

Assignment 2 Analysing Game Sales Across Regions and Platforms

Task 1:

Insert a new game with the title "Future Racing", genre "Racing", release date "2024-10-01", and developer "Speed Studios":

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. Below the navigation bar is the 'Navigator' pane, which displays the database schema. Under the 'SCHEMAS' section, 'coursesintshl' is selected. Under 'coursesintshl', there are 'employees', 'games', and 'game sales - game s'. Under 'games', there are 'Views', 'Stored Procedures', and 'Functions'. The 'internships' table is also listed under 'games'. The 'Information' tab is selected. The main workspace contains a query editor titled 'Assignment 2_SQL Game-Gam...'. The query is:

```
1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
```

Below the query editor is the 'Output' pane, which shows the execution log:

#	Time	Action	Message	Duration / Fetch
58	17:52:31	select * from games	150 row(s) returned	0.000 sec / 0.000 sec
59	17:54:26	insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios')	1 row(s) affected	0.000 sec

The status bar at the bottom left says 'Query Completed'.

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane is identical to the previous screenshot. The main workspace now displays a 'Result Grid' for the 'games' table. The grid has columns: GameID, GameTitle, Genre, ReleaseDate, and Developer. The data is as follows:

GameID	GameTitle	Genre	ReleaseDate	Developer
148	Cross-group 24/7 benchmark	Strategy	2023-05-30	SeaWave Games
149	Organized dynamic moderator	Strategy	2022-09-08	Sunrise Interactive
150	Front-line regional hierarchy	Strategy	2023-01-18	SeaWave Games
5001	Future Racing	Racing	2024-10-01	Speed Studios

Below the result grid is the 'Output' pane, which shows the execution log:

#	Time	Action	Message	Duration / Fetch
55	17:40:41	DEALLOCATE PREPARE stmt	OK	0.000 sec
56	17:51:28	select * from games	150 row(s) returned	0.015 sec / 0.000 sec
57	17:51:44	select * from gamesales	150 row(s) returned	0.016 sec / 0.000 sec
58	17:52:31	select * from games	150 row(s) returned	0.000 sec / 0.000 sec
59	17:54:26	insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios')	1 row(s) affected	0.000 sec
60	17:55:26	select * from games	151 row(s) returned	0.000 sec / 0.000 sec

The status bar at the bottom left says 'Object Info' and 'Session'.

Query:

```
insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
```

Task 2:

Update the price of the game with GameID 2 on the PlayStation platform to 60:

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the following SQL code is written:

```
1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
```

In the Output pane, the results of the last two statements are displayed:

#	Time	Action	Message	Duration / Fetch
67	18:06:23	select * from gamesales	150 row(s) returned	0.000 sec / 0.000 sec
68	18:06:30	update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation'	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the same SQL code is present:

```
1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
```

In the Result Grid pane, the data from the gamesales table is shown:

GameID	Platform	SalesRegion	UnitsSold	Price
1	Xbox	North America	861216	45.39
2	PlayStation	North America	444626	60
3	PlayStation	North America	569637	34.2
4	PlayStation	South America	977779	43.9
5	PC	South America	473349	21.61
6	PC	Europe	214870	37.55
7	PC	Europe	644295	33.37

In the Output pane, the results of the update statement are displayed:

#	Time	Action	Message	Duration / Fetch
68	18:06:30	update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation'	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
69	18:07:00	select * from gamesales	150 row(s) returned	0.000 sec / 0.000 sec

Query:

```
update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
```

Task 3:

Delete the record of the game with GameID 5 from the Game Sales table:

The screenshot shows the MySQL Workbench interface. In the SQL editor tab titled "Assignment 2 - SQL Game-Gam...", the following SQL code is written:

```
1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
5 • delete from gamesales where gameid = 5;
```

The Output pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
69	18:07:00	select * from gamesales	150 row(s) returned	0.000 sec / 0.000 sec
70	18:18:40	delete from gamesales where gameid = 5	1 row(s) affected	0.000 sec

The screenshot shows the MySQL Workbench interface. In the SQL editor tab titled "Assignment 2 - SQL Game-Gam...", the same SQL code is present:

```
1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
5 • delete from gamesales where gameid = 5;
```

The Result Grid pane displays the data from the "gamesales" table:

GameID	Platform	SalesRegion	UnitsSold	Price
1	Xbox	North America	861216	45.39
2	PlayStation	North America	444626	60
3	PlayStation	North America	569637	34.2
4	PlayStation	South America	977779	43.9
6	PC	Europe	214870	37.55
7	PC	Europe	644295	33.37
8	PlayStation	Asia	105599	48.93

The Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
70	18:18:40	delete from gamesales where gameid = 5	1 row(s) affected	0.000 sec
71	18:19:09	select * from gamesales	149 row(s) returned	0.000 sec / 0.000 sec

Query:

delete from gamesales where gameid = 5;

Task 4:

Calculate the total number of units sold for each game across all platforms and regions:

```

MySQL Workbench Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator: Assignment 2_SQL Game-Gam...
Schemas courseintshl employees games
Tables game sales - game s games - games Views Stored Procedures Functions Internalschema
Administration Schemas
Information Schema: games
Object Info Session
Query Completed

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** courseintshl, employees, games
- Tables:** game sales - game s, games - games
- Query:**

```

1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
5 • delete from gamesales where gameid = 5;
6

```
- Result Grid:** Shows the results of the initial query.
- Output:** Shows the execution log with three entries.

GameID	Platform	SalesRegion	UnitsSold	Price
1	Xbox	North America	861216	45.39
2	PlayStation	North America	444626	60
3	PlayStation	North America	569637	34.2
4	PlayStation	South America	977779	43.9
6	PC	Europe	214870	37.55
7	PC	Europe	644295	33.37
8	PlayStation	Asia	105599	48.93

```

MySQL Workbench Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator: Assignment 2_SQL Game-Gam...
Schemas courseintshl employees games
Tables game sales - game s games - games Views Stored Procedures Functions Internalschema
Administration Schemas
Information Schema: games
Object Info Session
Query Completed

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** courseintshl, employees, games
- Tables:** game sales - game s, games - games
- Query:**

```

1 • select * from games;
2 • select * from gamesales;
3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
5 • delete from gamesales where gameid = 5;
6 • select g.gameid,g.gametitle,gs.platform,gs.SalesRegion,sum(gs.unitsold) as Number_of_Units_Sold from games g
7 join gamesales gs ON g.gameID=gs.gameID group by g.gameID, g.gametitle, gs.platform, gs.SalesRegions

```
- Result Grid:** Shows the results of the modified query.
- Output:** Shows the execution log with two entries.

gameid	gametitle	platform	SalesRegion	Number_of_Units_Sold
1	Right-sized human-resource forecast	Xbox	North America	861216
2	Optional tertiary frame	PlayStation	North America	444626
3	Team-oriented static standardization	PlayStation	North America	569637
4	Inverse mobile database	PlayStation	South America	977779
6	Integrated tertiary installation	PC	Europe	214870
7	Team-oriented non-volatile initiative	PC	Europe	644295
8	Quality-focused static frame	PlayStation	Asia	105599

Query:

```

select g.gameid,g.gametitle,gs.platform,gs.SalesRegion,sum(gs.unitssold) as
Number_of_Units_Sold from games g join gamesales gs ON g.gameID=gs.gameid group by
g.gameID, g.gameTitle, gs.platform, gs.SalesRegion;

```

Task 5:

Identify the game with the highest number of units sold in North America:

gameid	gametitle	SalesRegion	Highest_Number_Unitsold
99	Face-to-face eco-centric throughput	North America	983132

Output

#	Time	Action	Message	Duration / Fetch
90	18:49:42	select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc limit 1 ;	1 row(s) returned	0.000 sec / 0.000 sec
91	18:51:47	select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc ;	1 row(s) returned	0.000 sec / 0.000 sec

gameid	gametitle	SalesRegion	Highest_Number_Unitsold
99	Face-to-face eco-centric throughput	North America	983132
21	Digitized client-driven Graphic Interface	North America	936854
52	Implemented system-worthy database	North America	919762
138	Persistent fault-tolerant budgetary management	North America	865487
65	Front-line high-level knowledge user	North America	863940
1	Right-sized human-resource forecast	North America	861216
37	Function-based upward-trending knowledge user	North America	850188

Output

#	Time	Action	Message	Duration / Fetch
91	18:51:47	select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc ;	1 row(s) returned	0.000 sec / 0.000 sec
92	18:52:08	select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc ;	41 row(s) returned	0.016 sec / 0.000 sec

Query:

```

select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold
from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc limit 1 ;

```

Task 6:

Get the game titles, platforms, and sales regions along with the units sold for each game:

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Assignment 2 SQL Game-Gam...
- Schemas:** courseintshl, employees, games (selected), game sales - game s, games - games, Views, Stored Procedures, Functions, Internalschema.
- Code Area:** Contains the following SQL code:

```

3 • insert into games (gametitle,genre,releasedate,developer) values('Future Racing','Racing','2024-10-01','Speed Studios');
4 • update gamesales set price = 60 where gameID = 2 and Platform = 'PlayStation';
5 • delete from gamesales where gameid = 5;
6 • select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc ;
8 • select g.gametitle, gs.platform, gs.salesregion, gs.unitssold from gamesales gs left join games g on gs.gameid = g.gameid;
9

```
- Result Grid:** Shows the results of the last query:

gametitle	platform	salesregion	unitsold
Right-sized human-resource forecast	Xbox	North America	861216
Optional tertiary frame	PlayStation	North America	444626
Team-oriented static standardization	PlayStation	North America	569637
Inverse mobile database	PlayStation	South America	977779
Integrated tertiary installation	PC	Europe	214870
Team-oriented non-volatile initiative	PC	Europe	644295
Quality-focused static frame	PlayStation	Asia	105599
- Output:** Action Output table showing two log entries:

#	Time	Action	Message	Duration / Fetch
101	19:16:33	select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc ;	41 row(s) returned	0.000 sec / 0.000 sec
102	19:16:42	select g.gametitle, gs.platform, gs.salesregion, gs.unitssold from gamesales gs left join games g on gs.gameid = g.gameid;	149 row(s) returned	0.000 sec / 0.000 sec

Query:

```

select g.gametitle, gs.platform, gs.salesregion, gs.unitssold from gamesales gs left join games
g on gs.gameid = g.gameid;

```

Task 7:

Find all games, including those that have no sales data in the Game Sales table:

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **games** selected.
- SQL Editor:** Contains the following SQL code:

```
5 • delete from gamesales where gameid = 5;
6 • select g.gameid, g.gametitle, gs.SalesRegion,max(gs.unitssold) as Highest_Number_Unitsold from games g join gamesales gs on g.gameid=gs.gameid
7 where gs.salesregion= 'North America' group by g.gameid,g.gametitle,gs.salesregion order by highest_number_unitsold desc ;
8 • select g.gameid, g.gametitle, gs.platform, gs.salesregion, gs.unitssold from gamesales gs left join games g on gs.gameid = g.gameid;
9 • select g.gameid, g.gametitle, coalesce(gs.platform, 'No Sales') as Platform, coalesce(gs.salesregion, 'No Sales') as SalesRegion,
10 coalesce(gs.unitssold, 0) as units_sold from games g
11 left join gamesales gs on g.gameid = gs.gameid;
```
- Result Grid:** Displays the results of the query, showing 151 rows. The columns are **gameid**, **gametitle**, **Platform**, **SaleRegion**, and **units_sold**. The results include various game titles and their sales data across different platforms and regions.
- Action Output:** Shows two log entries with timestamp, action, message, and duration.

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **games** selected.
- SQL Editor:** Contains the following SQL code:

```
8 • select g.gameid, gs.platform, gs.salesregion, gs.unitssold from gamesales gs left join games g on gs.gameid = g.gameid;
9 • select g.gameid, g.gametitle, coalesce(gs.platform, 'No Sales') as Platform, coalesce(gs.salesregion, 'No Sales') as SalesRegion,
10 coalesce(gs.unitssold, 0) as units_sold from games g left join gamesales gs on g.gameid = gs.gameid;
11 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
12 gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid;
```
- Result Grid:** Displays the results of the query, showing 151 rows. The columns are **gameid**, **gametitle**, **Platform**, **salesregion**, and **unitssold**. The results are identical to the first query.
- Action Output:** Shows two log entries with timestamp, action, message, and duration.

Query:

```
select g.gameid, g.gametitle, coalesce(gs.platform, 'No Sales') as Platform,
coalesce(gs.salesregion, 'No Sales') as SalesRegion, coalesce(gs.unitssold, 0) as units_sold
from games g left join gamesales gs on g.gameid = gs.gameid;

select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid;
```

Task 8:

Retrieve sales records where the game details are missing in the Games table:

The screenshot shows the MySQL Workbench interface. The query editor window contains the following SQL code:

```
9 • select g.gameid, g.gametitle, coalesce(gs.platform, 'No Sales') as Platform, coalesce(gs.salesregion, 'No Sales') as SalesRegion,
10 • coalesce(gs.unitssold, 0) as units_sold from games g left join gamesales gs on g.gameid = gs.gameid;
11 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
12 • gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid;
13 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
14 • gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid where g.gameid = null;
15
```

The results grid shows the following columns: gameid, gametitle, Platform, salesregion, unitssold. The output pane shows two log entries:

Action Output	Message	Duration / Fetch
127 21:12:50	select g.gameid, g.gametitle, gs.Platform, gs.salesregion, gs.unitssold from games g... 151 row(s) returned	0.000 sec / 0.000 sec
128 21:13:03	select g.gameid, g.gametitle, gs.Platform, gs.salesregion, gs.unitssold from games g... 0 row(s) returned	0.000 sec / 0.000 sec

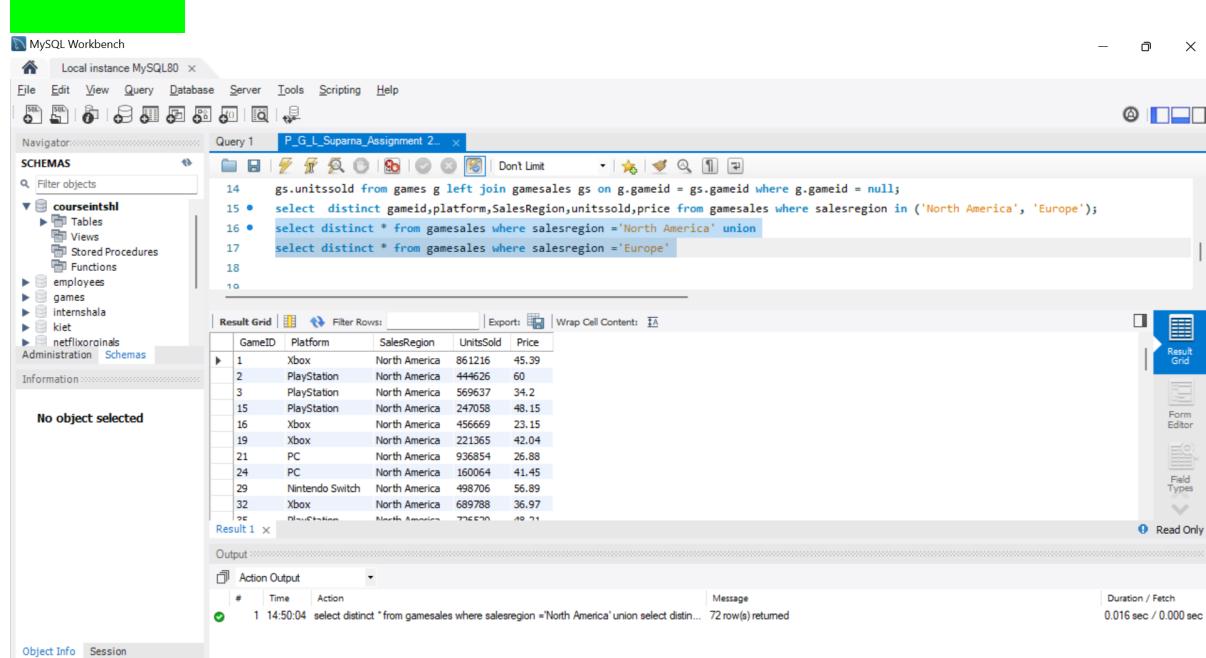
Query:

```
select g.gameid, g.gametitle, gs.Platform, gs.salesregion, gs.unitssold from games g
left join gamesales gs on g.gameid = gs.gameid where g.gameid = null;
```

Task 9:

Retrieve game sales data for North America and Europe removing duplicate records:

With UNION:



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid where g.gameid = null;
select distinct * from gamesales where salesregion in ('North America', 'Europe');
select distinct * from gamesales where salesregion = 'North America' union
select distinct * from gamesales where salesregion = 'Europe'.
```

The results grid displays the following data:

#	GameID	Platform	SalesRegion	UnitsSold	Price
1	Xbox	North America	861216	45.39	
2	PlayStation	North America	444626	60	
3	PlayStation	North America	569637	34.2	
15	PlayStation	North America	247058	48.15	
16	Xbox	North America	456669	23.15	
19	Xbox	North America	221365	42.04	
21	PC	North America	936854	26.88	
24	PC	North America	160064	41.45	
29	Nintendo Switch	North America	498706	56.89	
32	Xbox	North America	689788	36.97	
33	PlayStation	North America	742579	40.71	

The output pane shows the execution message: "1 14:50:04 select distinct * from gamesales where salesregion = 'North America' union select distin... 72 row(s) returned".

Query:

```
select distinct * from gamesales where salesregion ='North America' union
select distinct * from gamesales where salesregion ='Europe';
```

With IN:

The screenshot shows the MySQL Workbench interface with a query editor titled "Assignment 2 SQL Game-Gam...". The code entered is:

```
9 • select g.gameid, g.gametitle, coalesce(gs.platform, 'No Sales') as Platform, coalesce(gs.salesregion, 'No Sales') as SalesRegion,
10 •     coalesce(gs.unitssold, 0) as units_sold from games g left join gamesales gs on g.gameid = gs.gameid;
11 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
12 •     gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid;
13 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
14 •     gs.unitssold from games g left join gamesales gs on g.gameid = gs.gameid where g.gameid = null;
15 • select distinct gameid,platform,SalesRegion,unitssold,price from gamesales where salesregion in ('North America', 'Europe');
```

The results grid shows the following data:

gameid	platform	SalesRegion	unitssold	price
1	Xbox	North America	861216	45.39
2	PlayStation	North America	444626	60
3	PlayStation	North America	569637	34.2
6	PC	Europe	214870	37.55
7	PC	Europe	644295	33.37
9	Nintendo Switch	Europe	472076	46.39
14	PC	Europe	785993	23.56
15	PlayStation	North America	247058	48.15

The output section shows the following execution details:

#	Time	Action	Message	Duration / Fetch
5	14:32:08	select * from gamesales	149 row(s) returned	0.000 sec / 0.000 sec
6	14:32:16	select distinct gameid,platform,SalesRegion,unitssold,price from gamesales where s...	72 row(s) returned	0.016 sec / 0.000 sec

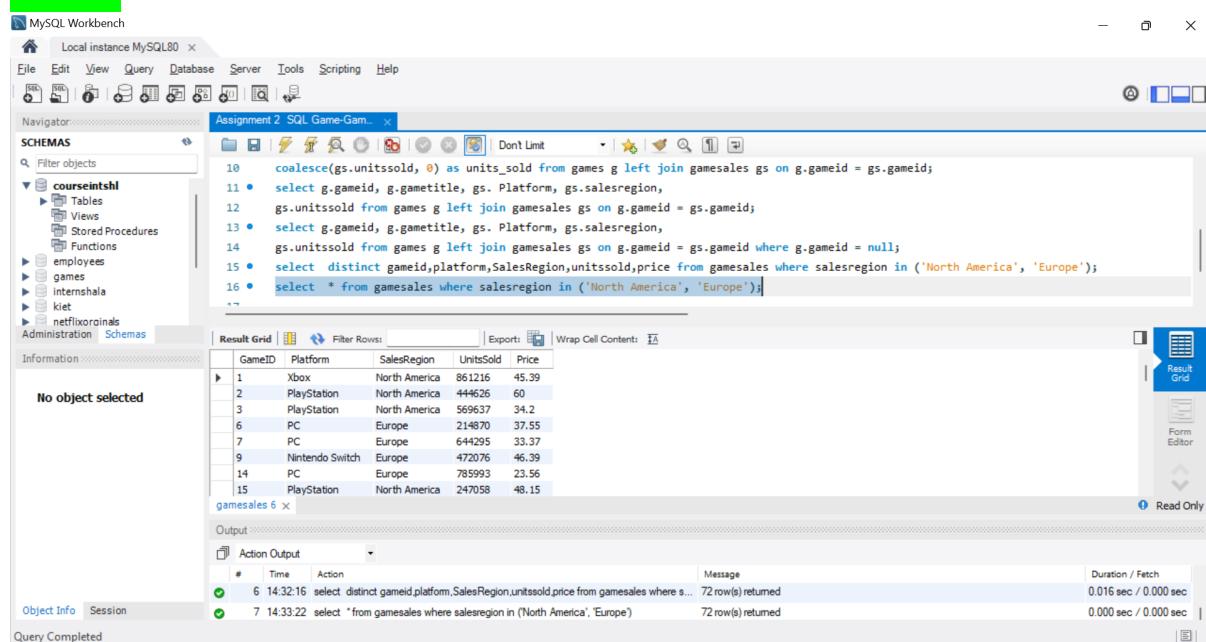
Query:

```
select distinct gameid,platform,salesregion,unitssold,price from gamesales where
salesregion in ('North America', 'Europe');
```

Task 10:

Retrieve all game sales data from North America and Europe without removing duplicate records:

With IN:



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
10 • coalesce(gs.unitsold, 0) as units_sold from games g left join gamesales gs on g.gameid = gs.gameid;
11 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
12 gs.unitsold from games g left join gamesales gs on g.gameid = gs.gameid;
13 • select g.gameid, g.gametitle, gs.Platform, gs.salesregion,
14 gs.unitsold from games g left join gamesales gs on g.gameid = gs.gameid where g.gameid = null;
15 • select distinct gameid,platform,SalesRegion,unitsold,price from gamesales where salesregion in ('North America', 'Europe');
16 • select * from gamesales where salesregion in ('North America', 'Europe');
```

The results grid displays the following data:

	GameID	Platform	SalesRegion	UnitsSold	Price
1	Xbox	North America	861216	45.39	
2	PlayStation	North America	444626	60	
3	PlayStation	North America	569637	34.2	
6	PC	Europe	214870	37.55	
7	PC	Europe	644295	33.37	
9	Nintendo Switch	Europe	472076	46.39	
14	PC	Europe	785993	23.56	
15	PlayStation	North America	247058	48.15	

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
6	14:32:16	select distinct gameid,platform,SalesRegion,unitsold,price from gamesales where salesregion in ('North America', 'Europe')	72 row(s) returned	0.016 sec / 0.000 sec
7	14:33:22	select * from gamesales where salesregion in ('North America', 'Europe')	72 row(s) returned	0.000 sec / 0.000 sec

Query:

```
select * from gamesales where salesregion in ('North America', 'Europe');
```

With UNION ALL:

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The 'Query' tab is active, displaying a query titled 'P_G_L_Suparna_Assignment 2...'. The query uses UNION ALL to combine results from two SELECT statements. The results are displayed in a grid with columns: GameID, Platform, SalesRegion, UnitsSold, and Price. The output pane shows the execution log with two entries, both indicating 72 row(s) returned.

GameID	Platform	SalesRegion	UnitsSold	Price
1	Xbox	North America	861216	45.39
2	PlayStation	North America	444626	60
3	PlayStation	North America	569637	34.2
15	PlayStation	North America	247058	48.15
16	Xbox	North America	456669	23.15
19	Xbox	North America	221365	42.04
21	PC	North America	936854	26.88
24	PC	North America	160064	41.45
...

QUERY:

```
select * from gamesales where salesregion ='North America' union all  
select * from gamesales where salesregion ='Europe' ;
```

Prepared By,
Utkarsh Anand