Databasesystemer, Forår 2006

Rasmus Pagh
Esben Rune Hansen

Øvelser d. 30. marts

Øvelserne for denne uge er relateret til forelæsningen om "Mere SQL". Der er øvelser i sub-queries og authetication. Øvelserne på det første ark omhandler sub-queries. De kan laves som maskine-øvelser men du opfordres til at skrive dem ned i hånden og kun bruge Oracle til at teste om dine forespørgsler er korrekte. Øvelserne på det andet ark er om authentication of er taget fra et eksamenssæt.

Sub-queries

Exercise 1

Write the following queries, based on the database schema:

```
Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
Battles(name, date)
OutComes(ship, battle, result)
```

You should use at least one sub-query in each of your answers and write each query in two significantly different ways (e.g. using different sets of the operator `EXISTS`, `IN`, `ALL` and `ANY`).

1. Find the countries whose ships had the largest number of guns.
2. Find the classes of the ships at least one of which was sunk in a battle
3. Find the names of the ships with a 16-inch bore.
4. Find the battles in which ships of the Kongo class participates

Test your queries in Oracle on the sample data you entered in the beginning of the course.

Exercise 2

Write the following queries, based on the database schema:

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

You should use at least one sub-query in each of your answers and write each query in two significantly different ways (e.g. using different sets of the operator `EXISTS`, `IN`, `ALL` and `ANY`).

1. find the printers with the highest price
2. find the laptops whose speed is slower than that of any PC.

Test your queries in Oracle on the sample data you entered in the beginning of the course.

Authentication (Exam question worth 15%)

The user alice has just created a relation R(user,info), and issues the following SQL commands:

```
GRANT INSERT ON R TO bob WITH GRANT OPTION;
GRANT SELECT ON R TO bob WITH GRANT OPTION;
GRANT SELECT ON R TO claire;
```

Consider the following SQL commands:

1. SELECT * FROM alice.R WHERE user='claire';
2. INSERT INTO alice.R VALUES ('claire','clairvoyant');
3. GRANT SELECT ON alice.R TO dorothy;

a) State for each of the three users bob, claire, and dorothy, which of the above SQL commands he/she has autorization to execute.

Now assume that bob executes the command

```
GRANT INSERT ON R TO claire;
```

and alice then executes the command

```
REVOKE INSERT ON R FROM bob CASCADE;
```

b) Again, state for each of the three users bob, claire, and dorothy, which of the above SQL commands he/she has autorization to execute at this point.