Information Management
Assignment 2

Pratik Gawli (pbg397)

The table structure contains the following info:
StudId is an integer Primary Key of 6 digits
StdFirstName has up to 30 characters and is not allowed to be empty
StdLastName has up to 30 characters and is allowed to be empty
TotalScore is a percentage number between 0 and 100, both inclusive expressed in up to two fractional digits
CourseName has up to 30 characters and can be NULL
Section can be either A or B
Stream can be either Accounting, Finance, IB, Marketing, MIS, or Analytics that the student belongs to.

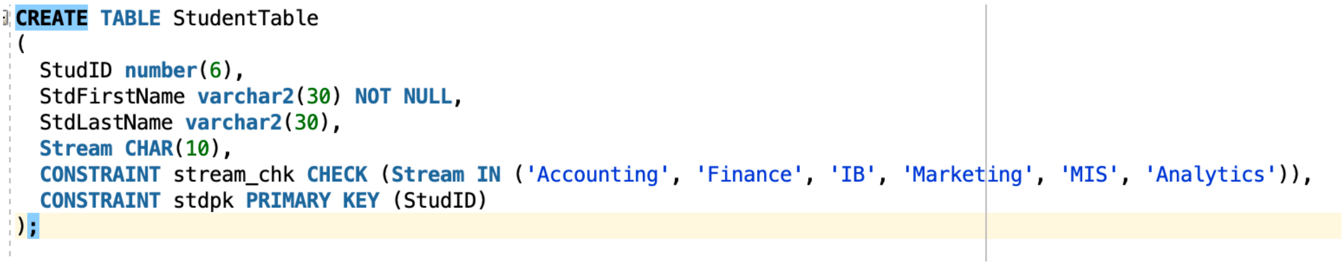
The following data has to be entered in the database:

The data to be entered is as follows

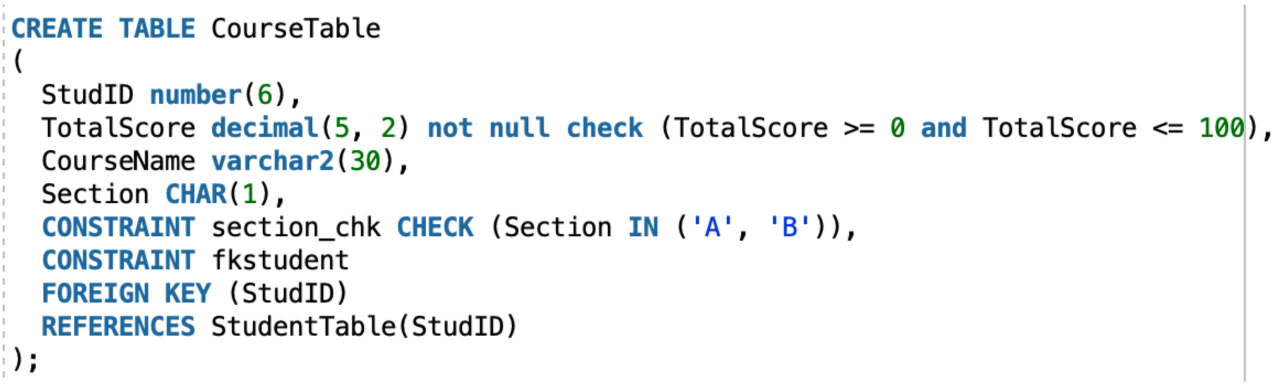
StudID	**StdFirstName**	**StdLastName**	**TotalScore**	**CourseName**	**Section**	**Stream**
135791	Albert	Einstein	99.98	Physics	A	Accounting
246802	Homi	Bhabha	99.99	Physics	B	Finance
147036	Marie	Daly	100	Chemistry	A	IB
260482	Srinivasa	Ramanuja	17.29	Math	A	Analytics
161616	Marie	Curie	88	Chemistry	B	Analytics
271828	Vikram	Sarabhai	19.19	Astronomy	A	MIS
314159	Chien	Wu	19.12	Physics	A	Marketing
314159	Chien	Wu	100	Chemistry	B	Marketing
135791	Albert	Einstein	75	Chemistry	A	Accounting

246802	Homi	Bhabha	48	Math	A	Finance
147036	Marie	Daly	67	Math	A	IB
260482	Srinivasa	Ramanuja	92.71	Chemistry	A	Analytics
161616	Marie	Curie	88.88	Astronomy	B	Analytics
271828	Vikram	Sarabhai	91.91	Physics	A	MIS
314159	Chien	Wu	91.21	Math	A	Marketing

To store the above information optimally, I’ve created two tables, StudentTable and CourseTable as below:

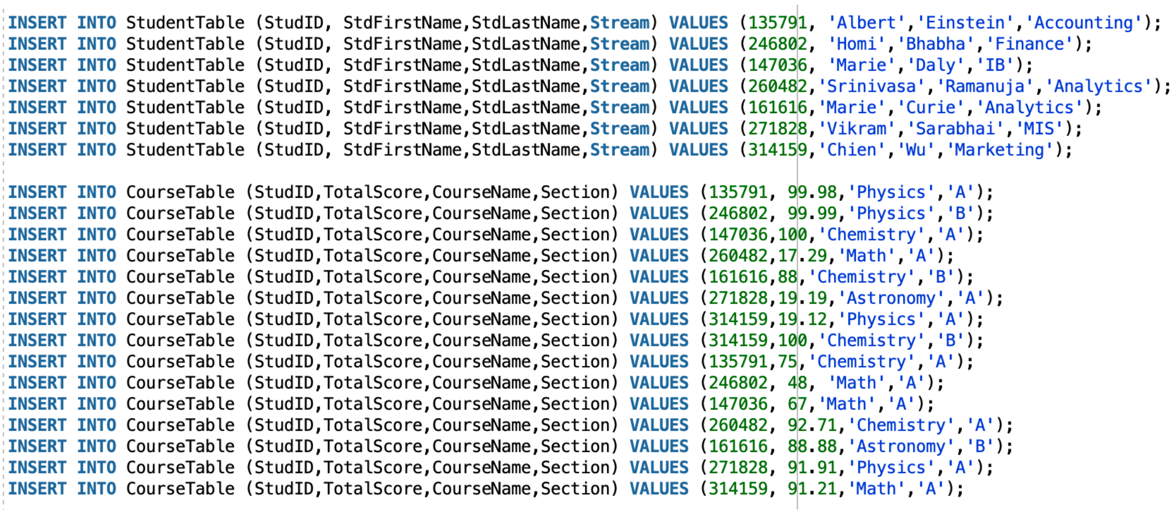


```
CREATE TABLE StudentTable
(
  StudID number(6),
  StdFirstName varchar2(30) NOT NULL,
  StdLastName varchar2(30),
  Stream CHAR(10),
  CONSTRAINT stream_chk CHECK (Stream IN ('Accounting', 'Finance', 'IB', 'Marketing', 'MIS', 'Analytics')),
  CONSTRAINT stdpk PRIMARY KEY (StudID)
);
```



```
CREATE TABLE CourseTable
(
  StudID number(6),
  TotalScore decimal(5, 2) not null check (TotalScore >= 0 and TotalScore <= 100),
  CourseName varchar2(30),
  Section CHAR(1),
  CONSTRAINT section_chk CHECK (Section IN ('A', 'B')),
  CONSTRAINT fkstudent
  FOREIGN KEY (StudID)
  REFERENCES StudentTable(StudID)
);
```

And the data was inserted into the tables using insert statements as below:

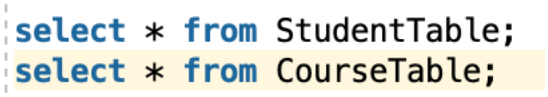


```
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (135791, 'Albert','Einstein','Accounting');
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (246802, 'Homi','Bhabha','Finance');
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (147036, 'Marie','Daly','IB');
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (260482,'Srinivasa','Ramanuja','Analytics');
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (161616,'Marie','Curie','Analytics');
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (271828,'Vikram','Sarabhai','MIS');
INSERT INTO StudentTable (StudID, StdFirstName,StdLastName,Stream) VALUES (314159,'Chien','Wu','Marketing');

INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (135791, 99.98,'Physics','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (246802, 99.99,'Physics','B');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (147036,100,'Chemistry','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (260482,17.29,'Math','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (161616,88,'Chemistry','B');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (271828,19.19,'Astronomy','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (314159,19.12,'Physics','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (314159,100,'Chemistry','B');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (135791,75,'Chemistry','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (246802, 48, 'Math','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (147036, 67,'Math','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (260482, 92.71,'Chemistry','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (161616, 88.88,'Astronomy','B');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (271828, 91.91,'Physics','A');
INSERT INTO CourseTable (StudID,TotalScore,CourseName,Section) VALUES (314159, 91.21,'Math','A');
```

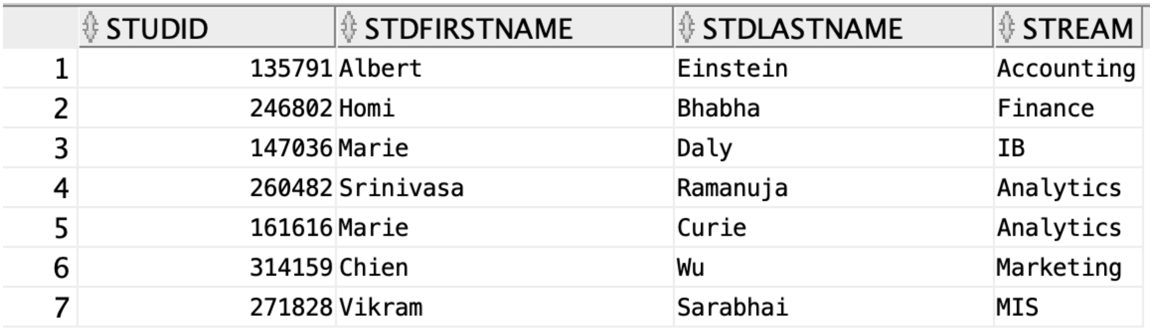
All the constraints are taken care of using the checks created at the ‘create table’ command.

Question 1: Show all the records from (all the) table(s) you have created



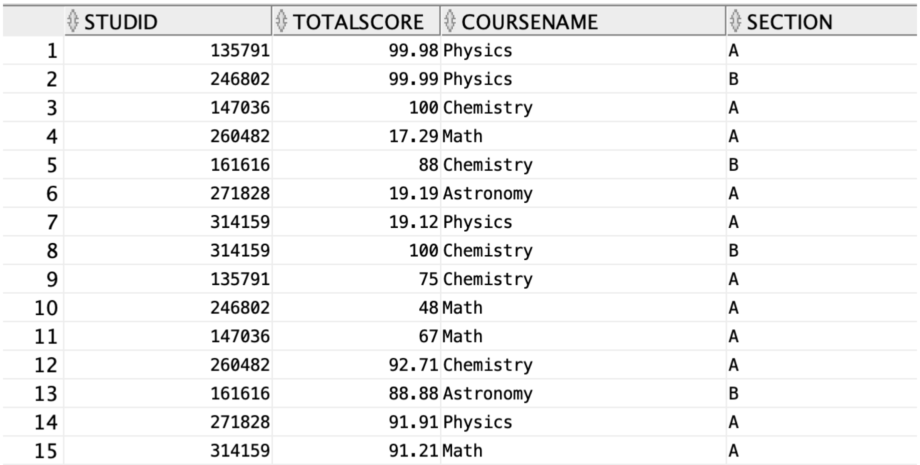
```
select * from StudentTable;
select * from CourseTable;
```

StudentTable:



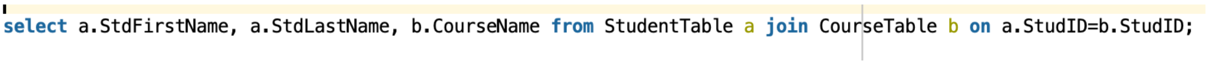
	STUDID	STDFIRSTNAME	STDLASTNAME	STREAM
1	135791	Albert	Einstein	Accounting
2	246802	Homi	Bhabha	Finance
3	147036	Marie	Daly	IB
4	260482	Srinivasa	Ramanuja	Analytics
5	161616	Marie	Curie	Analytics
6	314159	Chien	Wu	Marketing
7	271828	Vikram	Sarabhai	MIS

CourseTable:



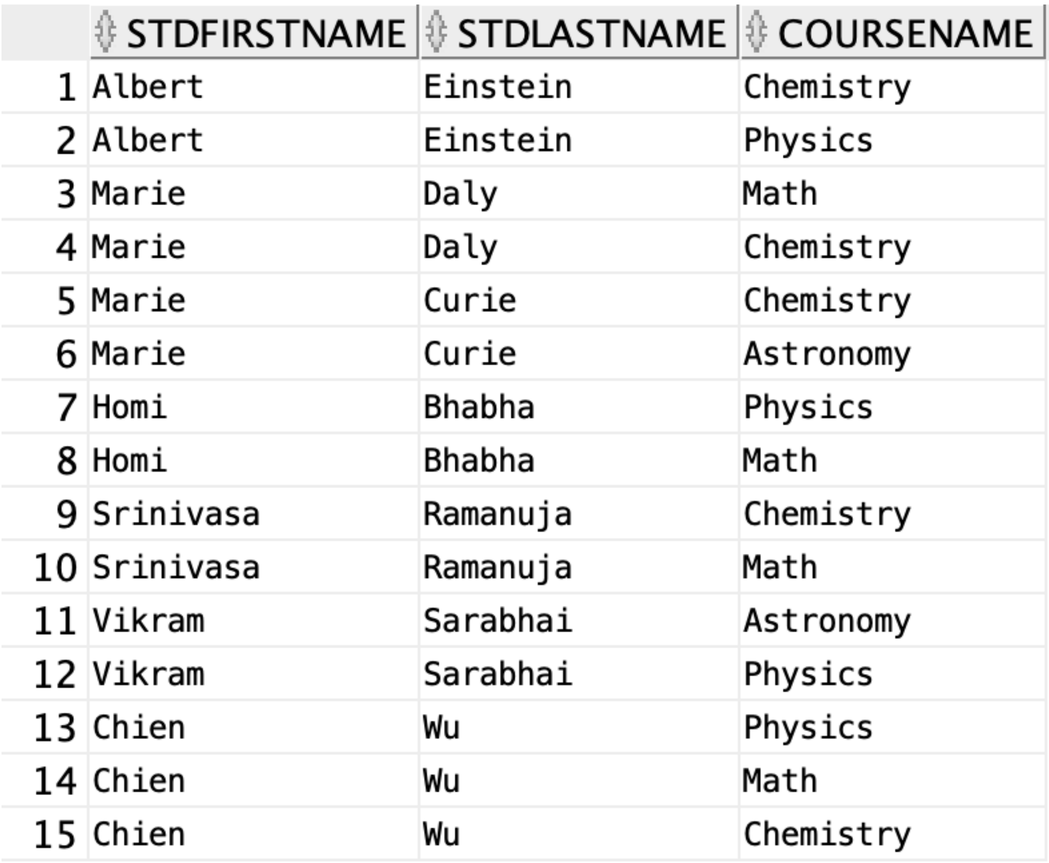
	STUDID	TOTALSCORE	COURSENAME	SECTION
1	135791	99.98	Physics	A
2	246802	99.99	Physics	B
3	147036	100	Chemistry	A
4	260482	17.29	Math	A
5	161616	88	Chemistry	B
6	271828	19.19	Astronomy	A
7	314159	19.12	Physics	A
8	314159	100	Chemistry	B
9	135791	75	Chemistry	A
10	246802	48	Math	A
11	147036	67	Math	A
12	260482	92.71	Chemistry	A
13	161616	88.88	Astronomy	B
14	271828	91.91	Physics	A
15	314159	91.21	Math	A

Question 2: Display only the first and last names, and courses each student is enrolled in



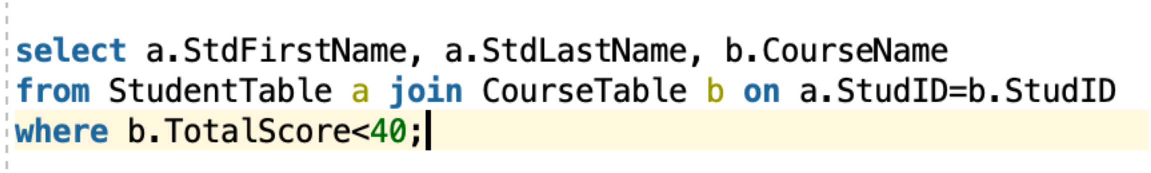
```
select a.StdFirstName, a.StdLastName, b.CourseName from StudentTable a join CourseTable b on a.StudID=b.StudID;
```

Output:



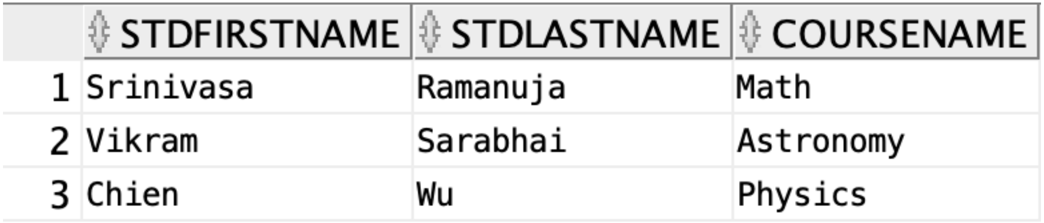
	STDFIRSTNAME	STDLASTNAME	COURSENAME
1	Albert	Einstein	Chemistry
2	Albert	Einstein	Physics
3	Marie	Daly	Math
4	Marie	Daly	Chemistry
5	Marie	Curie	Chemistry
6	Marie	Curie	Astronomy
7	Homi	Bhabha	Physics
8	Homi	Bhabha	Math
9	Srinivasa	Ramanuja	Chemistry
10	Srinivasa	Ramanuja	Math
11	Vikram	Sarabhai	Astronomy
12	Vikram	Sarabhai	Physics
13	Chien	Wu	Physics
14	Chien	Wu	Math
15	Chien	Wu	Chemistry

Question 3: Which students are failing in which classes, where the failing grade is 40%?



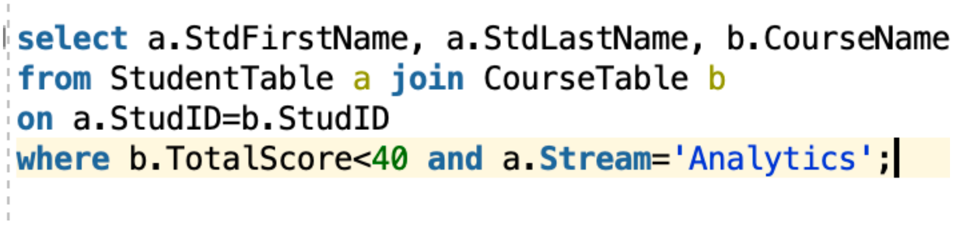
```
select a.StdFirstName, a.StdLastName, b.CourseName
from StudentTable a join CourseTable b on a.StudID=b.StudID
where b.TotalScore<40;
```

Output:



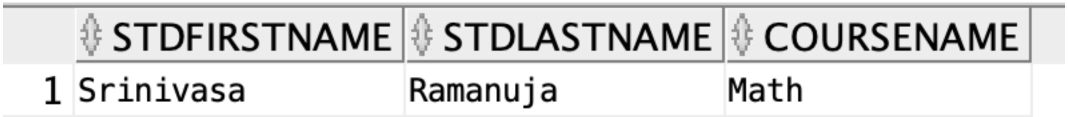
	STDFIRSTNAME	STDLASTNAME	COURSENAME
1	Srinivasa	Ramanuja	Math
2	Vikram	Sarabhai	Astronomy
3	Chien	Wu	Physics

Question 4: Which students from the Analytics stream are failing?



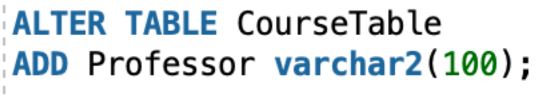
```
select a.StdFirstName, a.StdLastName, b.CourseName
from StudentTable a join CourseTable b
on a.StudID=b.StudID
where b.TotalScore<40 and a.Stream='Analytics';
```

Output:



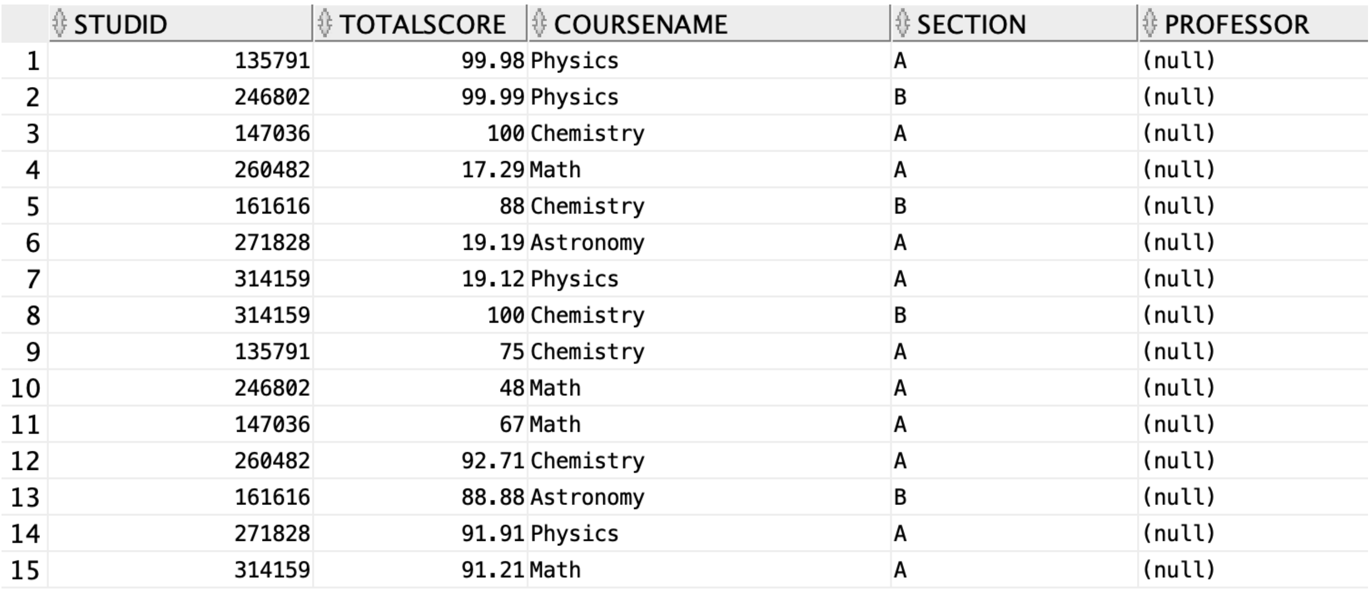
	STDFIRSTNAME	STDLASTNAME	COURSENAME
1	Srinivasa	Ramanuja	Math

Question 5: Now alter the table(s) by adding a Professor to each class being taught. Right now keep the professor name empty. Show the new table(s)



```
ALTER TABLE CourseTable
ADD Professor varchar2(100);
```

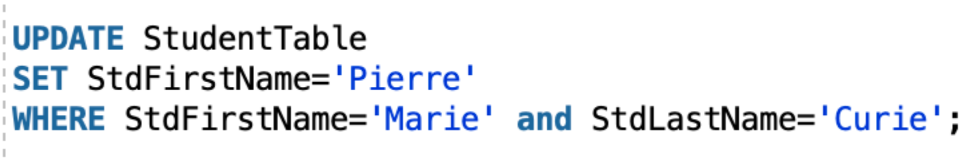
Output: (CourseTable appended with professor column)



	STUDID	TOTALSCORE	COURSENAME	SECTION	PROFESSOR
1	135791	99.98	Physics	A	(null)
2	246802	99.99	Physics	B	(null)
3	147036	100	Chemistry	A	(null)
4	260482	17.29	Math	A	(null)
5	161616	88	Chemistry	B	(null)
6	271828	19.19	Astronomy	A	(null)
7	314159	19.12	Physics	A	(null)
8	314159	100	Chemistry	B	(null)
9	135791	75	Chemistry	A	(null)
10	246802	48	Math	A	(null)
11	147036	67	Math	A	(null)
12	260482	92.71	Chemistry	A	(null)
13	161616	88.88	Astronomy	B	(null)
14	271828	91.91	Physics	A	(null)
15	314159	91.21	Math	A	(null)

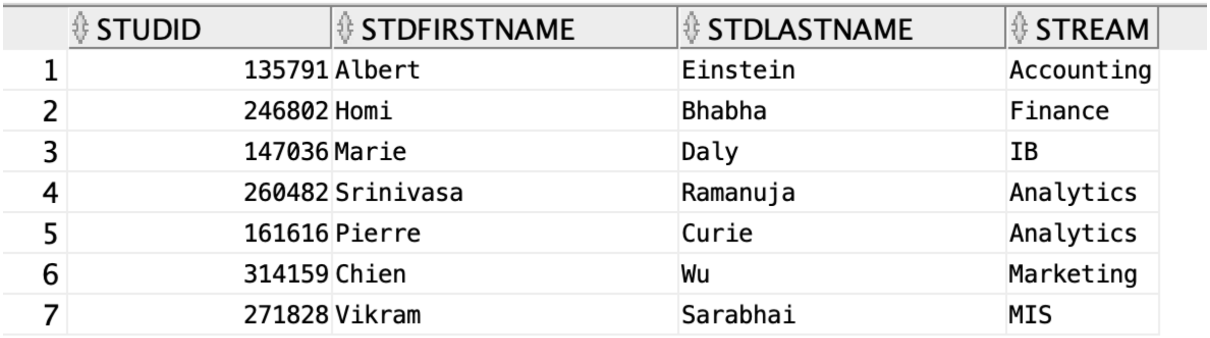
Question 6:

Change the student name ‘Marie Curie’ to ‘Pierre Curie’



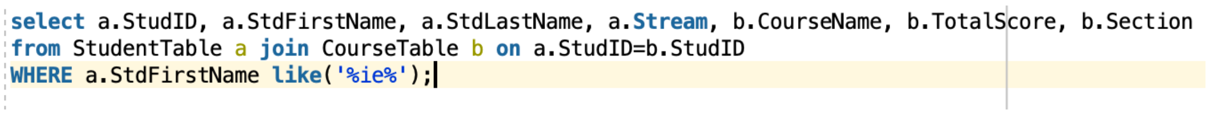
```
UPDATE StudentTable
SET StdFirstName='Pierre'
WHERE StdFirstName='Marie' and StdLastName='Curie';
```

Output



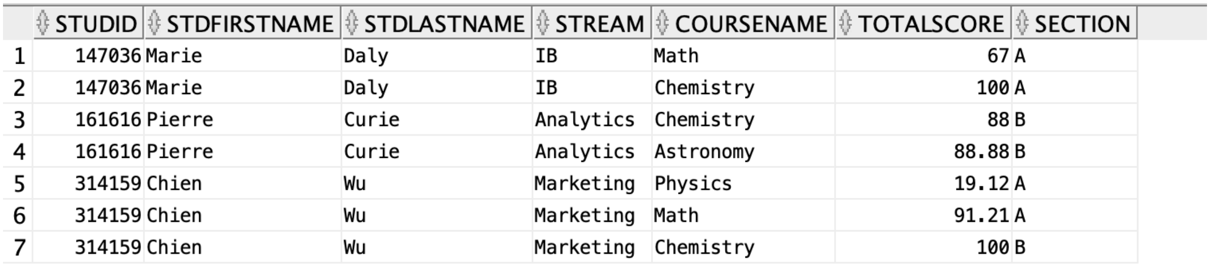
	STUDID	STDFIRSTNAME	STDLASTNAME	STREAM
1	135791	Albert	Einstein	Accounting
2	246802	Homi	Bhabha	Finance
3	147036	Marie	Daly	IB
4	260482	Srinivasa	Ramanuja	Analytics
5	161616	Pierre	Curie	Analytics
6	314159	Chien	Wu	Marketing
7	271828	Vikram	Sarabhai	MIS

Question 7: Display the full record for those students whose first name contains the regular expression ‘ie’. For example, the word **lie**d has the regular expression ‘**ie**’, while lai does not.



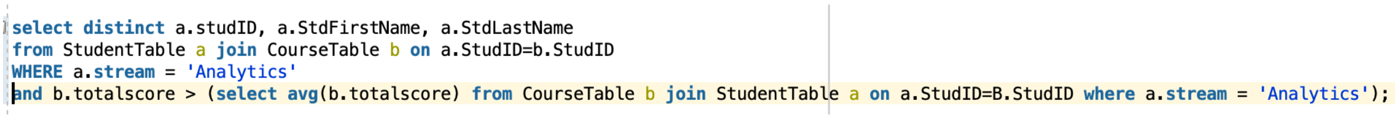
```
select a.StudID, a.StdFirstName, a.StdLastName, a.Stream, b.CourseName, b.TotalScore, b.Section
from StudentTable a join CourseTable b on a.StudID=b.StudID
WHERE a.StdFirstName like('%ie%');
```

Output:



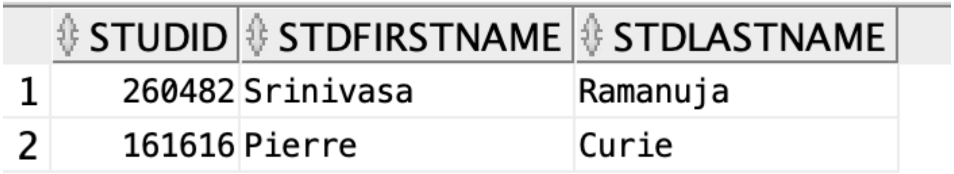
	STUDID	STDFIRSTNAME	STDLASTNAME	STREAM	COURSENAME	TOTALSCORE	SECTION
1	147036	Marie	Daly	IB	Math	67	A
2	147036	Marie	Daly	IB	Chemistry	100	A
3	161616	Pierre	Curie	Analytics	Chemistry	88	B
4	161616	Pierre	Curie	Analytics	Astronomy	88.88	B
5	314159	Chien	Wu	Marketing	Physics	19.12	A
6	314159	Chien	Wu	Marketing	Math	91.21	A
7	314159	Chien	Wu	Marketing	Chemistry	100	B

Question 8: Find all the students from the Analytics stream whose score is greater than the average of the Analytic stream students.



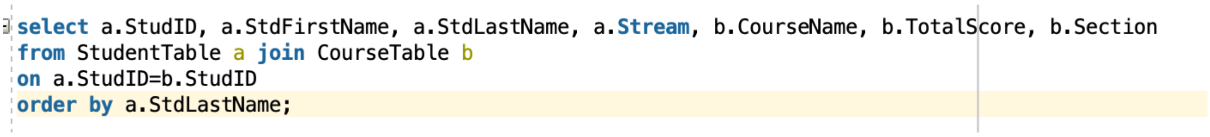
```
select distinct a.studID, a.StdFirstName, a.StdLastName
from StudentTable a join CourseTable b on a.StudID=b.StudID
WHERE a.stream = 'Analytics'
and b.totalscore > (select avg(b.totalscore) from CourseTable b join StudentTable a on a.StudID=B.StudID where a.stream = 'Analytics');
```

Output:



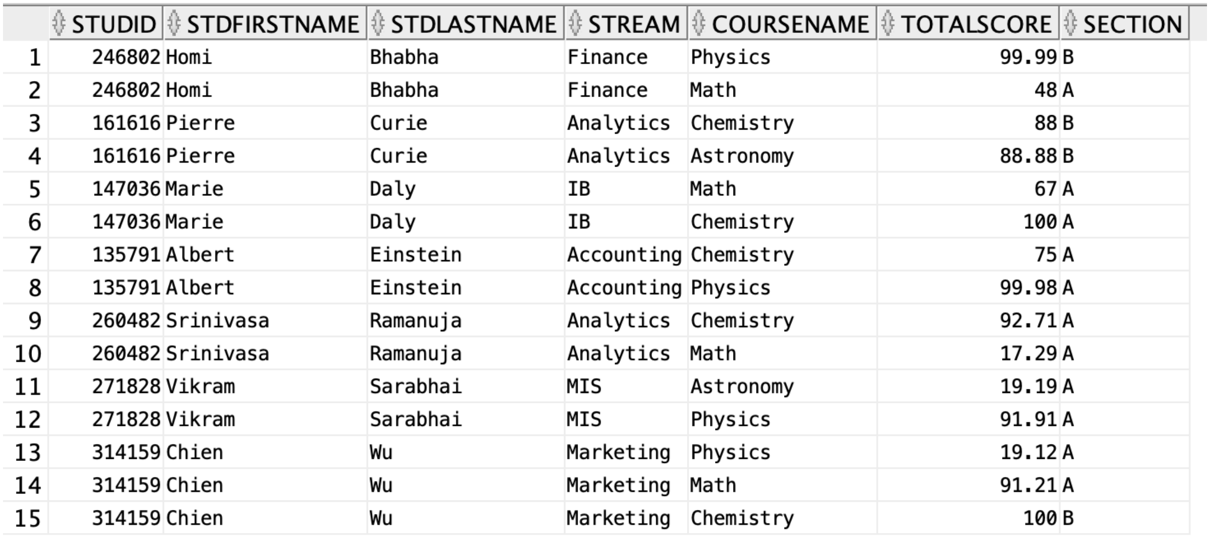
	STUDID	STDFIRSTNAME	STDLASTNAME
1	260482	Srinivasa	Ramanuja
2	161616	Pierre	Curie

Question 9: Print the information from these columns StudID, StdFirstName, StdLastName, TotalScore, CourseName, Section, Stream sorted on the last name of the students



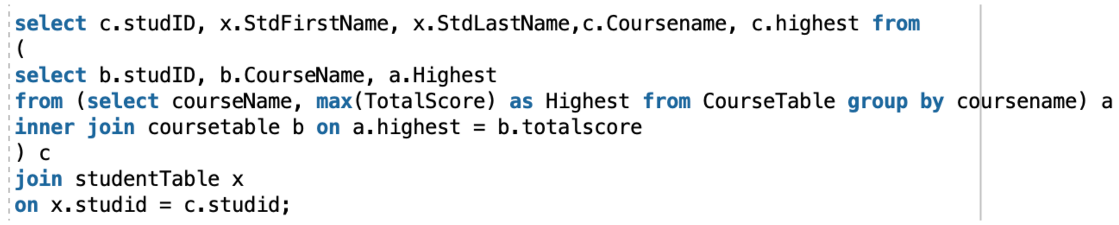
```
select a.StudID, a.StdFirstName, a.StdLastName, a.Stream, b.CourseName, b.TotalScore, b.Section
from StudentTable a join CourseTable b
on a.StudID=b.StudID
order by a.StdLastName;
```

Output:



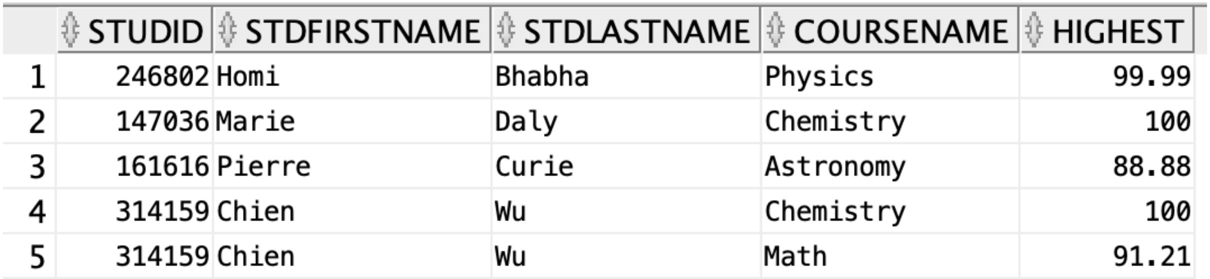
	STUDID	STDFIRSTNAME	STDLASTNAME	STREAM	COURSENAME	TOTALSCORE	SECTION
1	246802	Homi	Bhabha	Finance	Physics	99.99	B
2	246802	Homi	Bhabha	Finance	Math	48	A
3	161616	Pierre	Curie	Analytics	Chemistry	88	B
4	161616	Pierre	Curie	Analytics	Astronomy	88.88	B
5	147036	Marie	Daly	IB	Math	67	A
6	147036	Marie	Daly	IB	Chemistry	100	A
7	135791	Albert	Einstein	Accounting	Chemistry	75	A
8	135791	Albert	Einstein	Accounting	Physics	99.98	A
9	260482	Srinivasa	Ramanuja	Analytics	Chemistry	92.71	A
10	260482	Srinivasa	Ramanuja	Analytics	Math	17.29	A
11	271828	Vikram	Sarabhai	MIS	Astronomy	19.19	A
12	271828	Vikram	Sarabhai	MIS	Physics	91.91	A
13	314159	Chien	Wu	Marketing	Physics	19.12	A
14	314159	Chien	Wu	Marketing	Math	91.21	A
15	314159	Chien	Wu	Marketing	Chemistry	100	B

Question 10: Find the student who received the highest score on each subject (ignore the sections A and B for each subject to find the topper in each subject)



```
select c.studID, x.StdFirstName, x.StdLastName,c.Coursename, c.highest from
(
select b.studID, b.CourseName, a.Highest
from (select courseName, max(TotalScore) as Highest from CourseTable group by coursename) a
inner join coursetable b on a.highest = b.totalscore
) c
join studentTable x
on x.studid = c.studid;
```

Output:



	STUDID	STDFIRSTNAME	STDLASTNAME	COURSENAME	HIGHEST
1	246802	Homi	Bhabha	Physics	99.99
2	147036	Marie	Daly	Chemistry	100
3	161616	Pierre	Curie	Astronomy	88.88
4	314159	Chien	Wu	Chemistry	100
5	314159	Chien	Wu	Math	91.21

Inference:

Performing all the above operations has given me an understanding of how to structure and navigate through a database to get relevant records and perform operations ranging from select, joins, alter, update etc.