

Tentamen i EDAF75

12 mars 2024

Skrivtid: 14-19

- **SKRIV TYDLIGT:** om texten inte går att läsa kan du inte få några poäng.
- **SKRIV BARA PÅ ENA SIDAN AV PAPPRET:** tentorna kommer att scannas in, och endast framsidorna rättas.
- **SKRIV INGEN PERSONLIG KOD PÅ TENTAN:** den 'personliga kod' som används på en del tentor gör tentorna mindre anonyma, så vi vill inte ha den!
- **SÄTT IDENTITET OCH SIDNUMMER PÅ VARJE INLÄMNAT BLAD:** om din anonymkod skulle vara EDAF75-0012-ABC så skall du bara skriva 12-ABC längst upp i högerkanten på varje inlämnat blad (hoppa över kurskod och inledande nollor) – skriv dessutom sidnumret och antal inlämnade blad under identiteten, som nedan (exemplet visar sidan 3 av 5 på tentan med anonymkoden EDAF75-0012-ABC):

12-ABC

3/5

- **KONTROLLERA NOGA ATT DU FÅR MED ALLA BLAD, OCH LÄGG DEM I RÄTT ORDNING I OMSLAGET.** Om du markerar sidorna enligt ovan, *och lägger dem i ordning*, så är det enklare att kontrollera att alla sidor finns med.
- *Preliminär* maxpoäng anges vid varje uppgift.
- *Preliminära* betygsgränser:
 - 3: Minst ca 60% på *var och en* av uppgift 1-3 (alla tre måste vara godkända).
 - 4: Minst ca 70% på *var och en* av uppgift 1-3, och ca 70% totalt på uppgift 4-6.
 - 5: Minst ca 85% på *var och en* av uppgift 1-3, och ca 85% totalt på uppgift 4-6.

Uppgift 1

I denna uppgift förekommer ett antal termer som kan vara svåra att översätta till engelska under tentan, om du vill får du gärna använda svenska namn i din modell (å, ä och ö går bra!).

I Lunds kommun händer massor av saker, och det är lätt att missa evenemang som man gärna skulle ha gått på, så familjen S vill ha en databas där man kan lägga in de evenemang som någon i familjen kan tänkas vara intresserad av.

Det finns ett antal web-sidor med information om vad som händer i kommunen (sådana sidor finns exempelvis på kommunens hemsidor, och på olika tidnings- och turistsidor) – vi vill för varje sådan ”tips-sida” hålla reda på dess namn (typ ’Kommunens hemsida’), en URL (typ <https://lund.se/evenemang>), och den tidpunkt då vi senast tittade på sidan.

På tipssidorna anges olika evenemang, och vi vill för varje intressant evenemang spara följande information i databasen:

- Uppgift om när det börjar och, en kort egen textbeskrivning.
- Vilken eller vilka tipssidor som tipsat om evenemanget (så att vi senare kan se vilka tipssidor som vi faktiskt har nytta av att följa) – det är inte ovanligt att flera tipssidor rekommenderar samma evenemang.
- Eventuella länkar till mer information om evenemanget (dessa länkar är i princip oberoende av vilken/vilka tipssidor som tipsade om händelsen).

Vi vill dessutom kunna definiera olika kategorier av evenemang, som ’barn’, ’teater’, ’musik’, eller ’innebandymatch’, och vi vill i databasen kunna lägga in nya kategorier som vi själva definierar efterhand. Till varje evenemang vill vi kunna koppla en eller flera kategorier, så att en barnteaterföreställning kan markeras med både ’barn’ och ’teater’.

Databasen skall hålla koll på samtliga familjemedlemmar, och dessutom hålla koll på vilka kategorier de är intresserade av, så att vi kan generera utskrifter med lämpliga evenemang för en given familjemedlem.

Avslutningsvis vill vi kunna lagra recensioner av olika evenemang – en recension görs av en familjemedlem, och den består av ett betyg (ett heltal från 1-5), och en text. Inte alla aktiviteter behöver recenserars, och varje familjemedlem skriver högst en recension per evenemang.

- (a) Gör en E/R-modell som beskriver databasen, redovisa med ett UML-diagram, markera nycklar, och skriv ut multipliciteter på samtliga associationer. *Skriv inte ut främmande nycklar som redan finns markerade i diagrammet.* Du behöver inte markera dina entity-sets med “Weak”. 8 p

Skriv sedan SQL-satser för att definiera samtliga tabeller för din E/R-modell – var noga med att deklarerar primärnycklar och främmande nycklar. Använd samma namn som i E/R-modellen på tabeller och attribut. För att spara tidpunkter kan du använda typen DATETIME, som lagrar både datum och klockslag. Du behöver inte skriva `DROP TABLE ...` för dina tabeller. 11 p

- (b) Utgående från dina tabeller i uppgift a, skriv en SQL-sats som skriver ut URL-en för alla tipssidor från vilka vi lagt in tips om innebandymatcher som spelas under 2024 (du kan använda funktionen `year()` för att hämta året för en DATETIME). 2 p

Uppgift 2

Ett bibliotek håller reda på sina låntagare och sina böcker, varje boklån registreras med lånedatum, sista återlämningsdatum och faktiskt återlämningsdatum – faktiskt återlämningsdatum (`actual_return_date` nedan) är NULL om boken ännu inte är återlämnad. För enkelhets skull antar vi att vi bara har ett exemplar av varje bok.

Böckerna klassificeras enligt ”The Dewey Decimal Classification”, som kopplar treställiga koder till olika kategorier (300 är samhällsvetenskaper, 400 är språk, 500 är naturvetenskap och matematik, etc.). Koderna består bara av siffror, men vi lagrar dem som strängar. I uppgifterna nedan frågas på några ställen efter skönlitterära böcker, de har alltid en klassificering som börjar på 8.

Bibliotekets databas beskrivs i följande relationer (understrukna attribut är delar av primärnycklar, kursiverade fetstilta attribut är främmande nycklar):

```
borrowers(card_nbr, name, address)
books(isbn, title, author, publisher_id, classification)
loans(card_nbr, isbn, borrow_date, latest_return_date, actual_return_date)
publishers(publisher_id, publisher_name)
classifications(classification, description)
```

För uppgift (b)-(f) nedan skall du skriva SQL-satser.

- (a) Rita ett ER-diagram i UML-format för att beskriva databasen, rita diagrammet enligt anvisningarna i problem 1a. 2 p
- (b) Skriv ut titel och förlag för de böcker som är utlånade för tillfället (sorterat efter titel). 2 p
- (c) Ta bort alla böcker som aldrig varit utlånade. 2 p
- (d) Skriv ut namnet på de låntagare som aldrig lånat en skönlitterär bok (skönlitterära böcker har en klassificering som börjar på 8). 2 p
- (e) Skriv ut namnen på de låntagare som vid något tillfälle har lånat någon bok med fransk poesi som är publicerad av förlaget med namnet ”ACME böcker” – texten ”Fransk poesi” finns som en beskrivning (`description`) i tabellen `classifications`. 2 p
- (f) Skriv för de olika förlagen ut hur många av deras skönlitterära böcker biblioteket har i sin databas – *vi vill dock bara ha med förlag med minst 10 sådana böcker* (skönlitterära böcker har en klassificering som börjar på 8). 3 p

Uppgift 3

För att ingen skall tappa onödiga poäng på ett eventuellt misstag i (a), så har vi helt separerat uppgift (a) från uppgift (b)..(e).

(a) I en liten modell för filmvärlden har vi följande attribut:

- actor_id: IMDB-nyckeln för en skådespelare.
- actor_name: namnet på en skådespelare (behöver inte vara unikt).
- movie_id: IMDB-nyckeln för en film.
- movie_role: en roll i en film – vi antar för enkelhets skull att varje roll i en given film bara görs av en person, och att personen bara gör en roll i filmen. En roll (exempelvis "Hamlet") kan förekomma i många filmer.
- category: kategori för "Academy Awards" (Oscar) – exempelvis "Bästa kvinnliga skådespelare", "Bästa manliga skådespelare i biroll", etc.
- year: år för utdelning av en "Academy Award".

Vid Oscars-utdelningen delas varje år priser ut till bästa skådespelare i olika kategorier, *de får priset för en given roll i en given film*. Vi bryr oss i uppgiften bara om skådespelarpriserna (inte "Bästa film" och andra kategorier).

Bestäm de funktionella beroenden som gäller för attributen ovan, och använd beteckningarna ovan, dvs introducera *inte* nya namn som A, B, etc. För full poäng vill vi att dina beroenden skall vara så enkla som möjligt (men vi gör bara små avdrag om du lägger till 'onödiga' men korrekta attribut i högerleden).

4 p

(b) I relationen $R(A, B, C, D, E, F, G, H)$ gäller följande funktionella beroenden:

$$FD_1: ADF \rightarrow C$$

$$FD_2: D \rightarrow EH$$

$$FD_3: F \rightarrow BG$$

Bestäm relationens nycklar, motivera ditt svar.

2 p

(c) Visa att relationen inte är i BCNF.

1 p

(d) Bryt ner relationen R i (b) i mindre relationer som är i BCNF, och som kan rekonstrueras till den ursprungliga relationen utan att vi förlorar eller lägger till rader. Motivera nedbrytningen genom att i *varje steg* skriva dina relationer på nedanstående form, och *förklara de uppdelningar du gör*:

$R(A, B, C, D, E, \dots)$
Beroenden: $A \rightarrow BC, BD \rightarrow E, \dots$
Nycklar: AC, AD, \dots

Det vi är intresserade av är alltså framförallt stegen som leder fram till de slutliga relationerna, *att svara genom att bara ange relationer utan att förklara nedbrytningarna ger inga poäng*.

5 p

(e) Visa med en SQL-sats hur vi kan återskapa den ursprungliga relationen i (b) med hjälp av relationerna som du bröt upp den i ovan. Denna uppgift ger bara poäng om lösningen i (d) motiverar att vi får tillbaka exakt samma rader.

1 p

Uppgift 4

Definiera för databasen i uppgift 2 en trigger som gör att en låntagare aldrig kan låna mer än 10 böcker samtidigt.

4 p

Uppgift 5

Skriv ett Python- eller Javaprogram som med databasen i uppgift 2 löser nedanstående uppgift. Vi antar att vi i databasen har triggern från uppgift 4, oavsett om du löst uppgift 4 eller inte, och att vi definierat tabellen *loans* så att *borrow_date* och *latest_return_date* får rätt startvärden automatiskt.

Läs in ett lånekortnummer och en ISBN-kod, och notera i databasen att låntagaren med det givna lånekortnumret lånar boken med det givna ISBN-numret. Om låntagaren redan har 10 utestående lån skall lånet inte genomföras, och programmet skriva ut att ett nytt lån skulle leda till för många utestående lån.

Skriv därefter ut samtliga utestående lån för den aktuella låntagaren, och visa sista returdatum för varje lån (utskriften skall göras oavsett om lånet kunde genomföras eller ej).

Databasen ligger i SQLite-filen "library.sqlite".

Du skall skriva ett program som går att köra, men du behöver inte skriva några import-satser. Om du skriver Pythonkod måste du använda `sql` i `te3`-biblioteket, och om du skriver Javakod måste du använda JDBC-paketet.

För att läsa in en sträng kan du använda följande anrop:

- I Python: `s = input("Ange s: ")`
- I Java: `var s = EDAF75.nextLine("Ange s: ");`

4 p

Uppgift 6

(a) Förklara kortfattat termerna *isolation level*, *read committed* och *repeatable read*.

1.5 p

(b) Bevisa att en relation med bara två attribut alltid är i BCNF.

1.5 p