2016-2017 Fall Semester
Database Management Systems
Experiment 3

Selecting Data

The **select** statement is used to query the database and retrieve selected data that match the criteria that you specify. Here is the format of a simple select statement:

```
select "column1"
  [,"column2",etc]
  from "tablename"
  [where "condition"];
  [] = optional
```

The column names that follow the select keyword determine which columns will be returned in the results. You can select as many column names that you'd like, or you can use a "*" to select all columns.

The table name that follows the keyword **from** specifies the table that will be queried to retrieve the desired results.

The **where** clause (optional) specifies which data values or rows will be returned or displayed, based on the criteria described after the keyword **where**.

Conditional selections used in the **where** clause:

=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal to

LIKE *

*The **LIKE** pattern matching operator can also be used in the conditional selection of the where clause. Like is a very powerful operator that allows you to select only rows that are "like" what you specify. The percent sign "%" can be used as a wild card to match any possible character that might appear before or after the characters specified. For example:

```
select first, last, city
from empinfo
where first LIKE 'Er%';
```

This SQL statement will match any first names that start with 'Er'. **Strings must be in single quotes.**

Or you can specify,

```
select first, last
from empinfo
where last LIKE '%s';
```

This statement will match any last names that end in a 's'.

```
select *
from empinfo
where first = 'Eric';
```

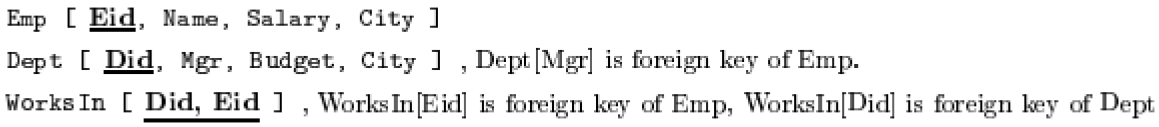
ALL and DISTINCT are keywords used to select either ALL (default) or the "distinct" or unique records in your query results. If you would like to retrieve just the unique records in specified columns, you can use the "DISTINCT" keyword. DISTINCT will discard the duplicate records for the columns you specified after the "SELECT" statement: For example:

```
select distinct age
from employee_info;
```

This statement will return all of the unique ages in the employee_info table.

ALL will display "all" of the specified columns including all of the duplicates. The ALL keyword is the default if nothing is specified.

Exercises :



Emp [Eid, Name, Salary, City]
Dept [Did, Mgr, Budget, City] , Dept[Mgr] is foreign key of Emp.
WorksIn [Did, Eid] , WorksIn[Eid] is foreign key of Emp, WorksIn[Did] is foreign key of Dept

1. Draw an ER diagram for the relation schema given above. Identify referential integrity constraints of each table.

2. Create the tables according to relation schema given above.

3. Insert at least 3 records into each table.

4. Insert below records into Emp table.

Emp Table			
Eid	**Name**	**Salary**	**City**
1	Ali Yılmaz	2000,00	Eskişehir
2	Veli Doğan	3000,00	Ankara
3	Ayşe Parlak	2950,00	Kütahya
4	Mehmet Duran	3500,00	Eskişehir
5	Aslı Çelik	2200,00	Eskişehir
6	Kader Kara	2700,00	Bursa
7	Merve Yılmaz	3800,00	İzmir

5. Display name and city for everyone that’s in the table.

6. Display name and city that is not from Eskişehir.

7. Display all columns for everyone that has salary over 2500,00.

8. Display name for everyone whose name ends in an ‘an’.

9. Display all columns for everyone whose name contains "Yılmaz".

10. Everyone that's making under 2500,00 are to receive a 1200,00 a year raise. Update the records according to this.

11. Veli Doğan just quit, remove his record from the table.

12. Display distinct city in the Emp table.