

2.1 Basic SELECT

1. Get all customers

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT * FROM customers;
```

customer_id	name	email	city	state
1	Aisha Khan	aisha@mail.com	Pune	MH
2	Rohan Patil	rohan@mail.com	Mumbai	MH
3	Sneha Joshi	sneha@mail.com	Nagpur	MH
4	Imran Shaikh	imran@mail.com	Delhi	DL
5	Sarah Ali	sarah@mail.com	Bangalore	KA

Execution finished without errors.
Result: 5 rows returned in 44ms
At line 1:
SELECT * FROM customers;

2. Show all products with price > 2000

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT * FROM products
2 WHERE price > 2000;
```

product_id	product_name	category	price
1	Laptop	Electronics	55000.0
2	Smartphone	Electronics	20000.0
3	Watch	Accessories	2500.0
4	Shoes	Fashion	3000.0

Execution finished without errors.
Result: 4 rows returned in 22ms
At line 1:
SELECT * FROM products
WHERE price > 2000;

2.2 ORDER BY

3. Sort products by price (high → low)

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT * FROM products
2 ORDER BY price DESC;
3
```

	product_id	product_name	category	price
1	1	Laptop	Electronics	55000.0
2	2	Smartphone	Electronics	20000.0
3	4	Shoes	Fashion	3000.0
4	3	Watch	Accessories	2500.0
5	5	Headphones	Electronics	1500.0

Execution finished without errors.
Result: 5 rows returned in 33ms
At line 1:
SELECT * FROM products
ORDER BY price DESC;

2.3 GROUP BY + Aggregate Functions

4. Count of orders per customer

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT c.name, COUNT(o.order_id) AS total_orders
2 FROM customers c
3 LEFT JOIN orders o ON c.customer_id = o.customer_id
4 GROUP BY c.customer_id;
5
```

	name	total_orders
1	Aisha Khan	2
2	Rohan Patil	1
3	Sneha Joshi	1
4	Imran Shaikh	1
5	Sarah Ali	0

Execution finished without errors.
Result: 5 rows returned in 21ms
At line 1:
SELECT c.name, COUNT(o.order_id) AS total_orders
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id;

5. Total revenue by product category

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT p.category, SUM(p.price * oi.quantity) AS total_revenue
2 FROM products p
3 JOIN order_items oi ON p.product_id = oi.product_id
4 GROUP BY p.category;
5
```

	category	total_revenue
1	Accessories	5000.0
2	Electronics	134500.0
3	Fashion	3000.0

Execution finished without errors.
Result: 3 rows returned in 24ms
At line 1:
SELECT p.category, SUM(p.price * oi.quantity) AS total_revenue
FROM products p
JOIN order_items oi ON p.product_id = oi.product_id
GROUP BY p.category;

2.4 JOINS

6. All orders with customer names

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT o.order_id, c.name, o.order_date
2 FROM orders o
3 INNER JOIN customers c ON o.customer_id = c.customer_id;
4
```

	order_id	name	order_date
1	101	Aisha Khan	2025-01-02
2	102	Rohan Patil	2025-01-05
3	103	Aisha Khan	2025-01-10
4	104	Sneha Joshi	2025-01-12
5	105	Imran Shaikh	2025-01-15

Execution finished without errors.
Result: 5 rows returned in 22ms
At line 1:
SELECT o.order_id, c.name, o.order_date
FROM orders o
INNER JOIN customers c ON o.customer_id = c.customer_id;

7. Full order details (customer + product + quantity + price)

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT
2     c.name AS customer,
3     p.product_name,
4     oi.quantity,
5     (p.price * oi.quantity) AS total_price
6 FROM order_items oi
7 JOIN orders o ON oi.order_id = o.order_id
8 JOIN customers c ON o.customer_id = c.customer_id
9 JOIN products p ON oi.product_id = p.product_id;
```

	customer	product_name	quantity	total_price
1	Aisha Khan	Laptop	1	55000.0
2	Aisha Khan	Watch	2	5000.0
3	Rohan Patil	Smartphone	1	20000.0
4	Aisha Khan	Headphones	3	4500.0
5	Sneha Joshi	Shoes	1	3000.0
6	Imran Shaikh	Laptop	1	55000.0

2.5 SUBQUERIES

8. Find customers who spent more than ₹50,000

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT name FROM customers
2 WHERE customer_id IN (
3     SELECT o.customer_id
4     FROM orders o
5     JOIN order_items oi ON o.order_id = oi.order_id
6     JOIN products p ON oi.product_id = p.product_id
7     GROUP BY o.customer_id
8     HAVING SUM(p.price * oi.quantity) > 50000
9 );
```

	name
1	Aisha Khan
2	Imran Shaikh

9. Get product with the highest price

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 SELECT product_name, price
2 FROM products
3 WHERE price = (SELECT MAX(price) FROM products);
4
```

	product_name	price
1	Laptop	55000.0

2.6 CREATE VIEWS

10. Create a revenue summary view

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1 CREATE VIEW revenue_summary AS
2 SELECT
3     p.product_name,
4     SUM(oi.quantity) AS total_quantity,
5     SUM(oi.quantity * p.price) AS total_revenue
6 FROM products p
7 JOIN order_items oi ON p.product_id = oi.product_id
8 GROUP BY p.product_id;
9
```

11. Use the view

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Databases

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1*

```
1 SELECT * FROM revenue_summary;
```

	product_name	total_quantity	total_revenue
1	Laptop	2	110000.0
2	Smartphone	1	20000.0
3	Watch	2	5000.0
4	Shoes	1	3000.0
5	Headphones	3	4500.0

p

2.7 INDEXES (for optimization)

12. Create indexes

DB Browser for SQLite - C:\Users\hp\Desktop\Elevate Labs - DA Internship\Day 3 - Task 3\sample.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Databases

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1*

```
1 CREATE INDEX idx_order_customer ON orders(customer_id);
2 CREATE INDEX idx_order_items_prod ON order_items(product_id);
3
```

Execution finished without errors.
Result: query executed successfully. Took 0ms
At line 2:
CREATE INDEX idx_order_items_prod ON order_items(product_id);