Problem Statement:

Efficient Database Management System for conduction of Examinations at school/college level.

PROJECT DESCRIPTION:

Examination database Management system is a relational data base which stores the basic information of student, Registration records,The basic examination details,student results and handles the basic relations which this holds between them.

This database is ER Modeled.

It helps to analyse and record student basic data and there results quite easily.

It's resemblance is to that our Institute examination Department database.

ENTITIES:

Student:

- The basic information of the student is stored so that to identify him/her and there are no redundance in registration of the student this entity is defined by attributes Name,contact,branch,year. This are the basic attributes which define a particular student in the database.

Examination:

- Now once the student is registered it gives the clear information about the subjects in which the student is enrolled.
- Now exam id is given to student, and seating arrangement is done accordingly.
- As per the subject id provided by his registration entity we can make seating arrangements.
- Department which is obtained from the student data eases the examination management.
- Exam id is candidate key here and it solely defines to which exams is the student applicable to write and what are it's requirements and credits.

Subject:

- Subject is the basic entity which gives the number of credits to the course and also the no of students enrolled to that particular course.
- Subject entity also helps in choosing out the invigilator.

Schedule:

- This is the basic work done by the examination Department schedule in the daily routine of a class in the particular year and yet more importantly the examination scheduling.
- The registration relation gives the number of students registered to the course and hence deciding the number of rooms required and also that there is no clash in subjects reserved by the student
- Exam id determines the schedule of a student for the particular season of examination.

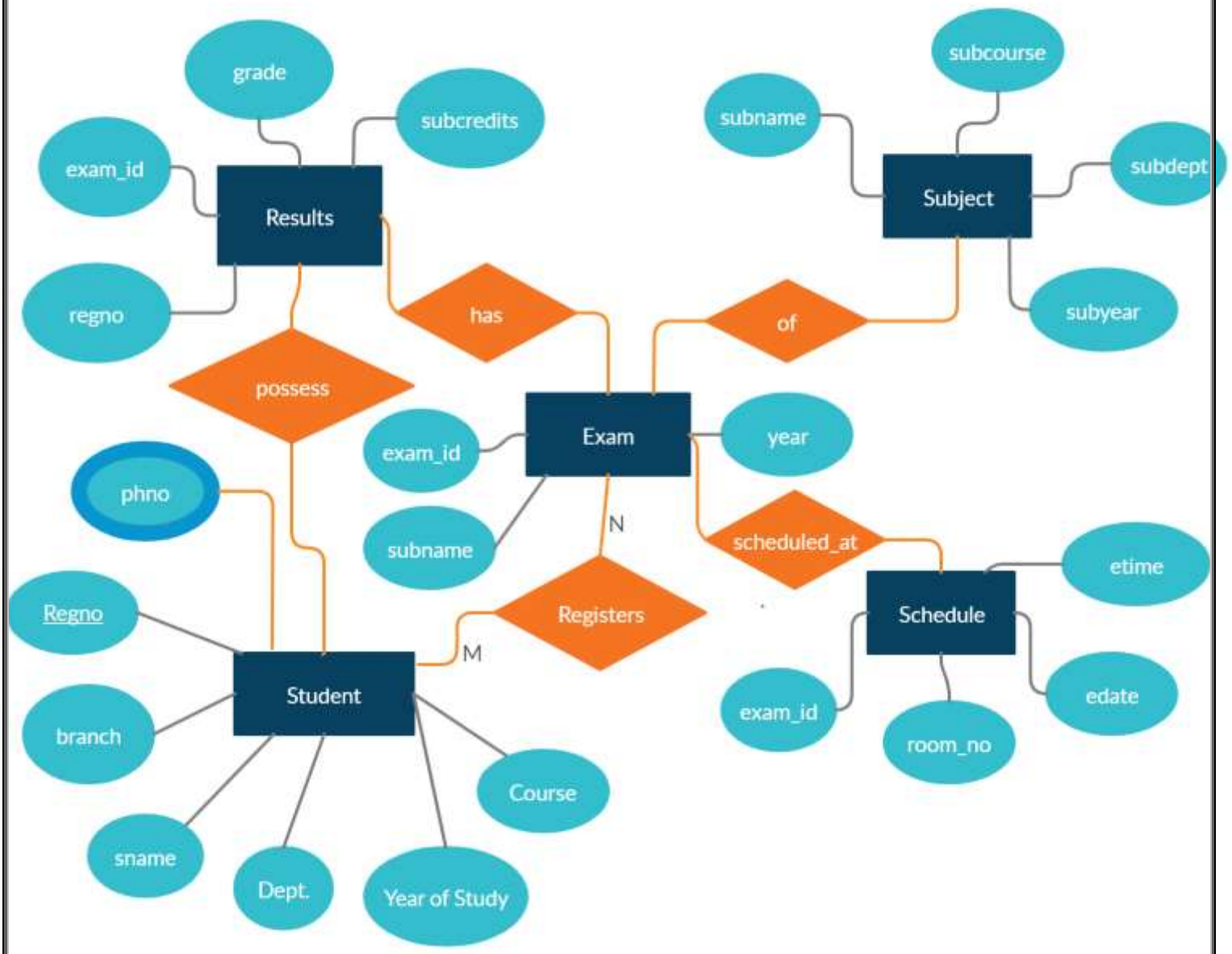
Result:

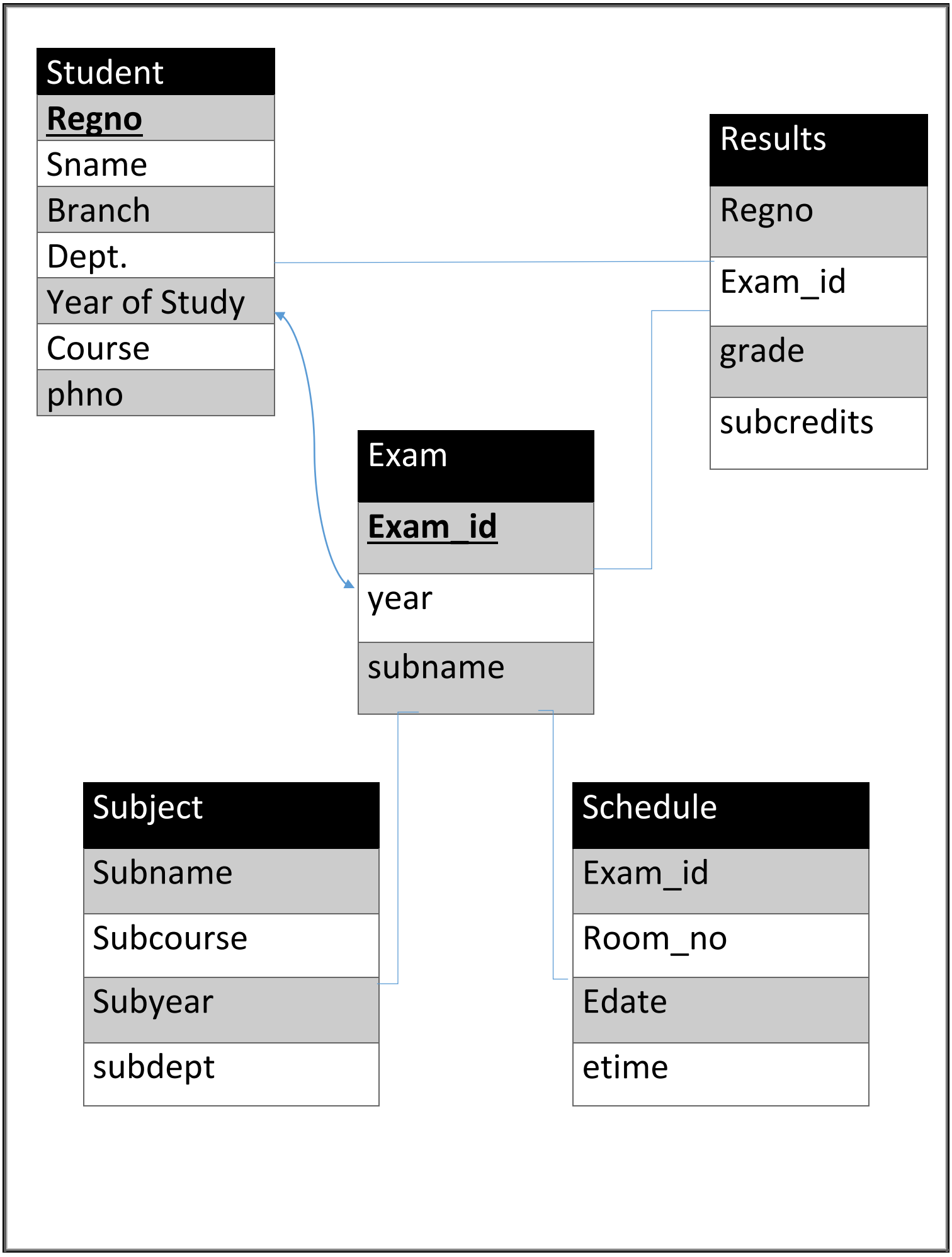
- Result is yet another important task performed by examination Department.
- Student_ID, exam_ID collectively gives the data of the students who attended the examination and also the marks scored by the person in the examination
- Grade is the entity which describes the grade obtain by student in the particular subject by relation with subject id.

THE RELATIONSHIPS

1. This is the 1-to-many binary relationship set between subject and exam entity sets
2. Possess is the one-to-one binary relationship sets between Exam and Results.
3. Registers is a many-to-many binary relationship sets between the Exam Candidate and the Examination.
4. Scheduled_at is the relationship that holds between the examination and the exam schedule.
5. The of relation is a one-to-one binary relationship set between the subject and the examination.

ER Diagram


grade
subcredits
exam_id
Results
regno
has
possess
subcourse
subname
Subject
subdept
subyear
of
Exam
exam_id
year
subname
N
scheduled_at
phno
Regno
Registers
M
Schedule
etime
edate
exam_id
room_no
Student
branch
Course
sname
Dept.
Year of Study


Student
Regno
Sname
Branch
Dept.
Year of Study
Course
phno
Results
Regno
Exam_id
grade
subcredits
Exam
Exam_id
year
subname
Subject
Subname
Subcourse
Subyear
subdept
Schedule
Exam_id
Room_no
Edate
etime

Student table:

Attributes:

regno sname contactno sec branch year course noofcredits

Functional Dependencie's:

regno→name

regno→contact

regno→sec

regno→branch

regno→year

regno→course

regno→noofcredits

(branch,year,course)→noofcredits

Transitive relation{

Regno→(branch,year,course)

(branch,year,course)→noofcredits

(branch,year,course) is a non key attribute

➔not in 3nf;

multivalued attribute:contactno

➔not in 1nf;

Therefore decompose

Student1

Attributes: regno name sec branch year course

Pk=regno

Fd's:

Regno→name

Regno→sec

Regno→branch

Regno→year

Regno→course

In 1f (no multi valued attributes)

In 2nf(no partial dependency)

In 3 nf(no transitive relation)

Student2

Attributes: regno contactno

Pk(regno,contactno)

➔it is in 3nf;

Student3

Attributes: branch year course noofcredits

Pk=(branch year course)

Fd’s:

(branch year course)→noofcredits

In 1nf(no multival attribute)

In 2nf(no partial dependency)

In 3nf(no transitive relation)

Subject table:

Attributes: subname subyear subdept subcourse

Pk=(subyear,subname)

Fd’s:

(subyear,subname)→dept

(subyear,subname)→course

In 1nf(no multivaluedattributes)

In 2nf(no partial dependency)

In 3nf(no transitive dependency)

Exam table:

Attribute: exam_id ,subname, subyear

Pk=examid

Fd’s:

Examid→subname

Examid→subyear

In 1nf(no multivaluedattributes)

In 2nf(no partial dependency)

In 3nf(no transitive dependency)

Registers table:

Attributes:regno,exam_id

Pk=(regno,examid)

In 1nf(no multivaluedattributes)

In 2nf(no partial dependency)

In 3nf(no transitive dependency)

Result table

Initially attributes: regno, examid ,grade, submax_credits

Primarykey(regno,examid)

Fd's:

(Regno,examid)→grade

(Regno,examid)→ submax_credits

Examid→ submax_credits

Partial relation:{

(Regno,examid)→ submax_credits

examid→ submax_credits

}

Therefore not in 2nf

Decompose..

Result1:

Attributes:regno examid grade

Pk=(regno ,examid)

Fd's:

(regno ,examid)→grade

In 1nf(no multivaluedattributes)

In 2nf(no partial dependency)

In 3nf(no transitive dependency)

Result2:

Attributes: examid,maxsub_credits

Pk(examid)

Fd's:

examid→maxsub_credits

In 1nf(no multivaluedattributes)

In 2nf(no partial dependency)

In 3nf(no transitive dependency)

Schedule table:

Attributes: examid, room_no,examtime,examdate

Pk(examid, room_no,examtime,examdate);

In 1nf(no multivaluedattributes)

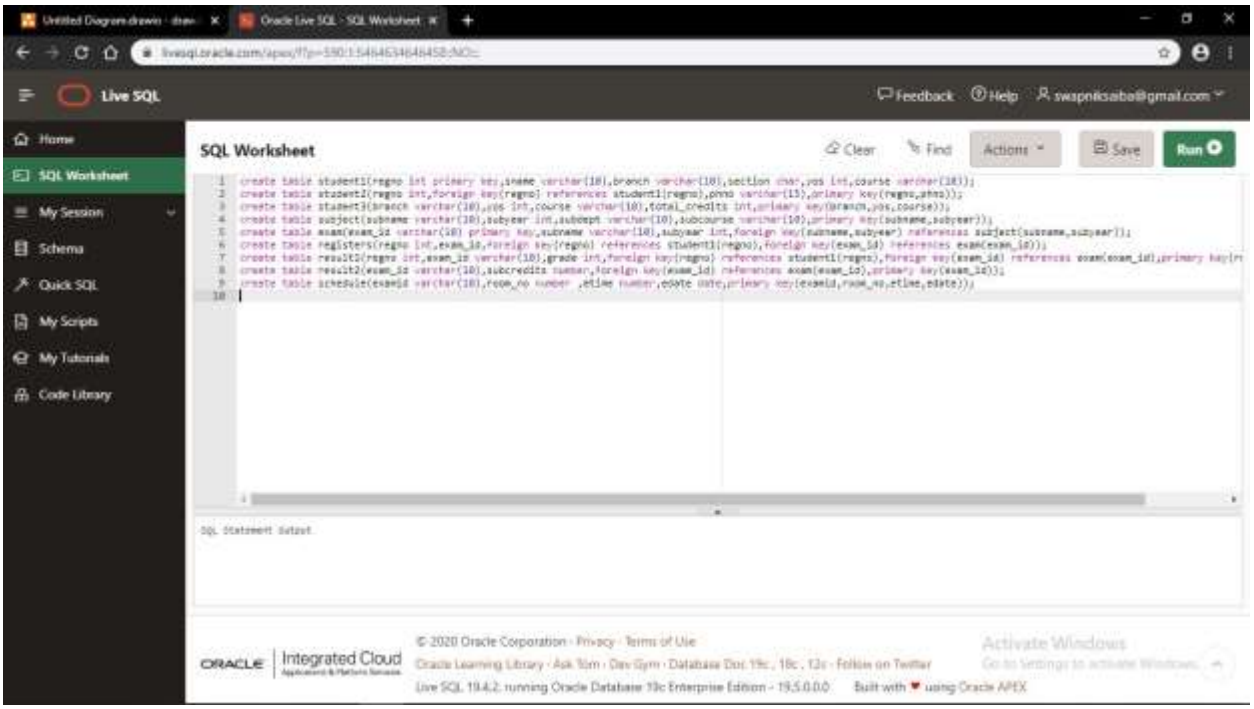
In 2nf(no partial dependency)

In 3nf(no transitive dependency)

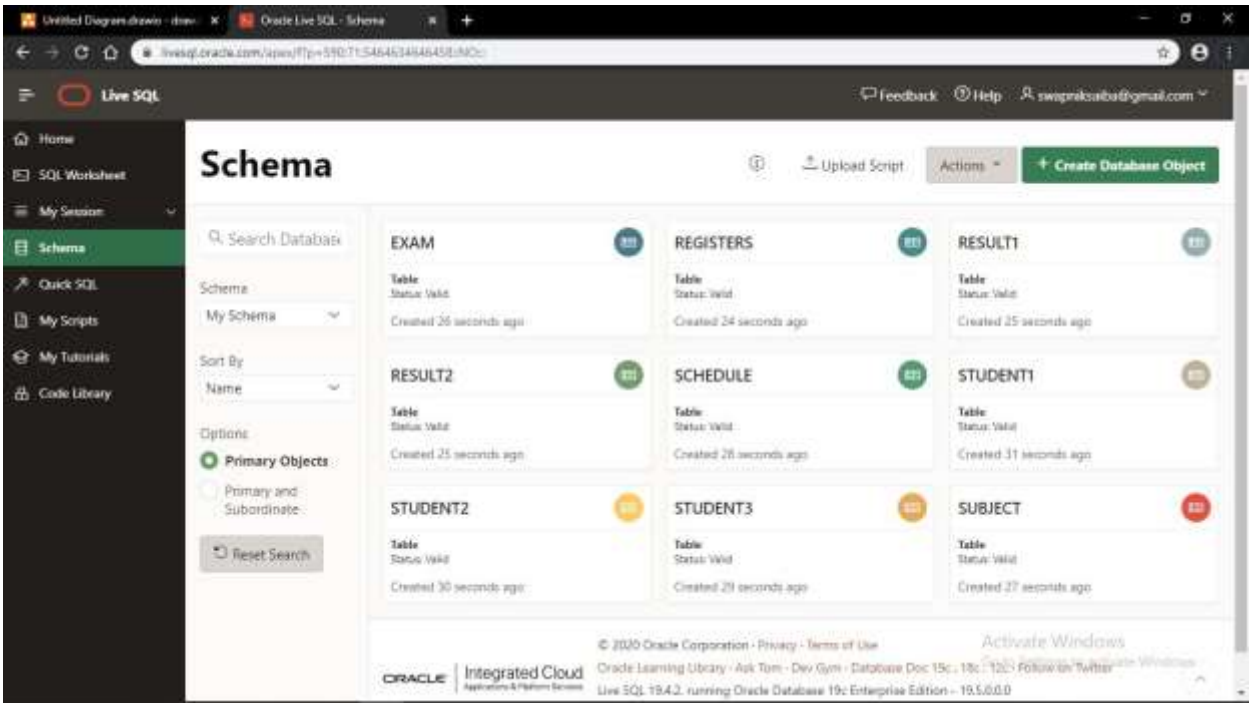
After Normalisation:

ENTITY	ATTRIBUTE
STUDENT1	pk(regno),sname,branch,section,yos,course
STUDENT1	Fk(regno),phno,pk(regno,phno)
STUDENT3	branch,yos,course,totalcredits,pk(branch,yos,course)
SUBJECT	Subname,subyear,subdept,subcourse,pk(subname,subyear)
EXAM	Pk(exam_id),fk(subname,subyear)
REGISTERS	Fk(Regno),fk(exam_id)
RESULT1	Fk(regno),fk(exam_id),pk(regno,examid),grade
RESULT2	Fk(exam_id),subcredits,pk(exam_id)
SCHEDULE	Fk(Exam_id),room_no,etime,edate,pk(examid,room_no,etime,edate)

SQL Statements to create tables:



Created tables in Database:



SQL Statements used for creation of Database:

Create and insert statements:

Student1 Table:

create table student1(regno int primary key,sname varchar(10),branch varchar(10),section char,yos int,course varchar(10));

insert statements:

insert into student1 values(1,'ainy','cse','b',2,'btech');

insert into student1 values(2,'swapna','mechanical','a',3,'btech');

insert into student1 values(3,'raka','mechanical','a',2,'btech');

insert into student1 values(4,'madhu','chemical','b',4,'btech');

insert into student1 values(5,'sanyu','chemical','a',3,'btech');

insert into student1 values(6,'virat','cse','a',2,'btech');

insert into student1 values(7,'sachin','eee','b',3,'btech');

insert into student1 values(8,'dhoni','eee','a',2,'btech');

insert into student1 values(9,'reddy','cse','a',1,'mtech');

insert into student1 values(10,'chidha','eee','a',1,'mtech');

Student2 Table:

create table student2(regno int,foreign key(regno) references student1(regno),phno varchar(15),primary key(regno,phno));

values:

insert into student2 values(1,'98765438262');

```
insert into student2 values(1,'97765438262');
insert into student2 values(3,'95565438262');
insert into student2 values(4,'93245438262');
insert into student2 values(5,'94675438262');
insert into student2 values(6,'98745438262');
insert into student2 values(7,'98762138262');
insert into student2 values(8,'98766738262');
insert into student2 values(9,'98235438262');
insert into student2 values(10,'99765438262');
```

Student3 Table:

```
create table student3(branch varchar(10),yos
int,course varchar(10),total_credits int,primary
key(branch,yos,course));
insert into student3 values('cse',1,'mtech',19);
insert into student3 values('mech',3,'btech',23);
insert into student3 values('mech',2,'btech',18);
insert into student3 values('chem',4,'btech',24);
insert into student3 values('chem',3,'btech',21);
insert into student3 values('EEE',3,'btech',20);
insert into student3 values('EEE',1,'mtech',20)
```

Subject Table:

create table subject(subname varchar(10),subyear int,subdept varchar(10),subcourse varchar(10),primary key(subname,subyear));

values:

insert into subject values('oop',2,'cse','btech');

insert into subject values('oop',1,'cse','mtech');

insert into subject values('dbms',2,'cse','btech');

insert into subject values('dbms',3,'cse','btech');

insert into subject values('fluidmech',4,'chem','btech');

insert into subject values('kinematics',3,'mech','btech');

insert into subject values('kinematics',2,'mech','btech');

insert into subject values('titration',3,'chem','btech');

insert into subject values('transform',3,'EEE','btech');

insert into subject values('em',1,'EEE','mtech');

Exam Table:

create table exam(exam_id varchar(10) primary key,subname varchar(10),subyear int,foreign key(subname,subyear) references subject(subname,subyear));

values:

insert into exam values('oop2','oop',2);

insert into exam values('oop1','oop',1);

insert into exam values('dbms2','dbms',2);

```
insert into exam values('dbms3','dbms',2);

insert into exam values('k3','kinematics',3);

insert into exam values('k2','kinematics',2);

insert into exam values('ti3','titration',3);

insert into exam values('tr3','transform',3);

insert into exam values('em1','em',1);

insert into exam values('fm4','fluidmech',4);
```

Registers Table:

```
create table registers(regno int,exam_id varchar(10),foreign key(regno)
references student1(regno),foreign key(exam_id) references
exam(exam_id));
```

values:

```
insert into registers values(1,'oop2');

insert into registers values(6,'oop1');

insert into registers values(6,'dbms2');

insert into registers values(5,'fm4');

insert into registers values(7,'tr3');

insert into registers values(2,'k3');
```

Result1 Table:

create table result1(regno int,exam_id varchar(10),grade int,foreign key(regno) references student1(regno),foreign key(exam_id) references exam(exam_id),primary key(regno,exam_id));

values:

insert into result1 values(1,'oop2',9);

insert into result1 values(6,'oop1',8);

insert into result1 values(6,'dbms2',7);

insert into result1 values(5,'fm4',9);

insert into result1 values(7,'tr3',9);

insert into result1 values(2,'k3',6);

Result2 Table:

create table result2(exam_id varchar(10),subcredits int,foreign key(exam_id) references exam(exam_id),primary key(exam_id));

values:

insert into result2 values('oop2',9);

insert into result2 values('oop1',8);

insert into result2 values('dbms2',7);

insert into result2 values('fm4',9);

insert into result2 values('tr3',9);

insert into result2 values('k3',6);

Schedule Table:

create table schedule(examid varchar(10),room_no int ,etime int,edate date,primary key(examid,room_no,etime,edate));

values:

insert into schedule values('oop2',102,1430,'19-nov-2020');

insert into schedule values('oop1',103,1000,'18-oct-2020');

insert into schedule values('dbms2',104,0930,'24-dec-2020');

insert into schedule values('fm4',102,1530,'25-dec-2020');

insert into schedule values('tr3',104,1330,'22-dec-2020');

insert into schedule values('k3',103,0830,'20-dec-2020');

insert into schedule values('oop2',105,0730,'13-dec-2020');

insert into schedule values('dbms2',106,1120,'09-dec-2020');