A
Project Report
on

UNIVERSITY EXAMINATION MANAGEMENT SYSTEM

Developed by

DARSHAKKUMAR KATHIRIYA (IT053)
NIHAL LIMBANI (IT060)

Guided By
Internal Guide:
PROF. MUKESH M. GOSWAMI
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
October – 2020

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled **"University Examination Management System"** is a bonafied report of the work carried out by

1) Mr Darshakkumar Kathiriya, Student ID No: 18ITUOS097
2) Mr Nihal Limbani, Student ID No: 18ITUOF051

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2020-2021.

Prof. Mukesh M. Goswami
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

On completion of project we would like to express our sincere thanks to all those who have guided, advised, inspired and supported during our project work.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. We hereby avail this opportunity to express our heartfelt gratitude to a number of people who extended their value time, full support and cooperation in developing this project.

We are heartily thankful to the qualified staff of the center and especially to our lab faculty Prof Mukesh M Goswami. We believe that her computer expertise and valuable guidance have made it possible to present such nice project report.

Thanking you,

Yours sincerely,

DARSHAKKUMAR KATHIRIYA (IT-053)

NIHAL LIMBANI (IT-060)

TABLE OF CONTENTS

SYSTEM OVERVIEW

1.Project Definition Scope

- Definition:

UNIVERSITY EXAMINATION MANAGEMENT SYSTEM

In this project, we are going to show how examination get handled in university. We will try to cover all the Courses in University for Examination.

- Purpose:

- The main objective of the Examination Management System is to manage the details of Examination , Branch , Marks, Courses, Session.
- It manages all the information about Examination, Student, Session, Examination
- The purpose of the project is to build an application program to reduce the manual work for managing the Examination, Branch, Student, Marks.
- It tracks all the details about the Marks, Courses, Session.

- Scope and uses:

  - This can be used for any university for their examination management as requirements.
  - Design for user as administrator, student and professor and also there are certain permission defined for security.
  - It's Stores data related exam as students information, professors information and all other exam related information such cours, department, subjects and results etc.

- Can be used anywhere anytime as it is web based application.

- Technology and Literature Review:

  - This project is based on Pl/SQl language.
  - Most of use Oracle database.

(Based On Time In future)

- We will try to create frontend related to this system with the use different languages Like javascript.
- And backend (language, server, database) try with node-js (javascript) And connect freamwork with express or any other.)

2.System Analysis (existing and new systems)

In Simple University Exam Management It's Difficult to manage all data.

By using web based system it's easy to maintaining all information related to exam.

- This System as Administrator can manages all the things related to Exam.
- Administrator also manage the student data, courses and professors.
- This system shows even details of the syllabus of subjects along with results of student in all the examinations taken.
- Student can know if there is allowed to appear in the examination based on attendance and fees payment.
- Student can even know his seating arrangement.
- The details of the bank in which fees is paid is also shown along with the fees id and fees amount of student.
- This system helps to know the current examination status of student as well as the information of the professor.
- System contain all records which help for university exam session

3.Requirements analysis (Roles , and functional requirements for each role, also list the reports needed for each role)

There are three type of Roles (users) require in this project :

1) Administrator : With full Access
2) Student : With Limited Access
3) Professor : With Limited Access

a) As Administrator:
    1. Login For Admin
    2. Manage Students: Add Students, View Details of the Students, Listing of all Students, Filter Students with their all information
    3. Manage Professor: Adding New Professor, Edit the Exiting Professor, View Profile of the Professor, Listing of all Professor
    4. Manage All requirements as: Courses, Departments, Exam all plan (syllabus, exam routine , Sub Information, Result information)

b) As Student:
    1. Login for Student
    2. Student can see own information
    3. Student also see their course along with department
    4. See their subject with professors
    5. See their exam related details and also own results
    6. See their attendance status and fees payments

c) As Professor:

1. Login for Professor
2. Professor can see and update own information
3. Professor also see their subject related course along with department

(IN our system Not Major Role of Professor but in Case of Online Examination system require than it's role major for Paper system or Checking Module.

E R DIAGRAM

4.Entity recognition (noun phrase analysis). Extract all nouns in the requirement description. Categorize nouns into entity, field or relation.

1) Login Users < Checked < Permission
2) As Administrator or Student or Professor
3) Student < own Information
4) Student < Has < Department.course
5) Student < enrolles < enrolled data <sits for< Exam < of< Subject< Routine < also gets < Result
6) Professor < own Information
7) Professor < belongs to <Department.course
8) Professor <teaches <Subjects
9) Administrator < own information
10) Administrator< manages < Students ,Professors
11) Administrator<manages<courses<<Department<<Exam<<Subject<<Syllabus<<Routine <<Supervisor<<Exam Attendance << Results(All the information and relations)

Entity Recognition:

1.) Login User
2.) Permission
3.) Student
4.) Administrator
5.) Professor
6.) Courses
7.) Department
8.) Exam

9.)Subject

10.) Attendance

11.) Fees

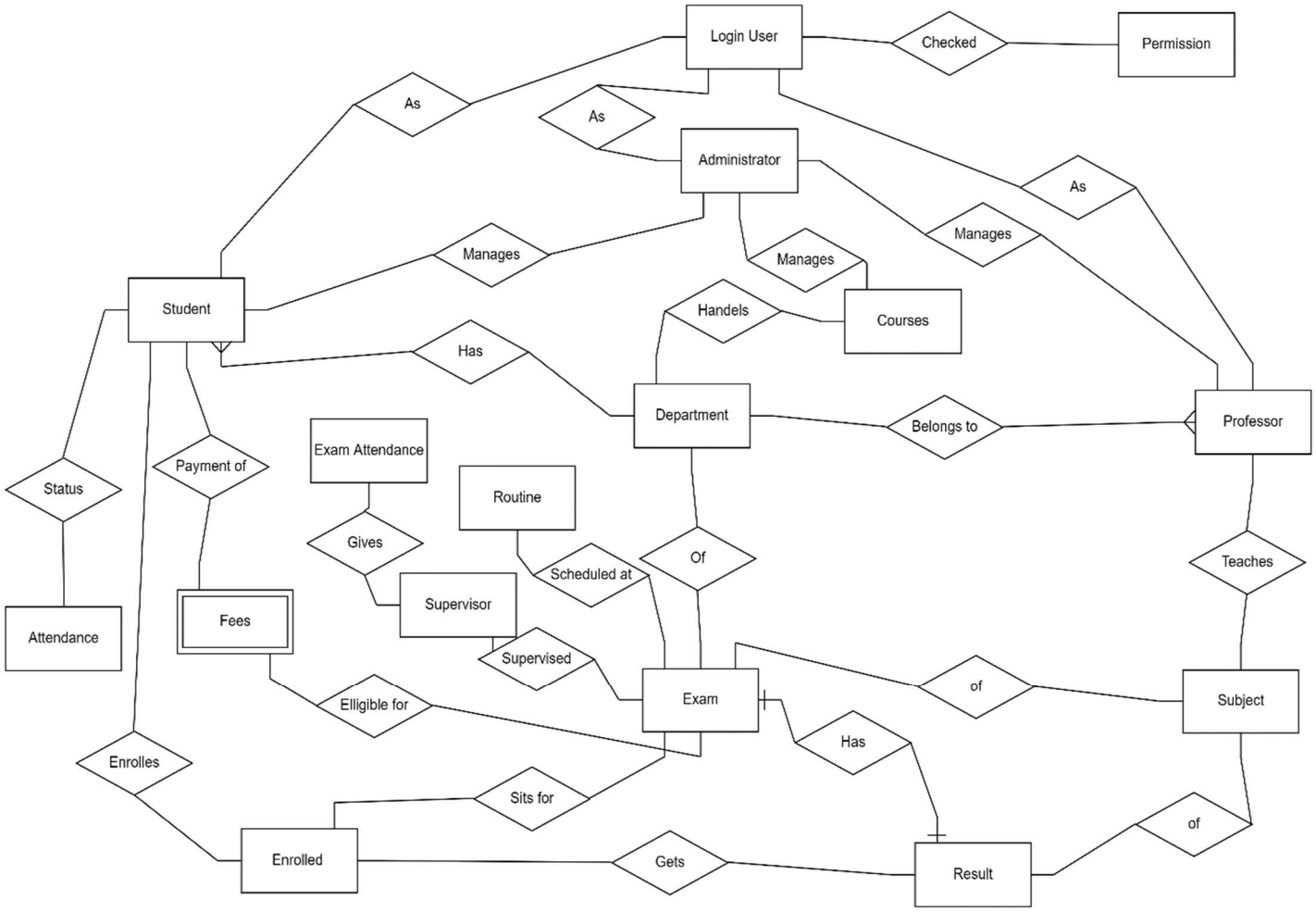
12.) Supervisor

13.) Routine

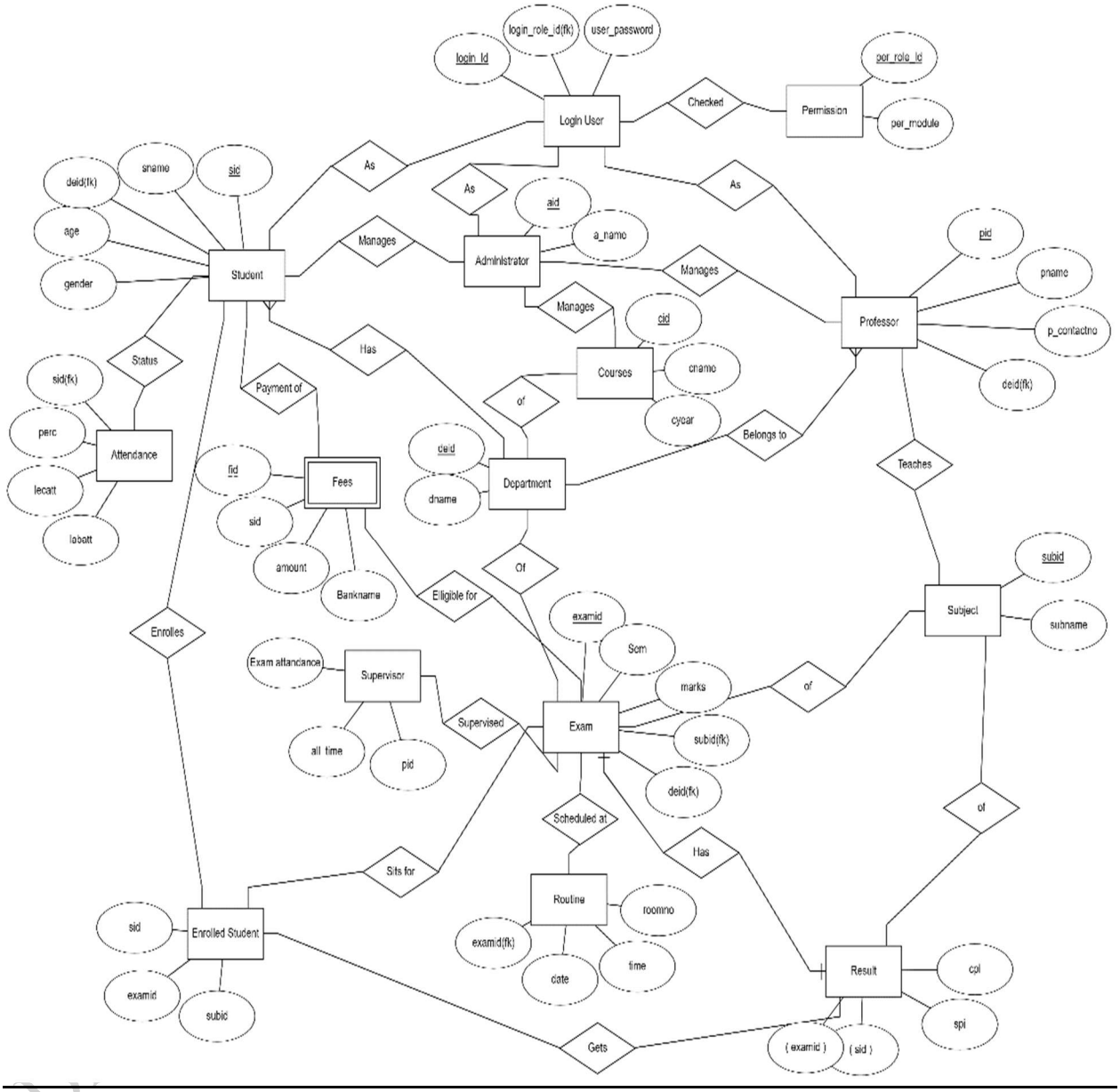
14.) Enrolled

15.) Results

5.Prepare a analysis level ER diagram (only entity and relation)



6.Prepare a design level ER (entity relationship fields):



DATA DICTIONARY

7.Prepare Data dictionary (details of each table structure with field name, type and constraint)

Login user :

Columns

#	Column	Type	Length	Precision	Scale	Null
1	LOGIN_ID	VARCHAR2	10			No
2	LOGIN_ROLE_ID	VARCHAR2	10			No
3	USER_PASSWORD	VARCHAR2	10			No

Constraints

Constraint	Type	Condition	On Delete	St
SYS_C0038858870	Check	"LOGIN_ROLE_ID" IS NOT NULL	-	ENABLE
SYS_C0038858871	Check	"USER_PASSWORD" IS NOT NULL	-	ENABLE
SYS_C0038858872	Primary Key	-	-	ENABLE

In our system login id is defined varchar with proper sequence. Because in our system only for exam section so there are other modules available in university so their defined other system.

Permission:

Columns

#	Column	Type	Length	Precision	Scale	Nullab
1	PER_ROLE_ID	VARCHAR2	10			Yes
2	PER_MODULE	VARCHAR2	50			Yes

In module its small work check in background who entered in system.

Administrator:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	AID	VARCHAR2	4			No
2	A_NAME	CHAR	20			No

Constraints

Constraint	Type	Condition	On Delete	Status
SYS_C0038858940	Check	"AID" IS NOT NULL	-	ENABLED
SYS_C0038858941	Check	"A_NAME" IS NOT NULL	-	ENABLED
SYS_C0038858942	Check	aid like 'A%'	-	ENABLED

In system aid is not defined auto increment, it's there are other admin available in other system.

And also it's proper sequence like start with A because there are many table id so we define unique start for data dictionary.

Courses:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	CID	VARCHAR2	4			No
2	CNAME	CHAR	20			No
3	CYEAR	NUMBER	22	1	0	Yes

Constraints

Constraint	Type	Condition	On Delete	Stat
SYS_C0038858879	Check	"CNAME" IS NOT NULL	-	ENABLED
SYS_C0038858880	Check	cid like 'C%'	-	ENABLED
SYS_C0038858881	Primary Key	-	-	ENABLED

As above mention there is apply in this table also id like cid.

And there it's ok if we decide cid auto increment but in our system only for exam section so we decide to it's course is varchar.

Department:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	DEID	VARCHAR2	4			No
2	DNAME	CHAR	25			No

Constraints

Constraint	Type	Condition	On Delete	Status
SYS_C0038858885	Check	"DNAME" IS NOT NULL	-	ENABLED
SYS_C0038858886	Check	deid like 'D%'	-	ENABLED
SYS_C0038858887	Primary Key	-	-	ENABLED

It's defined as course table.

Student:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	SID	VARCHAR2	10			No
2	SNAME	CHAR	25			No
3	DEID	VARCHAR2	4			No
4	DOB	DATE	7			Yes
5	GENDER	CHAR	1			Yes

Constraints

Constraint	Type	Condition	On Delete	Statu
SYS_C0038858888	Check	"SNAME" IS NOT NULL	-	ENABLED
SYS_C0038858889	Check	"DEID" IS NOT NULL	-	ENABLED
SYS_C0038858890	Check	sid like 'S%'	-	ENABLED
SYS_C0038858892	Foreign Key	-	NO ACTION	ENABLED
SYS_C0038858891	Primary Key	-	-	ENABLED

Sid varchar there can be main system university also has student module in this we guess it's auto increment.

Professor:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	PID	VARCHAR2	10			No
2	PNAME	CHAR	25			No
3	P_CONT	NUMBER	22	10	0	No
4	DEID	VARCHAR2	4			No

Constraints

Constraint	Type	Condition	On Delete	Statu
SYS_C0038858893	Check	"PNAME" IS NOT NULL	-	ENABLED
SYS_C0038858894	Check	"P_CONT" IS NOT NULL	-	ENABLED
SYS_C0038858895	Check	"DEID" IS NOT NULL	-	ENABLED
SYS_C0038858896	Check	pid like 'P%'	-	ENABLED
SYS_C0038858898	Foreign Key	-	NO ACTION	ENABLED
SYS_C0038858897	Primary Key	-	-	ENABLED

Professor module it's manageable because as we discuss requirement analysis here not professor as main module But if we decide to online exam system so there are many work associate with professor.

Attendance:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	SID	VARCHAR2	10			No
2	PERC	NUMBER	22	3	0	No
3	LECATT	NUMBER	22	3	0	Yes
4	LABATT	NUMBER	22	3	0	Yes

Constraints

Constraint	Type	Condition	On Delete	Status
SYS_C0038858899	Check	"SID" IS NOT NULL	-	ENABLED
SYS_C0038858900	Check	"PERC" IS NOT NULL	-	ENABLED
SYS_C0038858901	Foreign Key	-	NO ACTION	ENABLED

Fees:

Columns

#	Column	Type	Length	Precision	Scale	Nullab
1	FID	VARCHAR2	10			No
2	SID	VARCHAR2	10			No
3	AMOUNT	NUMBER	22			No
4	AMOUNT_PAID	NUMBER	22			Yes
5	BANKNAME	CHAR	25			Yes

Constraints

Constraint	Type	Condition	On Delete	Stat
SYS_C0038858912	Check	"SID" IS NOT NULL	-	ENABLED
SYS_C0038858913	Check	"AMOUNT" IS NOT NULL	-	ENABLED
SYS_C0038858914	Check	fid like 'F%'	-	ENABLED
SYS_C0038858916	Foreign Key	-	NO ACTION	ENABLED
SYS_C0038858915	Primary Key	-	-	ENABLED

Here defined amount how much paid and amount paid other attribute so admin and student see there status paid or not.

Subject:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	SUBID	VARCHAR2	4			No
2	SUBNAME	CHAR	15			No

Constraints

Constraint	Type	Condition	On Delete	Statu
SYS_C0038858917	Check	"SUBNAME" IS NOT NULL	-	ENABLED
SYS_C0038858918	Check	subid like 'SB%'	-	ENABLED
SYS_C0038858919	Primary Key	-	-	ENABLED

In also subject we not set auto increment so in other university department many subject it's possible there are any other module.

Exam:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	EXAMID	VARCHAR2	10			No
2	SEM	NUMBER	22	1	0	Yes
3	MARKS	NUMBER	22	3	0	Yes
4	SUBID	VARCHAR2	4			No
5	DEID	VARCHAR2	4			No

Constraints

Constraint	Type	Condition	On Delete	Status
SYS_C0038858902	Check	"SUBID" IS NOT NULL	-	ENABLED
SYS_C0038858903	Check	"DEID" IS NOT NULL	-	ENABLED
SYS_C0038858904	Check	examid like 'E%'	-	ENABLED
SYS_C0038858906	Foreign Key	-	NO ACTION	ENABLED
SYS_C0038858905	Primary Key	-	-	ENABLED

Supervisor:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	PID	VARCHAR2	10			No
2	EXAMID	VARCHAR2	10			Yes
3	ROOMNO	NUMBER	22	3	0	Yes
4	ALL_TIME	VARCHAR2	8			Yes

Constraints

Constraint	Type	Condition	On Delete	Status
SYS_C0038858925	Check	"PID" IS NOT NULL	-	ENABLED

Enrolled_std:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	SID	VARCHAR2	10			Yes
2	EXAMID	VARCHAR2	10			Yes
3	SUBID	VARCHAR2	4			Yes
4	ROOMNO	NUMBER	22	3	0	Yes

It's help for student show exam data.

Routine:

Columns

#	Column	Type	Length	Precision	Scale	Nullable
1	EXAMID	VARCHAR2	10			Yes
2	DT	DATE	7			Yes
3	TIME	VARCHAR2	8			Yes

Result:

Columns

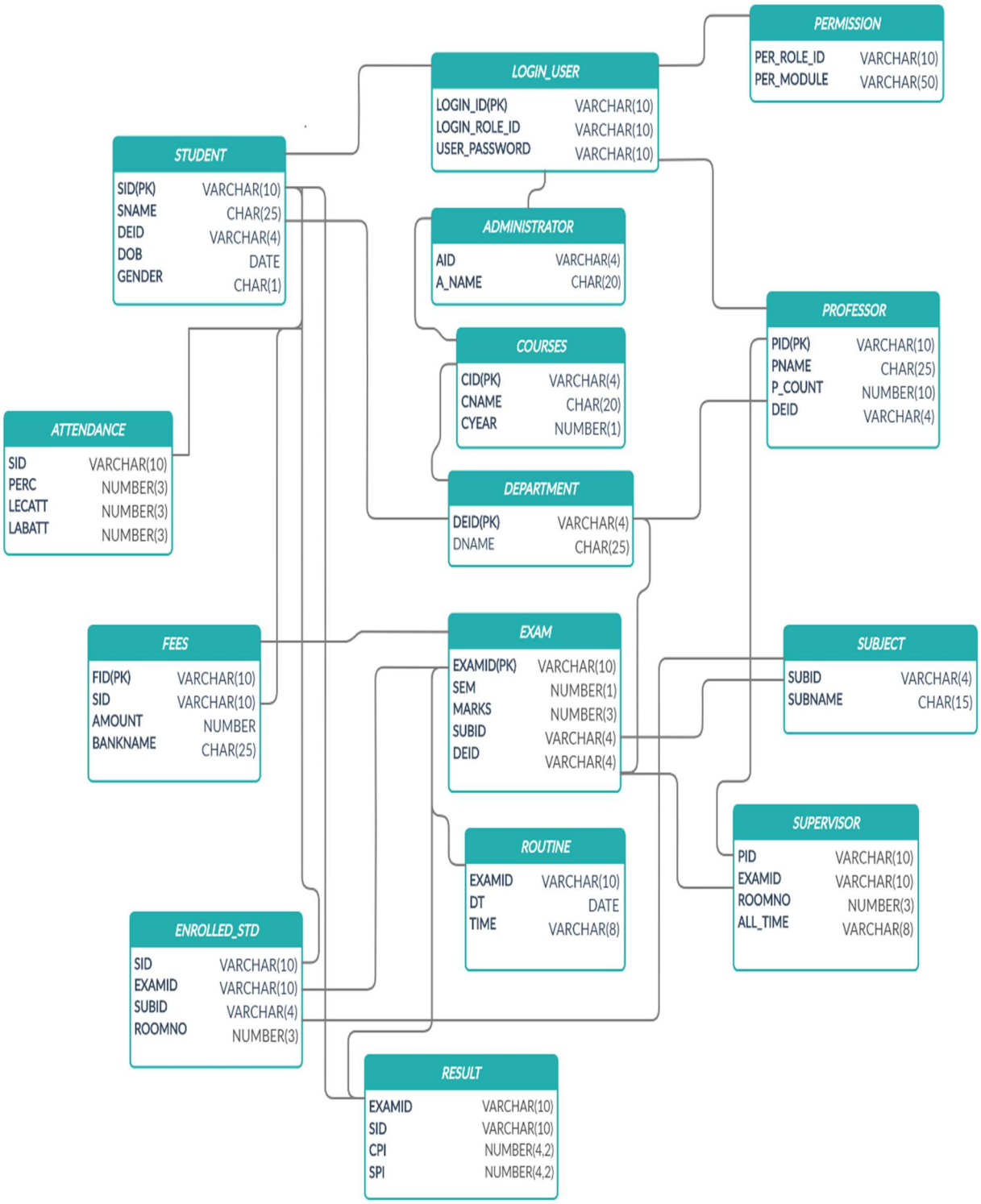
#	Column	Type	Length	Precision	Scale	Nullable
1	EXAMID	VARCHAR2	10			No
2	SID	VARCHAR2	10			No
3	SPI	NUMBER	22	4	2	No
4	CPI	NUMBER	22	4	2	No

Constraints

Constraint	Type	Condition	On Delete	Statu
SYS_C0038858934	Check	"EXAMID" IS NOT NULL	-	ENABLED
SYS_C0038858935	Check	"SID" IS NOT NULL	-	ENABLED
SYS_C0038858936	Check	"SPI" IS NOT NULL	-	ENABLED
SYS_C0038858937	Check	"CPI" IS NOT NULL	-	ENABLED
SYS_C0038858938	Foreign Key	-	NO ACTION	ENABLED
SYS_C0038858939	Foreign Key	-	NO ACTION	ENABLED

In spi and cpi there can point number so here scale 2 .

SCHEMA DIAGRAM



Implementation

8.1 Create Database Table

Login user :

```
create table loginuser(
        login_id varchar(10) primary key,
        login_role_id varchar(10) not null,
        user_password varchar(10) not null
        );
```

Permission:

```
create table permission(
        per_role_id varchar(10),
        per_module varchar(50)
        );
```

Administrator Login user:

```
create table administrator(
        aid varchar(4) not null check (aid like 'A%'),
        a_name char(20) not null
        );
```

Courses:

```
create table courses(
        cid varchar(4) primary key,
        cname char(20) not null,
```

```
    cyear number(1),
    check (cid like 'C%')
    );
```

Department:

```
create table department(
    deid varchar(4) primary key,
    dname char(25) not null,
    check (deid like 'D%')
    );
```

Student:

```
create table student(
    sid varchar(10) primary key,
    sname char(25) not null,
    deid varchar(4) not null,
    dob date,
    gender char(1),
    check (sid like 'S%'),
    foreign key(deid) references department(deid)
    );
```

Professor:

```
create table professor(
    pid varchar(10) primary key,
    pname char(25) not null,
    p_cont number(10) not null,
```

```
    deid varchar(4) not null,
    check (pid like 'P%'),
    foreign key(deid) references department(deid)
    );
```

Attendance:

```
create table attendance(
    sid varchar(10) not null,
    perc number(3) not null,
    lecatt number(3),
    labatt number (3),
    foreign key(sid) references student(sid)
    );
```

Fees:

```
create table fees(
    fid varchar(10) primary key,
    sid varchar(10) not null,
    amount number not null,
    amount_paid number ,
    bankname char(25),
    check (fid like 'F%'),
    foreign key(sid) references student(sid)
    );
```

Subject:

```
create table subject(
```

```
        subid varchar(4) primary key,
        subname char(15) not null,
        check (subid like 'SB%')
        );
```

Exam:

```
create table exam(
        examid varchar(10) primary key,
        sem number(1),
        marks number(3),
        subid varchar(4) not null,
        deid varchar(4) not null,
        check (examid like 'E%'),
        foreign key(deid) references department(deid)
);
```

Supervisor:

```
create table supervisor(
        pid varchar(10) not null,
        examid varchar(10),
        roomno number(3),
        all_time varchar(8)
);
```

Enrolled_std:

```
create table enrolled_std(
        sid varchar(10),
```

```
    examid varchar(10),
    subid varchar(4),
    roomno number(3)
    );
```

Routine:

```
create table routine(
    examid varchar(10),
    dt date,
    time varchar(8)
    );
```

Result:

```
create table result(
    examid varchar(10) not null,
    sid varchar(10) not null,
    spi number(4,2) not null,
    cpi number(4,2) not null,
    foreign key(sid) references student(sid),
    foreign key(examid) references exam(examid)
    );
```

8.2 Insert Update – Information

Login user :

```
INSERT INTO LOGINUSER(LOGIN_ID, LOGIN_ROLE_ID, USER_PASSWORD) VALUES
('10AD1','A1','AD991379');
INSERT INTO LOGINUSER(LOGIN_ID, LOGIN_ROLE_ID, USER_PASSWORD) VALUES
('10AD3','A1','A99179');
INSERT INTO LOGINUSER(LOGIN_ID, LOGIN_ROLE_ID, USER_PASSWORD) VALUES
('18ITUOS097','S1','13/07/2001');
INSERT INTO LOGINUSER(LOGIN_ID, LOGIN_ROLE_ID, USER_PASSWORD) VALUES
('18ITUOF051','S1','05/05/2001');
INSERT INTO LOGINUSER(LOGIN_ID, LOGIN_ROLE_ID, USER_PASSWORD) VALUES
('15POO1','P1','PR0909');
INSERT INTO LOGINUSER(LOGIN_ID, LOGIN_ROLE_ID, USER_PASSWORD) VALUES
('14P006','P1','P5590');
SELECT * FROM LOGINUSER;
```

LOGIN_ID	LOGIN_ROLE_ID	USER_PASSWORD
10AD1	A1	AD991379
10AD3	A1	A99179
18ITUOS097	S1	13/07/2001
18ITUOF051	S1	05/05/2001
15POO1	P1	PR0909
14P006	P1	P5590

Download CSV

6 rows selected.

Permission:

```
INSERT INTO PERMISSION VALUES
('A1','ALL PERMISSION INSERT VIEW DELETION GIVEN');
INSERT INTO PERMISSION VALUES
('S1','PERMISSION VIEW STUDENT GIVEN');
INSERT INTO PERMISSION VALUES
('P1','PERMISSION VIEW SOME DATA GIVEN');
SELECT * FROM PERMISSION;
```

PER_ROLE_ID	PER_MODULE
A1	ALL PERMISSION INSERT VIEW DELETION GIVEN
S1	PERMISSION VIEW STUDENT GIVEN
P1	PERMISSION VIEW SOME DATA GIVEN

Download CSV
3 rows selected.

Administrator Login user:

```
INSERT INTO ADMINISTRATOR VALUES
('A01','BHAVSHAR MAHESHSIR');
INSERT INTO ADMINISTRATOR VALUES
('A02','PRAJAPATI RAMESHSIR');
INSERT INTO ADMINISTRATOR VALUES
('A03','DESAI RAMSIR');
INSERT INTO ADMINISTRATOR VALUES
('A04','DESAI LAKHANSIR');
SELECT * FROM ADMINISTRATOR;
```

1 row(s) inserted.

AID	A_NAME
A01	BHAVSHAR MAHESHSIR
A02	PRAJAPATI RAMESHSIR
A03	DESAI RAMSIR
A04	DESAI LAKHANSIR

Download CSV
4 rows selected.

Courses:

```
INSERT INTO COURSES VALUES
('C01','ENGINEERING UGA','4');
INSERT INTO COURSES VALUES
('C02','MBA','2');
INSERT INTO COURSES VALUES
('C03','DENTAL','4');
INSERT INTO COURSES VALUES
('C04','BCA','3');
INSERT INTO COURSES VALUES
('C05','ENGINEERING PGA','2');
SELECT * FROM COURSES;
```

CID	CNAME	CYEAR
C01	ENGINEERING UGA	4
C02	MBA	2
C03	DENTAL	4
C04	BCA	3
C05	ENGINEERING PGA	2

Download CSV
5 rows selected.

Department:

```
INSERT INTO DEPARTMENT VALUES
('D01','INFORMATION TECHNOLOGY');
INSERT INTO DEPARTMENT VALUES
('D02','COMPUTER');
INSERT INTO DEPARTMENT VALUES
('D03','CIVIL');
INSERT INTO DEPARTMENT VALUES
('D011','FOOD MANAGEMENT');
INSERT INTO DEPARTMENT VALUES
('D012','EVENT MANAGEMENT');
INSERT INTO DEPARTMENT VALUES
('D021','DENTAL RADIOLOGIST');
SELECT * FROM DEPARTMENT;
```

DEID	DNAME
D01	INFORMATION TECHNOLOGY
D02	COMPUTER
D03	CIVIL
D011	FOOD MANAGEMENT
D012	EVENT MANAGEMENT
D021	DENTAL RADIOLOGIST

Download CSV
6 rows selected.

Student:

```
INSERT INTO STUDENT VALUES
('S01','DARSHAK KATHIRIYA','D01','13-JULY-2001','M');
INSERT INTO STUDENT VALUES
('S02','NIHAL LIMBANI','D01','05-AUG-2000','M');
```

```
INSERT INTO STUDENT VALUES
('S011','AYUSHI AJUDIYA','D02','10-JUNE-2000','F');
INSERT INTO STUDENT VALUES
('S012','VARJESH VADI','D02','13-JULY-2000','M');
INSERT INTO STUDENT VALUES
('S021','SRUSHTI PATEL','D011','20-JUNE-2002','F');
INSERT INTO STUDENT VALUES
('S022','AARYA MAHETA','D011','30-JAN-2002','M');
INSERT INTO STUDENT VALUES
('S031','DARSHAK KATHIRIYA','D012','01-FEB-2001','M');
INSERT INTO STUDENT VALUES
('S032','ANJAN VASOYA','D012','15-JUNE-2001','M');
INSERT INTO STUDENT VALUES
('S041','CHARMI BABARIYA','D021','05-JULY-2000','F');
INSERT INTO STUDENT VALUES
('S042','ADITYA PATEL','D021','14-JULY-2000','M');
SELECT * FROM STUDENT;
```

SID	SNAME	DEID	DOB	GENDER
S01	DARSHAK KATHIRIYA	D01	13-JUL-01	M
S02	NIHAL LIMBANI	D01	05-AUG-00	M
S011	AYUSHI AJUDIYA	D02	10-JUN-00	F
S012	VARJESH VADI	D02	13-JUL-00	M
S021	SRUSHTI PATEL	D011	20-JUN-02	F
S022	AARYA MAHETA	D011	30-JAN-02	M
S031	DARSHAK KATHIRIYA	D012	01-FEB-01	M
S032	ANJAN VASOYA	D012	15-JUN-01	M
S041	CHARMI BABARIYA	D021	05-JUL-00	F
S042	ADITYA PATEL	D021	14-JUL-00	M

Download CSV
10 rows selected.

Professor:

```
INSERT INTO PROFESSOR VALUES
('P01','MMG SIR','9913010327','D01');
INSERT INTO PROFESSOR VALUES
('P02','ANV MAM','9010919327','D01');
INSERT INTO PROFESSOR VALUES
('P011','HBP SIR','8739190270','D02');
INSERT INTO PROFESSOR VALUES
('P021','SRS SIR','9210909327','D011');
INSERT INTO PROFESSOR VALUES
('P031','VBC MAM','9913912349','D012');
INSERT INTO PROFESSOR VALUES
('P041','SKV SIR','9912123311','D021');
INSERT INTO PROFESSOR VALUES
('P042','VRP SIR','8899778850','D021');
INSERT INTO PROFESSOR VALUES
('P043','BDR MAM','9000050000','D021');
SELECT * FROM PROFESSOR;
```

PID	PNAME	P_CONT	DEID
P01	MMG SIR	9913010327	D01
P02	ANV MAM	9010919327	D01
P011	HBP SIR	8739190270	D02
P021	SRS SIR	9210909327	D011
P031	VBC MAM	9913912349	D012
P041	SKV SIR	9912123311	D021
P042	VRP SIR	8899778850	D021
P043	BDR MAM	9000050000	D021

Download CSV
8 rows selected.

Attendance:

```
INSERT INTO ATTENDANCE VALUES
('S01','99','100','98');
INSERT INTO ATTENDANCE VALUES
('S02','95','95','100');
INSERT INTO ATTENDANCE VALUES
('S011','85','80','90');
INSERT INTO ATTENDANCE VALUES
('S012','90','92','88');
INSERT INTO ATTENDANCE VALUES
('S021','90','90','100');
INSERT INTO ATTENDANCE VALUES
('S022','80','84','76');
INSERT INTO ATTENDANCE VALUES
('S031','95','100','95');
SELECT * FROM ATTENDANCE;
```

SID	PERC	LECATT	LABATT
S01	99	100	98
S02	95	95	100
S011	85	80	90
S012	90	92	88
S021	90	90	100
S022	80	84	76
S031	95	100	95

Download CSV
7 rows selected.

Fees:

```
INSERT INTO FEES VALUES
('F01','S01','10000','10000','HDFC');
INSERT INTO FEES VALUES
('F02','S02','20000','20000','HDFC');
INSERT INTO FEES VALUES
('F03','S021','10000','0','HDFC SFI');
INSERT INTO FEES VALUES
('F04','S022','5000','5000','DENA BANK');
INSERT INTO FEES VALUES
('F05','S031','10000','0','BOB');
INSERT INTO FEES VALUES
('F06','S01','8000','8000','SBI');
INSERT INTO FEES VALUES
('F07','S041','10000','10000','BOB');
SELECT * FROM FEES;
```

FID	SID	AMOUNT	AMOUNT_PAID	BANKNAME
F01	S01	10000	10000	HDFC
F02	S02	20000	20000	HDFC
F03	S021	10000	0	HDFC SFI
F04	S022	5000	5000	DENA BANK
F05	S031	10000	0	BOB
F06	S01	8000	8000	SBI
F07	S041	10000	10000	BOB

Download CSV

7 rows selected.

Subject:

```
INSERT INTO SUBJECT VALUES
('SB01','DBMS');
INSERT INTO SUBJECT VALUES
('SB02','DAA');
INSERT INTO SUBJECT VALUES
('SB03','CJT');
INSERT INTO SUBJECT VALUES
('SB11','TAFL');
INSERT INTO SUBJECT VALUES
('SB12','MATHS1');
INSERT INTO SUBJECT VALUES
('SB14','MATHS3');
INSERT INTO SUBJECT VALUES
('SB05','FMA');
INSERT INTO SUBJECT VALUES
('SB21','PC');
INSERT INTO SUBJECT VALUES
('SB22','FOODRES');
SELECT * FROM SUBJECT;
```

SUBID	SUBNAME
SB01	DBMS
SB02	DAA
SB03	CJT
SB11	TAFL
SB12	MATHS1
SB14	MATHS3
SB05	FMA
SB21	PC
SB22	FOODRES

Download CSV

Exam:

INSERT INTO EXAM VALUES

('E01','1','100','SB01','D01');

INSERT INTO EXAM VALUES

('E02','4','36','SB22','D011');

INSERT INTO EXAM VALUES

('E03','3','100','SB21','D011');

INSERT INTO EXAM VALUES

('E04','1','100','SB11','D01');

INSERT INTO EXAM VALUES

('E05','4','100','SB21','D021');

INSERT INTO EXAM VALUES

('E06','2','30','SB21','D021');

INSERT INTO EXAM VALUES

('E07','1','100','SB22','D012');

INSERT INTO EXAM VALUES

('E08','2','100','SB11','D011');

INSERT INTO EXAM VALUES

('E09','7','36','SB14','D01');

INSERT INTO EXAM VALUES

('E10','6','100','SB12','D01');

SELECT * FROM EXAM;

EXAMID	SEM	MARKS	SUBID	DEID
E01	1	100	SB01	D01
E02	4	36	SB22	D011
E03	3	100	SB21	D011
E04	1	100	SB11	D01
E05	4	100	SB21	D021
E06	2	30	SB21	D021
E07	1	100	SB22	D012
E08	2	100	SB11	D011
E09	7	36	SB14	D01
E10	6	100	SB12	D01

Download CSV

10 rows selected.

Supervisor:

INSERT INTO SUPERVISOR VALUES

('P01','E01','505','06:30:00');

INSERT INTO SUPERVISOR VALUES

('P011','E01','503','06:30:00');

INSERT INTO SUPERVISOR VALUES

('P011','E07','008','14:30:00');

INSERT INTO SUPERVISOR VALUES

('P031','E08','302','16:30:00');

INSERT INTO SUPERVISOR VALUES

('P041','E09','305','12:00:00');

SELECT * FROM SUPERVISOR;

PID	EXAMID	ROOMNO	ALL_TIME
P01	E01	505	06:30:00
P011	E01	503	06:30:00
P011	E07	8	14:30:00
P031	E08	302	16:30:00
P041	E09	305	12:00:00

Download CSV

5 rows selected.

Enrolled_std:

INSERT INTO ENROLLED_STD VALUES

('S01','E01','SB01','505');

INSERT INTO ENROLLED_STD VALUES

('S01','E02','SB03','505');

INSERT INTO ENROLLED_STD VALUES

('S01','E05','SB05','503');

INSERT INTO ENROLLED_STD VALUES

('S011','E06','SB11','008');

INSERT INTO ENROLLED_STD VALUES

('S011','E06','SB14','302');

INSERT INTO ENROLLED_STD VALUES

('S012','E07','SB03','302');

INSERT INTO ENROLLED_STD VALUES

('S021','E04','SB12','302');

INSERT INTO ENROLLED_STD VALUES

('S021','E01','SB12','505');

INSERT INTO ENROLLED_STD VALUES

('S022','E01','SB11','305');

INSERT INTO ENROLLED_STD VALUES

('S02','E02','SB03','305');

SELECT * FROM ENROLLED_STD;

SID	EXAMID	SUBID	ROOMNO
S01	E01	SB01	505
S01	E02	SB03	505
S01	E05	SB05	503
S011	E06	SB11	8
S011	E06	SB14	302
S012	E07	SB03	302
S021	E04	SB12	302
S021	E01	SB12	505
S022	E01	SB11	305
S02	E02	SB03	305

Download CSV
10 rows selected.

Routine:

INSERT INTO ROUTINE VALUES

('E01','12-AUG-2020','06:30:00');

INSERT INTO ROUTINE VALUES

('E02','14-AUG-2020','11:30:00');

INSERT INTO ROUTINE VALUES

('E07','16-AUG-2020','14:30:00');

INSERT INTO ROUTINE VALUES

('E08','17-AUG-2020','16:30:00');

INSERT INTO ROUTINE VALUES

('E09','30-DEC-2020','12:00:00');

INSERT INTO ROUTINE VALUES

('E03','19-DEC-2020','11:30:00');

SELECT * FROM ROUTINE;

EXAMID	DT	TIME
E01	12-AUG-20	06:30:00
E02	14-AUG-20	11:30:00
E07	16-AUG-20	14:30:00
E08	17-AUG-20	16:30:00
E09	18-AUG-20	12:00:00
E03	19-AUG-20	11:30:00

Download CSV

6 rows selected.

Result:

INSERT INTO RESULT VALUES

('E01','S01','9.10','8.70');

INSERT INTO RESULT VALUES

('E01','S02','8.67','8.90');

INSERT INTO RESULT VALUES

('E02','S021','7.1','8.0');

INSERT INTO RESULT VALUES

('E03','S022','9.1','8.69');

INSERT INTO RESULT VALUES

('E03','S031','7.66','8.10');

INSERT INTO RESULT VALUES

('E04','S032','9.8','8.7');

INSERT INTO RESULT VALUES

('E01','S011','9.6','9.8');

INSERT INTO RESULT VALUES

('E01','S012','6.65','7.1');

SELECT * FROM RESULT;

EXAMID	SID	SPI	CPI
E02	S021	7.1	8
E03	S022	9.1	8.69
E03	S031	7.66	8.1
E04	S032	9.8	8.7
E01	S011	9.6	9.8
E01	S012	6.65	7.1
E01	S01	9.1	8.7
E01	S02	8.67	8.9

Download CSV

8 rows selected

8.3 QUERIES

1. **Find the id of students who has not paid full fees.**

   Select sid from fees
   where amount_paid <amount;

SID
S021
S031

Download CSV
2 rows selected.

2. **Find sid whose labattendance is less then 80(%).**
   Select sid from attendance
   where Labatt < 80;

SID
S022

Download CSV

3. **Count the number of students paying fees in various banks.**

   Select bankname,count(sid) from fees
   group by bankname;

BANKNAME	COUNT(SID)
HDFC	2
HDFC SFI	1
SBI	1
BOB	2
DENA BANK	1

Download CSV
5 rows selected.

4. **Find exam examid and sem in which total marks is 100.**

```
Select examid,sem from exam
 where marks = 100;
```

EXAMID	SEM
E01	1
E03	3
E04	1
E05	4
E07	1
E08	2
E10	6

Download CSV
7 rows selected.

5. **Find examid and sid whose spi inbetween 7 to 9.**

```
Select examid,sid from result
 where spi between 7 and 9;
```

EXAMID	SID
E01	S02
E02	S021
E03	S031

Download CSV
3 rows selected.

JOIN

6. **Find student name whose cpi>8.5.**

```
Select student.sid,student.sname from student
inner join result on result.sid=student.sid
where cpi>8.5;
```

SID	SNAME
S01	DARSHAK KATHIRIYA
S02	NIHAL LIMBANI
S011	AYUSHI AJUDIYA
S022	AARYA MAHETA
S032	ANJAN VASOYA

Download CSV

5 rows selected.

7. Show the student name whose attendance how much.

Select student.sname,attendance.perc from student

inner join attendance

on student.sid=attendance.sid;

SNAME	PERC
DARSHAK KATHIRIYA	99
NIHAL LIMBANI	95
AYUSHI AJUDIYA	85
VARJESH VADI	90
SRUSHTI PATEL	90
AARYA MAHETA	80
DARSHAK KATHIRIYA	95

Download CSV

7 rows selected.

8. **display SID and SNAME who pays fees in HDFC bank.**

```
Select student.sid,student.sname from student
inner join fees on student.sid=fees.sid
where bankname= 'HDFC';
```

SID	SNAME
S01	DARSHAK KATHIRIYA
S02	NIHAL LIMBANI

Download CSV

2 rows selected.

9. **Find the professors name,pid who allocated supervision time in between 6 to 12.**

```
Select professor.pname,professor.pid from professor
inner join supervisor on professor.pid=supervisor.pid
where all_time between '06:00:00' and '12:00:00';
```

PNAME	PID
MMG SIR	P01
HBP SIR	P011
SKV SIR	P041

Download CSV

3 rows selected.

10. Show the subject details of sem 1 examination.

```
Select subject.* from subject
inner join exam on subject.subid=exam.subid
where sem='1';
```

SUBID	SUBNAME
SB01	DBMS
SB11	TAFL
SB22	FOODRES

Download CSV

3 rows selected.

SUBQUERY

11. Show the names of students having lecture attendance more then 90(%).

```
select sname from student
where sid in(select sid from attendance where lecatt>=90);
```

SNAME
DARSHAK KATHIRIYA
NIHAL LIMBANI
VARJESH VADI
SRUSHTI PATEL
DARSHAK KATHIRIYA

Download CSV

5 rows selected.

8.4 Implement functions, triggers, cursors, stored procedures:

Procedure:

Q) Create procedure user enter the course id and get output the course year.

```
Set serveroutput on;
DECLARE
   cid1 varchar2(3);
   year1 number(2);
   PROCEDURE Get_year
   IS
   BEGIN
      cid1 := '&cid1';
      Select cyear into year1 from courses where cid=cid1;
      dbms_output.put_line( ' Total year is ' || year1 );
   END Get_year;
BEGIN
   Get_year();
END;
/
```

```
SQL> DECLARE
  2       cid1 varchar2(3);
  3       year1 number(2);
  4       PROCEDURE Get_year
  5       IS
  6       BEGIN
  7           cid1 := '&cid1';
  8           Select cyear into year1 from courses where cid=cid1;
  9           dbms_output.put_line( ' Total year is ' || year1 );
 10       END Get_year;
 11   BEGIN
 12       Get_year();
 13   END;
 14   /
Enter value for cid1: C04
old   7:          cid1 := '&cid1';
new   7:          cid1 := 'C04';
Total year is 3

PL/SQL procedure successfully completed.
```

Function:

Q) Count student which result cpi gretterthan and eql to 8.

```
Create or replace function totalstudent
return number IS
 total number(2):= 0;
BEGIN
   Select count(sid) into Total from result where cpi>='8';
   return total;
END;
/
DECLARE
  c number(2);
BEGIN
  c :=totalstudent ;
  dbms_output.put_line('Total student is: ' || c);
END;
/
```

```
SQL> SET SERVEROUTPUT ON;
SQL> Create or replace function totalstudent
  2  return number IS
  3   total number(2):= 0;
  4  BEGIN
  5      Select count(sid) into Total from result where cpi>='8';
  6      return total;
  7  END;
  8  /

Function created.

SQL> DECLARE
  2     c number(2);
  3  BEGIN
  4     c :=totalstudent ;
  5     dbms_output.put_line('Total student is: ' || c);
  6  END;
  7  /
Total student is: 7

PL/SQL procedure successfully completed.
```

Triggers:

Q1) Remove the entries from student table will delete the record of that student in every other table.

Create or replace TRIGGER ELIMINATE after delete on student for each row

DECLARE

BEGIN

DELETE from fees where :OLD.sid = sid;

DELETE from attendance where :OLD.sid = sid;

DELETE from Enrolled_std where :OLD.sid = sid;

DELETE from result where :OLD.sid= sid;

END;

/

Before delete (all Data)

```
SQL> SELECT * FROM STUDENT;

SID        SNAME                     DEID DOB       G
---------- ------------------------- ---- --------- -
S01        DARSHAK KATHIRIYA         D01  13-JUL-01 M
S02        NIHAL LIMBANI             D01  05-AUG-00 M
S011       AYUSHI AJUDIYA            D02  10-JUN-00 F
S012       VARJESH VADI              D02  13-JUL-00 M
S021       SRUSHTI PATEL             D011 20-JUN-02 F
S022       AARYA MAHETA              D011 30-JAN-02 M
S031       DARSHAK KATHIRIYA         D012 01-FEB-01 M
S032       ANJAN VASOYA              D012 15-JUN-01 M
S041       CHARMI BABARIYA           D021 05-JUL-00 F
S042       ADITYA PATEL              D021 14-JUL-00 M

10 rows selected.
```

```
SQL> SELECT * FROM FEES;

FID        SID            AMOUNT AMOUNT_PAID BANKNAME
---------- ---------- ---------- ----------- ------------------------
F01        S01             10000       10000 HDFC
F02        S02             20000       20000 HDFC
F03        S021            10000           0 HDFC SFI
F04        S022             5000        5000 DENA BANK
F05        S031            10000           0 BOB
F06        S01              8000        8000 SBI
F07        S041            10000       10000 BOB

7 rows selected.
```

```
SQL> SELECT * FROM ATTENDANCE;

SID              PERC     LECATT     LABATT
---------- ---------- ---------- ----------
S01                99        100         98
S02                95         95        100
S011               85         80         90
S012               90         92         88
S021               90         90        100
S022               80         84         76
S031               95        100         95

7 rows selected.
```

```
SQL> SELECT * FROM ENROLLED_STD;

SID        EXAMID     SUBI     ROOMNO
---------- ---------- ---- ----------
S01        E01        SB01        505
S01        E02        SB03        505
S01        E05        SB05        503
S011       E06        SB11          8
S011       E06        SB14        302
S012       E07        SB03        302
S021       E04        SB12        302
S021       E01        SB12        505
S022       E01        SB11        305
S02        E02        SB03        305

10 rows selected.
```

```
SQL> SELECT * FROM RESULT;

EXAMID     SID                SPI        CPI
---------- ---------- ---------- ----------
E01        S01                9.1        8.7
E01        S02               8.67        8.9
E02        S021               7.1          8
E03        S022               9.1       8.69
E03        S031              7.66        8.1
E04        S032               9.8        8.7
E01        S011               9.6        9.8
E01        S012              6.65        7.1

8 rows selected.
```

delete from student where sid='S01';

(after delete trigger run)

```
SQL> select * from student;

SID        SNAME                     DEID DOB       G
---------- ------------------------- ---- --------- -
S02        NIHAL LIMBANI             D01  05-AUG-00 M
S011       AYUSHI AJUDIYA            D02  10-JUN-00 F
S012       VARJESH VADI              D02  13-JUL-00 M
S021       SRUSHTI PATEL             D011 20-JUN-02 F
S022       AARYA MAHETA              D011 30-JAN-02 M
S031       DARSHAK KATHIRIYA         D012 01-FEB-01 M
S032       ANJAN VASOYA              D012 15-JUN-01 M
S041       CHARMI BABARIYA           D021 05-JUL-00 F
S042       ADITYA PATEL              D021 14-JUL-00 M

9 rows selected.
```

```
SQL> select * from fees;

FID        SID            AMOUNT AMOUNT_PAID BANKNAME
---------- ---------- ---------- ----------- -------------------------
F02        S02             20000       20000 HDFC
F03        S021            10000           0 HDFC SFI
F04        S022             5000        5000 DENA BANK
F05        S031            10000           0 BOB
F07        S041            10000       10000 BOB
```

```
SQL> select * from attendance;

SID              PERC     LECATT     LABATT
---------- ---------- ---------- ----------
S02                95         95        100
S011               85         80         90
S012               90         92         88
S021               90         90        100
S022               80         84         76
S031               95        100         95

6 rows selected.
```

```
SQL> select * from enrolled_std;

SID        EXAMID     SUBI      ROOMNO
---------- ---------- ---- ----------
S011       E06        SB11           8
S011       E06        SB14         302
S012       E07        SB03         302
S021       E04        SB12         302
S021       E01        SB12         505
S022       E01        SB11         305
S02        E02        SB03         305
```

```
SQL> select * from result;

EXAMID     SID              SPI        CPI
---------- ---------- ---------- ----------
E01        S02             8.67        8.9
E02        S021             7.1          8
E03        S022             9.1       8.69
E03        S031            7.66        8.1
E04        S032             9.8        8.7
E01        S011             9.6        9.8
E01        S012            6.65        7.1

7 rows selected.
```

Q2) Deleting an examid in routine table gives an error if the exam is still not todays current system .

Create or replace TRIGGER restrict_exam_delete before delete on routine for each row

```
DECLARE
   my_exception EXCEPTION;
BEGIN
   IF :OLD.DT > TRUNC(SYSDATE) THEN
      RAISE my_exception;
   END IF;
EXCEPTION
   WHEN my_exception THEN
      raise_application_error(-20001,'This exam has not started yet !');
END;
/
```

```
SQL> select * from routine;

EXAMID     DT        TIME
---------- --------- --------
E01        12-AUG-20 06:30:00
E02        14-AUG-20 11:30:00
E07        16-AUG-20 14:30:00
E08        17-AUG-20 16:30:00
E09        30-DEC-20 12:00:00
E03        19-DEC-20 11:30:00

6 rows selected.
```

Delete from routine where examid=’E03’;

```
SQL> Delete from routine where examid='E03';
Delete from routine where examid='E03'
            *
ERROR at line 1:
ORA-20001: This exam has not started yet !
ORA-06512: at "SCOTT.RESTRICT_EXAM_DELETE", line 9
ORA-04088: error during execution of trigger 'SCOTT.RESTRICT_EXAM_DELETE'
```

Cursor

Q)create cursor to display the student attendance in decs order.

Create table temp(tsid varchar2(10),tperc number);

```
DECLARE
  CURSOR crs is
    SELECT sid,perc FROM attendance
      ORDER BY perc DESC;
  TSID VARCHAR2(10);
  Tperc NUMBER;
BEGIN
  OPEN crs;
  FOR i IN 1..5 LOOP
    FETCH crs INTO TSID,TPERC;
    EXIT WHEN crs%NOTFOUND;
    INSERT INTO temp VALUES (TSID,TPERC);
    COMMIT;
  END LOOP;
  CLOSE crs;
END;
/
```

before

```
SQL> select * from attendance;

SID              PERC     LECATT     LABATT
---------- ---------- ---------- ----------
S02                95         95        100
S011               85         80         90
S012               90         92         88
S021               90         90        100
S022               80         84         76
S031               95        100         95

6 rows selected.
```

After

```
SQL> select * from temp;

TSID             TPERC
---------- ----------
S02                95
S031               95
S012               90
S021               90
S011               85
```

6.FUTURE ENHANCEMENTS OF THE SYSTEM

We can make more advanced software for university examination management.

1. We also add online examination management.
   (We mention at requirement analysis like professor module some function add).
2. SMS system should be added as the result of student arrives.
3. Backup mechanism can be also implemented.

7. BIBLIOGRAPHY

- Implementation of this project we referred to many websites and book.
- We created the ER Diagram on “https://erdplus.com/” and Schema Diagram on “https://creately.com/”.
- We use Oracle Live SQL and Oracle 10g to perform all datatable.

Book:
Database System Concepts
By: Henry K Forth and A. Silberchatz

Reference online:
https://www.w3schools.com/sql/
https://www.mysqltutorial.org/
https://www.geeksforgeeks.org/sql-tutorial/