Experiment No: 8

8) Study and implementation of different types of constraints.

constraints :-

They are used to specify rules for the data in a table. It can be specified when the table is created.

1) NOT NULL :- It implies that a value must be entered into column if the record is to be accepted for storage in table.

Syntax :

```
create table table_name (column_name
data type (size) NOT Null );
```

Ex : create table employees (emp_no int(5),
emp_name varchar(10) NOT NULL);

2) unique :- To ensure that information in the column is unique i.e a value entered in column must not be repeated in a table.

Syntax :

```
create table table_name (column_name
    datatype (size) unique );
```

Ex: create table student (s_no Number (3) unique, name char (10));

3) check :- It specifies a condition that each row in a table must satisfy. Each row in table must make the condition either True or unknown.

Syntax :

```
create table table_name (column_name
    datatype (size) check (logical expression), ...);
```

Ex : create table student (sno number (3), name char (10), class char (5), check (class in ('CSE', 'CSD'));

4) Primary Key :- A column or combination of columns can be created as primary key, which can be used as a reference from other tables.

Syntax :

```
create table table_name (column_name
    datatype (size) primary key ... );
```

Ex :

```
create table faculty (fcode Number (3)
    primary key, fname char (10));
```

5) Foreign Key :- To reference any primary key column from other table this constraint can be used. The table in which the foreign key is defined is called detail table.

Syntax : create table table_name (column_name
datatype (size) foreign key (column_name)
references (table_name);

Ex : create table subject (scode Number (3) primary key, subname char (10), fcode Number (3), Foreign key (fcode) reference faculty);

6) Default :- It is used to insert a default value into a column. It will be added to all new records, if no other value is specified.

Syntax :

create table table_name (column_name1, column_name2, column_name3 default '<value>');

Ex :

create table student (sno Number (3) unique, name char (10), address varchar (20) default 'Aurangabad');

Defining integrity constraints :

Syntax : Alter table table_name ADD primary key (column_name);

Ex : Alter table student ADD primary key (sno);

Dropping integrity constraints :

Syntax : Alter table table_name drop constraint_name;

Ex : Alter table student drop primary key;

* write the queries for the following

1) create a table EMP with the following structure

   Emp_no int(5), Emp_name varchar(20), Job varchar(10), dept_no int(3), salary int(8)

   Allow Null for all columns except emp_name and Job

2) Add constraints to check, while entering the emp_no value i.e (emp_no > 100)

3) Define the field dept_no as unique

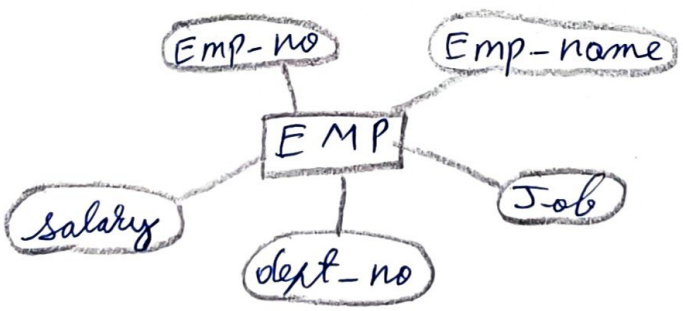
4) create a primary key constraint for Emp_no

schema diagram:

EMP

Emp_no	Emp_name	Job	dept_no	salary

ER- diagram:



1) create table EMP (emp_no int(5), emp_name varchar(20), Job varchar(10), dept_no int(3), salary int(8));

2) Alter table EMP add constraint emp_no check (emp_no > 100);

3) Alter table EMP add constraint dept_no unique (dept_no);

4) Alter table EMP add constraint emp_no primary Key (Emp_no);

insert into EMP (emp_no, emp_name, Job, dept_no, salary) values

(101, "John", "Manager", 10, 50000),

(102, "Suneet", "Analyst", 20, 60000),

(103, "James", "President", 30, 70000);

select * from EMP;

emp-no	emp-name	Job	dept-no
101	John	Manager	10
102	Sumeet	Analyst	20
103	James	Developer	30

Salary
50000
60000
70000