Problem with existing system

In the present School Management System, it is uneasy to store the information related to students and faculty in physical documents. As there is too much information, when someone tries to access any of the stored information it becomes a difficult and time-consuming task. While these days authorities have more work than just taking care of students it is difficult to monitor them. Whereas storing and retrieving information is a difficult task, it also requires a lot of unnecessary work to do the task.

Proposed system

The school management system will manage all the work in any school in particular order so that the time requirement and complexity of the system will be reduced, at first it will focus on student related information. As a student gets the admission in the school system will start managing the details regarding the students. Along with student details it will also manage faculty details and staff details.

School authorities will be able to retrieve all information regarding students, teachers and other staff members using our management system.

ER Diagram

The Entity Relationship (ER) Model is a high-level conceptual data model that provides a systematic approach to analyzing data requirements to create a well-designed database.

The ER Model captures real-world entities and the relationships between them, and is considered a best practice in database design.

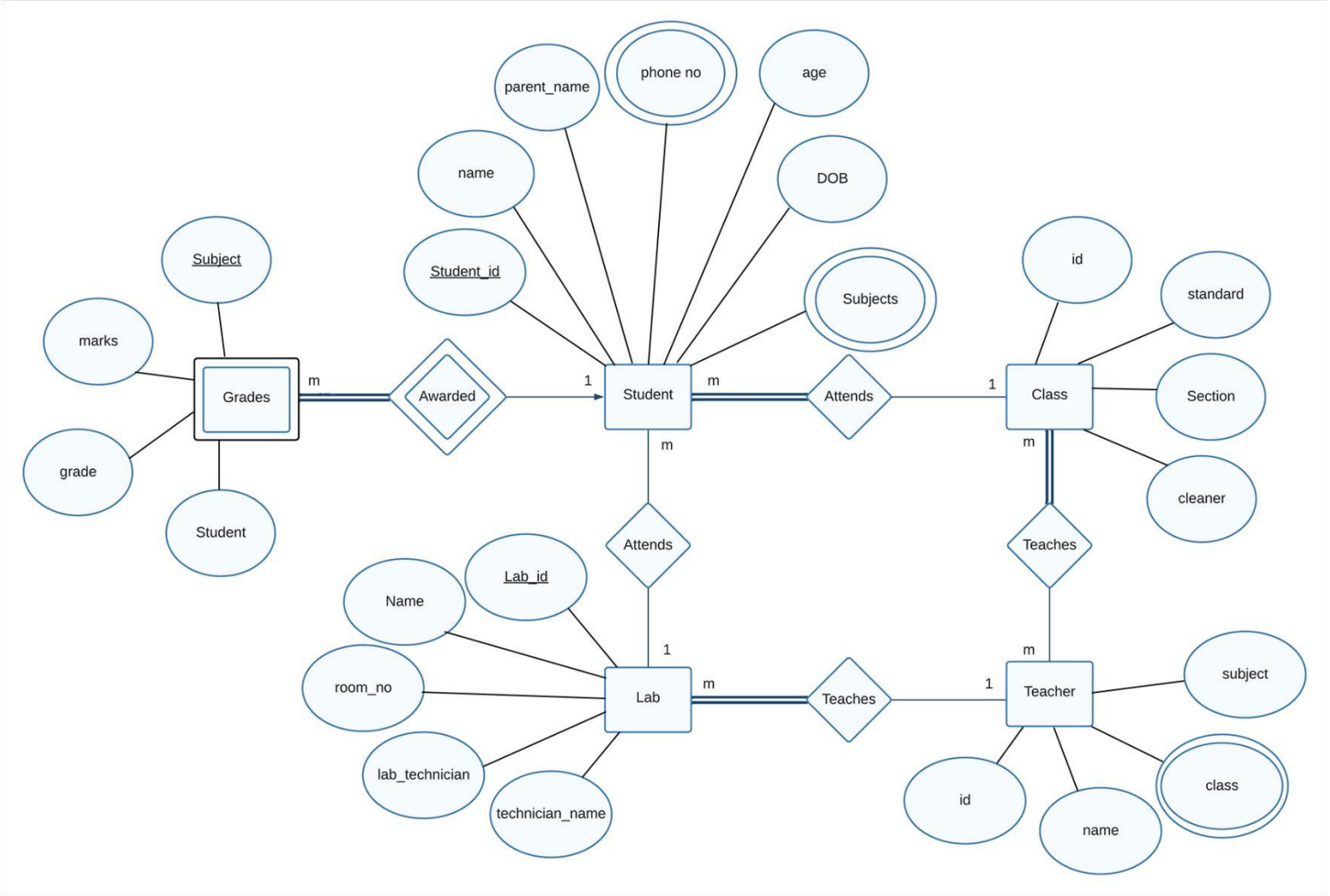
The ER Model is represented by an ER diagram, which is a visual depiction of all the entities, their attributes, and their relationships.

For our project, we have created an ER diagram that consists of five entities and a total of 30 attributes, which describe how the different processes and entities are related in our database.

By creating an ER Model before implementing a database, we can better understand the data requirements, relationships, and constraints involved in the system.

This helps us to design a database that is well structured, scalable, and easy to maintain**.**

Initial Conceptualization



Applying Normalization Up to 3NF

1. 1NF

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relationship is in the first normal form if it does not contain any composite or multi-valued attribute. A relation is in its first normal form if every attribute in that relation is a single valued attribute.

As per 1 NF

>Only atomic attributes should be there

>No attribute should hold multiple values

1.Student table

Before Normalization

sid	s_ name	P_ name	Phone No.	Strea m	age	Date Of birth	Date Of joining	class _id	Class standard	Class section	t_id	t_ name	t_dept	dept_room

Phone no and stream are multivalued functions and violate 1NF

s_id	S_ name	P_ name	age	Date Of birth	Date Of joining	Class standard	Class section	t_id	t_ name	T_dept	dept_room

s_id	Phone No.1	Phone No.2

s_id	Stream 1	Stream 2	Stream 3

2.Class Table

Before Normalization

Cleaner number, subject is multi–Valued Attribute and violate 1NF

c_id	Cleaner name	Cleaner id	cleaner number	Section	standard	Room no.	floor	Wing	Capacity	t_id	subject Name	Subject code	t_dept	Dept_ room

After Normalization

3.LAB TABLE

Class id	subject1	subject2

1-NF (equipment is multivalued)

lab_id	equipment1	equipment2	equipment3

Class id	cleaner1	Cleaner2

lab_id	lab_name	room_no	floor	wing	capacity	tech_id	tech_name	equipment

OTHER STAFF

staff_id	staff_name	work	salary	Working hours	phone_no

1-NF (Phone number is a multivalued attribute and Violates 1NF)

staff_id	staff_name	Work	salary	Working hours

staff_id	phone_no1	phone_no2

GRADES - ALREADY IN 1NF

st_id	st_name	subject_code	subject_name	marks	Grade

2NF

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

>Table should be in 1 NF

>Non-prime keys should not be partially dependent on primary key

1.Student table

Primary key (Student_id,teacher_id,class_id)

Class attributes fully functionally depend on class_id and partially on primary key **Hence Violates 2NF**

s_id	s_name	p_name	stream	age	D.O.B	Date of Joining

class_id	Class standard	Class section

t_id	t_name	t_dept

2.CLASS TABLE

c_id	Cleaner name	Cleaner id	cleaner number	section	standard	Room no.	floor	Wing	Capacity	t_id	subject Name	Subject code	t_dept	Dep

roo

Primary Key (Class_id, Cleaner_id, Teacher_id, Room_no)

Room Attributes, Cleaner Attributes and Teacher Attributes partially depend on Primary Key. **Hence Violates 2NF**

After Normalisation

room_no.	Floor	Wing	capacity

t_id	t-dept	t_room

cleaner_id	Cleaner name	Cleaner number

Subject code	Subject name

3.LAB TABLE

lab_id	lab_name	room_no	floor	wing	capacity	tech_id	tech_name	equipment

Primary key (lab_id, Technician_id, class_id)

Technician Attributes and Class attributes Partially Dependent

Hence Violates 2NF

room_no	Floor	wing	capacity

tech_id	tech_name

OTHER STAFF –

Already in 2NF as all fields fully depend in staff id

staff_id	staff_name	work	salary	Working hours	phone_no

GRADES –

Already in 2NF

st_id	st_name	subject_code	subject_name	marks	grade

3NF

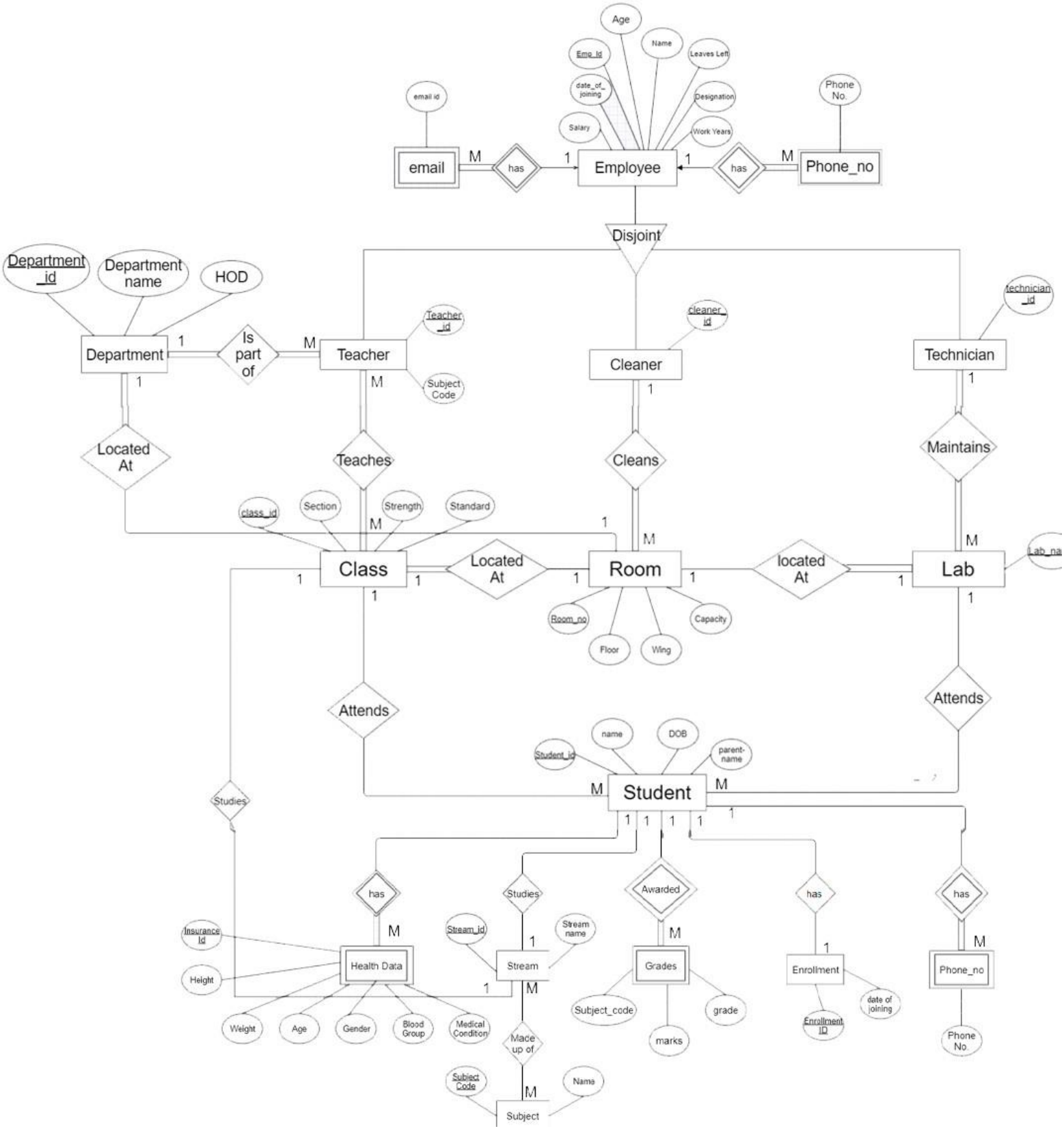
A relation will be in 3NF if it is in 2NF and does not contain any transitive partial dependency.

3NF is used to reduce data duplication. It is also used to achieve data integrity.

If there is no transitive dependency for non-prime attributes, then the relation must be in the third normal form.

ALL TABLES ARE IN 3NF

ER Diagram After Normalization

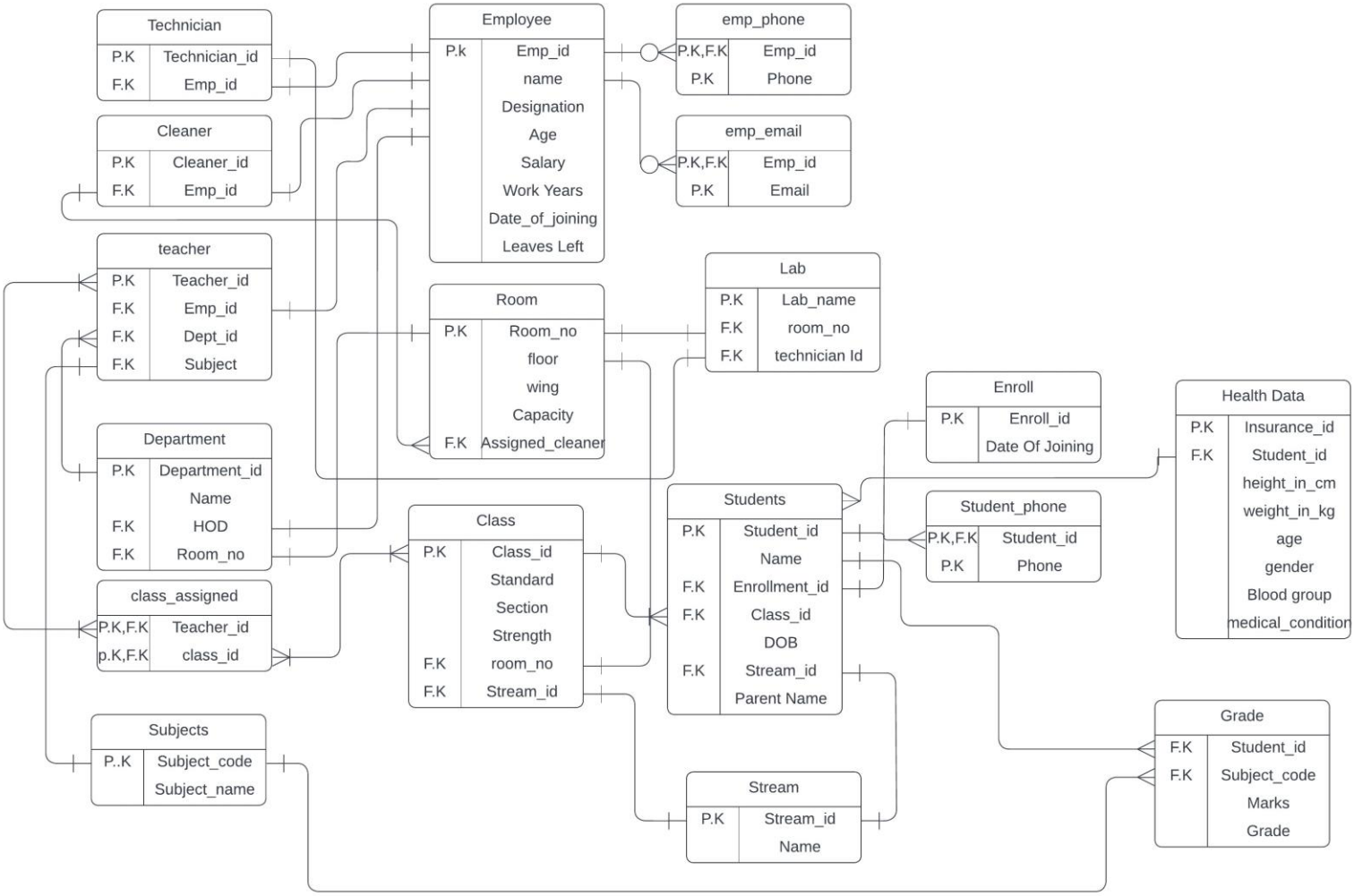


Entity-Relation Description

Entity 1	Relationship	Entity 2	Cardinalities
Employee	Has	Phone no	1:M
Employee	Has	Email	1:M
Cleaner	Cleans	Room	1:M
Teacher	Teaches	Class	M:M
Technician	Maintains	Lab	1:M
Class	Located At	Room	1:1
Lab	Located At	Room	1:1
Class	Has	Stream	1:1
Department	Located At	Room	1:1
Student	Attends	Class	M:1
Student	Attends	Lab	M:1
Student	Has	Health-data	1:1
Student	Has	Enrolment	1:1
Student	Awarded	Grades	1:M
Student	Has	Phone no	1:M
Student	Studies	Stream	M:1

Employee – A person that works at The School**.**
Cleaner – A Person that is employed for cleaning rooms.
Technician – A person Employed to maintain labs**.**
Room – A physical space for students or teachers to be in.
Class – A room where students are taught their Couse Subjects
Lab – A room where students perform practical.
Department – A room where Teachers can Sit.
Student – A person that attends the School for Gaining Knowledge

ER TO TABLE DIAGRAM



SQL and PL/SQL Commands

TABLE CREATIONS

```
1.) create table employee(emp_id number(2) primary key ,name
varchar2(50),age number(2),Designation varchar2(50),date_of_joining
date,salary number(6),work_years number(2),leave_left number(2)
Default(12));

2.)create table emp_phone(emp_id number(2)
references employee(emp_id),phone_no varchar2(10),
primary key(emp_id,phone_no));
3.)create table emp_email(emp_id number(2)
references employee(emp_id),email varchar2(30),
primary key(emp_id,email));

4.)create table cleaner(cleaner_id number(2) primary key,emp_id number(2)
references employee(emp_id));

5.)create table room(room_no varchar2(5) primary key,floor varchar2(2),wing
varchar2(2),capacity number(3),assigned_cleanerid number(2) references
cleaner(cleaner_id) ) ;


6.)create table technician(technician_id number(2) primary key,emp_id
number(2) references employee(emp_id));

7.)create table lab(lab_name varchar2(20),room_no varchar2(5) references
room(room_no),technician_id number(2) references technician(technician_id),
primary key(lab_name);

8.)create table subjects(Subject_code varchar(10) primary
key,Subject_name varchar(30));
9.)create table stream( stream_id number(2) ,stream_name varchar2(20)
,subject_code varchar2(10) references subjects(subject_code));

10.)create table class(class_id number(2) primary key,standard
varchar2(10),section_name varchar2(10),room_no varchar2(5) references
room(room_no),stream number(2) references stream(stream_id),strength
number(2));

11.)create table Department(department_id number(2) primary
key,department_name varchar2(50),HOD_id number(2) references
employee(emp_id), room_no varchar2(5) references room(room_no));


12.)create table teacher(teacher_id number(3) primary key, emp_id number(2)
references employee(emp_id),dept_id number(2) references
department(department_id),subject varchar2(10) references
subjects(subject_code) );

13.)create table enroll(enroll_id number(2) primary
key,date_of_joining date);
14.)create table students(Student_id number(2) primary key,Name
varchar2(100),enrollment_id number(2) references enroll(enroll_id),class_id
```

```
references class(class_id),DOB date,stream number(2) references
stream(stream_id),parent_name varchar2(100));

15.)create table health_data(insurance_id number(3) primary key ,student_id
number(3) references Students(student_id),height_in_cm
number(3),weight_in_kg number(3),age number(2),gender
varchar2(10),Blood_group varchar2(10),medical_conditions varchar2(100));
16.)create table Student_phone(Student_id references
students(student_id),phone varchar2(10),primary key(student_id,phone));
17.   )create table grades(student_id number(2) references
students(student_id),subject_code varchar2(10) references
subjects(subject_code),marks number(3),grade varchar2(2));
18.   ) create table class_assigned(teacher_id number(3) references
Teacher(Teacher_id),class_id number(2) references class(class_id),primary
key(teacher_id,class_id));
```

PL/SQL Procedures and Functions

1) Add teachers

```
CREATE OR REPLACE procedure add_teacher(

    emp_name IN employee.name%type,
age IN employee.age%type,
date_join In varchar2 ,         salary
IN employee.salary%type,
sub_code IN teacher.subject%type,
d_id department.department_id%type
)
IS
       id employee.emp_id%type;
id2 teacher.teacher_id%type;

BEGIN

    Select max(emp_id) + 1 into id from employee;
insert into employee
values(id,emp_name,age,'Teacher',to_date(date_join,'dd-mon-
yyyy'),salary,TRUNC(MONTHS_BETWEEN(SYSDATE, to_date(date_join,'dd-
monyyyy'))/12),12);
        SELECT max(teacher_id)+1 INTO id2 FROM teacher;
insert into teacher values(id2,id,d_id,sub_code);  END;
```

PROCEDURE ADD_TEACHER

Argument Name	Type	In/Out	Default?
EMP_NAME	VARCHAR2 (50)	IN	-
AGE	NUMBER	IN	-
DATE_JOIN	VARCHAR2	IN	-
SALARY	NUMBER	IN	-
SUB_CODE	VARCHAR2 (10)	IN	-
D_ID	NUMBER	IN	-

2) Add Cleaner

```
CREATE OR REPLACE procedure add_cleaner(

       emp_name IN employee.name%type,
age IN employee.age%type,       date_join
In varchar,
      salary IN employee.salary%type
)
IS
       id employee.emp_id%type;
id2 cleaner.cleaner_id%type;
BEGIN

    Select max(emp_id) + 1 into id from employee;
insert into employee
values(id,emp_name,age,'Cleaner',to_date(date_join,'dd-mon-
yyyy'),salary,TRUNC(MONTHS_BETWEEN(SYSDATE, to_date(date_join,'dd-
monyyyy'))/12),12);
        SELECT max(cleaner_id)+1 INTO id2 FROM
cleaner;          insert into cleaner values(id2,id);
END;
```

PROCEDURE ADD_CLEANER

Argument Name	Type	In/Out	Default?
EMP_NAME	VARCHAR2 (50)	IN	-
AGE	NUMBER	IN	-
DATE_JOIN	VARCHAR2	IN	-
SALARY	NUMBER	IN	-

3) Add HOD

```
create or replace procedure add_HOD(
emp_name IN employee.name%type,
age IN employee.age%type,       date_join
In varchar,
      salary IN employee.salary%type


)
IS
       id employee.emp_id%type;
BEGIN

    Select max(emp_id) + 1 into id from employee;
insert into employee
values(id,emp_name,age,'HOD',to_date(date_join,'dd-mon-
yyyy'),salary,TRUNC(MONTHS_BETWEEN(SYSDATE, to_date(date_join,'dd-
monyyyy'))/12),12); end;
```

PROCEDURE ADD_HOD

Argument Name	Type	In/Out	Default?
EMP_NAME	VARCHAR2 (50)	IN	-
AGE	NUMBER	IN	-
DATE_JOIN	VARCHAR2	IN	-
SALARY	NUMBER	IN	-

4) Add_technician

```
CREATE OR REPLACE procedure add_technician(

    emp_name IN employee.name%type,
age IN employee.age%type,
date_join In varchar2,
      salary IN employee.salary%type
)
IS
       id employee.emp_id%type;
      id2 technician.technician_id%type;
BEGIN

    Select max(emp_id) + 1 into id from employee;
insert into employee
values(id,emp_name,age,'Technician',to_date(date_join,'dd-
monyyyy'),salary,TRUNC(MONTHS_BETWEEN(SYSDATE, to_date(date_join,'dd-
monyyyy'))/12),12);
        SELECT max(technician_id)+1 INTO id2 FROM technician;
        insert into technician values(id2,id);
END;
```

PROCEDURE ADD_TECHNICIAN

Argument Name	Type	In/Out	Default?
EMP_NAME	VARCHAR2 (50)	IN	-
AGE	NUMBER	IN	-
DATE_JOIN	VARCHAR2	IN	-
SALARY	NUMBER	IN	-

5) Add subjects

```
CREATE OR REPLACE procedure add_subject(

    subcode
subjects.subject_code%type,
subname subjects.subject_name%type )
Is
    cntcode number(2);
cntname number(2); BEGIN
          select count(*) into cntcode from subjects where subject_code =
subcode;
        select count(*) into cntname from subjects where subject_name =
subname;
        if cntcode = 0 and cntname = 0 then
            insert into subjects values(subcode,subname);
else
            RAISE_APPLICATION_ERROR(-20001, 'Subject name or subject
code already used');          end if;

END;
```

PROCEDURE ADD_SUBJECT

Argument Name	Type	In/Out	Default?
SUBCODE	VARCHAR2 (10)	IN	-
SUBNAME	VARCHAR2 (30)	IN	-

6) Add Employee Email

```
     create or replace procedure
add_emp_mail(     emp_id IN
employee.emp_id%type,     email IN
emp_email.email%type
)
Is
begin
insert into emp_email values(emp_id,email);
end;
```

PROCEDURE ADD_EMP_MAIL

Argument Name	Type	In/Out	Default?
EMP_ID	NUMBER	IN	-
EMAIL	VARCHAR2 (30)	IN	-

7) Add room

```
CREATE OR REPLACE procedure add_room(
room_no room.room_no%type,
    floor room.floor%type,
wing room.wing%type,
capacity room.capacity%type
)     IS     cleaner_id
room.assigned_cleanerid%type; BEGIN
  SELECT cleaner_id
  INTO cleaner_id
  FROM (
    SELECT c.cleaner_id, COUNT(*) AS num_assigned
    FROM cleaner c
    LEFT JOIN room r ON c.cleaner_id = r.assigned_cleanerid
GROUP BY c.cleaner_id
    ORDER BY num_assigned
  )
  WHERE ROWNUM = 1;
  insert into room values(room_no,floor,wing,capacity,cleaner_id);
END;
```

PROCEDURE ADD_ROOM

Argument Name	Type	In/Out	Default?
ROOM_NO	VARCHAR2 (5)	IN	-
FLOOR	VARCHAR2 (2)	IN	-
WING	VARCHAR2 (2)	IN	-
CAPACITY	NUMBER	IN	-

8) Add class

```
create or replace  procedure add_class(
standard class.standard%type,       sect
class.section_name%type,            room
room.room_no%type,
    streamname stream.stream_name%type,
strength class.strength%type


) is
id class.class_id%type;
stream_id number(2);
cnt number(2); begin
select count(*) into cnt from class where class.standard = standard and
section_name = sect; if cnt = 0 then
    select max(class_id)+1 into id from class;
    select stream_id into stream_id from stream where stream_name =
streamname;
    insert into class values(id,standard,sect,room,stream_id,strength);

else
    Raise_application_error(-20001,'Class already exist'); end
if ;
End;
```

PROCEDURE ADD_CLASS

Argument Name	Type	In/Out	Default?
STANDARD	VARCHAR2 (10)	IN	-
SECT	VARCHAR2 (10)	IN	-
ROOM	VARCHAR2 (5)	IN	-
STREAMNAME	VARCHAR2 (20)	IN	-
STRENGTH	NUMBER	IN	-

9) Add department

```
create or replace  procedure add_department(
name Department.department_name%type,
    HOD employee.emp_id%type,
room room.room_no%type


) is
id department.department_id%type;
cnt number(2); begin
select count(*) into cnt from department where department_name = name ; if
cnt = 0 then
    select max(department_id)+1 into id from department;
insert into department values(id,name,HOD,room);

else
    Raise_application_error(-20001,'Department already exist');
end if ; end;
```

PROCEDURE ADD_DEPARTMENT

Argument Name	Type	In/Out	Default?
NAME	VARCHAR2 (50)	IN	-
HOD	NUMBER	IN	-
ROOM	VARCHAR2 (5)	IN	-

10) Add student

```
create or replace function add_student(
name students.name%type,
    class_id students.class_id%type,
Dob varchar2,
    stream_no students.stream%type,
parent_name students.parent_name%type,
date_join varchar2,
    height
health_data.height_in_cm%type,
weight health_data.weight_in_kg%type,
gender health_data.gender%type,
blood health_data.blood_group%type,
    medical health_data.medical_conditions%type
)return number
IS
    id students.student_id%type;
id2 students.enrollment_id%type;
id3 health_data.insurance_id%type;
Begin
    select max(enroll_id)+1 into id2 from enroll;
    insert into enroll values(id2,to_date(date_join,'dd-mon-yyyy'));
select max(student_id)+1 into id from students;
    insert into students values(id,name,id2,class_id,to_date(dob,'dd-
monyyyy'),stream_no,parent_name);
    select max(insurance_id) +1 into id3 from health_data;
insert into health_data
values(id3,id,height,weight,TRUNC(MONTHS_BETWEEN(SYSDATE,
to_date(dob,'ddmon-yyyy'))/12),gender,blood,medical);
    Update class
    Set strength = strength+1
    Where class_id = class_id;
Return id;end;
```

NAME	VARCHAR2 (100)	IN	-
CLASS_ID	NUMBER	IN	-
DOB	VARCHAR2	IN	-
STREAM_NO	NUMBER	IN	-
PARENT_NAME	VARCHAR2 (100)	IN	-
DATE_JOIN	VARCHAR2	IN	-
HEIGHT	NUMBER	IN	-
WEIGHT	NUMBER	IN	-
GENDER	VARCHAR2 (10)	IN	-
BLOOD	VARCHAR2 (10)	IN	-
MEDICAL	VARCHAR2 (100)	IN	-

11) Add Grade

```
create or replace procedure add_grade(
id students.student_id%type,
subcode subjects.subject_code%type,
mark grades.marks%type
)
IS
grade
grades.grade%type;
begin      grade := CASE
                  WHEN mark >= 90 THEN 'A'
                  WHEN mark >= 80 THEN 'B'
                  WHEN mark >= 70 THEN 'C'
                  WHEN mark >= 60 THEN 'D'
                  WHEN mark >= 50 THEN 'E'
                  ELSE 'F'
END;
insert into grades values(id,subcode,mark,grade);
END;
```

PROCEDURE ADD_GRADE

Argument Name	Type	In/Out	Default?
ID	NUMBER	IN	-
SUBCODE	VARCHAR2 (10)	IN	-
MARK	NUMBER	IN	-

12) Add student phone

```
create or replace procedure add_student_phone(
id students.student_id%type,      phone
Student_phone.phone%type
) is
begin
insert into student_phone values(id,phone); end;
```

PROCEDURE ADD_STUDENT_PHONE

Argument Name	Type	In/Out	Default?
ID	NUMBER	IN	-
PHONE	VARCHAR2 (10)	IN	-

13) Add employee phone

```
create or replace procedure
add_emp_phone(     emp_id IN
employee.emp_id%type,     phone IN
emp_phone.phone_no%type
)
is
begin
insert into emp_phone values(emp_id,phone);
end;
```

PROCEDURE ADD_EMP_PHONE

Argument Name	Type	In/Out	Default?
EMP_ID	NUMBER	IN	-
PHONE	VARCHAR2 (10)	IN	-

INSERTION CODES

```
insert into employee values(1,'Mahesh',28,'Cleaner',date '2018-02-20',15000,5,default);
 insert into employee values(2,'Mukesh',28,'Technician',date '2020-03-18',25000,3,default);
insert into employee values(3,'Gopal Sharma',35,'Teacher',date '2018-09-18',55000,5,default);
Insert into cleaner(1,1);
Insert into room(1,‘A101’,’1’,’A’,50,1);
Insert into room(2,‘A102’,’1’,’A’,50,1);
Insert into room(‘B101’,’1’,’B’,50,1);
Insert into room(‘B101’,’1’,’B’,50,1);
Insert into room(3,‘A301’,’3’,’A’,10,1);
Insert technician(1,2);
Insert into lab(‘Physics Lab’,’B101’,1); insert into employee
values(1,'Mahesh',28,'Cleaner',date '2018-02-20',15000,5,default); insert into employee
values(2,'Mukesh',28,'Technician',date '2020-03-18',25000,3,default); insert into employee
values(3,'Gopal Sharma',35,'Teacher',date '2018-09-18',55000,5,default); Insert into
cleaner(1,1);
Insert into room values( 1,‘A101’,’1’,’A’,50,1);
Insert into room values(2,‘A102’,’1’,’A’,50,1);
Insert into room values(‘B101’,’1’,’B’,50,1);
Insert into room values(‘B101’,’1’,’B’,50,1);
Insert into room values(3,‘A301’,’3’,’A’,10,1);
Insert technician values(1,2);
Insert into lab values(‘Physics Lab’,’B101’,1);
Insert into cleaner(1,1);
Insert into cleaner values(1,1);
Insert into room values( 1,‘A101’,’1’,’A’,50,1);
Insert into room values( 1,'A101','1','A101',50,1);
Insert into room values( 1,'A101','1','A',50,1);
Insert into room values( 'A101','1','A',50,1); Insert
into room values('A102','1','A',50,1);
Insert into room values('B101','1','B',50,1);
Insert into room values('B101','1','B',50,1);
Insert into room values('B102','1','B',50,1);
Insert into room values('A301','3','A',10,1);
```

```
Insert technician values(1,2);

Insert into technician values(1,2);

Insert into lab values('Physics Lab','B101',1); insert into stream
values(1,'PCM'); insert into class values(1,'11','A','A101',1,30);
insert into class values(1,'11','A','A101',1,30); insert into class
values(2,'11','B','A102',2,30); insert into teachers
values(3,1,'UCS001'); insert into teacher values(3,1,'UCS001'); insert
into teacher values(3,1,'UCS001'); insert into teacher
values(1,3,1,'UCS001'); insert into enroll values(1,date '2020-04-
15'); insert into students values(1,'Manik',1,1,date '2005-04-
25',1,'Atul'); insert into
healt_data(1,1,170,62,19,17,'M','O+','NONE'); insert into healt_data
values (1,1,170,62,19,17,'M','O+','NONE'); insert into health_data
values (1,1,170,62,19,17,'M','O+','NONE'); insert into health_data
values (1,1,170,62,17,'M','O+','NONE'); insert into lab_name
values('B102',2); insert into lab values('B102',2); insert into
class_assigned values(1,1); insert into class_assigned values(1,2);
insert into class_assigned values(2,1); insert into class_assigned
values(2,2); insert into class_assigned values(3,2); insert into
class_assigned values(4,1); insert into class_assigned values(4,2);
insert into class_assigned values(5,1); insert into class_assigned
values(5,2); insert into class_assigned values(5,3); insert into
class_assigned values(6,3); insert into class_assigned values(6,4);
insert into class_assigned values(7,3); insert into class_assigned
values(7,4); insert into class_assigned values(8,3);

insert into class_assigned values(8,4);
```

```
begin  add_subject('UMA001','Mathematics');

add_subject('UCS001','Physics');

add_subject('UCS002','Chemistry');

add_subject('UCS003','Biology');

add_subject('COM001','Bussiness Studies');

add_subject('COM002','Economics');

add_subject('COM003','Accounts');

add_subject('UMA002','Integration');

    end;

/
begin  add_hod('Dr.Sumit',50,'20-feb-
2008',75000);  add_HOD('Sumit',45,'12-Dec-
2002',65000);  add_HOD('Rajesh',51,'13-dec-
1998',78000);  add_HOD('Rahul',48,'04-feb-
2002',74000);  end;  / begin
add_cleaner('anjali',27,'12-dec-2018',17000);
add_cleaner('Mahesh',39,'14-aug-2011',12000);
add_cleaner('Raju',36,'19-aug-2015',15000);
add_cleaner('Ramu',38,'12-apr-2012',16000);
end;  / begin  add_room('A302','3','A',10);
add_room('A302','3','A',10);
add_room('A303','3','A',10);
add_room('A304','3','A',10);
add_room('A103','1','A',35);
add_room('A104','1','A',35); end;  / declare
```

```
    id number(2);  begin  id := add_student('Bhavesh',1,'20-Apr-2005',1,'Rupesh','15-Apr-
2022',169,70,'M','B+','Non');  id := add_student('Pratham',1,'18-Apr-2004',1,'Rinku','16-Apr-
2022',189,73,'M','A+','Non');  id := add_student('Sannidhhya',1,'15-Apr-2003',1,'Vinod','15-
Apr-2022',180,72,'M',')+','Non'); id := add_student('mansi',1,'04-sep-2003',1,'Umesh','15-Apr-
2022',160,56,'F','B+','Non');  id := add_student('Saanvi',1,'15-jan-2003',1,'Satya','15-Apr-
2022',160,52,'F','A+','Non');  id := add_student('Kartikay',2,'13-Apr-2003',2,'Preeti','15-jan-
2022',157,52,'M','B+','Non');  id := add_student('Abhinav',2,'17-jul-2003',2,'Rinku','15-Apr-
2022',185,75,'M','A+','Non');  id := add_student('Ria',2,'14-may-2003',2,'Rinki','15-Apr-
2022',170,62,'F','B+','Non');  id := add_student('Shivam',2,'18-nov-2003',2,'Lata','15-mar-
2022',186,79,'M','A+','Non');  id := add_student('Aditi',2,'15-dec-2003',2,'Nishu','17-jan-
2022',160,57,'F','B+','Non');  id := add_student('Hargun',3,'17-oct-2003',3,'Neelu','15-Apr-
2022',190,82,'M','A+','Non');  id := add_student('Gurpreet',3,'05-aug-2003',3,'Manpreet','15-
mar-2022',187,85,'M','B+','Non');  id := add_student('Akriti',3,'14-mar-2003',3,'Aditya','19-
mar-2022',150,62,'F','A-','Non');  id := add_student('Ishu',3,'08-jul-2003',3,'Ansh','15-Apr-
2022',160,52,'F','B-','Non');  id := add_student('Anmol',4,'16-Apr-2003',4,'Suresh','17-Apr-
2022',190,72,'M','O-','Non');  id := add_student('Suhani',4,'21-Apr-2003',4,'Pankaj','15-Apr-
2022',150,48,'F','AB+','Non');  id := add_student('Sankalp',4,'15-sep-2003',4,'Ronit','19-jun-
2022',165,67,'M','AB+','Non');  id := add_student('Sunidhi',4,'02-dec-2003',4,'Sanjay','01-may-
2022',159,53,'F','A-','Non');  id := add_student('Rupinder',4,'15-feb-2003',4,'Harpreet','11-
Apr-2022',187,74,'M','B-','Non');  id := add_student('Surinder',4,'13-oct-
2003',4,'Ravinder','14-Apr-2022',157,56,'F','O-','Non');  end;  / begin
add_department('Commerce',10,'A303');  add_department('Maths',11,'A304');
 end;
/  begin  add_teacher('Ajay',32,'16-09-
2014',46000,'UCS310',1);  add_teacher('Ajay',32,'16-jan-
2014',46000,'UCS310',1);  add_teacher('Shanu kuttan',42,'19-
jul-2008',49000,'UCS002',1);  add_teacher('Jayanti',44,'12-
jul-2009',53000,'UCS003',1);
```

```
add_teacher('Nikhil',48,'12-jul-2005',55000,'UMA001',2);

add_teacher('Abdul Bari',53,'11-jun-2002',76000,'COM001',3);

add_teacher('Ritu',46,'11-jun-2013',48000,'COM002',3);

add_teacher('Vikas',26,'19-jun-2021',36000,'COM003',3);

end;  / begin add_technician('Suman',29,'05-jun-

2018',31000);  add_technician('Daya',31,'07-jul-

2017',32000); end;  / begin

add_class('11','C','A103','Commerce,Maths',35);

add_class('11','D','A104','Commerce,No Maths',35);

/ begin
add_emp_phone(1,'9915600776');
add_emp_phone(2,'9915600702');
add_emp_phone(3,'9915600703');
add_emp_phone(4,'9915600704');
add_emp_phone(5,'9915600705');
add_emp_phone(6,'9915600706');
add_emp_phone(7,'9915600707');
add_emp_phone(8,'9915600708');
add_emp_phone(9,'9915600709');
add_emp_phone(10,'9915600710');
add_emp_phone(11,'9915600711');
add_emp_phone(12,'9915600712');
add_emp_phone(13,'9915600713');
add_emp_phone(14,'9915600714');
add_emp_phone(15,'9915600715');
add_emp_phone(16,'9915600716');
add_emp_phone(17,'9915600717');
add_emp_phone(18,'9915600718');
add_emp_phone(19,'9915600719');
add_emp_phone(20,'9915600720');
add_emp_phone(21,'9915600721');
```

```
add_emp_phone(22,'9915600722');

add_emp_phone(23,'9915600723');

end;  / begin

add_emp_mail(1,'emp1@BCM.edu');

add_emp_mail(2,'emp2@BCM.edu');

add_emp_mail(3,'emp3@BCM.edu');

add_emp_mail(4,'emp4@BCM.edu');

add_emp_mail(5,'emp5@BCM.edu');

add_emp_mail(6,'emp6@BCM.edu');

add_emp_mail(7,'emp7@BCM.edu');

add_emp_mail(8,'emp8@BCM.edu');

add_emp_mail(9,'emp9@BCM.edu');

add_emp_mail(10,'emp10@BCM.edu');

add_emp_mail(11,'emp11@BCM.edu');

add_emp_mail(12,'emp12@BCM.edu');

add_emp_mail(13,'emp13@BCM.edu');

add_emp_mail(14,'emp14@BCM.edu');

add_emp_mail(15,'emp15@BCM.edu');

add_emp_mail(16,'emp16@BCM.edu');

add_emp_mail(17,'emp17@BCM.edu');

add_emp_mail(18,'emp18@BCM.edu');

add_emp_mail(19,'emp19@BCM.edu');

add_emp_mail(20,'emp20@BCM.edu');

add_emp_mail(21,'emp21@BCM.edu');

add_emp_mail(22,'emp22@BCM.edu');

add_emp_mail(23,'emp23@BCM.edu');

add_emp_mail(24,'emp24@BCM.edu');

end;

/  begin
add_student_phone(1,'8433421992');
add_student_phone(2,'8433421902');
add_student_phone(3,'8433421903');
```

```
add_student_phone(4,'8433421904');  add_student_phone(5,'8433421905');

add_student_phone(6,'8433421906');  add_student_phone(7,'8433421907');

add_student_phone(8,'8433421908');  add_student_phone(9,'8433421909');

add_student_phone(10,'8433421910');

add_student_phone(11,'8433421911');

add_student_phone(12,'8433421912');

add_student_phone(13,'8433421913');

add_student_phone(14,'8433421914');

add_student_phone(15,'8433421915');

add_student_phone(16,'8433421916');

add_student_phone(17,'8433421917');

add_student_phone(18,'8433421918');

add_student_phone(19,'8433421919');

add_student_phone(20,'8433421920');

add_student_phone(21,'8433421921');

 begin
add_grade(1,'UCS001',60);
add_grade(2,'UMA002',50);
add_grade(3,'UMA001',70);
add_grade(4,'UCS001',64);
add_grade(5,'UCS002',62);
add_grade(6,'COM001',53);
add_grade(7,'UCS001',98);
add_grade(8,'COM003',97);
add_grade(9,'COM002',23);
add_grade(10,'UCS003',63);
add_grade(11,'UMA001',47);
add_grade(12,'UMA002',87);
add_grade(13,'UMA001',52);
add_grade(14,'COM003',72);
add_grade(15,'COM001',80);
add_grade(16,'COM002',34);
```

```
add_grade(17,'COM001',54);

add_grade(18,'UCS003',67);

add_grade(19,'UCS002',90);

add_grade(20,'UCS003',31);

add_grade(21,'UCS002',71); end;/
```

View Of all The Tables

1) Employee Table

EMP_ID	NAME	AGE	DESIGNATION	DATE_OF_JOINING	SALARY	WORK_YEARS	LEAVE_LEFT
1	Mahesh	28	Cleaner	20-FEB-18	15000	5	12
2	Mukesh	28	Technician	18-MAR-20	25000	3	12
3	Gopal Sharma	35	Teacher	18-SEP-18	55000	5	12
4	Sumit	45	HOD	12-DEC-02	65000	20	12
5	Dr.Sumit	50	HOD	20-FEB-08	75000	15	12
6	anjali	27	Cleaner	12-DEC-18	17000	4	12
7	Sumit	45	HOD	12-DEC-02	65000	20	12
8	Dr.Sumit	50	HOD	20-FEB-08	75000	15	12
9	anjali	27	Cleaner	12-DEC-18	17000	4	12
10	Rajesh	51	HOD	13-DEC-98	78000	24	12
11	Rahul	48	HOD	04-FEB-02	74000	21	12
12	Mahesh	39	Cleaner	14-AUG-11	12000	11	12
13	Raju	36	Cleaner	19-AUG-15	15000	7	12
14	Ramu	38	Cleaner	12-APR-12	16000	11	12
15	Suman	29	Technician	05-JUN-18	31000	4	12
16	Daya	31	Technician	07-JUL-17	32000	5	12
17	Daya	31	Technician	03-SEP-19	25000	3	12
18	Shanu kuttan	42	Teacher	19-JUL-08	49000	14	12
19	Jayanti	44	Teacher	12-JUL-09	53000	13	12
20	Jayanti	44	Teacher	12-JUL-09	53000	13	12
21	Nikhil	48	Teacher	12-JUL-05	55000	17	12
22	Abdul Bari	53	Teacher	11-JUN-02	76000	20	12
23	Ritu	46	Teacher	11-JUN-13	48000	9	12
24	Vikas	26	Teacher	19-JUN-21	36000	1	12

2) Employee Phone

EMP_ID	PHONE_NO
1	9915600776
2	9915600702
3	9915600703
4	9915600704
5	9915600705
6	9915600706
7	9915600707
8	9915600708
9	9915600709
10	9915600710
11	9915600711
12	9915600712
13	9915600713
14	9915600714
15	9915600715
16	9915600716
17	9915600717
18	9915600718
19	9915600719
20	9915600720
21	9915600721
22	9915600722
23	9915600723

3) Employee Email

EMP_ID	EMAIL
1	emp1@BCM.edu
2	emp2@BCM.edu
3	emp3@BCM.edu
4	emp4@BCM.edu
5	emp5@BCM.edu
6	emp6@BCM.edu
7	emp7@BCM.edu
8	emp8@BCM.edu
9	emp9@BCM.edu
10	emp10@BCM.edu
11	emp11@BCM.edu
12	emp12@BCM.edu
13	emp13@BCM.edu
14	emp14@BCM.edu
15	emp15@BCM.edu
16	emp16@BCM.edu
17	emp17@BCM.edu
18	emp18@BCM.edu
19	emp19@BCM.edu
20	emp20@BCM.edu
21	emp21@BCM.edu
22	emp22@BCM.edu
23	emp23@BCM.edu
24	emp24@BCM.edu

4) Cleaner

CLEANER_ID	EMP_ID
1	1
2	6
3	9
4	12
5	13
6	14

5) Technician

TECHNICIAN_ID	EMP_ID
1	2
2	15
3	16
4	17

6) Room

ROOM_NO	FLOOR	WING	CAPACITY	ASSIGNED_CLEANERID
A101	1	A	50	1
A102	1	A	50	1
B101	1	B	50	1
B102	1	B	50	1
A301	3	A	10	1
A302	3	A	10	2
A303	3	A	10	2
A304	3	A	10	3
A103	1	A	35	6
A104	1	A	35	6

7) Lab

LAB_NAME	ROOM_NO	TECHNICIAN_ID
Physics Lab	B101	1
Chemistry lab	B102	2

8) Subjects

SUBJECT_CODE	SUBJECT_NAME
UMA001	Mathematics
UCS001	Physics
UCS002	Chemistry
UCS003	Biology
COM001	Bussiness Studies
COM002	Economics
COM003	Accounts
UMA002	Integration

9) Stream

STREAM_ID	STREAM_NAME
1	PCM
2	PCMB
3	Commerce,Maths
4	Commerce,No Maths

10) Class

CLASS_ID	STANDARD	SECTION_NAME	ROOM_NO	STREAM	STRENGTH
4	11	D	A104	4	45
1	11	A	A101	1	50
2	11	B	A102	2	50
3	11	C	A103	3	45

11) Department

DEPARTMENT_ID	DEPARTMENT_NAME	HOD_ID	ROOM_NO
1	Science	1	A301
2	Commerce	10	A303
3	Maths	11	A304

12) Teachers

TEACHER_ID	EMP_ID	DEPT_ID	SUBJECT
1	3	1	UCS001
2	18	1	UCS002
3	19	1	UCS003
4	20	2	UMA002
5	21	2	UMA001
6	22	3	COM001
7	23	3	COM002
8	24	3	COM003

13) Students

STUDENT_ID	NAME	ENROLLMENT_ID	CLASS_ID	DOB	STREAM	PARENT_NAME
8	Abhinav	8	2	17-JUL-03	2	Rinku
11	Aditi	11	2	15-DEC-03	2	Nishu
6	Saanvi	6	1	15-JAN-03	1	Satya
7	Kartikay	7	2	13-APR-03	2	Preeti
9	Ria	9	2	14-MAY-03	2	Rinki
1	Manik	1	1	25-APR-05	1	Atul
2	Bhavesh	2	1	20-APR-05	1	Rupesh
3	Pratham	3	1	18-APR-04	1	Rinku
4	Sannidhhya	4	1	15-APR-03	1	Vinod
5	mansi	5	1	04-SEP-03	1	Umesh
10	Shivam	10	2	18-NOV-03	2	Lata
12	Hargun	12	3	17-OCT-03	3	Neelu
13	Gurpreet	13	3	05-AUG-03	3	Manpreet
14	Akriti	14	3	14-MAR-03	3	Aditya
15	Ishu	15	3	08-JUL-03	3	Ansh
16	Anmol	16	4	16-APR-03	4	Suresh
17	Suhani	17	4	21-APR-03	4	Pankaj
18	Sankalp	18	4	15-SEP-03	4	Ronit
19	Sunidhi	19	4	02-DEC-03	4	Sanjay
20	Rupinder	20	4	15-FEB-03	4	Harpreet
21	Surinder	21	4	13-OCT-03	4	Ravinder

14) Enroll

ENROLL_ID	DATE_OF_JOINING	10	15-MAR-22
8	15-APR-22	12	15-APR-22
11	17-JAN-22	13	15-MAR-22
6	15-APR-22	14	19-MAR-22
7	15-JAN-22	15	15-APR-22
9	15-APR-22	16	17-APR-22
1	15-APR-20	17	15-APR-22
2	15-APR-22	18	19-JUN-22
3	16-APR-22	19	01-MAY-22
4	15-APR-22	20	11-APR-22
5	15-APR-22	21	14-APR-22

15) Health_data

INSURANCE_ID	STUDENT_ID	HEIGHT_IN_CM	WEIGHT_IN_KG	AGE	GENDER	BLOOD_GROUP	MEDICAL_CONDITIONS
8	8	185	75	19	M	A+	Non
11	11	160	57	19	F	B+	Non
6	6	160	52	20	F	A+	Non
7	7	157	52	20	M	B+	Non
9	9	170	62	19	F	B+	Non
1	1	170	62	17	M	O+	NONE
2	2	169	70	18	M	B+	Non
3	3	189	73	19	M	A+	Non
4	4	180	72	20	M	)+	Non
5	5	160	56	19	F	B+	Non
10	10	186	79	19	M	A+	Non
12	12	190	82	19	M	A+	Non
13	13	187	85	19	M	B+	Non
14	14	150	62	20	F	A-	Non
15	15	160	52	19	F	B-	Non
16	16	190	72	20	M	O-	Non
17	17	150	48	20	F	AB+	Non
18	18	165	67	19	M	AB+	Non
19	19	159	53	19	F	A-	Non
20	20	187	74	20	M	B-	Non
21	21	157	56	19	F	O-	Non

16) Student_phone

STUDENT_ID	PHONE
1	8433421992
2	8433421902
3	8433421903
4	8433421904
5	8433421905
6	8433421906
7	8433421907
8	8433421908
9	8433421909
10	8433421910
11	8433421911
12	8433421912
13	8433421913
14	8433421914
15	8433421915
16	8433421916
17	8433421917
18	8433421918
19	8433421919
20	8433421920
21	8433421921

17) Grades

12	UMA002	87	B
13	UMA001	52	E
14	COM003	72	C
15	COM001	80	B
16	COM002	34	F
17	COM001	54	E
18	UCS003	67	D
19	UCS002	90	A
20	UCS003	31	F
21	UCS002	71	C
1	UCS001	60	D

STUDENT_ID	SUBJECT_CODE	MARKS	GRADE
2	UMA002	50	E
3	UMA001	70	C
4	UCS001	64	D
5	UCS002	62	D
6	COM001	53	E
7	UCS001	98	A
8	COM003	97	A
9	COM002	23	F
10	UCS003	63	D
11	UMA001	47	F

18) Class_assigned

TEACHER_ID	CLASS_ID
1	1
1	2
2	1
2	2
3	2
4	1
4	2
5	1

5	2
5	3
6	3
6	4
7	3
7	4
8	3
8	4

QUERIES

1. Display name of teachers having salary less than 50000.

```
select name from employee e join teacher t on e.emp_id=t.emp_id where
e.salary<50000;
```

NAME
Shanu kuttan
Ritu
Vikas

2. Calculate body mass index of student.

```
select
s.name,h.height_in_cm,h.weight_in_kg,(h.weight_in_kg/((h.height_in_cm/10
0)*(h.height_in_cm/100)))      as BMI
    from students s join
    health_data h on s.student_id=h.student_id
where
s.student_id=2;
```

NAME	HEIGHT_IN_CM	WEIGHT_IN_KG	BMI
Bhavesh	169	70	24.508945765204299569342810125695878995 83

3. Calculating work experience of teacher

```
select e.name,sum(e.work_years) as experience from teacher t
join employee e on t.emp_id=e.emp_id group by
name,work_years;
```

NAME	EXPERIENCE
Jayanti	26
Abdul Bari	20
Shanu kuttan	14
Nikhil	17
Ritu	9
Gopal Sharma	5
Vikas	1

4. Students having grade<=C

```
select s.student_id ,s.name
from students s join
grades g on s.student_id=g.student_id
join subjects sub on g.subject_code=sub.subject_code
where g.grade<'C';
```

STUDENT_ID	NAME
8	Abhinav
7	Kartikay
12	Hargun
15	Ishu
19	Sunidhi

5. Show Name of cleaners with rooms assigned

```
select e.name,m.room_no as room_name from cleaner c  join
employee e on c.emp_id=e.emp_id
join room m on m.assigned_cleanerid=c.emp_id;
```

NAME	ROOM_NAME
Mahesh	A101
Mahesh	A102
Mahesh	B101
Mahesh	B102
Mahesh	A301
anjali	A103
anjali	A104

6. Total salaries of different employees

```
select designation,sum(salary) from employee group
by designation order by sum(salary) DESC;
```

DESIGNATION	SUM(SALARY)
HOD	432000
Teacher	425000
Technician	113000
Cleaner	92000

7. Using Cursor to display employee name ,salary.

```
DECLARE
  v_emp_id employee.emp_id%TYPE;
v_name employee.name%TYPE;
v_salary employee.salary%TYPE;
  CURSOR c_employee IS
    SELECT emp_id, name, salary
    FROM employee;
BEGIN
  OPEN c_employee;
  LOOP
    FETCH c_employee INTO v_emp_id, v_name, v_salary;
    EXIT WHEN c_employee%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name || ', Salary: ' || v_salary);
END LOOP;
  CLOSE c_employee;
END;
```

```
Statement processed.
Name: Mahesh, Salary: 15000
Name: Mukesh, Salary: 25000
Name: Gopal Sharma, Salary: 55000
Name: Sumit, Salary: 65000
Name: Dr.Sumit, Salary: 75000
Name: anjali, Salary: 17000
Name: Sumit, Salary: 65000
Name: Dr.Sumit, Salary: 75000
Name: anjali, Salary: 17000
Name: Rajesh, Salary: 78000
Name: Rahul, Salary: 74000
Name: Mahesh, Salary: 12000
Name: Raju, Salary: 15000
Name: Ramu, Salary: 16000
Name: Suman, Salary: 31000
Name: Daya, Salary: 32000
Name: Daya, Salary: 25000
Name: Shanu kuttan, Salary: 49000
Name: Jayanti, Salary: 53000
Name: Jayanti, Salary: 53000
Name: Nikhil, Salary: 55000
Name: Abdul Bari, Salary: 76000
Name: Ritu, Salary: 48000
Name: Vikas, Salary: 36000
```

8. Using cursor to display name and salary of employees who have joined before a particular date.

```
DECLARE
  v_name employee.name%TYPE;
v_salary employee.salary%TYPE;
  CURSOR c_employee IS
    SELECT name, salary
    FROM employee
    WHERE date_of_joining < TO_DATE('2015-02-15', 'YYYY-MM-DD'); BEGIN
  OPEN c_employee;
  LOOP
    FETCH c_employee INTO v_name, v_salary;
    EXIT WHEN c_employee%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name || ', Salary: ' || v_salary);
END LOOP;
  CLOSE c_employee;
END;
```

```
Statement processed.
Name: Sumit, Salary: 65000
Name: Dr.Sumit, Salary: 75000
Name: Sumit, Salary: 65000
Name: Dr.Sumit, Salary: 75000
Name: Rajesh, Salary: 78000
Name: Rahul, Salary: 74000
Name: Mahesh, Salary: 12000
Name: Ramu, Salary: 16000
Name: Shanu kuttan, Salary: 49000
Name: Jayanti, Salary: 53000
Name: Jayanti, Salary: 53000
Name: Nikhil, Salary: 55000
Name: Abdul Bari, Salary: 76000
Name: Ritu, Salary: 48000
```

9. Using cursor to display maximum salary from employee table.

```
DECLARE
  max_sal NUMBER(6);
emp_sal NUMBER(6);
  CURSOR c_emp IS
    SELECT salary FROM employee;
BEGIN
  OPEN c_emp;
  FETCH c_emp INTO emp_sal;
max_sal := emp_sal;
  WHILE c_emp%FOUND LOOP
    IF emp_sal > max_sal THEN
max_sal := emp_sal;
    END IF;
    FETCH c_emp INTO emp_sal;
  END LOOP;
  CLOSE c_emp;
  DBMS_OUTPUT.PUT_LINE('Maximum salary is ' || max_sal);
END;
```

```
Statement processed.
Maximum salary is 78000
```

Triggers

1.Trigger to update grade when marks are updated

```
CREATE OR REPLACE TRIGGER update_grade
BEFORE UPDATE ON grades
FOR EACH ROW
BEGIN
    IF :OLD.marks <> :NEW.marks THEN
        IF :NEW.marks >= 90 THEN
            :NEW.grade := 'A';
        ELSIF :NEW.marks >= 80 THEN
            :NEW.grade := 'B';
        ELSIF :NEW.marks >= 70 THEN
            :NEW.grade := 'C';
        ELSIF :NEW.marks >= 60 THEN
            :NEW.grade := 'D';
        ELSIF : NEW.marks >= 50
THEN              :NEW.grade := 'E';
ELSE
            :NEW.grade := 'F';
        END IF;
    END IF;
END;
/
```

```
update grades set marks = 70
where student_id = 7;
```

Before

7	UCS001	98	A

After

7	UCS001	70	C