

SQL

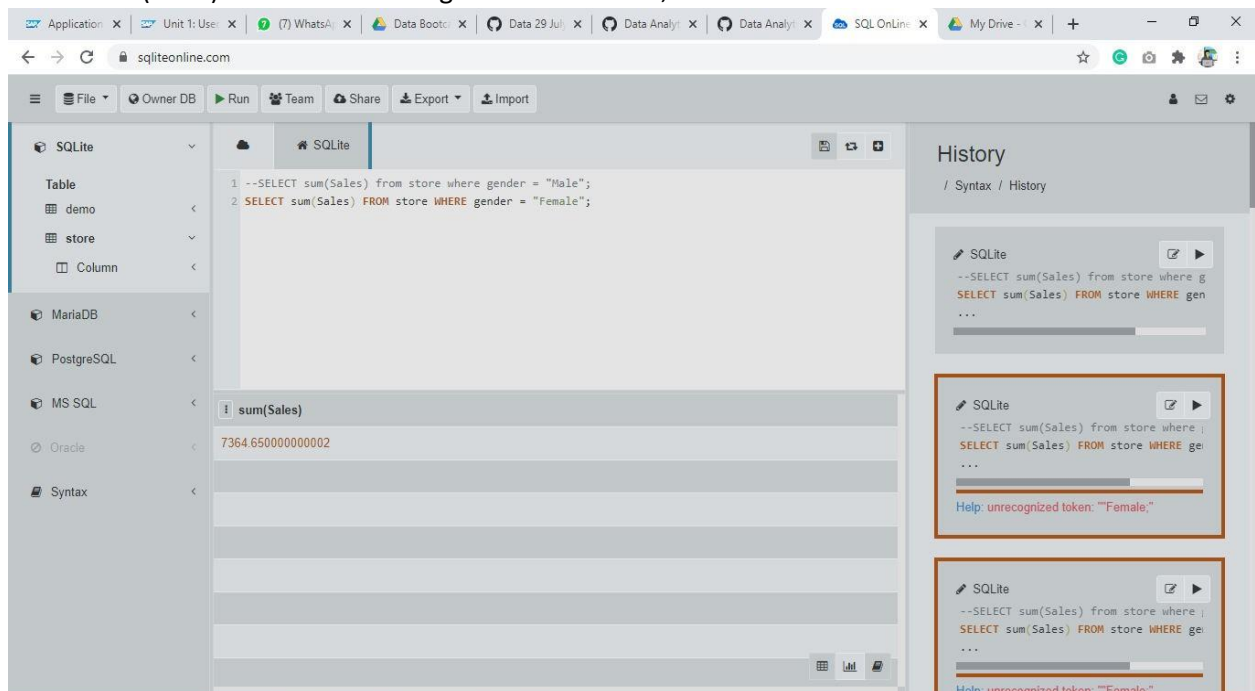
Paruvada sadwini Kumari

Database - Pelican Store

Q1. Which gender gives the maximum sales?

SELECT sum(Sales) from store where gender = "Male"; -> 384

SELECT sum(Sales) from store where gender = "Female"; -> 7854



Q2. Which age group and gender gives the maximum sales?

- Females and 30-50 age group gives the maximum sales.

Select Age, Gender, Sales from store order by cast (Sales as float) desc limit 5;

The screenshot shows the SQLiteOnline web application. The left sidebar lists databases: SQLite, MariaDB, PostgreSQL, MS SQL, Oracle, and Syntax. The main editor contains a SQL query:

```
1 --SELECT sum(Sales) from store where gender = "Male";
2 --SELECT sum(Sales) from store where gender = "Female";
3 --SELECT Sales from store where cast (Sales as float) ;
4 --select * from store;
5 SELECT Age, Gender, Sales FROM store ORDER BY CAST (Sales AS FLOAT) DESC LIMIT 5;
6
```

The results are displayed in a table with columns: Age, Gender, and Sales.

Age	Gender	Sales
52	Female	287.59
50	Female	266
30	Female	253
30	Female	229.5
42	Female	198.8

The right sidebar shows a History panel with three entries, each containing a snippet of the SQL query.

Q3. What is the highest discount given to females ? select * from store where gender = "Female" order by discount desc;

The screenshot shows the SQLiteOnline web application. The left sidebar lists databases: SQLite, MariaDB, PostgreSQL, MS SQL, Oracle, and Syntax. The main editor contains a SQL query:

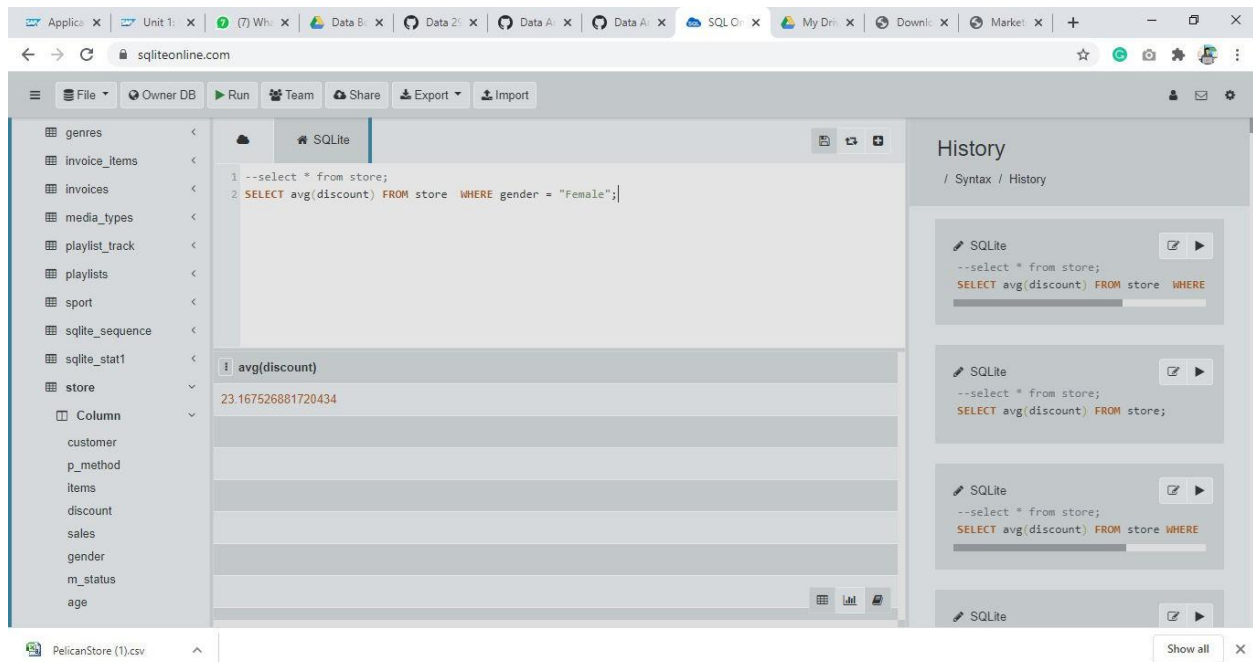
```
1 --select * from store;
2 SELECT * FROM store WHERE gender = "Female" ORDER BY discount DESC;
```

The results are displayed in a table with columns: custo..., p_method, items, discount, sales, gender, m_status, and age.

custo...	p_method	items	discount	sales	gender	m_status	age
50	Mastercard	9	94.3	145.2	Female	Married	46
62	Proprietar...	3	91.48	59.91	Female	Single	30
66	Proprietar...	1	9.9	39.6	Female	Married	62
65	Proprietar...	2	9.5	49.5	Female	Married	48
26	Proprietar...	1	9.48	30.02	Female	Married	60
97	Proprietar...	9	82.75	253	Female	Married	30
12	Proprietar...	1	7.9	31.6	Female	Married	40

The right sidebar shows a History panel with three entries, each containing a snippet of the SQL query.

What is the average discount given to females in the store? select avg(discount) from store where gender = "Female";



Database – Baseball_ball data.csv

Q4. Who are the Top 5 players having height more than average?

Average height – 72.73

Code:-

- select * from sport;
- Select avg(height) from sport;

Select name, height from sport where height > (Select cast (avg(height) as float) from sport) order by height desc limit 5;

The screenshot shows the SQLiteOnline.com web application. The left sidebar lists various database tables including 'artists', 'customers', 'employees', 'genres', 'invoice_items', 'invoices', 'media_types', 'playlist_track', 'playlists', 'sport', 'sqlite_sequence', and 'sqlite_stat1'. The 'sport' table is selected, and its columns (name, handedness, height, weight, avg, HR) are visible in the 'Column' section. The main query editor contains the following SQL code:

```
1 --select * from sport;
2 --Select avg(height) from sport;
3 SELECT name, height FROM sport WHERE height > ( SELECT CAST (avg(height) AS FLOAT) FROM sport) ORDER
```

The results table shows the following data:

name	height
Mike Smithson	80
J.R. Richard	80
Steve Ellsworth	80
Stefan Weber	80
Walt Bond	79

The right sidebar shows the 'History' tab with three previous queries.

Q5. Bottom 2 players having least weight and must be left handed? select

*** from sport where handedness = "L" order by weight ;**

The screenshot shows the SQLiteOnline.com web application. The left sidebar lists various database tables including 'artists', 'customers', 'employees', 'genres', 'invoice_items', 'invoices', 'media_types', 'playlist_track', 'playlists', 'sport', 'sqlite_sequence', and 'sqlite_stat1'. The 'sport' table is selected, and its columns (name, handedness, height, weight, avg, HR) are visible in the 'Column' section. The main query editor contains the following SQL code:

```
1 --select * from sport;
2 --Select avg(height) from sport;
3 --Select name, height from sport where height > ( Select cast (avg(height) as float) from sport) orde
4 --SELECT agent_name,phone_no,commission FROM AGENTS where agent_code IN
5 --(SELECT agent_code from CUSTOMER where outstanding_amt >
6 --(SELECT AVG(outstanding_amt) from CUSTOMER));
7
8 --select name, height, weight from sport order by height;
9 SELECT * FROM sport WHERE handedness = "L" ORDER BY weight ;
```

The results table shows the following data:

name	handedness	height	weight	avg	HR
Vic Davallillo	L	67	150	0.279	36
Sonny Jackson	L	69	150	0.251	7
Curt Ford	L	70	150	0.245	7
Tony Ferreira	L	73	160	0.0	0
Greg Gross	L	70	160	0.287	7
Don Mason	L	71	160	0.205	3
Narciso Elvira	L	70	160	0.0	0
Oscar Gamble	L	71	160	0.265	200
John Donaldson	L	71	160	0.238	4

The right sidebar shows the 'History' tab with three previous queries.

Database – Chinook.db

Q6. Top 2 invoices of Germany customers with names and Do the sum of the invoices of Germany customers.

- select total_invoices, country, firstname from invoices inner join customers on invoices.CustomerId = customers.customerid where country = "Germany" order by total_invoices DESC;

The screenshot shows the SQLiteOnline.com web interface. On the left, a sidebar lists database tables: albums, artists, customers, employees, genres, invoice_items, and invoices. The main area displays a SQL query in a text editor:

```
1 --select * from invoice_items;
2 --select * from invoices;
3 -- select trackid, customerid , invoices.InvoiceId from invoices left join invoice_items on invoice_i
4 --select firstname, trackid, country from invoice_items a left outer join invoices b on b.InvoiceId =
5 SELECT total_invoices, country, firstname FROM invoices INNER JOIN customers ON invoices.CustomerId =
6
7
```

Below the query, the results are shown in a table with columns: Total_Invoices, Country, and FirstName. The results are sorted by Total_Invoices in descending order.

Total_Invoices	Country	FirstName
14.91	Germany	Fynn
13.86	Germany	Leonie
13.86	Germany	Hannah
13.86	Germany	Fynn
13.86	Germany	Niklas
8.91	Germany	Leonie
8.91	Germany	Hannah
8.91	Germany	Niklas
5.94	Germany	Leonie

On the right side of the interface, there is a 'History' panel showing the executed SQL queries.

Q7. Which artist gives the maximum track? (Tables used – albums, artists and tracks)

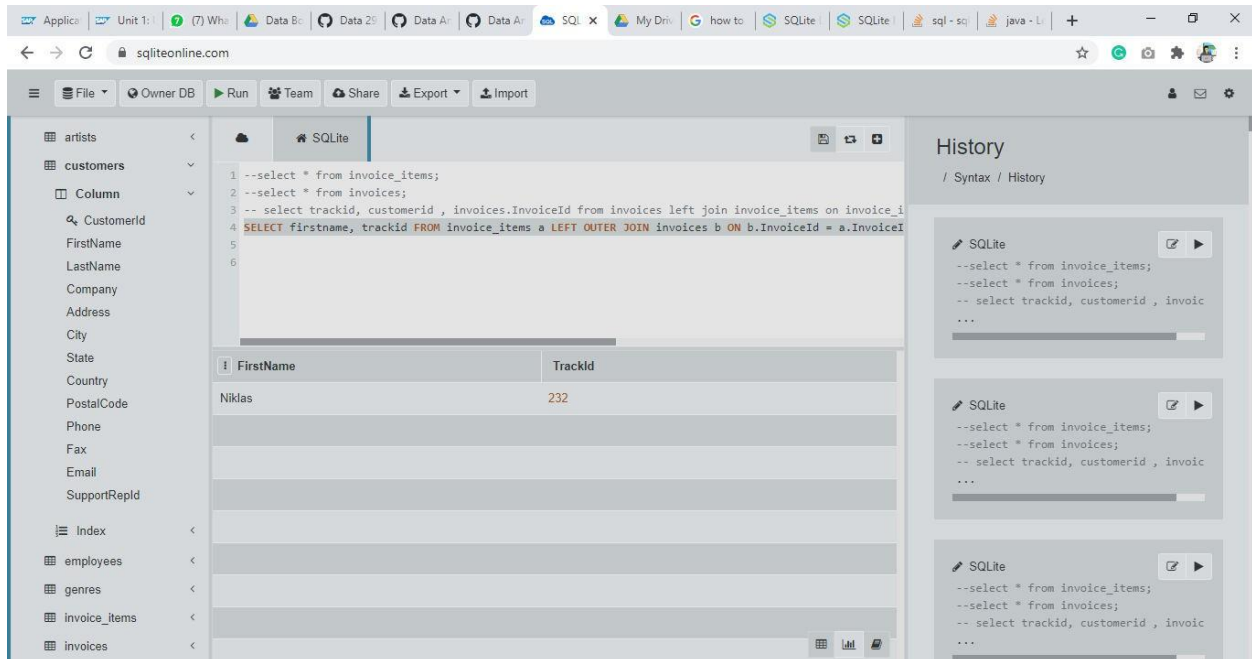
select a.ArtistId, b.AlbumId, c.TrackId, c.Name, a.Name, c.Composer from artists a left outer join albums b on b.ArtistId = a.ArtistId left outer join tracks c on c.AlbumId = b.AlbumId;

Further i need to discuss with vardaan..... (have a doubt in this question) **Q8.**

Name the customer have the track id 232?

- select trackid, customerid , invoices.InvoiceId from invoices left join invoice_items on invoice_items.InvoiceId = invoices.InvoiceId;

Final code - select firstname, trackid from invoice_items a left outer join invoices b on b.InvoiceId = a.InvoiceId left OUTER JOIN customers c on c.CustomerId = b.CustomerId where trackid = "232";



Dataset - imdb

Q9 -> top 5 directors having rating more then average?

Select director, actors, rating from imdb where rating > (SELECT avg(rating) from imdb) order by rating limit "5";

The screenshot shows the SQLiteOnline.com web interface. The browser tabs include 'Application', 'Unit 1: Use', '(8) WhatsA', 'Data Boot', 'Data 29 Jul', 'Data Analy', 'Data Analy', 'SQL Online', and 'My Drive'. The URL bar shows 'sqliteonline.com'. The interface has a top navigation bar with 'File', 'Owner DB', 'Run', 'Team', 'Share', 'Export', and 'Import' buttons. On the left, there's a sidebar with 'SQLite' selected, showing a table list with 'demo' and 'imdb'. The 'imdb' table is expanded, showing columns: Rank, Title, Genre, Description, Director, Actors, Year, Runtime (Minutes), Rating, Votes, Revenue (Millions), and Metascore. The main area displays a SQL query:


```
1 --SELECT avg(rating) from imdb;
2 --Select director, actors, rating from imdb where rating>"6.72";
3 SELECT director, actors, rating FROM imdb WHERE rating > (SELECT avg(rating) FROM imdb) ORDER BY rati
4
```

 Below the query, a table of results is shown:

Director	Actors	Rating
Jason Friedberg	Carmen Electra, Vanessa Lachey, Ni...	1.9
Shawn Burkett	Brittany Blanton, Ayse Howard, Ro...	2.7
James Wong	Justin Chatwin, James Marsters, Yu...	2.7
Jonathan Holbrook	Dan Crisafulli, Kay Whitney, Richard...	3.2
Femi Oyeniran	Dylan Duffus, Scorchier, Shone Rom...	3.5

 On the right, a 'History' panel shows the query history. The bottom of the interface shows other database options like 'MariaDB' and 'PostgreSQL'.

Database - Create two tables to practice sub queries (agents and customer)

Q10. Name the Agents whose customers have the maximum outstanding amount.

Code - select outstanding_amt,CUSTOMER.agent_code,agent_name,cust_name
from AGENTS inner join CUSTOMER on AGENTS.AGENT_CODE =
CUSTOMER.agent_code order by outstanding_amt desc;

The screenshot shows the SQLiteonline.com interface. The SQL editor contains the following query:

```

65 ai, 'India', '3', '7000.00', '11000.00', '7000.00', '11000.00', 'PPHGRTS', 'A010');
66
67
68 AGENTS INNER JOIN CUSTOMER ON AGENTS.AGENT_CODE = CUSTOMER.agent_code ORDER BY outstanding_amt DESC;

```

The results table is displayed below the query:

OUTSTANDING_AMT	AGENT_CODE	AGENT_NAME	CUST_NAME
12000	A002	Mukesh	Ramesh
12000	A001	Subbarao	Rangarappa
12000	A007	Ramasundar	Venkatpati
11000	A003	Alex	Stuart
11000	A002	Mukesh	Sasikant
11000	A004	Ivan	Shilton

Q11. Name all the agents whose customers have less than average outstanding amount.

select outstanding_amt,CUSTOMER.agent_code,agent_name,cust_name from
 AGENTS inner join CUSTOMER on AGENTS.AGENT_CODE =
 CUSTOMER.agent_code where outstanding_amt < (select avg(outstanding_amt)
 from CUSTOMER);

The screenshot shows the SQLiteonline.com interface. The SQL editor contains the following query:

```

55 INSERT INTO CUSTOMER VALUES ('C00023', 'Karl', 'London', 'London', 'UK', '0', '4000.00', '6000.00', '
56 INSERT INTO CUSTOMER VALUES ('C00006', 'Shilton', 'Toronto', 'Toronto', 'Canada', '1', '10000.00', '
57 INSERT INTO CUSTOMER VALUES ('C00010', 'Charles', 'Hampshire', 'Hampshire', 'UK', '3', '6000.00', '4
58 INSERT INTO CUSTOMER VALUES ('C00017', 'Srinivas', 'Bangalore', 'Bangalore', 'India', '2', '8000.00'
59 INSERT INTO CUSTOMER VALUES ('C00012', 'Steven', 'San Jose', 'San Jose', 'USA', '1', '5000.00', '700
60 INSERT INTO CUSTOMER VALUES ('C00008', 'Karolina', 'Toronto', 'Toronto', 'Canada', '1', '7000.00', '
61 INSERT INTO CUSTOMER VALUES ('C00003', 'Martin', 'Toronto', 'Toronto', 'Canada', '2', '8000.00', '70
62 INSERT INTO CUSTOMER VALUES ('C00009', 'Ramesh', 'Mumbai', 'Mumbai', 'India', '3', '8000.00', '7000.
63 INSERT INTO CUSTOMER VALUES ('C00014', 'Rangarappa', 'Bangalore', 'Bangalore', 'India', '2', '8000.0
64 INSERT INTO CUSTOMER VALUES ('C00016', 'Venkatpati', 'Bangalore', 'Bangalore', 'India', '2', '8000.0
65 INSERT INTO CUSTOMER VALUES ('C00011', 'Sundariya', 'Chennai', 'Chennai', 'India', '3', '7000.00', '
66 */
67 --select * from CUSTOMER;
68 SELECT outstanding_amt,CUSTOMER.agent_code,agent_name,cust_name FROM AGENTS INNER JOIN CUSTOMER ON A
69 --select agent_name,agent_code,outstanding_amt from AGENTS where agent_code IN ( select agent_code f

```

The results table is displayed below the query:

OUTSTANDING_AMT	AGENT_CODE	AGENT_NAME	CUST_NAME
4000	A003	Alex	Holmes
6000	A008	Alford	Micheal
6000	A008	Alford	Albert
6000	A006	McDen	Cook
3000	A008	Alford	Bolt
5000	A005	Anderson	Fleming

