**1. create database**

CMD==> sqlite3 inventory.db

sqlite>.databases

**2. create tables**

Table : users

sqlite> CREATE TABLE users (

user\_id INTEGER PRIMARY KEY,

username TEXT,

email TEXT,

created\_at TEXT

);

Table: products

sqlite> CREATE TABLE products (

product\_id INTEGER PRIMARY KEY,

product\_name TEXT,

price REAL,

description TEXT

);

Table: orders

sqlite> CREATE TABLE orders (

order\_id INTEGER PRIMARY KEY,

user\_id INTEGER,

product\_id INTEGER,

quantity INTEGER,

order\_date TEXT,

FOREIGN KEY (user\_id) REFERENCES users (user\_id),

FOREIGN KEY (product\_id) REFERENCES products (product\_id)

);

**3. insert values**

Table: users

INSERT INTO users (username, email, created\_at)

VALUES ('johndoe', 'johndoe@example.com', '2023-05-01');

INSERT INTO users (username, email, created\_at)

VALUES ('janedoe', 'janedoe@example.com', '2023-05-02');

INSERT INTO users (username, email, created\_at)

VALUES ('alexsmith', 'alexsmith@example.com', '2023-05-03');

INSERT INTO users (username, email, created\_at)

VALUES ('sarahj', 'sarahj@example.com', '2023-05-04');

INSERT INTO users (username, email, created\_at)

VALUES ('mikec', 'mikec@example.com', '2023-05-05');

INSERT INTO users (username, email, created\_at)

VALUES ('emilyw', 'emilyw@example.com', '2023-05-06');

INSERT INTO users (username, email, created\_at)

VALUES ('davidm', 'davidm@example.com', '2023-05-07');

INSERT INTO users (username, email, created\_at)

VALUES ('lisaw', 'lisaw@example.com', '2023-05-08');

INSERT INTO users (username, email, created\_at)

VALUES ('kevinp', 'kevinp@example.com', '2023-05-09');

INSERT INTO users (username, email, created\_at)

VALUES ('amandar', 'amandar@example.com', '2023-05-10');

Table: products

INSERT INTO products (product\_name, price, description)

VALUES ('Laptop', 999.99, 'High-performance laptop');

INSERT INTO products (product\_name, price, description)

VALUES ('Smartphone', 699.99, 'Latest smartphone model');

INSERT INTO products (product\_name, price, description)

VALUES ('Headphones', 149.99, 'Noise-canceling headphones');

INSERT INTO products (product\_name, price, description)

VALUES ('Smart TV', 1499.99, '55-inch 4K Smart TV');

INSERT INTO products (product\_name, price, description)

VALUES ('Fitness Tracker', 79.99, 'Activity tracker with heart rate monitor');

INSERT INTO products (product\_name, price, description)

VALUES ('Wireless Earbuds', 129.99, 'True wireless earbuds with Bluetooth 5.0');

INSERT INTO products (product\_name, price, description)

VALUES ('Gaming Console', 399.99, 'Next-gen gaming console');

INSERT INTO products (product\_name, price, description)

VALUES ('Camera', 599.99, 'DSLR camera with multiple lenses');

INSERT INTO products (product\_name, price, description)

VALUES ('Tablet', 349.99, '10-inch tablet with high-resolution display');

INSERT INTO products (product\_name, price, description)

VALUES ('Bluetooth Speaker', 89.99, 'Portable wireless speaker with long battery life');

Table: orders

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (1, 2, 1, '2023-05-05');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (2, 3, 2, '2023-05-06');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (1, 1, 1, '2023-05-07');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (3, 4, 1, '2023-05-08');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (4, 5, 1, '2023-05-09');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (5, 6, 1, '2023-05-10');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (6, 7, 1, '2023-05-11');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (7, 8, 1, '2023-05-12');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (8, 9, 1, '2023-05-13');

INSERT INTO orders (user\_id, product\_id, quantity, order\_date)

VALUES (9, 10, 1, '2023-05-14');

**4. Display all content of each table**

Select \* from users;

Select \* from products;

Select \* from orders;

**5. Retrieve all the usernames and emails from the users table.**

SELECT username, email FROM users;

**6. Find the product name and price for the product with product\_id equal to 3.**

SELECT product\_name, price FROM products WHERE product\_id = 3;

**7. Get the total quantity of orders placed by user with user\_id equal to 1.**

SELECT SUM(quantity) AS total\_quantity FROM orders WHERE user\_id = 1;

**8. List all the orders along with the corresponding usernames and product names.**

SELECT orders.order\_id, users.username, products.product\_name

FROM orders

JOIN users ON orders.user\_id = users.user\_id

JOIN products ON orders.product\_id = products.product\_id;

**9. Find the number of users who have created their accounts after a specific date.**

SELECT COUNT(\*) AS user\_count FROM users WHERE created\_at > '2023-05-05';

**10. Retrieve the usernames and order dates for all orders that were placed after a specific date.**

SELECT users.username, orders.order\_date

FROM orders

JOIN users ON orders.user\_id = users.user\_id

WHERE orders.order\_date > '2023-05-10';

**11. Get the average price of all the products in the products table.**

SELECT AVG(price) AS average\_price FROM products;

**12. Find the total quantity of a specific product ordered across all orders.**

SELECT SUM(quantity) AS total\_quantity FROM orders WHERE product\_id = 2;

**13. Retrieve the usernames of the users who have placed an order for a product with a price greater than 500.**

SELECT users.username

FROM users

JOIN orders ON users.user\_id = orders.user\_id

JOIN products ON orders.product\_id = products.product\_id

WHERE products.price > 500;

**14. Get the latest order placed along with the corresponding username and product name.**

SELECT orders.order\_id, users.username, products.product\_name

FROM orders

JOIN users ON orders.user\_id = users.user\_id

JOIN products ON orders.product\_id = products.product\_id

ORDER BY orders.order\_date DESC

LIMIT 1;

**15. Update the price of the product with product\_id equal to 5 to $89.99.**

UPDATE products SET price = 89.99 WHERE product\_id = 5;

**16. Change the email of the user with user\_id equal to 3 to 'alex@example.com'.**

UPDATE users SET email = 'alex@example.com' WHERE user\_id = 3;

**17. Increase the quantity of all orders placed by user with user\_id equal to 2 by 1.**

UPDATE orders SET quantity = quantity + 1 WHERE user\_id = 2;

**18. Add a new column called discount of type INTEGER to the products table.**

ALTER TABLE products ADD COLUMN discount INTEGER;

**19. Rename the column created\_at to registration\_date in the users table.**

ALTER TABLE users RENAME COLUMN created\_at TO registration\_date;

**20. Remove the column description from the products table.**

ALTER TABLE products DROP COLUMN description;

**21. Delete all orders placed by the user with user\_id equal to 4.**

DELETE FROM orders WHERE user\_id = 4;

**22. Delete all users who registered before the year 2022.**

DELETE FROM users WHERE registration\_date < '2022-01-01';

**23. Delete all data from the users table.**

DELETE FROM users;

or

TRUNCATE TABLE products;

**24. Remove(Drop) the orders table from the database.**

DROP TABLE orders;