CL-2005: Database Systems
Lab # 6: Introduction to SQL

Objective:

1-Database schema
2-Introduction to SQL (DML, DDL, DCL)
3-SQL sample commands and interactions
4-Learning and practice

Scope:

The student should know the following:

- Workaround SQL Server
- SQL Practice
- Basic exercises

```
Operators (IN, BETWEEN, LIKE, IS NULL)
ORDER BY with ASC and DESC sorting
Usage of AND, OR, NOT and their precedence
```

Useful Concepts:

Command level programming experience
How to see source data as a table
Table name, its column name, and column's datatypes

Discussion:

DML, DDL, DCL
Control Commands are Set, column, format, etc.

SCOTT Schema:

Scott is a database user used for demonstration purposes containing the famous **EMP, DEPT, BONUS,** and **SALGRADE** tables. According to legend, this account was named after Bruce Scott (co-author and co-architect of Oracle v1 to v3) and the password was the name of his daughter's cat, Tiger.

Create a SCOTT Schema:

To create a Scott schema, you have to write all the queries given below.

Employee Table (Creation):

```
CREATE TABLE EMP
       (EMPNO NUMERIC(4) NOT NULL,
        ENAME VARCHAR(10),
        JOB VARCHAR(9),
        MGR NUMERIC(4),
        HIREDATE DATE,
        SAL NUMERIC(7, 2),
        COMM NUMERIC(7, 2),
        DEPTNO NUMERIC(2));
```

Employee Table (Insertion):

```
INSERT INTO EMP VALUES
        (7369, 'SMITH','CLERK',7902, CONVERT(date,'17-DEC-1980'),800, NULL, 20);
INSERT INTO EMP VALUES
        (7499, 'ALLEN','SALESMAN',7698, CONVERT(date,'20-FEB-1981'), 1600,300, 30);
INSERT INTO EMP VALUES
        (7521, 'WARD','SALESMAN',7698, CONVERT(date,'22-FEB-1981'), 1250,500, 30);
INSERT INTO EMP VALUES
        (7566, 'JONES','MANAGER',7839, CONVERT(date,'2-APR-1981'), 2975, NULL, 20);
INSERT INTO EMP VALUES
        (7654, 'MARTIN','SALESMAN',7698, CONVERT(date,'28-SEP-1981'), 1250, 1400, 30);
INSERT INTO EMP VALUES
        (7698, 'BLAKE','MANAGER',7839, CONVERT (date,'1-MAY-1981'), 2850, NULL, 30);
INSERT INTO EMP VALUES
        (7782, 'CLARK','MANAGER',7839, CONVERT (date,'9-JUN-1981'), 2450, NULL, 10);
INSERT INTO EMP VALUES
        (7788, 'SCOTT','ANALYST',7566, CONVERT(date,'09-DEC-1982'), 3000, NULL, 20);
INSERT INTO EMP VALUES
        (7839, 'KING','PRESIDENT', NULL, CONVERT(date,'17-NOV-1981'), 5000, NULL, 10);
INSERT INTO EMP VALUES
        (7844, 'TURNER','SALESMAN',7698, CONVERT(date,'8-SEP-1981'), 1500, 0, 30);
INSERT INTO EMP VALUES
        (7876, 'ADAMS','CLERK',7788, CONVERT(date,'12-JAN-1983'), 1100, NULL, 20);
INSERT INTO EMP VALUES
        (7900, 'JAMES','CLERK',7698, CONVERT(date,'3-DEC-1981'), 950, NULL, 30);
INSERT INTO EMP VALUES
        (7902, 'FORD','ANALYST',7566, CONVERT(date,'3-DEC-1981'), 3000, NULL, 20);
INSERT INTO EMP VALUES
        (7934, 'MILLER', 'CLERK',7782, CONVERT(date,'23-JAN-1982'), 1300, NULL, 10);
```

Department Table (Creation):

```
CREATE TABLE DEPT
       (DEPTNO NUMERIC(2),
        DNAME VARCHAR(14),
        LOC VARCHAR(13));
```

Department Table (Insertion):

```
INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');
INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');
INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');
```

Bonus Table (Creation):

```
CREATE TABLE BONUS
       (ENAME VARCHAR(10),
        JOB   VARCHAR(9),
        SAL   NUMERIC,
        COMM NUMERIC);
```

Salary Grade Table (Creation):

```
CREATE TABLE SALGRADE
        (GRADE NUMERIC,
         LOSAL NUMERIC,
         HISAL NUMERIC);
```

Salary Grade Table (Insertion):

```
INSERT INTO SALGRADE VALUES (1, 700, 1200);
INSERT INTO SALGRADE VALUES (2, 1201, 1400);
INSERT INTO SALGRADE VALUES (3, 1401, 2000);
INSERT INTO SALGRADE VALUES (4, 2001, 3000);
INSERT INTO SALGRADE VALUES (5, 3001, 9999);
```

Dummy Table (Creation):

```
CREATE TABLE DUMMY
        (DUMMY NUMERIC);
```

Dummy Table (Insertion):

```
INSERT INTO DUMMY VALUES (0);
```

Tada Scott Schema is created successfully 😊

Simple SQL Commands:

To see the total table, count in the database

```
USE MyDatabase
SELECT COUNT(*)
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE';
```

To see description of tables in the database

```
EXEC sp_help 'dbo.mytable';

EXEC sp_columns mytable;

SELECT * FROM information_schema.columns
WHERE table_name = ' mytable';
```

To see the description of the employee table in the database

```
EXEC sp_help 'dbo.EMP';

EXEC sp_columns EMP;

SELECT * FROM information_schema.columns
WHERE table_name = 'EMP';
```

From the output of the above command choose column names and make SQL as

```
Select empno,ename,sal
from EMP;

Select hiredate
from EMP;
```

To see the description of the department table in the database

```
EXEC sp_help 'dbo.DEPT';

EXEC sp_columns DEPT;

SELECT * FROM information_schema.columns
WHERE table_name = 'DEPT';
```

From the output of the above command choose column names and make SQL as

```
Select dname,loc
from  DEPT;
```

Exercises:

```
Select *
from EMP;

Select job
from EMP;

Select distinct job
from EMP;
```

How a question can be asked?

Question:

Display job, hiredate, a salary of all employees order by department number?

Solution:

First to find out an exact table or tables provides required columns by writing the following SQL

```
SELECT
    *
FROM
    information_schema.tables;
```

Then to see exact names of column(s) from a table (in this case table you decided is EMP) , run command as

```
EXEC sp_help 'dbo.EMP';

EXEC sp_columns EMP;

SELECT * FROM information_schema.columns
WHERE table_name = 'EMP';
```

Based upon the above SQLs formulate a SQL statement as follows

```
Select job, hiredate, sal
from EMP
order by deptno;
```

Question:

List of employee name, hiring date, job title, commission, and salary of those employees who are clerks.

```
Select ename,sal,deptno
from EMP
order by ename;

Select job,deptno,sal
from EMP
order by job asc,deptno desc;

Select ename,hiredate,job,comm,sal
from EMP
where job='CLERK';
```

Strings enclosed in a single quotation are **Case Sensitive**.

```
Select ename,hiredate,job,comm,sal
from EMP
where job='CLERK'
AND
sal > 1000;
```

For **WHERE** you can use operators **IN, BETWEEN, LIKE, IS NULL**

Question:

List of employees who may CLERK, MANAGER, ANALYST having salary below 1200.

```
Select *
from EMP
where job='CLERK' OR
job='MANAGER' OR
job='ANALYST' OR
sal <1200;

Select *
from EMP
where job IN('CLERK','MANAGER','ANALYST')
AND
sal <1200;

Select *
from EMP
where (job='CLERK' AND sal<1200) OR
         (job='MANAGER'AND sal<1200) OR
         (job='ANALYST' AND sal<1200);

Select *
from EMP
where (job='CLERK' OR
          job='MANAGER' OR
       job='ANALYST') AND
       sal <1200;
```

Question:

List of employees having salary ranges from 1000 to 3000.

```
Select *
from EMP
where sal BETWEEN 1000 AND 3000;

Select *
from EMP
where sal>= 1000 AND sal<=3000;

Select *
from EMP
where deptno IN (10,30);
```

Like (% and _)

```
Select *
from EMP
where ename LIKE 'M%'
order by ename;

Select *
from EMP
where ename LIKE 'MAR%';

Select *
from EMP
where ename LIKE 'MARTI_';

Select *
from emp
where ename like '_A%';

Select *
from emp
where ename like '%A%';

Select *
from emp
where ename NOT like '%A%';

Select *
from emp
where comm IS NULL;

select *
from emp
where comm = NULL;  -- does not give you any output
```

Exercise and Logical Evaluation:

```
Select *
from EMP
where job='CLERK' OR
job='MANAGER' AND
sal < 1200;

Select *
from EMP
where job='CLERK' OR
(job='MANAGER' AND
sal < 1200);

Select *
from EMP
where (job='CLERK' OR
job='MANAGER') AND
sal < 1200;
```

List all the employee names, their salaries and add an increment of 300 in the salaries.

List all the employee names, jobs, and salaries who have not been given any commission.

List all the job titles one time

List all the employee identity numbers, names, jobs, and salaries are greater than 1500, and job title has MAN keywords.

List all the employee names and jobs who are not CLERK, ANALYST, and SALESMAN.

List all the employee's information by their hiring dates.

Evaluation:

Your Lab Work grade will depend on your active participation, individual efforts in solving Lab Problem and Seriousness during the lab.