**Programming - Databases**

**Exercise 1: Data Manipulation on Paper**

Use the Company Database in Elmasri, fig. 5.6 (in older version fig. 3.6) to answer the following questions:

a. What are the names of employees with a salary greater than 30000?

b. Retrieve the name and birthday of the employee with social security number (Ssn) 987987987.

c. What is the name of the department where Joyce A. English works?

d. List all last names of employees that work in department number 1.

e. List all names of employees that work in Administration.

f. What are the names of the projects that Alicia J. Zelaya works on?

g. How many hours are totally spent on the project “Reorganization”?

**Exercise 2 – Practical Exercise with SQL Server**

**The Database**

The database is the Company Database and the design is given in figure 3.7 (Elmasri). We talked about the database in the lecture and we worked with it in the previous exercise.

**Create the database**

The database is going to be created using MS SQL Server as DBMS. SQL scripts (small SQL programs) to creating and inserting sample data can be found in **CompanySqlScripts2.sql**.

**Do the following:**

A) Download and the scrípt-file.

B) Open the MS SQL Server Management Studio (SSMS) and connect to your server.

C) Open the CompanySqlScripts2.sql script and execute it.

Now you should be able to see the tables using the Object Explorer (It may be necessary to refresh the object explorer in order to view the changes). The tables were created by the “create table…” lines.

**Initializing**

The script has inserted the data shown in Elmasri, fig. 3.6.

Check how the “insert…” statements specify data that match the table definitions from the previous step.

The order in which the table definitions and the insert scripts were run is important, as this is how referential integrity is maintained. For instance, you cannot insert tuples into the Employee table referencing some department in the Department table if that department does not exist. The order that the tables were created in (see the CreateTables script) may give you a hint.

Finally, the “alter table department…” line is run, as this constraint would not have been possible to satisfy if it had existed on an empty table. Can you see why?

**Working with the Database.**

Currently, no application that accesses the database has been developed, so we will work through the interactive SQL workbench in the SQL Server Management Studio.

To interact with the database choose “**New Query**”, and you get an editor window where you can type SQL statements. To execute the statement click the “**Execute**” button or press **F5**. But before you do that you must choose the right database in the dropdown list at the top left or you must start the script with: “**use Company**;”.

To check if there are data in the tables, try the script:

select \* from Employee

**Searching:**Try out some queries (select statements):

**Searching in one table:**

select lname, fname   
from Employee   
where ssn='123456789'

select lname,fname  
from Employee   
where sex='F'

Note the differences in the results (1 row and several rows).

**Searching across multiple tables:**

select lname, fname, dname  
from Employee, Department

Does this result make sense? And why not? What is the problem?

Try this one instead – what is the difference?

select lname, fname, dname   
from Employee, Department   
where dnumber= dno

**Try out some queries of your own.**

Maybe some of the queries from the first exercise that you did on paper?

**Updating Rows:**

**Simple update:**Optionally try this search before and after the update:

select \*   
from Employee   
where fname = 'John'   
and lname = 'Smith'

Now do an update:

update Employee   
set fname = 'Johnny'   
where fname = 'John'   
and lname = 'Smith';

What happens when this statement is executed? Is it what one would expect? (Check the result by running a query!)

Consider this update:

Update Employee   
set fname = 'Joyce'   
where sex = 'F'

What would happen if this statement was executed? Is it reasonable to do this update?

**More complex update:**

Updating one table using another table to find the tuples to be updated:

update Employee   
set salary = salary\*1.1   
where Employee.dno =   
 (select Department.dnumber   
 from Department   
 where Department.dname = 'Research');

Which tuples are updated? How are they found?

**Inserting Rows:**

insert into employee values   
('Benno', null, 'Benzen', '876543210', '19581212', 'FarAway, Sæby, DK', 'M', 55000, null, 5);

Execute this. Does it behave as expected?

Now try this one:

insert into employee values   
('The', 'T', 'Kid', '876543210', '19581212', 'BadArea, Ålborg, DK', 'M', 55000, null, 5);

What is the problem?

**Deleting Rows:**

delete from Employee   
where fname= 'Benno'

What do you suppose will happen if there are more employees called ‘Benno’?

Deleting based on information from another table:

delete from works\_on   
where works\_on.essn in (   
 select employee.ssn   
 from employee   
 where employee.dno = (   
 select department.dnumber from department   
 where department.dnumber = (  
 select department.dnumber   
 from Department   
 where department.dname = 'Administration'   
 )   
 )   
)

Explain the query – what is deleted?

Hint: Understand the inner parts first

**Do some manipulation yourself:**

Use the examples in the previous sections as inspiration to do some manipulation on your own.

1) Do the following updates:

a) ’James’ should be ’Jim’

b) ’Jennifer’ is to be transferred to ’Headquarters’.

c) ’Franklin’ is to be transferred from the project ‘Reorganization’ to ‘Newbenefits’.

2) Deleting rows. Is this always possible? If not, what may the reasons be?

a) Try to sack (delete) ’Franklin’. Is this possible? Why not?

b) Read the error messages, and manipulate tables WORKS\_ON, DEPENDENT and EMPLOYEE to make it possible to delete Franklin. E.g he must be relieved from his position as department manager in employee table (why?). Do that (you decided who to promote in his place) – and then delete him!

3) Inserting new rows into a table:

a) Insert a new department – give it a name and number of your own choice, and set the manager’s ssn to ‘222222222’. Any problems? Try another manager.

b) Try to insert a new department with dnumber 1 – Why isn’t that possible?

**Reflections (“Lessons Learned”):**

Based on your experiences from the above exercises formulate general rules for updating, deleting, and inserting rows in tables.