

Örebro universitet
Institutionen för naturvetenskap och teknik
[Thomas Padron-McCarthy \(thomas.padron-mccarthy@oru.se\)](mailto:thomas.padron-mccarthy@oru.se)

Tentamen i

Databasteknik

för D1 m fl

onsdag 16 augusti 2017

Gäller som tentamen för:
DT105G Databasteknik, provkod 0100

Hjälpmedel:	Ordbok för översättning.
Poängkrav:	Maximal poäng är 35. För godkänt betyg krävs 21 poäng, varav minst fyra poäng på uppgift 1 och minst fem poäng på uppgift 2.
Resultat:	Meddelas på kursens hemsida eller via e-post senast onsdag 6 september 2017.
Återlämning av tentor:	Elektroniskt via Studentforum.
Examinator och jourhavande:	Thomas Padron-McCarthy, telefon 070 - 73 47 013.

- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara formuleringar kommer att misstolkas.
 - Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
 - Skriv bara på en sida av papperet. Använd inte röd skrift.
 - Antaganden utöver de som står i uppgifterna måste anges. Gjorda antaganden får inte förändra den givna uppgiften.
 - Skriv gärna förklaringar om hur du tänkt. Även ett svar som är fel kan ge poäng, om det finns med en förklaring som visar att huvudtankarna var rätt.
-

LYCKA TILL!

Scenario till uppgifterna

Vi ska göra en databas över flygplan och flygpassagerare. Som illustration visar vi Airbus A380, som är världens största passagerarflygplan. Beroende på vilken inredning man valt, kan det ha plats för upp till 853 passagerare.



Airbus A380. Av: Roman Zelentsov (Rulexip). Licens: Creative Commons Attribution-Share Alike 3.0 Unported.

I databasen ska vi lagra följande:

- **Passagerare.** Varje passagerare beskrivs med ett unikt nummer, ett namn och en nationalitet.
- **Flygplanstyper.** Varje flygplanstyp har ett unikt namn, och det namnet består av namnet på tillverkaren, till exempel **Airbus**, och ett modellnamn, till exempel **A380**. Dessutom finns ett maximalt antal passagerare för den flygplanstypen, till exempel **853**.
- **Flygplan.** Det här är de enskilda flygplanen. Varje flygplan är av en viss flygplanstyp, och har en unik registreringsbeteckning, till exempel **SE-KSG**.
- **Flighter.** En "flight" innebär att ett flygplan flyger från en flygplats till en annan. Vi bryr oss inte om att lagra flygplatserna i den här databasen. Förutom att flighten gäller ett visst flygplan har den ett unikt nummer, och startade ett visst datum.
- Vi ska också lagra vilka passagerare som flugit på vilka flighter.

Uppgift 1 (5 p)

Rita ett ER- eller EER-diagram för den beskrivna databasen. Använd informationen i scenariot ovan, men tänk också på att det ska gå att svara på frågorna i uppgift 3 nedan.

ER- och EER-diagram kan ritas på flera olika sätt. Om du använder en annan notation än kursboken, måste du förklara den notation som du använder.

Uppgift 2 (6 p)

Implementera den beskrivna databasen i relationsmodellen, dvs översätt ER-diagrammet till tabeller.

Du behöver inte skriva **create table**-kommandon i SQL, men du ska ange vilka relationer som finns och vilka attribut varje relation innehåller. Ange också alla kandidatnycklar, vilken av dessa som är primärnyckel, samt vilka referensattribut som finns och vad de refererar till.

Implementationen ska vara bra.

Uppgift 3 (10 p)

Formulera följande frågor i SQL. Definiera gärna vyer om det underlättar, men skapa inte nya tabeller.

- a) (1p) Vilka olika modeller av flygplan tillverkas av tillverkaren **Airbus**?
- b) (2p) Vilka nationaliteter finns representerade bland passagerarna på flight nummer **4711**?
- c) (2p) Det finns bara en enda passagerare som heter **Anna Alm** i databasen. Vilka flygplanstyper har hon flugit med?
- d) (2p) Vilka passagerare som *inte* är från **USA** har någon gång flugit med ett flygplan av typen **SR-71 Blackbird** från tillverkaren **Lockheed**?
- e) (3p) Vi ska leta efter fel i databasen, så vi vill ha svar på följande fråga: Vilka flights har fler passagerare än vad flygplanstypen har plats för?

Uppgift 4 (4 p)

Databasen innehåller realistiska mängder data, vilket innebär tusentals flygplanstyper, tiotusentals flygplan, miljoner flighter och hundratals miljoner passagerare. SQL-frågorna a, b och c i uppgift 3 körs väldigt ofta (men kanske med andra konstanter, till exempel att man söker efter **Boeing** i stället för **Airbus**). De tar för lång tid att köra, och behöver snabbas upp. Vi märker att det inte finns några index alls i databasen, inte ens på nycklar.

- a) Vilka index bör man skapa för att dessa frågor ska gå snabbare?
- b) Nämn ett index som **inte** skulle snabba upp dessa frågor, och förklara varför det inte skulle göra det.

Uppgift 5 (5 p)

I uppgift 2, när man ska skapa tabeller, står det att implementationen ska vara bra. Här är ett försök att skapa en av tabellerna, men tyvärr blev det inte så bra. Tabellen är inte så väl normaliserad som man helst skulle vilja.

Flighter					
Flightnr	Start	Plan	Tillverkare	Modell	Maxpassagerare
1	2017-01-02	SE-KSG	Airbus	A380	853
2	2017-02-09	SE-KSG	Airbus	A380	853
3	2017-02-09	SE-KOD	Boeing	747	660
4	2017-08-10	SE-KOD	Boeing	747	660
5	2017-08-12	SE-BLX	Boeing	747	660

- a) Vilka fullständiga funktionella beroenden finns i tabellen?
- b) Vilken är den högsta normalform, av 1NF, 2NF, 3NF och BCNF, som tabellen uppfyller?
- c) Varför uppfyller tabellen inte den närmast högre normalformen?
- d) Beskriv något problem som finns i den här tabellen, och som man skulle slippa med en högre normalform.

Uppgift 6 (5 p)

Förklara kort vad följande termer från databasområdet betyder! Det kan vara lämpligt med 2-4 meningar om varje term.

- a) integritetsvillkor
- b) transaktion
- c) lagrad procedur
- d) trigger
- e) XML