexfilen tar 400 byte inklusive blockpekaren (ganska stort för ett index och beror på att nyckeln är text). Hårddiskens blockstorlek är 4096 byte och accesstiden 10 ms. (7p)

- Hur stort blir indexet (hur många block tar det upp)?
- Hur lång tid tar det att söka fram en viss maträtt (du har hela nyckeln)?
- Om man skulle göra ett multipelindex, hur mycket snabbare skulle sökningen bli då?

Ledning: $\log_2(15000)=13,9$; $\log_2(1500)=10,5$; $\log_2(150)=7,2$; $\log_2(15)=3,9$

8. **Relationsmodell:** Nedanstående ER-diagram beskriver Sjukhemmet Solåsens data om patienter, läkare och prover som tas på patienterna. Konvertera ER-diagrammet till relationsschema. Markera nycklar med understrykning och främmande nycklar med pilar (som vi gjort i kursen). (8p)

