

1. Reading done.

2.

a.

Listing all CSC/IT courses at the 200 and 400 level.

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows the 'JVALLES' connection. The 'Tables (Filtered)' pane shows the 'COURSE' table. The 'Worksheet' pane contains the following SQL query:

```
SELECT CID, CourseName, Department, CourseNr
FROM Course CO
WHERE ((CO.DEPARTMENT = 'IT' OR CO.DEPARTMENT = 'CSC') AND CO.COURSENR LIKE '2_%' OR CO.COURSENR LIKE '4_%');
```

The 'Script Output' pane shows the query results:

CID	COURSENAME	DEPARTMENT	COURSENR
1	1020 Theory of Computation	CSC	489
2	1092 Cryptology	CSC	440
3	3201 Data Analysis	IT	223
4	9219 Databases Technologies	CSC	453

b.

Listing all graduate students enrolled in an undergraduate course (coursenr < 400).

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows the 'JVALLES' connection. The 'Tables (Filtered)' pane shows the 'ENROLLED' table. The 'Worksheet' pane contains the following SQL query:

```
SELECT DISTINCT ST.SID, ST.LastName, ST.FirstName
FROM Student ST, Enrolled EN, Course CR
WHERE (ST.CAREER = 'GRD' AND (EN.STUDENTID = ST.SID AND EN.COURSEID = CR.CID AND CAST(COURSENR AS NUMBER) < 400));
```

The 'Script Output' pane shows the query results:

SID	LASTNAME	FIRSTNAME
1	8871 Snowdon	Jonathan

SELECT DISTINCT ST.SID, ST.LastName, ST.FirstName
FROM Student ST, Enrolled EN, Course CR
WHERE (ST.CAREER = 'GRD' AND (EN.STUDENTID = ST.SID AND EN.COURSEID = CR.CID AND
CAST(COURSENR AS NUMBER) < 400));

I added course id 3201 for student 8871 to test query above.

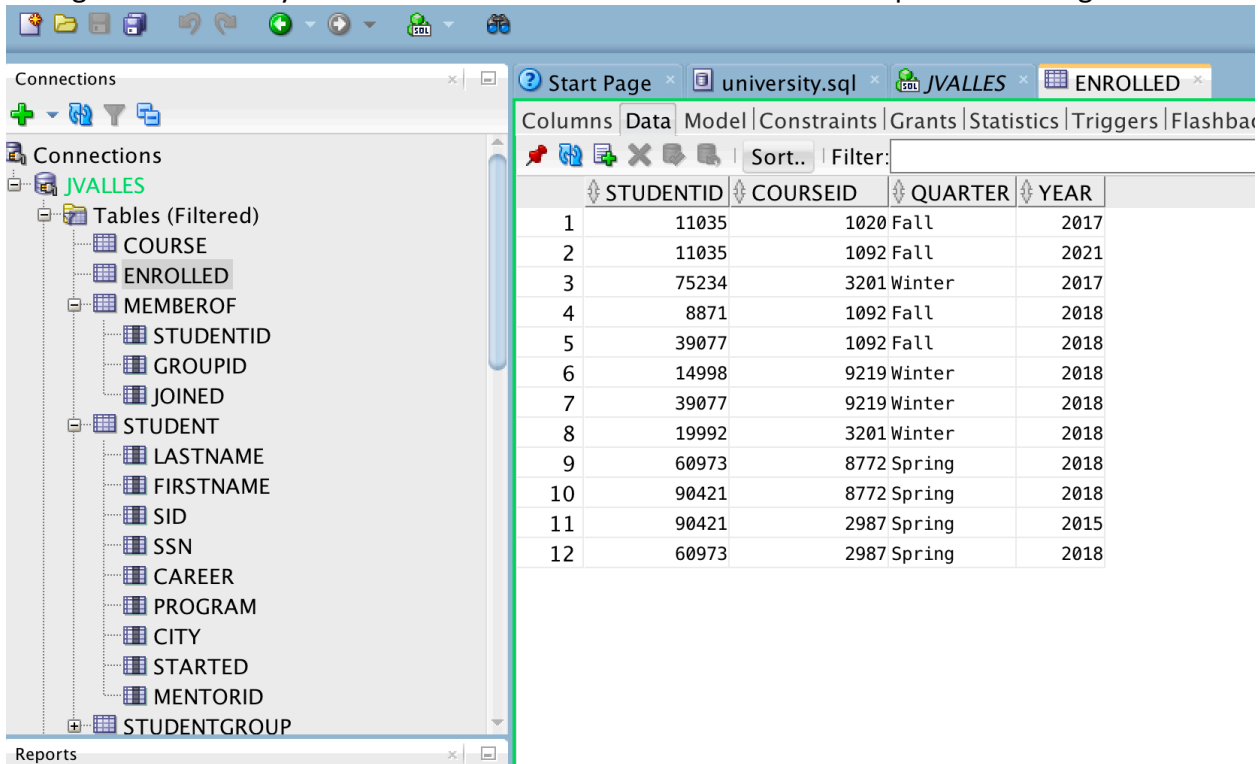
The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows the 'JVALLES' connection. The 'Tables (Filtered)' pane shows the 'ENROLLED' table. The 'Worksheet' pane contains the following SQL query:

```
SELECT ST.SID, ST.LastName, ST.FirstName, EN.COURSEID, EN.QUARTER, EN.YEAR
FROM Student ST, Enrolled EN, Course CR
WHERE (ST.CAREER = 'GRD' AND (EN.STUDENTID = ST.SID AND EN.COURSEID = CR.CID AND CAST(COURSENR AS NUMBER) < 400));
```

The 'Script Output' pane shows the query results:

SID	LASTNAME	FIRSTNAME	COURSEID	QUARTER	YEAR
1	11035		1020	Fall	2017
2	11035		1092	Fall	2021
3	75234		3201	Winter	2017
4	8871		3201	Fall	2018
5	39077		1092	Fall	2018
6	14998		9219	Winter	2018

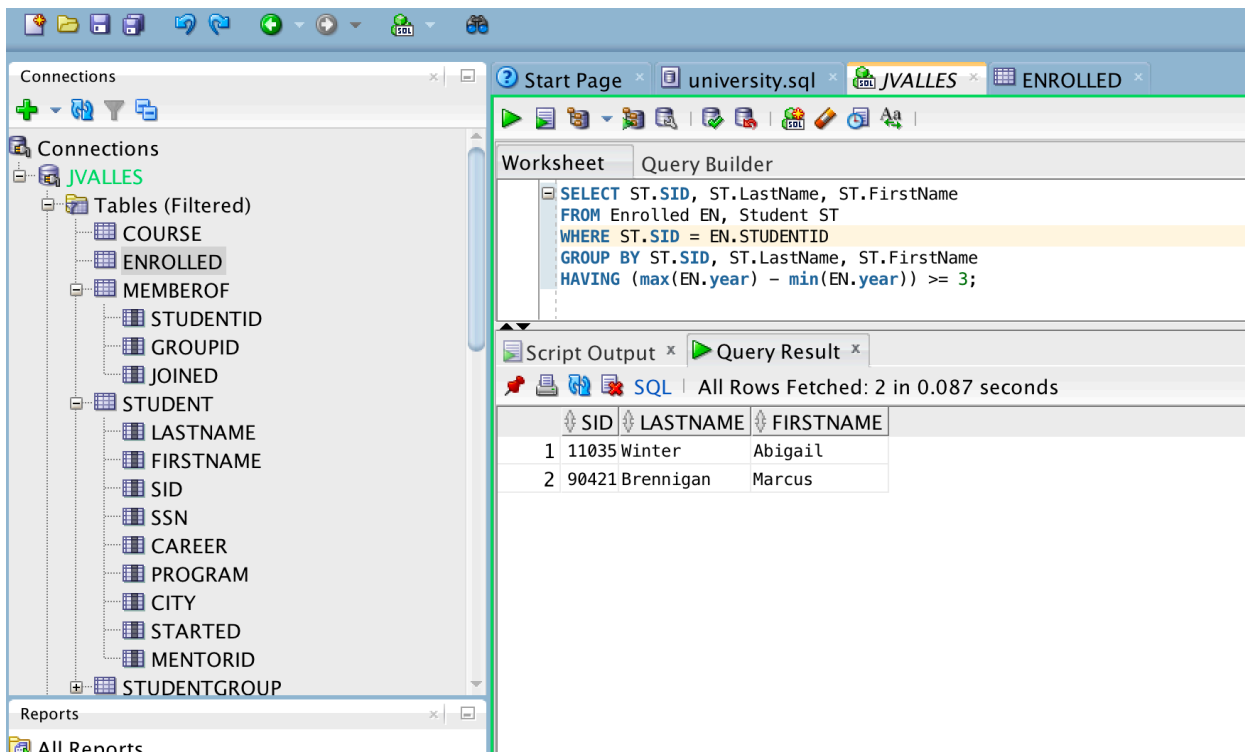
C.
Changed enrollment year for 11035 to 2012 and 90421 to 2015 for 'positive testing'.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Connections' pane displays the 'JVALLES' database with a tree view of tables: COURSE, ENROLLED, MEMBEROF, STUDENTID, GROUPID, JOINED, STUDENT, LASTNAME, FIRSTNAME, SID, SSN, CAREER, PROGRAM, CITY, STARTED, MENTORID, and STUDENTGROUP. The 'ENROLLED' table is selected. On the right, the 'Data' tab shows the contents of the 'ENROLLED' table. The table has columns: STUDENTID, COURSEID, QUARTER, and YEAR. The data is as follows:

	STUDENTID	COURSEID	QUARTER	YEAR
1	11035	1020	Fall	2017
2	11035	1092	Fall	2021
3	75234	3201	Winter	2017
4	8871	1092	Fall	2018
5	39077	1092	Fall	2018
6	14998	9219	Winter	2018
7	39077	9219	Winter	2018
8	19992	3201	Winter	2018
9	60973	8772	Spring	2018
10	90421	8772	Spring	2018
11	90421	2987	Spring	2015
12	60973	2987	Spring	2018

Based on query, there were two long-term students. Had to calculate diff between max, min years for each student. Used GROUP BY and HAVING.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Connections' pane displays the 'JVALLES' database with a tree view of tables: COURSE, ENROLLED, MEMBEROF, STUDENTID, GROUPID, JOINED, STUDENT, LASTNAME, FIRSTNAME, SID, SSN, CAREER, PROGRAM, CITY, STARTED, MENTORID, and STUDENTGROUP. The 'ENROLLED' table is selected. On the right, the 'Query Builder' tab shows a query:

```
SELECT ST.SID, ST.LastName, ST.FirstName
FROM Enrolled EN, Student ST
WHERE ST.SID = EN.STUDENTID
GROUP BY ST.SID, ST.LastName, ST.FirstName
HAVING (max(EN.year) - min(EN.year)) >= 3;
```

Below the query, the 'Script Output' and 'Query Result' tabs are visible. The 'Query Result' tab shows the results of the query:

	SID	LASTNAME	FIRSTNAME
1	11035	Winter	Abigail
2	90421	Brennigan	Marcus

3.
Create tables for Orders, Customer, StateTax. And populated tables with data.

a.

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The 'Worksheet' contains the following SQL code:

```
create table statetax (
  State          varchar(15),
  SalesTaxRate   number(5,2),
  primary key (State)
);

create table customer (
  CID            number(5),
  Name           varchar(15),
  State          varchar(15),
  primary key (CID),
  foreign key (State) references statetax(State)
);

create table orders (
  OrderID        number(5),
  CustomerID      number(5),
  NumberOfItems  number(7),
  ItemsTotal     number(9,2),
  primary key (OrderID),
  foreign key (CustomerID) references customer(CID)
);

insert into statetax(State, SalesTaxRate)
values ('Illinois', 6.25);
insert into statetax(State, SalesTaxRate)
values ('California', 8.01);
insert into statetax(State, SalesTaxRate)
values ('New York', 7);

insert into customer(CID, Name, State)
values (11035, 'James', 'Illinois');
insert into customer(CID, Name, State)
values (11036, 'Sam', 'California');
insert into customer(CID, Name, State)
values (11037, 'Leo', 'New York');

insert into orders(OrderID, CustomerID, NumberOfItems, ItemsTotal)
values (93321, 11035, 7, 239.49);
insert into orders(OrderID, CustomerID, NumberOfItems, ItemsTotal)
values (93322, 11036, 5, 339.49);
insert into orders(OrderID, CustomerID, NumberOfItems, ItemsTotal)
values (93323, 11037, 10, 299);
```

The 'Script Output' window at the bottom shows the execution results:

```
Task completed in 0.791 seconds
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
```

The screenshot shows the 'STATETAX' table data in the 'Columns' tab. The table has two columns: STATE and SALESTAXRATE.

STATE	SALESTAXRATE
1 Illinois	6.25
2 California	8.01
3 New York	7

The screenshot shows the 'CUSTOMER' table data in the 'Columns' tab. The table has three columns: CID, NAME, and STATE.

CID	NAME	STATE
1 11035	James	Illinois
2 11036	Sam	California
3 11037	Leo	New York

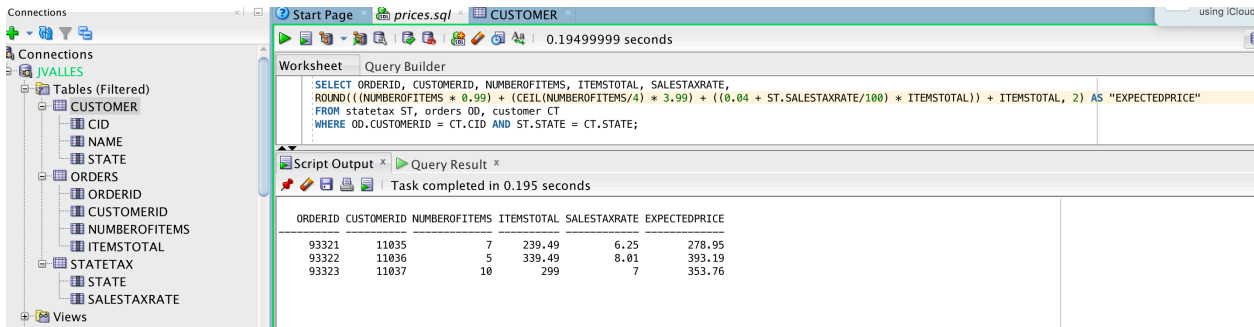
The screenshot shows the 'ORDERS' table data in the 'Columns' tab. The table has four columns: ORDERID, CUSTOMERID, NUMBEROFITEMS, and ITEMSTOTAL.

ORDERID	CUSTOMERID	NUMBEROFITEMS	ITEMSTOTAL
1 93321	11035	7	239.49
2 93322	11036	5	339.49
3 93323	11037	10	299

b.

List all orders and the expected price the customer has to pay. Expected price calculated according to rules. Used Ceil to calculate bundle shipment charge.

```
SELECT ORDERID, CUSTOMERID, NUMBEROFITEMS, ITEMSTOTAL, SALESTAXRATE,
ROUND(((NUMBEROFITEMS * 0.99) + (CEIL(NUMBEROFITEMS/4) * 3.99) + ((0.04 +
ST.SALESTAXRATE/100) * ITEMSTOTAL)) + ITEMSTOTAL, 2) AS "EXPECTEDPRICE"
FROM statetax ST, orders OD, customer CT
WHERE OD.CUSTOMERID = CT.CID AND ST.STATE = CT.STATE;
```



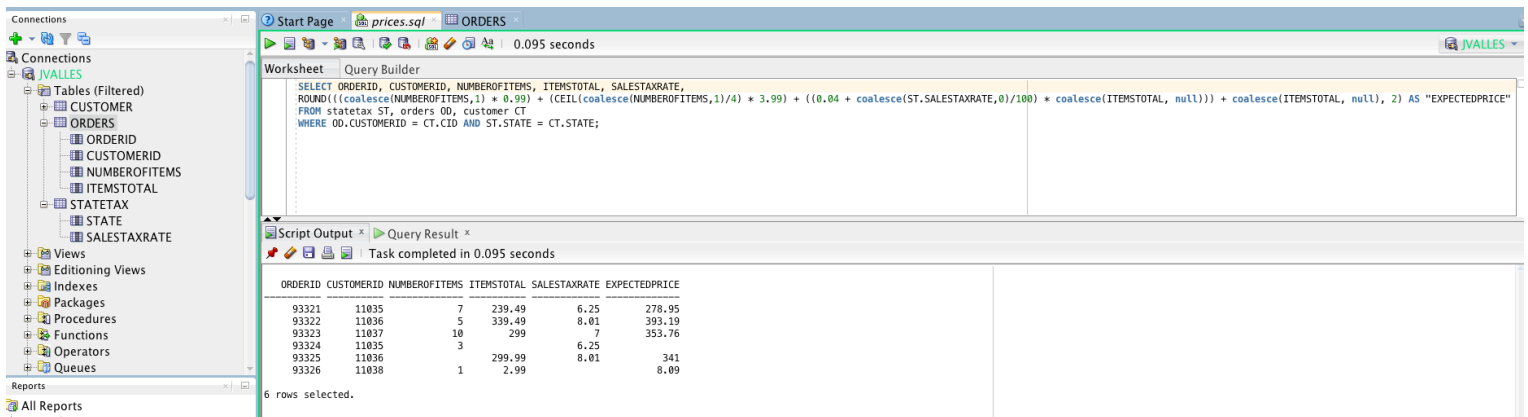
The screenshot shows a database query tool interface. On the left, a tree view displays the database schema with tables: CUSTOMER, ORDERS, STATETAX, and VIEWS. The main window shows a query in the 'Query Builder' tab. The query is the same as the one in block b. Below the query, the 'Script Output' tab shows the execution time as 0.19499999 seconds. The 'Query Result' tab displays the results of the query in a table format.

ORDERID	CUSTOMERID	NUMBEROFITEMS	ITEMSTOTAL	SALESTAXRATE	EXPECTEDPRICE
93321	11035	7	239.49	6.25	278.95
93322	11036	5	339.49	8.01	393.19
93323	11037	10	299	7	353.76

c.

Added coalesce function to input default values should for null values.

```
SELECT ORDERID, CUSTOMERID, NUMBEROFITEMS, ITEMSTOTAL, SALESTAXRATE,
ROUND(((coalesce(NUMBEROFITEMS,1) * 0.99) + (CEIL(coalesce(NUMBEROFITEMS,1)/4) * 3.99) + ((0.04
+ coalesce(ST.SALESTAXRATE,0)/100) * coalesce(ITEMSTOTAL, null))) + coalesce(ITEMSTOTAL, null), 2) AS
"EXPECTEDPRICE"
FROM statetax ST, orders OD, customer CT
WHERE OD.CUSTOMERID = CT.CID AND ST.STATE = CT.STATE;
```



The screenshot shows a database query tool interface. On the left, a tree view displays the database schema with tables: CUSTOMER, ORDERS, STATETAX, and VIEWS. The main window shows a query in the 'Query Builder' tab. The query is the same as the one in block c. Below the query, the 'Script Output' tab shows the execution time as 0.095 seconds. The 'Query Result' tab displays the results of the query in a table format.

ORDERID	CUSTOMERID	NUMBEROFITEMS	ITEMSTOTAL	SALESTAXRATE	EXPECTEDPRICE
93321	11035	7	239.49	6.25	278.95
93322	11036	5	339.49	8.01	393.19
93323	11037	10	299	7	353.76
93324	11035	3		6.25	
93325	11036		299.99	8.01	341
93326	11038	1	2.99		8.09

6 rows selected.

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashba
Sort..	Filter:						
ORDERID	CUSTOMERID	NUMBEROFITEMS	ITEMSTOTAL				
1	93321	11035	7	239.49			
2	93322	11036	5	339.49			
3	93323	11037	10	299			
4	93324	11035	3	(null)			
5	93325	11036	(null)	299.99			
6	93326	11038	1	2.99			

4.

a. Added mentorID field.

The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane shows the 'JVALLES' database with a tree view of tables: COURSE, ENROLLED, MEMBEROF, and STUDENT. The 'STUDENT' table is expanded, showing columns: LASTNAME, FIRSTNAME, SID, and SSN. In the center, the 'Query Builder' tab is active, displaying the SQL command: `ALTER TABLE student ADD MentorID NUMBER(5);`. Below the query, the 'Script Output' pane shows the message: 'Task completed in 0.51' and 'Table STUDENT altered.'

Also set up foreign key (MentorID) constraint referencing Student(SID).

The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane shows the 'JVALLES' database with a tree view of tables: COURSE, ENROLLED, MEMBEROF, STUDENT, and STUDENTGROUP. The 'STUDENT' table is expanded, showing columns: LASTNAME, FIRSTNAME, SID, and SSN. In the center, the 'Query Builder' tab is active, displaying the SQL command: `ALTER TABLE student ADD CONSTRAINT fk_mid_sid FOREIGN KEY(MentorID) REFERENCES student(SID);`. Below the query, the 'Script Output' pane shows the message: 'Task completed in 0.112 seconds' and 'Table STUDENT altered.'

connections

Connections

- JVALLES
 - Tables (Filtered)
 - COURSE
 - ENROLLED
 - MEMBEROF
 - STUDENT
 - STUDENTGROUP
 - Views
 - Editing Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues

Start Page | university.sql | JVALLES | STUDENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Depe

Actions...

	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME
1	Foreign_Key	(null)	JVALLES1	STUDENT
2	Check	"LASTNAME" IS NOT NULL	(null)	(null)
3	Primary_Key	(null)	(null)	(null)
4	Unique	(null)	(null)	(null)

Assigning mentors to some(not all) of the students.

connections

Connections

- JVALLES
 - Tables (Filtered)
 - COURSE
 - ENROLLED
 - MEMBEROF
 - STUDENT
 - STUDENTGROUP
 - Views
 - Editing Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues
 - Queues Tables
 - Triggers
 - Crossedition Triggers
 - Types

Start Page | university.sql | JVALLES | STUDENT

0.2579999 seconds

Worksheet | Query Builder

```
UPDATE student
SET mentorid = '39077'
WHERE SID = '14662';

UPDATE student
SET mentorid = '14998'
WHERE SID = '08871';

UPDATE student
SET mentorid = '58992'
WHERE SID = '19992';
```

Script Output x

Task completed in 0.151 seconds

1 row updated.

1 row updated.

1 row updated.

Oracle SQL Developer : Table JVALLES1.STUDENT@JVALLES

Connections

- JVALLES
 - Tables (Filtered)
 - COURSE
 - ENROLLED
 - MEMBEROF
 - STUDENT
 - STUDENTGROUP
 - Views
 - Editing Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues
 - Queues Tables
 - Triggers
 - Crossedition Triggers
 - Types

Start Page | university.sql | JVALLES | STUDENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

	LASTNAME	FIRSTNAME	SID	SSN	CAREER	PROGRAM	CITY	STARTED	MENTORID
1	Brennigan	Marcus	90421	987654321	UGRD	INFO-TECH	Evanston	2015	(null)
2	Patel	Deepa	14662	(null)	GRD	COMP-SCI	Evanston	2016	39077
3	Snowdon	Jonathan	8871	123123123	GRD	INFO-SYS	Springfield	2017	14998
4	Starck	Jason	19992	789789789	UGRD	INFO-SYS	Springfield	2014	58992
5	Johnson	Peter	32105	123456789	UGRD	COMP-SCI	Chicago	2017	(null)
6	Winter	Abigail	11035	111111111	GRD	PHD	Chicago	2016	(null)
7	Patel	Prakash	75234	(null)	UGRD	COMP-SCI	Chicago	2014	(null)
8	Snowdon	Jennifer	93321	321321321	GRD	COMP-SCI	Springfield	2015	(null)
9	Degroff	Jarvis	14998	113311331	GRD	INFO-TECH	Evanston	2016	(null)
10	Kubik	Dwayne	57923	979797979	UGRD	COMP-SCI	Springfield	2018	(null)
11	Skelly	Trinity	58992	555222555	GRD	PHD	Springfield	2018	(null)
12	Krol	Angelo	60973	(null)	UGRD	COMP-SCI	Springfield	2013	(null)
13	Pollard	Joya	39077	(null)	GRD	COMP-SCI	Springfield	2017	(null)

- b.
- Query to list all students which has not been assigned a mentor.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'JVALLES' database. The 'Tables (Filtered)' list includes COURSE, ENROLLED, MEMBEROF, STUDENT, and STUDENTGROUP. The 'Script Output' pane shows the results of a query executed in the 'Query Builder'.

Query:

```
SELECT SID, LastName, FirstName
FROM student
WHERE MENTORID is null;
```

Script Output:

SID	LASTNAME	FIRSTNAME
90421	Brennigan	Marcus
32105	Johnson	Peter
11035	Winter	Abigail
75234	Patel	Prakash
93321	Snowdon	Jennifer
14998	Degroff	Jarvis
57923	Kubik	Dwayne
58992	Skelly	Trinity
60973	Krol	Angelo
39077	Pollard	Joya

10 rows selected.

- c.
- List all mentees which have been assigned a mentor that do not have same careers.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'JVALLES' database. The 'Tables (Filtered)' list includes COURSE, ENROLLED, MEMBEROF, STUDENT, and STUDENTGROUP. The 'Script Output' pane shows the results of a query executed in the 'Query Builder'.

Query:

```
SELECT S2.SID, S2.LastName
FROM student S1, student S2
WHERE S1.SID = S2.MENTORID AND S1.CAREER <> S2.CAREER;
```

Script Output:

SID	LASTNAME
1	19992 Starck

Returned student whose last name is Starck using data set below.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'JVALLES' database. The 'Tables (Filtered)' list includes COURSE, ENROLLED, MEMBEROF, STUDENT, and STUDENTGROUP. The 'Script Output' pane shows the results of a query executed in the 'Query Builder'.

Query:

```
SELECT * FROM STUDENT;
```

Script Output:

SID	LASTNAME	FIRSTNAME	SID	SSN	CAREER	PROGRAM	CITY	STARTED	MENTORID
1	Brennigan	Marcus	90421	987654321	UGRD	INFO-TECH	Evanston	2015	(null)
2	Patel	Deepa	14662	(null)	GRD	COMP-SCI	Evanston	2016	39077
3	Snowdon	Jonathan	8871	123123123	GRD	INFO-SYS	Springfield	2017	14998
4	Starck	Jason	19992	789789789	UGRD	INFO-SYS	Springfield	2014	58992
5	Johnson	Peter	32105	123456789	UGRD	COMP-SCI	Chicago	2017	(null)
6	Winter	Abigail	11035	111111111	GRD	PHD	Chicago	2016	(null)
7	Patel	Prakash	75234	(null)	UGRD	COMP-SCI	Chicago	2014	(null)
8	Snowdon	Jennifer	93321	321321321	GRD	COMP-SCI	Springfield	2015	(null)
9	Degroff	Jarvis	14998	113311331	GRD	INFO-TECH	Evanston	2016	(null)
10	Kubik	Dwayne	57923	979797979	UGRD	COMP-SCI	Springfield	2018	(null)
11	Skelly	Trinity	58992	555222555	GRD	PHD	Springfield	2018	(null)
12	Krol	Angelo	60973	(null)	UGRD	COMP-SCI	Springfield	2013	(null)
13	Pollard	Joya	39077	(null)	GRD	COMP-SCI	Springfield	2017	(null)