VIDEO STORE TRACKING SYSTEM

CIS 228

MAY 9, 2020

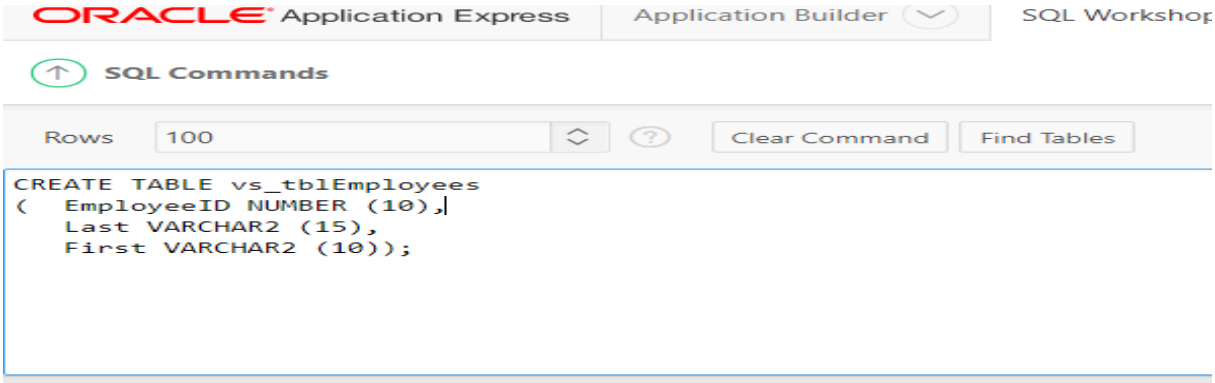
RAYMOND SARFO-AFRIHENE
BOB DESILETS

Summary

This database is designed to help Employees maintain a video store track system. This database shows us the relationship between other tables and how we can pull information from them. The database also helps us keep track of customers who have rented movies. By tracking this information, we can know how long the movie was rented for and what type of movie the customer rented. Lastly, we keep track of the movies based on the genre, distributor, boxes, type, cast, and actors.

Table of Contents

B1[1]



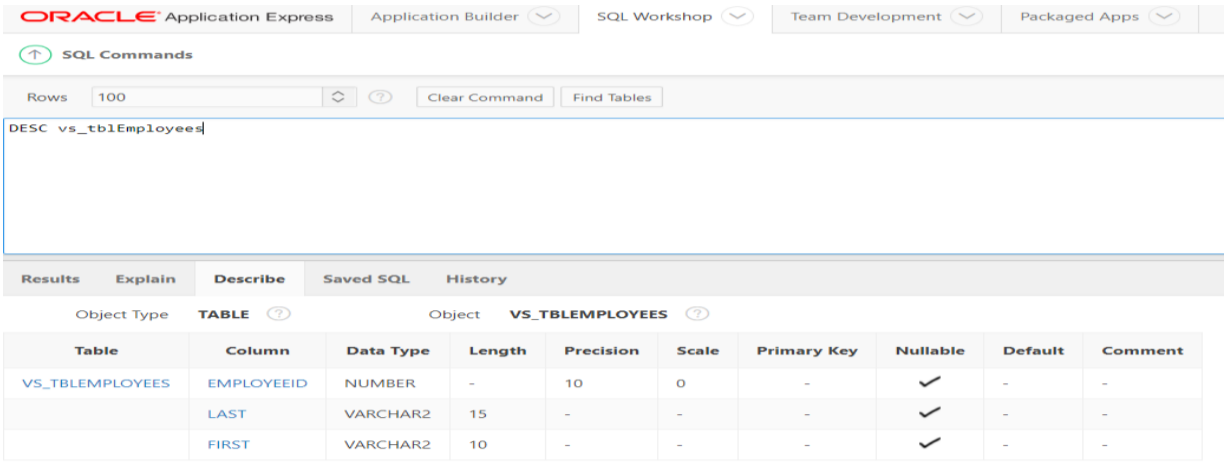
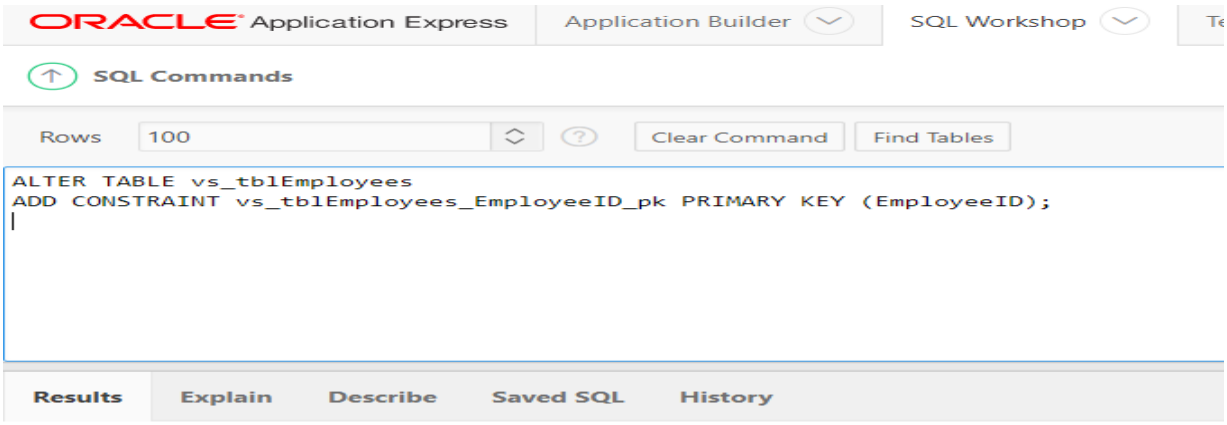
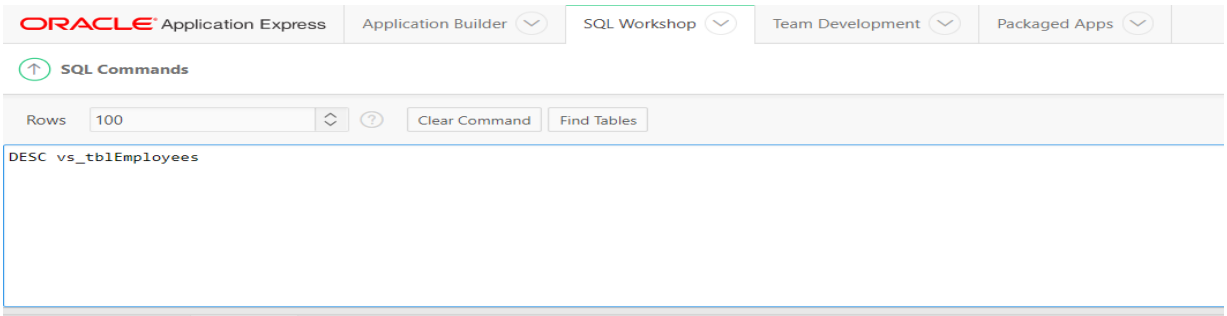


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VS_TBLEMPLOYEES	EMPLOYEEID	NUMBER	-	10	0	-	✓	-	-
	LAST	VARCHAR2	15	-	-	-	✓	-	-
	FIRST	VARCHAR2	10	-	-	-	✓	-	-

[1] B1
Add/Create vs_tblEmployee
a [PK] EmployeeID
b Add a foreign key EmployeeID to vs_tblRentals linkinking to vs_tblEmployee
c INSERT at least 10 employees with appropriate data





Object Type **TABLE** Object **VS_TBLEMPLOYEES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VS_TBLEMPLOYEES	EMPLOYEEID	NUMBER	-	10	0	1	-	-	-
	LAST	VARCHAR2	15	-	-	-	✓	-	-
	FIRST	VARCHAR2	10	-	-	-	✓	-	-

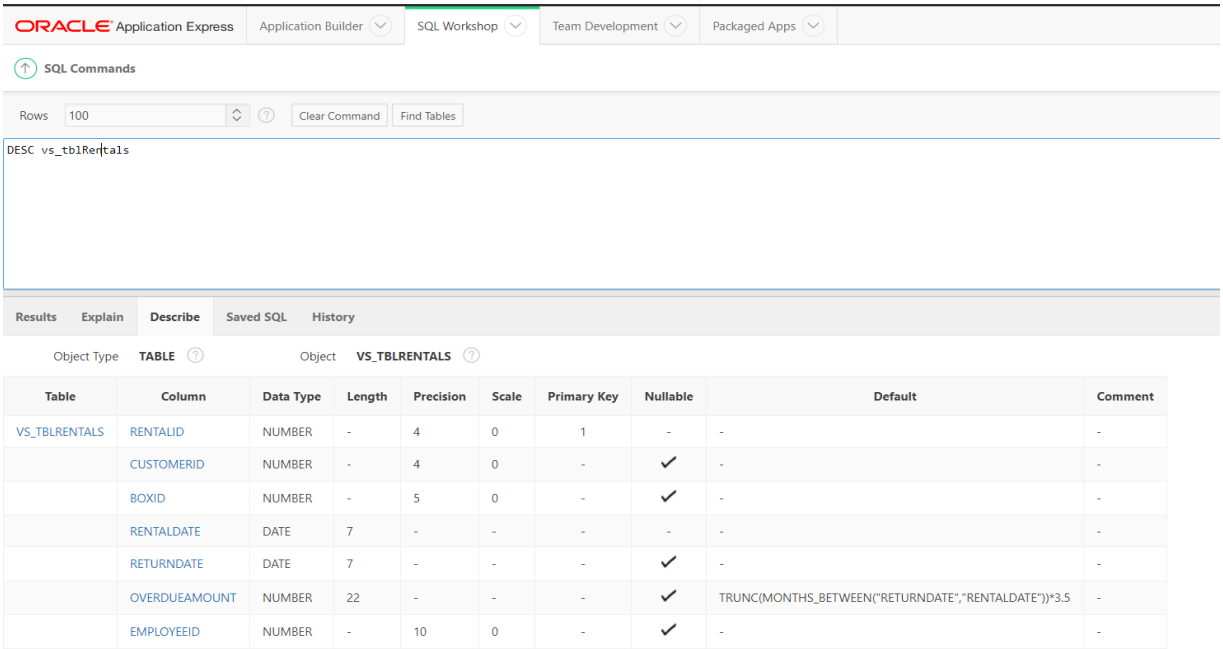
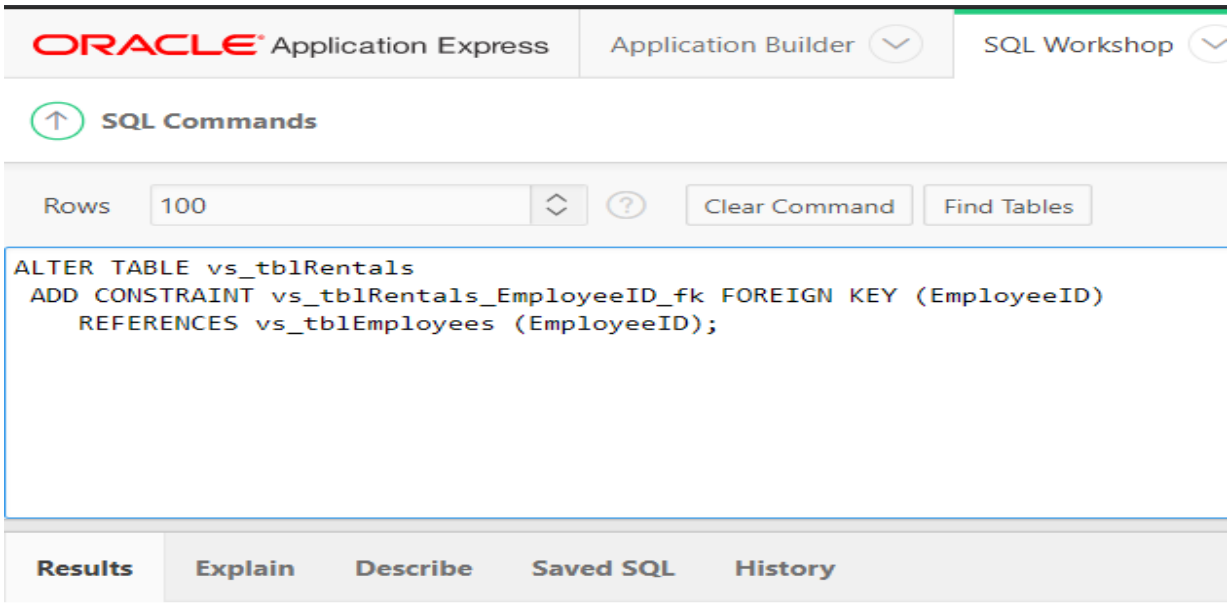
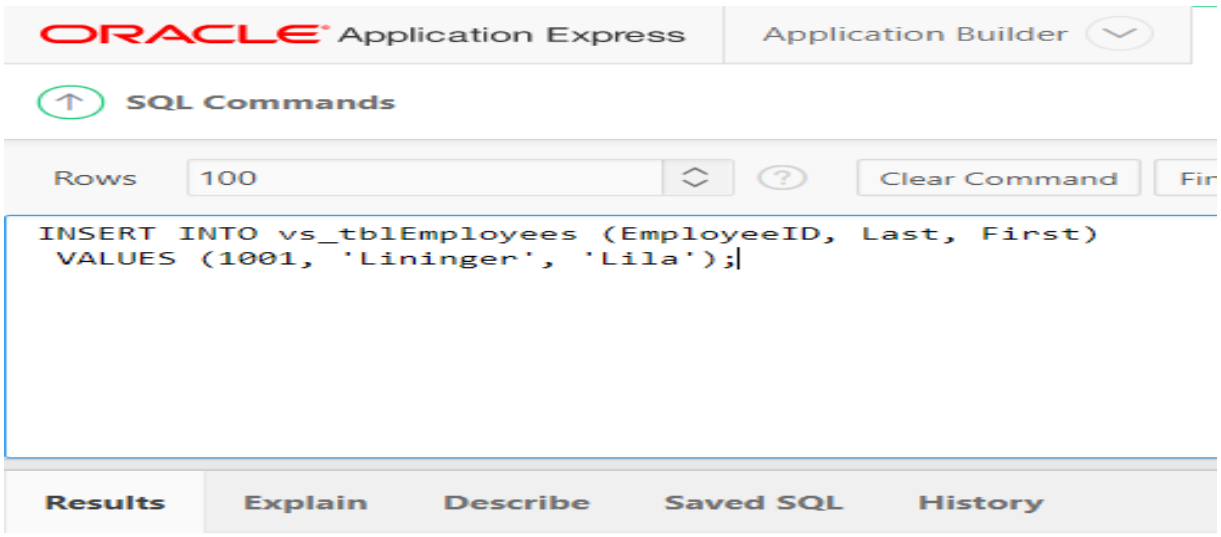
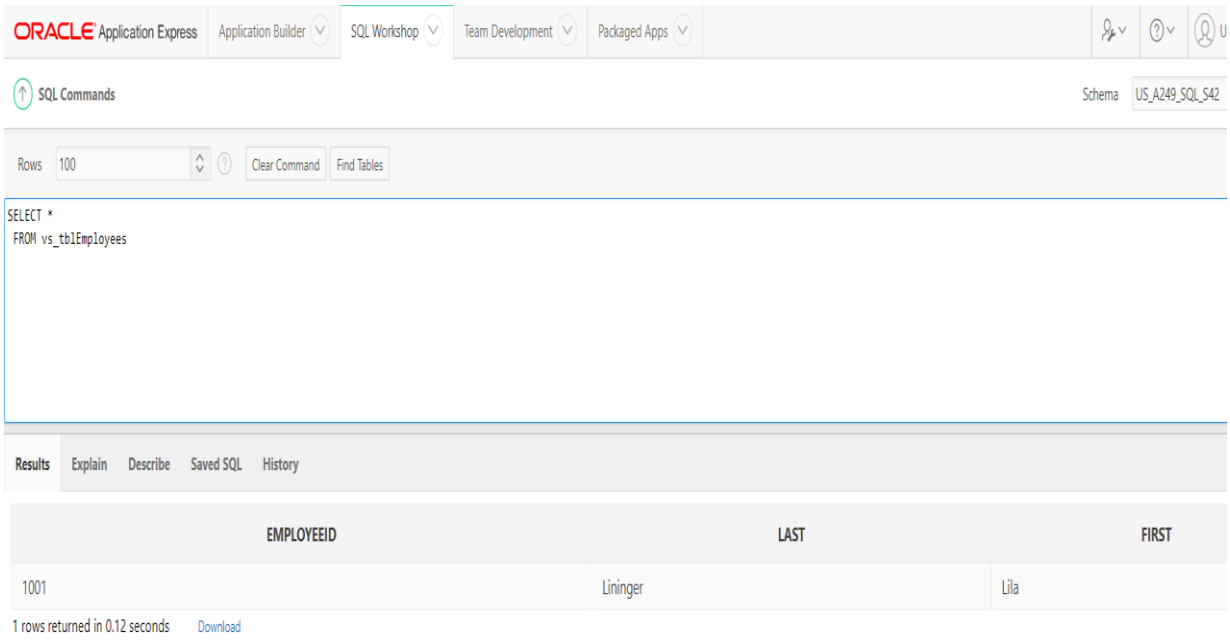
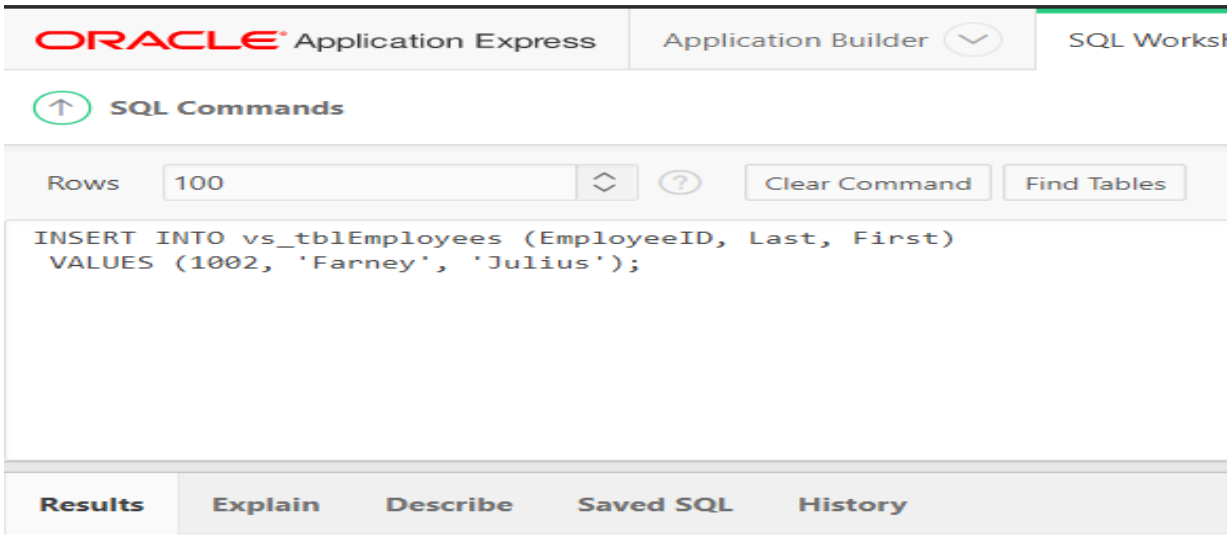


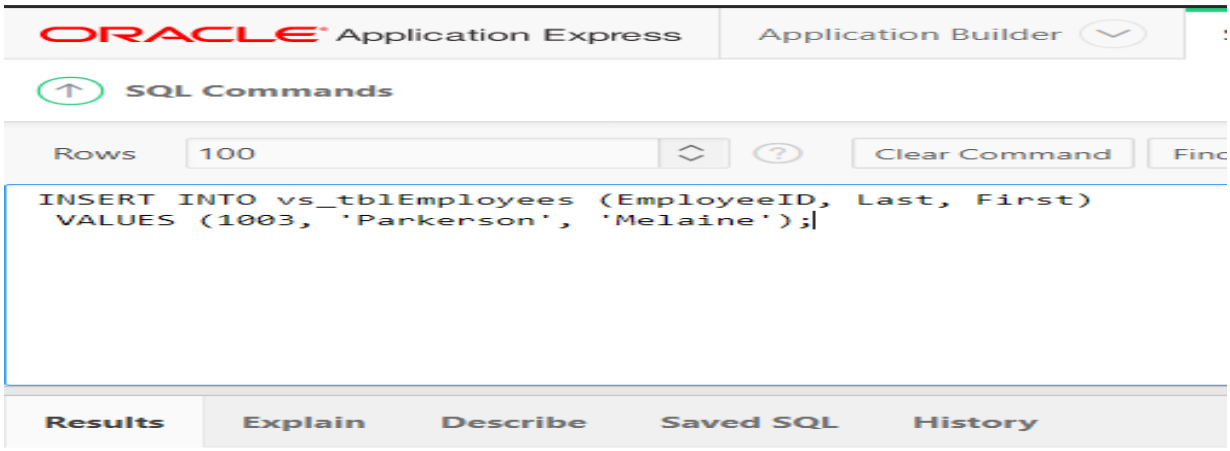
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VS_TBLRENTALS	RENTALID	NUMBER	-	4	0	1	-	-	-
	CUSTOMERID	NUMBER	-	4	0	-	✓	-	-
	BOXID	NUMBER	-	5	0	-	✓	-	-
	RENTALDATE	DATE	7	-	-	-	-	-	-
	RETURNDATE	DATE	7	-	-	-	✓	-	-
	OVERDUEAMOUNT	NUMBER	22	-	-	-	✓	TRUNC(MONTHS_BETWEEN("RETURNDATE","RENTALDATE"))*3.5	-
	EMPLOYEEID	NUMBER	-	10	0	-	✓	-	-

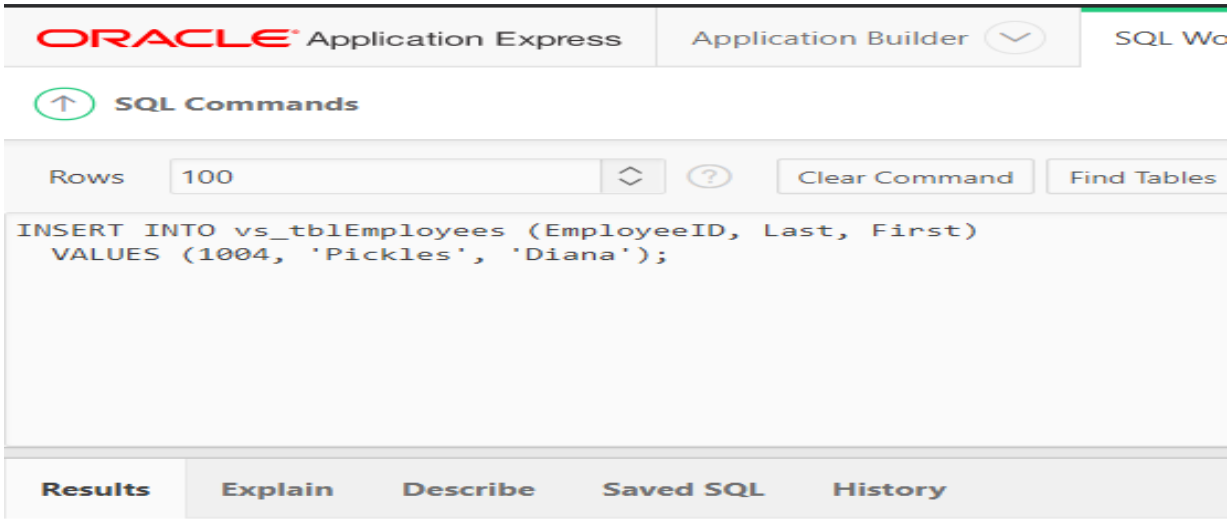

ORACLE Application Express
Application Builder
SQL Workshop
SQL Commands
Rows
100
Clear Command
Find Tables
ALTER TABLE vs_tblRentals
ADD CONSTRAINT vs_tblRentals_EmployeeID_fk FOREIGN KEY (EmployeeID)
REFERENCES vs_tblEmployees (EmployeeID);
Results
Explain
Describe
Saved SQL
History
Table altered.
0.11 seconds

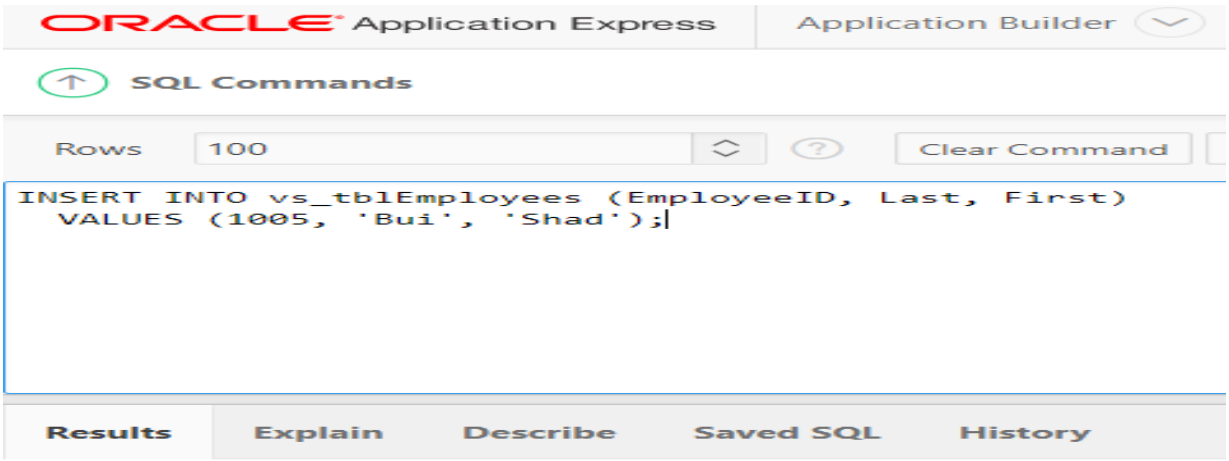

ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1001, 'Lininger', 'Lila');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.03 seconds

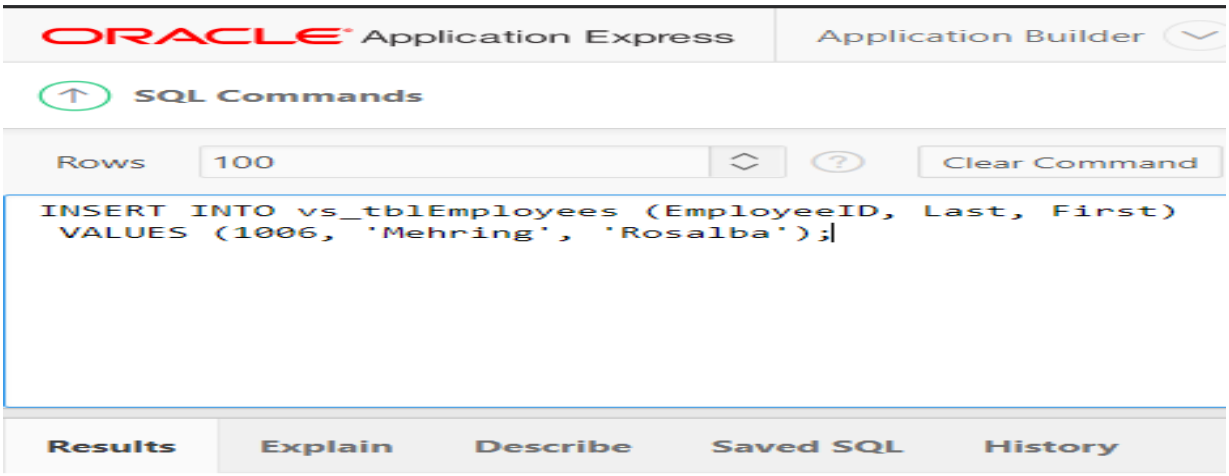

ORACLE Application Express
Application Builder
SQL Workshop
Team Development
Packaged Apps
SQL Commands
Schema US_A249_SQL_S42
Rows 100
Clear Command
Find Tables
SELECT *
FROM vs_tblEmployees
Results
Explain
Describe
Saved SQL
History
EMPLOYEEID
LAST
FIRST
1001
Lininger
Lila
1 rows returned in 0.12 seconds
Download

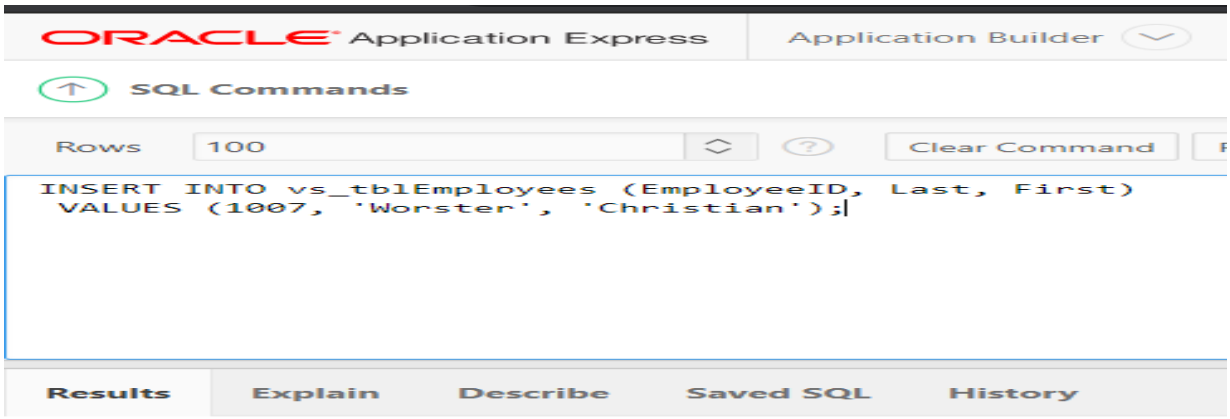

ORACLE Application Express
Application Builder
SQL Workshop
SQL Commands
Rows 100
Clear Command
Find Tables
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1002, 'Farney', 'Julius');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.01 seconds


ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1003, 'Parkerson', 'Melaine');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.03 seconds

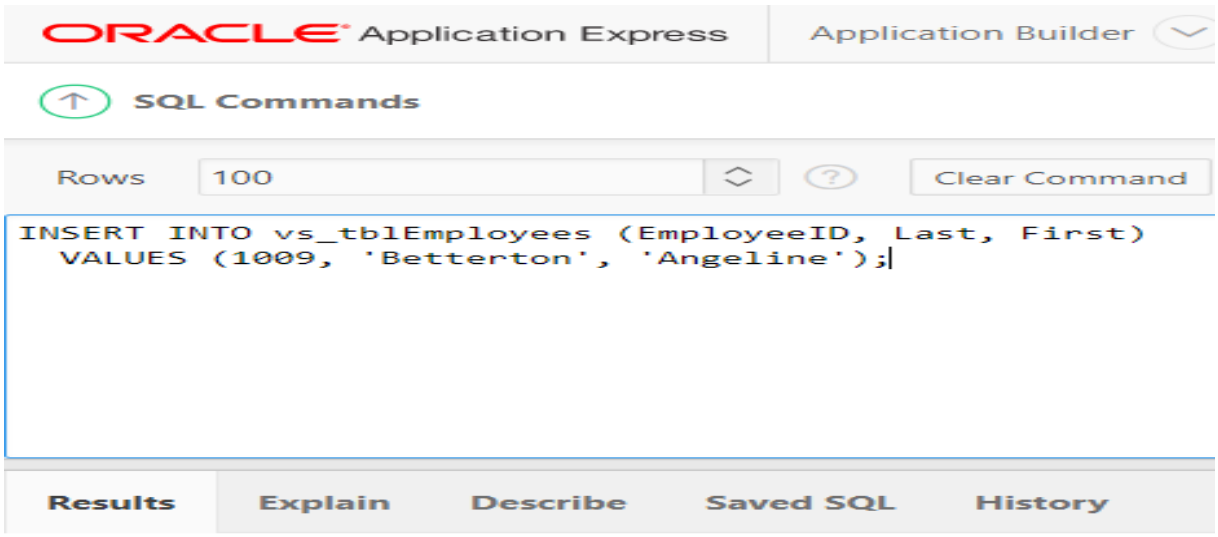

ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
Find Tables
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1004, 'Pickles', 'Diana');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.01 seconds

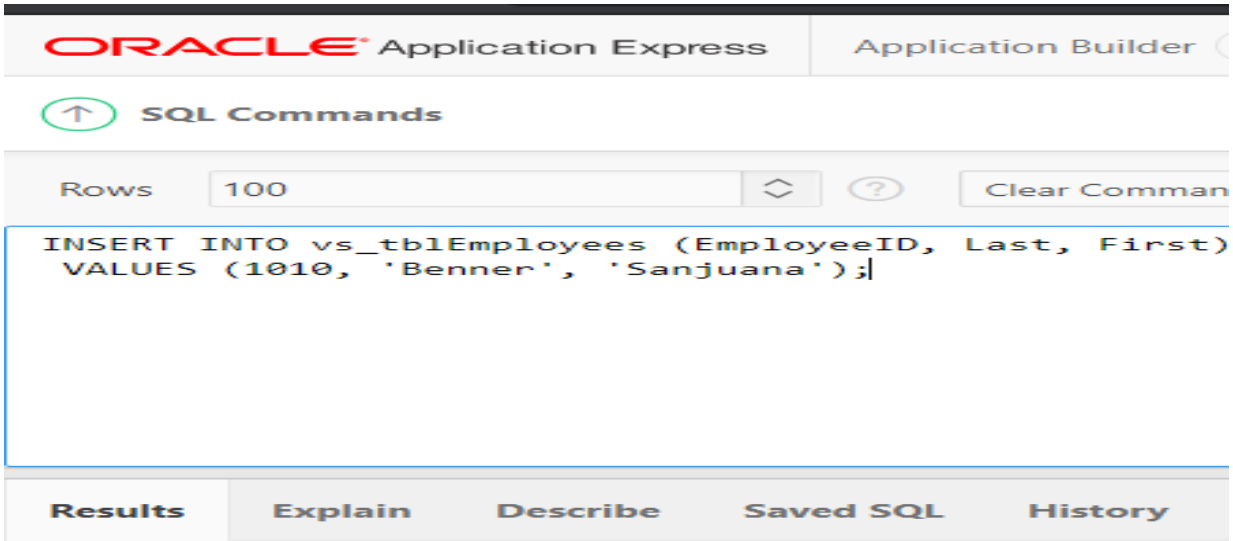

ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1005, 'Bui', 'Shad');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.04 seconds


ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1006, 'Mehring', 'Rosalba');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.01 seconds


ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1007, 'Worster', 'Christian');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.00 seconds


ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1008, 'Richie', 'Giselle');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.06 seconds


ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Command
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1009, 'Betterton', 'Angeline');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.02 seconds


ORACLE Application Express
Application Builder
SQL Commands
Rows
100
Clear Comman
INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
VALUES (1010, 'Benner', 'Sanjuana');
Results
Explain
Describe
Saved SQL
History
1 row(s) inserted.
0.00 seconds

```
SELECT *
 FROM vs_tblEmployees
  ORDER BY (EmployeeID);
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEEID	LAST	FIRST
1001	Lininger	Lila
1002	Farney	Julius
1003	Parkerson	Melaine
1004	Pickles	Diana
1005	Bui	Shad
1006	Mehring	Rosalba
1007	Worster	Christian
1008	Richie	Giselle
1009	Betterton	Angeline
1010	Benner	Sanjuana

10 rows returned in 0.01 seconds Download

B2[2]

Rows 1000 | Clear Command | Find Tables | Save | Run

```
SELECT a.StageLName, a.StageFName, g.GenreTitle "Genre", m.MovieName "Movie",
       ca.CastMemberRoleDesc "Cast"

FROM vs_tblMovies m

JOIN vs_tblCastMembers c ON m.MovieID = c.MovieID

JOIN vs_tblCastMemberRoles ca ON c.CastMemberRoleID = ca.CastMemberRoleID

JOIN vs_tblGenreBridge gb ON m.MovieID = gb.MovieID

JOIN vs_tblActors a ON a.ActorID = c.ActorID
```

Results | Explain | Describe | Saved SQL | History | Bottom Splitter

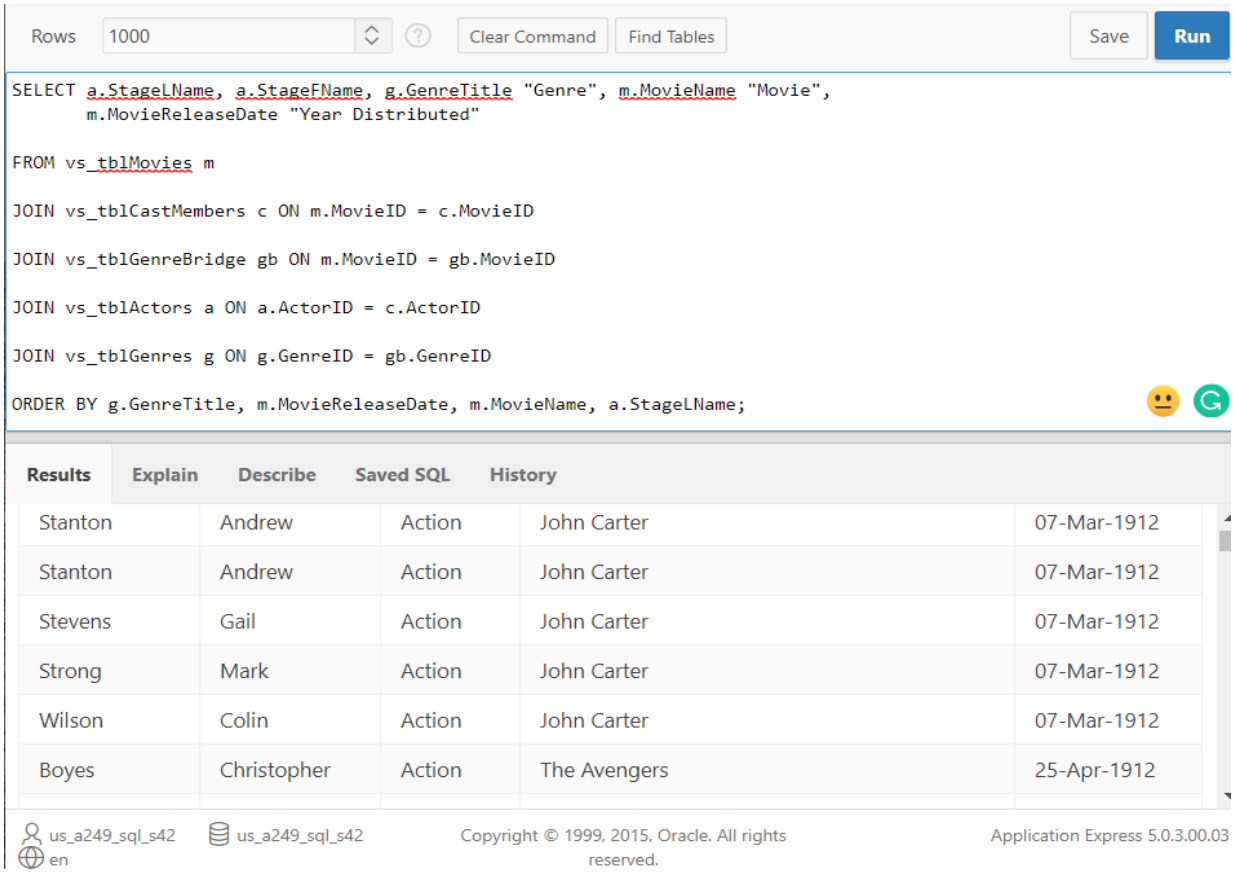
STAGELNAME	STAGEFNAME	Genre	Movie	Cast
Calhern	Louis	Action	Annie Get Your Gun	Col. Buffalo Bill Cody
Freed	Arthur	Action	Annie Get Your Gun	Producer
Hutton	Betty	Action	Annie Get Your Gun	Annie Oakley
Keel	Howard	Action	Annie Get Your Gun	Frank Butler
Naish	Carrol	Action	Annie Get Your Gun	Chief Sitting Bull

us_a249_sql_s42 us_a249_sql_s42 Copyright © 1999, 2015, Oracle. All rights reserved. Application Express 5.0.3.00.0

en

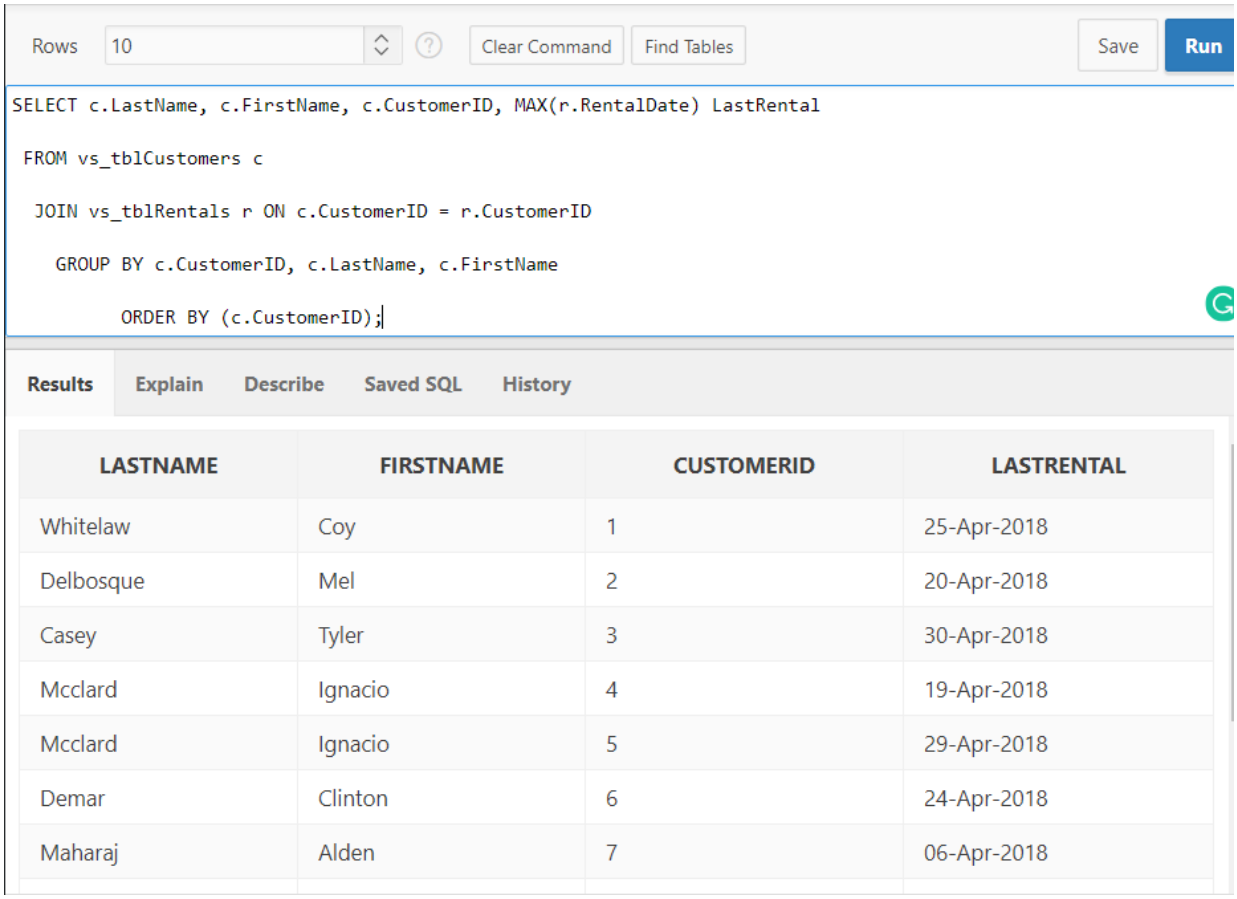
[2] B2 -Actors Role List for actors all their movies (sort by genre and movie title) (and the required columns)

B3[3]



[3] B3 - List of Actors per Movie sorted by Genre, year distributed and title

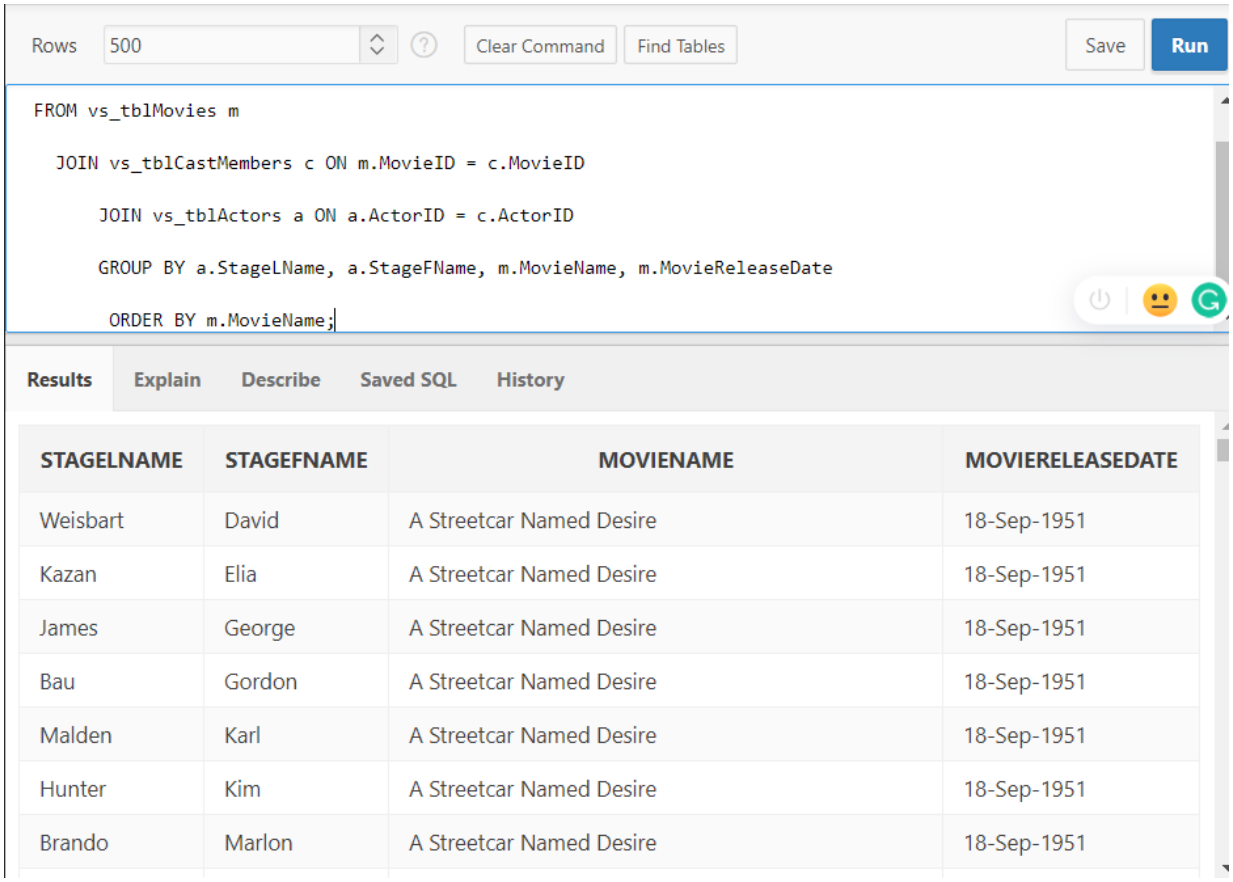
B4[4]



LASTNAME	FIRSTNAME	CUSTOMERID	LASTRENTAL
Whitelaw	Coy	1	25-Apr-2018
Delbosque	Mel	2	20-Apr-2018
Casey	Tyler	3	30-Apr-2018
Mcclard	Ignacio	4	19-Apr-2018
Mcclard	Ignacio	5	29-Apr-2018
Demar	Clinton	6	24-Apr-2018
Maharaj	Alden	7	06-Apr-2018

[4] B4 - List of Customer by last name, first name CustID that have not rented a video in x number of days. {Make sure to include the last rental date and all the customer demographics

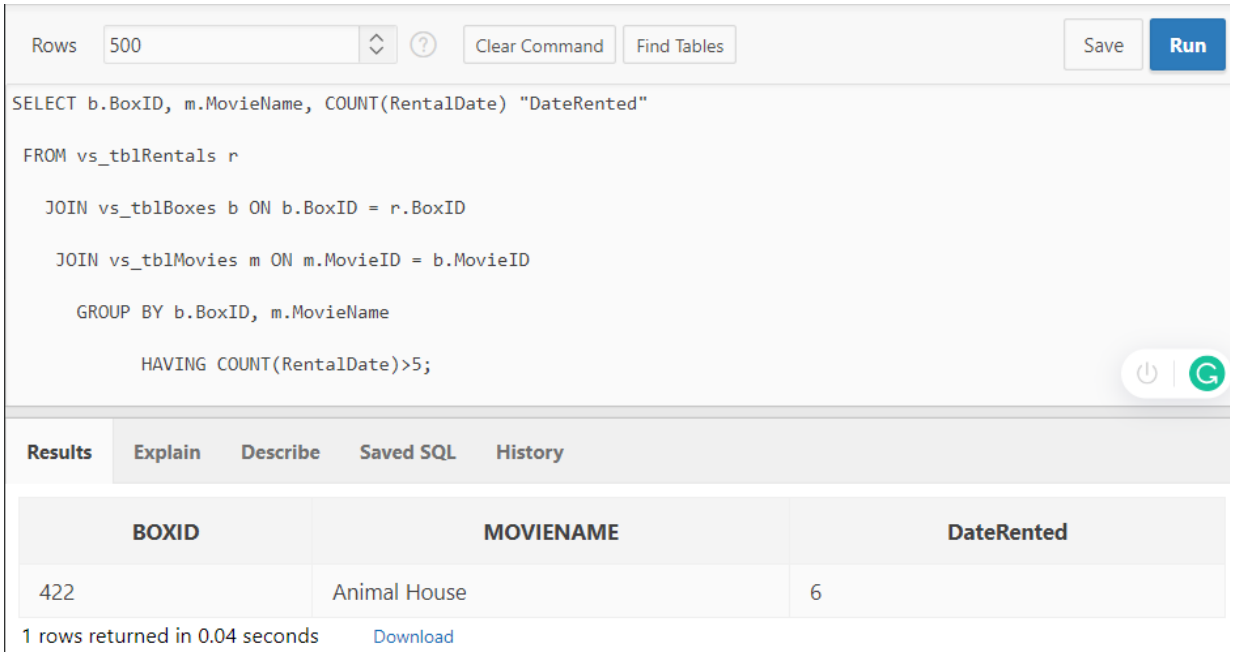
B5[5]



STAGELNAME	STAGEFNAME	MOVIENAME	MOVIERELEASEDATE
Weisbart	David	A Streetcar Named Desire	18-Sep-1951
Kazan	Elia	A Streetcar Named Desire	18-Sep-1951
James	George	A Streetcar Named Desire	18-Sep-1951
Bau	Gordon	A Streetcar Named Desire	18-Sep-1951
Malden	Karl	A Streetcar Named Desire	18-Sep-1951
Hunter	Kim	A Streetcar Named Desire	18-Sep-1951
Brando	Marlon	A Streetcar Named Desire	18-Sep-1951

[5] B5 - For a specific actor list all the other actors they have worked with along with the movie title and year.

B6[6]



[6] Generate a list of boxes that have been rented more than 5 time… Generate 1 row for each rental [BoxID, MovieName DateRented…] Make sure that you have all the required columns, grouped and sort in the proper sequence

B7[7]

Rows 500 Clear Command Find Tables Save Run

```
SELECT m.MovieName "MovieTitle", b.DateBoxReceived, b.RentalCount

FROM vs_tblBoxes b

 JOIN vs_tblMovies m ON m.MovieID = b.MovieID
```

Results Explain Describe Saved SQL History

MovieTitle	DATEBOXRECEIVED	RENTALCOUNT
The Living Daylights	16-Jul-1988	61
The Living Daylights	16-Jul-1988	12
The Living Daylights	16-Jul-1988	60
Tomorrow Never Dies	22-Apr-1999	43
Harry Potter and the Half-Blood Prince	23-Sep-2011	59
Superman Returns	27-Jan-2009	64
Superman Returns	27-Jan-2009	24
Pirates of the Caribbean: On Stranger Tides	22-Sep-2013	96
The Empire Strikes Back	14-Nov-1981	41

[7] B7 - Generate a list which includes the number of boxes per movie. Including Movie Title Dist. Date and count, …]

B8[8]

Rows 500 Clear Command Find Tables Save Run

```
SELECT m.MovieName, d.DistributorName, b.DateBoxReceived, b.RentalCount

  FROM vs_tblMovies m

   JOIN vs_tblDistributors d ON d.DistributorID = m.DistributorID

         JOIN vs_tblBoxes b ON b.MovieID = m.MovieID

     GROUP BY m.MovieName, d.DistributorName, b.DateBoxReceived, b.RentalCount;
```

Results | Explain | Describe | Saved SQL | History

MOVIENAME	DISTRIBUTORNAME	DATEBOXRECEIVED	RENTALCOUNT
A Streetcar Named Desire	Charles K. Feldman Group	22-Nov-1953	87
The Living Daylights	United Artists	16-Jul-1988	59
The Living Daylights	United Artists	16-Jul-1988	41
Harry Potter and the Half-Blood Prince	Warner Bros.	23-Sep-2011	11
The Color Purple	Amblin Entertainment	02-Jan-1987	80
Superman Returns	DC Comics	27-Jan-2009	66

B9[9]

Submit

Bind Variable	Value
:MOVIEID	30
:MOVIENAME	Bank Robbery
:MOVIERELEASEDATE	05 07 2020
:DISTRIBUTORID	4

[8] B8 - Generate a rental count for each movie [Movie Title, Distributor, Dist. Date, count, …]

[9] B9 - List all Movie titles in your inventory that have never been rented. (Included all the required fields to make this a useful business report.)

Submit

Bind Variable	Value
:MOVIEID	31
:MOVIENAME	Money Run
:MOVIERELEASEDATE	05 08 2020
:DISTRIBUTORID	4

Submit

Bind Variable	Value
:MOVIEID	32
:MOVIENAME	Ice Scam
:MOVIERELEASEDATE	05 09 2020
:DISTRIBUTORID	4

Submit

Bind Variable	Value
:MOVIEID	33
:MOVIENAME	Cold Money
:MOVIERELEASEDATE	05 10 2020
:DISTRIBUTORID	4

Submit

Bind Variable	Value
:MOVIEID	34
:MOVIENAME	Fast Cars
:MOVIERELEASEDATE	05 11 2020
:DISTRIBUTORID	4

Rows 5000 Clear Command Find Tables Save Run

```
SELECT *
 FROM vs_tblMovies
  ORDER BY MovieID;
```

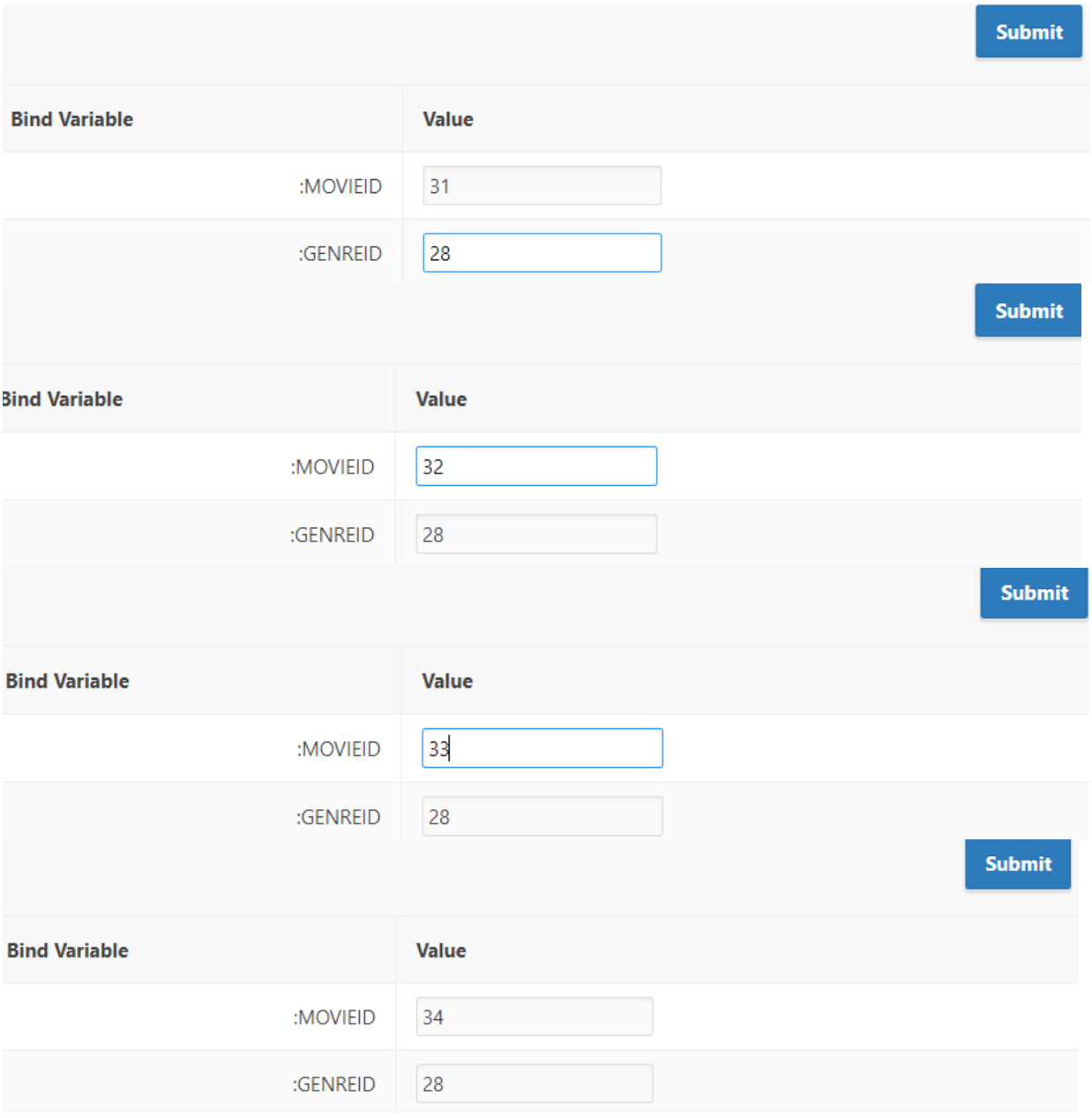
Results Explain Describe Saved SQL History

MOVIEID	MOVIENAME	MOVIERELEASEDATE	DISTRIBUTORID
11	Star Wars	25-May-1977	1
12	Finding Nemo	30-May-2000	3
22	Pirates of the Caribbean: The Curse of the Black Pearl	09-Jul-2003	2
30	Bank Robbery	07-May-2020	4
31	Money Run	08-May-2020	4
32	Ice Scam	09-May-2020	4
33	Cold Money	10-May-2020	4
34	Fast Cars	11-May-2020	4

Submit

Bind Variable	Value
:MOVIEID	30
:GENREID	28

.


Submit
Bind Variable
Value
:MOVIEID
31
:GENREID
28
Submit
Bind Variable
Value
:MOVIEID
32
:GENREID
28
Submit
Bind Variable
Value
:MOVIEID
33
:GENREID
28
Submit
Bind Variable
Value
:MOVIEID
34
:GENREID
28

Rows 5000 Clear Command Find Tables Save Run

```
SELECT *
 FROM vs_tblGenreBridge
  WHERE MovieID BETWEEN 30 AND 34
```

Results Explain Describe Saved SQL History

MOVIEID	GENREID
30	28
31	28
32	28
33	28
34	28

Submit

Bind Variable	Value
:BOXID	800
:MOVIEID	30
:MEDIUMTYPECODE	DVD
:DATEBOXRECEIVED	05 07 2020
:RENTALCOUNT	100

New Value for Bind Variable :BOXID

Submit

Bind Variable	Value
:BOXID	801
:MOVIEID	31
:MEDIUMTYPECODE	DVD
:DATEBOXRECEIVED	05 08 2020
:RENTALCOUNT	20

Submit

Bind Variable	Value
:BOXID	802
:MOVIEID	32
:MEDIUMTYPECODE	DVD
:DATEBOXRECEIVED	05 09 2020
:RENTALCOUNT	20

Submit

Bind Variable	Value
:BOXID	803
:MOVIEID	33
:MEDIUMTYPECODE	DVD
:DATEBOXRECEIVED	05 10 2020
:RENTALCOUNT	20

Submit

Bind Variable	Value
:BOXID	804
:MOVIEID	34
:MEDIUMTYPECODE	DVD
:DATEBOXRECEIVED	05 11 2020
:RENTALCOUNT	20

Rows 5000 Clear Command Find Tables Save Run

```
SELECT *
 FROM vs_tblBoxes
 WHERE BoxID > 799;
```

Results Explain Describe Saved SQL History

BOXID	MOVIEID	MEDIUMTYPECODE	DATEBOXRECEIVED	RENTALCOUNT
800	30	DVD	07-May-2020	100
801	31	DVD	08-May-2020	20
802	32	DVD	09-May-2020	20
803	33	DVD	10-May-2020	20
804	34	DVD	11-May-2020	20

5 rows returned in 0.02 seconds Download

Rows 5000 Clear Command Find Tables Save Run

```
SELECT m.MovieName "MovieTitle", m.MovieReleaseDate "MovieDate", d.DistributorName "Distributor",
 g.GenreTitle "Genre"

  FROM vs_tblMovies m

   JOIN vs_tblDistributors d ON d.DistributorID = m.DistributorID

   JOIN vs_tblGenreBridge gb ON gb.MovieID = m.MovieID

   JOIN vs_tblGenres g ON g.GenreID = gb.GenreID

   JOIN vs_tblBoxes b ON b.MovieID = m.MovieID
```

Results Explain Describe Saved SQL History

MovieTitle	MovieDate	Distributor	Genre
Cold Money	10-May-2020	Paramount Pictures	Action
Fast Cars	11-May-2020	Paramount Pictures	Action
Bank Robbery	07-May-2020	Paramount Pictures	Action
Ice Scam	09-May-2020	Paramount Pictures	Action
Money Run	08-May-2020	Paramount Pictures	Action

5 rows returned in 0.05 seconds Download

B10[10]

Rows 5000 Clear Command Find Tables Save Run

```
SELECT COUNT(m.MovieName) "MovieTitle", m.MovieReleaseDate "MovieDate", d.DistributorName "Distributor",
 g.GenreTitle "Genre"

  FROM vs_tblMovies m

   JOIN vs_tblDistributors d ON d.DistributorID = m.DistributorID

   JOIN vs_tblGenreBridge gb ON gb.MovieID = m.MovieID
```

Results Explain Describe Saved SQL History

MovieTitle	MovieDate	Distributor	Genre
1	10-May-2020	Paramount Pictures	Action
1	07-May-2020	Paramount Pictures	Action
1	11-May-2020	Paramount Pictures	Action
1	08-May-2020	Paramount Pictures	Action
1	09-May-2020	Paramount Pictures	Action
1	07-Feb-1974	Warner Bros.	Comedy
1	07-Feb-1974	Warner Bros.	Western

7 rows returned in 0.03 seconds Download

[10] B10 - List all movie titles that you have only 1 copy in your inventory. (Included all the required fields to make this a useful business report.)

Appendix - Scripts

B1

```
CREATE TABLE vs_tblEmployees
(  EmployeeID NUMBER (10)
   Last VARCHAR2 (15),
   First VARCHAR2 (10));

ALTER TABLE vs_tblEmployees
ADD CONSTRAINT vs_tblEmployees_EmployeeID_pk PRIMARY KEY (EmployeeID);

ALTER TABLE vs_tblRentals
 ADD (EmployeeID NUMBER (10));

ALTER TABLE vs_tblRentals
 ADD CONSTRAINT vs_tblRentals_EmployeeID_fk FOREIGN KEY (EmployeeID)
    REFERENCES vs_tblEmployees (EmployeeID));

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1001, 'Lininger', 'Lila');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1002, 'Farney', 'Julius');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1003, 'Parkerson', 'Melaine');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1004, 'Pickles', 'Diana');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1005, 'Bui', 'Shad');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1006, 'Mehring', 'Rosalba');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1007, 'Worster', 'Christian');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1008, 'Richie', 'Giselle');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1009, 'Betterton', 'Angeline');

 INSERT INTO vs_tblEmployees (EmployeeID, Last, First)
  VALUES (1010, 'Benner', 'Sanjuana');
```

B2

```
SELECT a.StageLName, a.StageFName, g.GenreTitle "Genre", m.MovieName "Movie",
    ca.CastMemberRoleDesc "Cast"

FROM vs_tblMovies m

JOIN vs_tblCastMembers c ON m.MovieID = c.MovieID

JOIN vs_tblCastMemberRoles ca ON c.CastMemberRoleID = ca.CastMemberRoleID

JOIN vs_tblGenreBridge gb ON m.MovieID = gb.MovieID

JOIN vs_tblActors a ON a.ActorID = c.ActorID

JOIN vs_tblGenres g ON g.GenreID = gb.GenreID

ORDER BY g.GenreTitle, m.MovieName, a.StageLName;
```

B4

```
SELECT a.StageLName, a.StageFName, g.GenreTitle "Genre", m.MovieName "Movie",
    m.MovieReleaseDate "Year Distributed"

FROM vs_tblMovies m

JOIN vs_tblCastMembers c ON m.MovieID = c.MovieID

JOIN vs_tblGenreBridge gb ON m.MovieID = gb.MovieID

JOIN vs_tblActors a ON a.ActorID = c.ActorID

JOIN vs_tblGenres g ON g.GenreID = gb.GenreID

ORDER BY g.GenreTitle, m.MovieReleaseDate, m.MovieName, a.StageLName;
```

B5

```
SELECT c.LastName, c.FirstName, c.CustomerID, MAX(r.RentalDate) LastRental

 FROM vs_tblCustomers c

  JOIN vs_tblRentals r ON c.CustomerID = r.CustomerID

    GROUP BY c.CustomerID, c.LastName, c.FirstName

            ORDER BY (c.CustomerID); data

       WHERE (Sysdate - 14) > data.LastRental;
```

B6

```
SELECT b.BoxID, m.MovieName, COUNT(RentalDate)

 FROM vs_tblRentals r

  JOIN vs_tblBoxes b ON b.BoxID = r.BoxID

   JOIN vs_tblMovies m ON m.MovieID = b.MovieID

    GROUP BY b.BoxID, m.MovieName

            HAVING COUNT(RentalDate)>5;
```

B7

```
SELECT m.MovieName "MovieTitle", b.DateBoxReceived, b.RentalCount

 FROM vs_tblBoxes b

  JOIN vs_tblMovies m ON m.MovieID = b.MovieID

   GROUP BY m.MovieName, b.DateBoxReceived, b.RentalCount;
```

B8

```
SELECT m.MovieName, d.DistributorName, b.DateBoxReceived, b.RentalCount

  FROM vs_tblMovies m

   JOIN vs_tblDistributors d ON d.DistributorID = m.DistributorID

          JOIN vs_tblBoxes b ON b.MovieID = m.MovieID

    GROUP BY m.MovieName, d.DistributorName, b.DateBoxReceived, b.RentalCount;
```

B9

```
SELECT m.MovieName "MovieTitle", m.MovieReleaseDate "MovieDate", d.DistributorName
"Distributor",
 g.GenreTitle "Genre"

  FROM vs_tblMovies m

   JOIN vs_tblDistributors d ON d.DistributorID = m.DistributorID

   JOIN vs_tblGenreBridge gb ON gb.MovieID = m.MovieID

   JOIN vs_tblGenres g ON g.GenreID = gb.GenreID

   JOIN vs_tblBoxes b ON b.MovieID = m.MovieID

   WHERE m.MovieID BETWEEN 30 AND 34

    GROUP BY m.MovieName, m.MovieReleaseDate,
          d.DistributorName, g.GenreTitle;


INSERT INTO vs_tblMovies (MovieID, MovieName, MovieReleaseDate, DistributorID)
          VALUES (:MovieID, :MovieName, TO_DATE(:MovieReleaseDate, 'MM DD
YYYY'),:DistributorID);

INSERT INTO vs_tblGenreBridge (MovieID, GenreID)
          VALUES (:MovieID, :GenreID);

INSERT INTO vs_tblBoxes (BoxID, MovieID, MediumTypeCode, DateBoxReceived, RentalCount)
          VALUES (:BoxID, :MovieID, :MediumTypeCode, TO_DATE(:DateBoxReceived, 'MM DD YYYY'),
:RentalCount);
```

B10

```
SELECT COUNT(m.MovieName) "MovieTitle", m.MovieReleaseDate "MovieDate", d.DistributorName
"Distributor",
 g.GenreTitle "Genre"

  FROM vs_tblMovies m

   JOIN vs_tblDistributors d ON d.DistributorID = m.DistributorID

   JOIN vs_tblGenreBridge gb ON gb.MovieID = m.MovieID

   JOIN vs_tblGenres g ON g.GenreID = gb.GenreID

   JOIN vs_tblBoxes b ON b.MovieID = m.MovieID

    GROUP BY m.MovieName, m.MovieReleaseDate,
          d.DistributorName, g.GenreTitle

      HAVING COUNT(m.MovieName)=1;
```