**Star Schema Implementation**

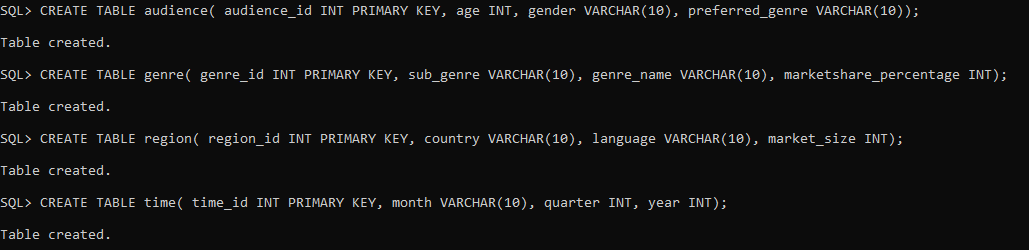
**-- Dimension Tables**

CREATE TABLE audience( audience\_id INT PRIMARY KEY, age INT, gender VARCHAR(10), preferred\_genre VARCHAR(10));

CREATE TABLE genre( genre\_id INT PRIMARY KEY, sub\_genre VARCHAR(10), genre\_name VARCHAR(10), marketshare\_percentage INT);

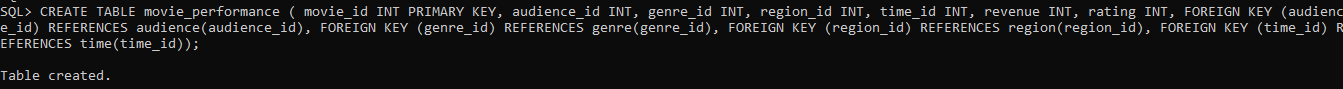
CREATE TABLE region( region\_id INT PRIMARY KEY, country VARCHAR(10), language VARCHAR(10), market\_size INT);

CREATE TABLE time( time\_id INT PRIMARY KEY, month VARCHAR(10), quarter INT, year INT);



**-- Fact Table:**

CREATE TABLE movie\_performance ( movie\_id INT PRIMARY KEY, audience\_id INT, genre\_id INT, region\_id INT, time\_id INT, revenue INT, rating INT, FOREIGN KEY (audience\_id) REFERENCES audience(audience\_id), FOREIGN KEY (genre\_id) REFERENCES genre(genre\_id), FOREIGN KEY (region\_id) REFERENCES region(region\_id), FOREIGN KEY (time\_id) REFERENCES time(time\_id));



**Insertion:**

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (1, 21, 'Male', 'mystery');

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (2, 25, 'Male', 'action');

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (3, 19, 'Female', 'thriller');

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (4, 20, 'Female', 'horror');

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (5, 30, 'Female', 'drama');

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (6, 35, 'Male', 'action');

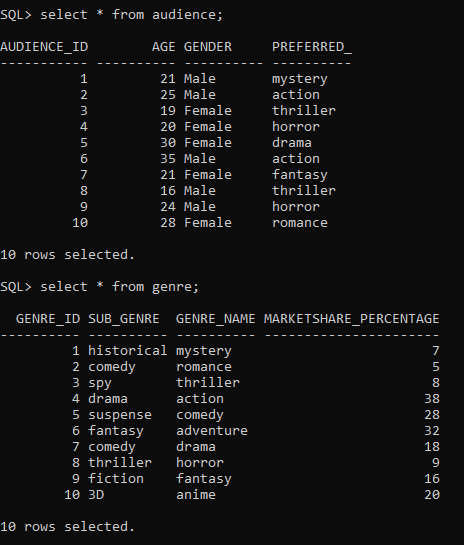
INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (7, 21, 'Female', 'fantasy');

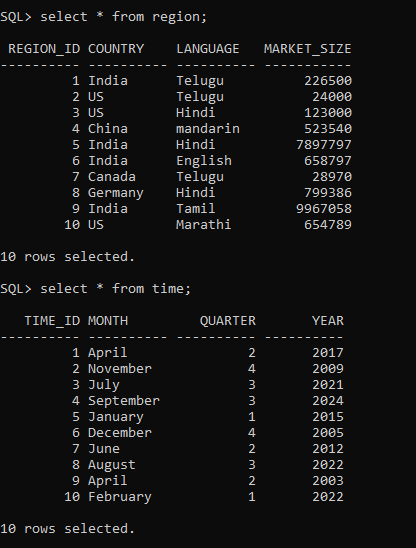
INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (8, 16, 'Male', 'thriller');

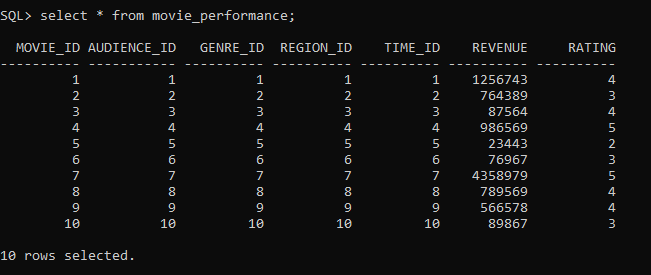
INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (9, 24, 'Male', 'horror');

INSERT INTO audience (audience\_id, age, gender, preferred\_genre) VALUES (10, 28, 'Female', 'romance');

**TABLES:**





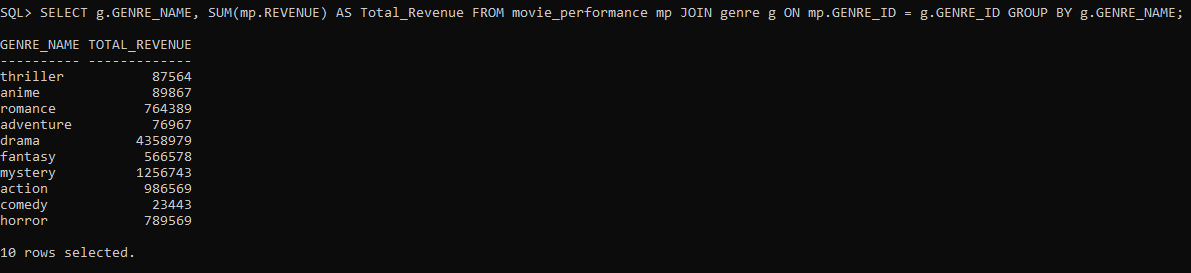


**OLAP Queries Using SQL**

Here are some practical SQL queries to analyse multidimensional data:

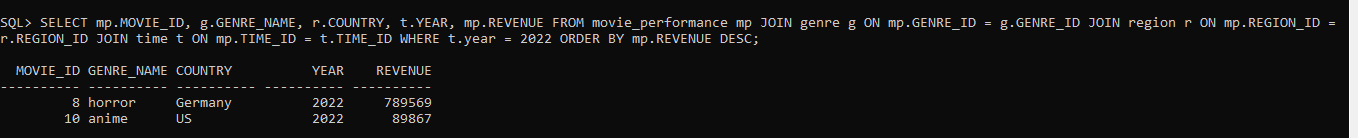
1. **Aggregate Revenue by Genre**

SELECT g.GENRE\_NAME, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID GROUP BY g.GENRE\_NAME;



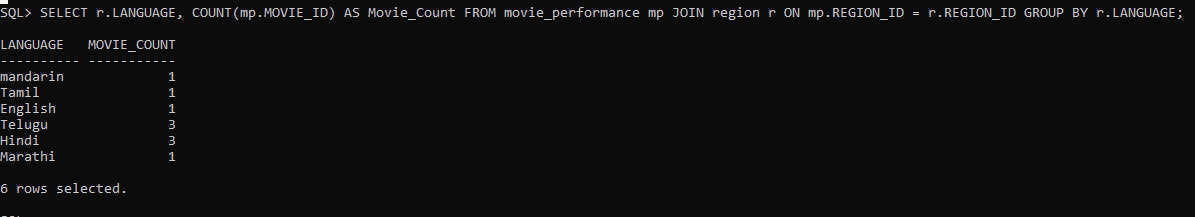
1. **Top 2 movies by Revenue in a Specific Year**

SELECT mp.MOVIE\_ID, g.GENRE\_NAME, r.COUNTRY, t.YEAR, mp.REVENUE FROM movie\_performance mp JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID JOIN region r ON mp.REGION\_ID = r.REGION\_ID JOIN time t ON mp.TIME\_ID = t.TIME\_ID WHERE t.year = 2022 ORDER BY mp.REVENUE DESC;



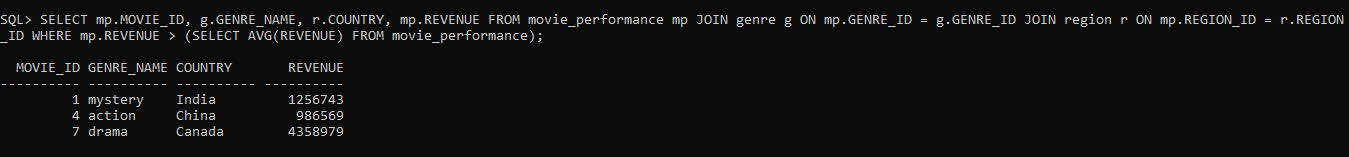
1. **Count of Movies by Language**

SELECT r.LANGUAGE, COUNT(mp.MOVIE\_ID) AS Movie\_Count FROM movie\_performance mp JOIN region r ON mp.REGION\_ID = r.REGION\_ID GROUP BY r.LANGUAGE;



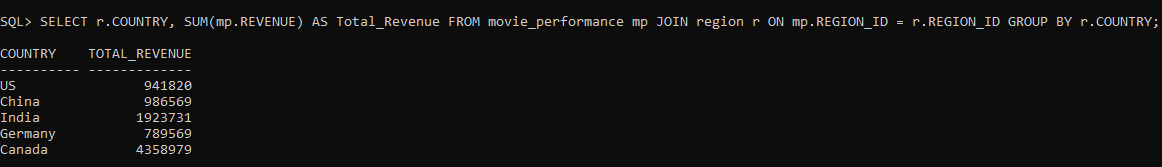
1. **Movies with Above Average Revenue**

SELECT mp.MOVIE\_ID, g.GENRE\_NAME, r.COUNTRY, mp.REVENUE FROM movie\_performance mp JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID JOIN region r ON mp.REGION\_ID = r.REGION\_ID WHERE mp.REVENUE > (SELECT AVG(REVENUE) FROM movie\_performance);



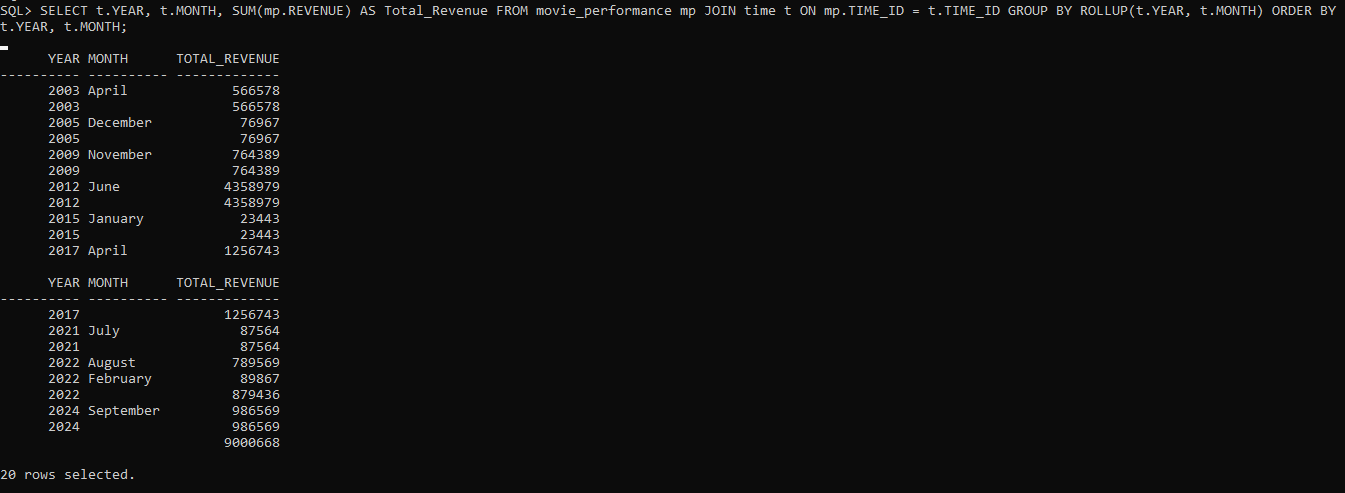
1. **Total Revenue by Country**

SELECT r.COUNTRY, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN region r ON mp.REGION\_ID = r.REGION\_ID GROUP BY r.COUNTRY;



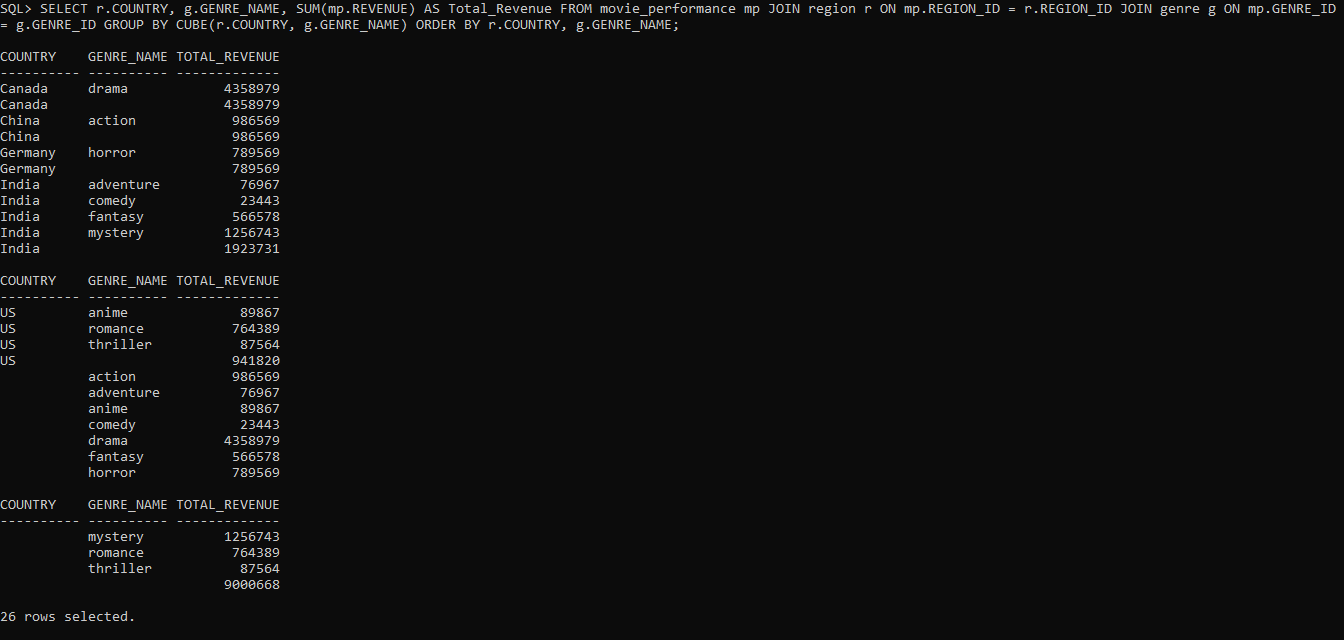
**Using ROLLUP (Hierarchical Aggregation)**

SELECT t.YEAR, t.MONTH, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN time t ON mp.TIME\_ID = t.TIME\_ID GROUP BY ROLLUP(t.YEAR, t.MONTH) ORDER BY t.YEAR, t.MONTH;



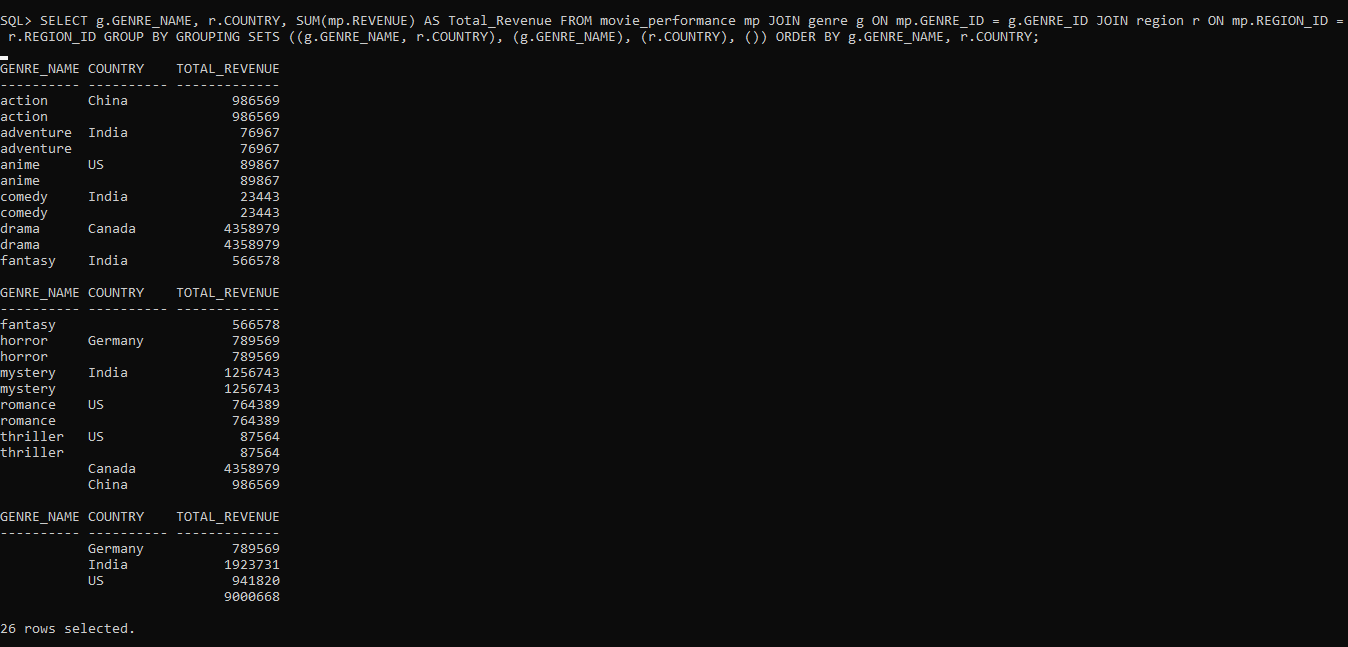
**Using CUBE (Multidimensional Aggregation)**

SELECT r.COUNTRY, g.GENRE\_NAME, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN region r ON mp.REGION\_ID = r.REGION\_ID JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID GROUP BY CUBE(r.COUNTRY, g.GENRE\_NAME) ORDER BY r.COUNTRY, g.GENRE\_NAME;



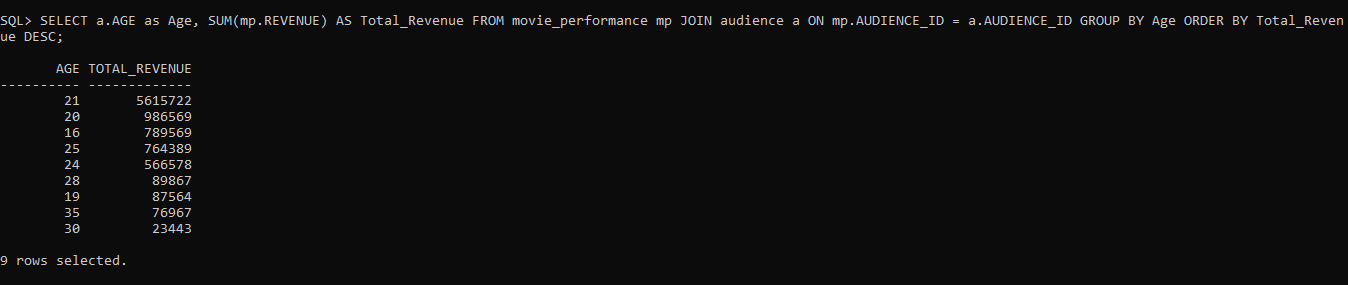
**Using GROUPING SETS (Custom Aggregation)**

SELECT g.GENRE\_NAME, r.COUNTRY, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID JOIN region r ON mp.REGION\_ID = r.REGION\_ID GROUP BY GROUPING SETS ((g.GENRE\_NAME, r.COUNTRY), (g.GENRE\_NAME), (r.COUNTRY), ()) ORDER BY g.GENRE\_NAME, r.COUNTRY;



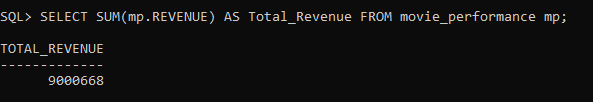
**Total Revenue by Age**

SELECT a.AGE as Age, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN audience a ON mp.AUDIENCE\_ID = a.AUDIENCE\_ID GROUP BY Age ORDER BY Total\_Revenue DESC;



**Total Revenue across all regions (Grand total)**

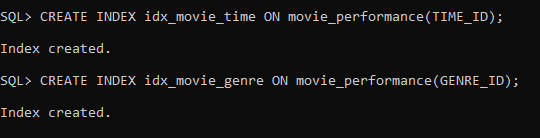
SELECT SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp;



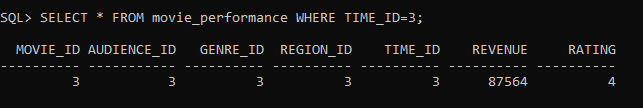
**Creating Indexes for Performance Optimization**

CREATE INDEX idx\_movie\_time ON movie\_performance(TIME\_ID);

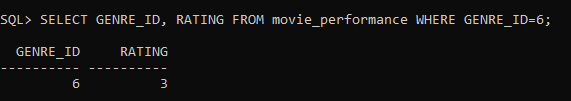
CREATE INDEX idx\_movie\_genre ON movie\_performance(GENRE\_ID);



SELECT \* FROM movie\_performance WHERE TIME\_ID=3;

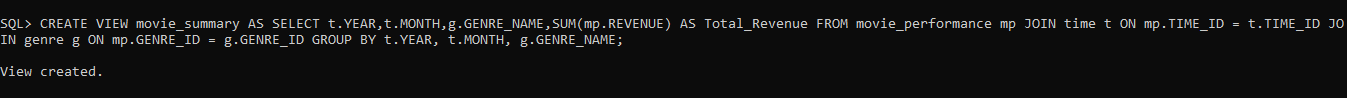


SELECT GENRE\_ID, RATING FROM movie\_performance WHERE GENRE\_ID=6;

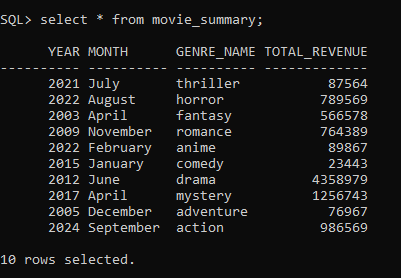


**Creating a View for Summary data**

CREATE VIEW movie\_summary AS SELECT t.YEAR,t.MONTH,g.GENRE\_NAME,SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN time t ON mp.TIME\_ID = t.TIME\_ID JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID GROUP BY t.YEAR, t.MONTH, g.GENRE\_NAME;



select \* from movie\_summary;



**Dynamic Queries with Parameters**

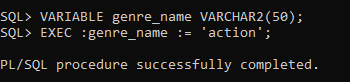
-- Declare and assign the variable

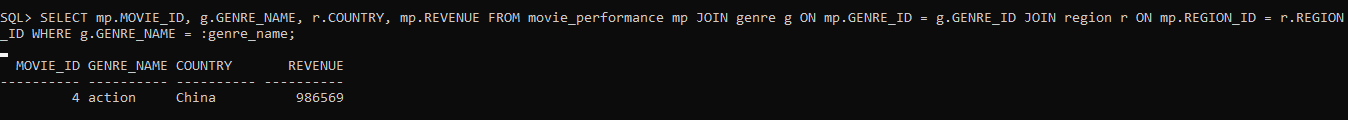
VARIABLE genre\_name VARCHAR2(50);

EXEC :genre\_name := 'action';

-- Use the bind variable in your query

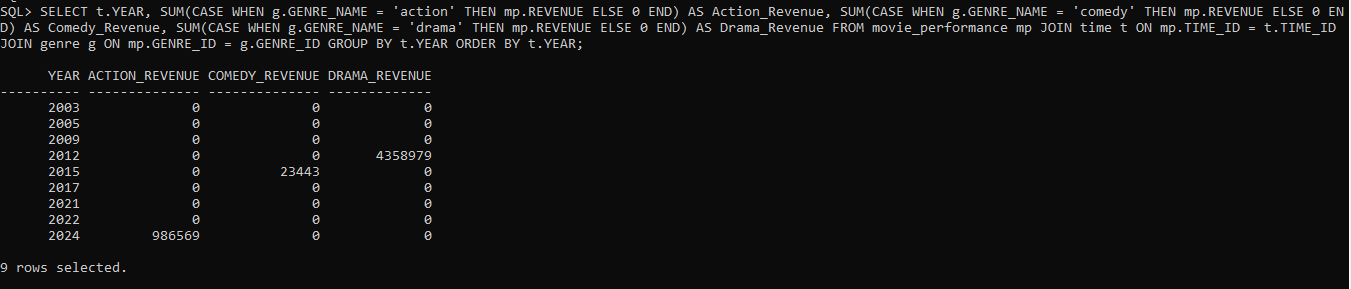
SELECT mp.MOVIE\_ID, g.GENRE\_NAME, r.COUNTRY, mp.REVENUE FROM movie\_performance mp JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID JOIN region r ON mp.REGION\_ID = r.REGION\_ID WHERE g.GENRE\_NAME = :genre\_name;





**Generate a Pivot Table Using CASE**

SELECT t.YEAR, SUM(CASE WHEN g.GENRE\_NAME = 'action' THEN mp.REVENUE ELSE 0 END) AS Action\_Revenue, SUM(CASE WHEN g.GENRE\_NAME = 'comedy' THEN mp.REVENUE ELSE 0 END) AS Comedy\_Revenue, SUM(CASE WHEN g.GENRE\_NAME = 'drama' THEN mp.REVENUE ELSE 0 END) AS Drama\_Revenue FROM movie\_performance mp JOIN time t ON mp.TIME\_ID = t.TIME\_ID JOIN genre g ON mp.GENRE\_ID = g.GENRE\_ID GROUP BY t.YEAR ORDER BY t.YEAR;



**Generate a Dynamic Monthly Revenue Report**

SELECT t.YEAR, t.MONTH, SUM(mp.REVENUE) AS Total\_Revenue FROM movie\_performance mp JOIN time t ON mp.TIME\_ID = t.TIME\_ID GROUP BY t.YEAR, t.MONTH ORDER BY t.YEAR, t.MONTH;

