

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

1. Exercise 1:

Identify the top 10 customers and their email so we can reward them

SQL Query:

```
SELECT concat (first_name,' ',last_name) as full_name, email, sum(amount) as  
total_amount  
FROM customer  
JOIN payment on payment.customer_id = customer.customer_id  
GROUP BY 1,2  
ORDER BY 3 DESC  
LIMIT 10;
```

Query Result:

full_name	email	total_amount
Eleanor Hunt	eleanor.hunt@sakilacustomer.org	211,55
Karl Seal	karl.seal@sakilacustomer.org	208,58
Marion Snyder	marion.snyder@sakilacustomer.org	194,61
Rhonda Kennedy	rhonda.kennedy@sakilacustomer.org	191,62
Clara Shaw	clara.shaw@sakilacustomer.org	189,6
Tommy Collazo	tommy.collazo@sakilacustomer.org	183,63
Ana Bradley	ana.bradley@sakilacustomer.org	167,67
Curtis Irby	curtis.irby@sakilacustomer.org	167,62
Marcia Dean	marcia.dean@sakilacustomer.org	166,61
Mike Way	mike.way@sakilacustomer.org	162,67

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Screenshot

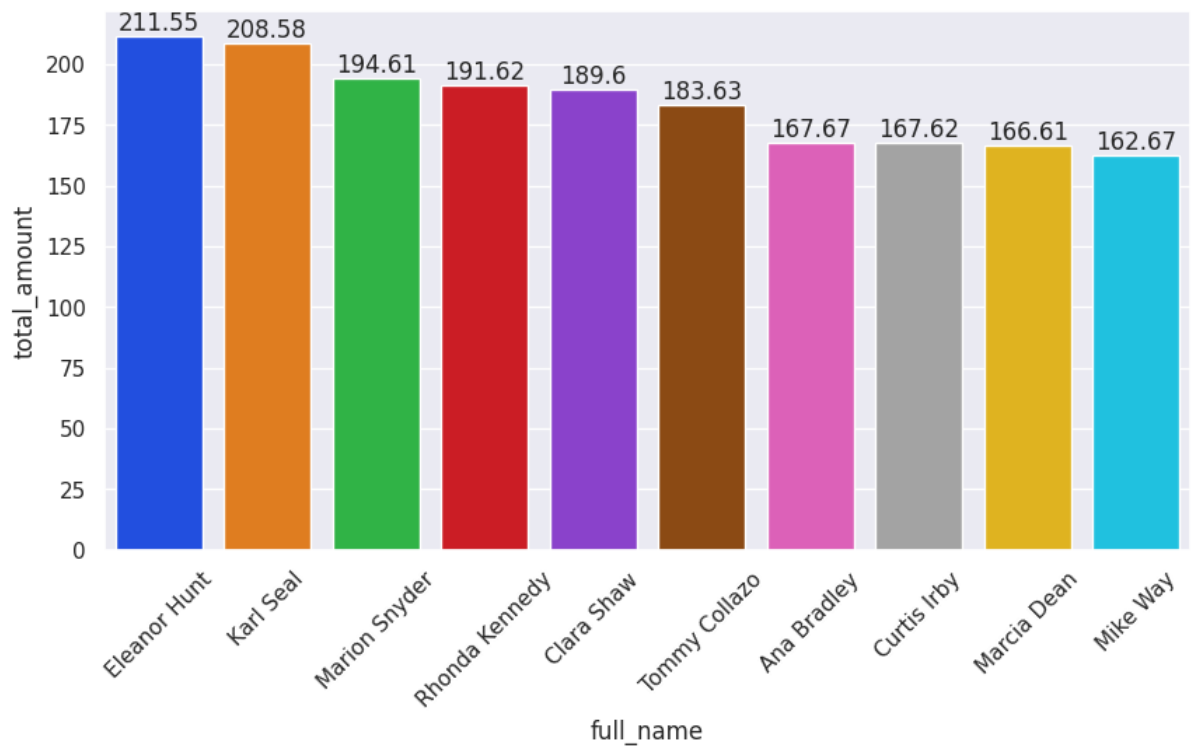
The screenshot shows the pgAdmin 4 interface. The left pane displays the database structure for 'PostgreSQL 15', including 'Lita Cantik' and 'dvdrental' databases. The central pane shows a SQL query:
1 SELECT concat (first_name,' ',last_name) as full_name, email, sum(amount) as
2 FROM customer
3 JOIN payment on payment.customer_id = customer.customer_id
4 GROUP BY 1,2
5 ORDER BY 3 DESC
6 LIMIT 10;
The bottom pane displays the query results in a table with 10 rows. The columns are full_name, email, and total_amount. The data is sorted by total_amount in descending order.

	full_name	email	total_amount
1	Eleanor Hunt	eleanor.hunt@sakilacustomer.org	211.55
2	Karl Seal	karl.seal@sakilacustomer.org	208.58
3	Marion Snyder	marion.snyder@sakilacustomer.org	194.61
4	Rhonda Kennedy	rhonda.kennedy@sakilacustomer.org	191.62
5	Clara Shaw	clara.shaw@sakilacustomer.org	189.60
6	Tommy Collazo	tommy.collazo@sakilacustomer.org	183.63
7	Ana Bradley	ana.bradley@sakilacustomer.org	167.67
8	Curtis Irby	curtis.irby@sakilacustomer.org	167.62
9	Marcia Dean	marcia.dean@sakilacustomer.org	166.61
10	Mike Way	mike.way@sakilacustomer.org	162.67

Total rows: 10 of 10 Query complete 00:00:00.420 Ln 6, Col 10

Visualization:

Top 10 DVD Customers



Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

2. Exercise 2:

Identify the bottom 10 customers and their emails

SQL Query:

```
SELECT concat (first_name, ' ',last_name) as full_name, email, sum(amount) as  
total_amount  
FROM customer  
JOIN payment on payment.customer_id = customer.customer_id  
GROUP BY 1,2  
ORDER BY 3 ASC  
LIMIT 10;
```

Query Result:

full_name	email	total_amount
Brian Wyman	brian.wyman@sakilacustomer.org	27,93
Leona Obrien	leona.obrien@sakilacustomer.org	32,9
Caroline Bowman	caroline.bowman@sakilacustomer.org	37,87
Anthony Schwab	anthony.schwab@sakilacustomer.org	47,85
Tiffany Jordan	tiffany.jordan@sakilacustomer.org	49,88
Kirk Stclair	kirk.stclair@sakilacustomer.org	50,83
Bobbie Craig	bobbie.craig@sakilacustomer.org	52,81
Jo Fowler	jo.fowler@sakilacustomer.org	54,85
Penny Neal	penny.neal@sakilacustomer.org	56,84
Johnny Turpin	johnny.turpin@sakilacustomer.org	57,81

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Screenshot:

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure for 'PostgreSQL 15', including 'Lita Cantik' and 'dvdrental' databases. The 'dvdrental' database is expanded, showing 'public' schema with 'Aggregates', 'Collations', 'Domains', and 'FTS Configurations'. The main pane shows a SQL query in the 'Query' tab:

```
1 SELECT concat (first_name,' ',last_name) as full_name, email, sum(amount) as  
2 FROM customer  
3 JOIN payment on payment.customer_id = customer.customer_id  
4 GROUP BY 1,2  
5 ORDER BY 3 ASC  
6 LIMIT 10;
```

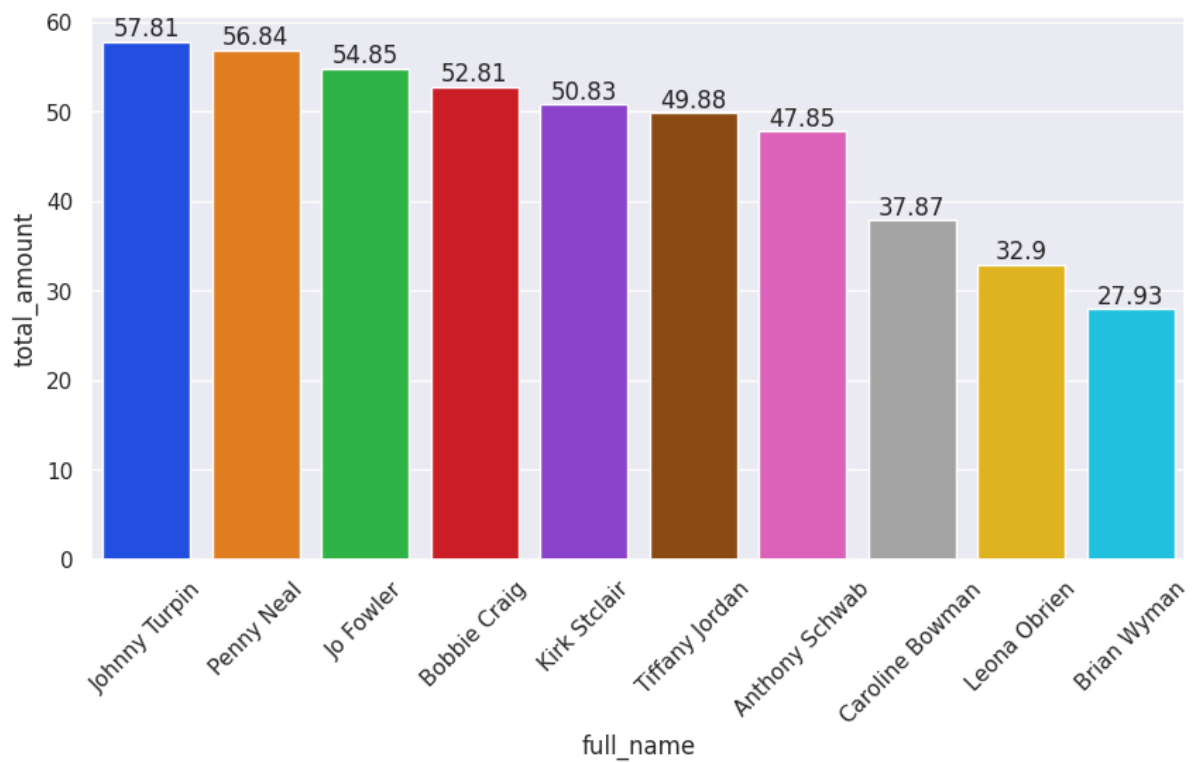
The 'Data Output' tab shows the results of the query:

	full_name text	email character varying	total_amount numeric
1	Brian Wyman	brian.wyman@sakilacustomer.org	27.93
2	Leona Obrien	leona.obrien@sakilacustomer.org	32.90
3	Caroline Bowman	caroline.bowman@sakilacustomer.org	37.87
4	Anthony Schwab	anthony.schwab@sakilacustomer.org	47.85
5	Tiffany Jordan	tiffany.jordan@sakilacustomer.org	49.88
6	Kirk Stclair	kirk.stclair@sakilacustomer.org	50.83
7	Bobbie Craig	bobbie.craig@sakilacustomer.org	52.81
8	Jo Fowler	jo.fowler@sakilacustomer.org	54.85
9	Penny Neal	penny.neal@sakilacustomer.org	56.84
10	Johnny Turpin	johnny.turpin@sakilacustomer.org	57.81

The status bar at the bottom indicates 'Total rows: 10 of 10' and 'Query complete 00:00:00.860'.

Visualization:

Bottom 10 DVD Customers



Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

3. Exercise 3:

What are the most profitable movie genres (ratings)?

SQL Query:

```
SELECT c.name AS genre, SUM(p.amount) AS total_sales
FROM payment p
JOIN rental r ON p.rental_id = r.rental_id
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id
JOIN film_category fc ON f.film_id = fc.film_id
JOIN category c ON fc.category_id = c.category_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3
```

Query Result:

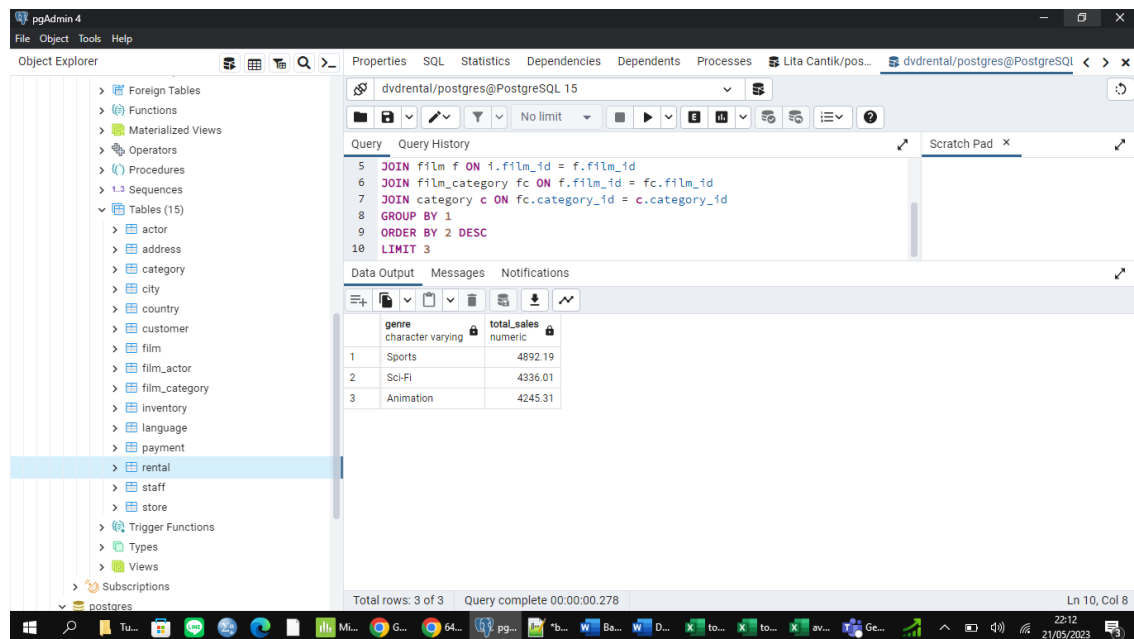
genre	total_sales
Sports	4892,19
Sci-Fi	4336,01
Animation	4245,31
Drama	4118,46
Comedy	4002,48
New	3966,38
Action	3951,84
Foreign	3934,47
Games	3922,18
Family	3830,15
Documentary	3749,65
Horror	3401,27
Classics	3353,38
Children	3309,39
Travel	3227,36
Music	3071,52

Nama : Nurlita Kholishotunnisa

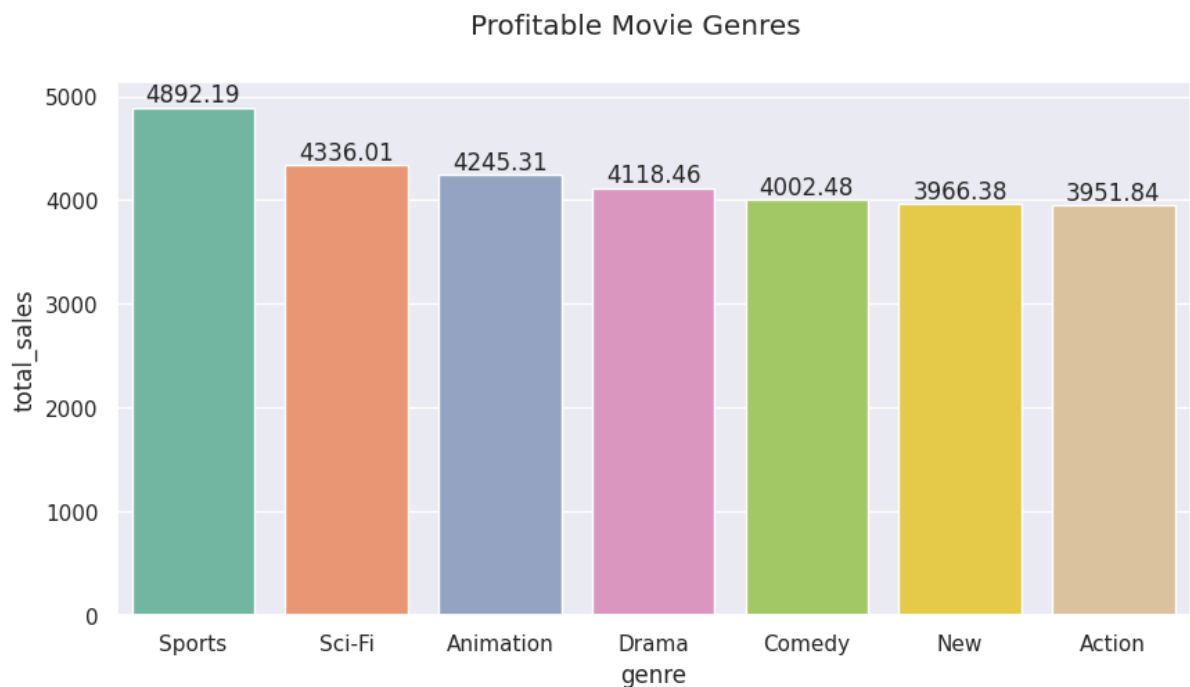
Spesialisasi : Data Analyst

Id Live Class : 041

Screenshot:



Visualization:



Based on the output above, it can be concluded that the most profitable movie genre is Sports with total sales of 4892,19.

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

4. Exercise 4:

How many rented movies were returned late, early, and on time?

SQL Query:

```
SELECT CASE
  WHEN rental_duration > date_part('day', return_date - rental_date)
  THEN 'Returned Early'
  WHEN rental_duration = date_part('day', return_date - rental_date)
  THEN 'On Time'
  WHEN rental_duration < date_part('day', return_date - rental_date)
  THEN 'Returned Late'
  ELSE 'Not Returned'
END AS status,
COUNT(*) AS jumlah_film
FROM film
INNER JOIN inventory USING(film_id)
INNER JOIN rental USING(inventory_id)
GROUP BY status
ORDER BY jumlah_film
```

Query Result:

status	jumlah_film
Not Returned	183
On Time	1720
Returned Late	6403
Returned Early	7738

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

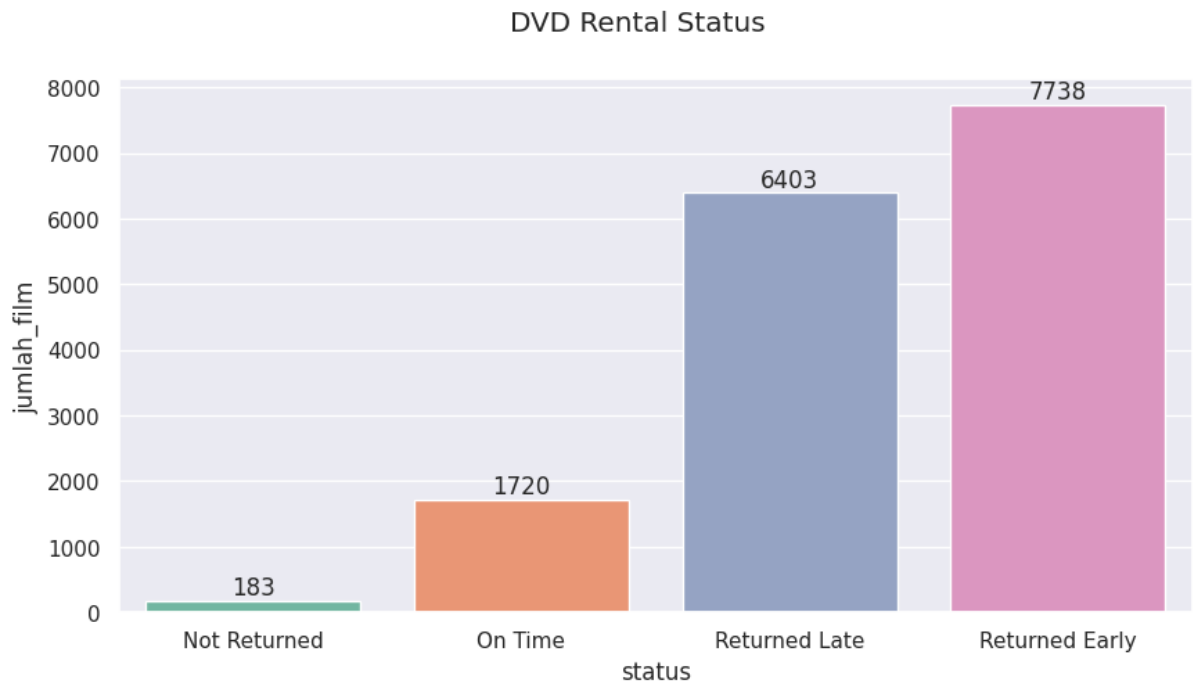
Id Live Class : 041

Screenshot:

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays a database schema with tables like rental, staff, and store. The central pane shows a SQL query that categorizes rental durations into four status groups: 'Returned Early', 'On Time', 'Returned Late', and 'Not Returned'. The query uses a CASE statement to assign these statuses based on the difference between return_date and rental_date. The results pane at the bottom shows a table with two columns: 'status' and 'jumlah_film' (count of films). The data is as follows:

status	jumlah_film
Not Returned	183
On Time	1720
Returned Late	6403
Returned Early	7738

Visualization:



Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

5. Exercise 5:

What is the customer base in the countries where we have a presence?

SQL Query:

```
SELECT country, COUNT(customer_id) AS total_customers
FROM country
JOIN city ON country.country_id = city.country_id
JOIN address ON city.city_id = address.city_id
JOIN customer ON address.address_id = customer.address_id
GROUP BY 1
ORDER BY 2 DESC
```

Query Result:

country	total_customers
India	60
China	53
United States	36
Japan	31
Mexico	30
Brazil	28
Russian Federation	28
Philippines	20
Turkey	15
Indonesia	14
Argentina	13
Nigeria	13
South Africa	11
Taiwan	10
United Kingdom	9
Poland	8
Iran	8
Germany	7

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Italy	7
Venezuela	7
Egypt	6
Colombia	6
Ukraine	6
Vietnam	6
Spain	5
Canada	5
South Korea	5
Pakistan	5
Netherlands	5
Saudi Arabia	5
Israel	4
France	4
Yemen	4
Peru	4
Dominican Republic	3
Bangladesh	3
Thailand	3
Algeria	3
Chile	3
Malaysia	3
Mozambique	3
Switzerland	3
Ecuador	3
Paraguay	3
Austria	3
United Arab Emirates	3
Morocco	3
Tanzania	3
Myanmar	2
Bulgaria	2

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Cameroon	2
Cambodia	2
Congo, The Democratic Republic of the	2
Sudan	2
Romania	2
Latvia	2
Yugoslavia	2
Puerto Rico	2
Kazakstan	2
Greece	2
Bolivia	2
Kenya	2
French Polynesia	2
Angola	2
Belarus	2
Oman	2
Azerbaijan	2
Hungary	1
American Samoa	1
Armenia	1
Sri Lanka	1
French Guiana	1
Sweden	1
Faroe Islands	1
Ethiopia	1
Bahrain	1
Czech Republic	1
Lithuania	1
Turkmenistan	1
Tunisia	1
Virgin Islands, U.S.	1

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Malawi	1
Chad	1
Afghanistan	1
Greenland	1
Moldova	1
Gambia	1
Holy See (Vatican City State)	1
Estonia	1
Slovakia	1
North Korea	1
Nauru	1
Liechtenstein	1
Senegal	1
Zambia	1
Hong Kong	1
Kuwait	1
Madagascar	1
Runion	1
Saint Vincent and the Grenadines	1
Tuvalu	1
Finland	1
Iraq	1
Anguilla	1
Brunei	1
Tonga	1
Nepal	1
New Zealand	1

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Screenshot:

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays a database schema with tables like actor, address, city, customer, film, inventory, language, payment, rental, staff, store, and subscription. The 'rental' table is selected. The main pane shows a SQL query in the Query Editor:

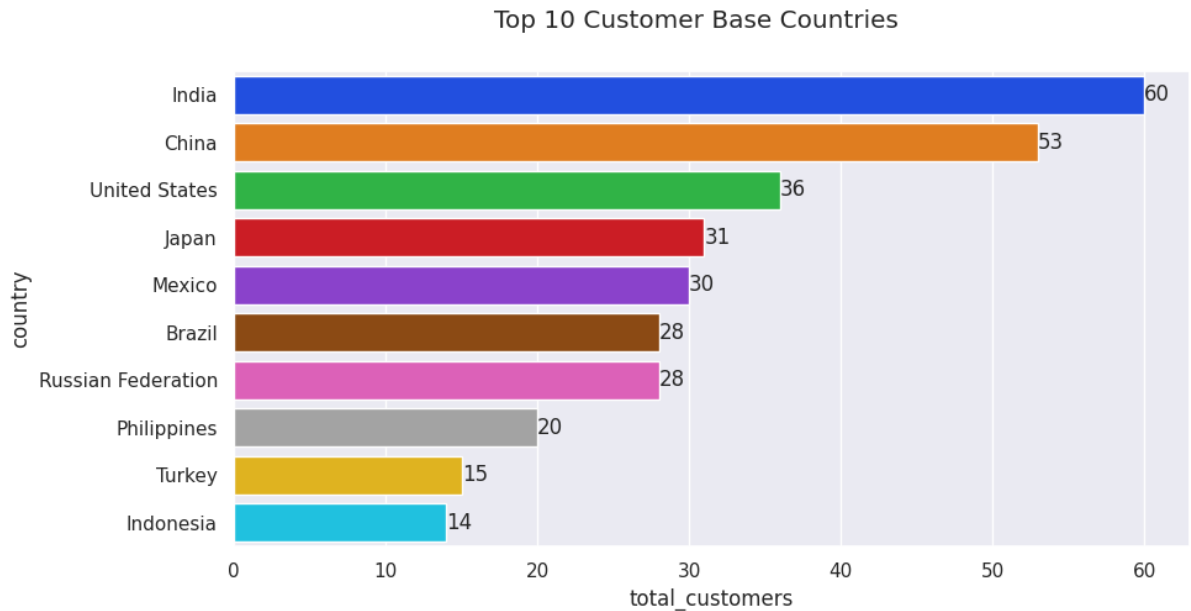
```
1 SELECT country, COUNT(customer_id) AS total_customers
2 FROM country
3 JOIN city ON country.country_id = city.country_id
4 JOIN address ON city.city_id = address.city_id
5 JOIN customer ON address.address_id = customer.address_id
6 GROUP BY 1
7 ORDER BY 2 DESC
```

Below the query editor, the Data Output pane shows the results of the query as a table:

country	total_customers
India	60
China	53
United States	36
Japan	31
Mexico	30
Brazil	28
Russian Federation	28
Philippines	20
Turkey	15
Indonesia	14

The status bar at the bottom indicates 'Total rows: 108 of 108' and 'Query complete 00:00:00.902'.

Visualization:



Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

6. Exercise 6:

Which country is the most profitable for the business?

SQL Query:

```
SELECT country, COUNT(*) AS total_customers, SUM(amount) AS total_sales
FROM country
JOIN city ON country.country_id=city.country_id
JOIN address ON city.city_id=address.city_id
JOIN customer ON address.address_id=customer.address_id
JOIN payment ON customer.customer_id=payment.customer_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10
```

Query Result:

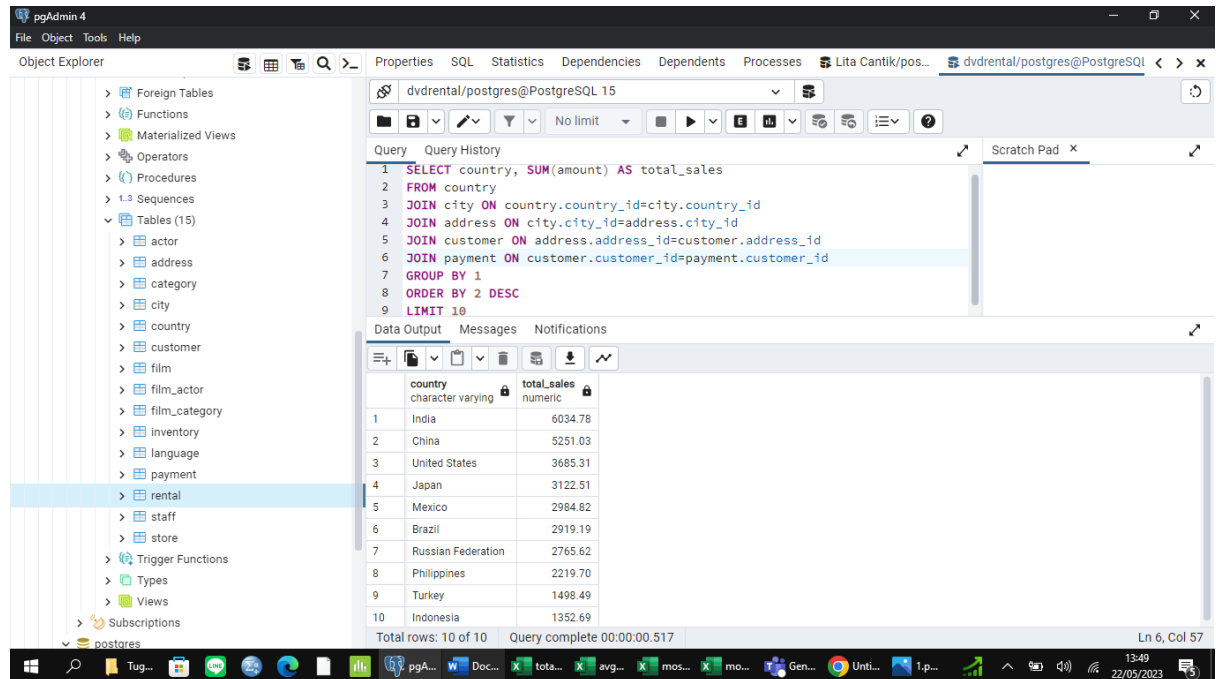
country	total_sales
India	6034,78
China	5251,03
United States	3685,31
Japan	3122,51
Mexico	2984,82
Brazil	2919,19
Russian Federation	2765,62
Philippines	2219,7
Turkey	1498,49
Indonesia	1352,69

Nama : Nurlita Kholishotunnisa

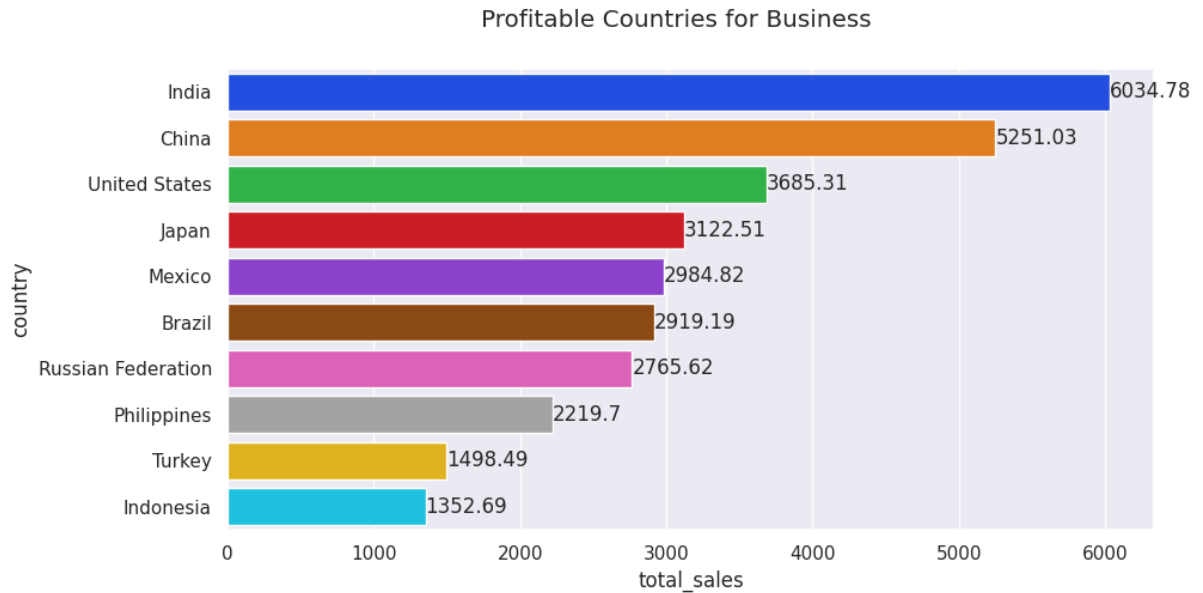
Spesialisasi : Data Analyst

Id Live Class : 041

Screenshot:



Visualization:



Based on the output above, it can be concluded that the most profitable country for business is India because India get total sales of 6034,78.

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

7. Exercise 7:

What is the average rental rate per movie genre (rating)?

SQL Query:

```
SELECT NAME AS movie_genre, AVG(rental_rate) AS average_rental_rate
FROM category
JOIN film_category USING (category_id)
JOIN film USING (film_id)
GROUP BY 1
ORDER BY 2 DESC
```

Query Result:

movie_genre	average_rental_rate
Games	3,252295
Travel	3,235614
Sci-Fi	3,219508
Comedy	3,162414
Sports	3,125135
New	3,116984
Foreign	3,099589
Horror	3,025714
Drama	3,022258
Music	2,950784
Children	2,89
Animation	2,808182
Family	2,758116
Classics	2,744386
Documentary	2,666471
Action	2,64625

Nama : Nurlita Kholishotunnisa

Spesialisasi : Data Analyst

Id Live Class : 041

Screenshot:

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays a database schema with tables like 'actor', 'address', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'inventory', 'language', 'payment', 'rental', 'staff', 'store', 'trigger_functions', 'types', 'views', and 'subscriptions'. The 'rental' table is selected. The main pane shows a SQL query in the 'Query' tab:

```
1 SELECT NAME AS movie_genre, AVG(rental_rate) AS average_rental_rate
2 FROM category
3 JOIN film_category USING (category_id)
4 JOIN film USING (film_id)
5 GROUP BY 1
6 ORDER BY 2 DESC
```

The 'Data Output' tab shows the results of the query as a table with two columns: 'movie_genre' and 'average_rental_rate'. The results are sorted by 'average_rental_rate' in descending order. The bottom status bar indicates 'Total rows: 16 of 16' and 'Query complete 00:00:00.608'.

movie_genre	average_rental_rate
Games	3.252950819672131
Travel	3.2356140350877193
Sci-Fi	3.2195081967213115
Comedy	3.1624137931034483
Sports	3.1251351351351351
New	3.1169841269841270
Foreign	3.095890410958904
Horror	3.0257142857142857
Drama	3.0222580645161290
Music	2.9507843137254902
Children	2.8900000000000000
Animation	2.8081818181818182

Visualization:

Average Rental Rate per Movie Genres

