



DELHI TECHNOLOGICAL UNIVERSITY

Name : Shrey
Roll Number : 2K19/MC/122
Subject : Database Management System Laboratory
Faculty : Prof. Aditya Kaushik

PRACTICAL - 4

AIM:

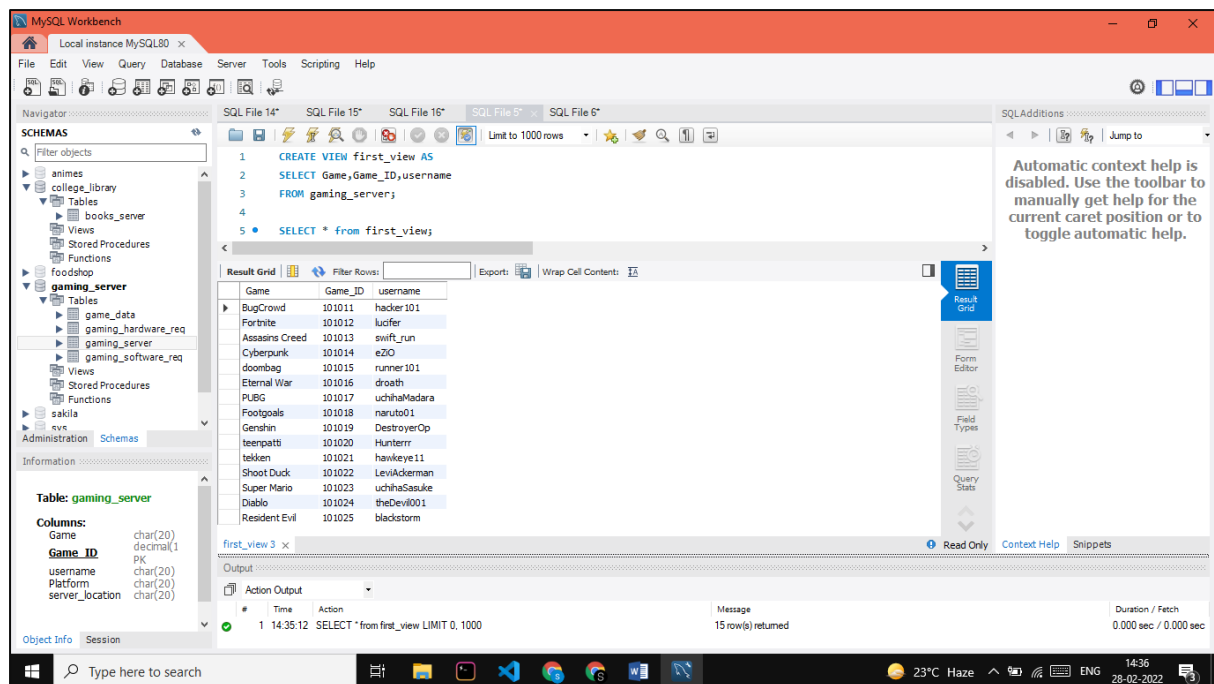
1. Use the tables created in assignment 3.
2. View only a subset of columns from the table.
3. View the data of any table giving conditions for choosing the rows to display.
4. Perform join:
 - a) Join 2 tables and display the details of the joined tables (do not write where condition)
 - b) Fetch the data from the tables by joining 2 and 3 tables.
5. Perform the same operations done in the 4th step by using correlated subqueries.

CODE AND OUTPUT:

Command to create a view as a subset of columns:

```
CREATE VIEW first_view AS  
SELECT Game,Game_ID,username  
FROM gaming_server;  
SELECT * from first_view;
```

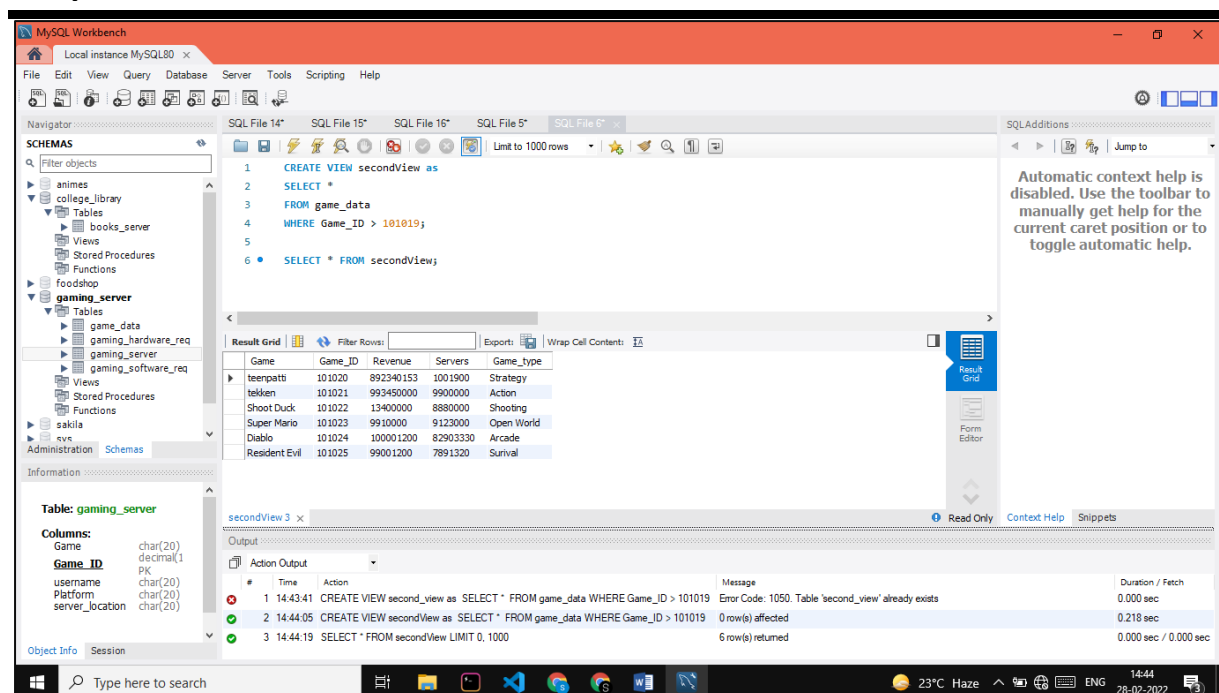
Output :



Command to create a view as a subset of columns with WHERE clause:

```
CREATE VIEW secondView as
SELECT *
FROM game_data
WHERE Game_ID > 101019;
SELECT * FROM secondView;
```

Output :



Command to JOIN 2 tables:

```
SELECT *
FROM game_data as gd
JOIN gaming_server as gs
ON gd.Game_ID = gs.Game_ID;
```

Output :

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

```
SELECT *
FROM game_data as gd
JOIN gaming_server as gs
ON gd.Game_ID = gs.Game_ID;
```

The Results panel displays the output of the query as a table with 10 columns: Game, Game_ID, Revenue, Servers, Game_type, Game, Game_ID, username, Platform, and server_location. The table contains 15 rows of data, including games like BugCrowd, Fortnite, Assassin's Creed, Cyberpunk, doombag, Eternal War, PUBG, Footgoals, Genshin, teenpatti, tekken, Shoot Duck, and Super Mario.

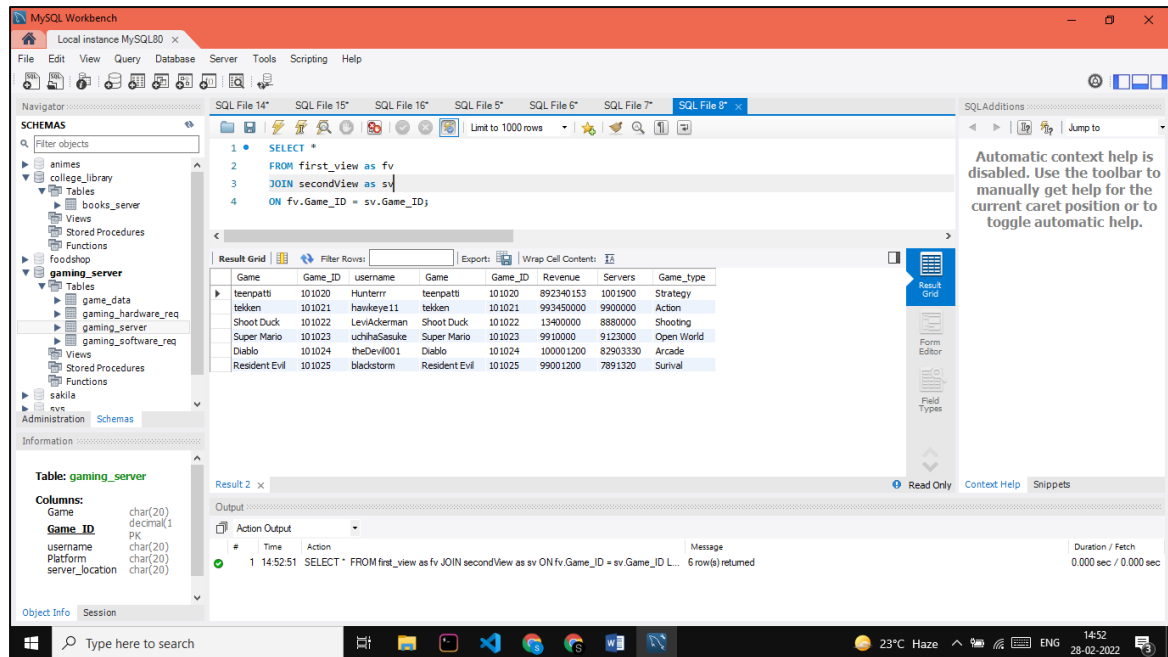
The Output panel shows the execution log with two entries:

#	Time	Action	Message	Duration / Fetch
1	14:47:04	SELECT * FROM game_data as gd JOIN gaming_server as gs WHERE gd.Game_ID = gs...	15 row(s) returned	0.016 sec / 0.000 sec
2	14:49:07	SELECT * FROM game_data as gd JOIN gaming_server as gs ON gd.Game_ID = gs.Game...	15 row(s) returned	0.000 sec / 0.000 sec

Command to JOIN 2 views created above:

```
SELECT *
FROM first_view as fv
JOIN secondView as sv
ON fv.Game_ID = sv.Game_ID;
```

Output :



Command to JOIN 2 tables using correlated subqueries:

```

SELECT *
FROM game_data as gd,gaming_server as gs
WHERE gd.Game_ID = gs.Game_ID;

```

Output :

