Query History

```sql
create database proect;
--create artist table
create table artists(
artist_id INT PRIMARY KEY,
name VARCHAR(50),
country VARCHAR(50)
);

-- Create Albums table
CREATE TABLE Albums (
    album_id INT PRIMARY KEY,
    artist_id INT,
    album_name VARCHAR(100),
    release_year INT,
    FOREIGN KEY (artist_id) REFERENCES Artists(artist_id)
);

-- Create Genres table
CREATE TABLE Genres (
    genre_id INT PRIMARY KEY,
    genre_name VARCHAR(50)
);

-- Create Tracks table
CREATE TABLE Tracks (
    track_id INT PRIMARY KEY,
```

Query History

```
    track_id INT PRIMARY KEY,
    album_id INT,
    track_name VARCHAR(100),
    duration INT, -- duration in seconds
    genre_id INT,
    FOREIGN KEY (album_id) REFERENCES Albums(album_id),
    FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)
);
-- Create Sales table
CREATE TABLE Sales (
    sale_id INT PRIMARY KEY,
    track_id INT,
    sales_amount INT, -- number of copies sold
    sale_date DATE,
    FOREIGN KEY (track_id) REFERENCES Tracks(track_id)
);

-- Insert data into Artists table
INSERT INTO Artists (artist_id, name, country)
VALUES
(1, 'The Beatles', 'UK'),
(2, 'Taylor Swift', 'USA'),
(3, 'Ed Sheeran', 'UK');

-- Insert data into Genres table
INSERT INTO Genres (genre_id, genre_name)
VALUES
```

Query | Query History

```
47    (2, 'Taylor Swift', 'USA'),
48    (3, 'Ed Sheeran', 'UK');
49
50    -- Insert data into Genres table
51    INSERT INTO Genres (genre_id, genre_name)
52    VALUES
53    (1, 'Rock'),
54    (2, 'Pop'),
55    (3, 'Folk');
56
57    -- Insert data into Albums table
58    INSERT INTO Albums (album_id, artist_id, album_name, release_year)
59    VALUES
60    (1, 1, 'Abbey Road', 1969),
61    (2, 2, '1989', 2014),
62    (3, 3, 'Divide', 2017);
63
64    -- Insert data into Tracks table
65    INSERT INTO Tracks (track_id, album_id, track_name, duration, genre_id)
66    VALUES
67    (1, 1, 'Come Together', 259, 1),
68    (2, 1, 'Something', 182, 1),
69    (3, 2, 'Shake It Off', 242, 2),
70    (4, 2, 'Blank Space', 231, 2),
71    (5, 3, 'Shape of You', 233, 3);
72
```

Query History

```sql
-- Insert data into Sales table
INSERT INTO Sales (sale_id, track_id, sales_amount, sale_date)
VALUES
(1, 1, 1000000, '2019-05-01'),
(2, 2, 800000, '2020-01-01'),
(3, 3, 2000000, '2021-03-01'),
(4, 4, 1500000, '2021-06-01'),
(5, 5, 3000000, '2022-07-01');

select * from  artists;
select * from Albums;
select * from Tracks;
select * from Genres;
select * from Sales;

--get the total number of albums released by each artist
select A.name as artist_name , count(*) AS total_albumns
from artists A
join Albums AL on A.artist_id = Al.artist_id
Group by A.name;

--find the average sales for each track
SELECT T.track_name ,AVG(S.sales_amount) AS average_sales
FROM Tracks T
JOIN Sales S ON T.track_id = S.track_id
GROUP BY T.track_name;
```

Query | Query History

```sql
--get the total number of albums released by each artist
select A.name as artist_name , count(*) AS total_albumns
from artists A
join Albums AL on A.artist_id = Al.artist_id
Group by A.name;

--find the average sales for each track
SELECT T.track_name ,AVG(S.sales_amount) AS average_sales
FROM Tracks T
JOIN Sales S ON T.track_id = S.track_id
GROUP BY T.track_name;

--List all tracks from a specific genere
select T.track_name , A.album_name , G.genre_name
from tracks T
join Albums A on T.album_id = A.album_id
join Genres G on T.genre_id = G.genre_id
where G.genre_name ='Pop';

--find the total sales for each album;
SELECT A.album_name ,SUM(S.sales_amount) AS total_sales
FROM Albums A
JOIN Tracks T on A.album_id = T.album_id
join Sales S ON T.track_id = S.track_id
GROUP BY A.album_name;

```

Query History

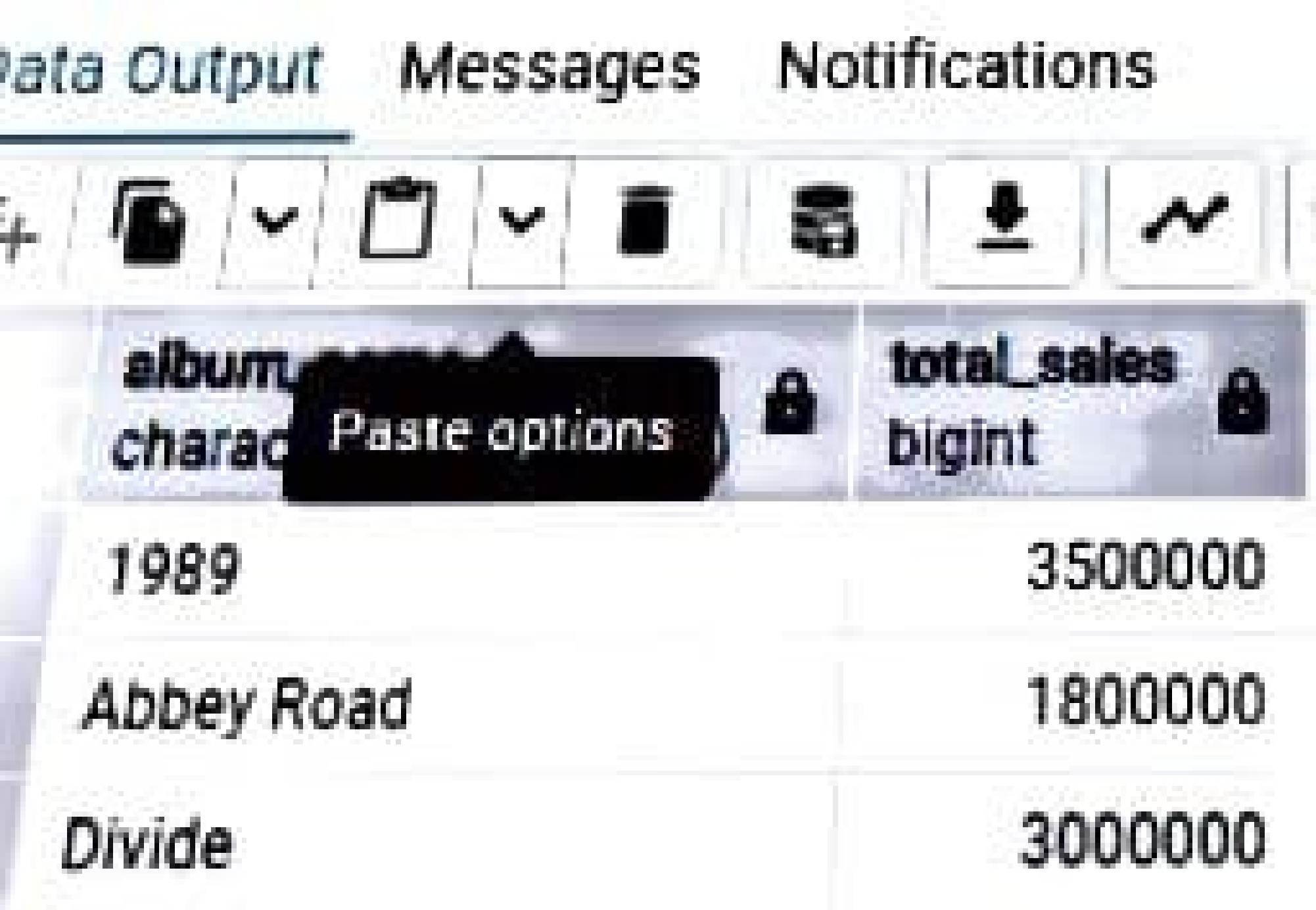
```
join Albums A on T.album_id = A.album_id
join Genres G on T.genre_id = G.genre_id
where G.genre_name ='Pop';

--find the total sales for each album;
SELECT A.album_name ,SUM(S.sales_amount) AS total_sales
FROM Albums A
JOIN Tracks T on A.album_id = T.album_id
join Sales S ON T.track_id = S.track_id
GROUP BY A.album_name;

--fibd the artist with highest total sales
SELECT A.name AS artist_name, SUM(S.sales_amount) AS tot
FROM Artists A
JOIN albums AL ON A.artist_id = AL.artist_id
JOIN Tracks T ON AL.album_id = T.album_id
JOIN Sales S ON T.track_id = S.track_id
GROUP BY A.name
ORDER BY total_sales DESC
LIMIT 1;
```

Data Output Messages Notifications

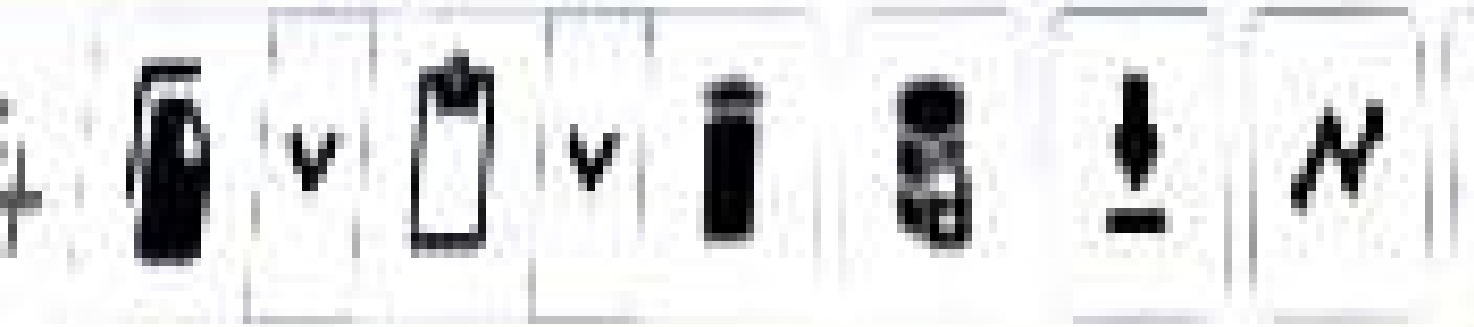
	artist_name character varying (50)	total_albumns bigint
1	The Beatles	1
2	Taylor Swift	1
3	Ed Sheeran	1

track_name character varying (100)	average_sales numeric
Blank Space	1500000.00000000000
Come Together	1000000.00000000000
Shake It Off	2000000.00000000000
Shape of You	3000000.00000000000
Something	800000.00000000000

track_name character varying (100)	album_name character varying (100)	genre_name character varying (50)
Shake It Off	1989	Pop
Blank Space	1989	Pop

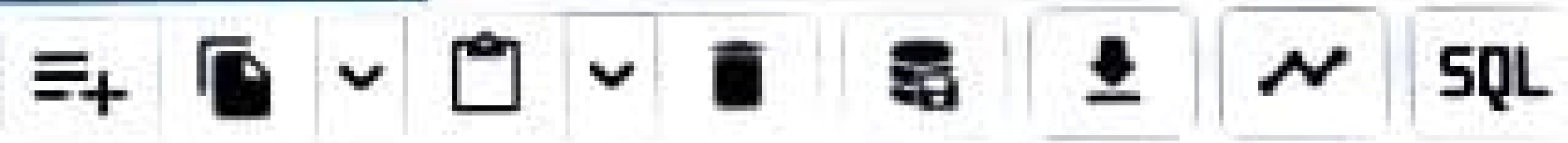

ata Output
Messages
Notifications
Paste options
totaL_sales
bigint
1989
3500000
Abbey Road
1800000
Divide
3000000

Data Output Messages Notifications



artist_name character varying (50)	total_sales bigint
Taylor Swift	3500000

Data Output Messages Notifications



	track_name character varying (100)	duration integer
1	Come Together	259