**UIT UNIVERSITY**

**Department of Computer Science**

**SET211 Database Systems/ CET224 Database Applications**

Lab#2

**Objective:**

Basic data retrieval operations in SQL\*Plus.

**Name of Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Sec. \_\_\_\_\_\_\_\_\_\_\_**

**Date of Experiment: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Marks Obtained/Remarks: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THEORY**

**SELECT Statement**

To extract data from the database, the SQL SELECT statement is used.

**Capabilities of SELECT statement**

Following are the various operations that can be performed using SELECT:-

i. **Selection**: The selection capability can be used to choose rows in a table depending on the criteria to selectively restrict the rows.

**Examples**

i. Selecting all employees whose salary is between 3500 and 5000 and who were hired after 31st July, 1981.

SELECT \*

FROM EMP

WHERE (SAL BETWEEN 3500 AND 5000) AND HIREDATE >

TO\_DATE('31-JUL-1981', 'DD-MON-YYYY');

SELECT \*

FROM EMP

WHERE (SAL BETWEEN 3500 AND 5000) AND HIREDATE >

TRY\_CONVERT(DATETIME, '31-JUL-1981', 102);

ii. Selecting all employees whose job is either clerk or analyst and were hired between 23rd July, 1981 and 14th May, 1982.

SELECT \* FROM EMP

WHERE (JOB = 'CLERK' OR JOB = 'ANALYST') AND HIREDATE BETWEEN TO\_DATE('23-JUL-1981', 'DD-MON-YYYY') AND

TO\_DATE('14-MAY-1982', 'DD-MON-YYYY');

SELECT \* FROM EMP

WHERE (JOB = 'CLERK' OR JOB = 'ANALYST') AND HIREDATE BETWEEN TRY\_CONVERT(DATETIME, '23-JUL-1981') AND

TRY\_CONVERT(DATETIME, '14-MAY-1982');



**Figure 2.1: Data in a single table can be useful for several employees**

ii. **Projection**: It refers to choosing the columns in a table that are to be returned by a query. We can choose as few or as many columns of the table as we require.

**Examples**

i. Selecting employee number, name and their job

SELECT EMPNO, ENAME, JOB

FROM EMP;

ii. Selecting employee number, name and their salary who do not earn commission

SELECT EMPNO, ENAME, SAL

FROM EMP

WHERE COMM IS NULL;

iii. **Join**: To bring together data that is stored in different tables by creating a link through a column that both the tables share.

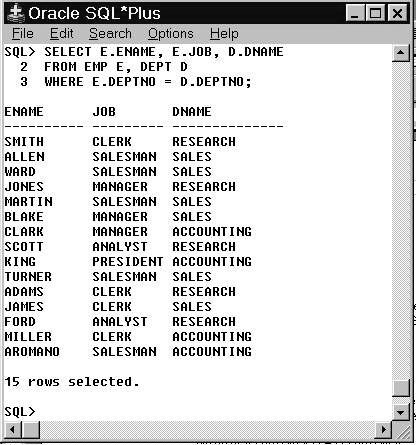
**Example**

To retrieve the employee name, their job and department name, we need to extract data from two tables, EMP and DEPT. This type of join is called *equijoin*-that is, values in the DEPTNO column on both tables must be equal. Equijoin is also called *simple join* or *inner join*. The output is shown in figure 2.2.

SELECT E.ENAME, E.JOB, D.DNAME

FROM EMP E, DEPT D

WHERE E.DEPTNO = D.DEPTNO;



**Figure 2.2: Joining tables using equi-join**

**NOTE:** Different join operations are discussed in detail in lab session 3.

**Comparison Operators**

Comparison operators are used in conditions that compare one expression to another. They are used in the WHERE or HAVING clause of the SELECT statement.

|  |  |
| --- | --- |
| **Operator** | **Meaning** |
| = | Equal to |
| > | Greater than |
| >= | Greater than or equal to |
| < | Less than |
| <= | Less than or equal to |
| <> | Not equal to |

**Table 2.1**

Besides basic comparison operators (>, <, >=, <=, =, <>), Oracle SQL also supports following comparison operators:-

|  |  |
| --- | --- |
| **Operator** | **Meaning** |
| BETWEEN … AND … | Between two values (inclusive) |
| IN (list) | Match any of a list of values |
| LIKE | Match a character pattern |
| IS NULL | Is a null value |

**Table 2.2**

**Examples**

1. To display record of employees who are not managers.

SELECT \* FROM EMP

WHERE JOB <> ‘MANAGER’;

1. To display the employee number, name, salary and the manager’s employee number of all the employees whose manager’s employee number is 7902, 7566, or 7788.

SELECT EMPNO, ENAME, SAL, MGR

FROM EMP

WHERE MGR IN (7902, 7566, 7788);

1. To display the names of all employees with names starting with **S,**

SELECT ENAME

FROM EMP

WHERE ENAME LIKE ‘S%’;

**Note**: Above query performs wildcard searches using LIKE operator. Here % symbol represents any sequence of zero or more characters.

1. To display the names of all employees with second character of name as **A,**

SELECT ENAME

FROM EMP

WHERE ENAME LIKE ‘\_A%’;

**Note**: Here \_ character represents any single character

**Logical Operators**

A logical operator combines the result of two component conditions to produce a single result based on them or to invert the result of a single condition. Three logical operators are available in SQL as shown below:-

|  |  |
| --- | --- |
| **Operator** | **Meaning** |
| AND | Returns TRUE if both component conditions are TRUE |
| OR | Returns TRUE if either component condition is TRUE |
| NOT | Returns TRUE if the following condition is FALSE |

**Table 2.3**

**Examples**

* To display record of all clerks who earn more than 1100

SELECT empno, ename, job, sal

FROM emp

WHERE sal >= 1100

AND job = ‘CLERK’;

* To display record of all employees who are either clerks or earn more than 1100.

SELECT empno, ename, job, sal

FROM emp

WHERE sal >= 1100

OR job = ‘CLERK’;

* To display name and job title of all the employees whose are not CLERK,

MANAGER, or ANALYST.

SELECT ename, job

FROM emp

WHERE job NOT IN (‘CLERK’, ‘MANAGER’, ‘ANALYST’);

**Rules of Precedence**

|  |  |
| --- | --- |
| **Order**  **Evaluated** | **Operator** |
| 1 | All comparison operators |
| 2 | NOT |
| 3 | AND |
| 4 | OR |

**Table 2.4**

For example, consider the following statement:- SELECT ename, job, sal FROM emp

WHERE job = ‘SALESMAN’

OR job = ‘PRESIDENT’

AND sal > 1500;

In the above example, there are two conditions:

* The first condition is that job is SALESMAN.
* The second condition is that job is CLERK and salary is greater than 1000.

Therefore the SELECT statement reads as follows:-

*Select the row if an employee is a SALESMAN or an employee is a CLERK and earns more than 1000*.

In order to force the OR operator to be evaluated before AND, use parentheses as follows:-.

SELECT ename, job, sal FROM emp

WHERE (job = ‘SALESMAN’

OR job = ‘PRESIDENT’)

AND sal > 1500;

**Ordering Data**

The order of rows returned in a query result is undefined. The ORDER BY clause can be used to sort the rows. This clause comes last in the SELECT statement. ASC at the end of the ORDER BY clause specifies ascending order where as DESC specifies descending order. ASC is the default order.

**Examples**

1. To select data in the increasing order of hiredate,

SELECT ENAME, JOB, DEPTNO, HIREDATE

FROM EMP

ORDER BY HIREDATE;

1. To select data in the decreasing order of hiredate,

SELECT ENAME, JOB, DEPTNO, HIREDATE

FROM EMP

ORDER BY HIREDATE DESC;

1. To sort by column alias,

SELECT EMPNO, ENAME, SAL\*12 ANNSAL FROM EMP

ORDER BY ANNSAL.

1. To sort by multiple columns,

SELECT ENAME, DEPTNO, SAL

FROM EMP

ORDER BY DEPTNO, SAL DESC;

**Note**: The DESC applies only to SAL column. The DEPTNO appears in ascending order.

1. To select list of names and jobs of all employees hired in 1987 in the alphabetical order of name

SELECT UPPER(ENAME) “EMP NAME”, JOB

FROM EMP

WHERE TO\_CHAR(HIREDATE, ‘YYYY’) = 1987

ORDER BY ENAME;

1. To print employee number, name, job, annual salary of all managers and clerks whose monthly salary is between 3000 and 5500 in descending order of annual salary.

SELECT EMPNO, ENAME, JOB, 12\*SAL + NVL(COMM, 0)

ANNUAL\_SALARY

FROM EMP

WHERE JOB = ‘MANAGER’ OR JOB = ‘CLERK’

AND SAL BETWEEN 3000 AND 5500

ORDER BY ANNUAL\_SALARY DESC;

**EXERCISES**

1. Define the different capabilities of SELECT statement? Give an example of each.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Write down SQL queries to perform following functions:-
   1. To display the name and department number of employee with number 7566.

* 1. To display the name and department number of all employees in departments 10 and 30 in alphabetical order by name.

* 1. To display the name, department number and hire date of all employees who were hired in *1982*.

* 1. To display the name of all employees who have two consecutive Ls in their name and are in department 30 or their manager is 7782

* 1. To display the name of all clerks of department 10 and 20 hired before 1983.

* 1. Display the name and salary for all employees whose salary is not in range of $1500 and $2850.

* 1. Display the name, salary and commission for all employees whose commission amount in greater than their salary increased by 10%.

\* \* \* \* \*