Bahçeşehir University

SEN2104 DATABASE MANAGEMENT SYSTEMS TERM PROJECT

University Database

Group Members :

Mert Oğuz	**1904634**
Mehmet Ali Kısacık	**1904537**
Buse Gizem Sezer	**1902029**

TABLE OF CONTENTS

A

B

C

D

E

F

Project Definition

Nowadays, a large amount of data is collected and saved for all parts of daily life, particularly in schooling and universities. The database is the only professional organization for organizing, manipulating, and retrieving disparate data structures.

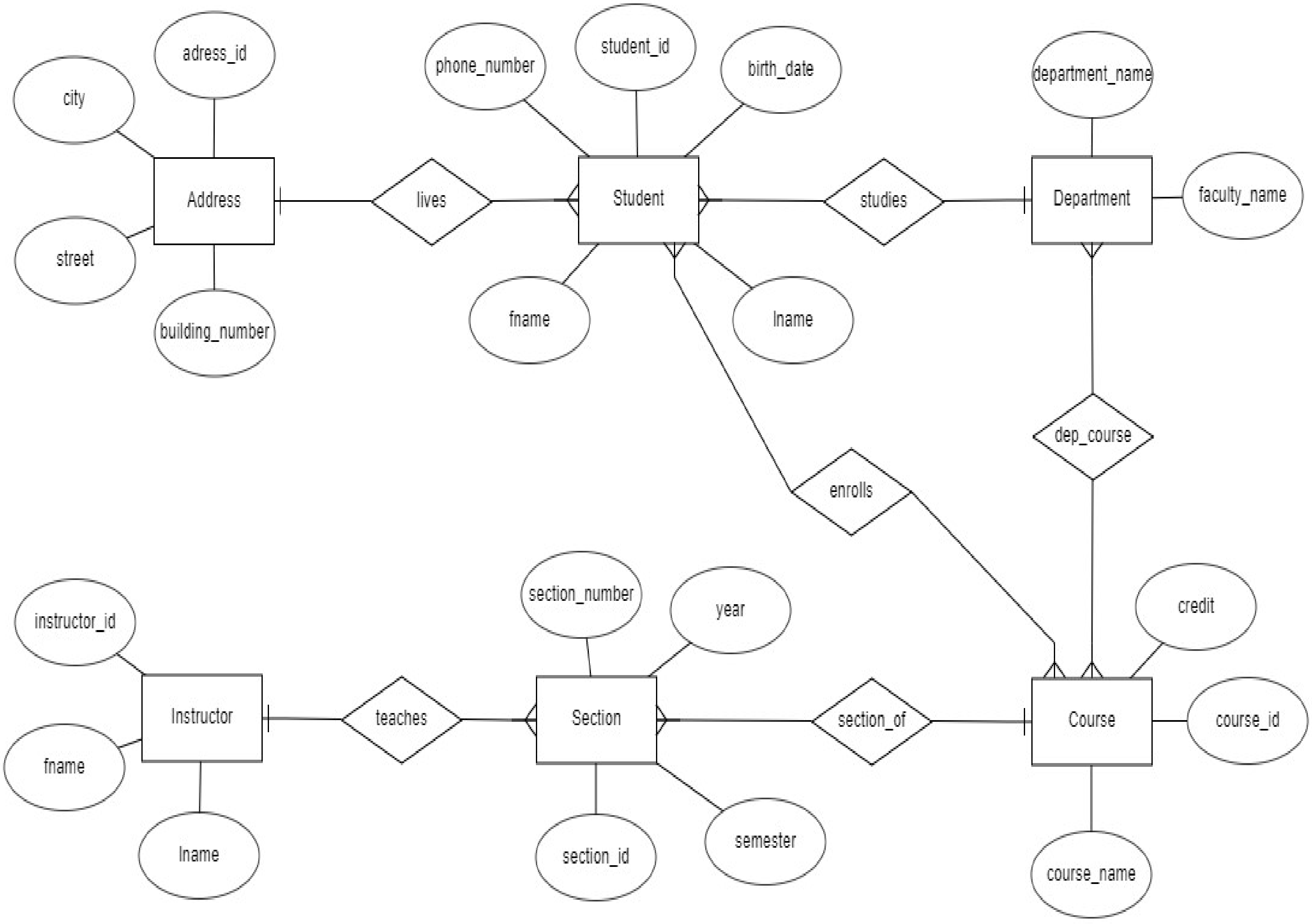
This project aims to manage the university with the given information and provide easy access to the personal information of students and instructors and keep track of their department, course, and section information.

There are many students at the college and each student has only one address, but there can be more than one student at an address. Each student has only one department but can take more than one course and each course has more than one section. These sections have an instructor, but these instructors can take more than one section.

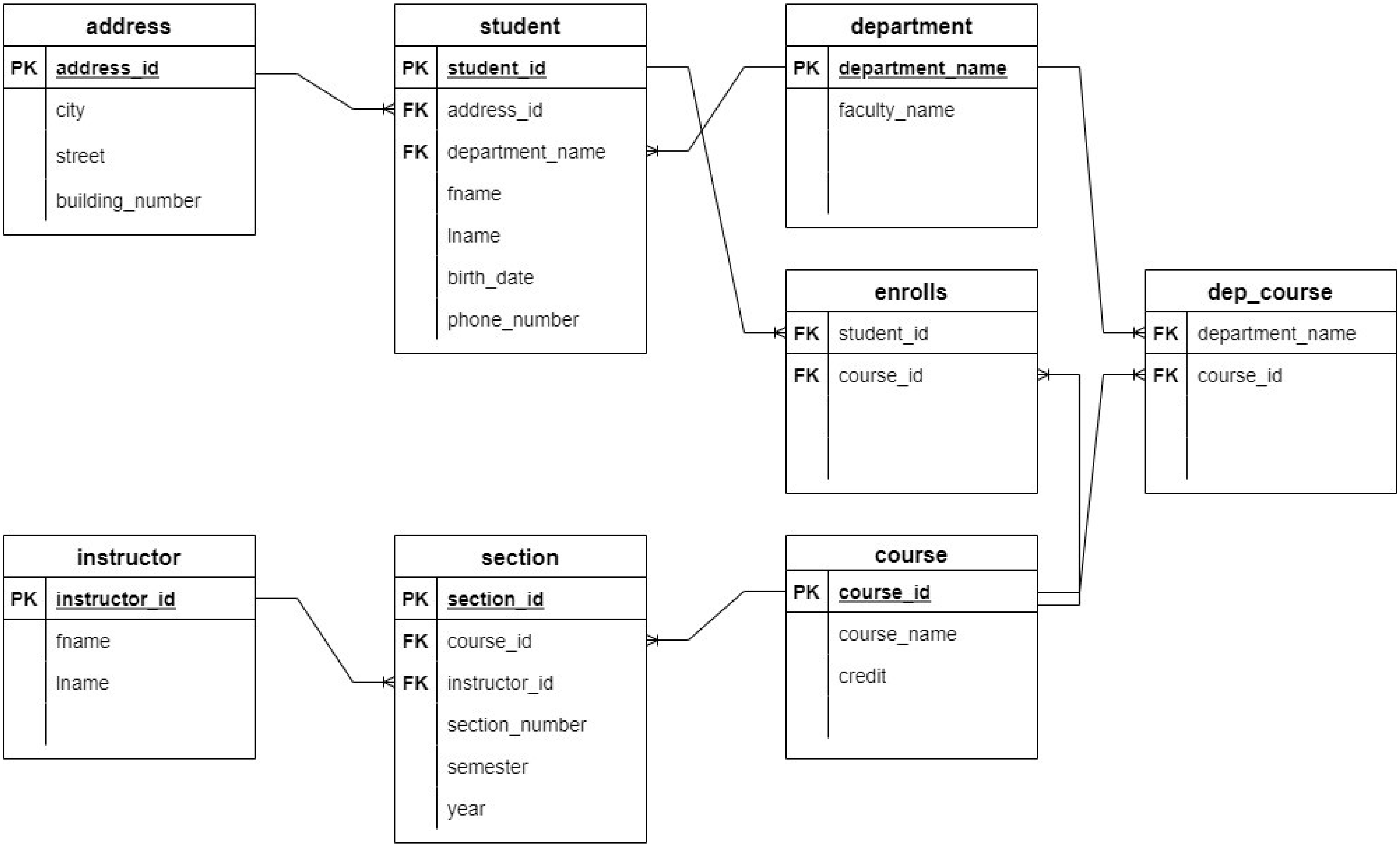
Task Responsibilities of Each Member

	Planning	A	B	C	D	E	F
Date	**28- 30th of December**	**30-31st of December**	**1-2nd of January**	**2-3rd of January**	**3rd of January**	**3rd of January**	**4-6th of January**
Mert	**Reviewing documents and making plans.**	Determining the subject of the project and making a plan.	Design E-R diagram based on aim of the project.	Drawing schema diagram based on an E-R diagram.			
Mehmet	**Identifying the places where we will have difficulty and focusing on them.**			Checking the tables whether the tables are in BCNF form.	Write create table statements for each of the tables.		
Buse	**Elimination of deficiencies in boyce codd normal form notation.**					Write at least 3 insert statements for each of the tables.	Write different queries with their descriptions and test them.

E-R Diagram


adress_id
city
Address
street
building_number
lives
phone_number
student_id
birth_date
Student
fname
lname
studies
department_name
Department
faculty_name
dep_course
enrolls
instructor_id
Instructor
fname
lname
teaches
section_number
year
Section
section_id
semester
section_of
credit
Course
course_id
course_name

Database Schema Diagram



Boyce-Codd Normal Form Notation

For Lives Relation

lives(student_id, fname, lname, birth_date, phone_number, address_id, department_name, city, street, building_number)

F = { student_id -> fname, lname, birth_date, phone_number, address_id, department_name ;

address_id -> city, street, building_number

}

Closures:

student_id+ = student_id ,fname, lname, birth_date, phone_number, address_id, department_name , city, street, building_number

address_id+ = address_id, city, street, building_number

Decomposition:

student_id is superkey. Because student_id+ contains all attributes of R (lives).

address_id is not a superkey because it doesn’t contain all attributes of R (lives) in its closure we need to decompose.

R2 = { student_id, fname, lname, birth_date, phone_number, address_id, department_name }

R3 = { address_id, city, street, building_number }

Final decomposition:

R2, R3 and the tables conform the relation.

For Studies Relation

studies(student_id, fname, lname, birth_date, phone_number, address_id, department_name,, faculty_name)

F = { student_id -> fname, lname, birth_date, phone_number, address_id, department_name ; department_name -> faculty_name

}

Closures:

student_id+ = student_id ,fname, lname, birth_date, phone_number, address_id, department_name, faculty_name

department_name+ = faculty_name

Decomposition:

student_id is superkey. Because student_id+ contains all attributes of R (studies).

department_name is not superkey therefore we need to decompose to conform BCNF Form.

R2 = { student_id, fname, lname, birth_date, phone_number, address_id, department_name}

R3 = { department_name, faculty_name }

Final decomposition:

R2, R3 and the tables conform the relation.

For Enrolls Relation

enrolls(student_id, fname, lname, birth_date, phone_number, address_id, departmen_name, course_id, course _name, credit)

F = { student_id -> fname, lname, birth_date, phone_number, address_id, department_name ;

course_id -> course _name, credit;

}

Key = { student_id, course_id }

Closures:

student_id+ = student_id ,fname, lname, birth_date, phone_number, address_id, department_name

course_id+ = course _name, credit

Decomposition:

student_id is not superkey. Because student_id+ doesn't contain all attributes of R (enrolls).

R1 = { student_id, course_id, course _name, credit }

R2 = { student_id, fname, lname, birth_date, phone_number, address_id, department_name}

course_id is not superkey because all attributes doesn't contain R1. So we decompose R1 again.

R3 = { course_id, course _name, credit } R4 = { student_id, course_id }

Final decomposition:

R2, R3, R4 and the tables conform the relation.

For Section_of Relation

section_of(section_id, section _number, semester, year, course_id, course _name, credit,instructor_id)

F = { section_id -> section _number, semester, year, course_id,instructor_id;

course_id -> course _name, credit

}

Closures:

section_id+ = section _number, semester, year, course_id, course _name, credit ,instructor_id

course_id+ = course _name, credit

Decomposition:

section _id is superkey. Because section _id+ contains all attributes of R (section_of).

course_id is not superkey because course_id+ doesn't contain all attributes of R (section_of).

R1 = { course_id , course_name,credit}

R2 = { section_id, section _number, semester, year, course_id,instructor_id }

Final decomposition:

R1, R2 and the tables conform the relation.

For Dep_course Relation

Dep_course(department_name, faculty_name, course_id, course_name, credit)

F = { department_name -> faculty_name course_id -> course _name, credit

}

Closures:

department_name+ = faculty_name

course_id+ = course _name, credit

Decomposition:

department_name is not superkey. Because department_name + doesn't contain all attributes of R (dep_course).

R1= { department_name, faculty_name }

R2 = { department_name,course_id,course_name,credit} course_id is not superkey therefore we decompose the R2.

R3 = { course_id ,course _name, credit}

R4 = {course_id, department_name}

Final decomposition:

R1, R3, R4 and the tables conform the relation.

For Teaches Relation

Teaches(section_id, section _number, semester, year, course_id, instructor_id, fname, name,)

F ={ section_id -> section _number, semester, year, course_id,instructor_id;

instructor_id -> fname, lname

}

Closures:

section_id+ = section _number, semester, year, course_id, instructor_id,fname,lname instructor_id+ = fname,lname

Decomposition:

instructor _id is not superkey because instructor _id+ doesn't contain all attributes of R (teaches).

So we decompose the relation to:

R2 = { section_id, section _number, semester, year, course_id, instructor_id }

R3 = { instructor_id, fname, lname }

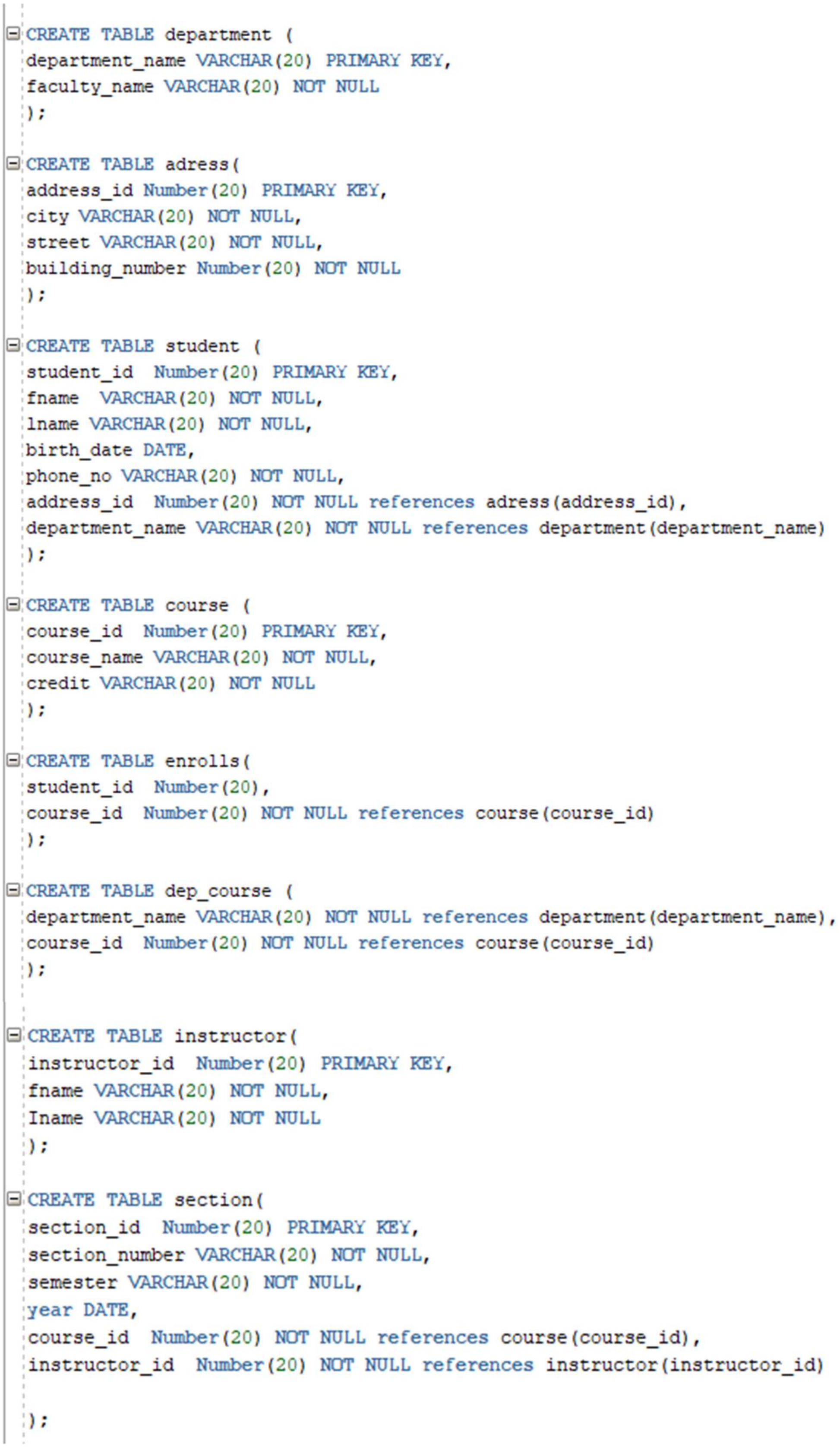
Final decomposition:

R2, R3 and the tables conform the relation.

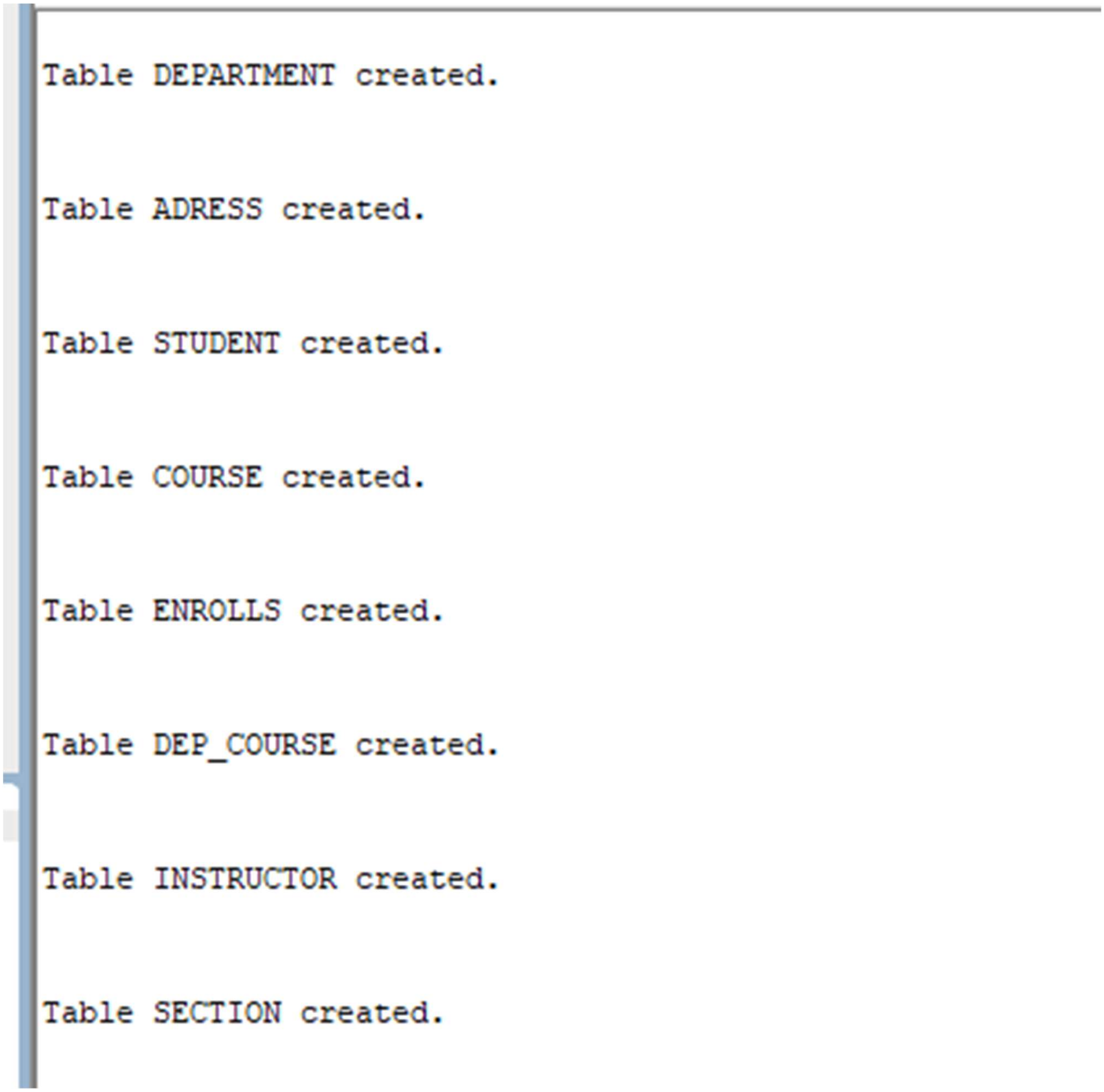
Create Table Statements

```
CREATE TABLE department (
department_name VARCHAR(20) PRIMARY KEY,
faculty_name VARCHAR(20) NOT NULL
);

CREATE TABLE address(
address_id Number(20) PRIMARY KEY,
city VARCHAR(20) NOT NULL,
street VARCHAR(20) NOT NULL,
building_number Number(20) NOT NULL
);

CREATE TABLE student (
student_id  Number(20) PRIMARY KEY,
fname  VARCHAR(20) NOT NULL,
lname VARCHAR(20) NOT NULL,
birth_date DATE,
phone_no VARCHAR(20) NOT NULL,
address_id  Number(20) NOT NULL references address(address_id),
department_name VARCHAR(20) NOT NULL references department(department_name)
);

CREATE TABLE course (
course_id  Number(20) PRIMARY KEY,
course_name VARCHAR(20) NOT NULL,
credit VARCHAR(20) NOT NULL
);
```

```
CREATE TABLE enrolls(
student_id  Number(20),
course_id  Number(20) NOT NULL references course(course_id)
);

CREATE TABLE dep_course (
department_name VARCHAR(20) NOT NULL references department(department_name),
course_id  Number(20) NOT NULL references course(course_id)
);

CREATE TABLE instructor(
instructor_id  Number(20) PRIMARY KEY,
fname VARCHAR(20) NOT NULL,
Iname VARCHAR(20) NOT NULL
);

CREATE TABLE section(
section_id  Number(20) PRIMARY KEY,
section_number VARCHAR(20) NOT NULL,
semester VARCHAR(20) NOT NULL,
year DATE,
course_id  Number(20) NOT NULL references course(course_id),
instructor_id  Number(20) NOT NULL references instructor(instructor_id)
);
```



```
CREATE TABLE department (
department_name VARCHAR(20) PRIMARY KEY,
faculty_name VARCHAR(20) NOT NULL
);

CREATE TABLE adress(
address_id Number(20) PRIMARY KEY,
city VARCHAR(20) NOT NULL,
street VARCHAR(20) NOT NULL,
building_number Number(20) NOT NULL
);

CREATE TABLE student (
student_id  Number(20) PRIMARY KEY,
fname  VARCHAR(20) NOT NULL,
lname VARCHAR(20) NOT NULL,
birth_date DATE,
phone_no VARCHAR(20) NOT NULL,
address_id  Number(20) NOT NULL references adress(address_id),
department_name VARCHAR(20) NOT NULL references department(department_name)
);

CREATE TABLE course (
course_id  Number(20) PRIMARY KEY,
course_name VARCHAR(20) NOT NULL,
credit VARCHAR(20) NOT NULL
);

CREATE TABLE enrolls(
student_id  Number(20),
course_id  Number(20) NOT NULL references course(course_id)
);

CREATE TABLE dep_course (
department_name VARCHAR(20) NOT NULL references department(department_name),
course_id  Number(20) NOT NULL references course(course_id)
);


CREATE TABLE instructor(
instructor_id  Number(20) PRIMARY KEY,
fname VARCHAR(20) NOT NULL,
Iname VARCHAR(20) NOT NULL
);

CREATE TABLE section(
section_id  Number(20) PRIMARY KEY,
section_number VARCHAR(20) NOT NULL,
semester VARCHAR(20) NOT NULL,
year DATE,
course_id  Number(20) NOT NULL references course(course_id),
instructor_id  Number(20) NOT NULL references instructor(instructor_id)

);
```

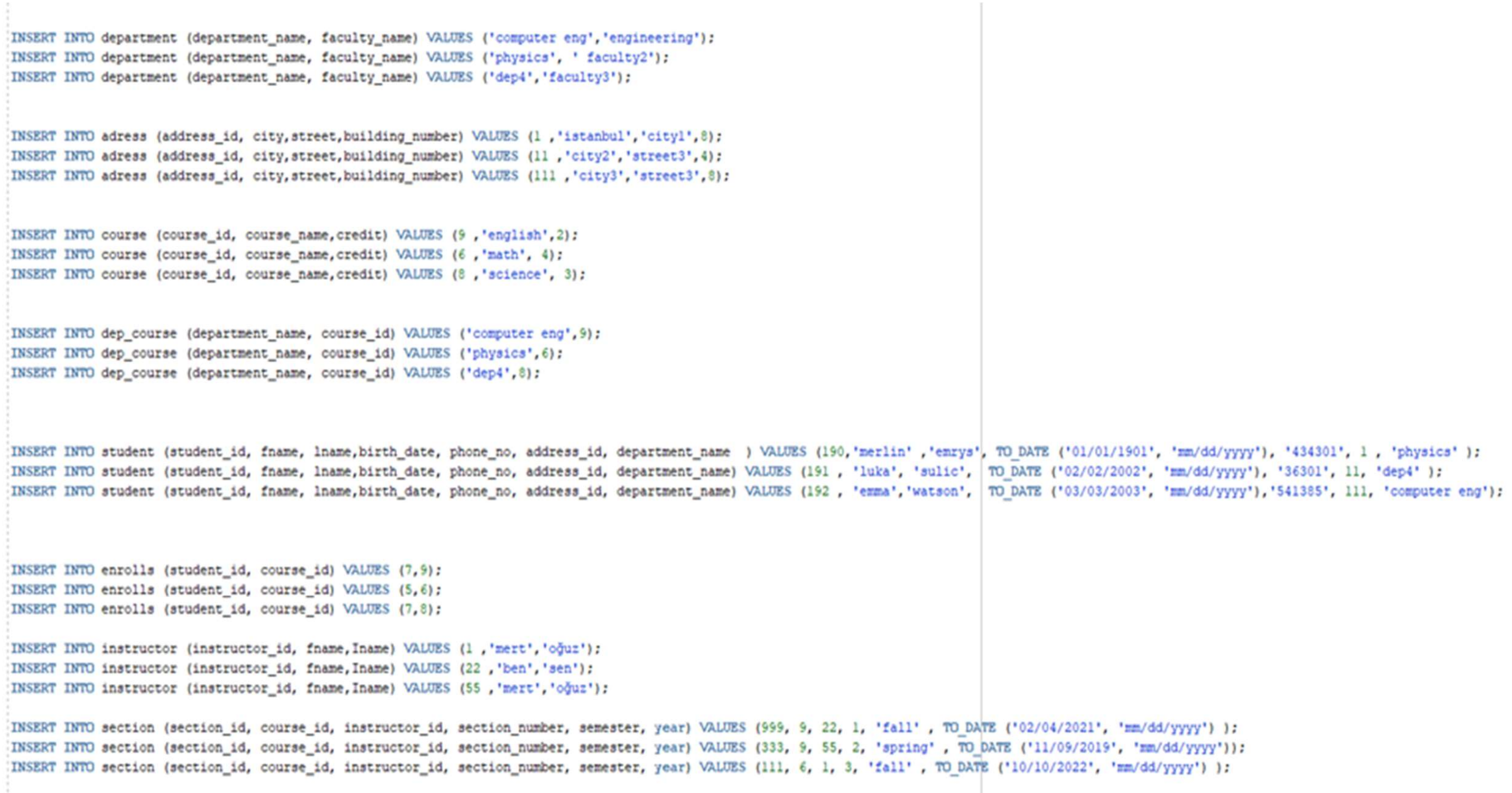


```
Table DEPARTMENT created.


Table ADRESS created.


Table STUDENT created.


Table COURSE created.


Table ENROLLS created.


Table DEP_COURSE created.


Table INSTRUCTOR created.


Table SECTION created.
```

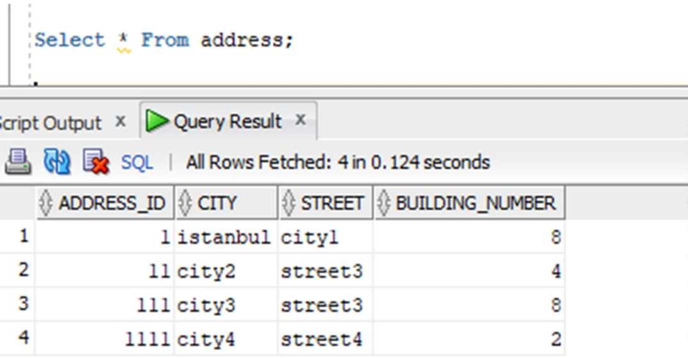
Insert Into Statements

```
INSERT INTO department (department_name, faculty_name) VALUES ('computer eng','engineering');
INSERT INTO department (department_name, faculty_name) VALUES ('physics', ' faculty2');
INSERT INTO department (department_name, faculty_name) VALUES ('dep4','faculty3');
INSERT INTO department (department_name, faculty_name) VALUES ('dep3','faculty3');

INSERT INTO address (address_id, city,street,building_number) VALUES (1 ,'istanbul','city1',8);
INSERT INTO address (address_id, city,street,building_number) VALUES (11 ,'city2','street3',4);
INSERT INTO address (address_id, city,street,building_number) VALUES (111 ,'city3','street3',8);
INSERT INTO address (address_id, city,street,building_number) VALUES (1111 ,'city4','street4',2);

INSERT INTO course (course_id, course_name,credit) VALUES (6 ,'english',2);
INSERT INTO course (course_id, course_name,credit) VALUES (7 ,'math', 4);
INSERT INTO course (course_id, course_name,credit) VALUES (8 ,'science', 3);
INSERT INTO course (course_id, course_name,credit) VALUES (9 ,'art', 3);

INSERT INTO dep_course (department_name, course_id) VALUES ('computer eng',9);
INSERT INTO dep_course (department_name, course_id) VALUES ('physics',6);
INSERT INTO dep_course (department_name, course_id) VALUES ('dep4',8);

INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name  )
VALUES (190,'merlin' ,'emrys', TO_DATE ('01/01/1901', 'mm/dd/yyyy'), '434301', 1 , 'physics' );
INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name)
VALUES (191 , 'luka', 'sulic',  TO_DATE ('02/02/2002', 'mm/dd/yyyy'), '36301', 11, 'dep4' );
INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name)
VALUES (192 , 'emma','watson',  TO_DATE ('03/03/2003', 'mm/dd/yyyy'),'541385', 111, 'computer
eng');
INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name  )
VALUES (193,'leonardo' ,'da vinci', TO_DATE ('01/21/1999', 'mm/dd/yyyy'), '434301', 111 , 'physics' );
```

```
INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name)
VALUES (194 , 'van', 'gogh',  TO_DATE ('12/02/2000', 'mm/dd/yyyy'), '36301', 11, 'dep3' );


INSERT INTO enrolls (student_id, course_id) VALUES (190,8);

INSERT INTO enrolls (student_id, course_id) VALUES (190,6);

INSERT INTO enrolls (student_id, course_id) VALUES (191,6);

INSERT INTO enrolls (student_id, course_id) VALUES (193,7);

INSERT INTO enrolls (student_id, course_id) VALUES (193,9);

INSERT INTO enrolls (student_id, course_id) VALUES (194,9);


INSERT INTO instructor (instructor_id, fname,Iname) VALUES (1 ,'mert','oğuz');

INSERT INTO instructor (instructor_id, fname,Iname) VALUES (22 ,'ben','sen');

INSERT INTO instructor (instructor_id, fname,Iname) VALUES (55 ,'mert','oğuz');

INSERT INTO instructor (instructor_id, fname,Iname) VALUES (77 ,'Tamer','Ucar');


INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES
(999, 9, 22, 1, 'fall' , TO_DATE ('02/04/2021', 'mm/dd/yyyy') );

INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES
(333, 9, 55, 2, 'spring' , TO_DATE ('11/09/2019', 'mm/dd/yyyy'));

INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES
(111, 6, 1, 3, 'fall' , TO_DATE ('10/10/2022', 'mm/dd/yyyy') );

INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES
(444, 7, 77, 3, 'fall' , TO_DATE ('11/05/2022', 'mm/dd/yyyy') );

INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES
(777, 6, 77, 4, 'spring' , TO_DATE ('12/08/2020', 'mm/dd/yyyy') );
```



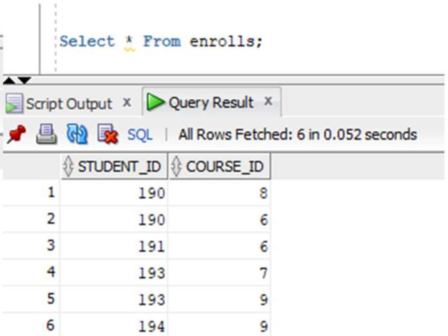
```sql
INSERT INTO department (department_name, faculty_name) VALUES ('computer eng','engineering');
INSERT INTO department (department_name, faculty_name) VALUES ('physics', ' faculty2');
INSERT INTO department (department_name, faculty_name) VALUES ('dep4','faculty3');


INSERT INTO adress (address_id, city,street,building_number) VALUES (1 ,'istanbul','city1',8);
INSERT INTO adress (address_id, city,street,building_number) VALUES (11 ,'city2','street3',4);
INSERT INTO adress (address_id, city,street,building_number) VALUES (111 ,'city3','street3',8);


INSERT INTO course (course_id, course_name,credit) VALUES (9 ,'english',2);
INSERT INTO course (course_id, course_name,credit) VALUES (6 ,'math', 4);
INSERT INTO course (course_id, course_name,credit) VALUES (8 ,'science', 3);


INSERT INTO dep_course (department_name, course_id) VALUES ('computer eng',9);
INSERT INTO dep_course (department_name, course_id) VALUES ('physics',6);
INSERT INTO dep_course (department_name, course_id) VALUES ('dep4',8);



INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name  ) VALUES (190,'merlin' ,'emrys', TO_DATE ('01/01/1901', 'mm/dd/yyyy'), '434301', 1 , 'physics' );
INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name) VALUES (191 , 'luka', 'sulic',  TO_DATE ('02/02/2002', 'mm/dd/yyyy'), '36301', 11, 'dep4' );
INSERT INTO student (student_id, fname, lname,birth_date, phone_no, address_id, department_name) VALUES (192 , 'emma','watson',  TO_DATE ('03/03/2003', 'mm/dd/yyyy'),'541385', 111, 'computer eng');



INSERT INTO enrolls (student_id, course_id) VALUES (7,9);
INSERT INTO enrolls (student_id, course_id) VALUES (5,6);
INSERT INTO enrolls (student_id, course_id) VALUES (7,8);

INSERT INTO instructor (instructor_id, fname,Iname) VALUES (1 ,'mert','oğuz');
INSERT INTO instructor (instructor_id, fname,Iname) VALUES (22 ,'ben','sen');
INSERT INTO instructor (instructor_id, fname,Iname) VALUES (55 ,'mert','oğuz');

INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES (999, 9, 22, 1, 'fall' , TO_DATE ('02/04/2021', 'mm/dd/yyyy') );
INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES (333, 9, 55, 2, 'spring' , TO_DATE ('11/09/2019', 'mm/dd/yyyy'));
INSERT INTO section (section_id, course_id, instructor_id, section_number, semester, year) VALUES (111, 6, 1, 3, 'fall' , TO_DATE ('10/10/2022', 'mm/dd/yyyy') );
```



```sql
Select * From address;
```

All Rows Fetched: 4 in 0.124 seconds

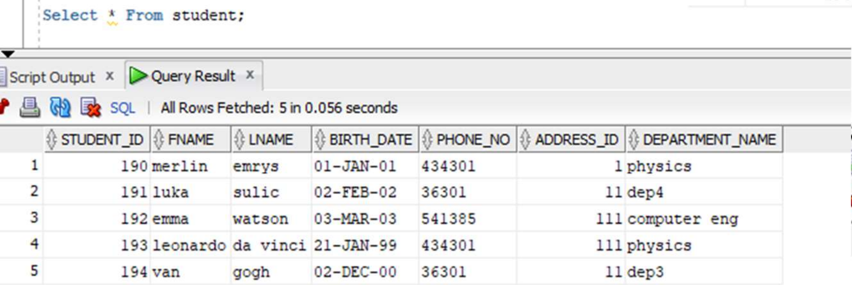
	ADDRESS_ID	CITY	STREET	BUILDING_NUMBER
1	1	istanbul	city1	8
2	11	city2	street3	4
3	111	city3	street3	8
4	1111	city4	street4	2



```sql
Select * From enrolls;
```

All Rows Fetched: 6 in 0.052 seconds

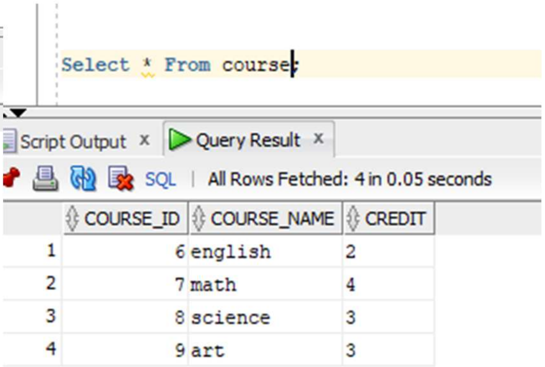
	STUDENT_ID	COURSE_ID
1	190	8
2	190	6
3	191	6
4	193	7
5	193	9
6	194	9



```sql
Select * From student;
```

All Rows Fetched: 5 in 0.056 seconds

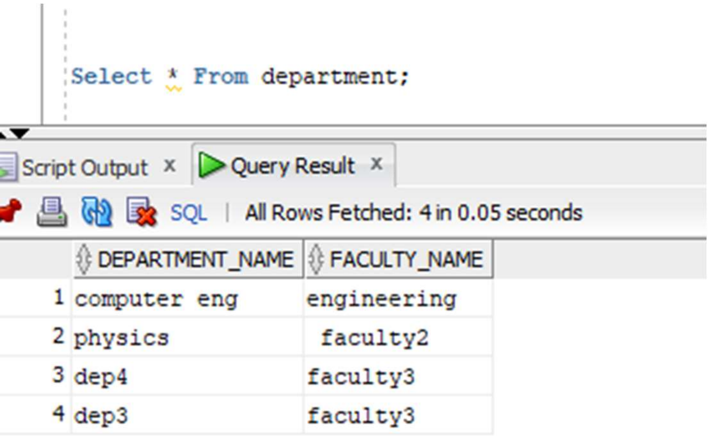
	STUDENT_ID	FNAME	LNAME	BIRTH_DATE	PHONE_NO	ADDRESS_ID	DEPARTMENT_NAME
1	190	merlin	emrys	01-JAN-01	434301	1	physics
2	191	luka	sulic	02-FEB-02	36301	11	dep4
3	192	emma	watson	03-MAR-03	541385	111	computer eng
4	193	leonardo	da vinci	21-JAN-99	434301	111	physics
5	194	van	gogh	02-DEC-00	36301	11	dep3



```sql
Select * From course;
```

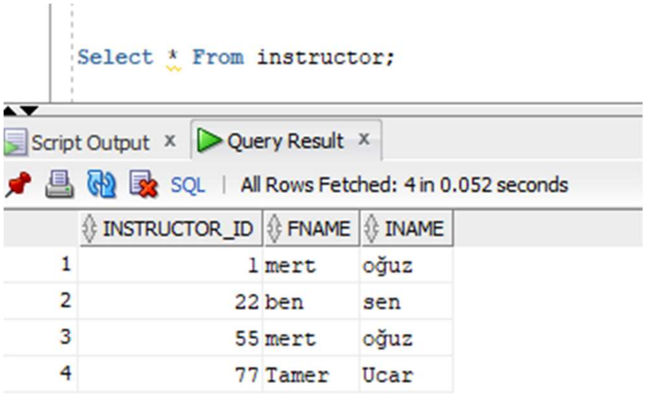
All Rows Fetched: 4 in 0.05 seconds

	COURSE_ID	COURSE_NAME	CREDIT
1	6	english	2
2	7	math	4
3	8	science	3
4	9	art	3



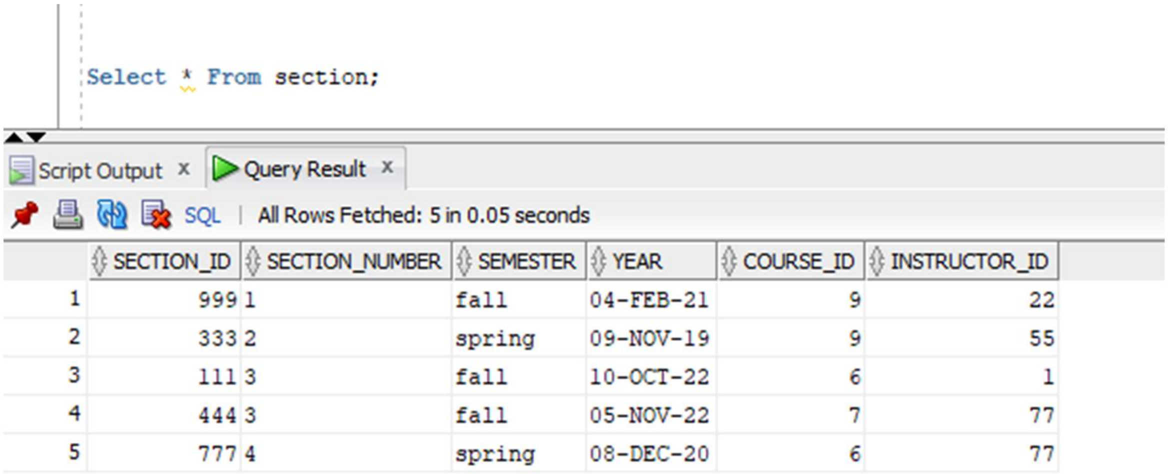
```
Select * From department;
```

	DEPARTMENT_NAME	FACULTY_NAME
1	computer eng	engineering
2	physics	faculty2
3	dep4	faculty3
4	dep3	faculty3



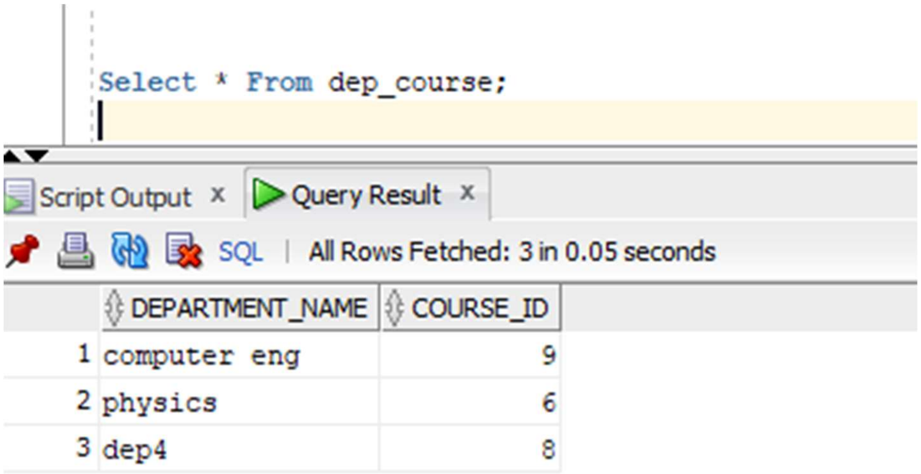
```
Select * From instructor;
```

	INSTRUCTOR_ID	FNAME	INAME
1	1	mert	oğuz
2	22	ben	sen
3	55	mert	oğuz
4	77	Tamer	Ucar



```
Select * From section;
```

	SECTION_ID	SECTION_NUMBER	SEMESTER	YEAR	COURSE_ID	INSTRUCTOR_ID
1	999	1	fall	04-FEB-21	9	22
2	333	2	spring	09-NOV-19	9	55
3	111	3	fall	10-OCT-22	6	1
4	444	3	fall	05-NOV-22	7	77
5	777	4	spring	08-DEC-20	6	77



```
Select * From dep_course;
```

	DEPARTMENT_NAME	COURSE_ID
1	computer eng	9
2	physics	6
3	dep4	8

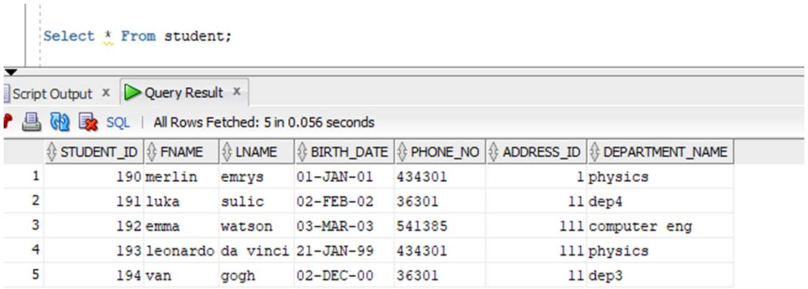
2 Joins

1_ join : Display the id, city and street of all students.

Select a.address_id, a.city, a.street

From student s, address a

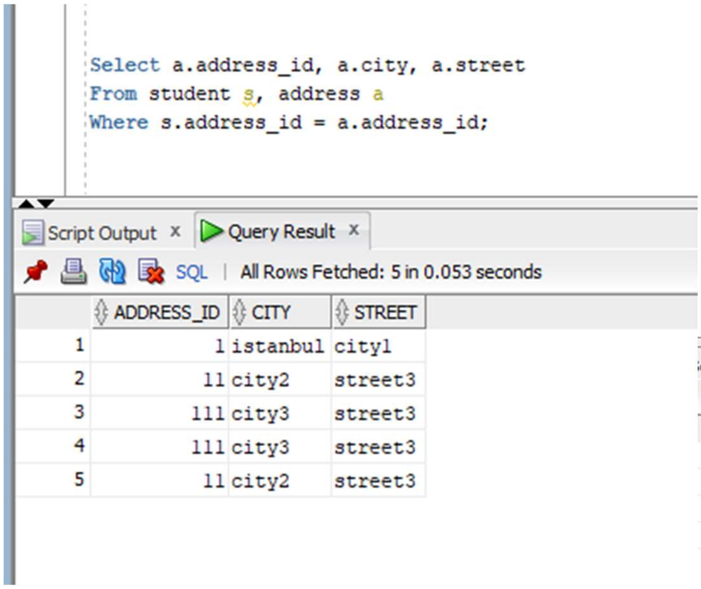
Where s.address_id = a.address_id;



```
Select * From student;
```

All Rows Fetched: 5 in 0.056 seconds

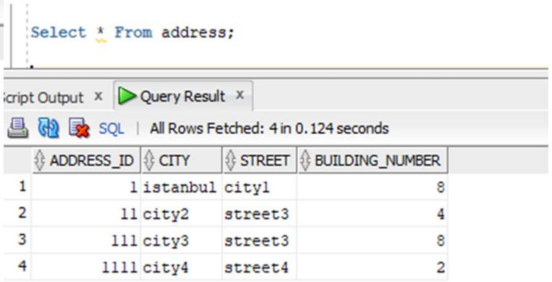
	STUDENT_ID	FNAME	LNAME	BIRTH_DATE	PHONE_NO	ADDRESS_ID	DEPARTMENT_NAME
1	190	merlin	emrys	01-JAN-01	434301	1	physics
2	191	luka	sulic	02-FEB-02	36301	11	dep4
3	192	emma	watson	03-MAR-03	541385	111	computer eng
4	193	leonardo	da vinci	21-JAN-99	434301	111	physics
5	194	van	gogh	02-DEC-00	36301	11	dep3



```
Select a.address_id, a.city, a.street
From student s, address a
Where s.address_id = a.address_id;
```

All Rows Fetched: 5 in 0.053 seconds

	ADDRESS_ID	CITY	STREET
1	1	istanbul	city1
2	11	city2	street3
3	111	city3	street3
4	111	city3	street3
5	11	city2	street3



```
Select * From address;
```

All Rows Fetched: 4 in 0.124 seconds

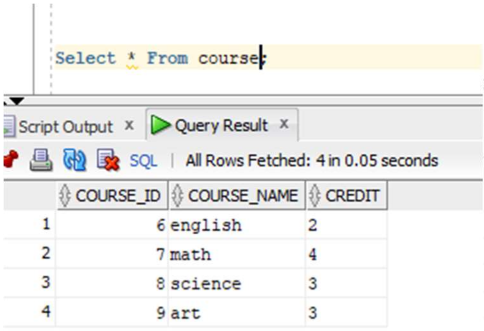
	ADDRESS_ID	CITY	STREET	BUILDING_NUMBER
1	1	istanbul	city1	8
2	11	city2	street3	4
3	111	city3	street3	8
4	1111	city4	street4	2

2_ join : Find the id of students who takes 'english' course.

Select distinct e.student_id

From enrolls e, course c

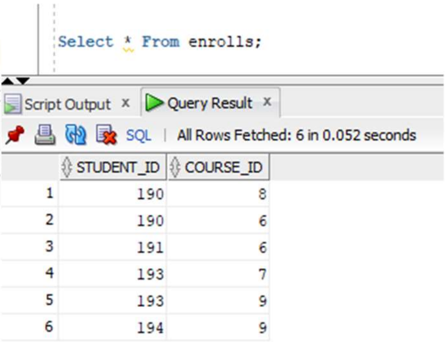
Where c.course_id = e.course_id AND LOWER(c.course_name) = 'english';



```
Select * From course;
```

All Rows Fetched: 4 in 0.05 seconds

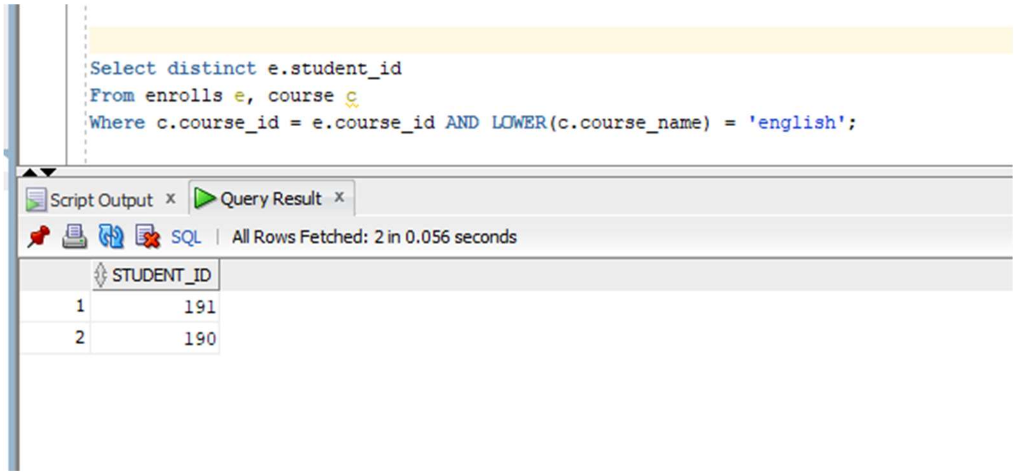
	COURSE_ID	COURSE_NAME	CREDIT
1	6	english	2
2	7	math	4
3	8	science	3
4	9	art	3



```
Select * From enrolls;
```

All Rows Fetched: 6 in 0.052 seconds

	STUDENT_ID	COURSE_ID
1	190	8
2	190	6
3	191	6
4	193	7
5	193	9
6	194	9



```
Select distinct e.student_id
From enrolls e, course c
Where c.course_id = e.course_id AND LOWER(c.course_name) = 'english';
```

All Rows Fetched: 2 in 0.056 seconds

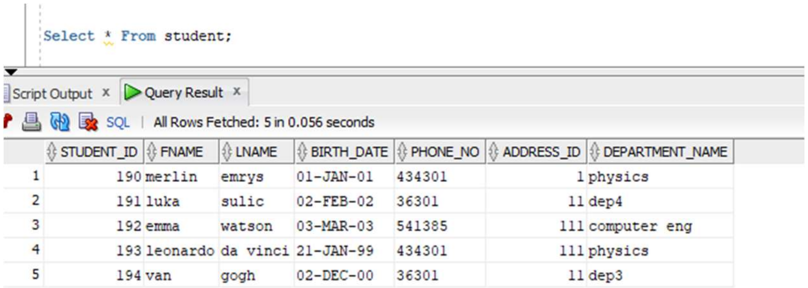
	STUDENT_ID
1	191
2	190

2 Nested Queries

3_(single row subquery): Find the names of students who studies in ‘physics’ department.

Select fname, lname From student Where department_name =

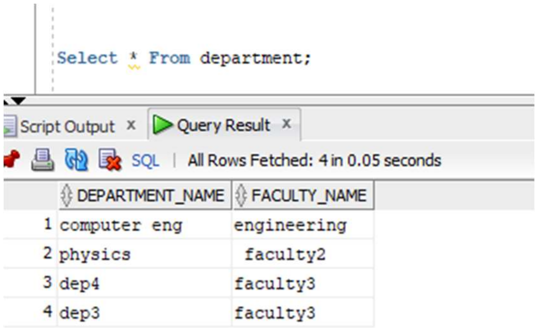
(Select department_name From department Where LOWER(department_name) = 'physics');



```
Select * From student;
```

All Rows Fetched: 5 in 0.056 seconds

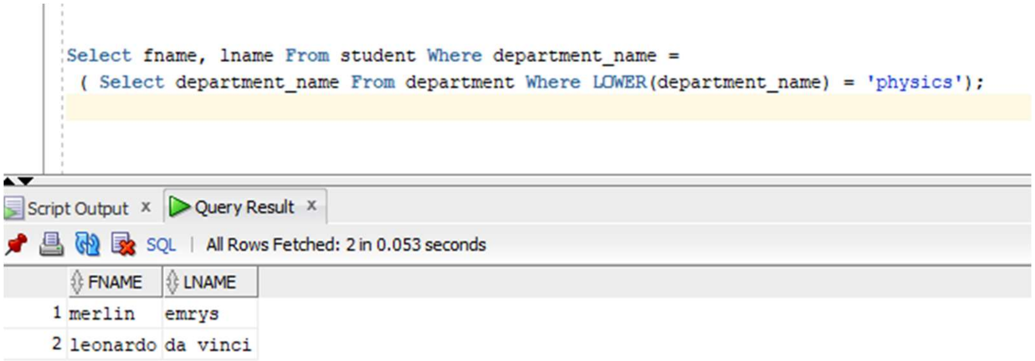
	STUDENT_ID	FNAME	LNAME	BIRTH_DATE	PHONE_NO	ADDRESS_ID	DEPARTMENT_NAME
1	190	merlin	emrys	01-JAN-01	434301	1	physics
2	191	luka	sulic	02-FEB-02	36301	11	dep4
3	192	emma	watson	03-MAR-03	541385	111	computer eng
4	193	leonardo	da vinci	21-JAN-99	434301	111	physics
5	194	van	gogh	02-DEC-00	36301	11	dep3



```
Select * From department;
```

All Rows Fetched: 4 in 0.05 seconds

	DEPARTMENT_NAME	FACULTY_NAME
1	computer eng	engineering
2	physics	faculty2
3	dep4	faculty3
4	dep3	faculty3



```
Select fname, lname From student Where department_name =
 ( Select department_name From department Where LOWER(department_name) = 'physics');
```

All Rows Fetched: 2 in 0.053 seconds

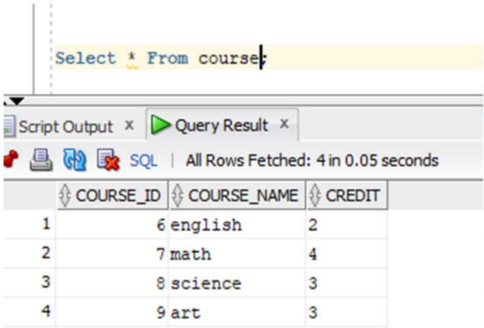
	FNAME	LNAME
1	merlin	emrys
2	leonardo	da vinci

4_(multiple row subquery): What is the names of courses which are teached by instructer ‘Tamer Ucar’ ?

Select course_name

From course Where course_id IN (Select course_id From instructor i, section s

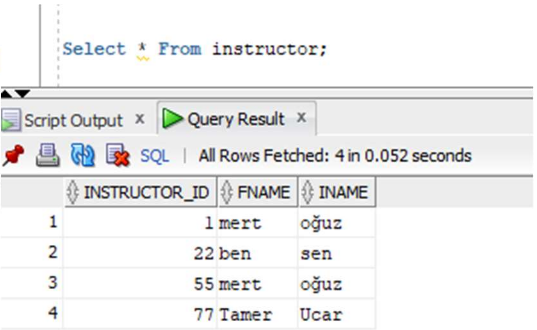
Where LOWER(i.fname) = 'tamer' AND LOWER(i.Iname) = 'ucar' AND i.instructor_id = s.instructor_id);



```
Select * From course;
```

All Rows Fetched: 4 in 0.05 seconds

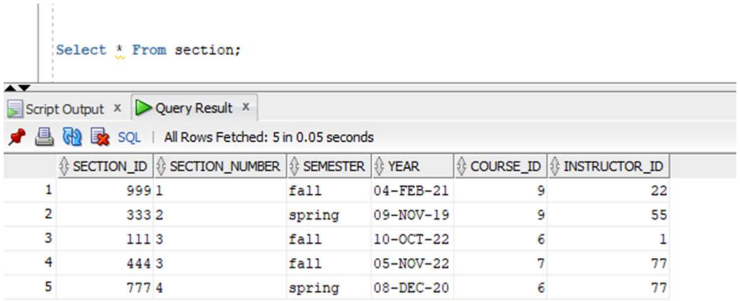
	COURSE_ID	COURSE_NAME	CREDIT
1	6	english	2
2	7	math	4
3	8	science	3
4	9	art	3



```
Select * From instructor;
```

All Rows Fetched: 4 in 0.052 seconds

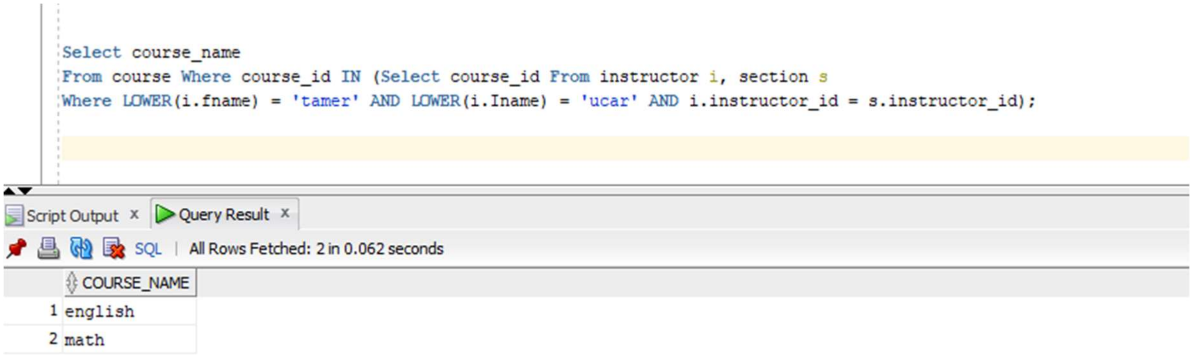
	INSTRUCTOR_ID	FNAME	INAME
1	1	mert	oğuz
2	22	ben	sen
3	55	mert	oğuz
4	77	Tamer	Ucar



```
Select * From section;
```

All Rows Fetched: 5 in 0.05 seconds

	SECTION_ID	SECTION_NUMBER	SEMESTER	YEAR	COURSE_ID	INSTRUCTOR_ID
1	999	1	fall	04-FEB-21	9	22
2	333	2	spring	09-NOV-19	9	55
3	111	3	fall	10-OCT-22	6	1
4	444	3	fall	05-NOV-22	7	77
5	777	4	spring	08-DEC-20	6	77



```
Select course_name
From course Where course_id IN (Select course_id From instructor i, section s
Where LOWER(i.fname) = 'tamer' AND LOWER(i.Iname) = 'ucar' AND i.instructor_id = s.instructor_id);
```

All Rows Fetched: 2 in 0.062 seconds

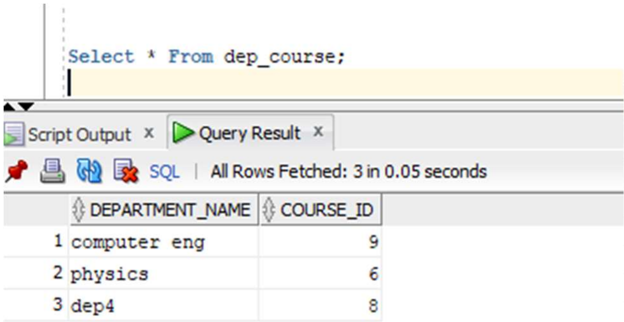
	COURSE_NAME
1	english
2	math

2 Set Operations

5_set: Findthe students who are taking courses which courses and students in the same department.

Select course_id From enrolls INTERSECT

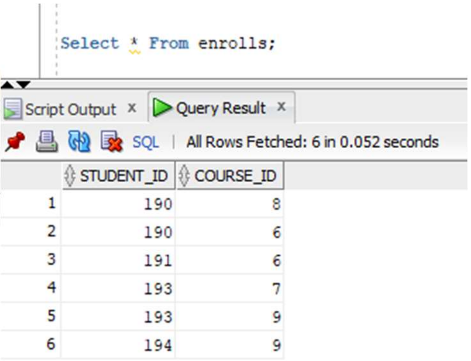
Select course_id From dep_course ;



```
Select * From dep_course;
```

All Rows Fetched: 3 in 0.05 seconds

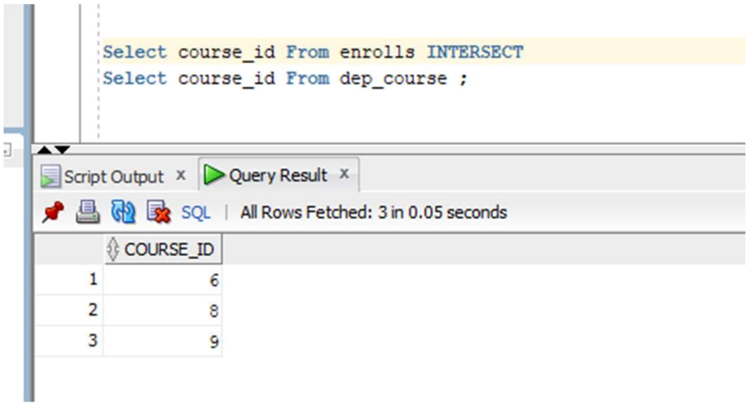
	DEPARTMENT_NAME	COURSE_ID
1	computer eng	9
2	physics	6
3	dep4	8



```
Select * From enrolls;
```

All Rows Fetched: 6 in 0.052 seconds

	STUDENT_ID	COURSE_ID
1	190	8
2	190	6
3	191	6
4	193	7
5	193	9
6	194	9



```
Select course_id From enrolls INTERSECT
Select course_id From dep_course ;
```

All Rows Fetched: 3 in 0.05 seconds

	COURSE_ID
1	6
2	8
3	9

6_set: Find the student names who takes courses with id of : 9 and 6.

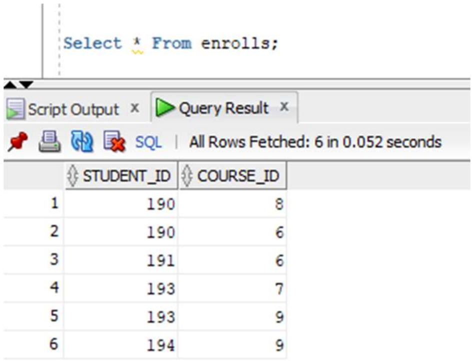
Select s.fname, s.lname

From enrolls e, student s Where e.student_id = s.student_id AND e.course_id = 6

UNION

Select s.fname, s.lname

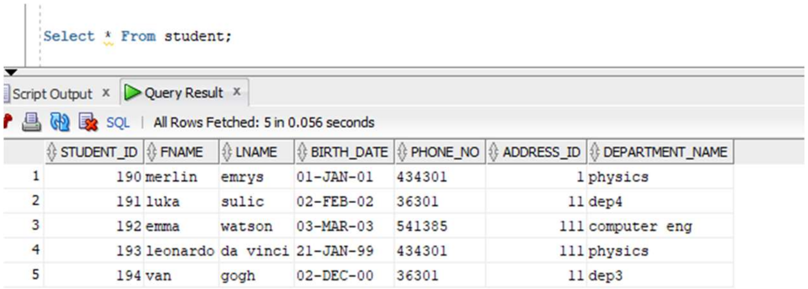
From enrolls e, student s Where e.student_id = s.student_id AND e.course_id = 9;



```
Select * From enrolls;
```

All Rows Fetched: 6 in 0.052 seconds

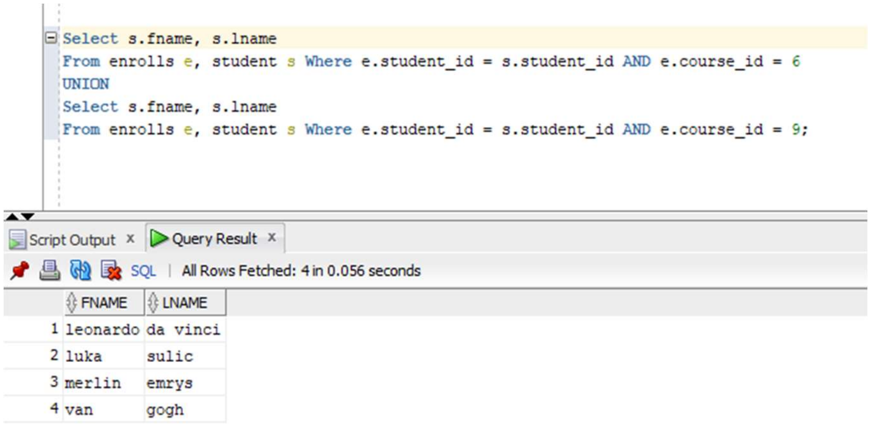
	STUDENT_ID	COURSE_ID
1	190	8
2	190	6
3	191	6
4	193	7
5	193	9
6	194	9



```
Select * From student;
```

All Rows Fetched: 5 in 0.056 seconds

	STUDENT_ID	FNAME	LNAME	BIRTH_DATE	PHONE_NO	ADDRESS_ID	DEPARTMENT_NAME
1	190	merlin	emrys	01-JAN-01	434301	1	physics
2	191	luka	sulic	02-FEB-02	36301	11	dep4
3	192	emma	watson	03-MAR-03	541385	111	computer eng
4	193	leonardo	da vinci	21-JAN-99	434301	111	physics
5	194	van	gogh	02-DEC-00	36301	11	dep3



```
Select s.fname, s.lname
From enrolls e, student s Where e.student_id = s.student_id AND e.course_id = 6
UNION
Select s.fname, s.lname
From enrolls e, student s Where e.student_id = s.student_id AND e.course_id = 9;
```

All Rows Fetched: 4 in 0.056 seconds

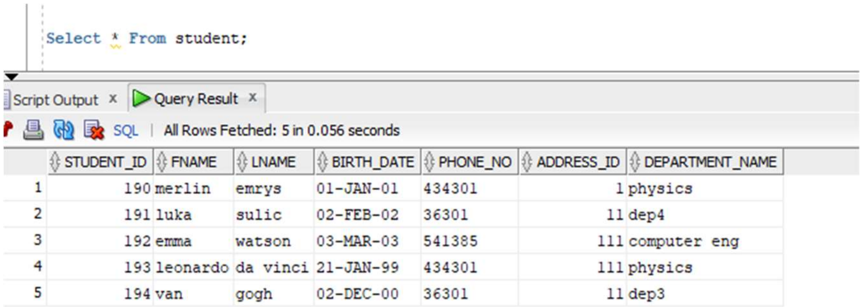
	FNAME	LNAME
1	leonardo	da vinci
2	luka	sulic
3	merlin	emrys
4	van	gogh

2 Aggregate Operations

7_ aggregate: Count the total number of students who is in ‘faculty3’.

Select count(s.student_id) as "Student Count" from student s, department d

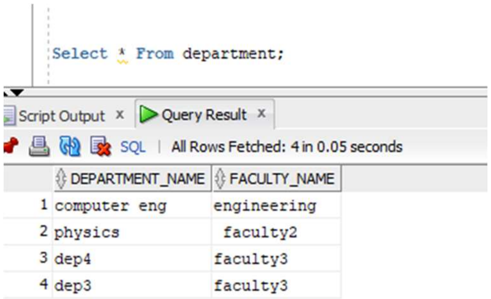
Where s.department_name = d.department_name AND LOWER(d.faculty_name) = 'faculty3';



```
Select * From student;
```

All Rows Fetched: 5 in 0.056 seconds

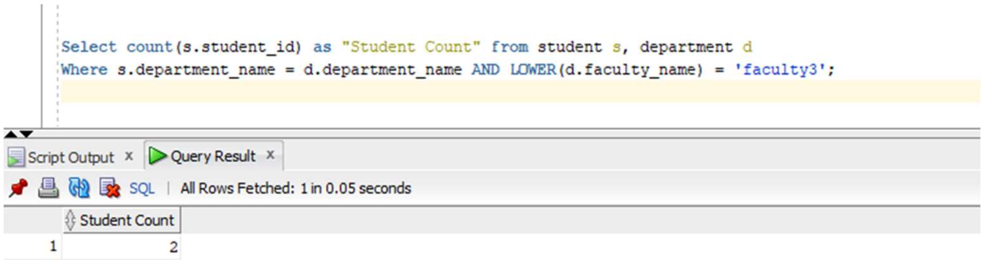
	STUDENT_ID	FNAME	LNAME	BIRTH_DATE	PHONE_NO	ADDRESS_ID	DEPARTMENT_NAME
1	190	merlin	emrys	01-JAN-01	434301	1	physics
2	191	luka	sulic	02-FEB-02	36301	11	dep4
3	192	emma	watson	03-MAR-03	541385	111	computer eng
4	193	leonardo	da vinci	21-JAN-99	434301	111	physics
5	194	van	gogh	02-DEC-00	36301	11	dep3



```
Select * From department;
```

All Rows Fetched: 4 in 0.05 seconds

	DEPARTMENT_NAME	FACULTY_NAME
1	computer eng	engineering
2	physics	faculty2
3	dep4	faculty3
4	dep3	faculty3



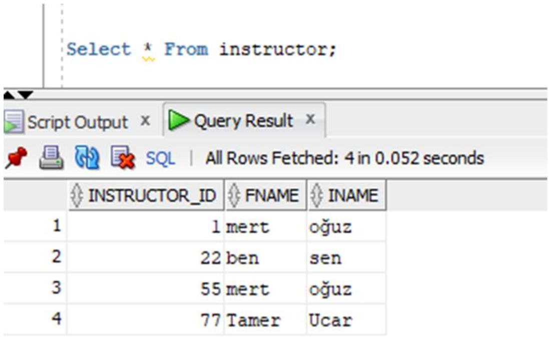
```
Select count(s.student_id) as "Student Count" from student s, department d
Where s.department_name = d.department_name AND LOWER(d.faculty_name) = 'faculty3';
```

All Rows Fetched: 1 in 0.05 seconds

	Student Count
1	2

8_ aggregate: Count the total number of intructors.

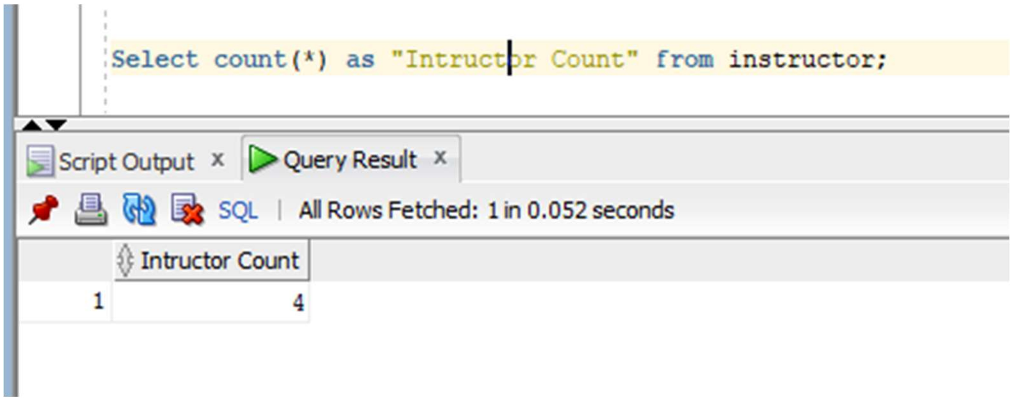
Select count(*) as "Intructor Count" from instructor;



```
Select * From instructor;
```

All Rows Fetched: 4 in 0.052 seconds

	INSTRUCTOR_ID	FNAME	INAME
1	1	mert	oğuz
2	22	ben	sen
3	55	mert	oğuz
4	77	Tamer	Ucar



```
Select count(*) as "Intructor Count" from instructor;
```

All Rows Fetched: 1 in 0.052 seconds

	Intructor Count
1	4