**SQL Query Analysis Outputs**

**1./\*To create table\*/**

CREATE TABLE employee (

employee\_id SERIAL PRIMARY KEY,

last\_name VARCHAR(100),

first\_name VARCHAR(100),

title VARCHAR(100),

reports\_to INTEGER,

levels varchar(10),

birthdate DATE,

hiredate DATE,

address VARCHAR(200),

city VARCHAR(100),

state VARCHAR(100),

country VARCHAR(100),

postal\_code VARCHAR(20),

phone VARCHAR(50),

fax VARCHAR(50),

email VARCHAR(100),

FOREIGN KEY (reports\_to) REFERENCES employee(employee\_id)

);

**2./\*To import data from excel to sql\*/**

COPY employee

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\employee.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE customer (

customer\_id INTEGER PRIMARY KEY,

first\_name TEXT NOT NULL,

last\_name TEXT NOT NULL,

company TEXT,

address TEXT,

city TEXT,

state TEXT,

country TEXT,

postal\_code TEXT,

phone TEXT,

fax TEXT,

email TEXT,

support\_rep\_id INTEGER,

FOREIGN KEY (support\_rep\_id) REFERENCES employee(employee\_id)

);

COPY customer

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\customer.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE invoice (

invoice\_id INTEGER PRIMARY KEY,

customer\_id INTEGER NOT NULL,

invoice\_date DATE NOT NULL,

billing\_address TEXT,

billing\_city TEXT,

billing\_state TEXT,

billing\_country TEXT,

billing\_postal\_code TEXT,

total NUMERIC NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id)

);

COPY invoice

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\invoice.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE invoice\_line (

invoice\_line\_id INTEGER PRIMARY KEY,

invoice\_id INTEGER NOT NULL,

track\_id INTEGER NOT NULL,

unit\_price NUMERIC NOT NULL,

quantity INTEGER NOT NULL,

FOREIGN KEY (invoice\_id) REFERENCES invoice(invoice\_id),

FOREIGN KEY (track\_id) REFERENCES track(track\_id)

);

COPY invoice\_line

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\invoice\_line.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE artist (

artist\_id INTEGER PRIMARY KEY,

name TEXT

);

COPY artist

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\artist.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE genre (

genre\_id INTEGER PRIMARY KEY,

name TEXT

);

COPY genre

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\genre.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE playlist (

playlist\_id INTEGER PRIMARY KEY,

name TEXT

);

COPY playlist

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\playlist.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE album (

album\_id INTEGER PRIMARY KEY,

title TEXT NOT NULL,

artist\_id INTEGER NOT NULL,

FOREIGN KEY (artist\_id) REFERENCES artist(artist\_id)

);

COPY album

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\album.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE media\_type (

media\_type\_id INTEGER PRIMARY KEY,

name TEXT

);

COPY media\_type

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\media\_type.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE track (

track\_id INTEGER PRIMARY KEY,

name TEXT NOT NULL,

album\_id INTEGER,

media\_type\_id INTEGER NOT NULL,

genre\_id INTEGER,

composer TEXT,

milliseconds INTEGER NOT NULL,

bytes INTEGER,

unit\_price NUMERIC NOT NULL,

FOREIGN KEY (album\_id) REFERENCES album(album\_id),

FOREIGN KEY (media\_type\_id) REFERENCES media\_type(media\_type\_id),

FOREIGN KEY (genre\_id) REFERENCES genre(genre\_id)

);

COPY track

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\track.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE playlist\_track (

playlist\_id INTEGER NOT NULL,

track\_id INTEGER NOT NULL,

PRIMARY KEY (playlist\_id, track\_id),

FOREIGN KEY (playlist\_id) REFERENCES playlist(playlist\_id),

FOREIGN KEY (track\_id) REFERENCES track(track\_id)

);

COPY playlist\_track

FROM 'C:\Users\Sandhya\OneDrive\Desktop\Digital Muasic Store Data Analysis\music store data\music store data\playlist\_track.csv'

DELIMITER ','

CSV HEADER;

**3./\* to updating nulls and Check and replacing dublicates \*/**

SELECT \* FROM customer

WHERE customer\_id IS NULL OR first\_name IS NULL OR last\_name IS NULL OR company IS NULL OR

address IS NULL OR city IS NULL OR state IS NULL OR country IS NULL OR postal\_code IS NULL OR

phone IS NULL OR fax IS NULL OR email IS NULL OR support\_rep\_id IS NULL;

UPDATE customer SET company = 'NA' WHERE company IS NULL;

UPDATE customer SET address = 'Not Provided' WHERE address IS NULL;

UPDATE customer SET city = 'Unknown' WHERE city IS NULL;

UPDATE customer SET state = 'Unknown' WHERE state IS NULL;

UPDATE customer SET country = 'Unknown' WHERE country IS NULL;

UPDATE customer SET postal\_code = '000000' WHERE postal\_code IS NULL;

UPDATE customer SET phone = '000-000-0000' WHERE phone IS NULL;

UPDATE customer SET fax = '000-000-0000' WHERE fax IS NULL;

UPDATE customer SET email = 'noemail@example.com' WHERE email IS NULL;

UPDATE customer SET support\_rep\_id = 0 WHERE support\_rep\_id IS NULL;

/\*to find null records\*/

SELECT \* FROM track

WHERE composer IS NULL;

**4./\*to update null values\*/**

UPDATE track SET composer = 'Unknown Composer' WHERE composer IS NULL;

**5./\*to count no. of duplicate records\*/**

SELECT name, COUNT(\*)

FROM genre

GROUP BY name

HAVING COUNT(\*) > 1;

**6./\* to remove duplicates from column\*/**

DELETE FROM genre

WHERE ctid NOT IN (

SELECT MIN(ctid)

FROM genre

GROUP BY name

);

**7./\*Total Revenue by Country\*/**

SELECT

c.country,

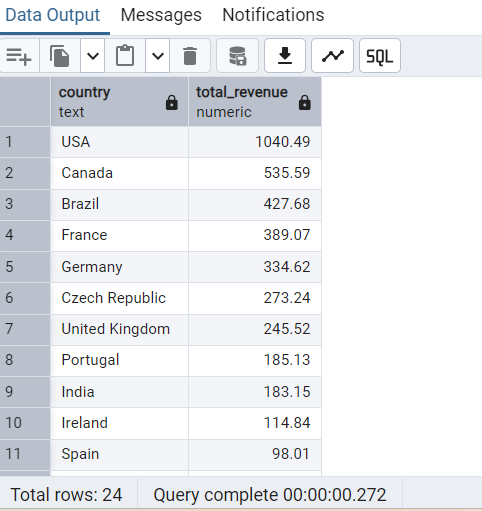
ROUND(SUM(i.total), 2) AS total\_revenue

FROM invoice i

JOIN customer c ON i.customer\_id = c.customer\_id

GROUP BY c.country

ORDER BY total\_revenue DESC;



**8./\*Customer Count by Country\*/**

SELECT

country,

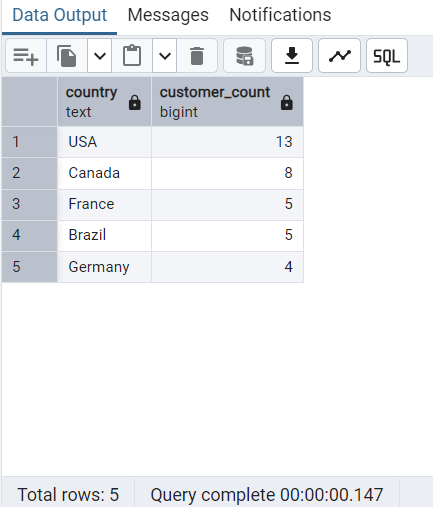
COUNT(customer\_id) AS customer\_count

FROM customer

GROUP BY country

ORDER BY customer\_count DESC

LIMIT 5;



**9./\* Monthly Invoice Count\*/**

SELECT

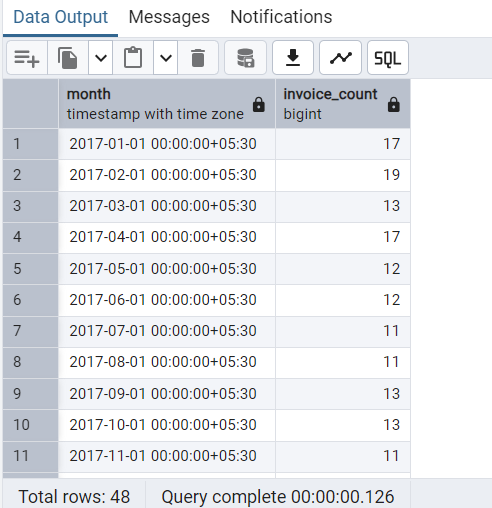
DATE\_TRUNC('month', invoice\_date) AS month,

COUNT(invoice\_id) AS invoice\_count

FROM invoice

GROUP BY month

ORDER BY month;



**10./\*Top 5 Customers by Total Spending\*/**

SELECT

c.first\_name || ' ' || c.last\_name AS customer\_name,

ROUND(SUM(i.total), 2) AS total\_spent

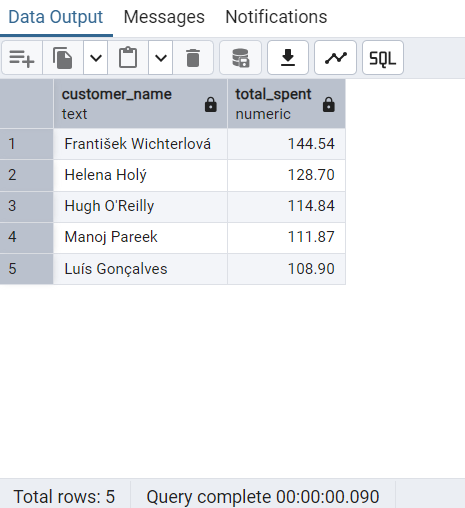
FROM invoice i

JOIN customer c ON i.customer\_id = c.customer\_id

GROUP BY customer\_name

ORDER BY total\_spent DESC

LIMIT 5;



**11./\*Top 10 Selling Tracks\*/**

SELECT

t.name AS track\_name,

SUM(il.quantity) AS units\_sold

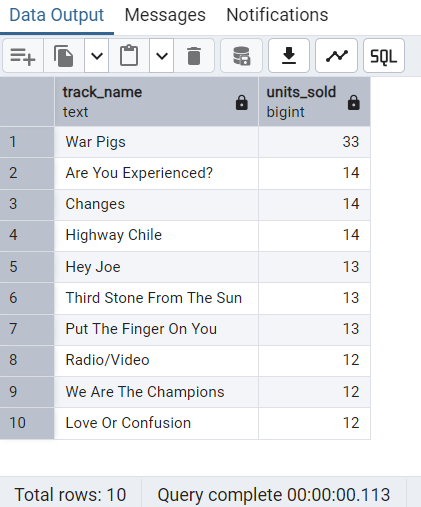
FROM invoice\_line il

JOIN track t ON il.track\_id = t.track\_id

GROUP BY t.name

ORDER BY units\_sold DESC

LIMIT 10;



**12./\*Revenue by Genre\*/**

SELECT

g.name AS genre,

ROUND(SUM(il.unit\_price \* il.quantity), 2) AS revenue

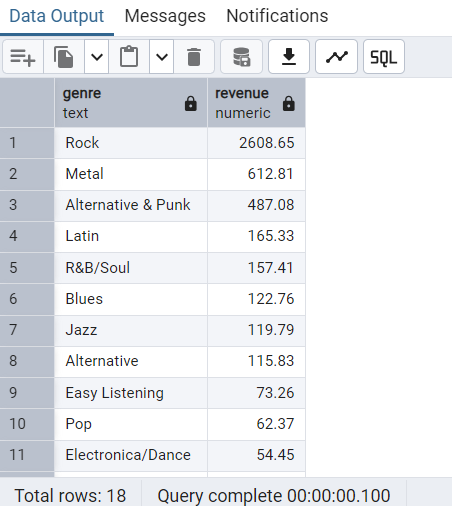
FROM invoice\_line il

JOIN track t ON il.track\_id = t.track\_id

JOIN genre g ON t.genre\_id = g.genre\_id

GROUP BY g.name

ORDER BY revenue DESC;



**13./\*Most Popular Artists by Sales Volume\*/**

SELECT

ar.name AS artist,

SUM(il.quantity) AS total\_tracks\_sold

FROM invoice\_line il

JOIN track t ON il.track\_id = t.track\_id

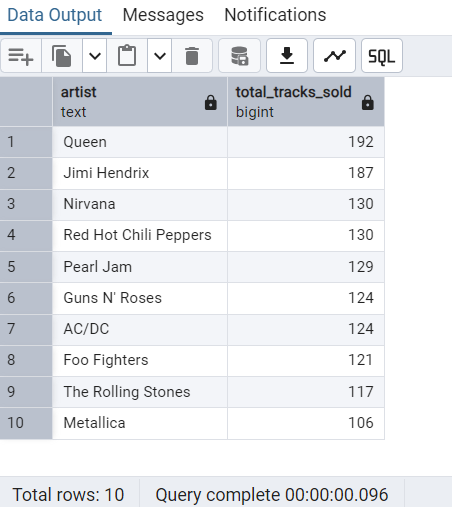
JOIN album a ON t.album\_id = a.album\_id

JOIN artist ar ON a.artist\_id = ar.artist\_id

GROUP BY ar.name

ORDER BY total\_tracks\_sold DESC

LIMIT 10;



**14./\*Most Used Media Types\*/**

SELECT

m.name AS media\_type,

COUNT(il.invoice\_line\_id) AS total\_sold

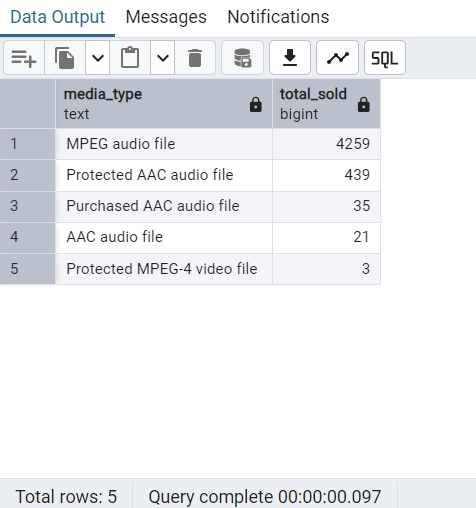
FROM invoice\_line il

JOIN track t ON il.track\_id = t.track\_id

JOIN media\_type m ON t.media\_type\_id = m.media\_type\_id

GROUP BY media\_type

ORDER BY total\_sold DESC;



**15./\*Monthly Revenue Trend\*/**

SELECT

DATE\_TRUNC('month', invoice\_date) AS month,

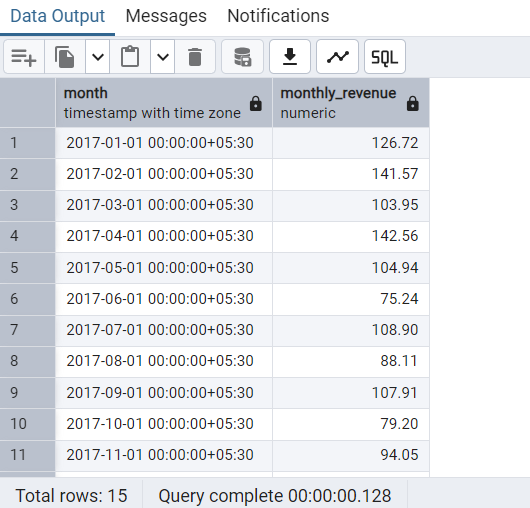
ROUND(SUM(total), 2) AS monthly\_revenue

FROM invoice

GROUP BY month

ORDER BY month

LIMIT 15;



**16./\*Top Albums by Revenue\*/**

SELECT

a.title AS album\_title,

ROUND(SUM(il.unit\_price \* il.quantity), 2) AS revenue

FROM invoice\_line il

JOIN track t ON il.track\_id = t.track\_id

JOIN album a ON t.album\_id = a.album\_id

GROUP BY a.title

ORDER BY revenue DESC

LIMIT 10;

