

USMAN INSTITUTE OF TECHNOLOGY

Department of Computer Science CS311 Introduction to Database Systems

Lab#2

Objective:

Basic data retrieval operations in SQL*Plus.

Name of Student: Muhammad Waleed

Roll No: 20b-115-SE Sec. A

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:-

1. **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');
```

- 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');
```



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

2. **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
```

Lab No. 2

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;
```

3. **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;
```

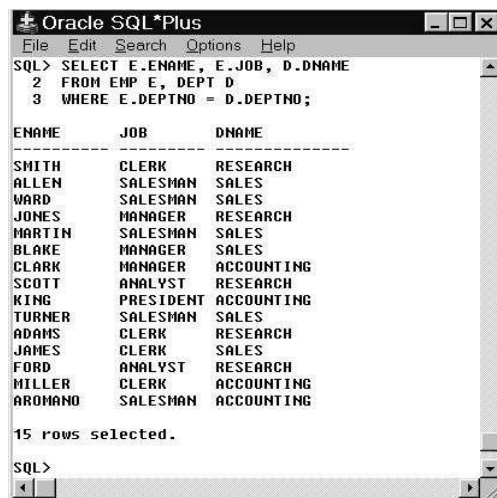


Figure 2.2: Joining tables using equi-join

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

Lab No. 2

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

- i. To display record of employees who are not managers.

```
SELECT * FROM EMP WHERE JOB <> 'MANAGER';
```

- ii. 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);
```

- iii. 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.

- iv. 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

Lab No. 2

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:-

Lab No. 2

```
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

- i. To select data in the increasing order of hiredate,

```
SELECT ENAME, JOB, DEPTNO, HIREDATE
FROM EMP
ORDER BY HIREDATE;
```

- ii. To select data in the decreasing order of hiredate,

```
SELECT ENAME, JOB, DEPTNO, HIREDATE
FROM EMP
ORDER BY HIREDATE DESC;
```

- iii. To sort by column alias,

```
SELECT EMPNO, ENAME, SAL*12 ANNSAL FROM EMP
ORDER BY ANNSAL;
```

- iv. 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.

- v. 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;
```

Lab No. 2

- vi. 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.

Following are the various operations that can be performed using SELECT.

1. Selection: The selection capability can be used to choose rows in a table depending on the criteria to selectively restrict the rows.
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');
```

2. 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.
Selecting employee number, name and their job

```
SELECT EMPNO, ENAME, JOB
FROM EMP;
```

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

```
SELECT E.ENAME, E.JOB, D.DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO;
```

-
2. Write down SQL queries to perform following functions:-

- i. To display the name and department number of employee with number 7566.

Lab No. 2

Worksheet Query Builder

```
select ename, deptno from emp where empno=7566;
```

Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 1 in 0.002 seconds

	ENAME	DEPTNO
1	JONES	20

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

Worksheet Query Builder

```
select ename, deptno from emp where deptno =10 or deptno=30 order by ename;
```

Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 9 in 0.026 seconds

	ENAME	DEPTNO
1	ALLEN	30
2	BLAKE	30
3	CLARK	10
4	JAMES	30
5	KING	10
6	MARTIN	30
7	MILLER	10
8	TURNER	30
9	WARD	30

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

Start Page x Ahtisham x

Worksheet Query Builder

```
select ename, deptno, hiredate from emp where hiredate BETWEEN '1-jan-1982' and '31-dec-1982';
```

Query... x

SQL | All Rows Fetched: 2 in 0.006 seconds

	ENAME	DEPTNO	HIREDATE
1	SCOTT	20	09-DEC-82
2	MILLER	10	23-JAN-82

- iv. 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.

Start Page x Ahtisham x

Worksheet Query Builder

```
select ename from emp where ename like '%LL%' and deptno =30 or mgr=7782;
```

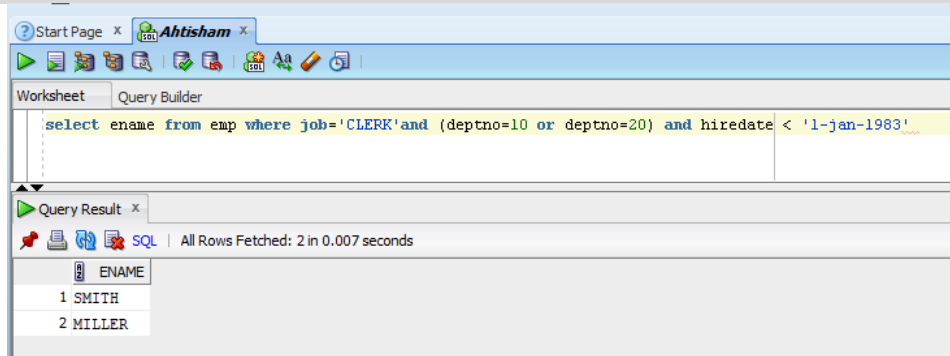
Query Result x

SQL | All Rows Fetched: 1 in 0.032 seconds

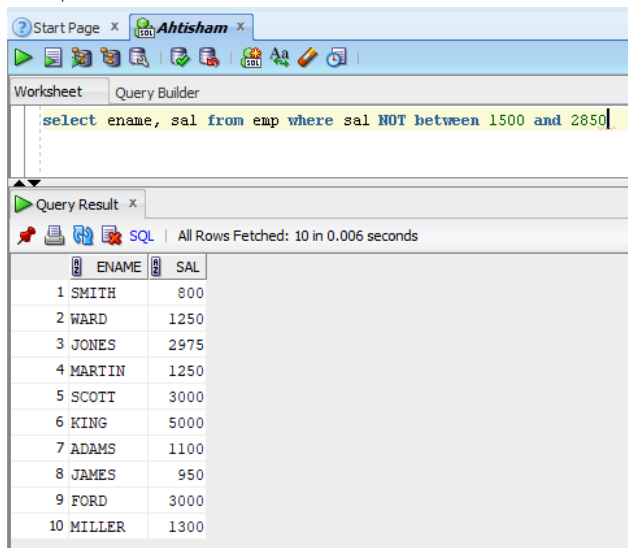
	ENAME
1	ALLEN

- v. To display the name of all clerks of department 10 and 20 hired before 1983.

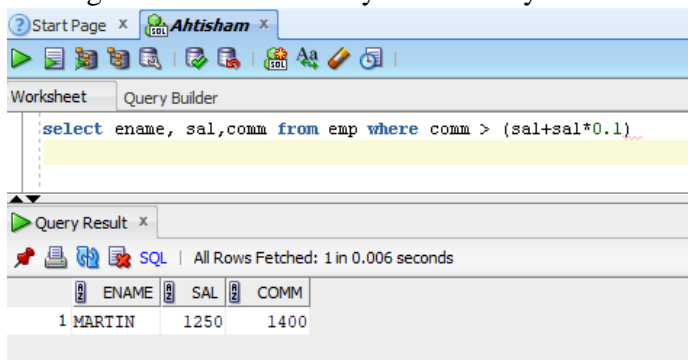
Lab No. 2



- vi. Display the name and salary for all employees whose salary is not in range of \$1500 and \$2850.



- vii. Display the name, salary and commission for all employees whose commission amount is greater than their salary increased by 10%.

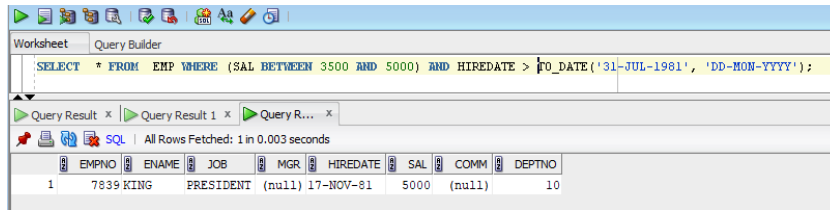


EXAMPLE

Lab No. 2

SELECTION

1)



Worksheet Query Builder

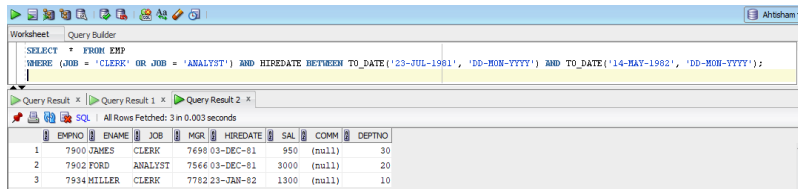
```
SELECT * FROM EMP WHERE (SAL BETWEEN 3500 AND 5000) AND HIREDATE > TO_DATE('31-JUL-1981', 'DD-MON-YYYY');
```

Query Result x Query Result 1 x Query R... x

All Rows Fetched: 1 in 0.003 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10

2)



Worksheet Query Builder

```
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');
```

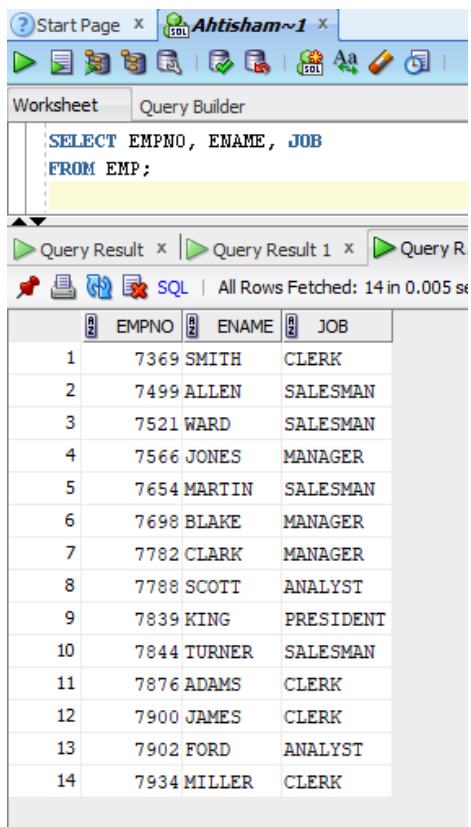
Query Result x Query Result 1 x Query Result 2 x

All Rows Fetched: 3 in 0.003 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7900	JAMES	CLERK	7690	03-DEC-81	950	(null)	30
2	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
3	7934	MILLER	CLERK	7762	23-JAN-82	1300	(null)	10

PROJECTION

1)



Start Page x Ahtisham~1 x

Worksheet Query Builder

```
SELECT EMPNO, ENAME, JOB FROM EMP;
```

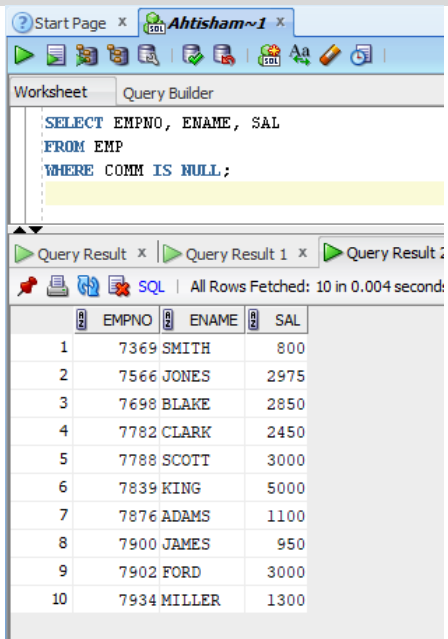
Query Result x Query Result 1 x Query R...

All Rows Fetched: 14 in 0.005 seconds

	EMPNO	ENAME	JOB
1	7369	SMITH	CLERK
2	7499	ALLEN	SALESMAN
3	7521	WARD	SALESMAN
4	7566	JONES	MANAGER
5	7654	MARTIN	SALESMAN
6	7698	BLAKE	MANAGER
7	7782	CLARK	MANAGER
8	7788	SCOTT	ANALYST
9	7839	KING	PRESIDENT
10	7844	TURNER	SALESMAN
11	7876	ADAMS	CLERK
12	7900	JAMES	CLERK
13	7902	FORD	ANALYST
14	7934	MILLER	CLERK

2)

Lab No. 2



The screenshot shows the SQL Developer interface with a query window titled 'Query Builder'. The query is:

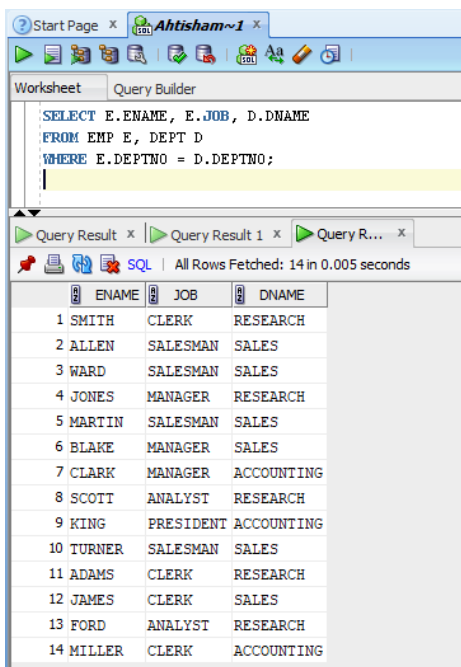
```
SELECT EMPNO, ENAME, SAL
FROM EMP
WHERE COMM IS NULL;
```

The query results are displayed in a table with 10 rows. The status bar indicates 'All Rows Fetched: 10 in 0.004 seconds'.

	EMPNO	ENAME	SAL
1	7369	SMITH	800
2	7566	JONES	2975
3	7698	BLAKE	2850
4	7782	CLARK	2450
5	7788	SCOTT	3000
6	7839	KING	5000
7	7876	ADAMS	1100
8	7900	JAMES	950
9	7902	FORD	3000
10	7934	MILLER	1300

JOIN

1)



The screenshot shows the SQL Developer interface with a query window titled 'Query Builder'. The query is:

```
SELECT E.ENAME, E.JOB, D.DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO;
```

The query results are displayed in a table with 14 rows. The status bar indicates 'All Rows Fetched: 14 in 0.005 seconds'.

	ENAME	JOB	DNAME
1	SMITH	CLERK	RESEARCH
2	ALLEN	SALESMAN	SALES
3	WARD	SALESMAN	SALES
4	JONES	MANAGER	RESEARCH
5	MARTIN	SALESMAN	SALES
6	BLAKE	MANAGER	SALES
7	CLARK	MANAGER	ACCOUNTING
8	SCOTT	ANALYST	RESEARCH
9	KING	PRESIDENT	ACCOUNTING
10	TURNER	SALESMAN	SALES
11	ADAMS	CLERK	RESEARCH
12	JAMES	CLERK	SALES
13	FORD	ANALYST	RESEARCH
14	MILLER	CLERK	ACCOUNTING

COMPARISON OPERATORS

1)

Lab No. 2

The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT * FROM EMP WHERE JOB = 'MANAGER';
```

The query has been executed, and the results are displayed in a table. The status bar indicates 'All Rows Fetched: 11 in 0.008 seconds'.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20
2	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
3	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
4	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
5	7788	SCOTT	ANALYST	7566	09-DEC-82	3000	(null)	20
6	7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10
7	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
8	7876	ADAMS	CLERK	7788	12-JAN-83	1100	(null)	20
9	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30
10	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
11	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10

2)

The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT EMPNO, ENAME, SAL, MGR FROM EMP WHERE MGR IN (7902,7566,7788);
```

The query has been executed, and the results are displayed in a table. The status bar indicates 'All Rows Fetched: 4 in 0.002 seconds'.

	EMPNO	ENAME	SAL	MGR
1	7369	SMITH	800	7902
2	7788	SCOTT	3000	7566
3	7876	ADAMS	1100	7788
4	7902	FORD	3000	7566

3)

The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

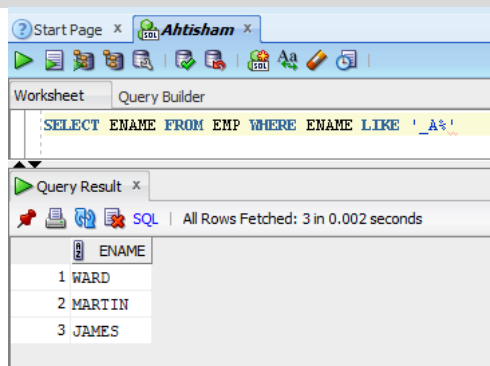
```
SELECT ENAME FROM EMP WHERE ENAME LIKE 'S%';
```

The query has been executed, and the results are displayed in a table. The status bar indicates 'All Rows Fetched: 2 in 0.025 seconds'.

	ENAME
1	SMITH
2	SCOTT

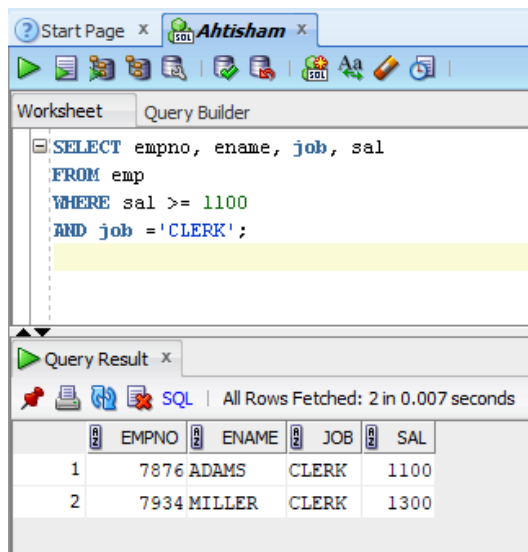
4)

Lab No. 2



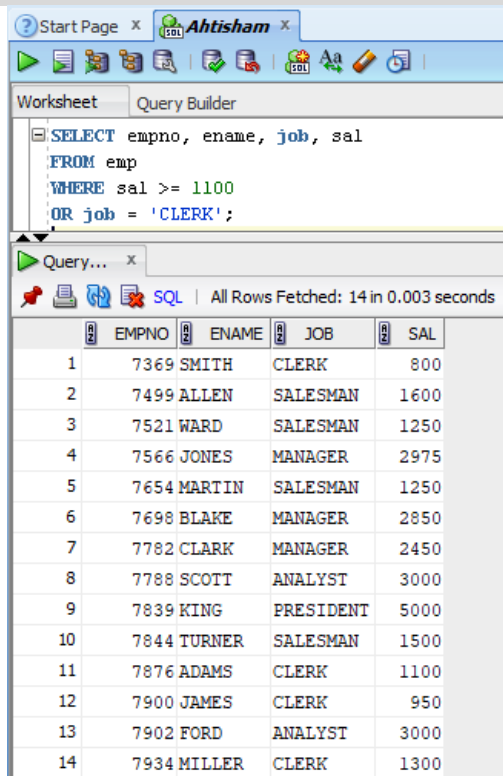
LOGICAL OPERATOR

1)



2)

Lab No. 2



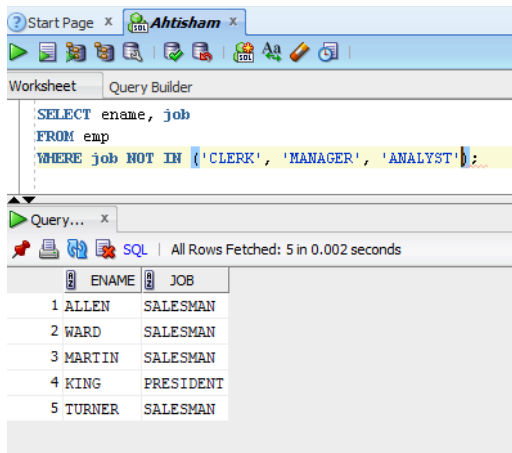
The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT empno, ename, job, sal
FROM emp
WHERE sal >= 1100
OR job = 'CLERK';
```

Below the query editor, the 'Query...' window shows the results of the query. The status bar indicates 'All Rows Fetched: 14 in 0.003 seconds'. The results are displayed in a table with the following columns: EMPNO, ENAME, JOB, and SAL.

	EMPNO	ENAME	JOB	SAL
1	7369	SMITH	CLERK	800
2	7499	ALLEN	SALESMAN	1600
3	7521	WARD	SALESMAN	1250
4	7566	JONES	MANAGER	2975
5	7654	MARTIN	SALESMAN	1250
6	7698	BLAKE	MANAGER	2850
7	7782	CLARK	MANAGER	2450
8	7788	SCOTT	ANALYST	3000
9	7839	KING	PRESIDENT	5000
10	7844	TURNER	SALESMAN	1500
11	7876	ADAMS	CLERK	1100
12	7900	JAMES	CLERK	950
13	7902	FORD	ANALYST	3000
14	7934	MILLER	CLERK	1300

3)



The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT ename, job
FROM emp
WHERE job NOT IN ('CLERK', 'MANAGER', 'ANALYST');
```

Below the query editor, the 'Query...' window shows the results of the query. The status bar indicates 'All Rows Fetched: 5 in 0.002 seconds'. The results are displayed in a table with the following columns: ENAME and JOB.

	ENAME	JOB
1	ALLEN	SALESMAN
2	WARD	SALESMAN
3	MARTIN	SALESMAN
4	KING	PRESIDENT
5	TURNER	SALESMAN

4)

Lab No. 2

The screenshot shows the SQL Developer interface with a query in the Query Builder. The query is:

```
SELECT ename, job, sal FROM emp
WHERE job = 'SALESMAN'
OR job = 'PRESIDENT'
AND sal > 1500;
```

The results pane shows 5 rows fetched in 0.002 seconds:

	ENAME	JOB	SAL
1	ALLEN	SALESMAN	1600
2	WARD	SALESMAN	1250
3	MARTIN	SALESMAN	1250
4	KING	PRESIDENT	5000
5	TURNER	SALESMAN	1500

5)

The screenshot shows the same SQL Developer interface with the same query. The results pane shows 1 row fetched in 0.006 seconds:

	ENAME	JOB	SAL
1	ALLEN	SALESMAN	1600

ORDERING DATA:

1)

The screenshot shows the SQL Developer interface with a query in the Query Builder. The query is:

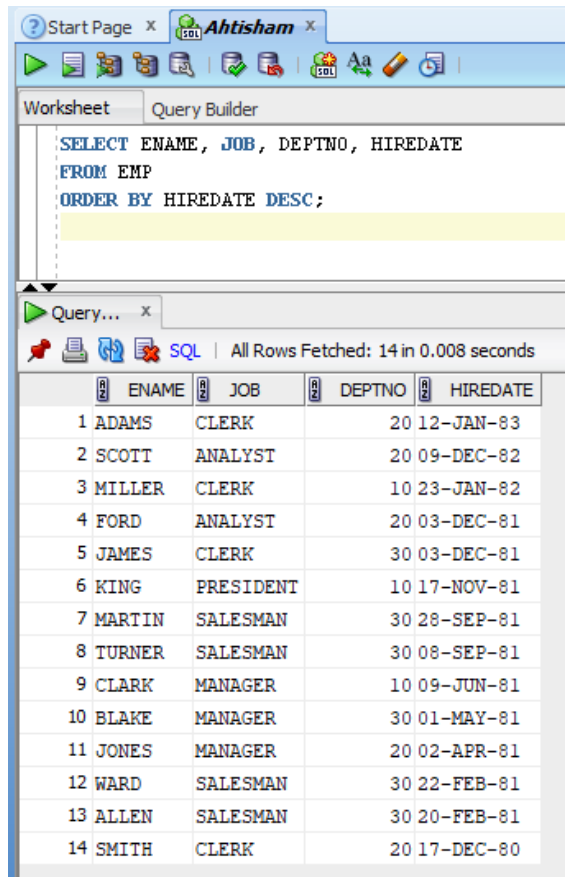
```
SELECT ENAME, JOB, DEPTNO, HIREDATE
FROM EMP
ORDER BY HIREDATE;
```

The results pane shows 14 rows fetched in 0.057 seconds:

	ENAME	JOB	DEPTNO	HIREDATE
1	SMITH	CLERK	20	17-DEC-80
2	ALLEN	SALESMAN	30	20-FEB-81
3	WARD	SALESMAN	30	22-FEB-81
4	JONES	MANAGER	20	02-APR-81
5	BLAKE	MANAGER	30	01-MAY-81
6	CLARK	MANAGER	10	09-JUN-81
7	TURNER	SALESMAN	30	08-SEP-81
8	MARTIN	SALESMAN	30	28-SEP-81
9	KING	PRESIDENT	10	17-NOV-81
10	JAMES	CLERK	30	03-DEC-81
11	FORD	ANALYST	20	03-DEC-81
12	MILLER	CLERK	10	23-JAN-82
13	SCOTT	ANALYST	20	09-DEC-82
14	ADAMS	CLERK	20	12-JAN-83

Lab No. 2

2)



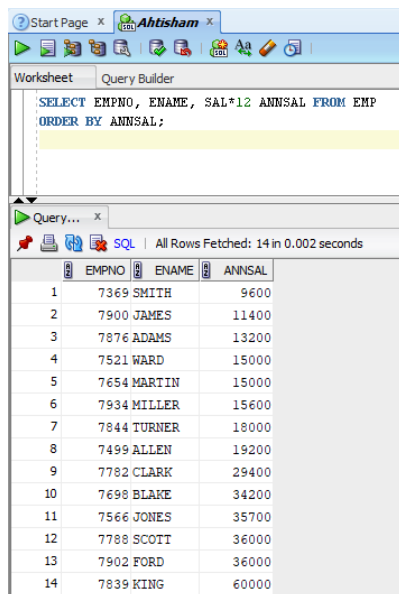
The screenshot shows the SQL Developer interface with a query window titled 'Query...'. The query is:

```
SELECT ENAME, JOB, DEPTNO, HIREDATE
FROM EMP
ORDER BY HIREDATE DESC;
```

The results are displayed in a table with 14 rows, ordered by hire date in descending order.

	ENAME	JOB	DEPTNO	HIREDATE
1	ADAMS	CLERK	20	12-JAN-83
2	SCOTT	ANALYST	20	09-DEC-82
3	MILLER	CLERK	10	23-JAN-82
4	FORD	ANALYST	20	03-DEC-81
5	JAMES	CLERK	30	03-DEC-81
6	KING	PRESIDENT	10	17-NOV-81
7	MARTIN	SALESMAN	30	28-SEP-81
8	TURNER	SALESMAN	30	08-SEP-81
9	CLARK	MANAGER	10	09-JUN-81
10	BLAKE	MANAGER	30	01-MAY-81
11	JONES	MANAGER	20	02-APR-81
12	WARD	SALESMAN	30	22-FEB-81
13	ALLEN	SALESMAN	30	20-FEB-81
14	SMITH	CLERK	20	17-DEC-80

3)



The screenshot shows the SQL Developer interface with a query window titled 'Query...'. The query is:

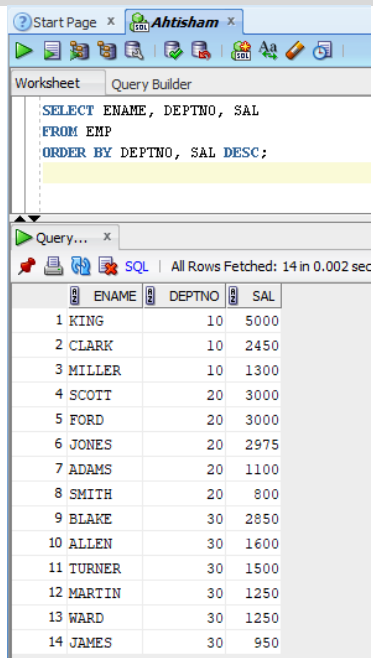
```
SELECT EMPNO, ENAME, SAL*12 ANNSAL FROM EMP
ORDER BY ANNSAL;
```

The results are displayed in a table with 14 rows, ordered by annual salary in ascending order.

	EMPNO	ENAME	ANNSAL
1	7369	SMITH	9600
2	7900	JAMES	11400
3	7876	ADAMS	13200
4	7521	WARD	15000
5	7654	MARTIN	15000
6	7934	MILLER	15600
7	7844	TURNER	18000
8	7499	ALLEN	19200
9	7782	CLARK	29400
10	7698	BLAKE	34200
11	7566	JONES	35700
12	7788	SCOTT	36000
13	7902	FORD	36000
14	7839	KING	60000

4)

Lab No. 2



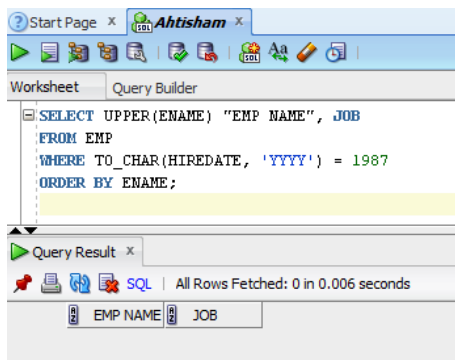
The screenshot shows the SQL Developer interface. The Query Builder tab is active, displaying the following SQL query:

```
SELECT ENAME, DEPTNO, SAL
FROM EMP
ORDER BY DEPTNO, SAL DESC;
```

The Query window below shows the results of the query, with 14 rows fetched in 0.002 seconds. The results are as follows:

	ENAME	DEPTNO	SAL
1	KING	10	5000
2	CLARK	10	2450
3	MILLER	10	1300
4	SCOTT	20	3000
5	FORD	20	3000
6	JONES	20	2975
7	ADAMS	20	1100
8	SMITH	20	800
9	BLAKE	30	2850
10	ALLEN	30	1600
11	TURNER	30	1500
12	MARTIN	30	1250
13	WARD	30	1250
14	JAMES	30	950

5)

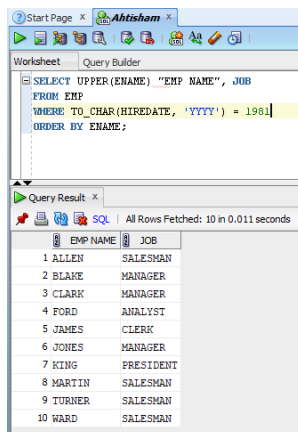


The screenshot shows the SQL Developer interface. The Query Builder tab is active, displaying the following SQL query:

```
SELECT UPPER(ENAME) "EMP NAME", JOB
FROM EMP
WHERE TO_CHAR(HIREDATE, 'YYYY') = 1987
ORDER BY ENAME;
```

The Query window below shows the results of the query, with 0 rows fetched in 0.006 seconds. The results are as follows:

	EMP NAME	JOB
--	----------	-----



The screenshot shows the SQL Developer interface. The Query Builder tab is active, displaying the following SQL query:

```
SELECT UPPER(ENAME) "EMP NAME", JOB
FROM EMP
WHERE TO_CHAR(HIREDATE, 'YYYY') = 1981
ORDER BY ENAME;
```

The Query window below shows the results of the query, with 10 rows fetched in 0.011 seconds. The results are as follows:

	EMP NAME	JOB
1	ALLEN	SALESMAN
2	BLAKE	MANAGER
3	CLARK	MANAGER
4	FORD	ANALYST
5	JAMES	CLERK
6	JONES	MANAGER
7	KING	PRESIDENT
8	MARTIN	SALESMAN
9	TURNER	SALESMAN
10	WARD	SALESMAN

6)

Lab No. 2

The screenshot shows a database query tool interface. At the top, there are tabs for 'Start Page' and 'Ahtisham'. Below the tabs is a toolbar with various icons. The main window is divided into two sections: 'Worksheet' and 'Query Builder'. The 'Query Builder' section contains a SQL query:

```
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;
```

Below the query, the 'Query Result' section shows the results of the query. It indicates that all rows were fetched in 0.002 seconds. The results are displayed in a table with the following columns: EMPNO, ENAME, JOB, and ANNUAL_SALARY.

EMPNO	ENAME	JOB	ANNUAL_SALARY
7566	JONES	MANAGER	35700
7698	BLAKE	MANAGER	34200
7782	CLARK	MANAGER	29400