



Database Management System (DBMS – 204)

Experiment # 01

Writing Basic SQL SELECT Statements

Student Name: Kabeer Ahmed

Roll Number: SE-19028

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name: Engr. Adiba Jafar

Signature and Date:

OUTCOMES

After completing this lesson, you should be able to do the following:

- List the capabilities of SQL SELECT statements
- Execute a basic SELECT statement
- Differentiate between SQL statements and iSQL*Plus commands

THEORY

Capabilities of SQL SELECT Statements

- 1. Projection**
- 2. Selection**
- 3. Join**

A **SELECT** statement retrieves information from the database. Using a SELECT statement, you can do the following:

- **Projection:** You can use the projection capability in SQL to choose the columns in a table that you want returned by your query. You can choose as few or as many columns of the table as you require.
- **Selection:** You can use the selection capability in SQL to choose the rows in a table that you want to returned by a query. You can use various criteria to restrict the rows that you see.
- **Joining:** You can use the join capability in SQL to bring together data that is stored in different tables by creating a link between them. You learn more about joins in a later lesson.

Basic SELECTS Statement:

SELECT *|{[DISTINCT] column | expression [alias],...} FROM
table.

- **SELECT identifies what columns**
 - **FROM identifies which table**
- (*)Selecting All Columns
SELECT * FROM dept;

Selecting Specific Columns:

SELECT deptno, loc FROM dept;

Arithmetic Expressions

- *Create expressions with number and date data by*
- *using arithmetic operators.*
- | <i>Operator</i> | <i>Description</i> |
|-----------------|--------------------|
| • + | Add |
| • - | Subtract |
| • * | Multiply |
| • / | Divide |

Using Arithmetic Operators

```
SELECT ename, sal, sal + 300 FROM emp;
```

Operator Precedence

- / * + -
- Multiplication and division take priority over addition and subtraction.
- Operators of the same priority are evaluated from left to right.
- Parentheses are used to force prioritized evaluation and to clarify statements.
- Operator Precedence
SELECT ename, sal, 12*sal+100 FROM emp;

Using Parentheses.

```
SELECT ename, sal, 12*(sal+100) FROM emp;
```

Defining a Null Value

- A null is a value that is unavailable, unassigned, unknown, or inapplicable.
- A null is not the same as zero or a blank space.
SELECT ename, empno, sal, comm FROM emp;

Null Values in Arithmetic Expressions

Arithmetic expressions containing a null value evaluate to null.

```
SELECT ename, 12*sal*comm FROM emp;
```

Defining a Column Alias

A column alias:

- Renames a column heading
- Is useful with calculations
- Immediately follows the column name: there can also be the optional AS keyword between the column name and alias
- Requires double quotation marks if it contains spaces or special characters or is case sensitive

Using Column Aliases

```
SELECT ename AS name, comm FROM emp;
```

NAME	COMM
King	
Kochhar	

```
SELECT ename "Name", sal*12 "Annual Salary" FROM emp;
```

Name	Annual Salary
King	288000
Kochhar	204000

A concatenation operator:

- Concatenates columns or character strings to other columns.
- Is represented by two vertical bars (||)
- Creates a resultant column that is a character Expression.
- A literal value is a character, a number, or a date included in the SELECT list.
- Date and character literal values must be enclosed within single quotation marks.
- Each character string is output once for each row returned.

SELECT ename||job AS "Employees" FROM emp;

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP
HunoldIT_PROG

Using Literal Character Strings

**SELECT ename || ' is a '||job AS "Employee Details"
FROM emp;**

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG

Duplicate Rows:

The default display of queries is all rows, including duplicate rows.

SELECT deptno AS DEPARTMENT_ID FROM emp;

DEPARTMENT_ID
10
20
50
60
80
90
110

8 rows selected.

Eliminating Duplicate Rows

**Eliminate duplicate rows by using the
DISTINCT keyword in the SELECT clause.**

- **SELECT DISTINCT deptno FROM emp;**

SQL and SQL*Plus Interaction

SQL statements SQL*Plus Oracle Internet server Browser
Query results SQL*Plus commands Formatted report Client

SQL and SQL*Plus :SQL is a command language for communication with the Oracle server from any tool or application. Oracle SQL contains many extensions. SQL*Plus is an Oracle tool that recognizes and submits SQL statements to the Oracle server for execution and contains its own command language.

Features of SQL

- Can be used by a range of users, including those with little or no programming experience
- Is a nonprocedural language
- Reduces the amount of time required for creating and maintaining systems
- Is an English-like language

Features of SQL*Plus

- Accessed from a browser
- Accepts ad hoc entry of statements
- Provides online editing for modifying SQL statements
- Controls environmental settings
- Formats query results into a basic report
- Accesses local and remote databases

SQL Statements versus iSQL*Plus Commands

SQL iSQL*Plus

- A language
- An environment
- ANSI standard
- Oracle proprietary
- Keywords can be abbreviated
- Statements manipulate
- Commands do not allow data and table definitions manipulation of values in the database.
- Runs on a browser
- Centrally loaded, does not have to be implemented on each machine.

LAB # 01

Writing Basic SQL SELECT Statements

Paper-Based Questions

The following SELECT statement executes successfully:

1. `SELECT ename, job, sal AS Salary FROM emp;`

Ans: True

2. `SELECT * FROM salgrades;`

Ans: False

Practice Problem

1. There are four coding errors in this statement. Can you identify them?

`SELECT empno, ename sal x 12 ANNUAL SALARY FROM emp;`

ANS.

- The operator for multiplication is * whereas in the code it is 'x' which is wrong.
- The ANNUAL SALARY cannot not have space between it. It should be ANNUAL_SALARY.
- There is no 'AS' rename operation after sal x 12.
- There is a comma missing after ename.

2. Show the structure of the Dept table. Select all data from the table

ANS.

```
SQL> DESCRIBE DEPT
Name                               Null?    Type
-----
DEPTNO                             NOT NULL NUMBER
DNAME                              VARCHAR2(20)
LOC                                 VARCHAR2(20)

SQL> select * from dept;

DEPTNO DNAME                LOC
-----
10 ACCOUNTING              NEW YORK
20 RESEARCH                 DALLAS
30 SALES                     CHICAGO
40 OPERATIONS               BOSTON
```

3. Show the structure of the EMP table. Create a query to display the ename, job , hiredate, and employee number for each employee, with employee number appearing first. Save your SQL statement to a file named lab1_7.sql.

ANS.

```
SQL> DESCRIBE emp;
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER
ENAME		VARCHAR2(20)
JOB		VARCHAR2(20)
MGR		NUMBER
HIREDATE		DATE
SAL		NUMBER
COMM		NUMBER
DEPTNO		NUMBER

```
SQL> select empno, ename, job, hiredate from emp;
```

EMPNO	ENAME	JOB	HIREDATE
7369	SMITH	CLERK	17-DEC-80
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7788	SCOTT	ANALYST	09-DEC-82
7839	KING	PRESIDENT	17-NOV-81
7844	TURNER	SALESMAN	08-SEP-81
7876	ADAMS	CLERK	12-JAN-83
7900	JAMES	CLERK	03-DEC-81
7902	FORD	ANALYST	03-DEC-81
7902	MILLER	CLERK	23-JAN-82

14 rows selected.

```
SQL> save e://lab1_7.sql
```

```
Created file e://lab1_7.sql
```

4. Run your query in the file lab1_7.sql.

ANS.

```
SQL> run e://lab1_7.sql
1* select empno, ename, job, hiredate from emp
```

EMPNO	ENAME	JOB	HIREDATE
7369	SMITH	CLERK	17-DEC-80
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7788	SCOTT	ANALYST	09-DEC-82
7839	KING	PRESIDENT	17-NOV-81
7844	TURNER	SALESMAN	08-SEP-81
7876	ADAMS	CLERK	12-JAN-83

EMPNO	ENAME	JOB	HIREDATE
7900	JAMES	CLERK	03-DEC-81
7902	FORD	ANALYST	03-DEC-81
7902	MILLER	CLERK	23-JAN-82

```
14 rows selected.
```

5. Load lab1_7.sql into the SQL buffer. Name the column headings Emp#. Employee, Job and Hire Date, Respectively. Return your query.

ANS.

```
SQL> edit e://lab1_7.sql;

SQL> get e://lab1_7.sql
1* select empno AS Emp#, ename AS Employee, job, hiredate from emp
SQL>
```



```
SQL> @ e://lab1_7.sql;
```

EMP#	EMPLOYEE	JOB	HIREDATE
7369	SMITH	CLERK	17-DEC-80
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7788	SCOTT	ANALYST	09-DEC-82
7839	KING	PRESIDENT	17-NOV-81
7844	TURNER	SALESMAN	08-SEP-81
7876	ADAMS	CLERK	12-JAN-83

EMP#	EMPLOYEE	JOB	HIREDATE
7900	JAMES	CLERK	03-DEC-81
7902	FORD	ANALYST	03-DEC-81
7902	MILLER	CLERK	23-JAN-82

14 rows selected.

6. Create a query to display unique job codes from the EMPLOYEES table.

ANS.

```
SQL> select distinct mgr from emp;
```

MGR
7788
7782
7698
7902
7566
7839

7 rows selected.

7. Copy the statement from lab1_7.sql into the iSQL*Plus Edit window. Name the column headings Emp #,Employee,Job, and Hire Date, respectively. Run your query again.

ANS.

```
SQL> edit e://lab1_7.sql;

SQL> get e://lab1_7.sql
  1* select empno AS Emp#, ename AS Employee, job, hiredate from emp
SQL>
```

```
SQL> @ e://lab1_7.sql;

      EMP# EMPLOYEE              JOB              HIREDATE
-----
      7369 SMITH                CLERK              17-DEC-80
      7499 ALLEN                SALESMAN            20-FEB-81
      7521 WARD                 SALESMAN            22-FEB-81
      7566 JONES                MANAGER            02-APR-81
      7654 MARTIN               SALESMAN            28-SEP-81
      7698 BLAKE                MANAGER            01-MAY-81
      7782 CLARK                MANAGER            09-JUN-81
      7788 SCOTT                ANALYST            09-DEC-82
      7839 KING                 PRESIDENT           17-NOV-81
      7844 TURNER               SALESMAN            08-SEP-81
      7876 ADAMS                CLERK              12-JAN-83

      EMP# EMPLOYEE              JOB              HIREDATE
-----
      7900 JAMES                CLERK              03-DEC-81
      7902 FORD                 ANALYST            03-DEC-81
      7902 MILLER               CLERK              23-JAN-82

14 rows selected.
```

8. Display the employee name concatenated with the job ID, separated by a comma and space, and name the column Employee and Title

ANS.

```
SQL> edit e://lab1_7.sql;

SQL> get e://lab1_7.sql;
  1  select ename||', '||job "Employee and Title"
  2* from emp;
SQL> @e://lab1_7.sql;

Employee and Title
-----
SMITH, CLERK
ALLEN, SALESMAN
WARD, SALESMAN
JONES, MANAGER
MARTIN, SALESMAN
BLAKE, MANAGER
CLARK, MANAGER
SCOTT, ANALYST
KING, PRESIDENT
TURNER, SALESMAN
ADAMS, CLERK

Employee and Title
-----
JAMES, CLERK
FORD, ANALYST
MILLER, CLERK

14 rows selected.
```

9. Create a query to display all the data from the EMP table. Separate each column by a comma. Name the column THE_OUTPUT.

ANS.

```
SQL> get e://lab1_7.sql;
  1  select empno||', '||ename||', '||job||', '||mgr||', '||hiredate||', '||sal||', '||comm||', '||deptno "THE OUTPUT"
  2* from emp;
SQL> @e://lab1_7.sql;

THE OUTPUT
-----
7369, SMITH, CLERK, 7902, 17-DEC-80, 800, , 20
7499, ALLEN, SALESMAN, 7698, 20-FEB-81, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 22-FEB-81, 1250, 500, 30
7566, JONES, MANAGER, 7839, 02-APR-81, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 28-SEP-81, 1250, 1400, 30
7698, BLAKE, MANAGER, 7839, 01-MAY-81, 2850, , 30
7782, CLARK, MANAGER, 7839, 09-JUN-81, 2450, , 10
7788, SCOTT, ANALYST, 7566, 09-DEC-82, 3000, , 20
7839, KING, PRESIDENT, , 17-NOV-81, 5000, , 10
7844, TURNER, SALESMAN, 7698, 08-SEP-81, 1500, 0, 30
7876, ADAMS, CLERK, 7788, 12-JAN-83, 1100, , 20

THE OUTPUT
-----
7900, JAMES, CLERK, 7698, 03-DEC-81, 950, , 30
7902, FORD, ANALYST, 7566, 03-DEC-81, 3000, , 20
7902, MILLER, CLERK, 7782, 23-JAN-82, 1300, , 10

14 rows selected.
```