# index

## College Management System: Purpose & Overview

## Purpose of the System

The College Management System is designed to digitally manage various aspects of a college, ensuring smooth and efficient operations. The system aims to:

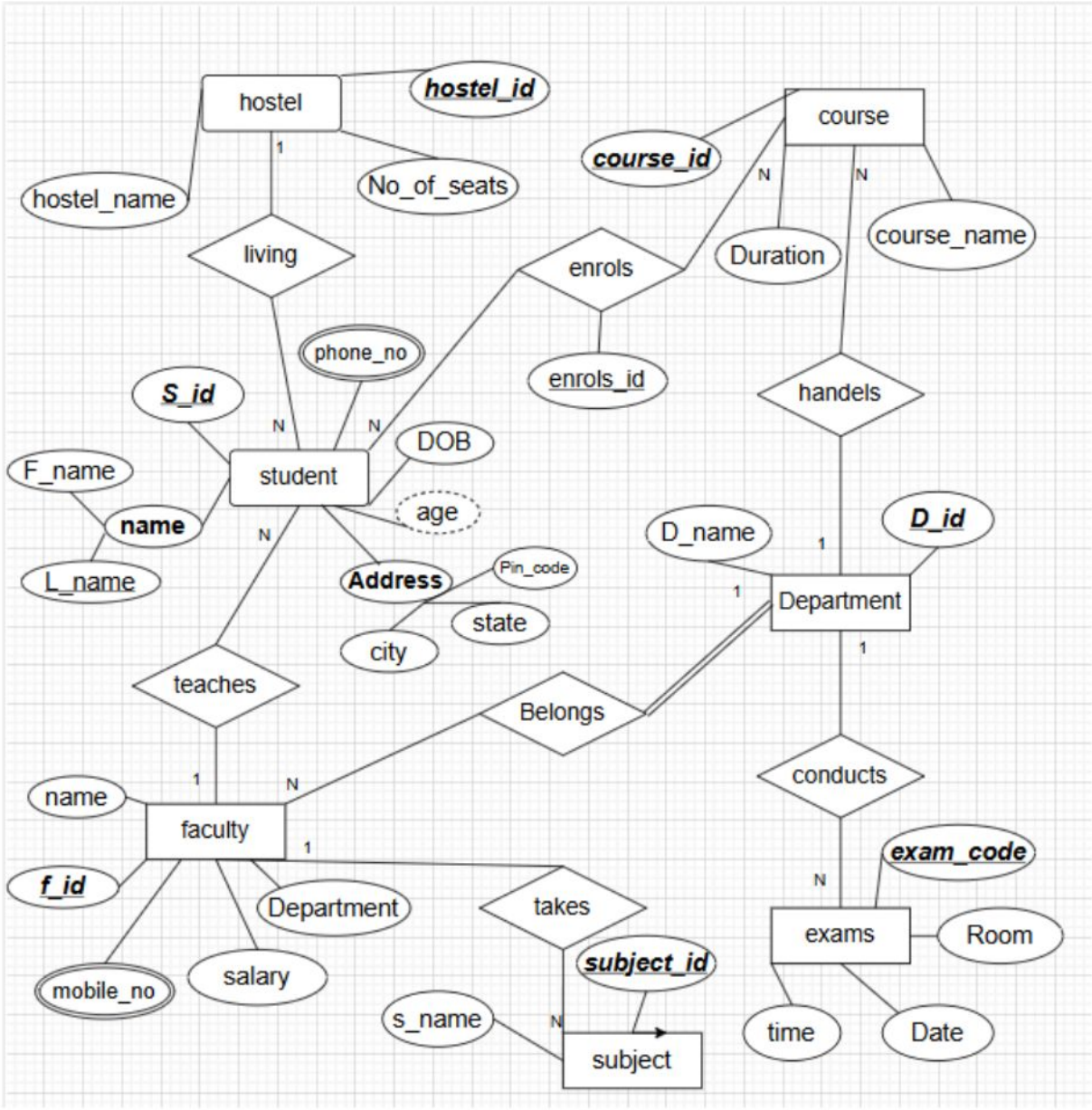
1. Automate Administrative Tasks – Reduce paperwork by managing faculty, students, courses, exams, and hostels electronically.

2.Improve Data Organization – Store and retrieve student, faculty, and department details efficiently.

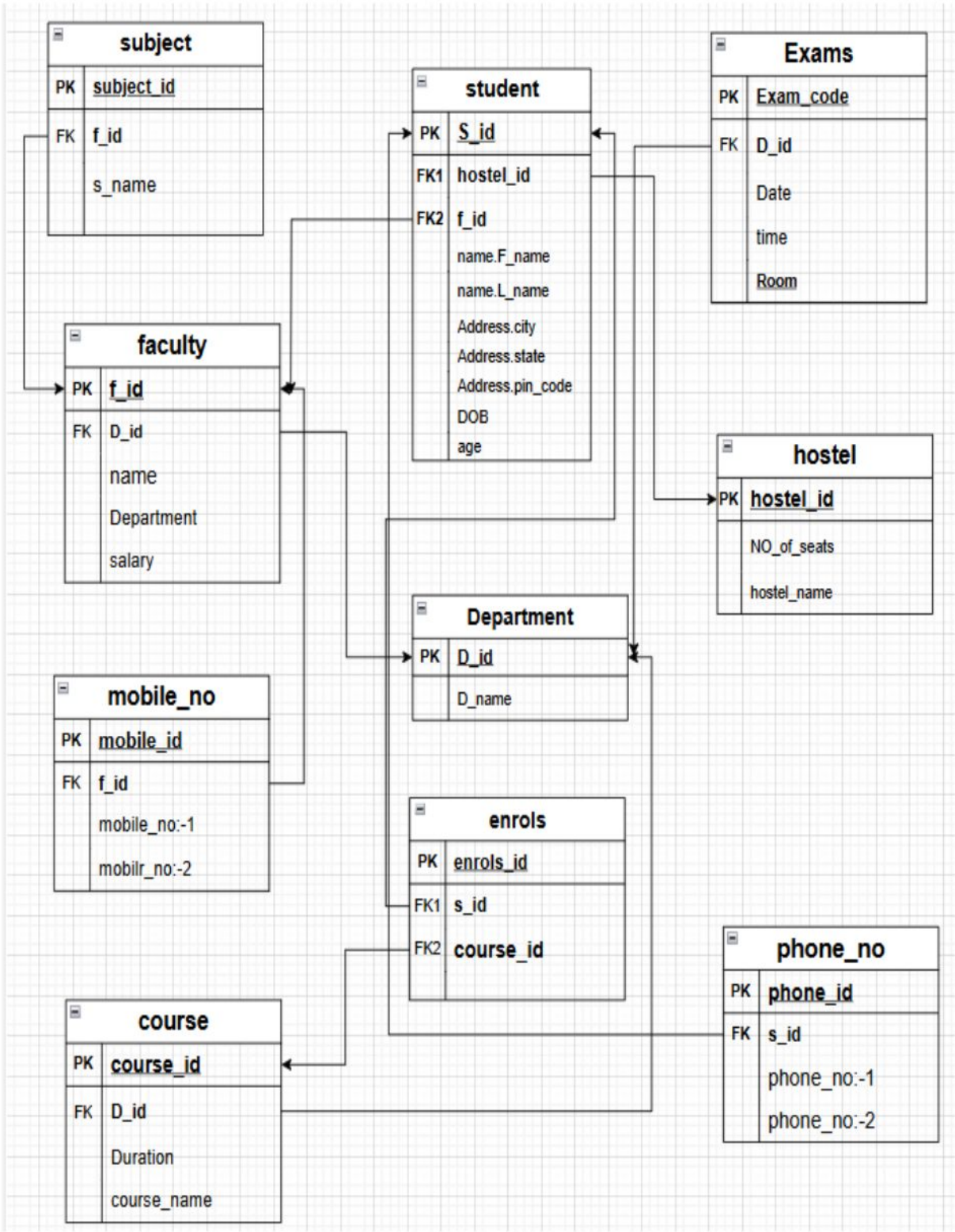
3.Enhance Communication – Provide easy access to student records, course enrollments, and exam details for administrators and faculty.

4.Optimize Student Management – Track student enrollment, hostel accommodations, and academic performance.

# 2.ENTITY-RELATIONSHIP MODEL



# 3.RELATION SCHEMA



# 4.Data Dictionary

## 4.1 Department

```
postgres=# create table Department(D_id numeric(5) primary key,D_name varcha
r(10));
CREATE TABLE
postgres=# \d Department
                    Table "public.department"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 d_id   | numeric(5,0)          |           | not null |
 d_name | character varying(10) |           |          |
Indexes:
    "department_pkey" PRIMARY KEY, btree (d_id)
```

## 4.2 hostel

```
postgres=# \d hostel
                      Table "public.hostel"
   Column    |         Type          | Collation | Nullable | Default
-------------+-----------------------+-----------+----------+---------
 hostel_id   | numeric(5,0)          |           | not null |
 no_of_seats | numeric(5,0)          |           |          |
 hostel_name | character varying(20) |           |          |
Indexes:
    "hostel_pkey" PRIMARY KEY, btree (hostel_id)
```

## 4.3 faculty

```
postgres=# \d faculty;
                    Table "public.faculty"
   Column   |         Type          | Collation | Nullable | Default
------------+-----------------------+-----------+----------+---------
 f_id       | numeric(5,0)          |           | not null |
 name       | character varying(20) |           |          |
 department | character varying(20) |           |          |
 salary     | numeric(7,0)          |           |          |
 d_id       | numeric(5,0)          |           |          |
Indexes:
    "faculty_pkey" PRIMARY KEY, btree (f_id)
Foreign-key constraints:
    "faculty_d_id_fkey" FOREIGN KEY (d_id) REFERENCES department(d_id)
Referenced by:
    TABLE "mobileno" CONSTRAINT "mobileno_f_id_fkey" FOREIGN KEY (f_id) REFERENCES faculty(f_id)
    TABLE "student" CONSTRAINT "student_f_id_fkey" FOREIGN KEY (f_id) REFERENCES faculty(f_id)
    TABLE "subjects" CONSTRAINT "subjects_f_id_fkey" FOREIGN KEY (f_id) REFERENCES faculty(f_id)
Triggers:
    min_salary_trigger BEFORE UPDATE ON faculty FOR EACH ROW EXECUTE FUNCTION enforce_min_salary()
```

## 4.4 course

```
postgres=# \d course;
                      Table "public.course"
     Column      |         Type          | Collation | Nullable | Default
-----------------+-----------------------+-----------+----------+---------
 course_id       | numeric(5,0)          |           | not null |
 duration_minute | numeric(3,0)          |           |          |
 course_name     | character varying(10) |           |          |
 d_id            | numeric(5,0)          |           |          |
Indexes:
    "course_pkey" PRIMARY KEY, btree (course_id)
Foreign-key constraints:
    "course_d_id_fkey" FOREIGN KEY (d_id) REFERENCES department(d_id)
Referenced by:
    TABLE "enrols" CONSTRAINT "enrols_course_id_fkey" FOREIGN KEY (course_id) REFERENCES course(course_id)
```

## 4.5 exams

```
postgres=# \d exams;
                   Table "public.exams"
  Column   |     Type     | Collation | Nullable | Default
-----------+--------------+-----------+----------+---------
 exam_code | numeric(5,0) |           | not null |
 room      | numeric(4,0) |           | not null |
 exam_date | date         |           |          |
 time      | numeric(5,0) |           |          |
 d_id      | numeric(5,0) |           |          |
Indexes:
    "exams_pkey" PRIMARY KEY, btree (exam_code)
Foreign-key constraints:
    "exams_d_id_fkey" FOREIGN KEY (d_id) REFERENCES department(d_id)
```

## 4.6 subject

```
postgres=# \d subjects;
                      Table "public.subjects"
    Column    |         Type          | Collation | Nullable | Default
--------------+-----------------------+-----------+----------+---------
 subject_id   | numeric(5,0)          |           | not null |
 subject_name | character varying(20) |           |          |
 f_id         | numeric(5,0)          |           |          |
Indexes:
    "subjects_pkey" PRIMARY KEY, btree (subject_id)
Foreign-key constraints:
    "subjects_f_id_fkey" FOREIGN KEY (f_id) REFERENCES faculty(f_id)
```

## 4.7 mobileno

```
postgres=# \d mobileno;
                      Table "public.mobileno"
     Column      |          Type          | Collation | Nullable | Default
-----------------+------------------------+-----------+----------+---------
 mobile_id       | numeric(5,0)           |           | not null |
 f_id            | numeric(5,0)           |           |          |
 first_mobileno  | character varying(10) |           | not null |
 second_mobileno | character varying(10) |           |          |
Indexes:
    "mobileno_pkey" PRIMARY KEY, btree (mobile_id)
Foreign-key constraints:
    "mobileno_f_id_fkey" FOREIGN KEY (f_id) REFERENCES faculty(f_id)
```

## 4.8 student

```
postgres=# \d student;
                    Table "public.student"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 s_id      | numeric(5,0)          |           | not null |
 f_name    | character varying(10) |           |          |
 l_name    | character varying(10) |           |          |
 city      | character varying(10) |           |          |
 state     | character varying(10) |           |          |
 pin_code  | character varying(10) |           |          |
 dob       | date                  |           |          |
 age       | numeric(5,0)          |           |          |
 hostel_id | numeric(5,0)          |           |          |
 f_id      | numeric(5,0)          |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (s_id)
Foreign-key constraints:
    "student_f_id_fkey" FOREIGN KEY (f_id) REFERENCES faculty(f_id)
    "student_hostel_id_fkey" FOREIGN KEY (hostel_id) REFERENCES hostel(hostel_id)
Referenced by:
    TABLE "enrols" CONSTRAINT "enrols_s_id_fkey" FOREIGN KEY (s_id) REFERENCES student(s_id)
    TABLE "phoneno" CONSTRAINT "phoneno_s_id_fkey" FOREIGN KEY (s_id) REFERENCES student(s_id)
Triggers:
    student_trg BEFORE DELETE OR UPDATE ON student FOR EACH ROW EXECUTE FUNCTION log_student_changes()
```

## 4.9 phoneno

```
postgres=# \d phoneno;
                      Table "public.phoneno"
     Column      |          Type          | Collation | Nullable | Default
-----------------+------------------------+-----------+----------+---------
 phone_id        | numeric(5,0)           |           | not null |
 s_id            | numeric(5,0)           |           |          |
 first_phone_no  | character varying(10) |           | not null |
 second_phone_no | character varying(10) |           |          |
Indexes:
    "phoneno_pkey" PRIMARY KEY, btree (phone_id)
Foreign-key constraints:
    "phoneno_s_id_fkey" FOREIGN KEY (s_id) REFERENCES student(s_id)
```

## 4.10 enrols

```
postgres=# \d enrols;
                  Table "public.enrols"
  Column   |     Type     | Collation | Nullable | Default
-----------+--------------+-----------+----------+---------
 enrols_id | numeric(5,0) |           | not null |
 s_id      | numeric(5,0) |           |          |
 course_id | numeric(5,0) |           |          |
Indexes:
    "enrols_pkey" PRIMARY KEY, btree (enrols_id)
Foreign-key constraints:
    "enrols_course_id_fkey" FOREIGN KEY (course_id) REFERENCES course(course_id)
    "enrols_s_id_fkey" FOREIGN KEY (s_id) REFERENCES student(s_id)
```

# 5.data implementation

## A) schema

### 5.1.1 department

create table Department(D_id numeric(5) primary key,D_name varchar(10));

### 5.1.2 hostel

create table hostel(hostel_id numeric(5) primary key,No_of_seats numeric(5),hostel_name varchar(20));

### 5.1.3 faculty

create table faculty(f_id numeric(5) primary key,name varchar(20),department varchar(20),salary numeric(7),D_id numeric(5),foreign key (D_id) references Department (D_id));

### 5.1.4 course

create table course(course_id numeric(5) primary key,duration numeric(3),course_name varchar(10),D_id numeric(5),foreign key (D_id) references Department (D_id));

### 5.1.5 exams

create table exams(exam_code numeric(5) primary key,room numeric(4) not null,exam_date Date,time numeric(5),D_id numeric(5),foreign key (D_id) references Department (D_id));

### 5.1.6 subjects

create table subjects(subject_id numeric(5) primary key, subject_name varchar(20),f_id numeric(5),foreign key (f_id) references faculty(f_id));

### 5.1.7 mobileno

create table mobileno(mobile_id numeric(5) primary key,f_id numeric(5),first_mobileno varchar(10) not null,second_mobileno varchar(10),foreign key (f_id) references faculty (f_id));

### 5.1.8 student

create table student(s_id numeric(5) primary key,f_name varchar(10),l_name varchar(10),city varchar(10),state varchar(10),pin_code varchar(10),DOB Date,age numeric(5),hostel_id numeric(5),f_id numeric(5),foreign key (f_id) references faculty (f_id),foreign key (hostel_id) references hostel (hostel_id));

### 5.1.9 phoneno

create table phoneno(phone_id numeric(5) primary key,s_id numeric(5),first_phone_no varchar(10) not null,second_phone_no varchar(10),foreign key (s_id) references student (s_id));

### 5.1.10 enrols

create table enrols(enrols_id numeric(5) primary key,s_id numeric(5),course_id numeric(5),foreign key (s_id) references student (s_id),foreign key (course_id) references course (course_id));

## B) data insertion

### 5.2.1 department

```
insert into Department (d_id,d_name) values (1,'IT'),(2,'CE'),(3,'EC');
```

### 5.2.2 faculty

```
insert into faculty values (11,'virat','IT',40000,1),

(12,'hardik','CE',80000,2),

(13,'rahul','EC',20000,3),

(14,'rohit','IT',50000,1),

(15,'shreyas','EC',12000,3),

(16,'bumrah','CE',90000,2);
```

### 5.2.3 hostel

```
insert into hostel values
(21,50,'h1'),(22,60,'h2'),(23,80,'h3'),(24,55,'h4');
```

### 5.2.4 mobileno

```
insert into mobileno values
(51,11,'5467894561'),(52,12,'7532145645'),(53,13,'78956123'),(54,1
4,'4561237890'),(55,15,'7568421312'),(56,16,'9979729436');
```

### 5.2.5 course

```
Insert into course values
(101,60,'DBMS',1),(102,60,'LE',3),(103,45,'DSA',2),(104,45,'DE',3),(10
5,45,'BEE',3),(106,50,'os',1),(107,50,'pps',1),(108,45,'ccn',2),

(109,45,'mapi',2);
```

### 5.2.6 subjects

insert into subjects values
(61,'DBMS',14),(62,'LE',13),(63,'DSA',12),(64,'DE',15),(65,'BEE',13),

(66,'os',11),(67,'pps',11),(68,'ccn',16),(69,'mapi',12);

### 5.2.7 exams

insert into exams values (91,2,'2025-01-04',10,1),

(92,25,'2025-02-08',10,2),(93,26,'2025-02-10',10,1),

(94,2,'2025-03-20',10,2),(95,27,'2025-03-11',10,3);

### 5.2.8 student

Insert into student values
(111,'yash','thummar','surat','gujarat','395010','2006-05-
15',19,22,11),(112,'nirmit','savaliya','rajkot','gujarat','394020','2005-
10-02',19,21,14),(113,'jenil','vaghasiya','bhopal','madhya','395012','2

005-07-10',19,22,11)
,(114,'dhrumil','lathiya','banaras','up','357684','2005-10-
02',19,22,11),(115,'shreya','thummar','jaipur','rajashthan','564564','2
000-10-05',20,21,14),
(115,'shreya','thummar','jaipur','rajashthan','564564','2000-10-
05',20,21,14), (116,'rishi','patel','surat','gujarat','564231','2002-11-
13',22,24,12), (117,'prince','patel','godhra','gujarat','852436','2004-
06-08',21,23,16),
(118,'harshil','bhungaliya','surat','gujarat','365214','2005-06-
07',20,22,12), (119,'hirav','pansuriya','rajkot','gujarat','395006','2001-
10-02',24,23,12),
(120,'manthan','parekh','rajkot','gujarat','395100','2000-10-
10',25,23,16), (121,'meet','sojitra','junagadh','gujarat','395600','2000-
10-02',24,23,13),(122,'trushil','patel','vadodra','gujarat',

'395101','2000-11-10',25,23,15),

(123,'prashant','patel','jamnagar','gujarat','321654','2003-12-25',22,24,13) ,(124,'kavish','patel','dahod','gujarat','321547','2005-10-14',20,24,15);

### 5.2.9 enrols

insert into enrols values
(161,111,106),(162,111,107),(163,112,101),(164,113,106),(165,114,107),(166,115,101),(167,116,103),(168,117,108),(169,118,103),(170,118,109),(171,119,109),(172,120,108),(173,121,102),(174,121,105),(175,122,104),(176,123,105),(177,124,104);

### 5.2.10 phoneno

Insert into phoneno values(151, 111, '9595959595', '6565666765'),(152, 112, '9510607101', '9173216352'),(153, 113, '9876543210', '8765432109'),(154, 114, '9123456789', '8234567890'),(155, 115, '9012345678', '7890123456'),(156, 116, '9321654987', '8541236790'),(157, 117, '9876123456', '8765094321'),(158, 118, '9654321876', '8321098765'),(159, 119, '9567891234', '7865432109'),(160, 120, '9456789123', '7654321098'),(161, 121, '9345678912', '7543210987'),(162, 122, '9234567891', '7432109876'),(163, 123, '9123456780', '7321098765'),(164, 124, '9012345671', '7210987654');

## Insertion output:

5.2.1 department

```
postgres=# select * from Department;
     1 | IT
     2 | CE
     3 | EC
```

### 5.2.2 faculty

```
postgres=# select * from faculty;
    11 | virat   | IT          |  40000 |    1
    12 | hardik  | CE          |  80000 |    2
    13 | rahul   | EC          |  20000 |    3
    14 | rohit   | IT          |  50000 |    1
    15 | shreyas | EC          |  12000 |    3
    16 | bumrah  | CE          |  90000 |    2
```

5.2.3 hostel

```
postgres=# select * from hostel;
         21 |           50 | h1
         22 |           60 | h2
         23 |           80 | h3
         24 |           55 | h4
```

### 5.2.4 mobileno

```
postgres=# select * from mobileno;
         51 |    11 | 5467894561     |
         52 |    12 | 7532145645     |
         53 |    13 | 78956123       |
         54 |    14 | 4561237890     |
         55 |    15 | 7568421312     |
         56 |    16 | 9979729436     |
```

### 5.2.5 course

```
postgres=# insert into course values(101,60,'DBMS',1),(102,60,'LE',3),(103,45,'DSA',2),(104,45,'DE',3),(105,45,'BEE',3),
(106,50,'os',1),(107,50,'pps',1),(108,45,'ccn',2),(109,45,'mapi',2);
INSERT 0 9
postgres=# select * from course;
 course_id | duration_minute | course_name | d_id
-----------+-----------------+-------------+------
       101 |              60 | DBMS        |    1
       102 |              60 | LE          |    3
       103 |              45 | DSA         |    2
       104 |              45 | DE          |    3
       105 |              45 | BEE         |    3
       106 |              50 | os          |    1
       107 |              50 | pps         |    1
       108 |              45 | ccn         |    2
       109 |              45 | mapi        |    2
```

5.2.6 subjects

```
postgres=# select * from subjects;
 subject_id | subject_name | f_id
------------+--------------+------
         61 | DBMS         |   14
         62 | LE           |   13
         63 | DSA          |   12
         64 | DE           |   15
         65 | BEE          |   13
         66 | os           |   11
         67 | pps          |   11
         68 | ccn          |   16
         69 | mapi         |   12
(9 rows)
```

5.2.7 exams

```
postgres=# select * from exams;
 exam_code | room | exam_date  | time | d_id
-----------+------+------------+------+------
        91 |    2 | 2025-01-04 |   10 |    1
        92 |   25 | 2025-02-08 |   10 |    2
        93 |   26 | 2025-02-10 |   10 |    1
        94 |    2 | 2025-03-20 |   10 |    2
        95 |   27 | 2025-03-11 |   10 |    3
(5 rows)
```

## 5.2.8 student

```
postgres=# select * from student;
 s_id |  f_name  |   l_name   |   city   |   state    | pin_code |    dob     | age | hostel_id | f_id
------+----------+------------+----------+------------+----------+------------+-----+-----------+------
  111 | yash     | thummar    | surat    | gujarat    | 395010   | 2006-05-15 |  19 |        22 |   11
  112 | nirmit   | savaliya   | rajkot   | gujarat    | 394020   | 2005-10-02 |  19 |        21 |   14
  113 | jenil    | vaghasiya  | surat    | gujarat    | 395012   | 2005-07-10 |  19 |        22 |   11
  114 | dhrumil  | lathiya    | banaras  | up         | 357684   | 2005-10-02 |  19 |        22 |   11
  115 | shreya   | thummar    | jaipur   | rajashthan | 564564   | 2000-10-05 |  20 |        21 |   14
  116 | rishi    | patel      | surat    | gujarat    | 564231   | 2002-11-13 |  22 |        24 |   12
  117 | prince   | patel      | godhra   | gujarat    | 852436   | 2004-06-08 |  21 |        23 |   16
  118 | harshil  | bhungaliya | surat    | gujarat    | 365214   | 2005-06-07 |  20 |        22 |   12
  119 | hirav    | pansuriya  | rajkot   | gujarat    | 395006   | 2001-10-02 |  24 |        23 |   12
  121 | meet     | sojitra    | junagadh | gujarat    | 395600   | 2000-10-02 |  24 |        23 |   13
  122 | trushil  | patel      | vadodra  | gujarat    | 395101   | 2000-11-10 |  25 |        23 |   15
  123 | prashant | patel      | jamnagar | gujarat    | 321654   | 2003-12-25 |  22 |        24 |   13
  124 | kavish   | patel      | dahod    | gujarat    | 321547   | 2005-10-14 |  20 |        24 |   15
  120 | manthan  | parekh     | rajkot   | gujarat    | 395100   | 2000-10-10 |  25 |        23 |   16
(14 rows)
```

## 5.2.9 enrols

```
postgres=# select * from enrol
 enrols_id | s_id | course_id
-----------+------+-----------
       161 |  111 |       106
       162 |  111 |       107
       163 |  112 |       101
       164 |  113 |       106
       165 |  114 |       107
       166 |  115 |       101
       167 |  116 |       103
       168 |  117 |       108
       169 |  118 |       103
       170 |  118 |       109
       171 |  119 |       109
       172 |  120 |       108
       173 |  121 |       102
       174 |  121 |       105
       175 |  122 |       104
       176 |  123 |       105
       177 |  124 |       104
(17 rows)
```

5.2.10 phoneno

```
postgres=# select * from phoneno;
 phone_id | s_id | first_phone_no | second_phone_no
----------+------+----------------+-----------------
      151 |  111 | 9595959595     | 6565666765
      152 |  112 | 9510607101     | 9173216352
      153 |  113 | 9876543210     | 8765432109
      154 |  114 | 9123456789     | 8234567890
      155 |  115 | 9012345678     | 7890123456
      156 |  116 | 9321654987     | 8541236790
      157 |  117 | 9876123456     | 8765094321
      158 |  118 | 9654321876     | 8321098765
      159 |  119 | 9567891234     | 7865432109
      160 |  120 | 9456789123     | 7654321098
      161 |  121 | 9345678912     | 7543210987
      162 |  122 | 9234567891     | 7432109876
      163 |  123 | 9123456780     | 7321098765
      164 |  124 | 9012345671     | 7210987654
(14 rows)
```

## 5.3 queries using basic DBMS constructs join & subqueries:

5.3.1:- find no.of students living in each hostel

```
postgres=# select hostel_id,count(hostel_id) from student group by hostel_id;
 hostel_id | count
-----------+-------
        21 |     2
        23 |     5
        24 |     3
        22 |     4
(4 rows)
```

5.3.2:- find the student name start with character P

```
postgres=# select * from student where f_name like 'p%';
 s_id |  f_name  | l_name |   city   |  state  | pin_code |    dob     | age | hostel_id | f_id
------+----------+--------+----------+---------+----------+------------+-----+-----------+------
  117 | prince   | patel  | godhra   | gujarat | 852436   | 2004-06-08 |  21 |        23 |   16
  123 | prashant | patel  | jamnagar | gujarat | 321654   | 2003-12-25 |  22 |        24 |   13
(2 rows)
```

5.3.3:- find faculty name whose salary between 50000 to 100000 ordered by name

```
postgres=# select name from faculty where salary>=50000 and salary<=100000 order by name asc;
  name
--------
 bumrah
 hardik
 rohit
(3 rows)
```

5.4.4:- count the no.of student whose birthday date lessthen 10

```
postgres=# select count(*) from student where extract(day from student.dob)<10;
 count
-------
     7
(1 row)
```

5.5.5:- find out course_id with more than 2 students enrolled,ordered

By s_id (DESC)

```
postgres=# select s_id from enrols group by s_id having count(s_id)>1 order by s_id DESC;
 s_id
------
  121
  118
  111
(3 rows)
```

5.5.6:- find the name of the student with their dept name who has student id  s_id=’114’

```
postgres=# select student.s_id,student.f_name,student.l_name,department.d_na
me from student join faculty on student.f_id=faculty.f_id join department on
 department.d_id=faculty.d_id where student.s_id=114;
 s_id | f_name  | l_name  | d_name
------+---------+---------+--------
  114 | dhrumil | lathiya | IT
(1 row)
```

5.5.7:- find the name of the course which exam has conducted in month of February

```
postgres=# select course_name from course where d_id in(select d_id from exams where extract(month from exams.exam_date)=02);
 course_name
-------------
 DBMS
 DSA
 os
 pps
 ccn
 mapi
(6 rows)
```

5.5.8:- find the student id and their course_id (including student not enrolled in any course)

```
postgres=# SELECT c.course_id, c.course_name, e.s_id
postgres-# FROM enrols e
postgres-# RIGHT JOIN course c ON e.course_id = c.course_id;
 course_id | course_name | s_id
-----------+-------------+------
       106 | os          |  111
       107 | pps         |  111
       101 | DBMS        |  112
       106 | os          |  113
       107 | pps         |  114
       101 | DBMS        |  115
       103 | DSA         |  116
       108 | ccn         |  117
       103 | DSA         |  118
       109 | mapi        |  118
       109 | mapi        |  119
       108 | ccn         |  120
       102 | LE          |  121
       105 | BEE         |  121
       104 | DE          |  122
       105 | BEE         |  123
       104 | DE          |  124
(17 rows)
```

5.5.9:- find student id,name who enrolled greater then 1 course

```
postgres=# select s.s_id,s.f_name,count(e.course_id) from student s join enrols e on s.s_id=e.s_id group by s.s_id,s.f_name having count(e.course_id)>1;
 s_id | f_name  | count
------+---------+-------
  118 | harshil |     2
  121 | meet    |     2
  111 | yash    |     2
(3 rows)
```

5.5.10:- Find student id and name whose department =’CE’ and hostel no_of_seats<=60

```
postgres=# select s_id,f_name from student where f_id in(select f_id from faculty where d_id=(select d_id from department where d_name='CE')) and hostel_id
in(select hostel_id from hostel where no_of_seats<=60);
 s_id | f_name
------+--------
  116 | rishi
  118 | harshil
(2 rows)
```

# 5.6 function and trigger

## 5.6.1:- create function to get salary of faculty

CREATE OR REPLACE FUNCTION get_faculty_salary_by_id(faculty_idINT)

RETURNS DECIMAL(10,2)

AS $$ DECLARE

salary_value DECIMAL(10,2);

BEGIN

SELECT salary INTO salary_value FROM faculty

WHERE f_id = faculty_id;

IF salary_value IS NULL THEN

RAISE EXCEPTION 'Faculty not found for the given ID';

END IF;

RETURN salary_value;

END;

$$ LANGUAGE plpgsql;

```
postgres=# CREATE OR REPLACE FUNCTION get_faculty_salary_by_id(
postgres(#  faculty_id INT)
postgres-#  RETURNS DECIMAL(10,2) AS $$
postgres$# DECLARE
postgres$#     salary_value DECIMAL(10,2);
postgres$#  BEGIN
postgres$#      SELECT salary INTO salary_value
postgres$#      FROM faculty
postgres$#      WHERE f_id = faculty_id;
postgres$#
postgres$#      IF salary_value IS NULL THEN
postgres$#          RAISE EXCEPTION 'Faculty not found for the given ID';
postgres$#      END IF;
postgres$#  RETURN salary_value;
postgres$#  END;
postgres$#  $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

```
postgres=# select get_faculty_salary_by_id(12);
 get_faculty_salary_by_id
--------------------------
                 80000.00
(1 row)
```

## 5.6.2:-create function get student dtails

```
CREATE OR REPLACE FUNCTION get_student_details_by_id(student_id INT)
RETURNS TABLE (s_id INT, f_name VARCHAR, dob DATE)
LANGUAGE plpgsql
AS $$
BEGIN
   RETURN QUERY
   SELECT student.s_id::INT, student.f_name, student.dob
   FROM student
   WHERE student.s_id = student_id;
END;
$$;
```

```
postgres=# CREATE OR REPLACE FUNCTION get_student_details_by_id(
postgres(#     student_id INT)
postgres-# RETURNS TABLE (s_id INT, f_name VARCHAR, dob DATE)
postgres-# LANGUAGE plpgsql
postgres-# AS $$
postgres$# BEGIN
postgres$#     RETURN QUERY
postgres$#     SELECT student.s_id::INT, student.f_name, student.dob  -- Cast s_id to INT
postgres$#     FROM student
postgres$#     WHERE student.s_id = student_id;
postgres$# END;
postgres$# $$;
CREATE FUNCTION
```

```
postgres=# select * from get_student_details_by_id(114);
 s_id | f_name  |    dob
------+---------+------------
  114 | dhrumil | 2005-10-02
(1 row)
```

## 5.6.3:- create trigger if salary < 50000 then set minimum 50000

```
CREATE OR REPLACE FUNCTION enforce_min_salary()

RETURNS TRIGGER AS $$

BEGIN

 IF NEW.salary < 50000 THEN

NEW.salary := 50000;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;


CREATE TRIGGER min_salary_trigger

BEFORE update ON faculty

FOR EACH ROW

EXECUTE FUNCTION enforce_min_salary();
```

```
postgres=# CREATE OR REPLACE FUNCTION enforce_min_salary()
postgres-# RETURNS TRIGGER AS $$
postgres$# BEGIN
postgres$#     IF NEW.salary < 50000 THEN
postgres$#         NEW.salary := 50000;
postgres$# END IF;
postgres$#     RETURN NEW;
postgres$# END;
postgres$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

```
postgres=# CREATE TRIGGER min_salary_trigger
postgres-# BEFORE update ON faculty
postgres-# FOR EACH ROW
postgres-# EXECUTE FUNCTION enforce_min_salary();
CREATE TRIGGER
```

```
postgres=# update faculty set salary=70000 where f_id=11;
UPDATE 1
postgres=# update faculty set salary=40000 where f_id=11;
UPDATE 1
postgres=# select * from faculty;
 f_id |  name   | department | salary | d_id
------+---------+------------+--------+------
   12 | hardik  | CE         |  80000 |    2
   14 | rohit   | IT         |  50000 |    1
   16 | bumrah  | CE         |  90000 |    2
   15 | shreyas | EC         |  50000 |    3
   13 | rahul   | EC         |  50000 |    3
   11 | virat   | IT         |  50000 |    1
(6 rows)
```

5.6.4:- create trigger when faculty update or delete the student details then old value automatically insert into another table with time

```
postgres=# CREATE OR REPLACE FUNCTION log_student_changes()
postgres-# RETURNS TRIGGER
postgres-# LANGUAGE plpgsql
postgres-# AS $$
postgres$# BEGIN
postgres$#     INSERT INTO student_logs (s_id, f_name, dob, city, state, hostel_id, f_id, action_type)
postgres$#     VALUES (OLD.s_id, OLD.f_name, OLD.dob, OLD.city, OLD.state, OLD.hostel_id, OLD.f_id, TG_OP);
postgres$#
postgres$#     RETURN NEW; -- Allow the update to persist
postgres$# END;
```

```
postgres=# CREATE TRIGGER student_trig
postgres-# BEFORE DELETE OR UPDATE
postgres-# ON student
postgres-# FOR EACH ROW
postgres-# EXECUTE PROCEDURE log_student_changes();
CREATE TRIGGER
```

```
postgres=# update student set city='rajkot' where s_id=121;
UPDATE 1
postgres=# select * from student_logs;
 log_id | s_id | f_name  |    dob     |   city   |  state  | hostel_id | f_id | action_type |          log_time
--------+------+---------+------------+----------+---------+-----------+------+-------------+----------------------------
      1 |  120 | manthan | 2000-10-10 | rajkot   | gujarat |        23 |   16 | UPDATE      | 2025-03-05 09:05:52.694512
      2 |  121 | meet    | 2000-10-02 | junagadh | gujarat |        23 |   13 | UPDATE      | 2025-03-05 21:46:01.354249
(2 rows)
```

## 5.7 cursor

Create cursor to select f_name and f_id if salary>=60000

```
postgres=# begin;
BEGIN
postgres=*# declare f_name_id cursor for
postgres-*# select f_id,name from faculty where salary>=60000;
DECLARE CURSOR
postgres=*# fetch next from f_name_id;
 f_id |  name
------+--------
   12 | hardik
(1 row)
```

```
postgres=# begin;
BEGIN
postgres=*# declare f_name_id cursor for
postgres-*# select f_id,name from faculty where salary>=60000;
DECLARE CURSOR
postgres=*# fetch next from f_name_id;
 f_id |  name
------+--------
   12 | hardik
(1 row)


postgres=*# fetch next from f_name_id;
 f_id |  name
------+--------
   16 | bumrah
(1 row)


postgres=*# fetch next from f_name_id;
 f_id | name
------+------
(0 rows)
```