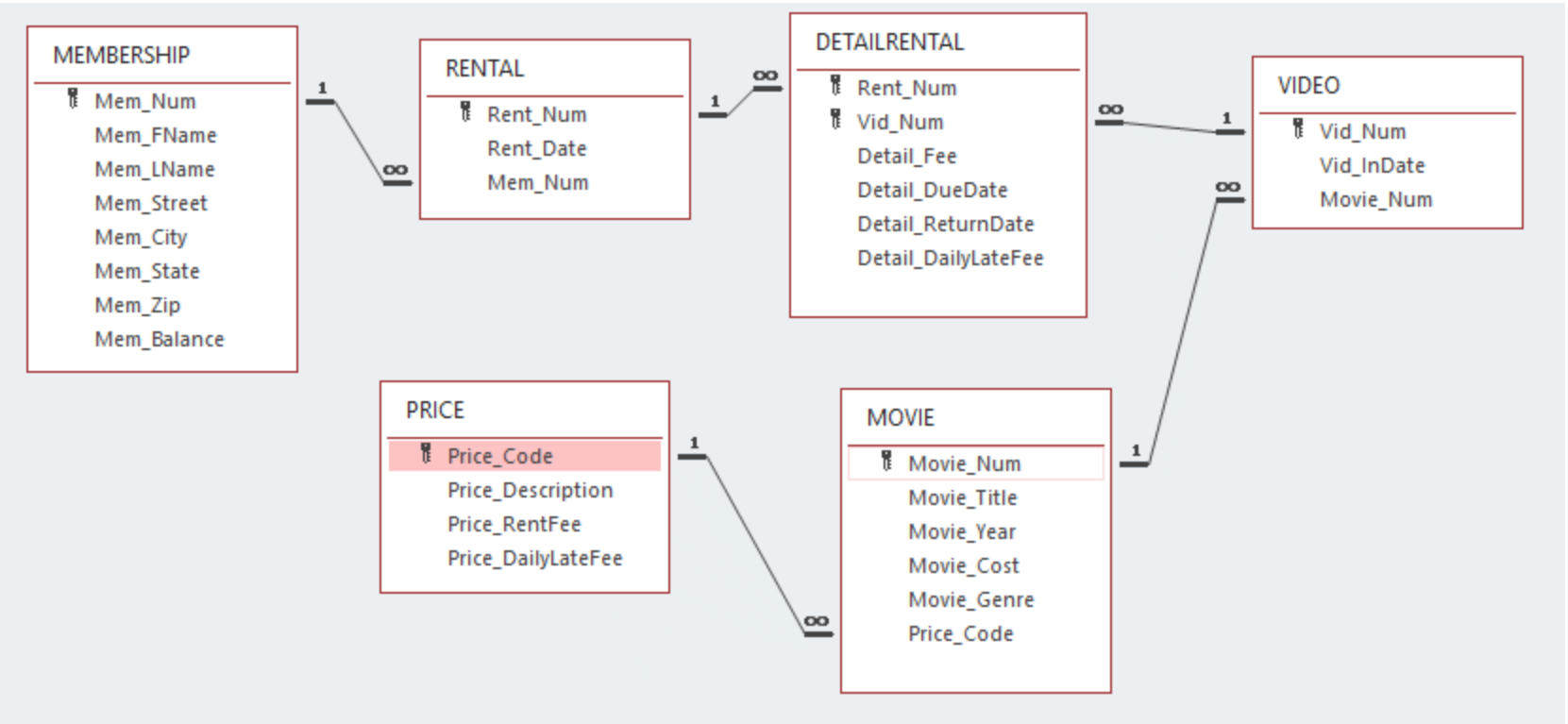
CIDS 333 FINAL EXAM
Points: 15%

Justin Akwuba

Sample

Question. Write a query to show all the records (members) from the Membership table



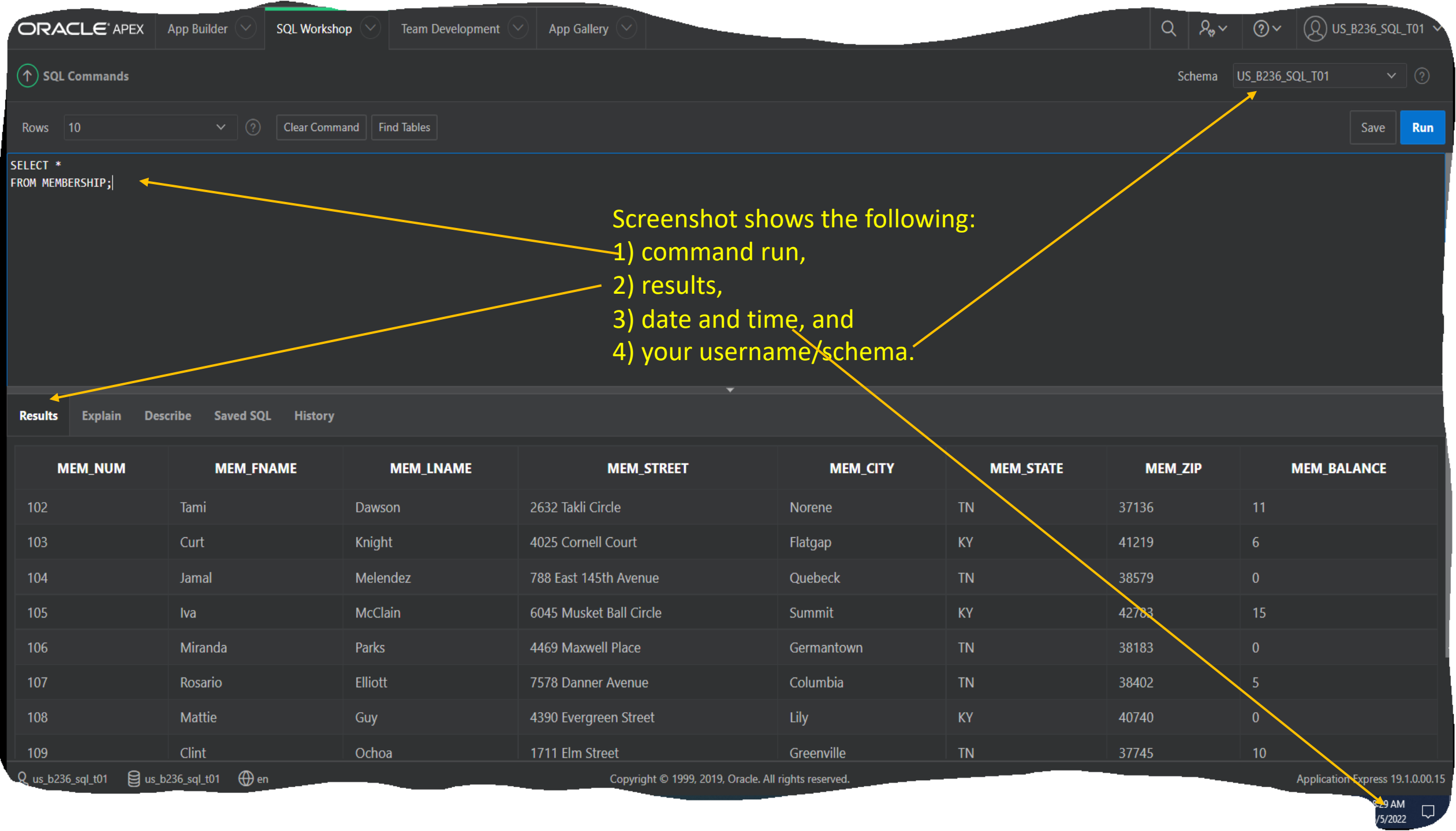
Answer

SQL:

```
Select *
From membership;
```

Notice the requirements for the screenshots in the next slide

Make sure to follow the format for the screenshot.


ORACLE APEX
App Builder
SQL Workshop
Team Development
App Gallery
US_B236_SQL_T01
SQL Commands
Schema
US_B236_SQL_T01
Rows
10
Clear Command
Find Tables
Save
Run
SELECT *
FROM MEMBERSHIP;
Screenshot shows the following:
1) command run,
2) results,
3) date and time, and
4) your username/schema.
Results
Explain
Describe
Saved SQL
History
MEM_NUM
MEM_FNAME
MEM_LNAME
MEM_STREET
MEM_CITY
MEM_STATE
MEM_ZIP
MEM_BALANCE
102 Tami Dawson 2632 Takli Circle Norene TN 37136 11
103 Curt Knight 4025 Cornell Court Flatgap KY 41219 6
104 Jamal Melendez 788 East 145th Avenue Quebeck TN 38579 0
105 Iva McClain 6045 Musket Ball Circle Summit KY 42783 15
106 Miranda Parks 4469 Maxwell Place Germantown TN 38183 0
107 Rosario Elliott 7578 Danner Avenue Columbia TN 38402 5
108 Mattie Guy 4390 Evergreen Street Lily KY 40740 0
109 Clint Ochoa 1711 Elm Street Greenville TN 37745 10
us_b236_sql_t01
us_b236_sql_t01
en
Copyright © 1999, 2019, Oracle. All rights reserved.
Application Express 19.1.0.00.15

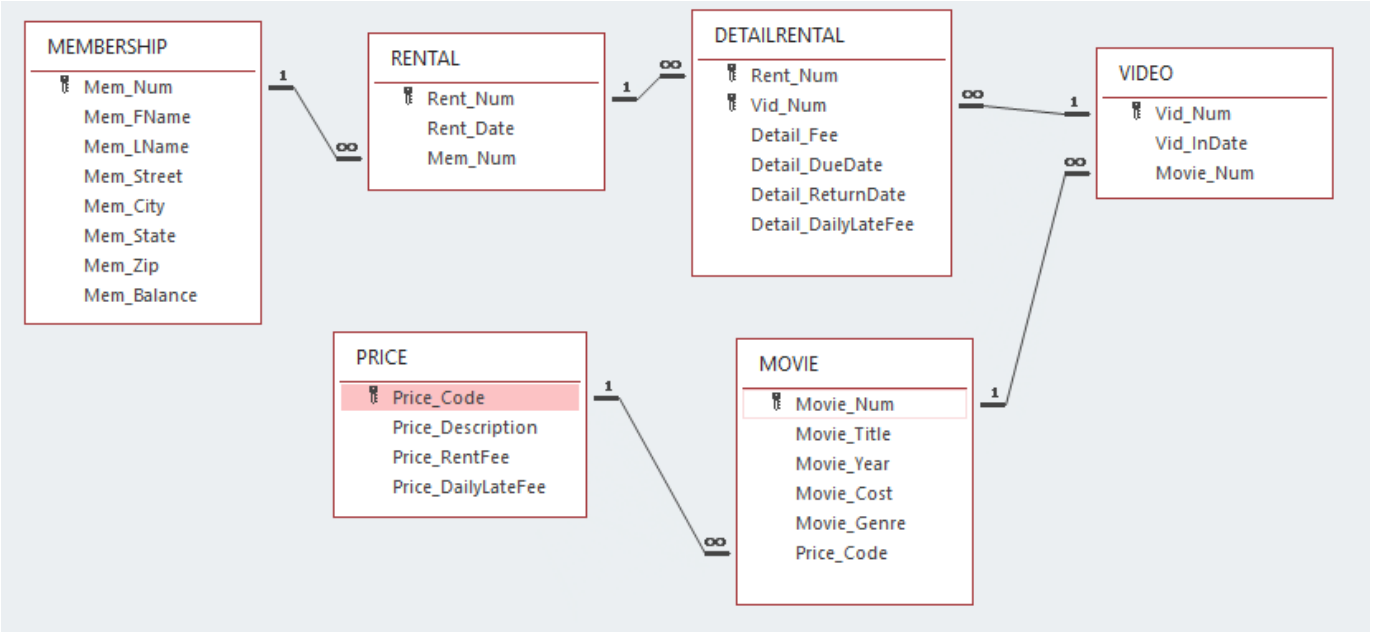
Begin

0. Data Entry

Enter a new member record for the **Membership** table. Assume you are the member of this "movie club."

- Use any integer between **3000** and **4000** as **Mem_Num**.
- Put your **own name** as a member.
- Put a *fictional street address* for **Mem_Street** and a *fictional name* for the **City** and **Zip**.
- Put any integer between **0** and **100** for **Member Balance** (**Mem_Balance**).
- For **State (Mem_State)** put it as **ZZ**.

Since you will enter a new record, the results of some of the queries may be different from what is shown in the expected results for the query, but you will be able to figure that out and account for it.



I refuse to join any club that would have me as a member.

~ Groucho Marx

Answer

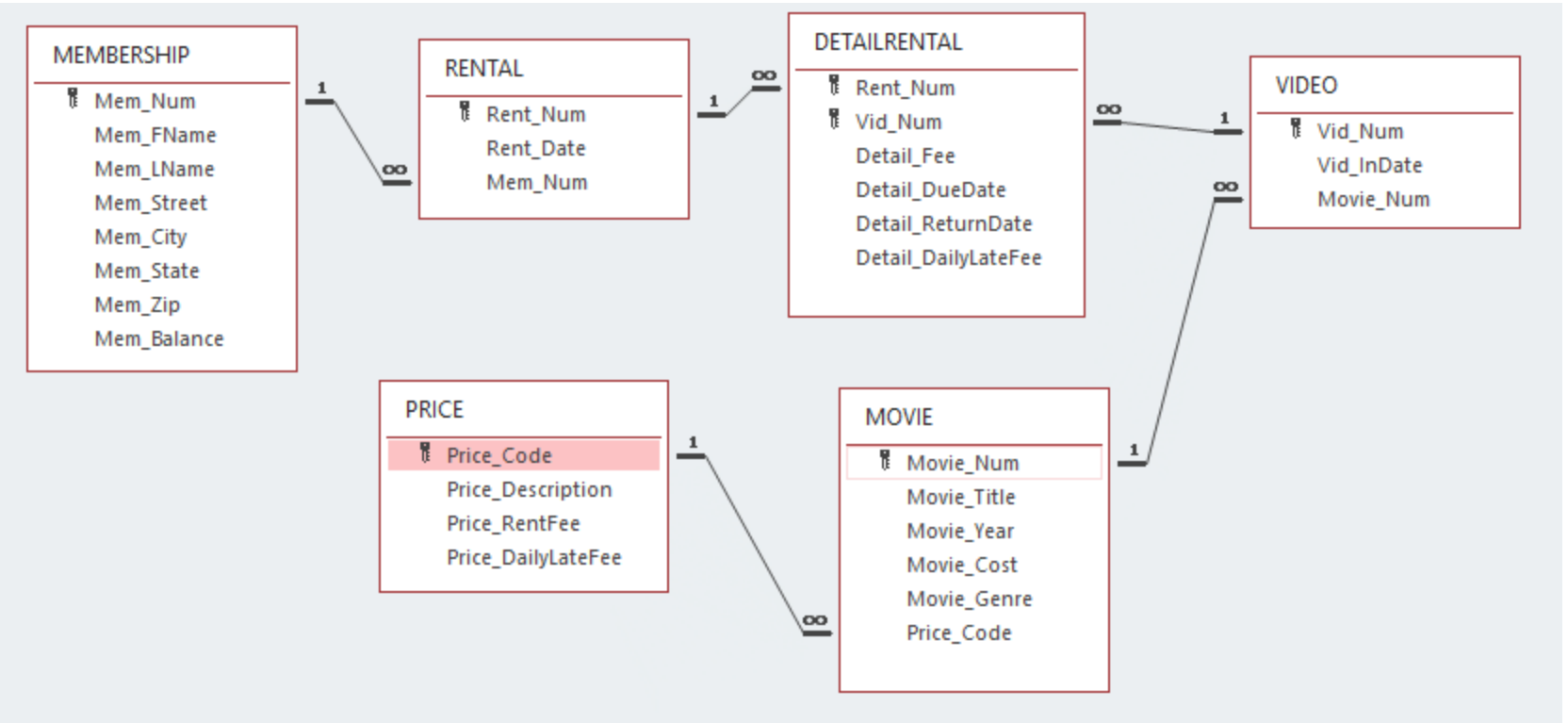
SQL:

```
INSERT INTO MEMBERSHIP (Mem_Num, Mem_FName, Mem_LName, Mem_Street, Mem_City, Mem_State, Mem_Zip, Mem_Balance)
VALUES (3200, 'Justin', 'Akwuba', '736 Sonic dr', 'Green Hills', 'ZZ', 55123, 45);
```

1a. Retrieve all records from the table Membership. Sort the result by Member State in a descending order. You should get the following result:

MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
107	Rosario	Elliott	7578 Danner Avenue	Columbia	TN	38402	5
112	Luis	Trujillo	7267 Melvin Avenue	Heiskell	TN	37754	3
111	Stacy	Mann	2789 East Cook Avenue	Murfreesboro	TN	37132	8
104	Jamal	Melendez	788 East 145th Avenue	Quebeck	TN	38579	0
106	Miranda	Parks	4469 Maxwell Place	Germantown	TN	38183	0
113	Minnie	Gonzales	6430 Vasili Drive	Williston	TN	38076	0
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10
110	Lewis	Rosales	4524 SouthWind Circle	Counce	TN	38326	0
105	Iva	McClain	6045 Musket Ball Circle	Summit	KY	42783	15
108	Mattie	Guy	4390 Evergreen Street	Lily	KY	40740	0
103	Curt	Knight	4025 Cornell Court	Flatgap	KY	41219	6

Your own name should also appear as the first record since the records are sorted in a descending order on Mem_State which you would have entered earlier as ZZ.



SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

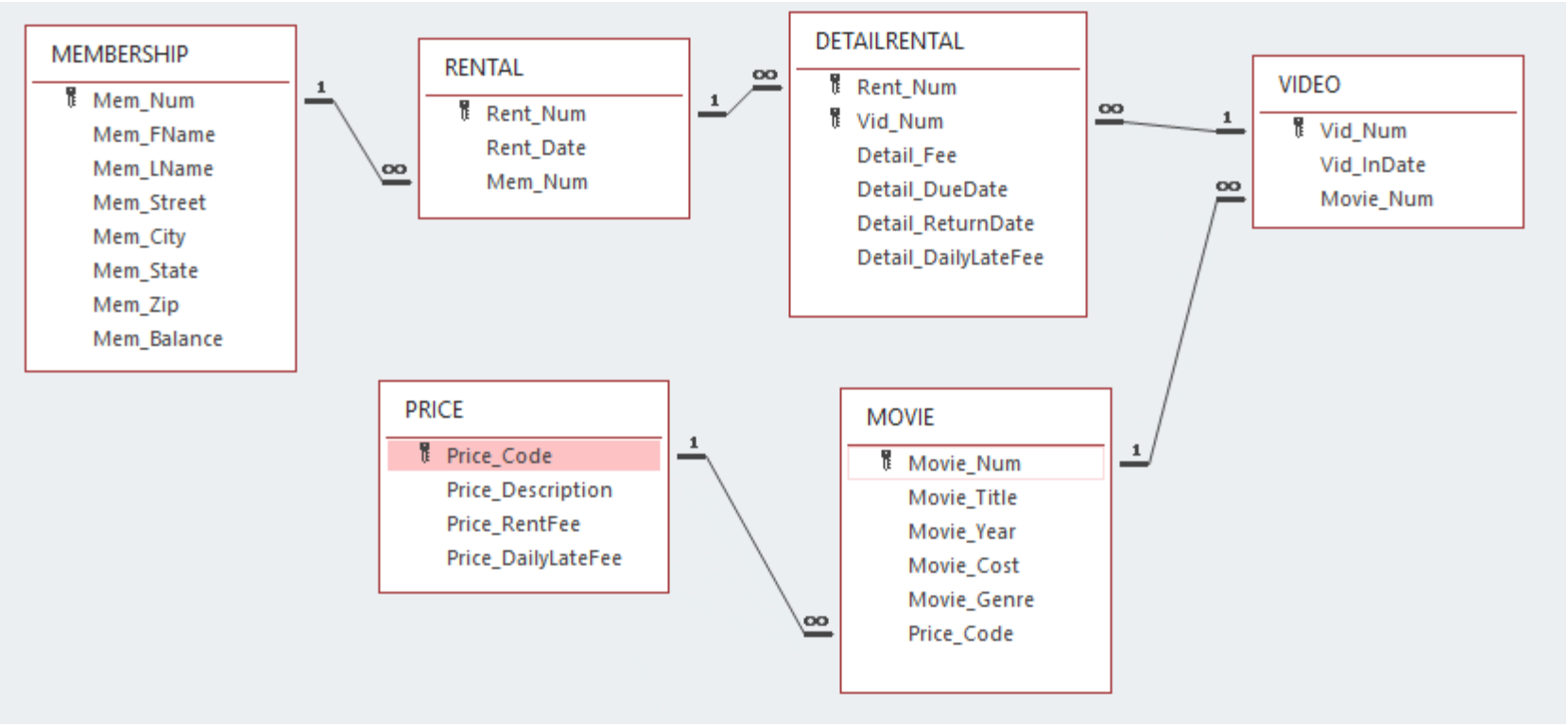
```
1  INSERT INTO MEMBERSHIP (Mem_Num, Mem_FName, Mem_LName, Mem_Street, Mem_City, Mem_State, Mem_Zip, Mem_Balance)
2  VALUES (3200, 'Justin', 'Akwuba', '736 Sonic dr', 'Green Hills', 'ZZ', 55123, 45);
3
4
5
6  SELECT *
7  FROM MEMBERSHIP
8  ORDER BY Mem_State DESC;
9
```

Results Explain Describe Saved SQL History

3200	Justin	Akwuba	736 Sonic dr	Green Hills	ZZ	55123	45
104	Jamal	Melendez	788 East 145th Avenue	Quebeck	TN	38579	0
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
107	Rosario	Elliott	7578 Danner Avenue	Columbia	TN	38402	5
113	Minnie	Gonzales	6430 Vasili Drive	Williston	TN	38076	0

Copyright © 1999, 2022, Oracle and/or its affiliates.

1b. Retrieve all records from any table **other than** Membership. **Sort** the result by any field of your choice in an ***ascending*** order.



Answer

SQL:

```
SELECT *
FROM MOVIE
ORDER BY Movie_Title ASC;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1  INSERT INTO MEMBERSHIP (Mem_Num, Mem_FName, Mem_LName, Mem_Street, Mem_City, Mem_State, Mem_Zip, Mem_Balance)
2  VALUES (3200, 'Justin', 'Akwuba', '736 Sonic dr', 'Green Hills', 'ZZ', 55123, 45);
3
4
5
6  SELECT *
7  FROM MEMBERSHIP
8  ORDER BY Mem_State DESC;
9
10
11
12 SELECT *
13 FROM MOVIE
14 ORDER BY Movie_Title ASC;
15
```

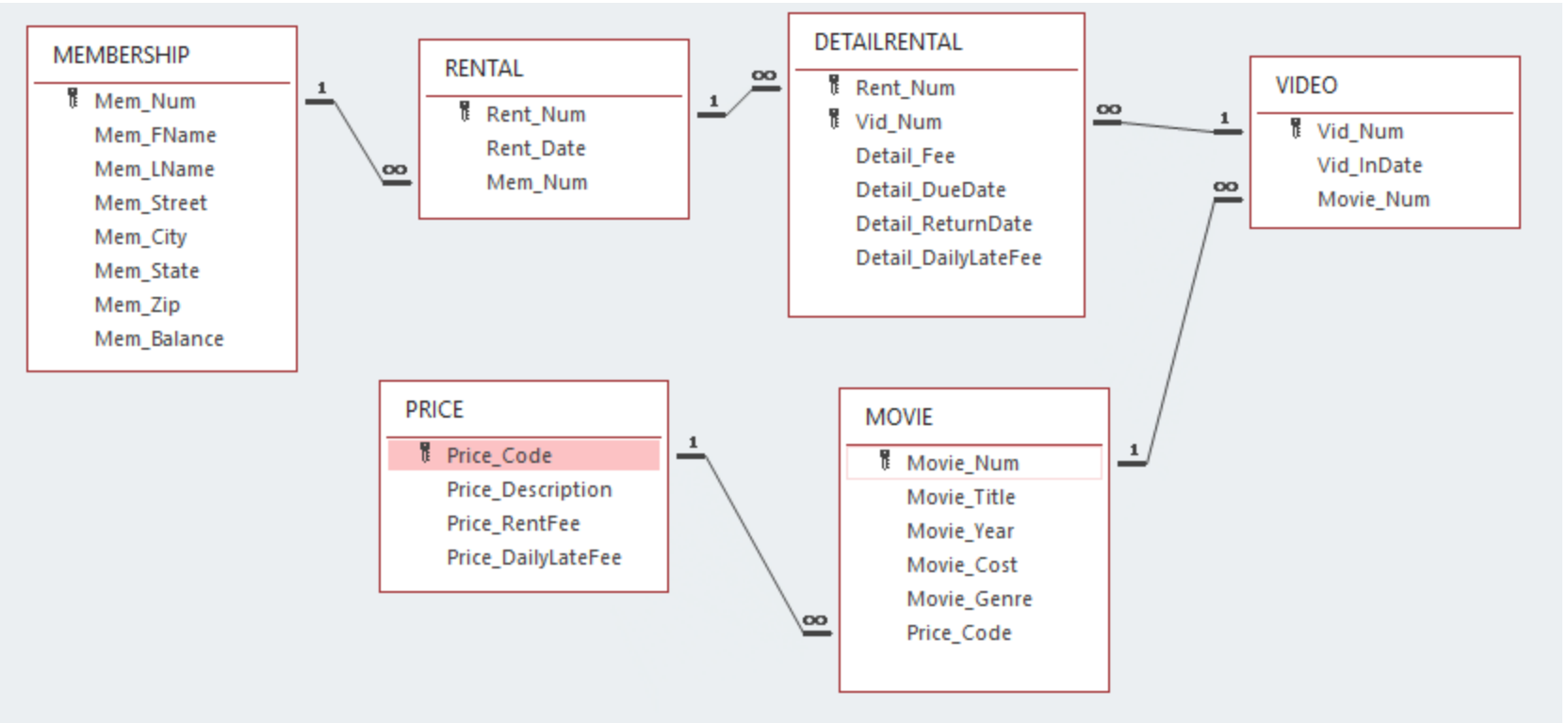
Results Explain Describe Saved SQL History

MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST	MOVIE_GENRE	PRICE_CODE
1237	Beatnik Fever	2009	29.95	COMEDY	2
1238	Constant Companion	2010	89.95	DRAMA	2
1236	Richard Goodhope	2010	59.95	DRAMA	2
1235	Smokey Mountain Wildlife	2006	59.95	ACTION	3

Copyright © 1999, 2022, Oracle and/or its affiliates.

2. Retrieve only those member records for members from the **Membership** table who belong to the state of **Tennessee** and have a *balance* greater than **five**. **Sort** the result based on **Member Balance** in *descending* order.

MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10
111	Stacy	Mann	2789 East Cook Avenue	Murfreesboro	TN	37132	8



Answer

SQL:

```
SELECT *
FROM MEMBERSHIP
WHERE Mem_State = 'TN' AND Mem_Balance > 5
ORDER BY Mem_Balance DESC;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1  INSERT INTO MEMBERSHIP (Mem_Num, Mem_FName, Mem_LName, Mem_Street, Mem_City, Mem_State, Mem_Zip, Mem_Balance)
2  VALUES (3200, 'Justin', 'Akwuba', '736 Sonic dr', 'Green Hills', 'ZZ', 55123, 45);
3
4
5
6  SELECT *
7  FROM MEMBERSHIP
8  ORDER BY Mem_State DESC;
9
10
11
12 SELECT *
13 FROM MOVIE
14 ORDER BY Movie_Title ASC;
15
16
17 SELECT *
18 FROM MEMBERSHIP
19 WHERE Mem_State = 'TN' AND Mem_Balance > 5
20 ORDER BY Mem_Balance DESC;
21
```

Results Explain Describe Saved SQL History

MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10
111	Stacy	Mann	2789 East Cook Avenue	Murfreesboro	TN	37132	8

3 rows returned in 0.00 seconds Download

Copyright © 1999, 2022, Oracle and/or its affiliates.

For your reference only:
Refer to the SQL code through which you can investigate the data types of a particular table

```
ELECT
   COLUMN_NAME,
   DATA_TYPE,
   DATA_LENGTH,
   DATA_PRECISION,
   DATA_SCALE
ROM ALL_TAB_COLUMNS
HERE TABLE_NAME = 'MEMBERSHIP';
```

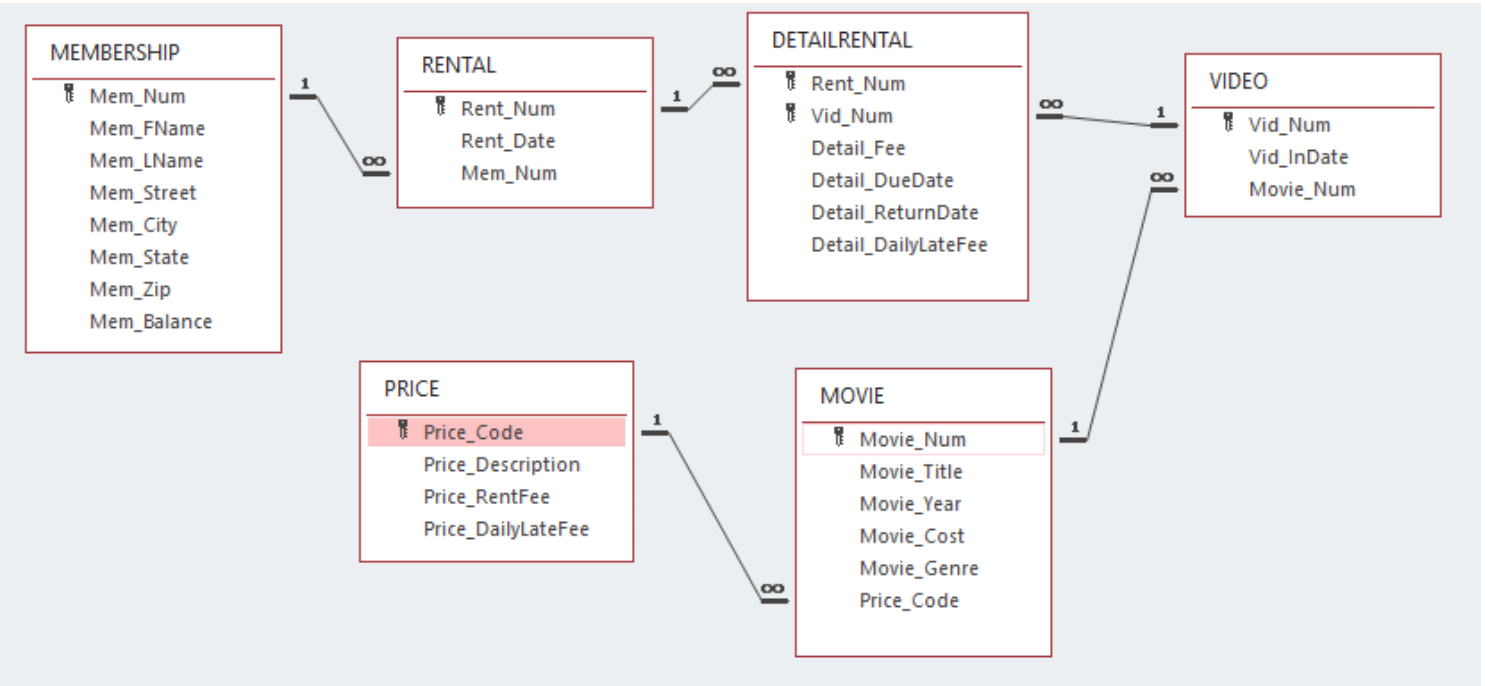
Results | Explain | Describe | Saved SQL | History

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	DATA_SCALE
MEM_NUM	NUMBER	22	-	0
MEM_FNAME	VARCHAR2	50	-	-
MEM_LNAME	VARCHAR2	50	-	-
MEM_STREET	VARCHAR2	50	-	-
MEM_CITY	VARCHAR2	50	-	-
MEM_STATE	CHAR	2	-	-
MEM_ZIP	NUMBER	22	-	0
MEM_BALANCE	NUMBER	22	-	0

3. Making use of the **IN** keyword, retrieve only those member records for members from the **Membership** table whose address is in one of **three zip codes:** 37754, 37132, 38579.

Note: while Oracle Apex may be forgiving in case of data types, make sure you know what is the data type for MEM_ZIP column. In this case it is defined as a number(integer) though VARCHAR or CHAR would have been better

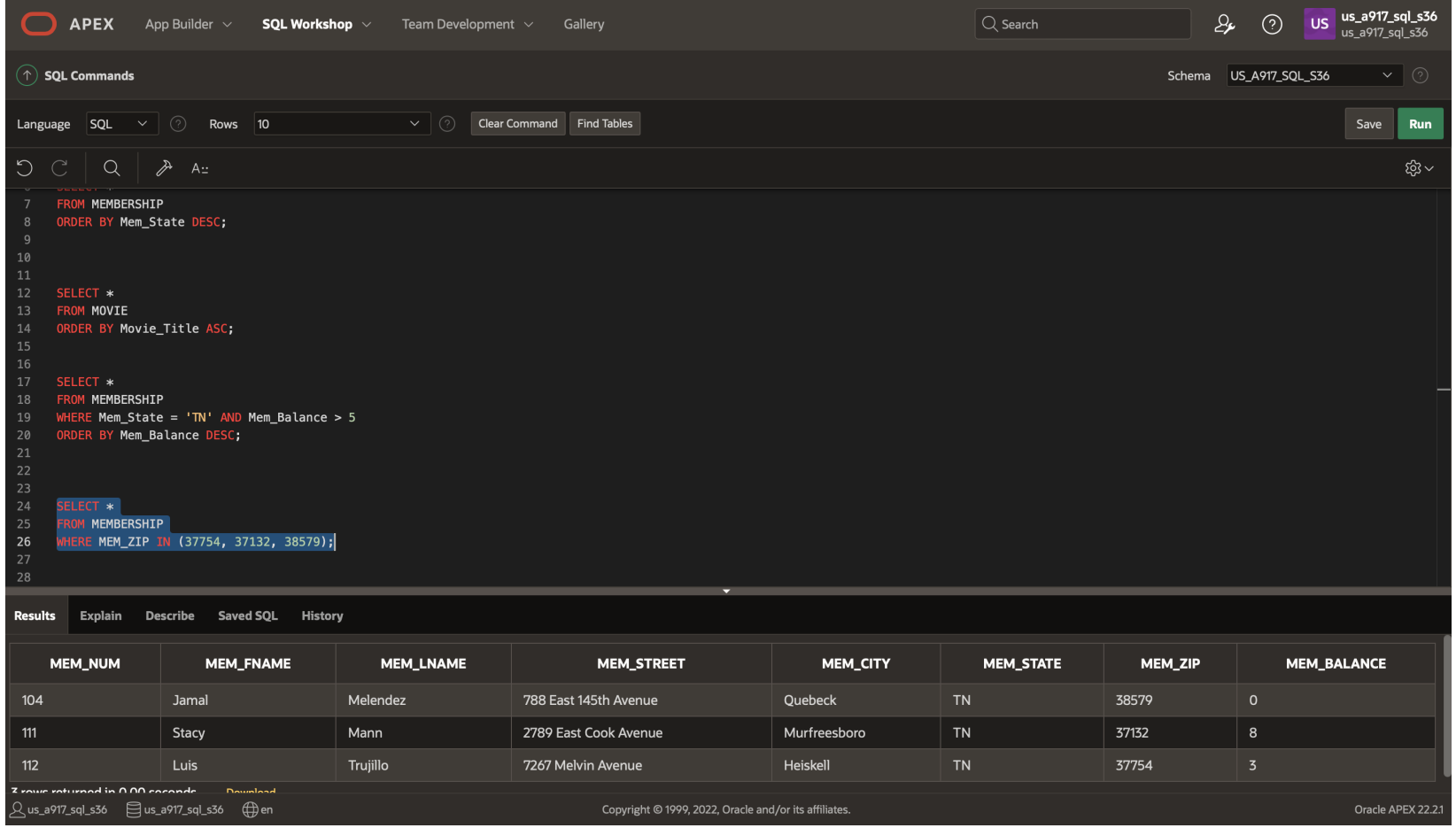
MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
104	Jamal	Melendez	788 East 145th Avenue	Quebeck	TN	38579	0
111	Stacy	Mann	2789 East Cook Avenue	Murfreesboro	TN	37132	8
112	Luis	Trujillo	7267 Melvin Avenue	Heiskell	TN	37754	3



Answer

SQL:

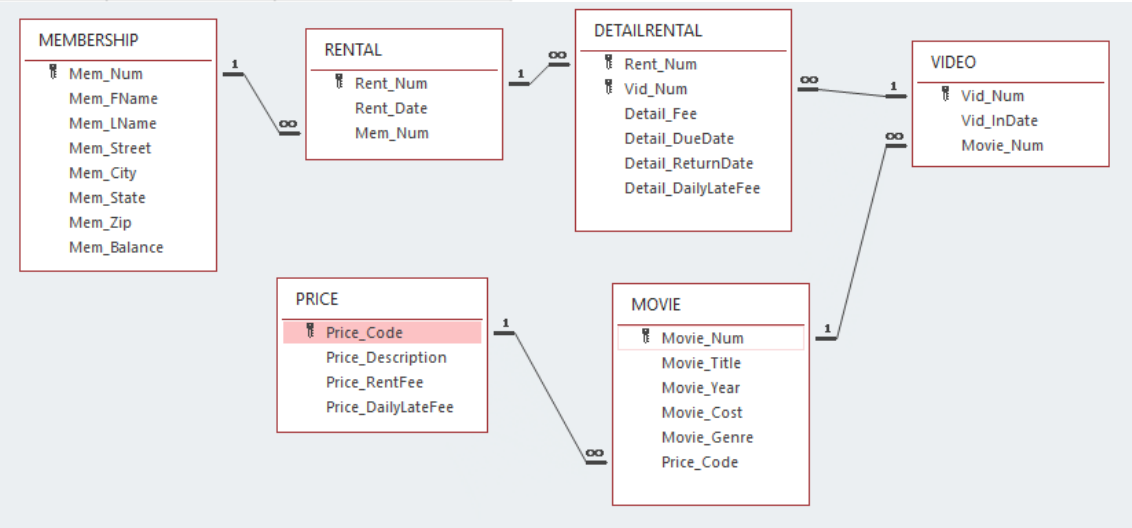
```
SELECT *
FROM MEMBERSHIP
WHERE MEM_ZIP IN (37754, 37132, 38579);
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
FROM MEMBERSHIP
ORDER BY Mem_State DESC;
SELECT *
FROM MOVIE
ORDER BY Movie_Title ASC;
SELECT *
FROM MEMBERSHIP
WHERE Mem_State = 'TN' AND Mem_Balance > 5
ORDER BY Mem_Balance DESC;
SELECT *
FROM MEMBERSHIP
WHERE MEM_ZIP IN (37754, 37132, 38579);
Results
Explain
Describe
Saved SQL
History
MEM_NUM
MEM_FNAME
MEM_LNAME
MEM_STREET
MEM_CITY
MEM_STATE
MEM_ZIP
MEM_BALANCE
104
Jamal
Melendez
788 East 145th Avenue
Quebeck
TN
38579
0
111
Stacy
Mann
2789 East Cook Avenue
Murfreesboro
TN
37132
8
112
Luis
Trujillo
7267 Melvin Avenue
Heiskell
TN
37754
3
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

4. Modify the previous query to now retrieve only those member records for members from the **Membership** table whose address **DOES NOT** belong in one of three zip codes (37754, 37132, 38579).

MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
103	Curt	Knight	4025 Cornell Court	Flatgap	KY	41219	6
105	Iva	McClain	6045 Musket Ball Circle	Summit	KY	42783	15
106	Miranda	Parks	4469 Maxwell Place	Germantown	TN	38183	0
107	Rosario	Elliott	7578 Danner Avenue	Columbia	TN	38402	5
108	Mattie	Guy	4390 Evergreen Street	Lily	KY	40740	0
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10
110	Lewis	Rosales	4524 SouthWind Circle	Counce	TN	38326	0
113	Minnie	Gonzales	6430 Vasili Drive	Williston	TN	38076	0

You may also have an additional row for the new data that you had entered earlier.



Answer

SQL:

```
SELECT *
FROM MEMBERSHIP
WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
14  ORDER BY Movie_Title ASC;
15
16
17  SELECT *
18  FROM MEMBERSHIP
19  WHERE Mem_State = 'TN' AND Mem_Balance > 5
20  ORDER BY Mem_Balance DESC;
21
22
23
24  SELECT *
25  FROM MEMBERSHIP
26  WHERE MEM_ZIP IN (37754, 37132, 38579);
27
28
29
30
31  SELECT *
32  FROM MEMBERSHIP
33  WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
34
35
```

Results Explain Describe Saved SQL History

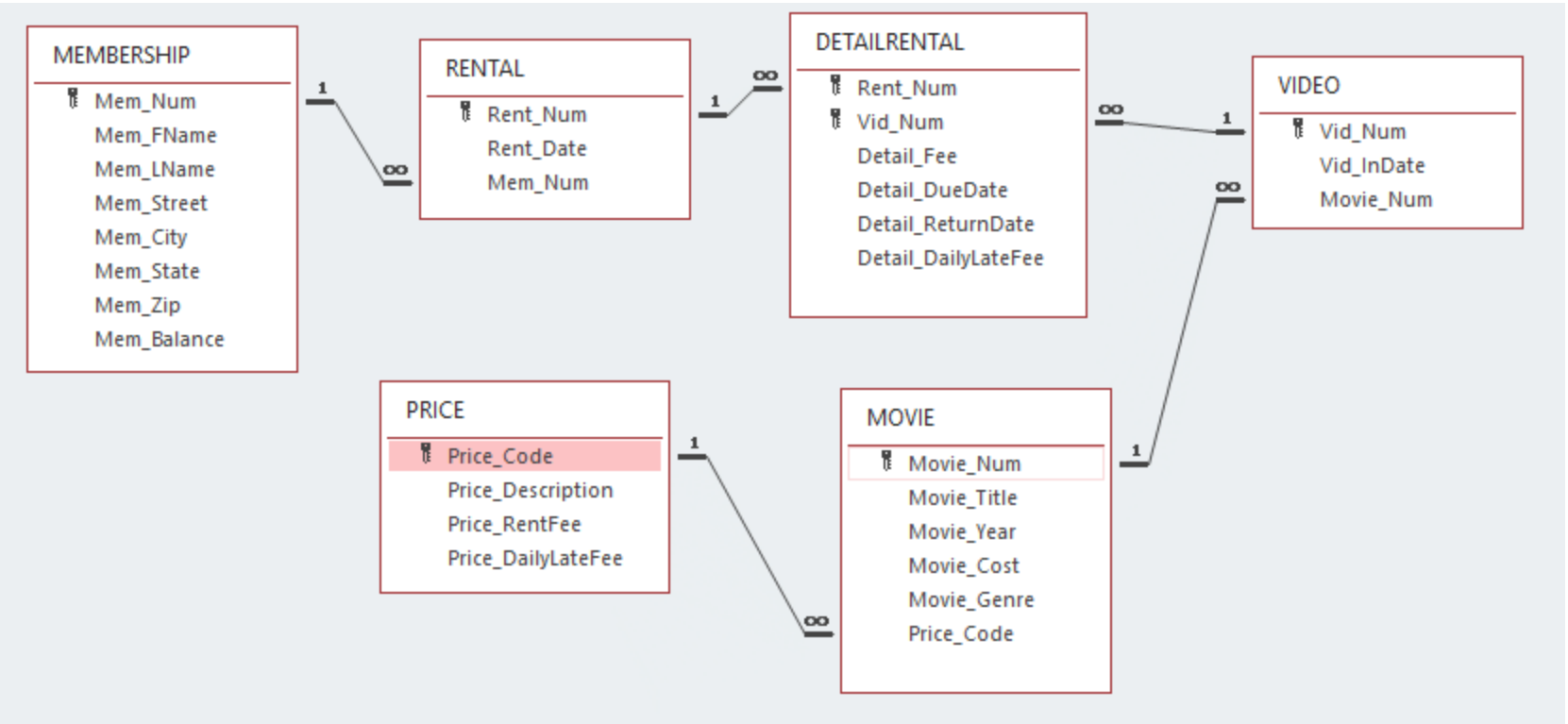
3500	Justin	-	736 Sonic dr	Green Hills	ZZ	55123	45
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
103	Curt	Knight	4025 Cornell Court	Flatgap	KY	41219	6
105	Iva	McClain	6045 Musket Ball Circle	Summit	KY	42783	15
106	Miranda	Parks	4469 Maxwell Place	Germantown	TN	38183	0

Copyright © 1999, 2022, Oracle and/or its affiliates.

5. Modify the previous query to now count the number of member records for members whose address DOES NOT belong in one of three zip codes (37754, 37132, 38579).
Name the column that has the count as **"Number of Members"**

Number of Members
9

Your number may say 10 instead of 9.



Answer

SQL:

```
SELECT COUNT(*) AS "Number of Members"
FROM MEMBERSHIP
WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
26  WHERE MEM_ZIP IN (37754, 37132, 38579);
27
28
29
30
31  SELECT *
32  FROM MEMBERSHIP
33  WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
34
35
36
37
38  SELECT COUNT(*) AS "Number of Members"
39  FROM MEMBERSHIP
40  WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
41
42
43
```

Results Explain Describe Saved SQL History

Number of Members
11

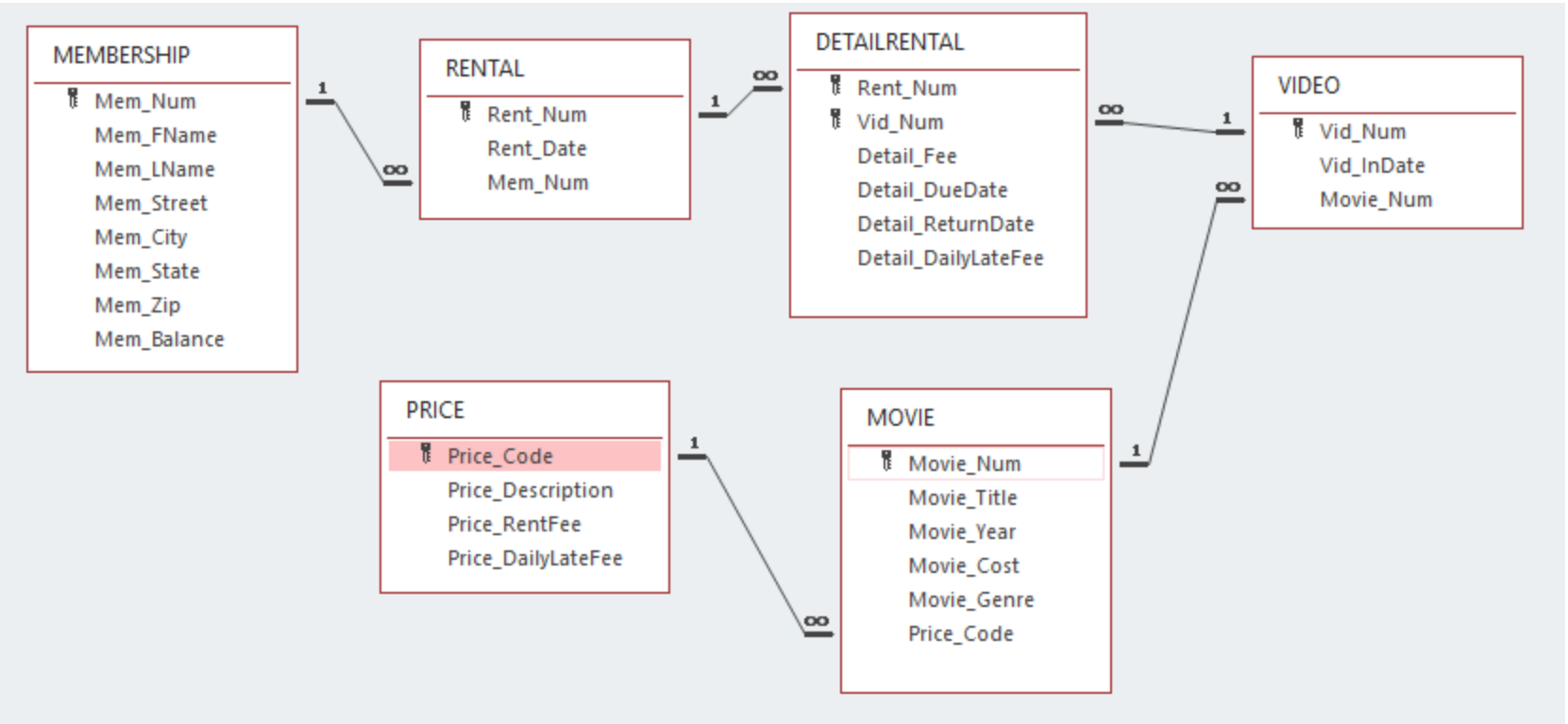
1 rows returned in 0.00 seconds Download

Copyright © 1999, 2022, Oracle and/or its affiliates.

6. Count the number of members who reside in each state. The columns should be renamed using aliases as shown below. The count should be sorted in descending order.

Member State	Number of Members
TN	9
KY	3

You should also have an additional row for the new data that you had entered earlier.



Answer

SQL:

```
SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
FROM MEMBERSHIP
GROUP BY Mem_State
ORDER BY "Number of Members" DESC;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
29
30
31  SELECT *
32  FROM MEMBERSHIP
33  WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
34
35
36
37
38  SELECT COUNT(*) AS "Number of Members"
39  FROM MEMBERSHIP
40  WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
41
42
43
44
45  SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
46  FROM MEMBERSHIP
47  GROUP BY Mem_State
48  ORDER BY "Number of Members" DESC;
49
```

Results Explain Describe Saved SQL History

State	Number of Members
TN	9
KY	3
ZZ	2

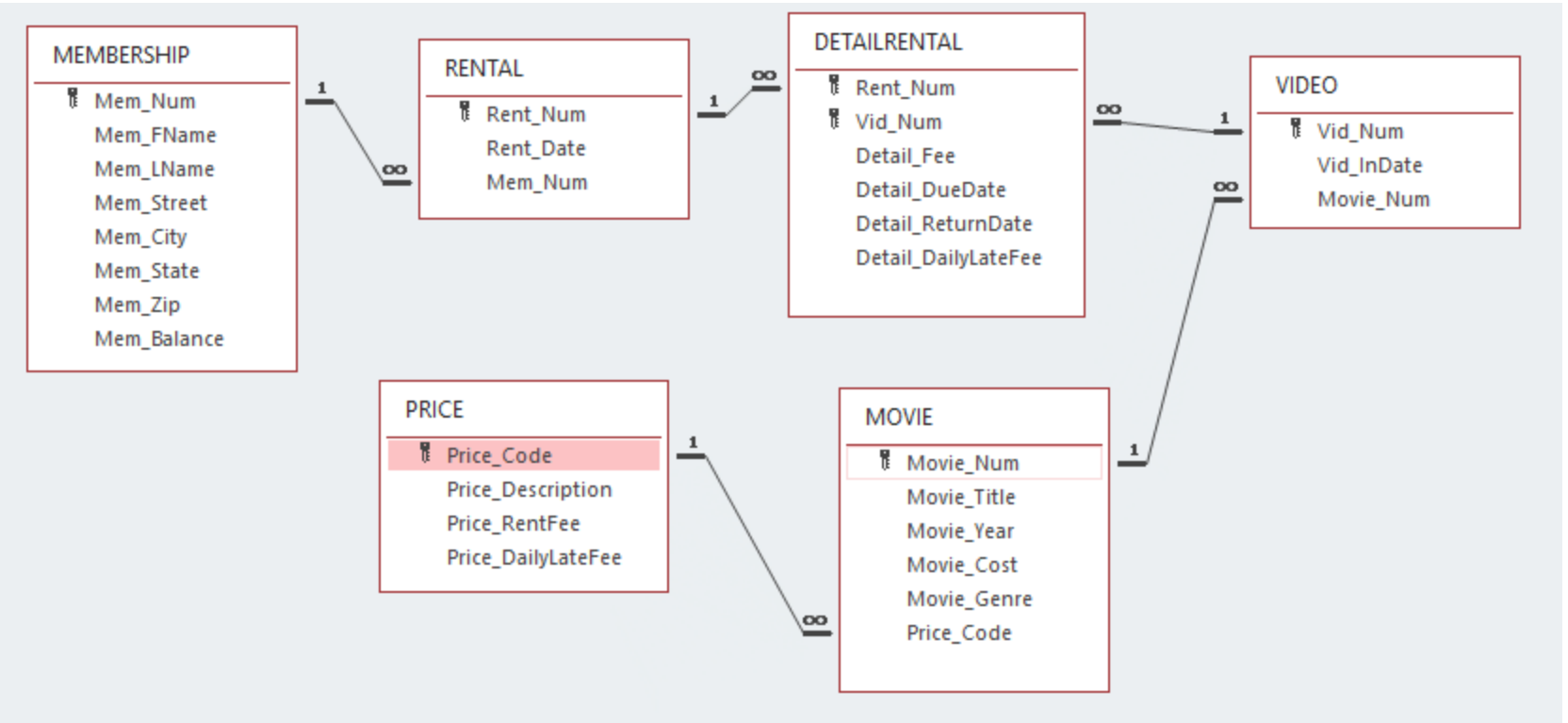
3 rows returned in 0.00 seconds Download

Copyright © 1999, 2022, Oracle and/or its affiliates.

7. Modify the previous query to now only show records where the total number of members are less than **5**.

Member State	Number of Members
KY	3

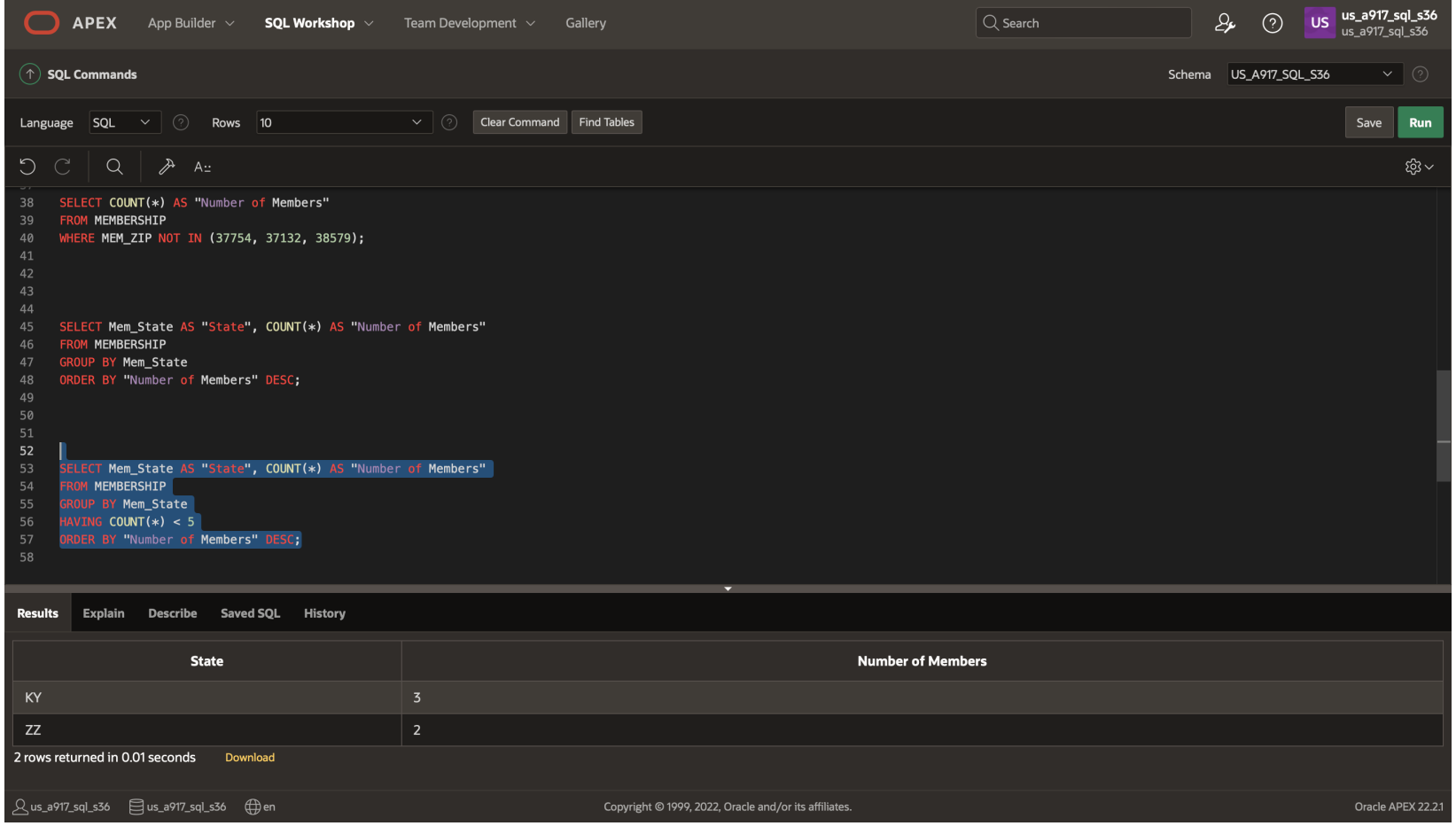
You should also have an additional row for the new data that you had entered earlier.



Answer

SQL:

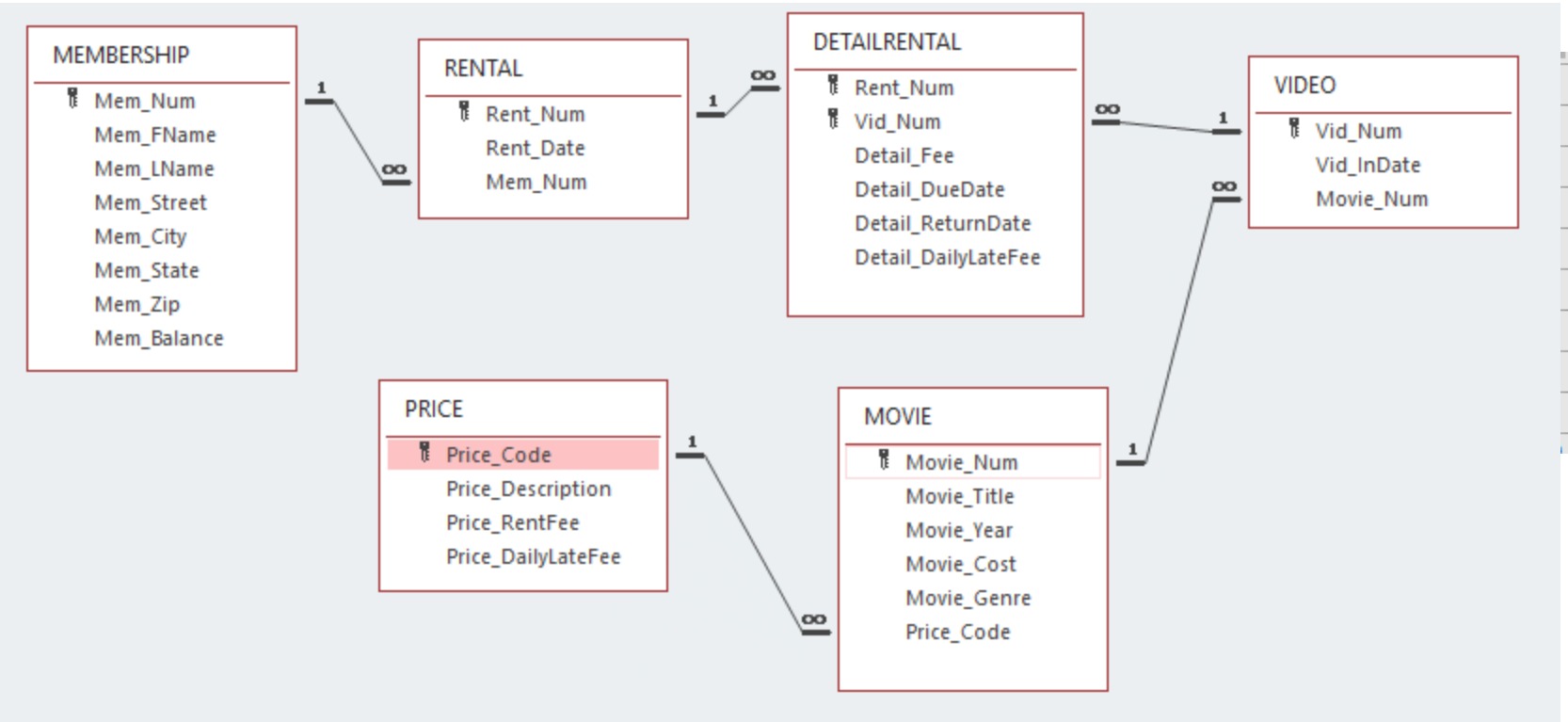
```
SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
FROM MEMBERSHIP
GROUP BY Mem_State
HAVING COUNT(*) < 5
ORDER BY "Number of Members" DESC;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
SELECT COUNT(*) AS "Number of Members"
FROM MEMBERSHIP
WHERE MEM_ZIP NOT IN (37754, 37132, 38579);
SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
FROM MEMBERSHIP
GROUP BY Mem_State
ORDER BY "Number of Members" DESC;
SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
FROM MEMBERSHIP
GROUP BY Mem_State
HAVING COUNT(*) < 5
ORDER BY "Number of Members" DESC;
Results
Explain
Describe
Saved SQL
History
State
Number of Members
KY
3
ZZ
2
2 rows returned in 0.01 seconds
Download
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

8. Retrieve only those records for members from the Membership table that have the word **"Street"** ending in the Street Address column (**Mem_Street**). You should get the following result:

MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
108	Mattie	Guy	4390 Evergreen Street	Lily	KY	40740	0
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10

Your result may have an additional row depending on the data you had entered earlier.



Some meta-data for Membership table

Column Name	Data Type	Nullable	Default	Primary Key
MEM_NUM	NUMBER	No		Yes
MEM_FNAME	VARCHAR2(50)	Yes		
MEM_LNAME	VARCHAR2(50)	Yes		
MEM_STREET	VARCHAR2(50)	Yes		
MEM_CITY	VARCHAR2(50)	Yes		
MEM_STATE	CHAR(2)	Yes		
MEM_ZIP	NUMBER	Yes		
MEM_BALANCE	NUMBER	Yes		

Answer

SQL:

```
SELECT *
FROM MEMBERSHIP
WHERE Mem_Street LIKE '%Street';
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
44
45  SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
46  FROM MEMBERSHIP
47  GROUP BY Mem_State
48  ORDER BY "Number of Members" DESC;
49
50
51
52
53  SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
54  FROM MEMBERSHIP
55  GROUP BY Mem_State
56  HAVING COUNT(*) < 5
57  ORDER BY "Number of Members" DESC;
58
59
60
61  SELECT *
62  FROM MEMBERSHIP
63  WHERE Mem_Street LIKE '%Street';
64
65
```

Results Explain Describe Saved SQL History

MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
108	Mattie	Guy	4390 Evergreen Street	Lily	KY	40740	0
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10

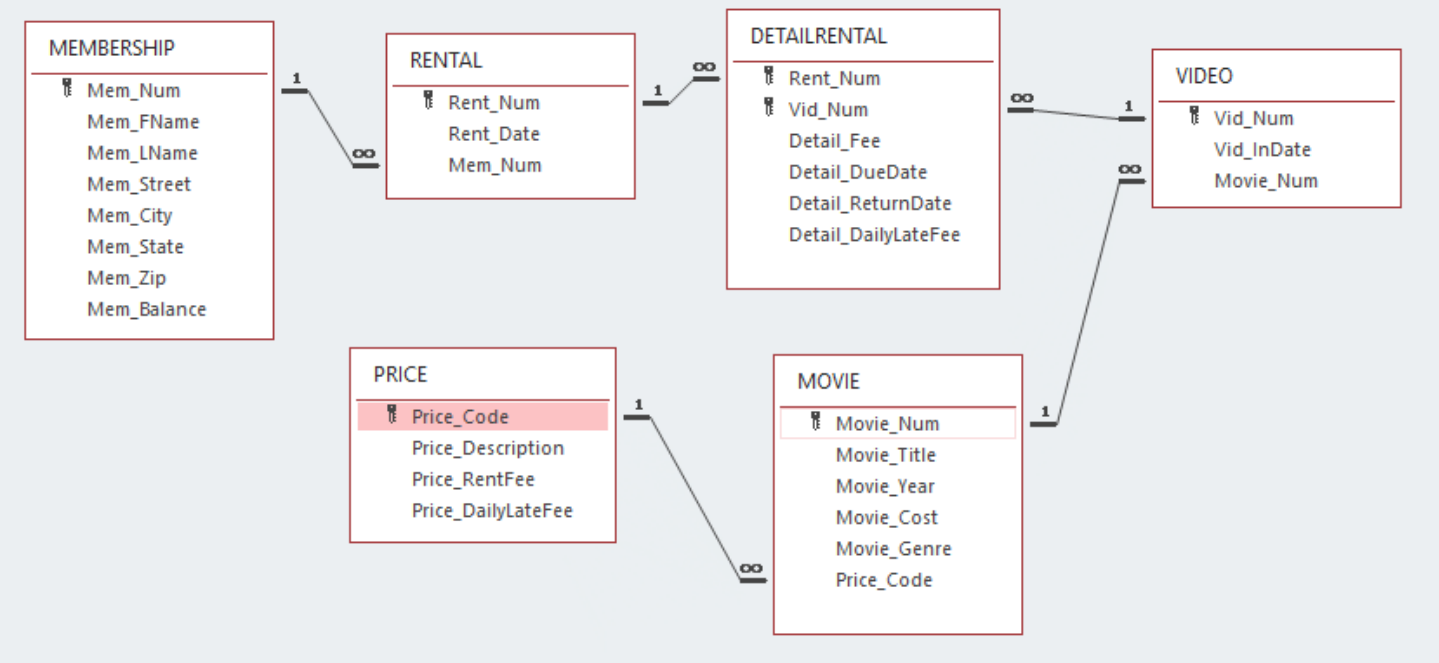
2 rows returned in 0.00 seconds Download

Copyright © 1999, 2022, Oracle and/or its affiliates.

9. Retrieve the Member First Name, Last Name, and Street Address for only those members from the Membership table that have the letter **'c'** anywhere in their street address i.e., the letter **'c'** can be found anywhere in the **Mem_Street** column. You should get the following result:

MEM_FNAME	MEM_LNAME	MEM_STREET
Tami	Dawson	2632 Takli Circle
Iva	McClain	6045 Musket Ball Circle
Miranda	Parks	4469 Maxwell Place
Lewis	Rosales	4524 SouthWind Circle

Depending on the data you had entered, you might have an additional row.



Answer

SQL:

```
SELECT Mem_FName, Mem_LName, Mem_Street
FROM MEMBERSHIP
WHERE Mem_Street LIKE '%c%';
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
50
51
52
53  SELECT Mem_State AS "State", COUNT(*) AS "Number of Members"
54  FROM MEMBERSHIP
55  GROUP BY Mem_State
56  HAVING COUNT(*) < 5
57  ORDER BY "Number of Members" DESC;
58
59
60
61  SELECT *
62  FROM MEMBERSHIP
63  WHERE Mem_Street LIKE '%Street';
64
65
66
67  SELECT Mem_FName, Mem_LName, Mem_Street
68  FROM MEMBERSHIP
69  WHERE Mem_Street LIKE '%c%';
70
71
```

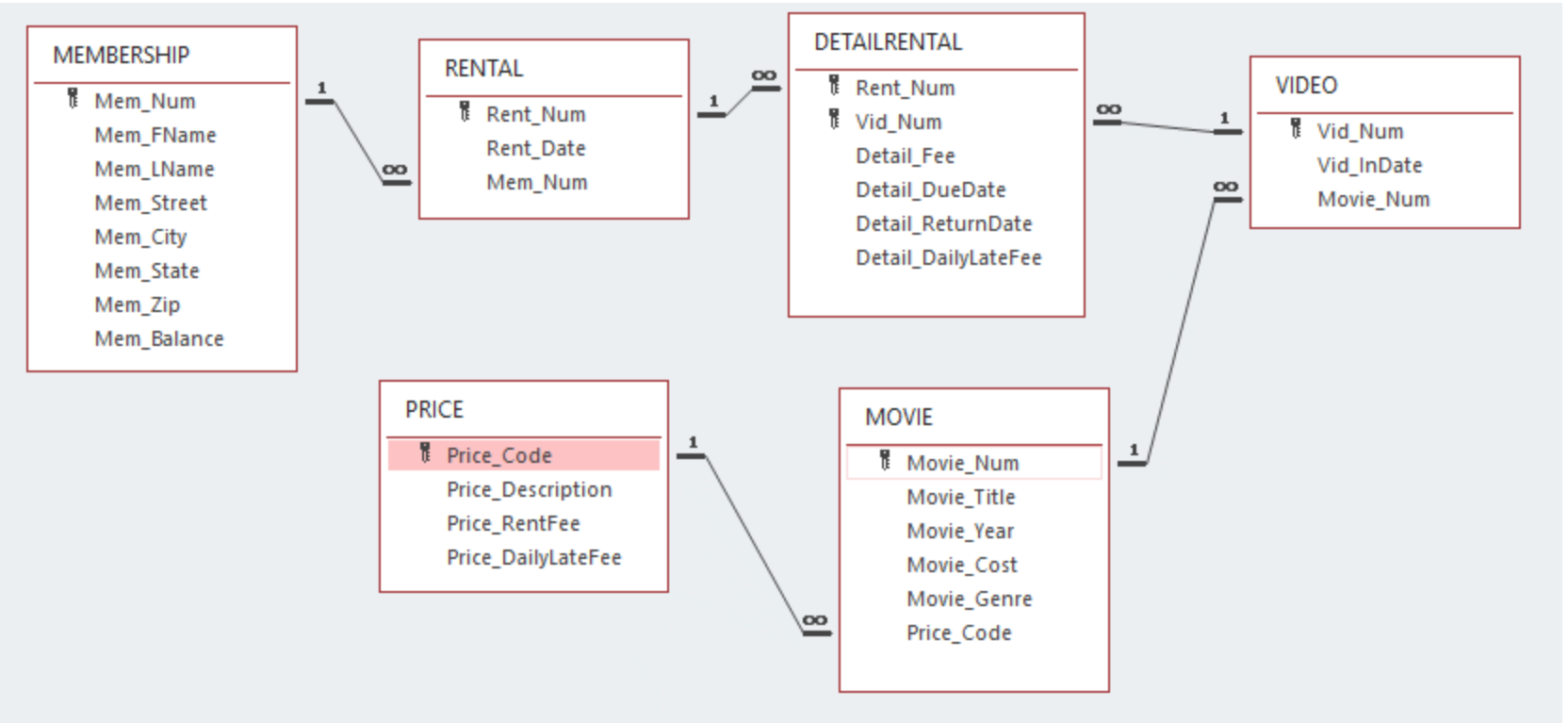
Results Explain Describe Saved SQL History

Justin	-	736 Sonic dr
Tami	Dawson	2632 Takli Circle
Iva	McClain	6045 Musket Ball Circle
Miranda	Parks	4469 Maxwell Place
Lewis	Rosales	4524 SouthWind Circle

Copyright © 1999, 2022, Oracle and/or its affiliates.

10. Retrieve only those records for members from the Membership table with Zip code that **begins** with 37. **Note:** we do not want zip codes that may have 37 in between e.g., we do not want zip code such as 23768, 45378, or 56837 and so forth.

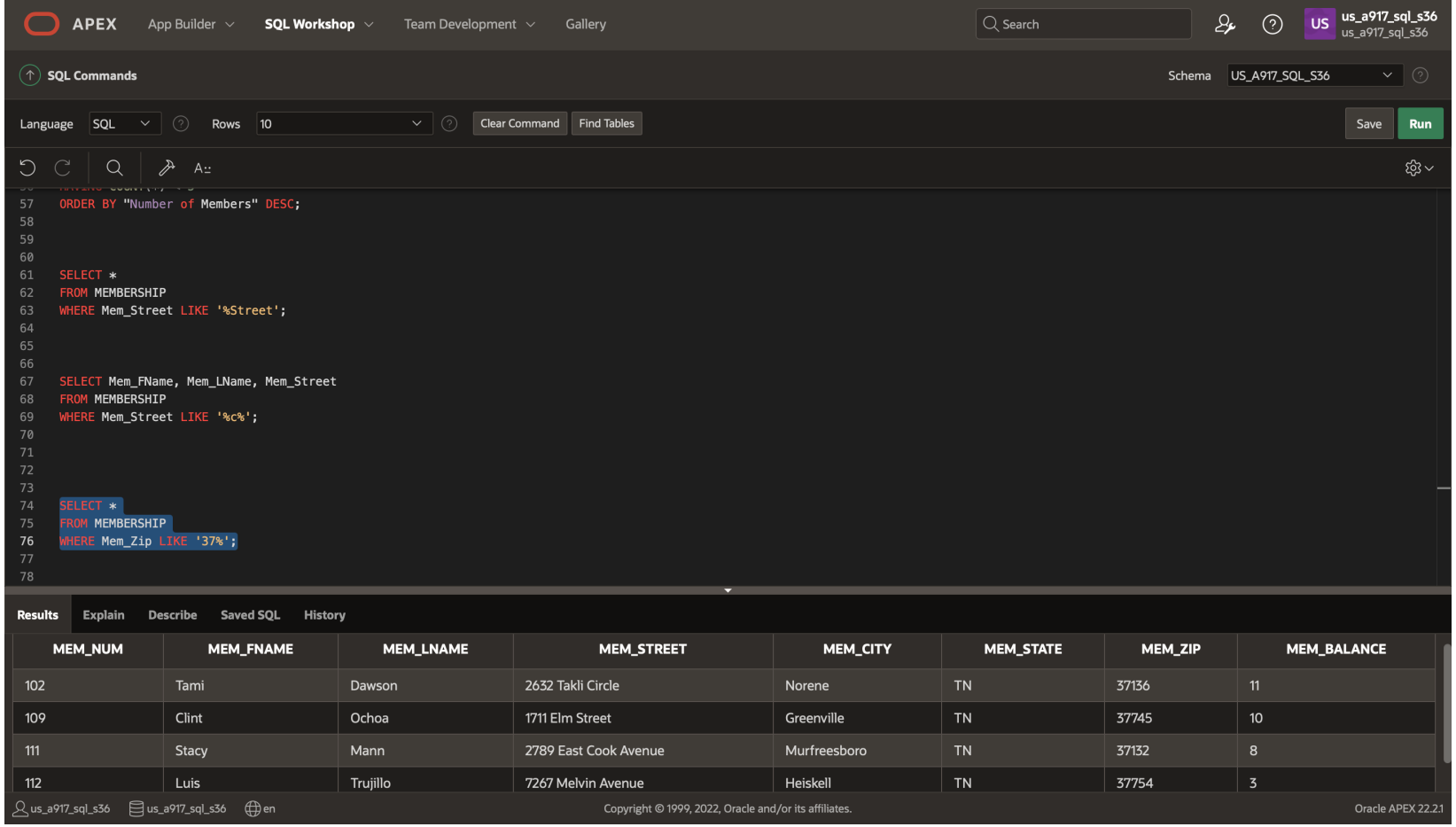
MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_STREET	MEM_CITY	MEM_STATE	MEM_ZIP	MEM_BALANCE
102	Tami	Dawson	2632 Takli Circle	Norene	TN	37136	11
109	Clint	Ochoa	1711 Elm Street	Greenville	TN	37745	10
111	Stacy	Mann	2789 East Cook Avenue	Murfreesboro	TN	37132	8
112	Luis	Trujillo	7267 Melvin Avenue	Heiskell	TN	37754	3



Answer

SQL:

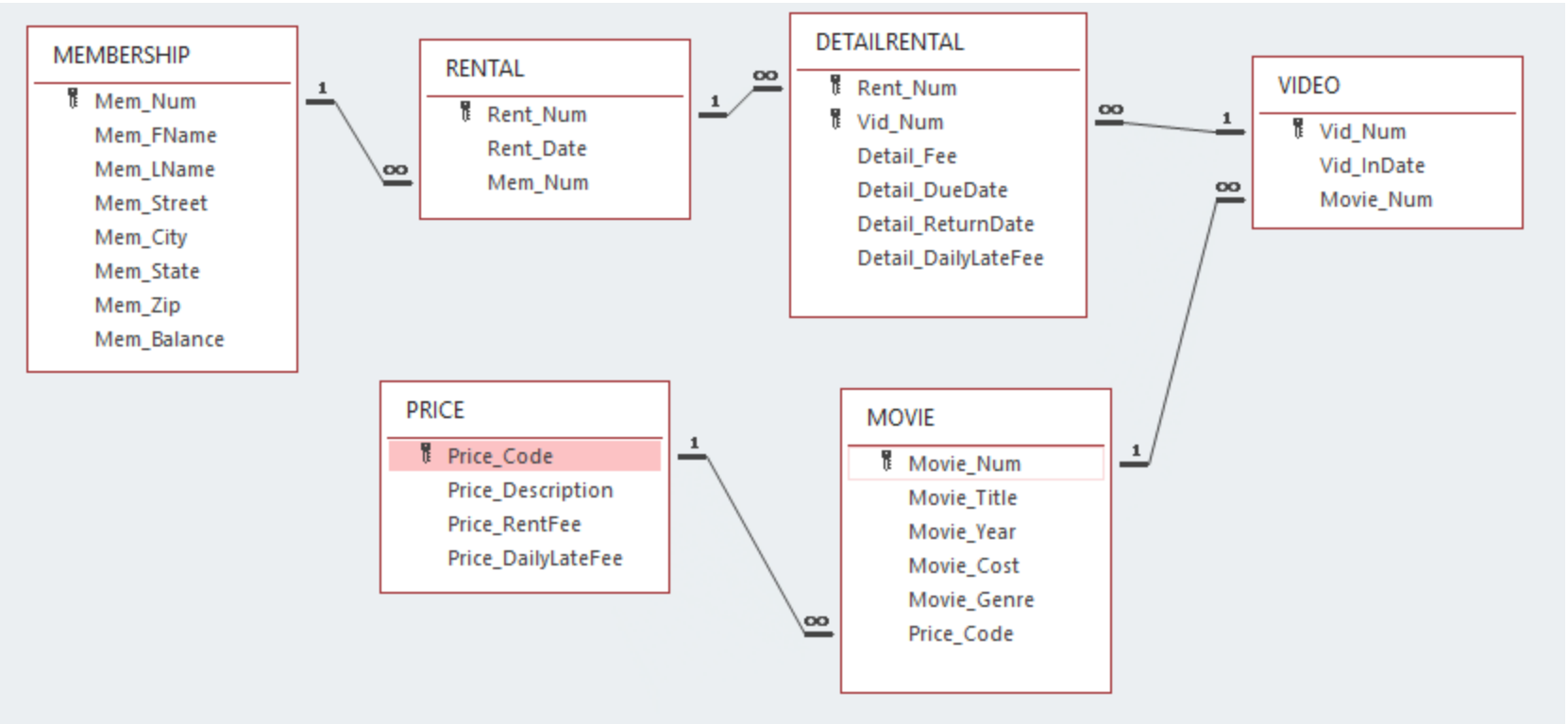
```
SELECT *
FROM MEMBERSHIP
WHERE Mem_Zip LIKE '37%';
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
ORDER BY "Number of Members" DESC;
SELECT *
FROM MEMBERSHIP
WHERE Mem_Street LIKE '%Street';
SELECT Mem_FName, Mem_LName, Mem_Street
FROM MEMBERSHIP
WHERE Mem_Street LIKE '%c%';
SELECT *
FROM MEMBERSHIP
WHERE Mem_Zip LIKE '37%';
Results
Explain
Describe
Saved SQL
History
MEM_NUM
MEM_FNAME
MEM_LNAME
MEM_STREET
MEM_CITY
MEM_STATE
MEM_ZIP
MEM_BALANCE
102
Tami
Dawson
2632 Takli Circle
Norene
TN
37136
11
109
Clint
Ochoa
1711 Elm Street
Greenville
TN
37745
10
111
Stacy
Mann
2789 East Cook Avenue
Murfreesboro
TN
37132
8
112
Luis
Trujillo
7267 Melvin Avenue
Heiskell
TN
37754
3
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

11. Retrieve only the Member State from the table Membership. The column **Mem_State** should be renamed using an alias “Member State”. The results should only show ***unique*** records.

Member State
TN
KY

You should also have an additional row for the new data that you had entered earlier.



Answer

SQL:

```
SELECT DISTINCT Mem_State AS "Member State"
FROM MEMBERSHIP;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
62  FROM MEMBERSHIP
63  WHERE Mem_Street LIKE '%Street';
64
65
66
67  SELECT Mem_FName, Mem_LName, Mem_Street
68  FROM MEMBERSHIP
69  WHERE Mem_Street LIKE '%c%';
70
71
72
73
74  SELECT *
75  FROM MEMBERSHIP
76  WHERE Mem_Zip LIKE '37%';
77
78
79
80  SELECT DISTINCT Mem_State AS "Member State"
81  FROM MEMBERSHIP;
82
```

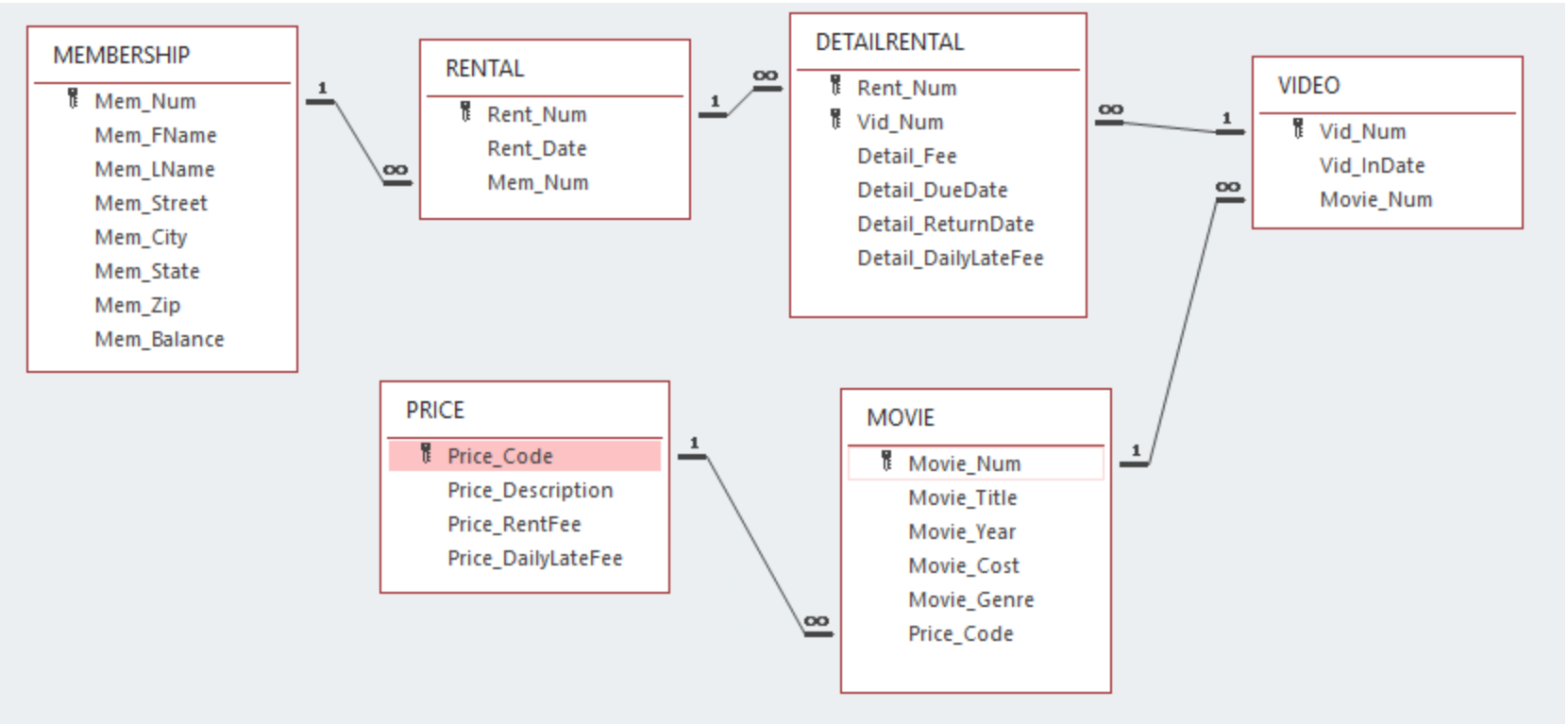
Results Explain Describe Saved SQL History

Member State
TN
KY
ZZ

3 rows returned in 0.00 seconds Download

12. Create a query that shows the number of rentals for each member State. Use an alias to rename the column Mem_State as “Member State” and the count of the number of rental records associated with each member state as “Number of Rentals”. The Number of Rentals should be in ***descending*** order. **Hint:** You will need to join the Membership and Rental tables.

Member State	Number of Rentals
TN	6
KY	3



Answer

SQL:

```
SELECT M.Mem_State AS "Member State", COUNT(R.Rent_Num) AS "Number of Rentals"
FROM MEMBERSHIP M
LEFT JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
GROUP BY M.Mem_State
ORDER BY "Number of Rentals" DESC;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
69  WHERE Mem_Street LIKE '%c%';
70
71
72
73
74  SELECT *
75  FROM MEMBERSHIP
76  WHERE Mem_Zip LIKE '37%';
77
78
79
80  SELECT DISTINCT Mem_State AS "Member State"
81  FROM MEMBERSHIP;
82
83
84  SELECT M.Mem_State AS "Member State", COUNT(R.Rent_Num) AS "Number of Rentals"
85  FROM MEMBERSHIP M
86  LEFT JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
87  GROUP BY M.Mem_State
88  ORDER BY "Number of Rentals" DESC;
89
90
```

Results Explain Describe Saved SQL History

Member State	Number of Rentals
TN	6
KY	3
ZZ	0

3 rows returned in 0.01 seconds Download

Copyright © 1999, 2022, Oracle and/or its affiliates.

13. Select **Price Description** from the **Price** table and **Movie Title, Movie Year** and **Movie Cost** from the **Movie** table. **Sort** the data on **Price Description** in *descending* order (**Hint:** you will need to join the tables Price and Movie). You should get the following result:

PRICE_DESCRIPTION	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST
Standard	Gone With the Wind	1939	19.99
Standard	Smokey Mountain Wildlife	2006	59.95
Standard	What He Doesn't Know	2008	58.29
Standard	Time to Burn	2007	45.49
New Release	Fantastic Beasts and Where to Find Them	2016	29.99
New Release	Doctor Strange	2016	29.99
New Release	Constant Companion	2010	89.95
New Release	Beatnik Fever	2009	29.95
New Release	Richard Goodhope	2010	59.95
New Release	The Cesar Family Christmas	2009	39.95
Discount	Where Hope Dies	2000	25.49

Note: If your data is slightly different, depending on what data was entered by you previously, it is okay. The query should be correct so that it will return the results based on what data exists in the database.

Answer

SQL:

```
SELECT P.Price_Description, M.Movie_Title, M.Movie_Year, M.Movie_Cost
FROM PRICE P
JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
ORDER BY P.Price_Description DESC;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
77
78
79
80  SELECT DISTINCT Mem_State AS "Member State"
81  FROM MEMBERSHIP;
82
83
84  SELECT M.Mem_State AS "Member State", COUNT(R.Rent_Num) AS "Number of Rentals"
85  FROM MEMBERSHIP M
86  LEFT JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
87  GROUP BY M.Mem_State
88  ORDER BY "Number of Rentals" DESC;
89
90
91
92
93  SELECT P.Price_Description, M.Movie_Title, M.Movie_Year, M.Movie_Cost
94  FROM PRICE P
95  JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
96  ORDER BY P.Price_Description DESC;
97
```

Results Explain Describe Saved SQL History

Standard	What He Doesn't Know	2008	58.29
New Release	The Cesar Family Christmas	2009	39.95
New Release	Richard Goodhope	2010	59.95
New Release	Beatnik Fever	2009	29.95
New Release	Constant Companion	2010	89.95

Copyright © 1999, 2022, Oracle and/or its affiliates.

14. Using SQL insert statement, insert a new record for the table **Movie** for the 2019 movie **Avengers: Endgame** using the following details. Notice that the Movie_Num assigned is **1260**. The Movie Year is 2019, Movie Cost is 39.95, Movie Genre is Fantasy, and Price Code is 2.

MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST	MOVIE_GENRE	PRICE_CODE
1260	Avengers: Endgame	2019	39.95	FANTASY	2

For your reference: Refer to the schema for the Movie table below that shows the columns and data types for each column. Make sure that you match the values to the data type e.g., Varchar2, Number etc.

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	MOVIE_NUM	NUMBER	22		0	No		
2	MOVIE_TITLE	VARCHAR2	50			Yes	Byte	
3	MOVIE_YEAR	NUMBER	22		0	Yes		
4	MOVIE_COST	NUMBER	22	5	2	Yes		
5	MOVIE_GENRE	VARCHAR2	20			Yes	Byte	
6	PRICE_CODE	NUMBER	22		0	Yes		

Answer

SQL:

```
INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
VALUES (1260, 'Avengers: Endgame', 2019, 39.95, 'Fantasy', 2);
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
87  GROUP BY M.Mem_State
88  ORDER BY "Number of Rentals" DESC;
89
90
91
92
93  SELECT P.Price_Description, M.Movie_Title, M.Movie_Year, M.Movie_Cost
94  FROM PRICE P
95  JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
96  ORDER BY P.Price_Description DESC;
97
98
99
100
101
102 INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
103 VALUES (1260, 'Avengers: Endgame', 2019, 39.95, 'Fantasy', 2);
104
105
106
107
108
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.

0.00 seconds
```


Copyright © 1999, 2022, Oracle and/or its affiliates.

15. Using SQL insert statement, insert a new record for the table **Movie** for any other Movie. You can enter a fictional name of the Movie or use a real name.

Use **Movie_Num** as **1261**. Set **Price_Code** for this Movie as **Null**. You can set the Movie Cost on your own, and Movie Genre accordingly.

For your reference: Refer to the schema for the Movie table below that shows the columns and data types for each column. Make sure that you match the values to the data type e.g. Varchar2, Number etc.

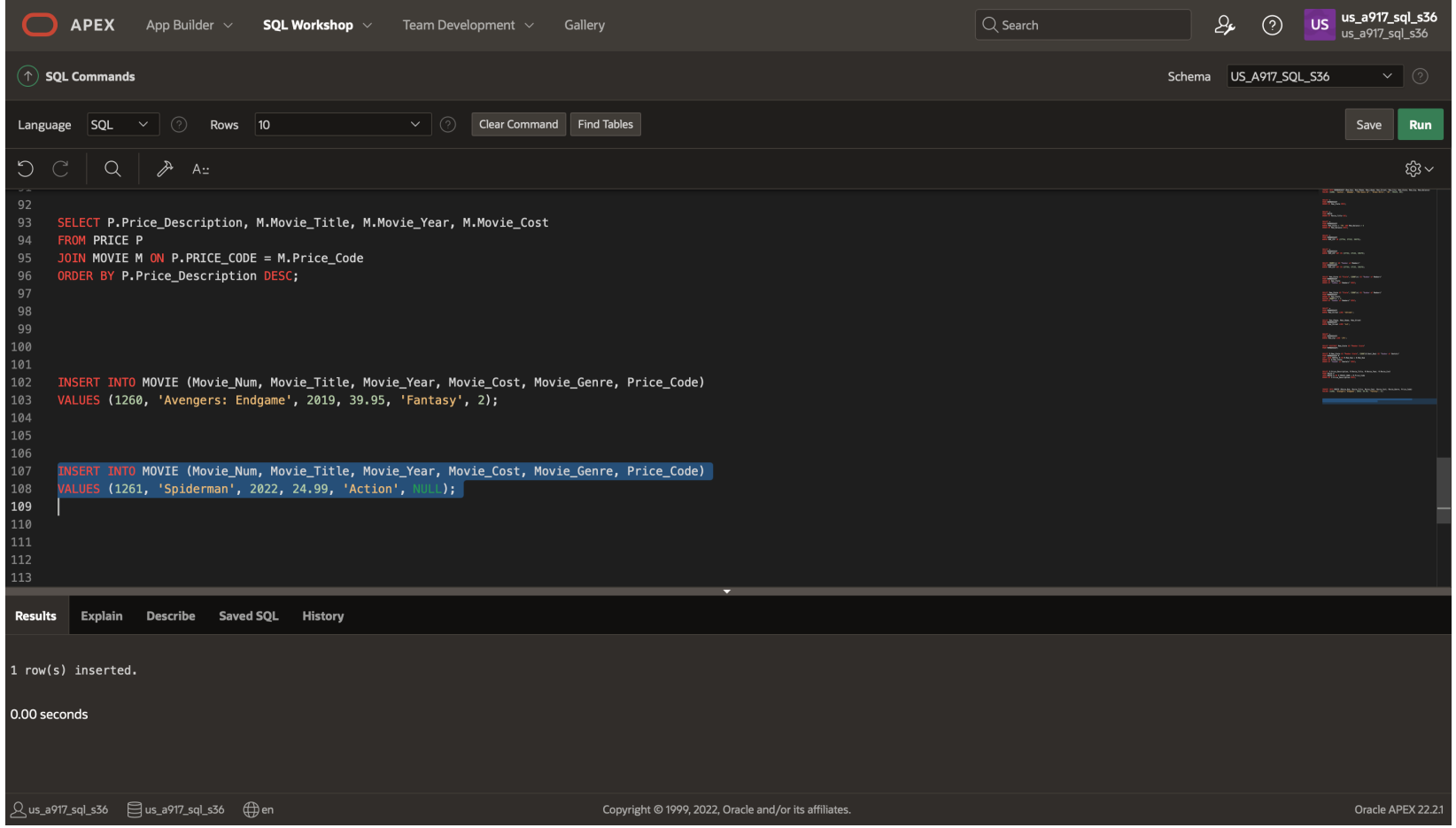
Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	MOVIE_NUM	NUMBER	22		0	No		
2	MOVIE_TITLE	VARCHAR2	50			Yes	Byte	
3	MOVIE_YEAR	NUMBER	22		0	Yes		
4	MOVIE_COST	NUMBER	22	5	2	Yes		
5	MOVIE_GENRE	VARCHAR2	20			Yes	Byte	
6	PRICE_CODE	NUMBER	22		0	Yes		

Answer

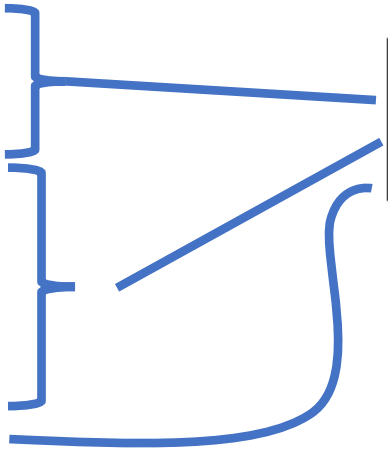
SQL:

```
INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
VALUES (1261, 'Spiderman', 2022, 24.99, 'Action', NULL);
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
92
93 SELECT P.Price_Description, M.Movie_Title, M.Movie_Year, M.Movie_Cost
94 FROM PRICE P
95 JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
96 ORDER BY P.Price_Description DESC;
97
98
99
100
101
102 INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
103 VALUES (1260, 'Avengers: Endgame', 2019, 39.95, 'Fantasy', 2);
104
105
106
107 INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
108 VALUES (1261, 'Spiderman', 2022, 24.99, 'Action', NULL);
109
110
111
112
113
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.00 seconds
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

16. Modify the previous query (or write a new query) to show the **average movie cost** for each **Price Description**. The column showing the average movie cost should be labeled as ***"Average Movie Cost"*** and should show no more than **two decimal places**. The result should be as follows, **sorted** on average cost in ***descending*** order. **Hint:** Use the Cast or Round function to limit the decimal places for the average movie cost to only two.

PRICE_DESCRIPTION	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST
Standard	Gone With the Wind	1939	19.99
Standard	Smokey Mountain Wildlife	2006	59.95
Standard	What He Doesn't Know	2008	58.29
Standard	Time to Burn	2007	45.49
New Release	Fantastic Beasts and Where to Find Them	2016	29.99
New Release	Doctor Strange	2016	29.99
New Release	Constant Companion	2010	89.95
New Release	Beatnik Fever	2009	29.95
New Release	Richard Goodhope	2010	59.95
New Release	The Cesar Family Christmas	2009	39.95
Discount	Where Hope Dies	2000	25.49



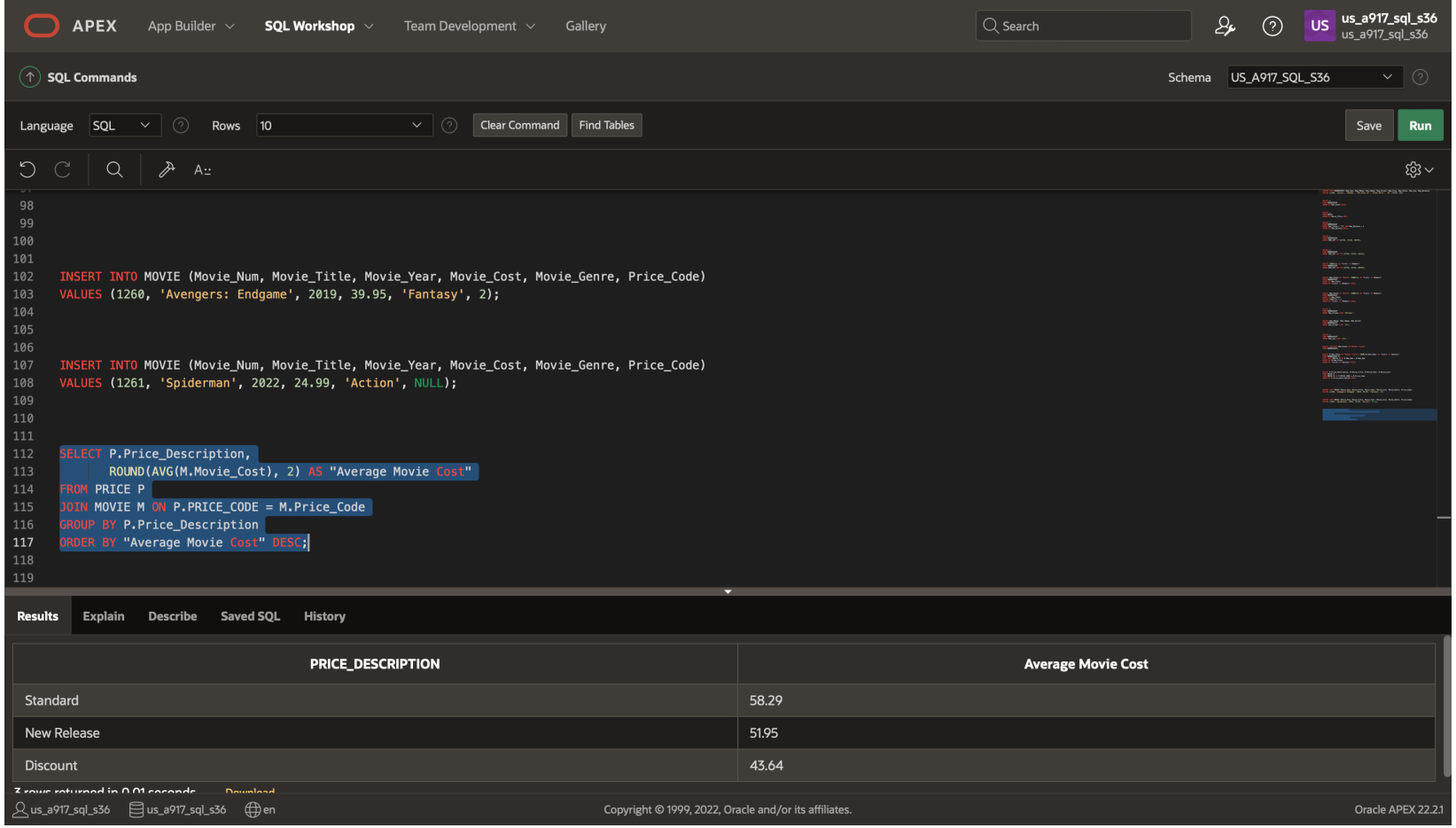
Price Description	Average Movie Cost
Standard	45.93
New Release	46.63
Discount	25.49

Note: Your data probably would be slightly different since you entered new data. However, the aggregation should be consistent with the data.

Answer

SQL:

```
SELECT P.Price_Description,
    ROUND(AVG(M.Movie_Cost), 2) AS "Average Movie Cost"
FROM PRICE P
JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
GROUP BY P.Price_Description
ORDER BY "Average Movie Cost" DESC;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
VALUES (1260, 'Avengers: Endgame', 2019, 39.95, 'Fantasy', 2);
INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
VALUES (1261, 'Spiderman', 2022, 24.99, 'Action', NULL);
SELECT P.Price_Description,
ROUND(AVG(M.Movie_Cost), 2) AS "Average Movie Cost"
FROM PRICE P
JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
GROUP BY P.Price_Description
ORDER BY "Average Movie Cost" DESC;
Results
Explain
Describe
Saved SQL
History
PRICE_DESCRIPTION
Average Movie Cost
Standard
58.29
New Release
51.95
Discount
43.64
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

17. Now retrieve all movie records from Movie table where the Price Code is null.

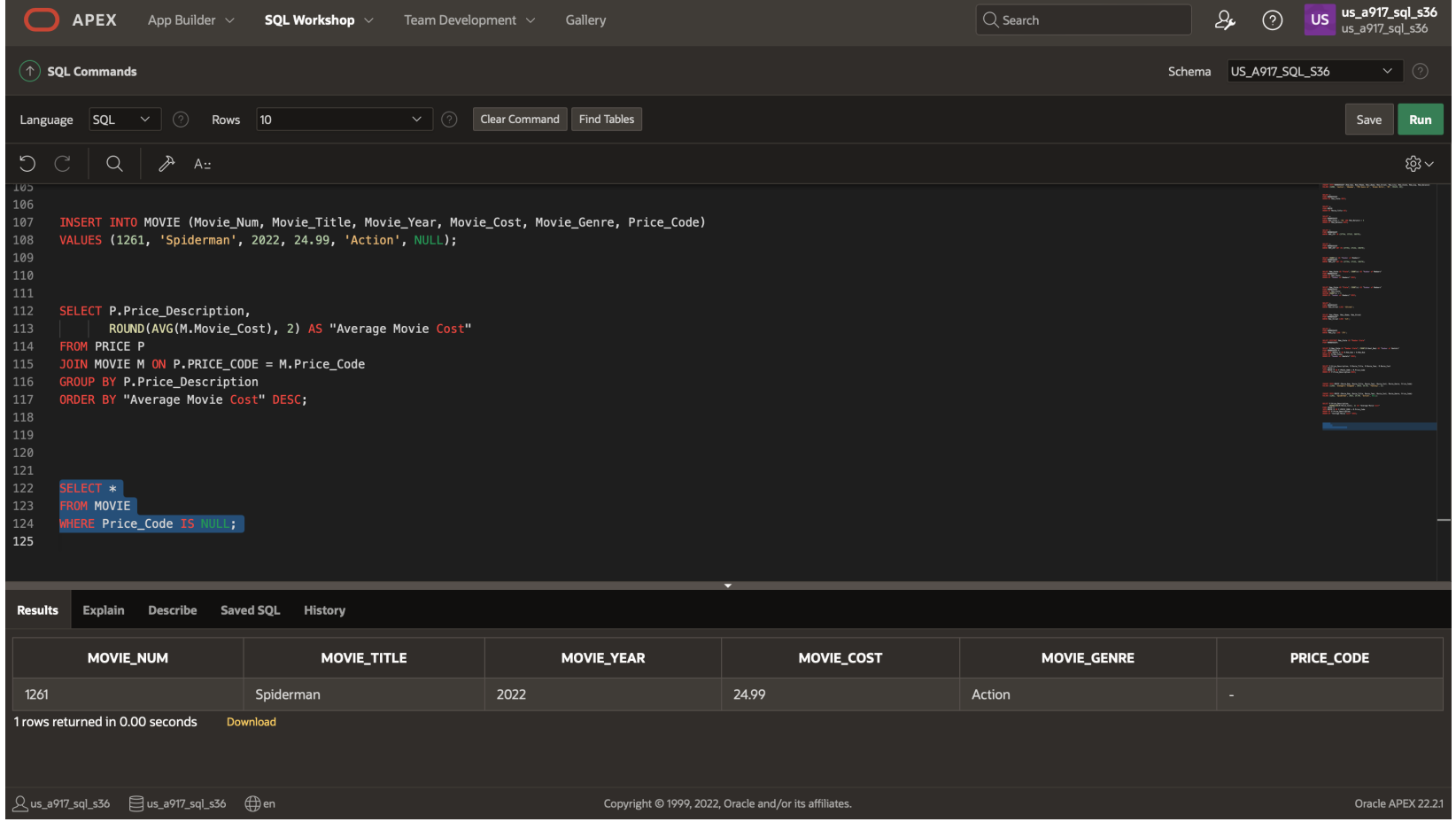
MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST	MOVIE_GENRE	PRICE_CODE
1261	Avengers: Infinity War	2018	39.95	FANTASY	-

Your result should show the record of the movie that you just entered with the price code as null.

Answer

SQL:

```
SELECT *
FROM MOVIE
WHERE Price_Code IS NULL;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
INSERT INTO MOVIE (Movie_Num, Movie_Title, Movie_Year, Movie_Cost, Movie_Genre, Price_Code)
VALUES (1261, 'Spiderman', 2022, 24.99, 'Action', NULL);
SELECT P.Price_Description,
ROUND(AVG(M.Movie_Cost), 2) AS "Average Movie Cost"
FROM PRICE P
JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
GROUP BY P.Price_Description
ORDER BY "Average Movie Cost" DESC;
SELECT *
FROM MOVIE
WHERE Price_Code IS NULL;
Results
Explain
Describe
Saved SQL
History
MOVIE_NUM
MOVIE_TITLE
MOVIE_YEAR
MOVIE_COST
MOVIE_GENRE
PRICE_CODE
1261
Spiderman
2022
24.99
Action
-
1 rows returned in 0.00 seconds
Download
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

18. Now perform a FULL OUTER JOIN using tables Price and Movie to retrieve Price Description from the Price table and Movie Title, Movie Year and Movie Genre from the Movie table. Sort the data on Movie Year in descending order. You should get the following result (see below):

PRICE_DESCRIPTION	MOVIE_TITLE	MOVIE_YEAR	MOVIE_GENRE
Weekly Special	-	-	-
New Release	Avengers: Endgame	2019	FANTASY
-	Avengers: Infinity War	2018	FANTASY
New Release	Doctor Strange	2016	FANTASY
New Release	Fantastic Beasts and Where to Find Them	2016	FANTASY
New Release	Richard Goodhope	2010	DRAMA
New Release	Constant Companion	2010	DRAMA
New Release	The Cesar Family Christmas	2009	FAMILY
New Release	Beatnik Fever	2009	COMEDY
Standard	What He Doesn't Know	2008	COMEDY
Discount	Time to Burn	2008	ACTION
Discount	Smokey Mountain Wildlife	2006	ACTION
Discount	Where Hope Dies	2000	DRAMA
Standard	Gone With the Wind	1939	DRAMA

Record of movie that you had entered

Answer

SQL:

```
SELECT P.Price_Description, M.Movie_Title, M.Movie_Year, M.Movie_Genre
FROM PRICE P
FULL OUTER JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
ORDER BY M.Movie_Year DESC;
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
117 ORDER BY "Average Movie Cost" DESC;
118
119
120
121
122
123
124
125 SELECT P.Price_Description, M.Movie_Title, M.Movie_Year, M.Movie_Genre
126 FROM PRICE P
127 FULL OUTER JOIN MOVIE M ON P.PRICE_CODE = M.Price_Code
128 ORDER BY M.Movie_Year DESC;
129
```

Results Explain Describe Saved SQL History

PRICE_DESCRIPTION	MOVIE_TITLE	MOVIE_YEAR	MOVIE_GENRE
Weekly Special	-	-	-
-	Spiderman	2022	Action
New Release	Avengers: Endgame	2019	Fantasy
New Release	Richard Goodhope	2010	DRAMA

Copyright © 1999, 2022, Oracle and/or its affiliates.

19.

a. First create a query that returns all the columns by joining the tables Membership, Rental and DetailRental

b. Now modify this query to only include Member First Name, Member Last Name and a new column that computes the late charges as the difference between the return date (Detail_ReturnDate) and due date (Detail_DueDate) multiplied by the daily late fee (Detail_DailyLateFee): (Detail_ReturnDate - Detail_DueDate) * Detail_DailyLateFee . Name this derived (calculated) field as “Late Charges”. Name Member First Name as “First Name” and Last Name as “Last Name”.

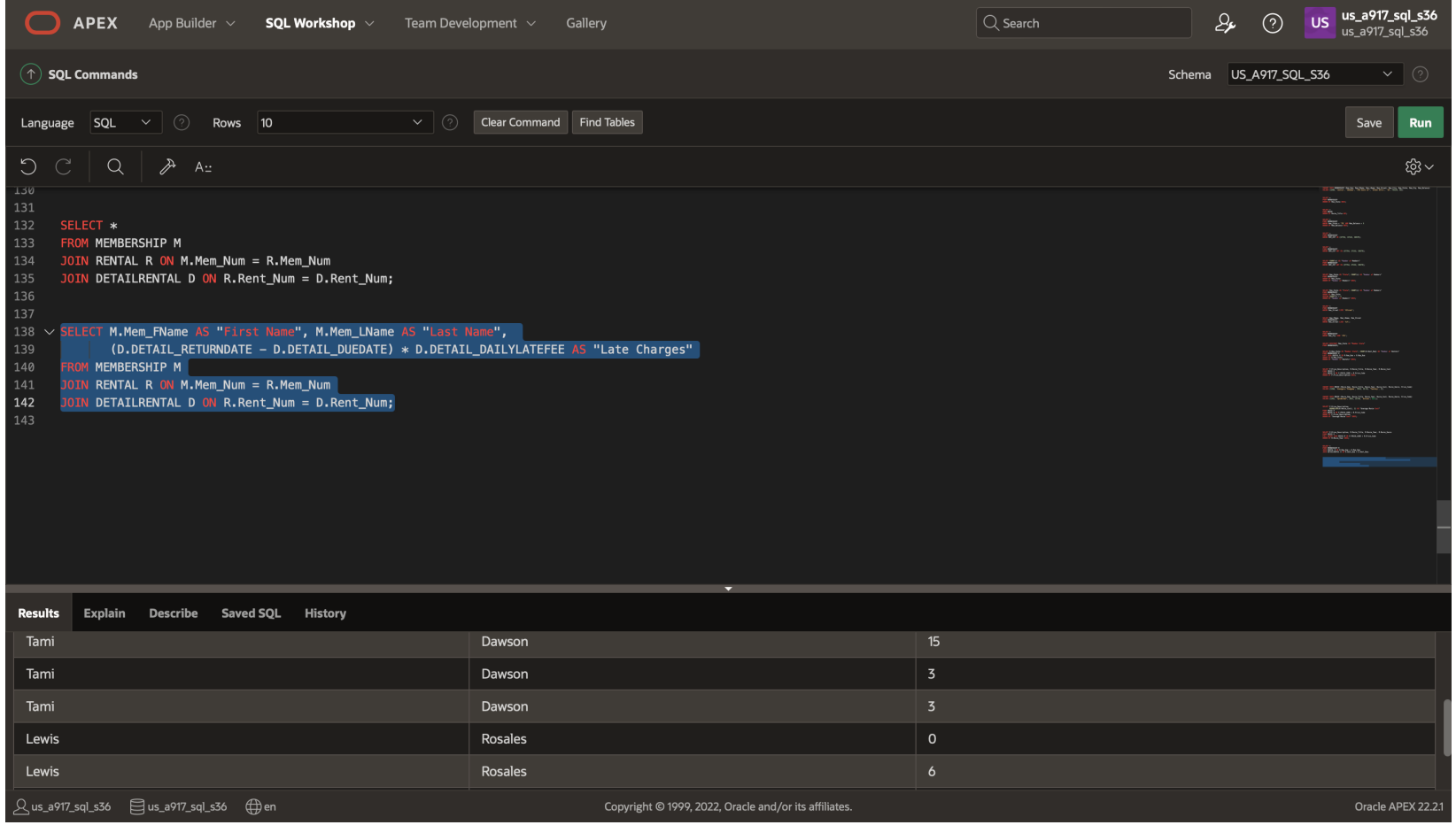
First Name	Last Name	Late Charges
Tami	Dawson	15
Tami	Dawson	3
Tami	Dawson	3
Lewis	Rosales	6
Lewis	Rosales	6

Note: Only those records should be included where the Return Date is later than the Due Date.

Answer

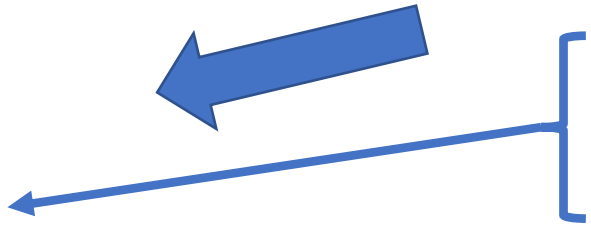
SQL:

```
SELECT M.Mem_FName AS "First Name", M.Mem_LName AS "Last Name",
    (D.DETAIL_RETURNDATE - D.DETAIL_DUEDATE) * D.DETAIL_DAILYLATEFEE AS "Late Charges"
FROM MEMBERSHIP M
JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
130
131
132 SELECT *
133 FROM MEMBERSHIP M
134 JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
135 JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num;
136
137
138 SELECT M.Mem_FName AS "First Name", M.Mem_LName AS "Last Name",
139 (D.DETAIL_RETURNDATE - D.DETAIL_DUEDATE) * D.DETAIL_DAILYLATEFEE AS "Late Charges"
140 FROM MEMBERSHIP M
141 JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
142 JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num;
143
Results
Explain
Describe
Saved SQL
History
Tami Dawson 15
Tami Dawson 3
Tami Dawson 3
Lewis Rosales 0
Lewis Rosales 6
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

20. Notice that there were multiple late charges for some members. Modify the previous query to show the Total Late Charges for each Member.

First Name	Last Name	Total Late Charges
Tami	Dawson	21
Lewis	Rosales	12

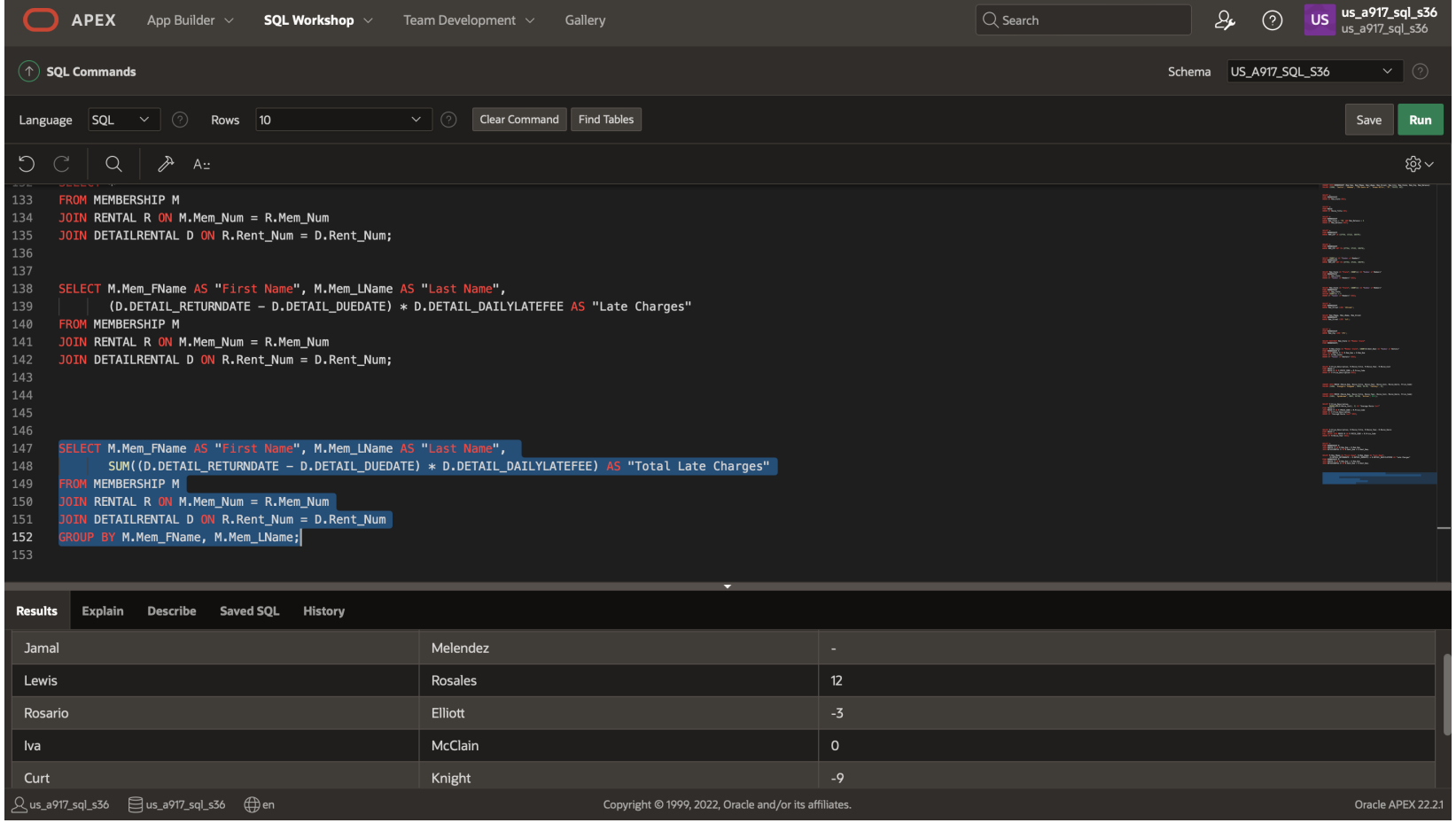


First Name	Last Name	Late Charges
Tami	Dawson	15
Tami	Dawson	3
Tami	Dawson	3
Lewis	Rosales	6
Lewis	Rosales	6

Answer

SQL:

```
SELECT M.Mem_FName AS "First Name", M.Mem_LName AS "Last Name",
    SUM((D.DETAIL_RETURNDATE - D.DETAIL_DUEDATE) * D.DETAIL_DAILYLATEFEE) AS "Total Late Charges"
FROM MEMBERSHIP M
JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num
GROUP BY M.Mem_FName, M.Mem_LName;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
FROM MEMBERSHIP M
JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num;
SELECT M.Mem_FName AS "First Name", M.Mem_LName AS "Last Name",
(D.DETAIL_RETURNDATE - D.DETAIL_DUEDATE) * D.DETAIL_DAILYLATEFEE AS "Late Charges"
FROM MEMBERSHIP M
JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num;
SELECT M.Mem_FName AS "First Name", M.Mem_LName AS "Last Name",
SUM((D.DETAIL_RETURNDATE - D.DETAIL_DUEDATE) * D.DETAIL_DAILYLATEFEE) AS "Total Late Charges"
FROM MEMBERSHIP M
JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num
GROUP BY M.Mem_FName, M.Mem_LName;
Results
Explain
Describe
Saved SQL
History
Jamal Melendez -
Lewis Rosales 12
Rosario Elliott -3
Iva McClain 0
Curt Knight -9
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

21. Create a query that returns the Member First Name, Member Last Name, City and State from the Membership table.

MEM_FNAME	MEM_LNAME	MEM_CITY	MEM_STATE
Tami	Dawson	Norene	TN
Curt	Knight	Flatgap	KY
Jamal	Melendez	Quebeck	TN
Iva	McClain	Summit	KY
Miranda	Parks	Germantown	TN
Rosario	Elliott	Columbia	TN
Mattie	Guy	Lily	KY
Clint	Ochoa	Greenville	TN
Lewis	Rosales	Counce	TN
Stacy	Mann	Murfreesboro	TN
Luis	Trujillo	Heiskell	TN
Minnie	Gonzales	Williston	TN

This is an intermediate query, so screenshots are not needed for this query. See next question after creating this query.

21. Now modify this query using concatenation (||) operator that returns a single column labeled as “Member Location” that shows the following result.

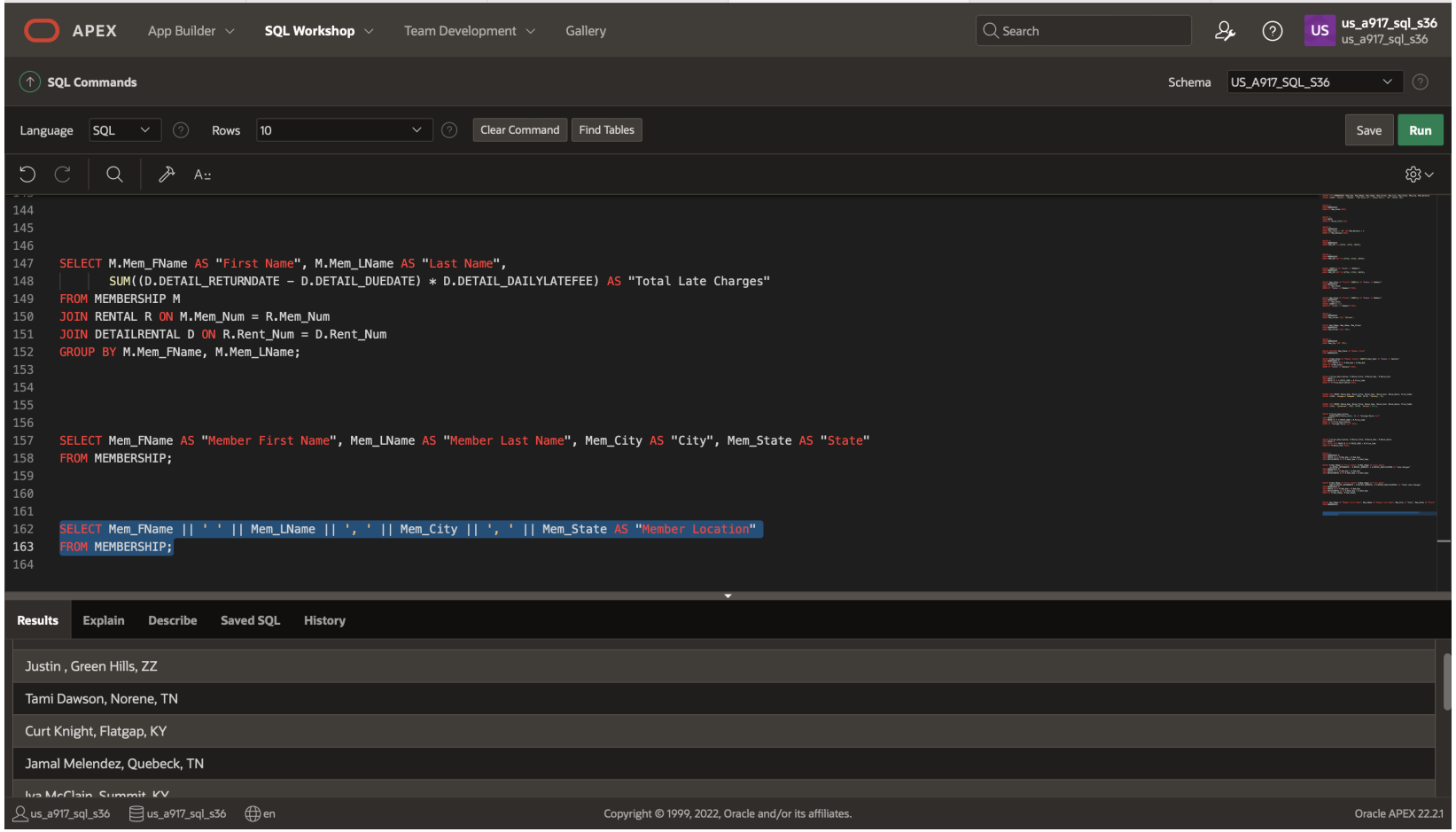
MEM_FNAME	MEM_LNAME	MEM_CITY	MEM_STATE
Tami	Dawson	Norene	TN
Curt	Knight	Flatgap	KY
Jamal	Melendez	Quebeck	TN
Iva	McClain	Summit	KY
Miranda	Parks	Germantown	TN
Rosario	Elliott	Columbia	TN
Mattie	Guy	Lily	KY
Clint	Ochoa	Greenville	TN
Lewis	Rosales	Counce	TN
Stacy	Mann	Murfreesboro	TN
Luis	Trujillo	Heiskell	TN
Minnie	Gonzales	Williston	TN

Member Location
Tami Dawson lives in Norene,TN
Curt Knight lives in Flatgap,KY
Jamal Melendez lives in Quebeck,TN
Iva McClain lives in Summit,KY
Miranda Parks lives in Germantown,TN
Rosario Elliott lives in Columbia,TN
Mattie Guy lives in Lily,KY
Clint Ochoa lives in Greenville,TN
Lewis Rosales lives in Counce,TN
Stacy Mann lives in Murfreesboro,TN
Luis Trujillo lives in Heiskell,TN
Minnie Gonzales lives in Williston,TN

Answer

SQL:

```
SELECT Mem_FName || ' ' || Mem_LName || ', ' || Mem_City || ', ' || Mem_State AS "Member Location"
FROM MEMBERSHIP;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
144
145
146
147 SELECT M.Mem_FName AS "First Name", M.Mem_LName AS "Last Name",
148 SUM((D.DETAIL_RETURNDATE - D.DETAIL_DUEDATE) * D.DETAIL_DAILYLATEFEE) AS "Total Late Charges"
149 FROM MEMBERSHIP M
150 JOIN RENTAL R ON M.Mem_Num = R.Mem_Num
151 JOIN DETAILRENTAL D ON R.Rent_Num = D.Rent_Num
152 GROUP BY M.Mem_FName, M.Mem_LName;
153
154
155
156
157 SELECT Mem_FName AS "Member First Name", Mem_LName AS "Member Last Name", Mem_City AS "City", Mem_State AS "State"
158 FROM MEMBERSHIP;
159
160
161
162 SELECT Mem_FName || ' ' || Mem_LName || ', ' || Mem_City || ', ' || Mem_State AS "Member Location"
163 FROM MEMBERSHIP;
164
Results
Explain
Describe
Saved SQL
History
Justin , Green Hills, ZZ
Tami Dawson, Norene, TN
Curt Knight, Flatgap, KY
Jamal Melendez, Quebeck, TN
us_a917_sql_s36
us_a917_sql_s36
en
Copyright © 1999, 2022, Oracle and/or its affiliates.
Oracle APEX 22.2.1

22. List the names of members with **above average** member balance.

Hint: First write a query that shows the average member balance. Using the result generated by this query, return the member first name and last name for members with above average balance.

MEM_FNAME	MEM_LNAME
Tami	Dawson
Curt	Knight
Iva	McClain
Rosario	Elliott
Clint	Ochoa
Stacy	Mann

Your results may have one more record depending on what data you had entered earlier.

Answer

SQL:

```
SELECT Mem_FName AS "Member First Name", Mem_LName AS "Member Last Name", Mem_Balance
FROM MEMBERSHIP
WHERE Mem_Balance > (SELECT AVG(Mem_Balance) FROM MEMBERSHIP);
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```
155
156
157 SELECT Mem_FName AS "Member First Name", Mem_LName AS "Member Last Name", Mem_City AS "City", Mem_State AS "State"
158 FROM MEMBERSHIP;
159
160
161
162 SELECT Mem_FName || ' ' || Mem_LName || ', ' || Mem_City || ', ' || Mem_State AS "Member Location"
163 FROM MEMBERSHIP;
164
165
166
167 SELECT AVG(Mem_Balance) AS Avg_Balance
168 FROM MEMBERSHIP;
169
170
171
172 SELECT Mem_FName AS "Member First Name", Mem_LName AS "Member Last Name", Mem_Balance
173 FROM MEMBERSHIP
174 WHERE Mem_Balance > (SELECT AVG(Mem_Balance) FROM MEMBERSHIP);
175
```

Results Explain Describe Saved SQL History

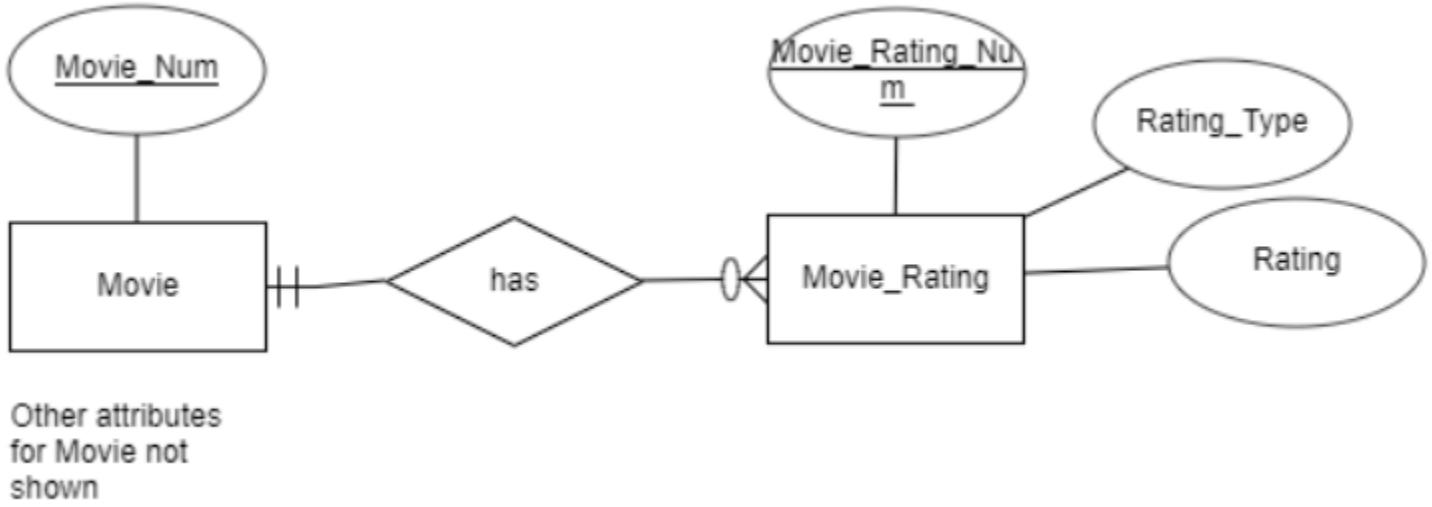
Member First Name	Member Last Name	MEM_BALANCE
Justin	-	45
Tami	Dawson	11
Iva	McClain	15
Justin	Akwuba	45

Copyright © 1999, 2022, Oracle and/or its affiliates.

Time for database renovation! Now we would like to capture movie ratings for each movie so that customers can be provided this information.

Source: Flickr Creative Commons/Ann Wuyts

23. Suppose we want to track the movie ratings esp. the ratings on IMDB and on Rotten Tomatoes. Create a table named **Movie_Rating** that has columns **Movie_Rating_Num** with datatype **INT**, **Rating_Type** with data type **Varchar2(50)**, **Rating** with datatype **Number(10,2).** Each Movie could have multiple ratings (e.g., IMDB, tomatometer, Audience score from Rotten Tomatoes) and each rating is for a specific movie. Thus, **Movie_Num** should appear as a foreign key in the **Movie_Rating** table with INT data type (foreign keys should have same data type as the primary key in the "parent" table). While Rotten Tomatoes report their ratings in %, we will just use the number without worrying about the % format.



#	Column	Type	Length	Precision	Scale	Nullable
1	MOVIE_RATING_NUM	NUMBER	22		0	No
2	RATING_TYPE	VARCHAR2	50			No
3	RATING	NUMBER	22	10	2	No
4	MOVIE_NUM	NUMBER	22		0	No

Whole number, integer data type

Answer

SQL:

```
CREATE TABLE Movie_Rating (
   Movie_Rating_Num INT,
   Movie_Num INT,
   Rating_Type VARCHAR2(50),
   Rating NUMBER(10,2),
   CONSTRAINT Movie_Rating_PK PRIMARY KEY (Movie_Rating_Num),
   CONSTRAINT Movie_Rating_FK FOREIGN KEY (Movie_Num) REFERENCES Movie (Movie_Num)
);
```

SQL Commands Schema US_A917_SQL_S36

Language SQL Rows 10 Clear Command Find Tables Save Run

```sql
SELECT Mem_FName AS "Member First Name", Mem_LName AS "Member Last Name", Mem_Balance
FROM MEMBERSHIP
WHERE Mem_Balance > (SELECT AVG(Mem_Balance) FROM MEMBERSHIP);


CREATE TABLE Movie_Rating (
    Movie_Rating_Num INT,
    Movie_Num INT,
    Rating_Type VARCHAR2(50),
    Rating NUMBER(10,2),
    CONSTRAINT Movie_Rating_PK PRIMARY KEY (Movie_Rating_Num),
    CONSTRAINT Movie_Rating_FK FOREIGN KEY (Movie_Num) REFERENCES Movie (Movie_Num)
);
```

Results Explain Describe Saved SQL History

```
Table created.

0.02 seconds
```


Copyright © 1999, 2022, Oracle and/or its affiliates.

24. After creating the table, enter three records for the Movie Avengers: Endgame as shown in the table below. Remember, **Movie_Num** for this movie is **1260**.

MOVIE_RATING_NUM	RATING_TYPE	RATING	MOVIE_NUM
1	IMDB	8.4	1260
3	Audience Score	90	1260
2	Tomatometer	94	1260

Answer

SQL:

Put the screenshot here.

25. Having entered the three records in the **Movie_Rating** table, now create a query that shows the results as below. **Hint:** You will need to join the **Movie** and **Movie_Rating** tables.

MOVIE_TITLE	RATING_TYPE	RATING
Avengers: Endgame	IMDB	8.4
Avengers: Endgame	Audience Score	90
Avengers: Endgame	Tomatometer	94

Answer

SQL:

Put the screenshot here.

For the next five queries, you will create **your own questions** that demonstrate a specific use of a command and create queries that return results that answer your own question.

🛑**Do not directly reuse queries from assignment A6.7**

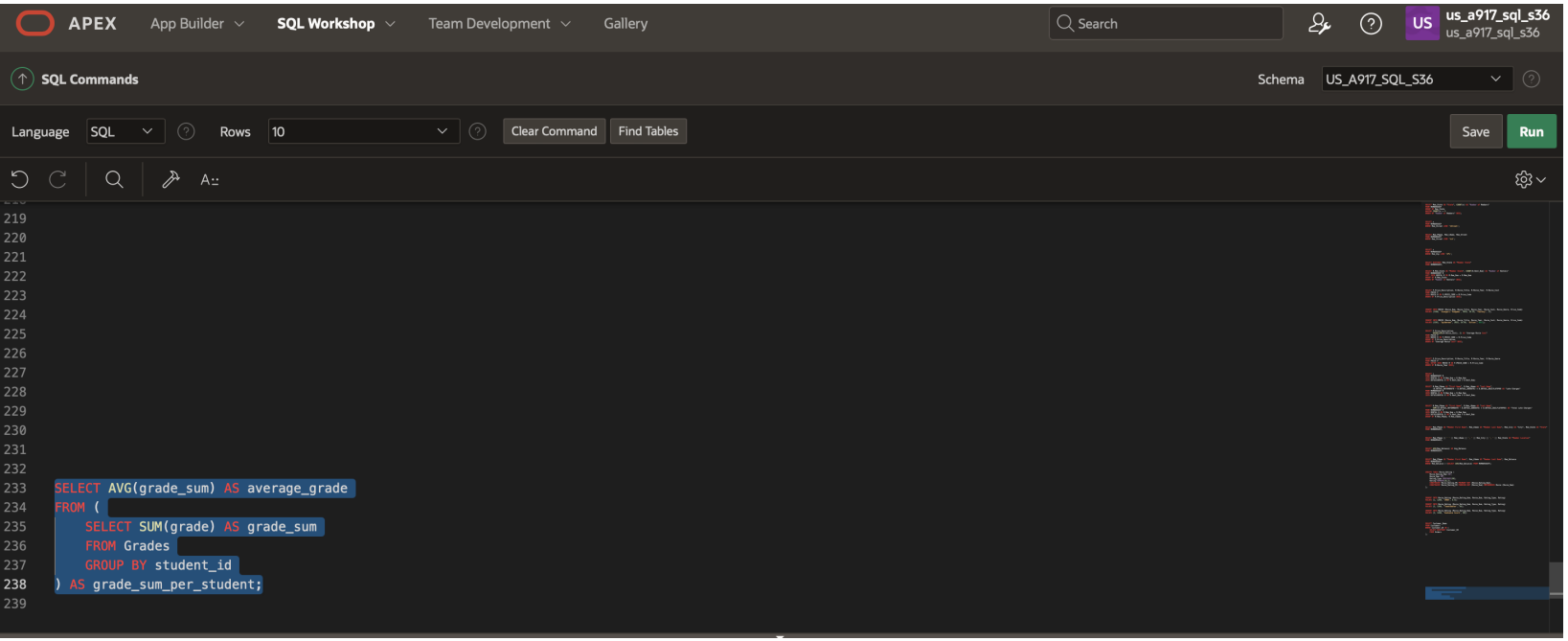
26. Create a query of your own where you are demonstrating the use of a **nested query** so that the results of one query are used within another query.

Your Question Here: find the average grade of all students.

Answer

SQL:

```
SELECT AVG(grade_sum) AS average_grade
FROM (
    SELECT SUM(grade) AS grade_sum
    FROM Grades
    GROUP BY student_id
) AS grade_sum_per_student;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233 SELECT AVG(grade_sum) AS average_grade
234 FROM (
235 SELECT SUM(grade) AS grade_sum
236 FROM Grades
237 GROUP BY student_id
238) AS grade_sum_per_student;
239

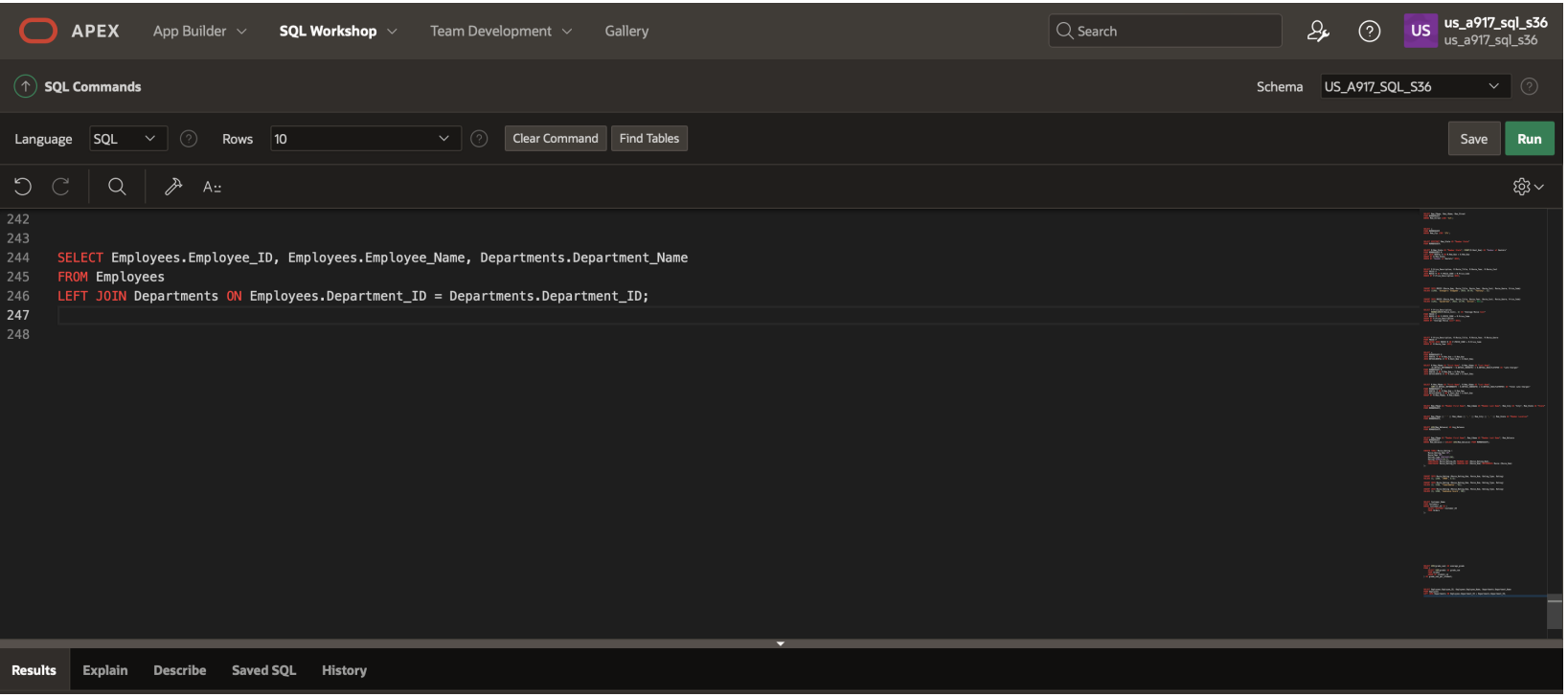
27. Create a query of your own where you are demonstrating the use of a LEFT JOIN or a RIGHT JOIN.

Your Question Here: retrieve a list of all employees along with their respective departments.
also want to include employees who are not assigned to any department

Answer

SQL:

```
SELECT Employees.Employee_ID, Employees.Employee_Name, Departments.Department_Name
FROM Employees
LEFT JOIN Departments ON Employees.Department_ID = Departments.Department_ID;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
242
243
244 SELECT Employees.Employee_ID, Employees.Employee_Name, Departments.Department_Name
245 FROM Employees
246 LEFT JOIN Departments ON Employees.Department_ID = Departments.Department_ID;
247
248
Results
Explain
Describe
Saved SQL
History

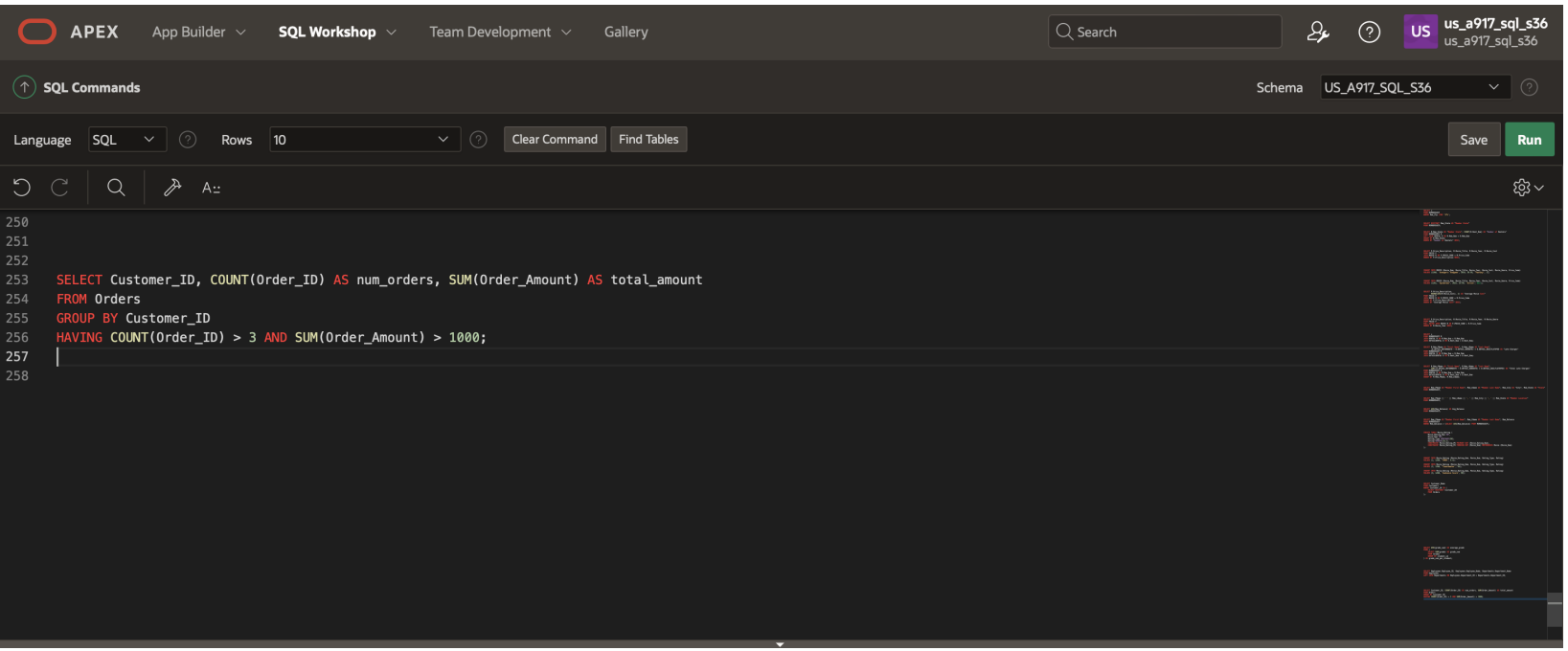
28. Create a query of your own where you are demonstrating the use of aggregation of results (e.g., use of **COUNT, MIN, MAX, SUM, AVG** etc.) and also the use of the **HAVING** Clause to filter the aggregated results.

Your Question Here: find customers who have placed 3 orders and their total order amount exceeds to $1000.

Answer

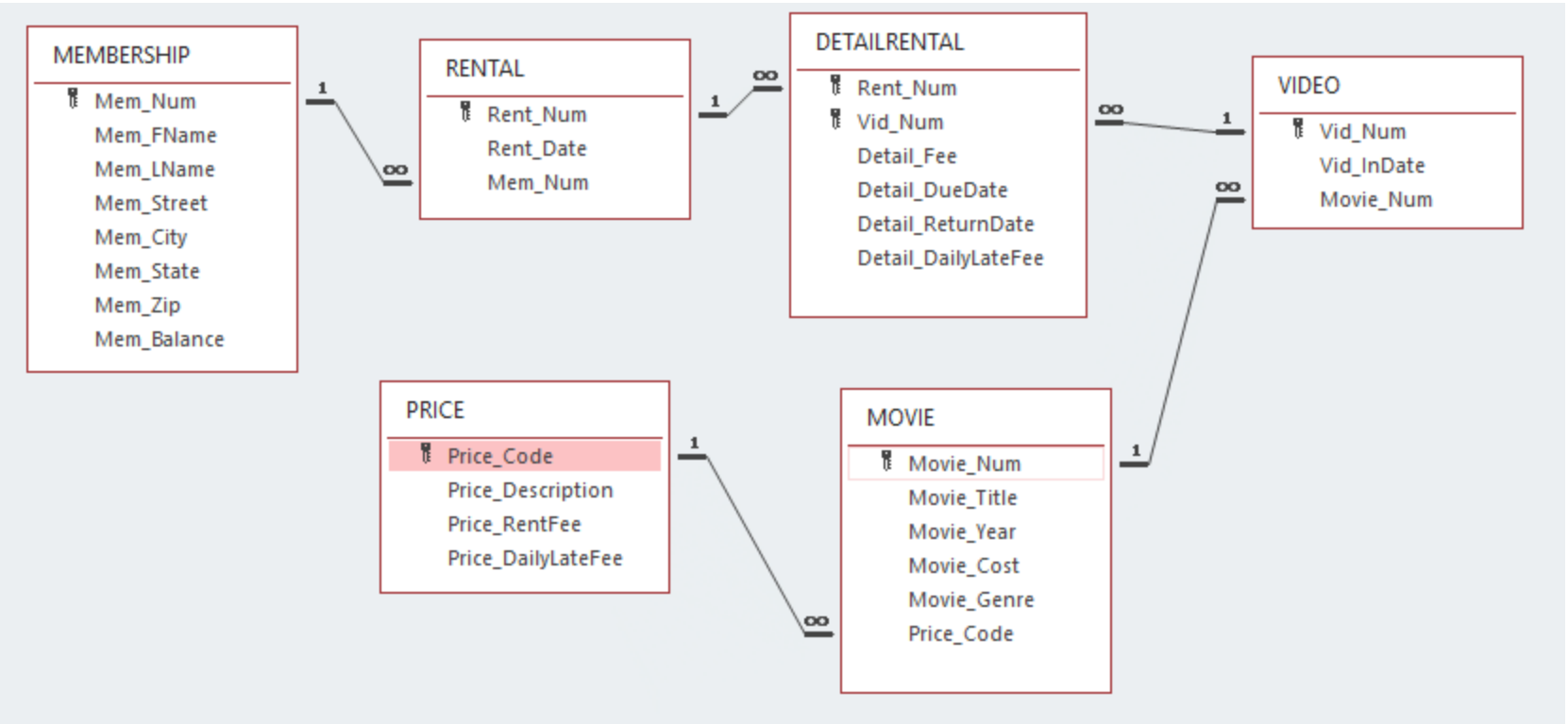
SQL:

```
SELECT Customer_ID, COUNT(Order_ID) AS num_orders, SUM(Order_Amount) AS total_amount
FROM Orders
GROUP BY Customer_ID
HAVING COUNT(Order_ID) > 3 AND SUM(Order_Amount) > 1000;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
250
251
252
253 SELECT Customer_ID, COUNT(Order_ID) AS num_orders, SUM(Order_Amount) AS total_amount
254 FROM Orders
255 GROUP BY Customer_ID
256 HAVING COUNT(Order_ID) > 3 AND SUM(Order_Amount) > 1000;
257
258

29. Create a query of your own where you are demonstrating the use of JOIN of three or more tables. Your query should join at least three tables from the six tables below. Make sure you are selecting the three tables that are directly connected or else you would need more joins.

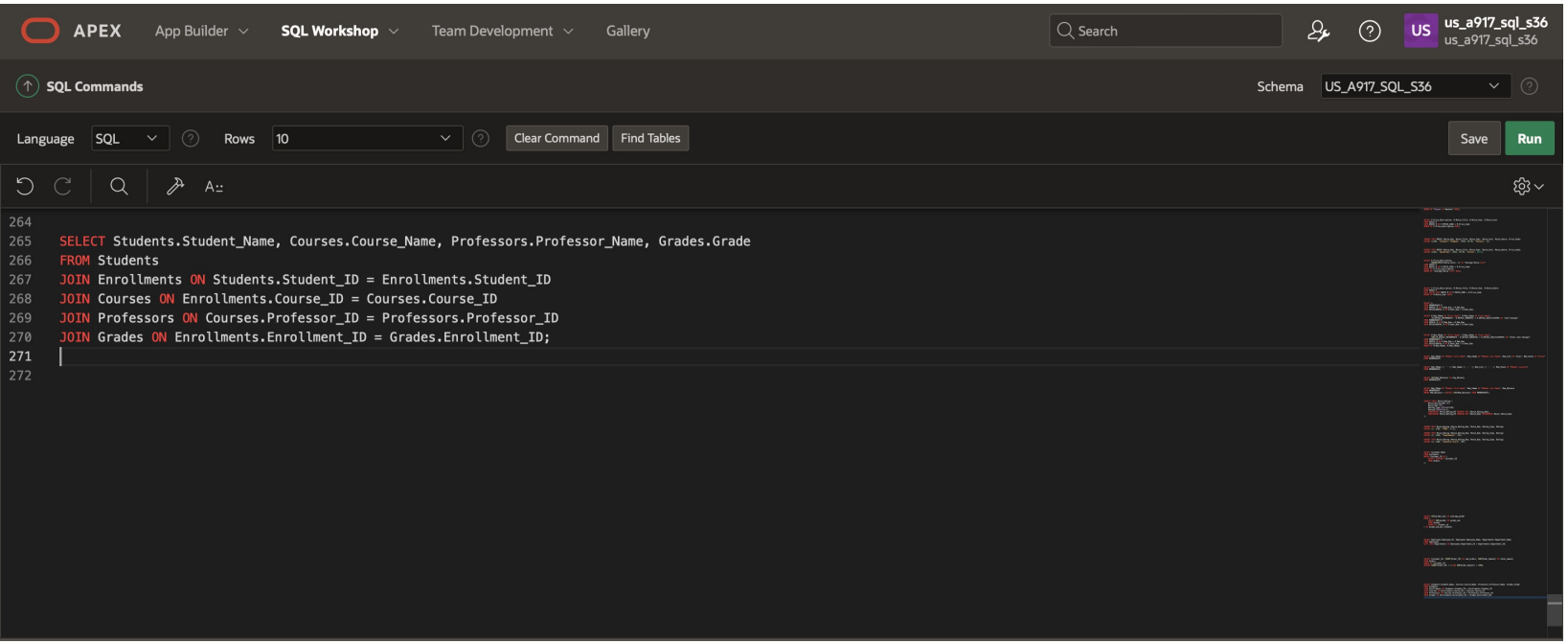
Your Question Here: retrieve the grades of students along with their corresponding course names and the names of professors who taught those courses.



Answer

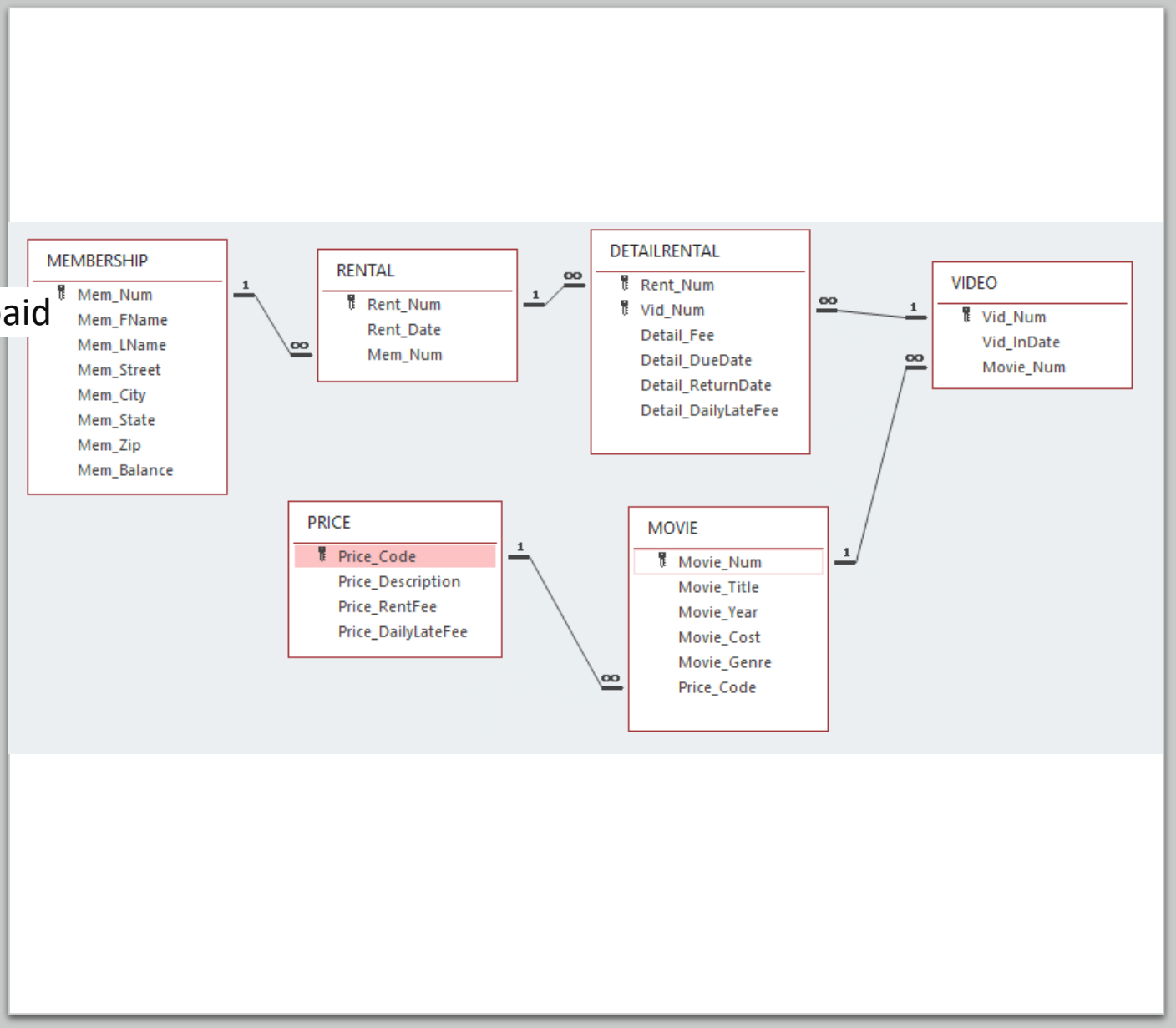
SQL:

```
SELECT Students.Student_Name, Courses.Course_Name, Professors.Professor_Name, Grades.Grade
FROM Students
JOIN Enrollments ON Students.Student_ID = Enrollments.Student_ID
JOIN Courses ON Enrollments.Course_ID = Courses.Course_ID
JOIN Professors ON Courses.Professor_ID = Professors.Professor_ID
JOIN Grades ON Enrollments.Enrollment_ID = Grades.Enrollment_ID;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
264
265 SELECT Students.Student_Name, Courses.Course_Name, Professors.Professor_Name, Grades.Grade
266 FROM Students
267 JOIN Enrollments ON Students.Student_ID = Enrollments.Student_ID
268 JOIN Courses ON Enrollments.Course_ID = Courses.Course_ID
269 JOIN Professors ON Courses.Professor_ID = Professors.Professor_ID
270 JOIN Grades ON Enrollments.Enrollment_ID = Grades.Enrollment_ID;
271
272

30. Create a query of your own where you are demonstrating a feature of your choice.

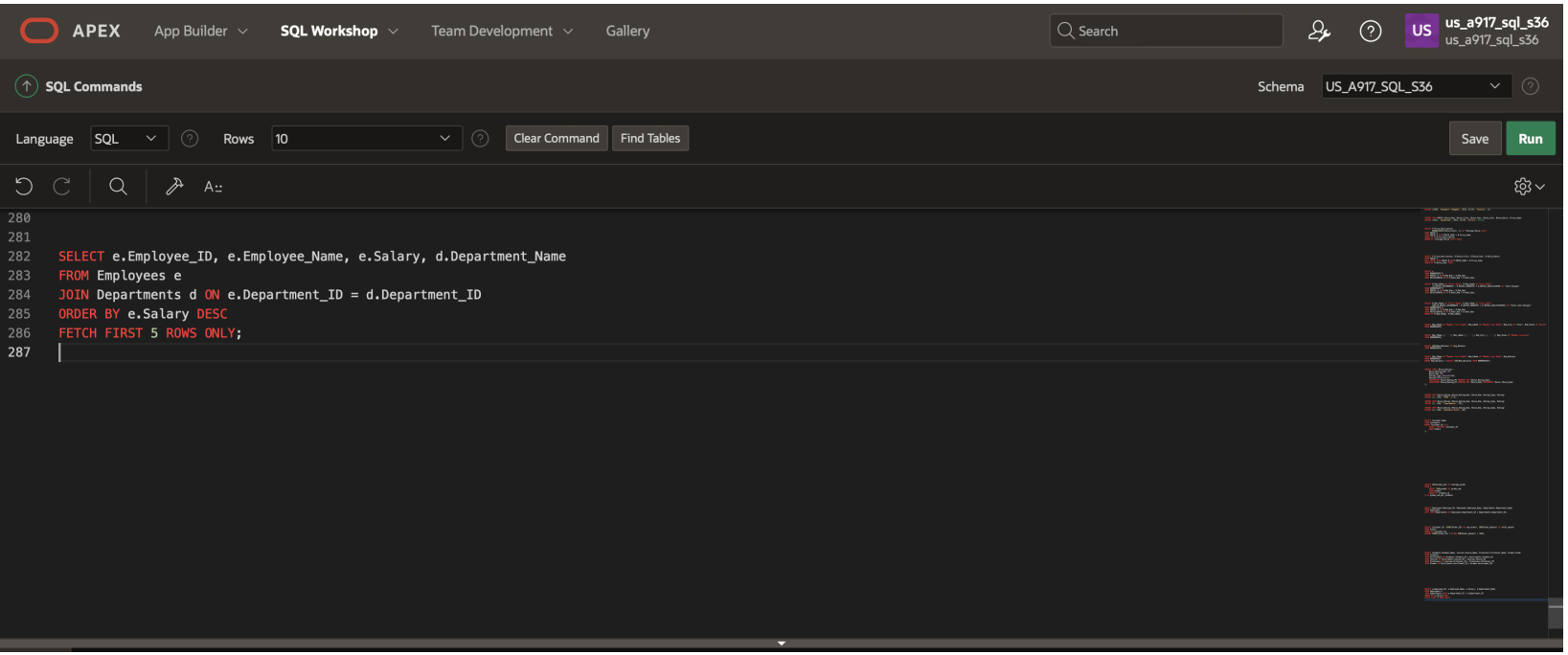
Your Question Here: Retrieve the top 5 highest-paid employees along with their department names.



Question: Put the question you are answering here.

SQL:

```
SELECT e.Employee_ID, e.Employee_Name, e.Salary, d.Department_Name
FROM Employees e
JOIN Departments d ON e.Department_ID = d.Department_ID
ORDER BY e.Salary DESC
FETCH FIRST 5 ROWS ONLY;
```


APEX
App Builder
SQL Workshop
Team Development
Gallery
Search
US
us_a917_sql_s36
us_a917_sql_s36
SQL Commands
Schema
US_A917_SQL_S36
Language
SQL
Rows
10
Clear Command
Find Tables
Save
Run
280
281
282 SELECT e.Employee_ID, e.Employee_Name, e.Salary, d.Department_Name
283 FROM Employees e
284 JOIN Departments d ON e.Department_ID = d.Department_ID
285 ORDER BY e.Salary DESC
286 FETCH FIRST 5 ROWS ONLY;
287

The End