

Databasesystemer, forår 2006

Rasmus Pagh
Esben Rune Hansen

Dette opgavesæt indeholder øvelser i at skrive SQL-forespørgler. Øvelserne er af forskellig sværhedsgrad. Den første opgavesektion (1) skulle være rimelig nem at gå til. Den anden sektion (2) kræver en god forståelse af JOINS. Den tredje sektion (3) kræver en god forståelse af det meste af stoffet der blev gennemgået i denne uges forelæsning.

1 Opgaver der bør kunne løses ud fra sidste uges forelæsning

Øvelserne i denne sektion involverer relationerne `Product`, `PC` og `Laptop`, som kan hentes via kursushjemmesiden som beskrevet på første øvelsesark (dvs. hvis du har lavet disse øvelser, har du allerede relationerne i din Oracle database). Relationerne har flg. skemaer:

```
Product (id, maker, model, type)
PC (model, speed, ram, hd, rd, price)
Laptop (model, speed, ram, hd, screen, price)
```

Skriv forespørgsler, der returnerer følgende:

1. Alle laptops med pris mindre end 2500.
2. Modelnumret for alle laptops med pris mindre end 2500 eller med `speed` på mindst 800.
3. Alle tupler i `Product`, der omhandler laptops.

2 Opgaver der kræver stof fra denne uges forelæsning

Syntaksen `Product NATURAL JOIN Laptop` kan bruges til at kombinere tupler i `Product` og `Laptop` med samme værdi i `model` attributten. Afprøv dette i Oracle ved at skrive flg:

```
SELECT * FROM (Product NATURAL JOIN Laptop);
```

Gentag øvelserne fra sidste sektion (1) med den tilføjelse, at relateret data i den anden relation skal medtages i resultatet. For forespørgsel 1 skal information fra `Product` f.eks. medtages i resultatet. Prøv derefter flg. forespørgsel:

```
SELECT * FROM (PC NATURAL JOIN Laptop);
```

Forklar hvorfor resultatet er tomt. Skriv derefter en mindre streng join-forespørgsel på formen:

```
SELECT * FROM PC, Laptop where ... = ...
```

som ikke giver et tomt resultat. Tilføj derefter en tupel i `Laptop`, som bevirker, at forespørgslen:

```
SELECT * FROM (PC NATURAL JOIN Laptop);
```

ikke længere giver et tomt resultat.

3 Videregående SQL-øvelser

De følgende øvelser omhandler SQL forespørgsler på data fra bogens materiale om Pine Valley Furniture. For at oprette (eller genoprette) data i Oracle skal du logge ind via SQL*Plus, og skrive følgende kommando (kopier evt. direkte fra denne pdf fil, men vær i så fald opmærksom på, hvorvidt _ kommer med):

```
start /import/home/pagh/public.html/DBS05/Intranet/CREATEPVFC7e.sql
```

Relationsskemaerne er vist på de næste sider. De er udledt fra E-R diagrammet i figur 3-22 (s. 125) i MDM, hvor der også kan findes en kort forklaring på betydningen af attributter og relationernes navne.

Øvelserne er følgende exercises i MDM:

1. Kapitel 7, s. 323, opgave 14–18. (Find den rigtige tabel, udfør den rigtige forespørgsel. **Hint:** du for brug for LIKE, GROUP BY, AVG, AND, '<' og '>')
2. Kapitel 8, s. 352, opgave 6–8. (Øvelser i Joins)
3. Kapitel 7, s. 323, opgave 20–23. I spørgsmål 21–23 skal I endvidere modificere forespørgslen eller tilføje tupler til relationerne, så resultatet ikke er tomt.
4. Gå evt. videre med resten af opgaverne 10–31 på s. 322–323.

Skemaer over tabeller i Pine Valley Furnitures Database

CUSTOMER_t(CUSTOMER_ID,CUSTOMER_NAME,CUSTOMER_ADDRESS,CUSTOMER_CITY,CUSTOMER_STATE,POSTAL_CODE)

Does_Business_In_t(CUSTOMER_ID,TERRITORY_ID)

EMPLOYEE_t(EMPLOYEE_ID,EMPLOYEE_NAME,EMPLOYEE_ADDRESS,EMPLOYEE_BIRTHDATE,EMPLOYEE_CITY,
EMPLOYEE_DATE_HIRED,EMPLOYEE_STATE,EMPLOYEE_SUPERVISOR,EMPLOYEE_ZIP)

Employee_Skills_t(EMPLOYEE_ID,SKILL_ID)

ORDER_t(ORDER_ID,CUSTOMER_ID,ORDER_DATE)

Order_line_t(ORDER_ID,PRODUCT_ID,ORDERED_QUANTITY)

Produced_in_t(PRODUCT_ID,WORK_CENTER_ID)

PRODUCT_t(PRODUCT_ID,PRODUCT_LINE_ID,PRODUCT_DESCRIPTION,PRODUCT_FINISH,STANDARD_PRICE)

PRODUCT_LINE_t(PRODUCT_LINE_ID,PRODUCT_LINE_NAME)

Raw_material_t(MATERIAL_ID,MATERIAL_NAME,STANDARD_COST,UNIT_OF_MEASURE)

SALESPERSON_t(SALESPERSON_ID,SALESPERSON_NAME,SALESPERSON_PHONE,SALESPERSON_FAX,TERRITORY_ID)

SKILL_t(SKILL_ID,SKILL_DESCRIPTION)

Supplies_t(VENDOR_ID,MATERIAL_ID,SUPPLY_UNIT_PRICE)

TERRITORY_t(TERRITORY_ID,TERRITORY_NAME)

Uses_t(GOES_INTO_QUANTITY,MATERIAL_ID,PRODUCT_ID)

Vendor_t(VENDOR_ID,VENDOR_NAME,VENDOR_ADDRESS,VENDOR_CITY,VENDOR_CONTACT,VENDOR_FAX,
VENDOR_PHONE,VENDOR_STATE,VENDOR_TAX_ID,VENDOR_ZIPCODE)

Works_In_t(EMPLOYEE_ID,WORK_CENTER_ID)

WORK_CENTER_t(WORK_CENTER_ID,WORK_CENTER_LOCATION)