Part A
Projection, ORDER BY clause, Altering and dropping of tables (use constraints while creating tables) examples using SELECT command.

Aim: SQL commands:
i) To sort data in a table
ii) To create a table from an existing table
iii) To insert data into a table from another table
i) To delete data from a table
ii) To update the contents of a table
iii) To modify the structure of a table (Alter table for adding/deleting columns and modifying columns)
iv) To rename a table
v) To truncate a table
vi) To drop a table

Prerequisite: Oracle.

Outcome: Table is created and records are inserted, viewed, and structure is modified.

Theory:

Sorting data in a table

Select * from <TableName> order by <ColumnName1>, <ColumnName2> <SortOrder>;

Creating a table from a table

Create table <TableName> (<ColumnName1>, <ColumnName2>) as select <ColumnName1>, <ColumnName2> from <TableName>;

Inserting data into a table from another table

Insert into <TableName> select <ColumnName1>, <ColumnName2> from <TableName>;

Delete data from a table

Delete from <TableName>;

Updating the contents of a table

Update <TableName> set <ColumnName1>=<expression1>,<ColumnName2>=<expression2>;

Modifying the structure of tables (SQL ALTER TABLE Syntax)

alter table <TableName> add (<ColumnName1><datatype1>, <ColumnName2><datatype2>);

Adding columns
```
ALTER TABLE table_name
ADD column_name datatype;
```

```
Deleting columns
ALTER TABLE table_name
DROP COLUMN column_name;

Modifying columns (Prior Oracle 10G)
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;

ALTER TABLE table_name (Oracle 10G and later:)
MODIFY column_name datatype;
```

Renaming tables

Rename <TableName> to <NewTableName>;

Truncating Tables

Truncate table <TableName>;

Destroying Tables

drop table <TableName>;

Procedure:
1. Formulate the query for given problem.
2. Write the SQL query with proper input.
3. Execute the query.

Practice Exercise

1	List the emps in the asc order of their Salaries?
2	List the details of the emps in asc order of the Dptnos and desc of Jobs?
3	Display all the unique job groups in the descending order?
4	Display all the details of all 'Mgrs'
5	Display all the details of the emps whose Comm. Is more than their Sal.
6	List the emps who are either 'CLERK' or 'ANALYST' in the Desc order.
7	List the emps Who Annual sal ranging from 22000 and 45000.
8	Create a table Emp_Income(E_ID,Name, Salary, Commission) from emp.
9	Create a table Emp_data(Name, Gender, Post, Dept_in) from emp.
10	Create a table Section(S_ID, S_name). (create by normal create table query)
11	Copy required data into Section table from dapt table.
12	Create table Relation(Ename, Manager, Dept_no). (create by normal create table query)
13	Add age at attribute to the employee table
14	Remove customer name attribute from the depositor table
15	Increase the size of column ename in table employee to 15.

16	Add New column' Data_of_Birth' in the Emp_data table. Which is of type Date.
17	Change the name of the Section table to Office, and Relation to Association.
18	Update the commission of all clerks to 500 in EMP.
19	Change the salary of every manager to 50000 in Emp_income.
20	Increase the salary of every salesman by 500 in Emp_income.
21	Delete the information of all employees of dept_no 20 in Emp_data.
22	Delete all the data from Association and Office.
23	Delete the information of 'martin' from Emp_data table.
24	Increase the commission of all the females by 10%.
25	Completely delete all the tables created today.

Instructions:
1. Write and execute the query in Oracle SQL server.
2. Paste the snapshot of the output in input & output section.

Part B

Code and Output:

1)Select Ename from Emp1 order by Salary;

☑ Autocommit   Rows   30 ⌄ 🖊 🖊   (Save) (Run)

select Ename from Emp1 order by Salary;

Results Explain Describe Saved SQL History

ENAME
Smith
Adams
Ward
Martin
Miller
Turner
Allen
Clark
Blake
Jones
Ford
Scott
King
James

14 rows returned in 0.01 seconds   Download

Workspace: GAYATHRI User: GAYATHRI

----

2)select * from Emp1 order by Dept_no asc,Job desc;

Results Explain Describe Saved SQL History

EMP_NO	ENAME	GENDER	JOB	MGR	HIREDATE	SALARY	COMM	DEPT_NO
7839	King	M	PRESIDENT	-	11/17/1981	50000	-	10
7782	Clark	M	MANAGER	7839	06/09/1981	24500	-	10
7934	Miller	F	CLERK	7782	01/23/1982	13000	-	10
7566	Jones	F	MANAGER	7839	04/02/1981	29750	-	20
7876	Adams	M	CLERK	7788	01/12/1983	11000	-	20
7369	Smith	M	CLERK	7902	12/17/1980	8000	-	20
7788	Scott	M	ANALYST	7566	12/09/1982	30000	-	20
7902	Ford	M	ANALYST	7566	12/03/1981	30000	-	20
7654	Martin	M	SALESMAN	7698	09/28/1981	12500	14000	30
7844	Turner	M	SALESMAN	7698	09/08/1981	15000	-	30
7521	Ward	M	SALESMAN	7698	02/22/1981	12500	5000	30
7499	Allen	F	SALESMAN	7698	02/20/1981	16000	3000	30
7698	Blake	M	MANAGER	7839	05/01/1981	28500	-	30
7900	James	M	CLERK	7698	12/03/1981	95000	-	30

14 rows returned in 0.01 seconds   Download

Workspace: GAYATHRI User: GAYATHRI

----

3)select distinct Job from Emp1 order by Job desc;

Results   Explain   Describe   Saved SQL   History

JOB
SALESMAN
PRESIDENT
MANAGER
CLERK
ANALYST

5 rows returned in 0.01 seconds   Download

Workspace: GAYATHRI User: GAYATHRI

----

4) select * from Emp1 where Mgr is not null;

Results   Explain   Describe   Saved SQL   History

EMP_NO	ENAME	GENDER	JOB	MGR	HIREDATE	SALARY	COMM	DEPT_NO
7369	Smith	M	CLERK	7902	12/17/1980	8000	-	20
7499	Allen	F	SALESMAN	7698	02/20/1981	16000	3000	30
7521	Ward	M	SALESMAN	7698	02/22/1981	12500	5000	30
7566	Jones	F	MANAGER	7839	04/02/1981	29750	-	20
7654	Martin	M	SALESMAN	7698	09/28/1981	12500	14000	30
7698	Blake	M	MANAGER	7839	05/01/1981	28500	-	30
7782	Clark	M	MANAGER	7839	06/09/1981	24500	-	10
7788	Scott	M	ANALYST	7566	12/09/1982	30000	-	20
7844	Turner	M	SALESMAN	7698	09/08/1981	15000	-	30
7876	Adams	M	CLERK	7788	01/12/1983	11000	-	20
7900	James	M	CLERK	7698	12/03/1981	95000	-	30
7902	Ford	M	ANALYST	7566	12/03/1981	30000	-	20
7934	Miller	F	CLERK	7782	01/23/1982	13000	-	10

13 rows returned in 0.01 seconds   Download

Workspace: GAYATHRI User: GAYATHRI

----

5)select * from Emp1 where comm>salary;

☑ Autocommit   Rows   30 ▼ ✏ ✒   Save   Run

select * from Emp1 where comm>salary;

Results Explain Describe Saved SQL History

EMP_NO	ENAME	GENDER	JOB	MGR	HIREDATE	SALARY	COMM	DEPT_NO
7654	Martin	M	SALESMAN	7698	09/28/1981	12500	14000	30

1 rows returned in 0.01 seconds   Download

----

6)select ename from emp1 where job='CLERK' or job='ANALYST' order by ename desc;

☑ Autocommit   Rows   30 ▼ ✏ ✒   Save   Run

select ename from emp1 where job='CLERK' or job='ANALYST' order by ename desc;

Results Explain Describe Saved SQL History

ENAME
Smith
Scott
Miller
James
Ford
Adams

6 rows returned in 0.00 seconds   Download

Workspace: GAYATHRI User: GAYATHRI

----

7)select ename from emp1 where salary>22000 and salary<45000;

Results Explain Describe Saved SQL History

ENAME
Jones
Blake
Clark
Scott
Ford

5 rows returned in 0.01 seconds   Download

Workspace: GAYATHRI User: GAYATHRI

----

8)create table emp_income as select emp_no as e_id,ename as name,salary as salary,comm as commission from emp;

☑ Autocommit   Rows   [30 ▼] ✐ ✐   (Save) (Run)

```
create table emp_income as select emp_no as e_id,ename as name,salary as salary,comm as commission from emp1;
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

----

9)
create table emp_data as select ename as name , gender as gender,job as post,Dept_no as dept_in from emp1;

☑ Autocommit   Rows   [30 ▼] ✐ ✐   (Save) (Run)

```
create table emp_data as select ename as name , gender as gender,job as post,Dept_no as dept_in from emp1;
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

----

10)
create table section(s_id int,sname varchar(20));

☑ Autocommit   Rows   30 ▾ 🧽 ✏   (Save) (Run)

```
create table section(s_id int,sname varchar(20));
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

----

11)insert into section dept_no,dname from department;

☑ Autocommit   Rows   30 ▾ 🧽 ✏   (Save) (Run)

```
insert into section select dept_no,dname from department;
```

Results Explain Describe Saved SQL History

4 row(s) inserted.

0.00 seconds

----

12) create table relation(ename varchar(20),manager varchar(20),dept_no int);

☑ Autocommit   Rows   30 ▾ 🧽 ✏   (Save) (Run)

```
create table relation(ename varchar(20),manager varchar(20),dept_no int);
```

Results   Explain   Describe   Saved SQL   Histor

Table created.

0.01 seconds

----

13)alter table emp1 add(age int);

☑ Autocommit   Rows   `30` ▽   ✎   ✐   (Save) (Run)

```
alter table emp1 add(age int);
```

Results   Explain   Describe   Saved SQL   History

Table altered.

0.04 seconds
----

14)alter table depositor drop column custname;

☑ Autocommit   Rows   `30` ▽   ✎   ✐   (Save) (Run)

```
alter table depositor drop column custname;
```

Results   Explain   Describe   Saved SQL   History

Table altered.

0.33 seconds

----

15)
alter table emp1 modify ename varchar(15);

☑ Autocommit   Rows   `30` ▽   ✎   ✐   (Save) (Run)

```
alter table emp1 modify ename varchar(15);
```

Results Explain Describe Saved SQL History

Table altered.

0.03 seconds

----

16)alter table emp1 add(date_of_birth date);

☑ Autocommit   Rows   [30   ⌄] 🖊 ✎   (Save) (Run)

```
alter table emp1 add(date_of_birth date);
```

Results Explain Describe Saved SQL History

Table altered.

0.00 seconds

----

17)
alter table section rename to Office;
alter table relation rename to association;

☑ Autocommit   Rows   [30   ⌄] 🖊 ✎   (Save) (Run)

```
alter table section rename to Office;
alter table relation rename to association;
```

Results Explain Describe Saved SQL History

Table altered.

0.01 seconds

----

18)
update emp set comm=500 where job='CLERK';

✅ Autocommit   Rows   [30 ⌄] 🖊 ✏️   (Save) (Run)

```
update emp set comm=500 where job='CLERK';
```

Results Explain Describe Saved SQL History

4 row(s) updated.

0.01 seconds

----

19) update emp_income set salary=50000 where e_id in (select emp_no from emp where job='manager');

✅ Autocommit   Rows   [30 ⌄] 🖊 ✏️   (Save) (Run)

```
update emp_income set salary=50000 where e_id in (select emp_no from emp1 where job='MANAGER');
```

Results Explain Describe Saved SQL History

3 row(s) updated.

0.02 seconds

----

20)update emp_income set salary=salary+500 where e_id in(select emp_no from emp1 where job='SALESMAN');

✅ Autocommit   Rows   [30 ⌄] 🖊 ✏️   (Save) (Run)

```
update emp_income set salary=salary+500 where e_id in(select emp_no from emp1 where job='SALESMAN');
```

Results Explain Describe Saved SQL History

4 row(s) updated.

0.01 seconds

----

21)delete from emp_data where dept_in=20;

```
Autocommit    Rows    30 ∨    Save    Run
delete from emp_data where dept_in=20;

Results  Explain  Describe  Saved SQL  History

5 row(s) deleted.

0.00 seconds
```

----

22)
delete from association;

```
Autocommit    Rows    30 ∨    Save    Run
delete from association;

Results  Explain  Describe  Saved SQL  History

0 row(s) deleted.

0.01 seconds
```

delete from office;

```
Autocommit    Rows    30 ∨    Save    Run
delete from office;

Results  Explain  Describe  Saved SQL  History

4 row(s) deleted.

0.00 seconds
```

----

23)delete from emp_data where name='Martin';

```
✓ Autocommit    Rows   [30  ∨]  🖊 ✏        ( Save )  ( Run )
delete from emp_data where name='Martin';
```

Results Explain Describe Saved SQL History

1 row(s) deleted.

0.00 seconds

----

24)update emp1 set comm=comm+0.1*comm where gender='F';

```
✓ Autocommit    Rows   [30  ∨]  🖊 ✏        ( Save )  ( Run )
update emp1 set comm=comm+0.1*comm where gender='F';
```

Results Explain Describe Saved SQL History

3 row(s) updated.

0.00 seconds

----

25)
drop table office;
drop table association;
drop table emp_data;
drop table emp_income;

```
✓ Autocommit    Rows   [30  ∨]  🖊 ✏        ( Save )  ( Run )
drop table office;
drop table association;
drop table emp_data;
drop table emp_income;
```

Results Explain Describe Saved SQL History

Table dropped.

0.02 seconds

Observation & Learning:
Learned and executed following SQL commands on database
sorted data in a table
created a table from an existing table
inserted data into a table from another table
deleted data from a table
updated the contents of a table
modified the structure of a table (Alter table for adding/deleting columns and modifying columns)
renamed a table
truncated a table
dropped a table

Conclusion:
Learned and practiced DML commands and recorded the outputs perfectly.

Questions:
1. What happen if where clause is not given in query?
2. What are the various comparison operator used in condition part?
3. Give the difference between delete, truncate, and destroy command.
4. What happen if domain type of data inserted is different from that of column?

5. What are the various comparison operator used in condition part?
6. What is the meaning of NULL in DBMS?

ANSWERS:

1. **Unnecessary tuples will also get selected.**
   And our work will not be efficient.
2. **= (equal to)**
   < > (not equal to)
   > (greater than)
   < (less than)
   >= (greater than or equal to)
   <= (less than or equal to)
3. **Difference between Truncate ,delete and destroy is**
   TRUNCATE SQL query removes all rows from a table, without logging the individual row deletions. TRUNCATE is faster than the DELETE query.
   DELETE query deletes all records from a database table. To execute a DELETE query, delete permissions are required on the target table. If you need to use a WHERE clause in a DELETE, select permissions are required as well.
   DROP table query removes one or more table definitions and all data, indexes, triggers, constraints, and permission specifications for those tables. DROP command requires to ALTER permission on the schema to which the table belongs, CONTROL permission on the table, or membership in the db_ddladmin fixed database role.

4. Data cant be inserted as the datatype of column does not match with given input data and error message will be displayed.
5. = (equal to)
   < > (not equal to)
   > (greater than)
   < (less than)
   >= (greater than or equal to)
   <= (less than or equal to)
6. Null means having no value; in other words null is zero.