

```
-- CREATE DATABASE
CREATE DATABASE user_db;

-- CREATE TABLE
CREATE TABLE user_db.users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age INT NOT NULL,
    is_student TINYINT NOT NULL,
    course VARCHAR(100)
);

-- INSERT DATA
INSERT INTO user_db.users (name, age, is_student, course)
VALUES ('Alice Johnson', 21, 1, 'Computer Science');

INSERT INTO user_db.users (name, age, is_student, course)
VALUES ('Bob Smith', 30, 0, NULL);

-- SELECT ALL DATA
SELECT * FROM user_db.users;

-- SELECT SPECIFIC COLUMNS
SELECT name, age FROM user_db.users;

-- SORT RESULTS
SELECT * FROM user_db.users
ORDER BY is_student DESC, age ASC;

-- LIMIT RESULTS
SELECT * FROM user_db.users LIMIT 10;

-- UPDATE DATA
UPDATE user_db.users
SET age = 25
WHERE id = 1;

-- DELETE DATA
DELETE FROM user_db.users WHERE id = 2;

-- FILTERING DATA WITH WHERE
SELECT * FROM user_db.users WHERE is_student = 1;

-- FILTERING WITH AND / OR
SELECT * FROM user_db.users WHERE age > 20 AND is_student = 1;

-- WILDCARD SEARCH
```

```

SELECT * FROM user_db.users WHERE name LIKE '%sam%';

-- DISTINCT VALUES
SELECT DISTINCT course FROM user_db.users;

-- BETWEEN OPERATOR
SELECT * FROM user_db.users WHERE age BETWEEN 18 AND 25;

-- IN OPERATOR
SELECT * FROM user_db.users WHERE course IN ('Computer Science', 'Engineering');

-- AGGREGATE FUNCTIONS
SELECT COUNT(*) AS total_users FROM user_db.users;
SELECT SUM(age) AS total_age FROM user_db.users;
SELECT AVG(age) AS avg_age FROM user_db.users WHERE is_student = 1;

-- GROUP BY AND HAVING
SELECT course, COUNT(*) AS total_students
FROM user_db.users
GROUP BY course
HAVING COUNT(*) > 1;

-- INNER JOIN
SELECT u.name, c.course_name
FROM user_db.users u
INNER JOIN user_db.courses c ON u.course = c.course_name;

-- LEFT JOIN
SELECT u.name, c.course_name
FROM user_db.users u
LEFT JOIN user_db.courses c ON u.course = c.course_name;

-- CASE STATEMENT
SELECT name, age,
       CASE
           WHEN age < 18 THEN 'Minor'
           WHEN age BETWEEN 18 AND 25 THEN 'Young Adult'
           ELSE 'Adult'
       END AS age_group
FROM user_db.users;

-- CREATE INDEX
CREATE INDEX idx_course ON user_db.users(course);

-- DROP INDEX
DROP INDEX idx_course ON user_db.users;

-- ALTER TABLE (ADD COLUMN)

```

```
ALTER TABLE user_db.users ADD email VARCHAR(150);
```

```
-- ALTER TABLE (MODIFY COLUMN)
```

```
ALTER TABLE user_db.users MODIFY age SMALLINT;
```