Ex.No.: 1		**CREATION OF BASE TABLE AND DML OPERATIONS**
Date:		

AIM:

ALGORITHM:

STEP-1: Start.

STEP-2: Create a base Table

Syntax:

```
CREATE TABLE <table name> (column1 type, column2 type, …);
```

STEP-3: Describe the Table structure

Syntax:

```
DESC  <table name>
```

STEP-4: Add a new row to a Table using INSERT statement.

Syntax:

- INSERT INTO <table name> VALUES (value1, value2..);
- INSERT INTO <table name> (column1, column2..) VALUES (value1, value2..);
- INSERT INTO <table name>VALUES (&column1,’&column’);

STEP-5: Modify the existing rows in the base Table with UPDATE statement.

Syntax:

```
UPDATE <table name> SET column1=value, column2 = ’value’
WHERE (condition);
```

STEP-6: Remove the existing rows from the Table using DELETE statement.

Syntax:

```
DELETE FROM <table name> WHERE <condition>;
```

STEP-7: Perform a Query using SELECT statement.

Syntax:

```
SELECT  [DISTINCT] {*,<column1,,..>} FROM <table name>
WHERE <condition>;
```

STEP-8: The truncate command deletes all rows from the table. Only the structure of the table remains.

Syntax:

```
TRUNCATE TABLE <table name>;
```

STEP-9: Alter the existing table using ALTER statement.

Syntax:

Add Column:

```
ALTER TABLE <table name> ADD (column data type
[DEFAULTexpr][,column data type]);
```

Modify Column:

```
ALTER TABLE <table name> MODIFY (column data type
[DEFAULT expr], [,column data type]);
```

Drop Column:

```
ALTER  TABLE <table name> DROP COLUMN <column name>;
```

STEP-10: To drop the entire table using DROP statement.

Syntax:

```
DROP TABLE <table name>;
```

STEP-11: Exit.

1. Create MY_EMPLOYEE table with the following structure

```
CReate table my-Employee ( id. number (4) not null,
                           last_name  varchar(25),
                           first_name  varchar (25),
                           user_id  varchar (25),
                           salary  number (9,2));
```

3.

ID	LAST NAME	FIRSL NAME	USER_ID	SALARY
1	patel	Ralph	rpatel	895
2.	Dancs	Betty	bdancs	860

4.

ID	LAST_NAME	FIRST_NAME	USER_ID	SALARY
1.	Patel	Ralph	rpatel	895
2.	Dancs	Betty	bdancs	860
3.	Biri	Ben	bbiri	1100
4.	Newman	chad	cnewman	750

5.

ID	Last_name	First_name	userid	Salary
1.	Patel	ralph	rpatel	895
3.	Biri	Ben	bbiri	1100
4.	Newman	Chad	chewman	750-

NAME	NULL?	TYPE
ID	Not null	Number(4)
Last_name		Varchar(25)
First_name		Varchar(25)
Userid		Varchar(25)
Salary		Number(9,2)

2. Add the first and second rows data to MY_EMPLOYEE table from the following sample data.

ID	Last_name	First_name	Userid	salary
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	Cnewman	750
5	Ropebur	Aud-ey	aropebur	1550

insert into my-employee values (1, 'Patel', 'Ralph', 'rpatel', 895);
insert into my-employee values (2, 'Dancs', 'Betty', 'bdancs', 860);

3. Display the table with values.

select * from my-employee ;

4. Populate the next two rows of data from the sample data. Concatenate the first letter of the first_name with the first seven characters of the last_name to produce Userid.

insert into my-employee values (3, 'Biri', 'Ben', 'bbiri', 1100);
insert into my-employee values (4, 'Newman', 'Chad', 'Cnewman', 7

select * from my-employee;

5. Delete Betty dancs from MY _EMPLOYEE table.

delete from my-employee where ID=2

6)

ID	Last_name	First name	userid	salary
1.	Patel	ralph	rpatel	895
3.	Biri	ben	bbiri	1100

8)

ID	Last_name	First_name	userid	Salary
1.	Patel	ralph	rpatel	895
3.	Drexler	ben	bbiri	1100

9.

ID	last_name	First_name	User_id	salary.
1.	Patel	ralph	rpatel	1000
2.	Drexler	ben	bbiri	1100.

6. Empty the fourth row of the emp table.

```
delete from my-employee where ID =4;
```

7. Make the data additions pe manent.

```
commit;
```

8. Change the last name of employee 3 to Drexler.

```
update my-employee set last_name =" Drexler"
where ID = 3;
```

9. Change the salary to 1000 for all the employees with a salary less than 900.

```
update my-employee set salary = 1000 where salary <900;
```

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	

TABLE CREATION :-

```
creat table employees (employee-id number (6) not null,
    first-name varchar2(6) not null ;
    last name varchar 2 (25) not nulle ;
    email varChar 2 (25) not null ;
    phone number varchar 2 (20) ;
    hoie_ date date not null ;
    joh_ id varchar 2 (10) not null ;
    salary number (8,2) ;
    commis sion_pct number (2,2) ;
    manger-id number (6) ;
    department -id number (4) ;
    primary key (emplayee-id)) ;
```

a)

employeeid	First_name	Last_name	salary
1002	jane	smith	4200
1003	alice	Johnson	7500
1001	John	robert	5000
1004	robert	alice	4800
1005	emily	emily	5300

b)

employee-id	First_name	Last_name
1001	John	doe
1004	robert	brown

)

First_name	Last_name
alice	doe
john	brown
robert	davis
emily	Johnson

Ex.No.: 2		DATA MANIPULATIONS
Date:		

Create the following tables with the given structure.

EMPLOYEES TABLE

NAME	NULL?	TYPE
Employee_id	Not null	Number(6)
First_Name		Varchar(20)
Last_Name	Not null	Varchar(25)
Email	Not null	Varchar(25)
Phone_Number		Varchar(20)
Hire_date	Not null	Date
Job_id	Not null	Varchar(10)
Salary		Number(8,2)
Commission_pct		Number(2,2)
Manager_id		Number(6)
Department_id		Number(4)

(a) Find out the employee id, names, salaries of all the employees

```
select employee_id , first_name, last_name ,
 salary from employees;
```

(b) List out the employees who works under manager 100

```
select employee_id , first_name , last_name .
  from employees where manager_id = 100;
```

(c) Find the names of the employees who have a salary greater than or equal to 4800

select first_name , last_name ~~where~~ from
employees where salary >= 4800;

d)
select fir

d)

Employee-id	First_name	Last_name	email
1004	robert	brown	robertbrown @example .com

e)

First_name	Last_name
john	doe
Jane	smith
alice	Johnson
robert	brown
emily	davis.

f)

unique_manager_id
104
100
103
102

DEPARTMENT TABLE :-

QUERY:

```
create table dept ( dept_id number (6) not null,
                    dept_name varchar (20) not null,
           manager_id number (6);
             Location_id number (4);
```

```
 desc dept;                          (1, 'AIML', 82, 101);
insert into dept values              (2, 'AIDS', 83, 102);
insert into dept values              (3, 'CSE', 84, 103);
insert into dept values
select * from dept;
```

Dept_id	Dept_name	manager_id	Location id
			101
1	AIML	82	102
2	AIDS	83	103.
3	CSE	84	

Job grade Table:-

```
create table job (grade_level varchar (2),
                  low_sal number (10);
                  high_sal number (10));
```

Location table:-

```
create table locat ( location_id varchar (4) not null,
                     st_addr varchar (40);
                     postal_code varchar (12);
                     city varchar (30) not null;
                      state_province varchar (25);
                      country_id . char (2));
```

DEPARTMENT TABLE

NAME	NULL?	TYPE
Dept_id	Not null	Number(6)
Dept_name	Not null	Varchar(20)
Manager_id		Number(6)
Location_id		Number(4)

JOB_GRADE TABLE

NAME	NULL?	TYPE
Grade_level		Varchar(2)
Lowest_sal		Number
Highest_sal		Number

LOCATION TABLE

NAME	NULL?	TYPE
Location_id	Not null	Number(4)
St_addr		Varchar(40)
Postal_code		Varchar(12)
City	Not null	Varchar(30)
State_province		Varchar(25)
Country_id		Char(2)

1. Create the DEPT table based on the DEPARTMENT following the table instance chart below. Confirm that the table is created.

Column name	ID	NAME
Key Type		
Nulls/Unique		
FK table		
FK column		
Data Type	Number	Varchar2
Length	7	25

```
create table dept_new (id_no number (7) not null,
                        dept_name varchar 2 (25);
                        primary key (id_no));
```

```
disc dept_new;
insert into dept_new values (1 , "HR' );
insert into dept_new values (2, 'Fenance');
insert into dept_new values (3,'Engineering');
select * from dept_new;
```

Idno	Dept_name
1	HR
2	Finance
3	Engineering.

2.

2. Create the EMP table based on the following instance chart. Confirm that the table is created.

Column name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Key Type				
Nulls/Unique				
FK table				
FK column				
Data Type	Number	Varchar2	Varchar2	Number
Length	7	25	25	7

create table emp(Id-number (7) not null ,
last_name varchar -2 (25);
First_name varchar2 (25),
Dept_id number (7));

desc emp;

3 Modify the EMP table to allow for longer employee last names. Confirm the modification.(Hint: Increase the size to 50)

alter table emp modify (last_name varchar (50));

4 Create the EMPLOYEES2 table based on the structure of EMPLOYEES table. Include Only the Employee_id, First_name, Last_name, Salary and Dept_id coloumns. Name the columns Id, First_name, Last_name, salary and Dept_id respectively.

create table employees 2 as select
ID as Id ,
First_name as First_name ;
LAST_name as Last_name ;
SALARY vas Salary ;
Dept_Id as Dept_id
from emp;

5 Drop the EMP table.

Drop table emp;

6 Rename the EMPLOYEES2 table as EMP.

alter table employees 2 rename to Emp;

7 Add a comment on DEPT and EMP tables. Confirm the modification by describing the table.

comment on table Dept is 'Table containing dept details';

comment on table emp is 'Table containing emp details';

8 Drop the First_name column from the EMP table and confirm it.

alter table emp drop Column First name ;

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	R

Ex.No.: 3		WRITING BASIC SQL SELECT STATEMENTS
Date:		

OBJECTIVES

After the completion of this exercise, the students will be able to do the following:

- List the capabilities of SQL SELECT Statement
- Execute a basic SELECT statement

Capabilities of SQL SELECT statement

A SELECT statement retrieves information from the database. Using a select statement, we can perform

- ✔ Projection: To choose the columns in a table
- ✔ Selection: To choose the rows in a table
- ✔ Joining: To bring together the data that is stored in different tables

Basic SELECT Statement

Syntax

```
SELECT *|DISTINCT Column_ name| alias
        FROM table_name;
```

NOTE:

DISTINCT—Suppress the duplicates.

Alias—gives selected columns different headings.

Example: 1

```
SELECT * FROM departments;
```

Example: 2

```
SELECT location_id, department_id FROM departments;
```

Writing SQL Statements

- SQL statements are not case sensitive
- SQL statements can be on one or more lines.

Using Literal Character String

- A literal 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.

Example:

SELECT last_name||'is a'||job_id AS "EMPLOYEES JOB" FROM employees;

Eliminating Duplicate Rows

- Using DISTINCT keyword.

Example:

SELECT DISTINCT deparment_id FROM employees;

Displaying Table Structure

- Using DESC keyword.

Syntax

DESC table_name;

Example:

DESC employees;

Find the Solution for the following:

True OR False

1. The following statement executes successfully.

Identify the Errors

SELECT employee_id, last_name
sal*12 ANNUAL SALARY
FROM employees;

select employee-id , last-name ,
sal * 12 ·AS ANNUAL-SALARY from ~~SALARY~~
-employees;

Queries

2. Show the structure of departments the table. Select all the data from it.

describe departments;

select * from departments;

3. Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first.

select employe_number , last -name, job-code , hire-date
from employees;

4. Provide an alias STARTDATE for the hire date.

select employee-number, last_name, job-code , hire-date
as startdate From employees;

5. Create a query to display unique job codes from the employee table.

select ~~employee-number, last-name, job-code, hire/date~~
distinct job-code ~~AS startdate from employees~~
from employees;

6. Display the last name concatenated with the job ID , separated by a comma and space, and name the column EMPLOYEE and TITLE.

select CONCAT (last_name ,',', job-id) AS
Employee_And_Title from employees;

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

select concat_ws (',', employee-number, last_name;
job_code, hire_date) As The_output
from employees;

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	R

Ex.No.: 4		WORKING WITH CONSTRAINTS
Date:		

OBJECTIVE

After the completion of this exercise the students should be able to do the following

- Describe the constraints
- Create and maintain the constraints

What are Integrity constraints?

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies

The following types of integrity constraints are valid

a) **Domain Integrity**

✓ NOT NULL
✓ CHECK

b) **Entity Integrity**

✓ UNIQUE
✓ PRIMARY KEY

c) **Referential Integrity**

✓ FOREIGN KEY

Constraints can be created in either of two ways

1. At the same time as the table is created
2. After the table has been created.

Defining Constraints

Create table tablename (column_name1 data_ type constraints, column_name2 data_ type constraints ...);

Example:

Create table employlees (employee_id number(6), first_name varchar2(20), ..job_id varchar2 (10), CONSTRAINT emp_emp_id_pk PRIMARY KEY (employlee_id));

(OR)

```
ALTER TABLE test 1 DROP(pk, fk, col1) CASCADE CONSTRAINTS;
```

VIEWING CONSTRAINTS

Query the USER_CONSTRAINTS table to view all the constraints definition and names.

Example:

```
SELECT constraint_name, constraint_type, search_condition FROM user_constraints
WHERE table_name='employees';
```

Viewing the columns associated with constraints

```
SELECT constraint_name, constraint_type, FROM user_cons_columns
WHERE table_name='employees';
```

Find the Solution for the following:

1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column.The constraint should be named at creation. Name the constraint my_emp_id_pk.

```
Alter table emp add constraint my-emp-id-pk
        primary key (ID);
```

2. Create a PRIMAY KEY constraint to the DEPT table using the ID colum. The constraint should be named at creation. Name the constraint my_dept_id_pk.

```
Alter table dept add constraint my-dept-id-pk
        primary key (ID);
```

3. Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to nonexistent deparment. Name the constraint my_emp_dept_id_fk.

```
Alter Table emp add Dept_Id number;
alter table emp add constraint my-emp-dept_id-fk
foreign key (Dept_id)
references Dept (ID);
```

4. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

```
Alter table emp add commission number(2,2)
    check (commission >0);
```

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	