Database Management Lab - 1

DDL	**DML**

Creating a table
CREATE TABLE table_name
(column_definition_list)
	Inserting a new row to a table:
	INSERT INTO table_name
	[(column_list)]
	VALUES (value_list)

Column Definiton: Column_name data_type_of_column	**Deleting an existing row from table:** DELETE FROM table_name [WHERE deleting_condition]
Droping a table DROP TABLE table_name	

Renaming an existing table:
RENAME old_name_of_table
TO new_name_of_table
	Updating an existing row of table:
	
	UPDATE table_name
	SET assignment_list
	[WHERE updating_condition]

Adding a new column to an existing table: ALTER TABLE table_name ADD column_definition	

DDL (Data Definition Language) Commands

"CREATE TABLE" command is used for creating a new table in the database. Usage of this command is:

```
Create table TableName
(
FieldName  FieldType (Width) PrimaryKey,
FieldName  FieldType (Width) Default (Default value),
FieldName  FieldType (Width) not null,
FieldName  FieldType (Width),
FieldName  FieldType,
Primary key (PkeyField1,PkeyField2)
) ;
```

(TableName) : is the name of the table which will be created.
(FieldName) : is name of the column/field of the new table.
(FieldType) : is data type of column/field. Data type can be one of those: Char(n), Varchar(n), Numeric(p,s), double(p,s) , Int4, Integer, bigint, int8, real, float4, double, float8, Smallint, byte, boolean. For more information about data types: http://www.postgresql.org/docs/8.4/interactive/datatype.html
(Width) : is width of the field.
(Default Value) : is default value of a column. It should be use together with the keyword "Default". When inserting a new row to the table, if any value is not inserted to the field which has a default value, that field will get that default value.
PrimaryKey : is used for integrity and consistency. If only one column is defined as primary key, simply you can write "primary key" after the definition of that column.
Primary Key ( ) : also can be used for primary key definiton. If more than one column get together and defined as a primary key, you should use this type of primary key definiton. Also only one column can be defined as a primary key by this type definiton.

Example 1 :

NAME	SURNAME	NO
char[20]	char[20]	numeric

Table "Student" is a table which has 3 fields/columns. The names of the fields are: Name, surname, and no. Their types are char(20), char(20) and numeric respectively. Create that table with an appropriate SQL sentence:

Create table Student
(
Name Char(20) not null,
Surname Char(20) not null,
No Numeric,
Primary Key (**No**)
);

- **ALTER TABLE … ADD** command is used for adding a new column to an existing table

Example 2 : Add a new column to the table Student, whose name is "Telephone" and type is char(11):

Alter table Student **add** Telephone char(11);

- **DROP TABLE** command is used for removing an existing table from database:

Example 3 : Destroy the table Student:

Drop table Student;

DML (Data Manipulation Language) Commands

We use DML statements to insert new rows to a table in the database; to delete existing rows from table; to update rows in the table.

INSERT command is used for adding a new row to a table:
Insert Into Table1 (Field1, Field2) **Values** ('String statement', numeric value ...)

DELETE FROM is used for deleting existing rows from a table: (If you want to delete all of the rows of a table, you should not use **Where deleting_condition** clause!)
Delete From Table1 **Where** deleting_condition

UPDATE ... SET is used for updating the information of the specified rows of a table: (If you want to update all of the rows of a table in the same way, you should not use **Where updating_condition** clause!)
Update Table1 **set** Field1='String statement', Field2=Numeric value, ...
Where updating_condition

DML Examples:

NAME	SURNAME	NO
char[20]	char[20]	Numeric

Example 1 : Let us use the table Student again. Add a new student to the table whose information is NAME = "Serkan" SURNAME = "Türkel" NO = 4683 by using an appropriate SQL sentence:

Insert Into student (NAME, SURNAME, NO) **Values**('Serkan','Türkel',4683)

Example 2 : Delete the information of a student whose number is 4556 from table by using the appropriate SQL clause:

Delete From student **Where** student.NO=4556

Example 3 : Change the information of the student whose number is 4683. Store new name of the student as “Ali”, new surname of the student as “Sert”:

UPDATE student **SET** NAME = 'Ali', SURNAME = 'Sert'
Where student.NO = 4683

Queries

We also use the SQL language for querying the information which is stored in the tables at our database.

We use command **SELECT** to query specified informations in the tables. Usage of SELECT:
Select TableName.Field1, TableName.Field2, ... [*] **From** TableName **Where** querying_condition

Examples of Queries:

Example 1 : List the NAME and SURNAME information of the students from table Student.

Select student.NAME, student.SURNAME **From** student

If we want to list all of the columns of a table, we can use sign “*” instead of listing all off the columns:

Select student.NAME, student.SURNAME, student.NO **From** student
Or we can write simply:
Select * From student

Example 2 : List all of the students whose first name is “Serkan”:

Select * From student **Where** student.NAME= "Serkan"

Example 3: List all of the students who has the letter " r " in her/his surname information:

Select * From student **Where** student.SURNAME **like** "%r%"

Example 4: List all of the students whose student number is between **1044 and 2866**:

Select * From student **Where** 1044<student.NO<2866
Or we can write:
Select * From student **Where** student.NO>1044 **and** student.NO<2866