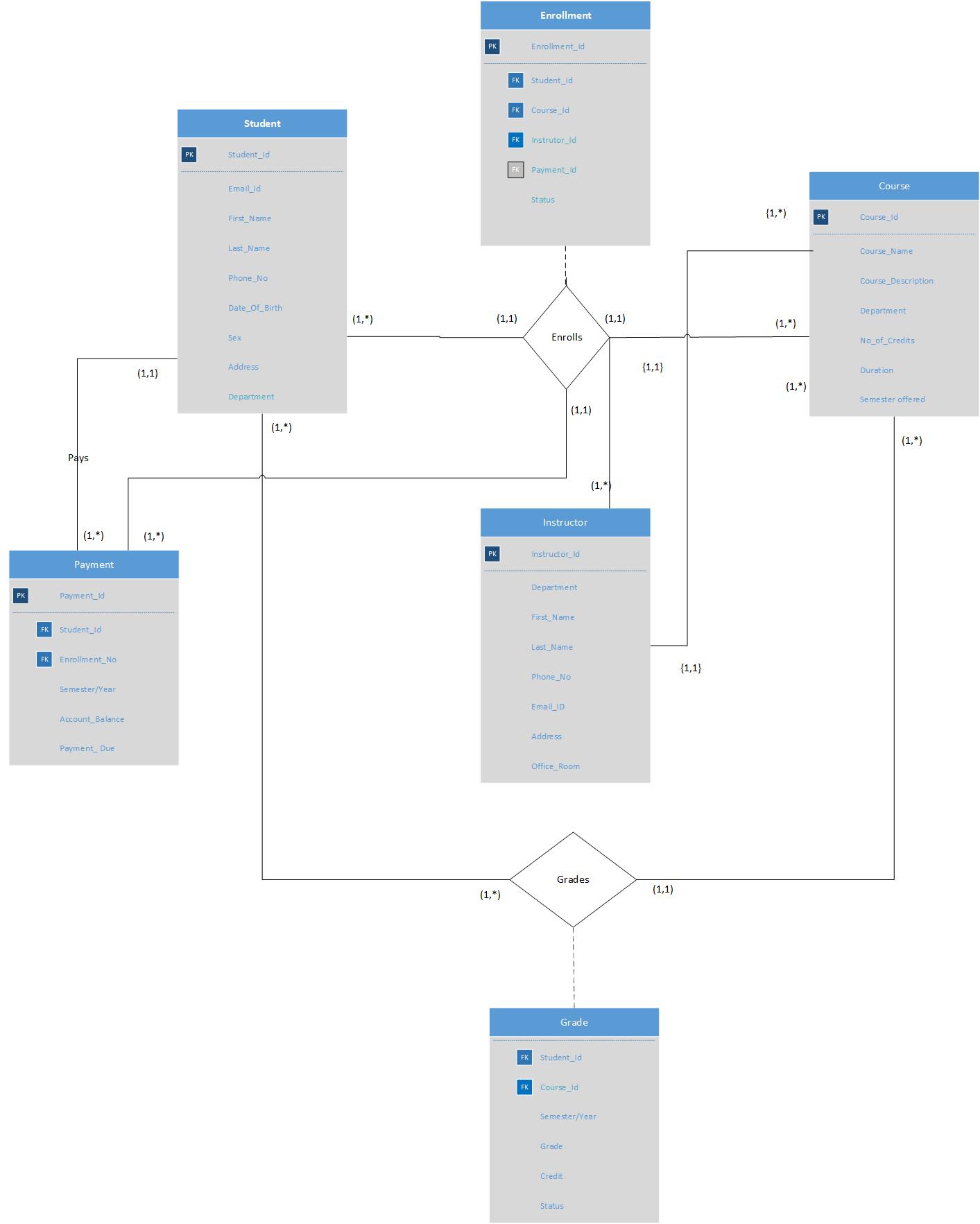
AIT 632 – Database Management Systems

D3 Physical Database Design

Team SSR

The Entity Relationship Diagram is as shown below:



In the Entity Relationship diagram, we have a many to many relationship between student and course. Hence there would be 3 tables and they are student, course and grade. The grade table would have two primary keys course\_ID, and Student\_ID. Student\_ID and course\_ID are foreign keys since they reference to the primary keys of other tables.

There exists a quaternary relationship between student, course, instructor and payment. The relationship enroll has a primary key Enrollment\_ID and Student\_ID, Course\_ID, Instructor\_ID Payment\_ID are foreign keys since they reference primary keys of other tables.

The relationship between student and payment is a one to many relationship. Hence the primary key from the one side (student) migrates as a foreign key to the many side (payment).

The relationship between instructor and enrollment is a one to many relationship. Hence the primary key from the one side the (instructor) migrates as a foreign key to the many side (enrollment).

The tables are as follows:

For the quaternary relationship between student, course and instructor.

Student (Student\_ID {PK}, Email\_ID, First\_name, Last\_name, Phone\_No, Address, Date of birth, Gender, Department)

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | DataType | constraint | null/notnull |
| Student\_ID | int | Primary key | Not null |
| Email\_ID | Varchar(30) |  | NotNull |
| First\_name | Varchar(20) |  | NotNull |
| Last\_name | Varchar(20) |  | NotNull |
| Phone\_No | Varchar(30) |  | NotNull |
| Address | Varchar(100) |  | NotNull |
| Date of birth | Date |  | NotNull |
| Gender | Varchar(6) |  | NotNull |

Course (Course\_ID {PK}, Course\_Name, No of credits, Duration, Semester offered, Class\_room, Department)

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | DataType | constraint | Null/not null |
| Course\_ID | int | Primary key | NotNull |
| Course\_Name | Varchar(20) |  | NotNull |
| No of credits | Number |  | NotNull |
| Duration | varchar(20) |  | NotNull |
| Semester offered | varchar(20) |  | NotNull |
| Class\_room | varchar(20) |  | NotNull |
| Department | varchar(30) |  | NotNull |

Instructor(Instructor\_ID{PK}, First\_name, Last\_name , Phone\_No, Email\_ID, Office\_room, Department)

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | DataType | constraint | Null/not null |
| Instructor\_ID | int | Primary key | Not null |
| First\_name | varchar(20) |  | Not null |
| Last\_name | varchar(20) |  | Not null |
| Phone\_No | varchar(20) |  | Not null |
| Email\_ID | varchar(20) |  | Not null |
| Office\_room | varchar(5) |  | Not null |
| Department | varchar(20) |  | Not null |

Payment (Payment\_ID{PK}, Student\_ID{FK}, Enrollment\_ID {FK}, Sem/Year Account balance, Payment\_due).

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | DataType | constraint | Null/not null |
| Payment\_ID | int | primary key | not null |
| Student\_ID | int | foreign key | not null |
| Enrollment\_ID | int | foreign key | not null |
| Account balance | numeric(12,2) |  | Not null |
| Payment\_due | numeric(12,2) |  | Not null |
| Sem\_year | Varchar(20) |  | Not null |

Enrollment (Enrollment{PK}, Student\_ID{FK}, Course\_ID{FK}, Payment\_ID{FK}, Instructor\_ID{FK}, Payment\_ID{FK}, status)

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | DataType | constraint | Null/not null |
| Enrollment\_ID | int | Primary key | not null |
| Student\_ID | int | Foreign key | not null |
| Course\_ID | int | Foreign key | not null |
| Payment\_ID | int | Foreign key | null |
| Instructor\_ID | int | Foreign key | not null |
| status | varchar(20) |  | not null |

For the grade table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | DataType | constraint | Null/not null |
| Student\_ID | int | Foreign key | not null |
| Sem/year | varchar(20) |  | not null |
| Course\_ID | Int | Foreign key | not null |
| Grade | varchar(2) |  | not null |
| Credit | int |  | not null |
| status | varchar(20) |  | not null |

The relational model is as shown below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Payment\_ID | Student\_ID | Enrollment\_ID | Semester/Year | Account balance | Payment\_due |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student\_ID | Email\_ID | First\_  name | Last\_  name | Phone\_  No | Address | Date Of  Birth | Gender | Department |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Enrollment\_ID | Student\_ID | Course\_ID | Instructor\_ID | Payment\_ID | Status |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instructor\_ID | Email\_ID | First\_  name | Last\_  name | Phone\_  No | Office\_room | Department |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Course\_ID | Course\_  Name | No of Credits | Duration | Semester  Offered | Class\_room | Department |

For the many to many relationship between student and course the tables are:

Student (Student\_ID {PK}, Email\_ID, First\_name, Last\_name, Phone\_No, Address, Date of birth, Gender, Department

Course (Course\_ID {PK}, Course\_Name, No of credits, Duration, Semester offered, Class\_room, Department)

Grade (Student\_ID{FK}, Course\_ID{FK},Sem/Year, grade, credit, status)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student\_ID | Email\_ID | First\_  name | Last\_  name | Phone\_  No | Address | Date Of  Birth | Gender | Department |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Course\_ID | Student\_ID | Semester/Year | Grade | Credit | Status |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Course\_ID | Course\_  Name | No of Credits | Duration | Semester  Offered | Class\_room | Department |

For the one to many relationship between student and payment, the student pays for the courses he has enrolled. Hence the enrollment has to be included as a table since payment cannot be found without enrollment ID.

Student (Student\_ID {PK}, Email\_ID, First\_name, Last\_name, Phone\_No, Address, Date of birth, Gender, Department )

Payment (Payment\_ID{PK}, Student\_ID{FK}, Enrollment\_ID {FK}, Account balance, Payment\_due)

Enrollment (Enrollment{PK}, Student\_ID{FK}, Course\_ID{FK},Instructor\_ID{FK}, status)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student\_ID | Email\_ID | First\_  name | Last\_  name | Phone\_  No | Address | Date Of  Birth | Gender | Department |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Payment\_ID | Student\_ID | Enrollment\_ID | Semester/Year | Account balance | Payment\_due |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Enrollment\_ID | Student\_ID | Course\_ID | Instructor\_ID | Payment\_ID | Status |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Course\_ID | Course\_  Name | No of Credits | Duration | Semester  Offered | Class\_room | Department |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instructor\_ID | Email\_ID | First\_  name | Last\_  name | Phone\_  No | Office\_room | Department |

The last two tables are present to show the referential integrity constraint.

**Business Rules:**

1. There exists a quaternary relationship between student, course, instructor and payment.
2. Many students enroll in many courses. Hence the relationship between student and courses is many to many relationship.
3. Many students can get grades for many courses that they have enrolled for.
4. One student can make payment for one or more courses. Hence the relationship between student and payment is a one to many relationship.
5. Many Instructors can teach many courses. Hence the relationship between instructor and courses is also a many to many relationship.
6. One instructor can enroll to teach many courses.
7. Many payments can be made for multiple enrollments.

**Functional Dependencies:**

Student\_ID Email\_ID, First\_name, Last\_name, Phone\_No, Address, Date of birth, Gender, Department

Course\_ID Course\_Name, Course\_Description, No\_of\_credits, Duration, Semester\_offered, Class\_room

Instructor\_ID First\_name, Last\_name , Phone\_No, Email\_ID, Office\_room, Address, Department

Payment\_ID Account balance, Payment\_due, semester/year.

Enrollment\_ID Status

**INTEGRITY CONSTRAINTS**

**Entity Integrity**

* Student\_ID in Student table cannot be NULL.
* Course\_ID in Course table cannot be NULL.
* Instructor\_ID in Instructor table cannot be NULL.
* Enrollment\_ID in enrollment table cannot be NULL.
* Payment\_ID in Payment table cannot be NULL.
* Grade in the Grade table can only be two characters.

Domain constraints

For Student table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Domain Name | Meaning | Domain Definition |
| Student\_ID | Student identification | Set of all possible student id’s | Integer size = 7, range =0000000 to 9999999 |
| Email\_ID | Student’s Email ID | Set of all possible student’s email ID’s | Character size=30 |
| First\_name | Student’s First name | Set of all possible student’s first names | Character size =20 |
| Last\_name | Student’s Last name | Set of all possible student’s last names | Character size =20 |
| Phone\_No | Student’s Phone Number | Set of all possible student phone Numbers | Character size =30  Range =000-000-0000  To 999-999-9999 |
| Address | Student’s Address | Set of all possible student’s addresses | Character size = 100 |
| Date of birth | Student’s Date of Birth | Set of all possible student’s date of births | Date format -mm/dd/yyyy |
| Gender | Student’s gender | Set of all possible student’s gender | Character size from 1 to 6 |
| Department | Student’s Department | Set of all possible student’s department | Character size = 20 |

For Course table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Domain Name | Meaning | Domain Definition |
| Course\_ID | Course identification | Set of all possible course id’s | Integer size = 7, range =0000000 to 9999999 |
| Course\_Name | Courses Names | Set of all possible course names | Character size=20 |
| No\_of\_credits | Number of credits | Set of all possible course credits | Number (5,2)  Example (3.00) |
| Duration | Course duration | Set of all course’s duration | Character size = 20 |
| Semester\_offered | Semester during which course offered | Set of all courses offered during each semester | Character size = 20 |
| Class\_room | Classroom where each course | Set of all possible course classrooms | Character size = 20 |
| Department | Department’s | Set of all possible department’s courses | Character size = 30 |

For instructor table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Domain Name | Meaning | Domain Definition |
| Instructor\_ID | Student identification | Set of all possible Instructor ID’s | Integer size = 7, range =0000000 to 9999999 |
| First\_name | Instructor’s First name | Set of all possible Instructor’s first names | Character size =20 |
| Last\_name | Instructor’s Last name | Set of all possible Instructor’s last names | Character size =20 |
| Phone\_No | Instructor’s Phone Number | Set of all possible Instructors phone Numbers | Character size =20 |
| Address | Instructor’s Address | Set of all possible Instructor’s addresses | Character size = 100 |
| Email\_ID | Instructor’s Email\_ID | Set of all possible Instructor’s Email ID | Character size =20 |
| Office\_room | Instructor’s Office\_room | Set of all possible Instructor’s Office\_rooms | Character size from 1 to 5 |
| Department | Instructor’s Department | Set of all possible Instructor’s departments | Character size = 20 |

For payment table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Domain Name | Meaning | Domain Definition |
| Payment\_ID | Payment identification | Set of all possible payment id’s | Integer size = 7, range =0000000 to 9999999 |
| Student\_ID | Student identification | Set of all possible student ID’s | Integer size = 7, range =0000000 to 9999999 |
| Enrollment\_ID | Enrollment identification | Set of all possible student’s enrolled courses | Integer size = 7, range =0000000 to 9999999 |
| Sem/Year | Semester and year | Semester and year each course is enrolled | Character size =20 |
| Account\_Balance | Student’s account balance | Balance that is to be paid by the student is reflected by this attribute | Number size =(12,2)  For example (000000.00) |
| Payment due | Student’s Payment due | If there is any amount due, they should pay the amount within the due date. | Number size =(12,2)  For example (000000.00) |

For enrollment table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Domain Name | Meaning | Domain Definition |
| Enrollment\_ID | Enrollment identification | Set of all possible student enrollment ID’s | Integer size = 7, range =0000000 to 9999999 |
| Student\_ID | Student identification | Set of all possible student id’s | Integer size = 7, range =0000000 to 9999999 |
| Instructor\_ID | Student identification | Set of all possible Instructor ID’s | Integer size = 7, range =0000000 to 9999999 |
| Payment\_ID | Payment identification | Set of all possible payment id’s | Integer size = 7, range =0000000 to 9999999 |
| Course\_ID | Course identification | Set of all possible course id’s | Integer size = 7, range =0000000 to 9999999 |
| Status | Enrollment status | Shows whether the instructor or student has enrolled for a subject or not | Character Size = 20 |

For Grade table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Domain Name | Meaning | Domain Definition |
| Student\_ID | Student identification | Set of all possible student id’s | Integer size = 7, range =0000000 to 9999999 |
| Course\_ID | Course identification | Set of all possible course id’s | Integer size = 7, range =0000000 to 9999999 |
| Sem/Year | Semester and Year | Semester and year the student or instructor has enrolled. | Character size = 20  Example could be Fall2018 |
| Grade | Grade | Grades obtained by the student | Character size = 2  Example (A-) |
| Credit | Course credits | Credits of each course | Number (5,2) |
| Status | Status | Status indicates whether the student has passed, failed or withdrawn a subject | Character size =20 |

Student table:

Create query:

CREATE TABLE STUDENT

(

student\_ID int NOT NULL,

Email\_ID varchar(30) NOT NULL,

First\_Name varchar(20) NOT NULL,

Last\_Name varchar(20) NOT NULL,

Phone\_NO varchar(30) NOT NULL,

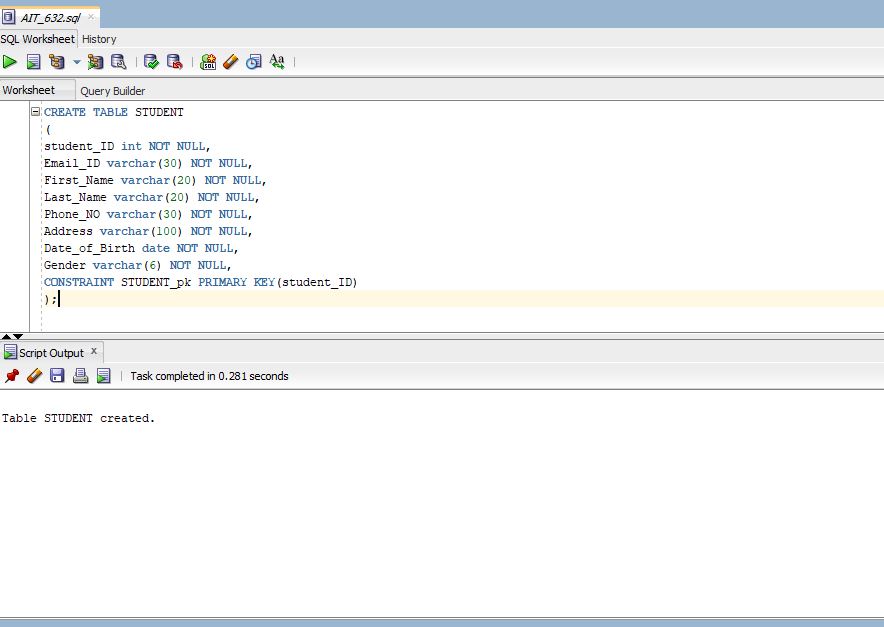
Address varchar(100) NOT NULL,

Date\_of\_Birth date NOT NULL,

Gender varchar(6) NOT NULL,

CONSTRAINT STUDENT\_pk PRIMARY KEY(student\_ID)

);



Insert query:

INSERT INTO STUDENT

(Student\_ID,Email\_ID,First\_Name,Last\_Name,Phone\_No,Address,Date\_of\_Birth,Gender) values('680234','rkachi1@students.towson.edu','Raj','kk','443-901-9595','6729 Airlie Way Baltimore','30-JUNE-1995','Male');

INSERT INTO STUDENT

(Student\_ID,Email\_ID,First\_Name,Last\_Name,Phone\_No,Address,Date\_of\_Birth,Gender) values('680132','sgobu2@students.towson.edu','Saranya','Gobu','443-231-1776','950 walnut wood road Baltimore','26-MAY-1995','Female');

INSERT INTO STUDENT

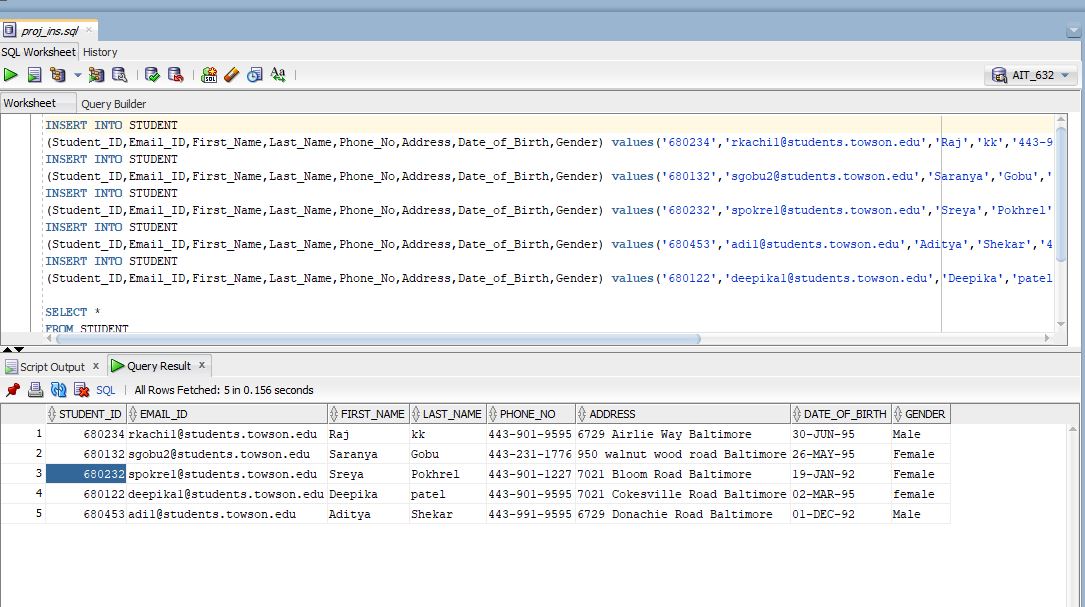
(Student\_ID,Email\_ID,First\_Name,Last\_Name,Phone\_No,Address,Date\_of\_Birth,Gender) values('680232','spokre1@students.towson.edu','Sreya','Pokhrel','443-901-1227','7021 Bloom Road Baltimore','19-JAN-1992','Female');

INSERT INTO STUDENT

(Student\_ID,Email\_ID,First\_Name,Last\_Name,Phone\_No,Address,Date\_of\_Birth,Gender) values('680453','adi1@students.towson.edu','Aditya','Shekar','443-991-9595','6729 Donachie Road Baltimore','01-DEC-1992','Male');

INSERT INTO STUDENT

(Student\_ID,Email\_ID,First\_Name,Last\_Name,Phone\_No,Address,Date\_of\_Birth,Gender) values('680122','deepika1@students.towson.edu','Deepika','patel','443-901-9595','7021 Cokesville Road Baltimore','02-MARCH-1995','female');



COURSE table

Create query:

CREATE TABLE COURSE

(

Course\_ID int NOT NULL,

Course\_name varchar(20) NOT NULL,

No\_of\_credits numeric(38) NOT NULL,

Duration varchar(20) NOT NULL,

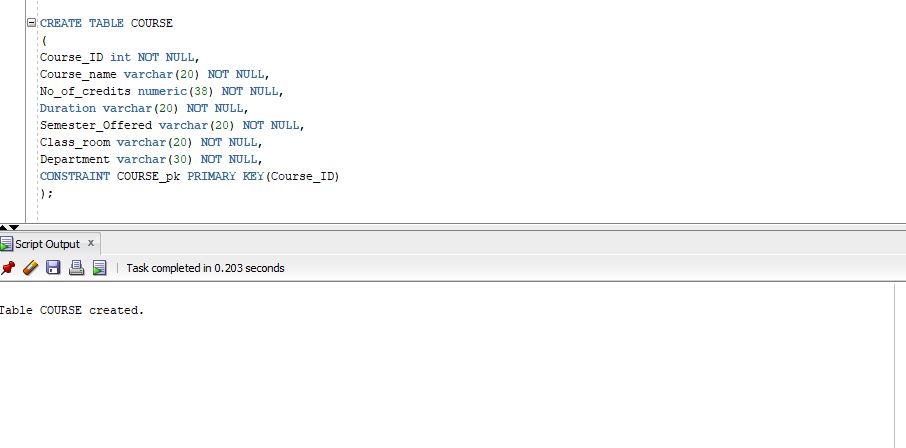
Semester\_Offered varchar(20) NOT NULL,

Class\_room varchar(20) NOT NULL,

Department varchar(30) NOT NULL,

CONSTRAINT COURSE\_pk PRIMARY KEY(Course\_ID)

);



Insert query:

INSERT INTO COURSE

(Course\_ID,Course\_Name,No\_of\_credits,duration,semester\_offered,class\_room,department) values('632','DBMS','3','6 months','Fall 2018','ES102','AIT');

INSERT INTO COURSE

(Course\_ID,Course\_Name,No\_of\_credits,duration,semester\_offered,class\_room,department) values('624','Softengfund','3','6 months','Spring 2018','YR104','AIT');

INSERT INTO COURSE

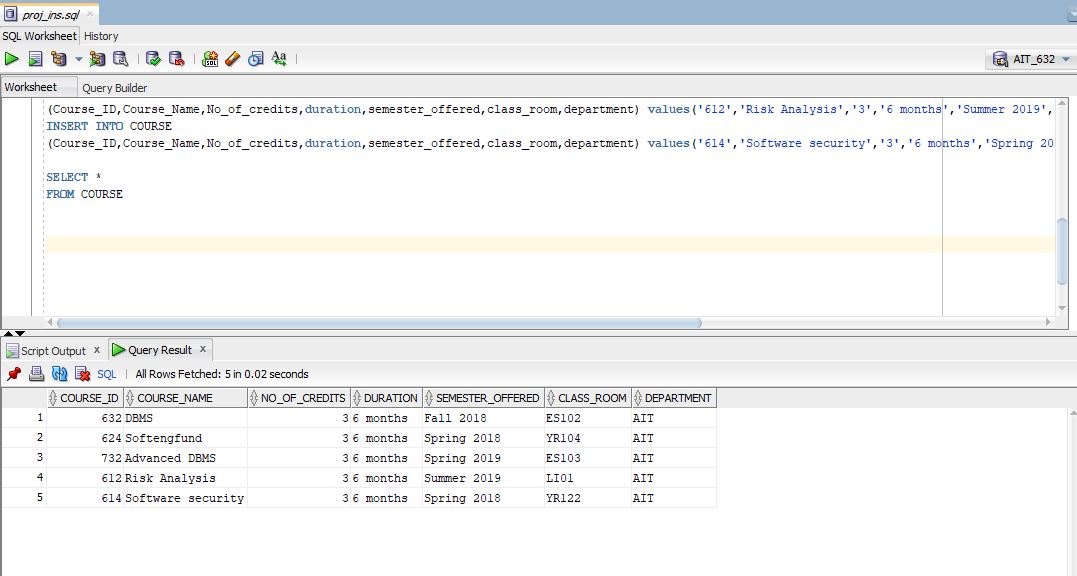
(Course\_ID,Course\_Name,No\_of\_credits,duration,semester\_offered,class\_room,department) values('732','Advanced DBMS','3','6 months','Spring 2019','ES103','AIT');

INSERT INTO COURSE

(Course\_ID,Course\_Name,No\_of\_credits,duration,semester\_offered,class\_room,department) values('612','Risk Analysis','3','6 months','Summer 2019','LI01','AIT');

INSERT INTO COURSE

(Course\_ID,Course\_Name,No\_of\_credits,duration,semester\_offered,class\_room,department) values('614','Software security','3','6 months','Spring 2018','YR122','AIT');



Instructor table

Create query:

CREATE TABLE INSTRUCTOR

(

Instructor\_ID int NOT NULL,

First\_name varchar(20) NOT NULL,

Last\_name varchar(20) NOT NULL,

Phone\_no varchar(20) NOT NULL,

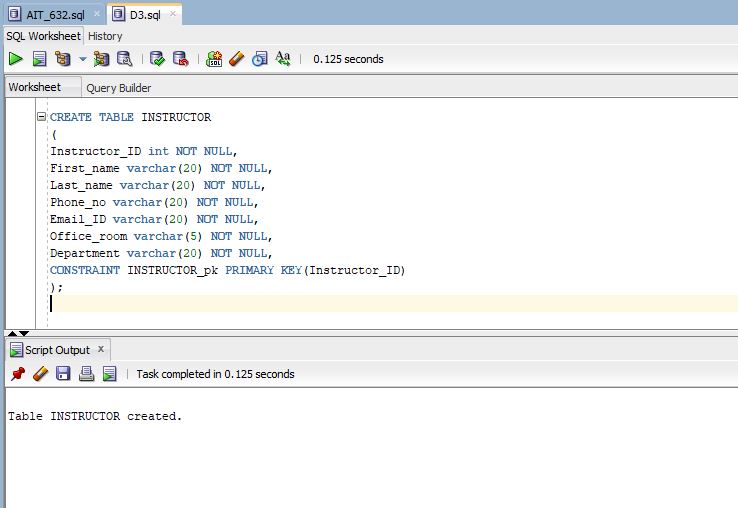
Email\_ID varchar(20) NOT NULL,

Office\_room varchar(5) NOT NULL,

Department varchar(20) NOT NULL,

CONSTRAINT INSTRUCTOR\_pk PRIMARY KEY(Instructor\_ID)

);



Insert Query:

INSERT INTO INSTRUCTOR

(Instructor\_ID,First\_Name,Last\_Name,Phone\_No,Email\_ID,Office\_room,department)values('4012','Jinie','Pak','443-301-8989','jpak1@towson.edu','YR401','CIS');

INSERT INTO INSTRUCTOR

(Instructor\_ID,First\_Name,Last\_Name,Phone\_No,Email\_ID,Office\_room,department)values('4011','Cheryl','Brown','443-121-8989','cbrown1@towson.edu','YR444','CIS');

INSERT INTO INSTRUCTOR

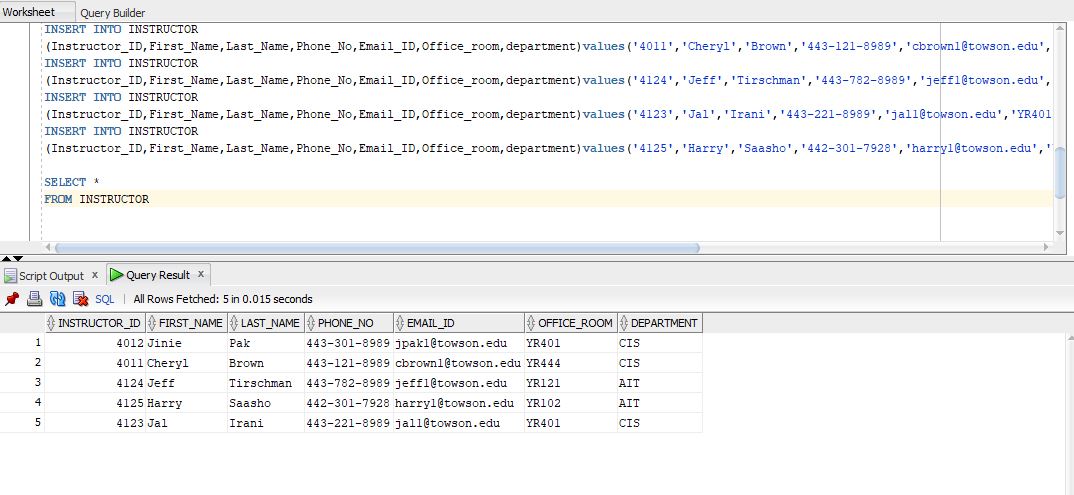
(Instructor\_ID,First\_Name,Last\_Name,Phone\_No,Email\_ID,Office\_room,department)values('4124','Jeff','Tirschman','443-782-8989','jeff1@towson.edu','YR121','AIT');

INSERT INTO INSTRUCTOR

(Instructor\_ID,First\_Name,Last\_Name,Phone\_No,Email\_ID,Office\_room,department)values('4123','Jal','Irani','443-221-8989','jal1@towson.edu','YR401','CIS');

INSERT INTO INSTRUCTOR

(Instructor\_ID,First\_Name,Last\_Name,Phone\_No,Email\_ID,Office\_room,department)values('4125','Harry','Saasho','442-301-7928','harry1@towson.edu','YR102','AIT');



Enrollment table

Create query:

CREATE TABLE ENROLLMENT

(

Enrollment\_ID int NOT NULL,

Student\_ID int NOT NULL,

Course\_ID int NOT NULL,

Payment\_ID int NULL,

Instructor\_ID int NOT NULL,

Status varchar(20) NOT NULL,

CONSTRAINT ENROLLMENT\_pk PRIMARY KEY(Enrollment\_ID),

CONSTRAINT Student\_ID\_fk FOREIGN KEY(Student\_ID) REFERENCES STUDENT(Student\_ID),

CONSTRAINT Course\_ID\_fk FOREIGN KEY(Course\_ID) REFERENCES COURSE(Course\_ID),

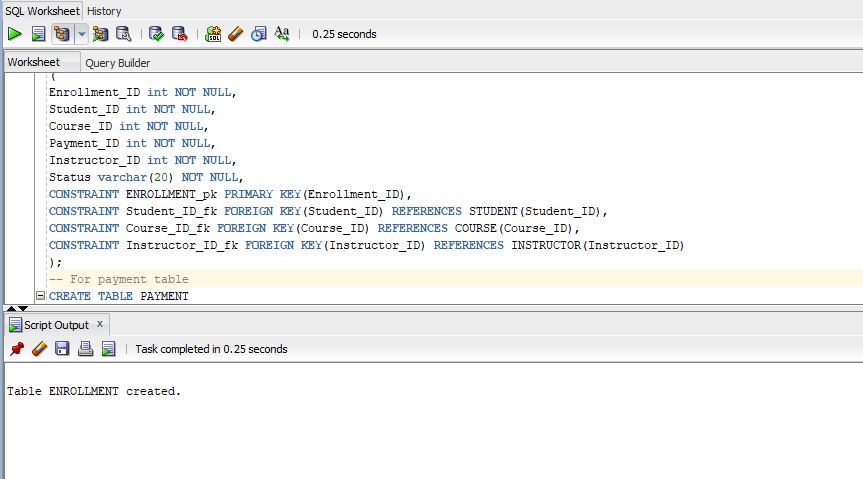
CONSTRAINT Instructor\_ID\_fk FOREIGN KEY(Instructor\_ID) REFERENCES INSTRUCTOR(Instructor\_ID)

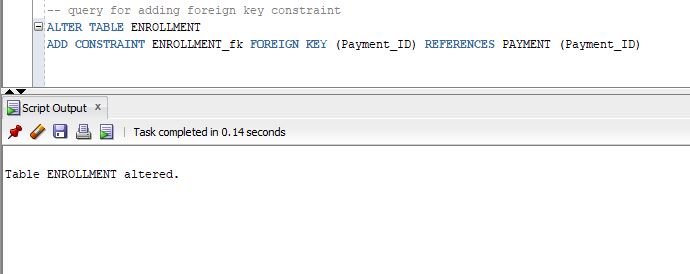
);

-- query for adding foreign key constraint

ALTER TABLE ENROLLMENT

ADD CONSTRAINT ENROLLMENT\_fk FOREIGN KEY (Payment\_ID) REFERENCES PAYMENT (Payment\_ID)





Insert Query:

INSERT INTO ENROLLMENT

(Enrollment\_ID,Student\_ID,Course\_ID,Payment\_ID,Instructor\_ID,Status)values ('64023','680234','632','','4012','enrolled');

INSERT INTO ENROLLMENT

(Enrollment\_ID,Student\_ID,Course\_ID,Payment\_ID,Instructor\_ID,Status)values ('64054','680132','624','','4011','withdrawn');

INSERT INTO ENROLLMENT

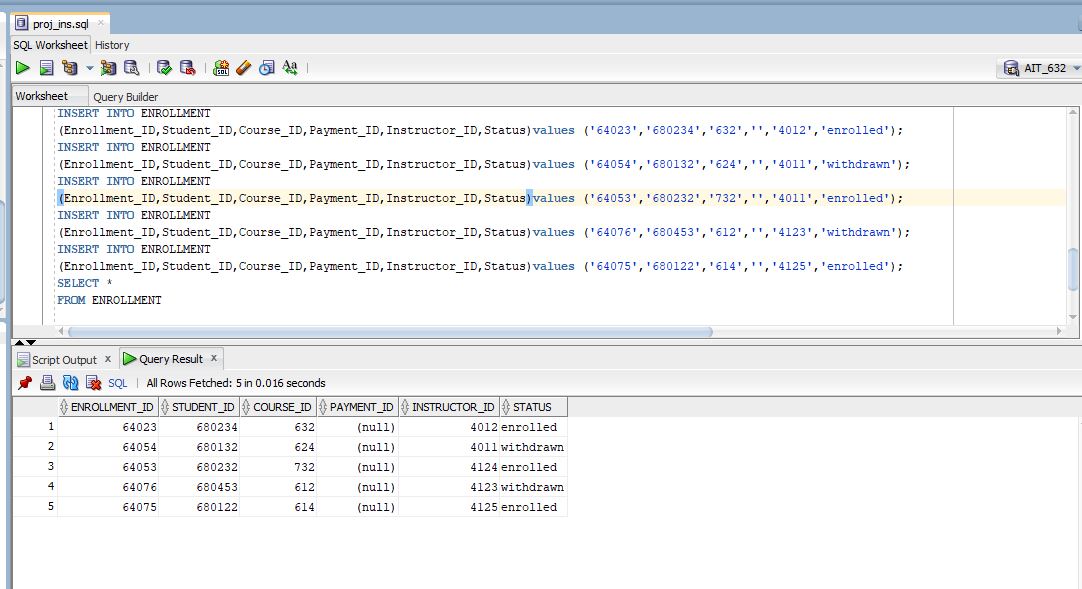
(Enrollment\_ID,Student\_ID,Course\_ID,Payment\_ID,Instructor\_ID,Status)values ('64053','680232','732','','4011','enrolled');

INSERT INTO ENROLLMENT

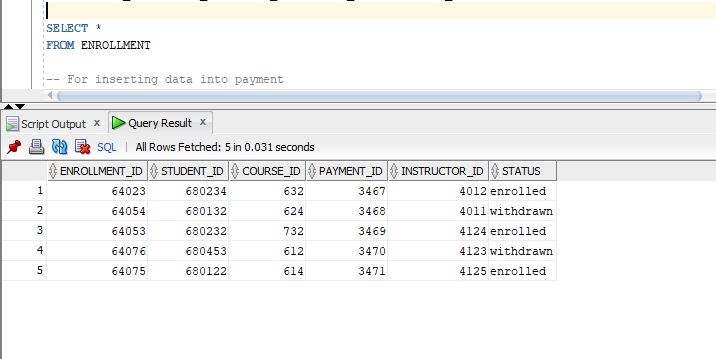
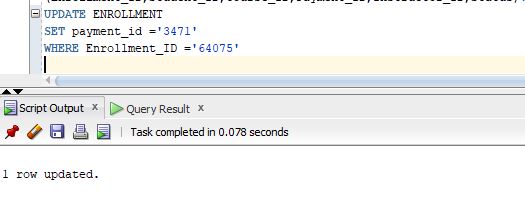
(Enrollment\_ID,Student\_ID,Course\_ID,Payment\_ID,Instructor\_ID,Status)values ('64076','680453','612','','4123','withdrawn');

INSERT INTO ENROLLMENT

(Enrollment\_ID,Student\_ID,Course\_ID,Payment\_ID,Instructor\_ID,Status)values ('64075','680122','614','','4125','enrolled');



Note: Since the table enrollment has foreign key constrain on payment\_ID. Null value has been inserted initially and updated with Payment\_ID later when payment\_ID has been created.



Payment table:

CREATE TABLE PAYMENT

(

Payment\_ID int NOT NULL,

student\_ID int NOT NULL,

Enrollment\_ID int NOT NULL,

Account\_Balance numeric(12,2) NOT NULL,

Payment\_due numeric(12,2) NOT NULL,

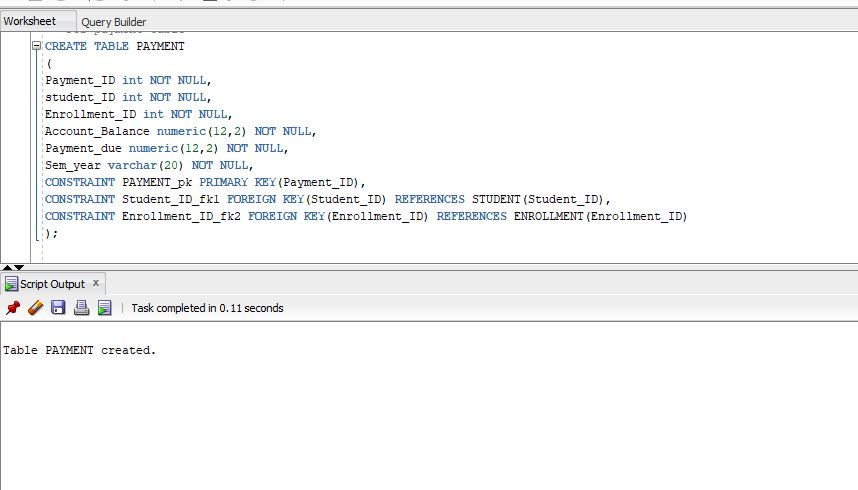
Sem\_year varchar(20) NOT NULL,

CONSTRAINT PAYMENT\_pk PRIMARY KEY(Payment\_ID),

CONSTRAINT Student\_ID\_fk1 FOREIGN KEY(Student\_ID) REFERENCES STUDENT(Student\_ID),

CONSTRAINT Enrollment\_ID\_fk2 FOREIGN KEY(Enrollment\_ID) REFERENCES ENROLLMENT(Enrollment\_ID)

);



Insert payment:

INSERT INTO PAYMENT

(Payment\_ID,Student\_ID,enrollment\_ID,account\_balance,payment\_due,sem\_year)values('3467','680234','64023','234.80','100.80','Fall2018');

INSERT INTO PAYMENT

(Payment\_ID,Student\_ID,enrollment\_ID,account\_balance,payment\_due,sem\_year)values('3468','680132','64054','118.80','10.80','Spring2019');

INSERT INTO PAYMENT

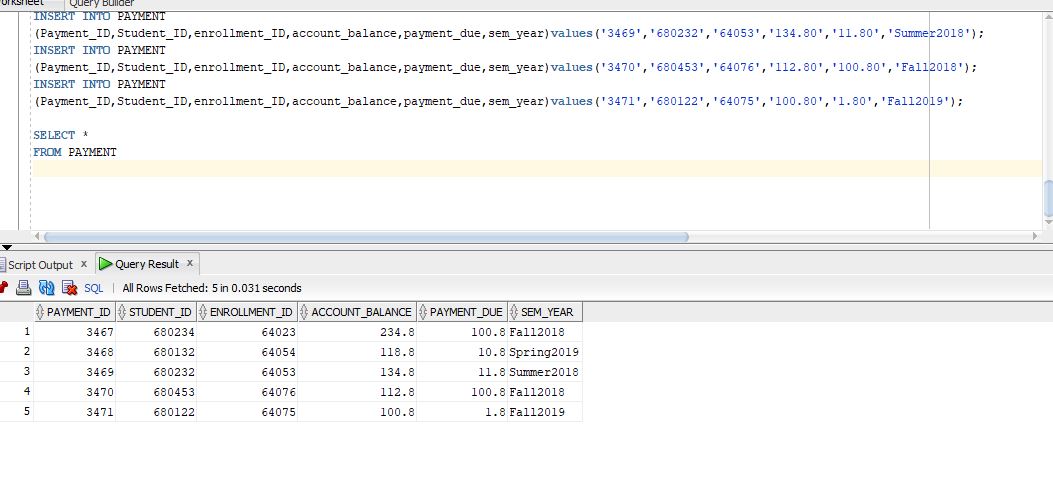
(Payment\_ID,Student\_ID,enrollment\_ID,account\_balance,payment\_due,sem\_year)values('3469','680232','64053','134.80','11.80','Summer2018');

INSERT INTO PAYMENT

(Payment\_ID,Student\_ID,enrollment\_ID,account\_balance,payment\_due,sem\_year)values('3470','680453','64076','112.80','100.80','Fall2018');

INSERT INTO PAYMENT

(Payment\_ID,Student\_ID,enrollment\_ID,account\_balance,payment\_due,sem\_year)values('3471','680122','64075','100.80','1.80','Fall2019');



Grade table:

CREATE TABLE GRADE

(

Student\_ID int NOT NULL,

Course\_ID int NOT NULL,

Grade varchar(2) NOT NULL,

Credit int NOT NULL,

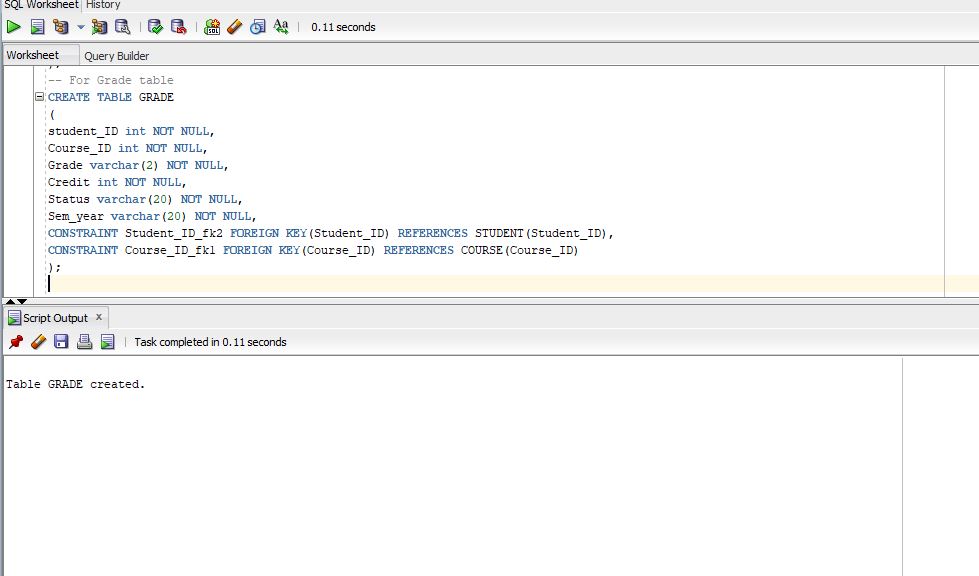
Status varchar(20) NOT NULL,

Sem\_year varchar(20) NOT NULL,

CONSTRAINT Student\_ID\_fk2 FOREIGN KEY(Student\_ID) REFERENCES STUDENT(Student\_ID),

CONSTRAINT Course\_ID\_fk2 FOREIGN KEY(Course\_ID) REFERENCES COURSE(Course\_ID)

);



Insert Query:

INSERT INTO GRADE

(Student\_ID,Course\_ID,Grade,Credit,Status,Sem\_year)VALUES ('680234','632','A-','3','Completed','Fall 2018');

INSERT INTO GRADE

(Student\_ID,Course\_ID,Grade,Credit,Status,Sem\_year) VALUES('680132','624','B+','3','Completed','Spring 2018');

INSERT INTO GRADE

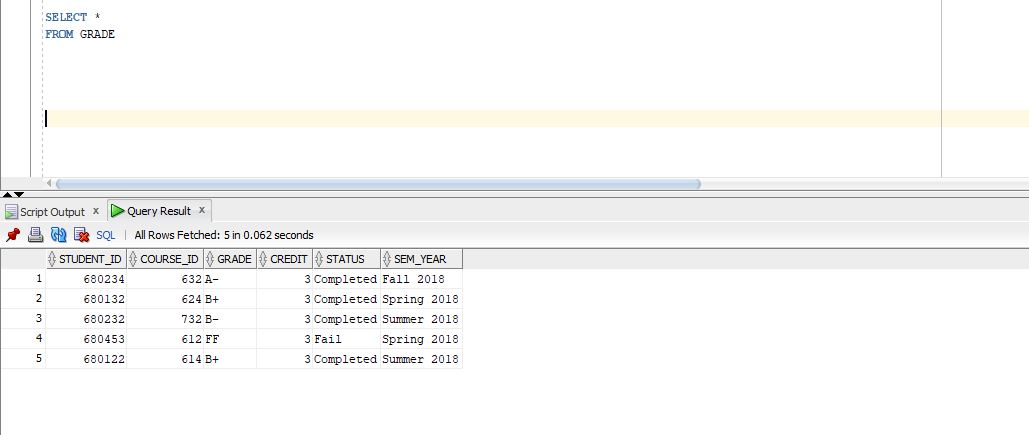
(Student\_ID,Course\_ID,Grade,Credit,Status,Sem\_year) VALUES('680232','732','B-','3','Completed','Summer 2018');

INSERT INTO GRADE

(Student\_ID,Course\_ID,Grade,Credit,Status,Sem\_year) VALUES('680453','612','FF','3','Fail','Spring 2018');

INSERT INTO GRADE

(Student\_ID,Course\_ID,Grade,Credit,Status,Sem\_year) VALUES('680122','614','B+','3','Completed','Summer 2018');



Real world Queries:

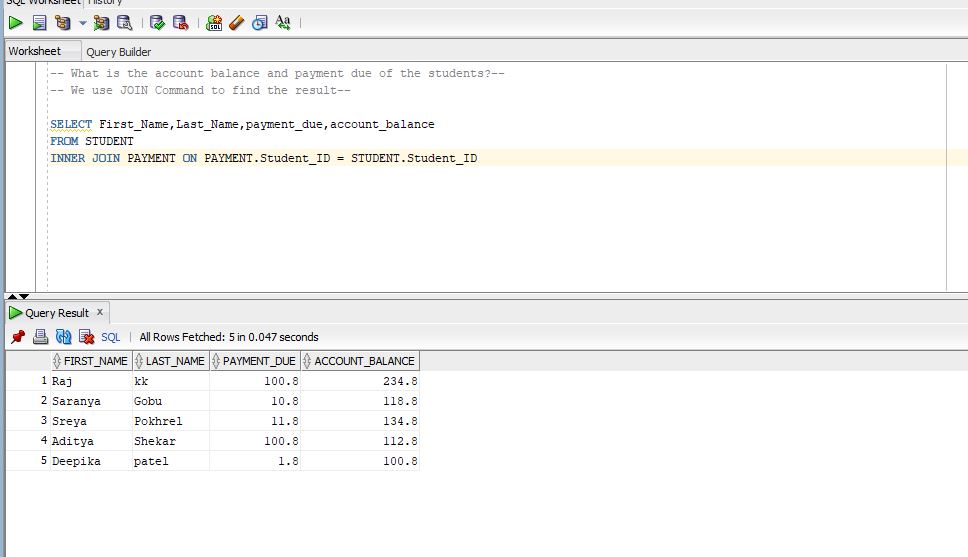
Question 1: What is the account balance and payment due of the students?

Solution: We use JOIN Command to find the result--

SELECT First\_Name,Last\_Name,payment\_due,account\_balance

FROM STUDENT

INNER JOIN PAYMENT ON PAYMENT.Student\_ID = STUDENT.Student\_ID



Question 2:

Show all the Email ID's of every person(Student and Instructor) associated with all Deprtments

Solution:

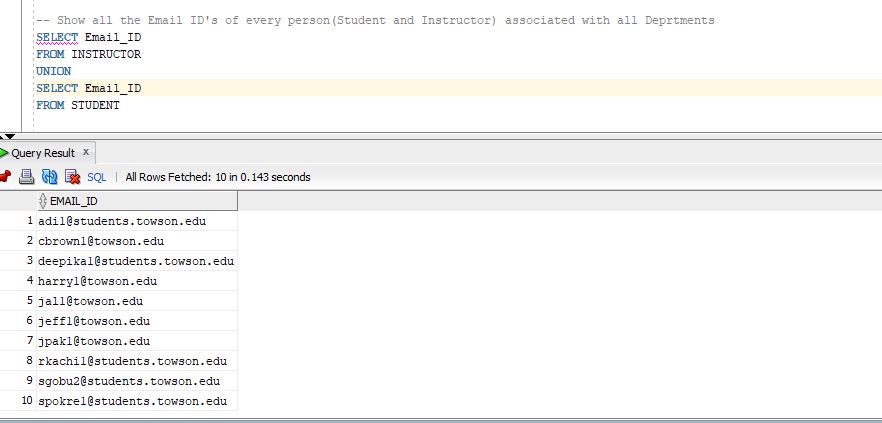
SELECT Email\_ID

FROM INSTRUCTOR

UNION

SELECT Email\_ID

FROM STUDENT



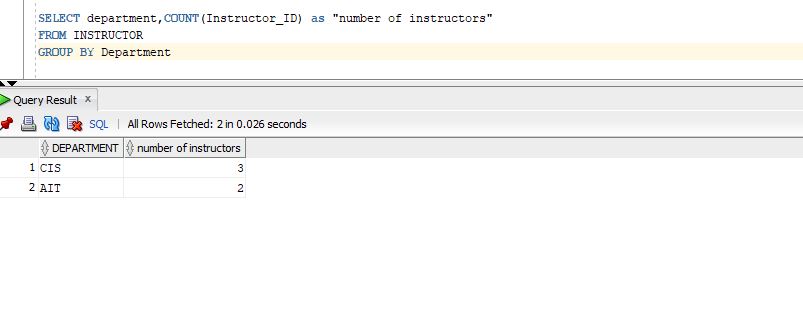
Question 3: How many instructors are present in each department?

Solution:

SELECT department,COUNT(Instructor\_ID) as "number of instructors"

FROM INSTRUCTOR

GROUP BY Department



Question 4: Show all the STUDENT\_ID who have withdrawn the courses?

Solution:

SELECT STUDENT\_ID

FROM STUDENT

INTERSECT

SELECT STUDENT\_ID

FROM ENROLLMENT

WHERE STATUS = ’withdrawn’



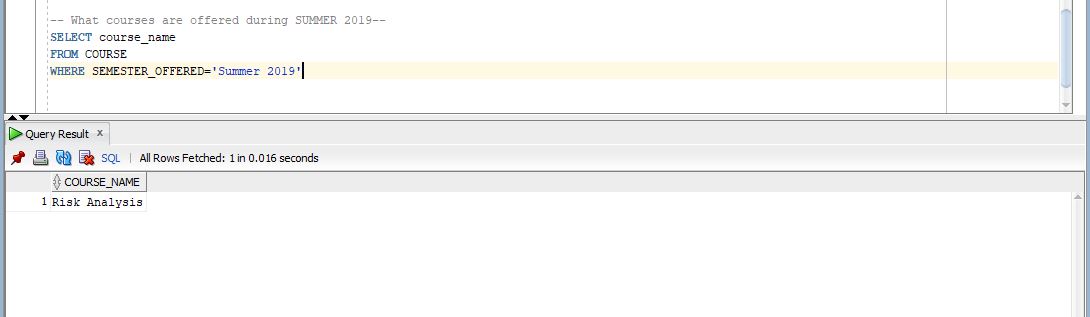
Question 5: What courses are offered during SUMMER 2019?

Solution:

SELECT COURSE\_NAME

FROM COURSE

WHERE SEMESTER\_OFFERED='Summer 2019'



Question 6: Show the student who has the maximum account balance?

Solution:

SELECT FIRST\_NAME,LAST\_NAME

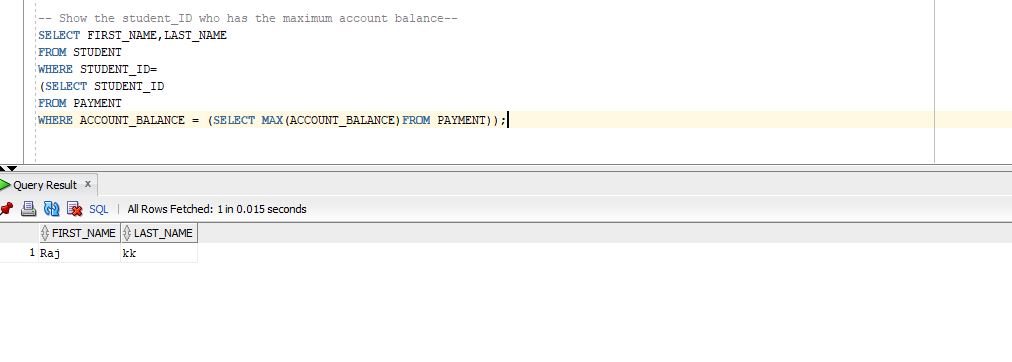
FROM STUDENT

WHERE STUDENT\_ID=

(SELECT STUDENT\_ID

FROM PAYMENT

WHERE ACCOUNT\_BALANCE = (SELECT MAX(ACCOUNT\_BALANCE)FROM PAYMENT));



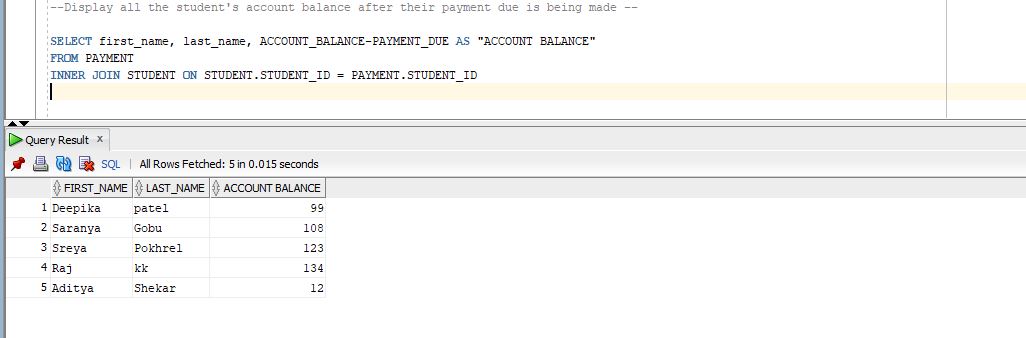
Question 7: Display all the student's account balance after their payment due is being made

Solution:

SELECT first\_name, last\_name, ACCOUNT\_BALANCE-PAYMENT\_DUE AS "ACCOUNT BALANCE"

FROM PAYMENT

INNER JOIN STUDENT ON STUDENT.STUDENT\_ID = PAYMENT.STUDENT\_ID



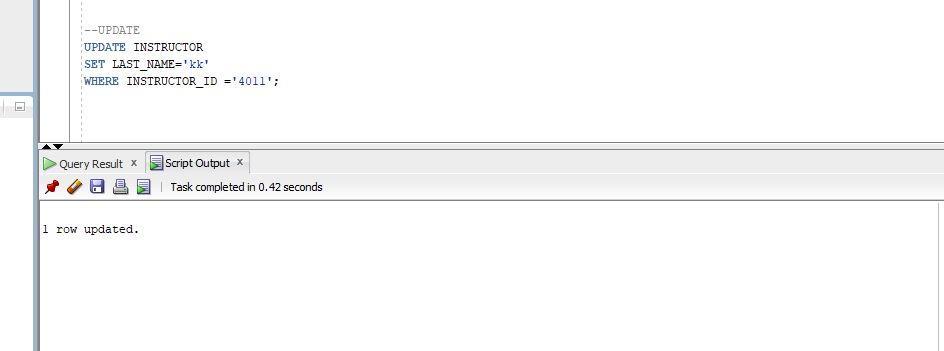
Question 8: Update last name of the instructor whose instructor\_ID is 4011.

Solution:

UPDATE INSTRUCTOR

SET LAST\_NAME='kk'

WHERE INSTRUCTOR\_ID ='4011';



Question 9:Display the last names of the student's which are distinct from instructor's last name--

Solution:

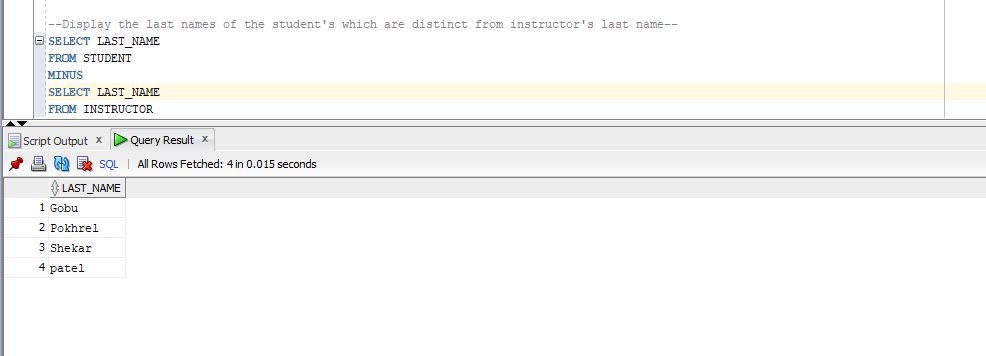
SELECT LAST\_NAME

FROM STUDENT

MINUS

SELECT LAST\_NAME

FROM INSTRUCTOR



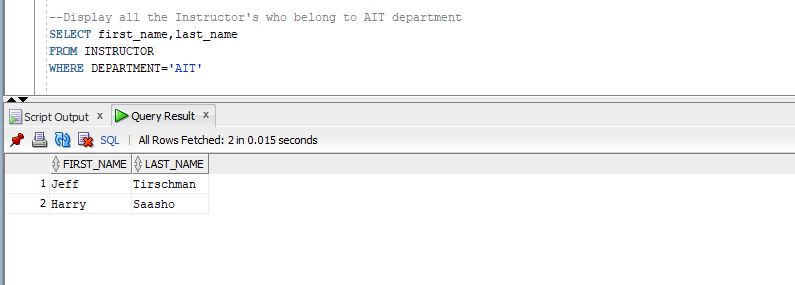
Question 10: --Display all the Instructor's who belong to AIT department

Solution:

SELECT first\_name,last\_name

FROM INSTRUCTOR

WHERE DEPARTMENT=’AIT’



Question 12: List all the names and IDs of students who have a B grade.

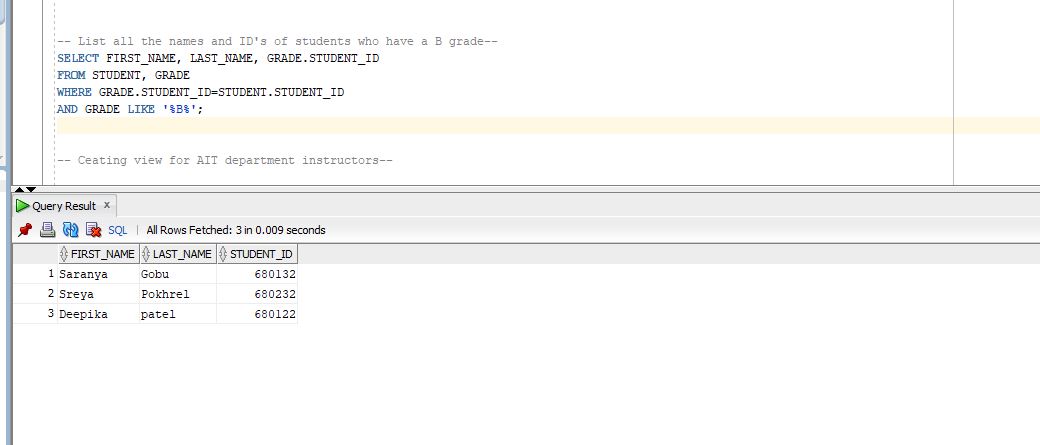
Solution:

SELECT FIRST\_NAME, LAST\_NAME, GRADE.STUDENT\_ID

FROM STUDENT, GRADE

WHERE GRADE.STUDENT\_ID=STUDENT.STUDENT\_ID

AND GRADE LIKE '%B%';



Question 11: List the instructor(s) situated in office\_room YR102 in AIT department

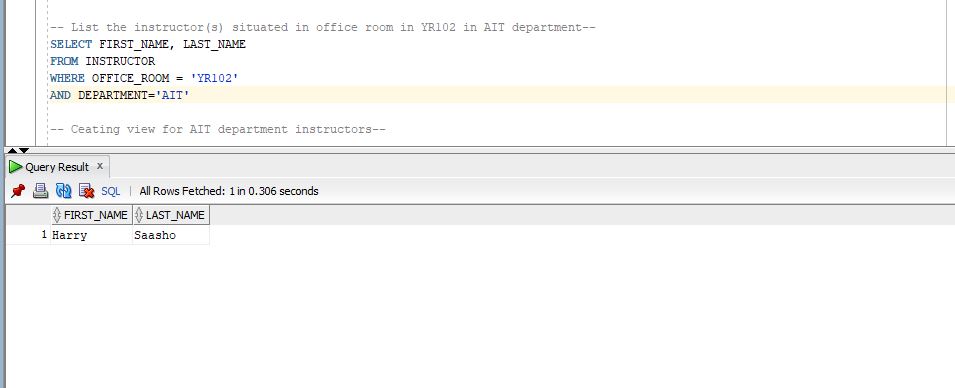
Solution:

SELECT FIRST\_NAME, LAST\_NAME

FROM INSTRUCTOR

WHERE OFFICE\_ROOM = 'YR102'

AND DEPARTMENT='AIT'



Creating view for AIT department instructors

CREATE VIEW AIT\_department\_instructors AS

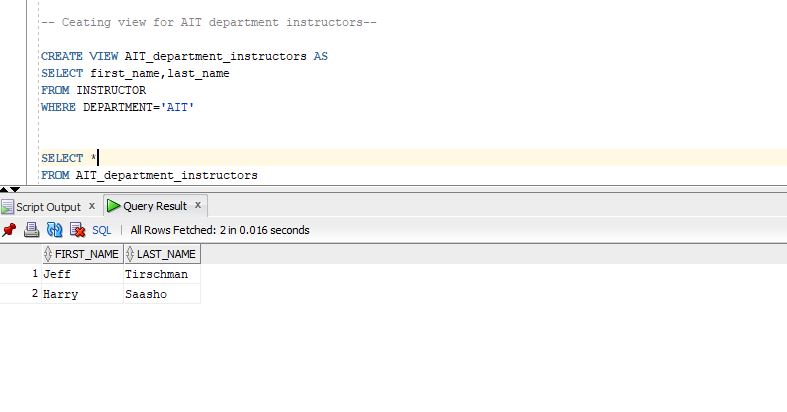
SELECT first\_name,last\_name

FROM INSTRUCTOR

WHERE DEPARTMENT='AIT'

SELECT \*

FROM AIT\_department\_instructors



Creating view for students who have withdrawn courses

CREATE VIEW withdrawn\_students AS

SELECT STUDENT\_ID

FROM STUDENT

INTERSECT

SELECT student\_ID

FROM ENROLLMENT

WHERE STATUS='withdrawn'

SELECT \*

FROM withdrawn\_students

