```sql
--q1
CREATE TABLE Sailors (sid INT,sname VARCHAR(255),rating INT,age REAL);
CREATE TABLE Boats (bid INT,bname VARCHAR(255),color VARCHAR(255);
CREATE TABLE Reserves (sid INT,bid INT);

INSERT INTO Sailors VALUES (1, 'John', 5, 25.5),(2, 'Alice', 3, 22.0);
INSERT INTO Boats VALUES (1, 'Boat1', 'Red'),(2, 'Boat2', 'Blue');
INSERT INTO Reserves VALUES (1, 1),(2, 2);

--1 name and ages
SELECT sname, age FROM Sailors;

--2 at least one boat
SELECT DISTINCT sname
FROM Sailors
WHERE sid IN (SELECT sid FROM Reserves);

--3 not reseved at least 1 boat
SELECT DISTINCT sname
FROM Sailors
WHERE sid NOT IN (SELECT sid FROM Reserves WHERE bid IN (SELECT bid FROM Boats WHERE
color = 'Red'));

--4 name and age of oldest
SELECT TOP 1 sname, age
FROM Sailors
ORDER BY age DESC;

--5 age of youngest 18
SELECT MIN(age) AS youngest_age
FROM Sailors
WHERE age >= 18;
```

```sql
--q2
CREATE TABLE Sailors (sid INT,sname VARCHAR(255),rating INT,age REAL);
CREATE TABLE Boats (bid INT,bname VARCHAR(255),color VARCHAR(255);
CREATE TABLE Reserves (sid INT,bid INT);

INSERT INTO Sailors VALUES (1, 'John', 5, 25.5),(2, 'Alice', 3, 22.0);
INSERT INTO Boats VALUES (1, 'Boat1', 'Red'),(2, 'Boat2', 'Blue');
INSERT INTO Reserves VALUES (1, 1),(2, 2);

--1 rating above 7
SELECT * FROM Sailors WHERE rating > 7;

--2 reserved red or green boat
SELECT DISTINCT sname
FROM Sailors
WHERE sid IN (SELECT sid FROM Reserves WHERE bid IN (SELECT bid FROM Boats WHERE color =
'Red' OR color = 'Green'));

--3 ratting better than ram
SELECT sname, rating
FROM Sailors
WHERE rating > (SELECT rating FROM Sailors WHERE sname = 'Ram');

--4 number of sailors
SELECT COUNT(*) AS sailor_count FROM Sailors;
```

```sql
--5 number of reservations for boat
SELECT bname AS boat_name, COUNT(sid) AS reservation_count
FROM Boats b
LEFT JOIN Reserves r ON b.bid = r.bid
WHERE color = 'Red'
GROUP BY bname;
```

```sql
--q3

CREATE TABLE Sailors (sid INT,sname VARCHAR(255),rating INT,age REAL);
CREATE TABLE Boats (bid INT,bname VARCHAR(255),color VARCHAR(255);
CREATE TABLE Reserves (sid INT,bid INT);

INSERT INTO Sailors VALUES (1, 'John', 5, 25.5),(2, 'Alice', 3, 22.0);
INSERT INTO Boats VALUES (1, 'Boat1', 'Red'),(2, 'Boat2', 'Blue');
INSERT INTO Reserves VALUES (1, 1),(2, 2);

--1Find the name of sailors who have reserved boat number 103:
SELECT sname
FROM Sailors
WHERE sid IN (SELECT sid FROM Reserves WHERE bid = 103);

--2 Find the name of sailors who have reserved both a red and a green boat:
SELECT sname
FROM Sailors
WHERE sid IN (
    SELECT sid
    FROM Reserves
    WHERE bid IN (SELECT bid FROM Boats WHERE color IN ('Red', 'Green'))
    GROUP BY sid
    HAVING COUNT(DISTINCT bid) = 2
);

--3 sailors who have reserved all boats:
SELECT sname
FROM Sailors s
WHERE NOT EXISTS (
    SELECT b.bid
    FROM Boats b
    EXCEPT
    SELECT r.bid
    FROM Reserves r
    WHERE r.sid = s.sid
);

--4 Count the number of different sailor names:
SELECT COUNT(DISTINCT sname) AS unique_sailor_count
FROM Sailors;

--5  Find the average age of sailors for each rating level:
SELECT rating, AVG(age) AS average_age
FROM Sailors
GROUP BY rating;
```

```sql
--q4
CREATE TABLE Suppliers (
    sid INT,
    sname VARCHAR(255),
    address VARCHAR(255)
);

CREATE TABLE Parts (
    pid INT,
    pname VARCHAR(255),
    color VARCHAR(255)
);

CREATE TABLE Catalog (
    sid INT,
    pid INT,
    cost REAL
);


INSERT INTO Suppliers VALUES (1, 'Supplier1', 'Address1'),(2, 'Supplier2', 'Address2');

INSERT INTO Parts VALUES (1, 'Part1', 'Red'),(2, 'Part2', 'Green');

INSERT INTO Catalog VALUES (1, 1, 10.0),(1, 2, 15.0),(2, 1, 12.0),(2, 2, 18.0);


--1 Find the pnames of
SELECT DISTINCT pname
FROM Catalog
JOIN Parts ON Catalog.pid = Parts.pid;

--2 Find the snames of suppliers who supply every red part:
SELECT DISTINCT sname
FROM Suppliers
WHERE sid IN (SELECT sid FROM Catalog WHERE pid IN (SELECT pid FROM Parts WHERE color =
'Red'));

--3  who supply that part
SELECT DISTINCT sid
FROM Catalog c1
WHERE cost > (SELECT AVG(cost) FROM Catalog c2 WHERE c2.pid = c1.pid);

--4  red part and a green part:
SELECT sid
FROM Catalog
WHERE pid IN (SELECT pid FROM Parts WHERE color IN ('Red', 'Green'))
GROUP BY sid
HAVING COUNT(DISTINCT pid) = 2;
```

```sql
--q5
CREATE TABLE Suppliers (
    sid INT,
    sname VARCHAR(255),
    address VARCHAR(255)
);
```

```sql
CREATE TABLE Parts (
    pid INT,
    pname VARCHAR(255),
    color VARCHAR(255)
);

CREATE TABLE Catalog (
    sid INT,
    pid INT,
    cost REAL
);

INSERT INTO Suppliers VALUES (1, 'Supplier1', 'Address1'),(2, 'Supplier2', 'Address2');
INSERT INTO Parts VALUES (1, 'Part1', 'Red'),(2, 'Part2', 'Green');
INSERT INTO Catalog VALUES (1, 1, 10.0),(1, 2, 15.0),(2, 1, 12.0),(2, 2, 18.0);

--1 snames of suppliers
SELECT DISTINCT sname
FROM Suppliers
WHERE NOT EXISTS (
    SELECT pid
    FROM Parts
    WHERE pid NOT IN (SELECT pid FROM Catalog WHERE sid = Suppliers.sid)
);

--2 Acme Widget
SELECT DISTINCT pname
FROM Parts
WHERE pid IN (
    SELECT pid
    FROM Catalog
    WHERE sid = (SELECT sid FROM Suppliers WHERE sname = 'Acme Widget Suppliers')
)
AND pid NOT IN (
    SELECT pid
    FROM Catalog
    WHERE sid <> (SELECT sid FROM Suppliers WHERE sname = 'Acme Widget Suppliers')
);



--3 supplier who charges the most for that part:
SELECT pname, MAX(sname) AS supplier_name
FROM Catalog
JOIN Parts ON Catalog.pid = Parts.pid
JOIN Suppliers ON Catalog.sid = Suppliers.sid
GROUP BY pname;

--4 red part or a green part
SELECT DISTINCT sid
FROM Catalog
WHERE pid IN (SELECT pid FROM Parts WHERE color IN ('Red', 'Green'));
```

```sql
--q6

CREATE TABLE Flights (
```

```
    flno INT,
    frm VARCHAR(255),
    to_city VARCHAR(255),
    distance INT,
    price INT
);

CREATE TABLE Aircraft (
    aid INT,
    aname VARCHAR(255),
    cruisingrange INT
);

CREATE TABLE Certified (
    pid INT,
    aid INT
);

CREATE TABLE Pilot (
    pid INT,
    pname VARCHAR(255),
    salary INT
);

INSERT INTO Flights VALUES (1, 'Los Angeles', 'Chicago', 2000, 500),
INSERT INTO Flights VALUES (2, 'Washington', 'Chicago', 800, 300);

INSERT INTO Aircraft VALUES (101, 'Boeing 747', 1500);
INSERT INTO Aircraft VALUES (102, 'Airbus A320', 1200);

INSERT INTO Certified VALUES (1, 101);
INSERT INTO Certified VALUES (2, 102);

INSERT INTO Pilot VALUES (1, 'John', 90000);
INSERT INTO Pilot VALUES (2, 'Alice', 85000);


--1 earn more than 80,000:
SELECT DISTINCT aname
FROM Aircraft
WHERE NOT EXISTS (
    SELECT *
    FROM Certified
    JOIN Pilot ON Certified.pid = Pilot.pid
    WHERE Certified.aid = Aircraft.aid AND Pilot.salary <= 80000
);

--2 For all aircraft with cruisingrange over 1,000 miles, find the name of the aircraft:
SELECT aname
FROM Aircraft
WHERE cruisingrange > 1000;

--3 Find the aids of all aircraft that can be used on routes from Los Angeles to Chicago:
SELECT DISTINCT a.aid
FROM Aircraft a
JOIN Certified c ON a.aid = c.aid
JOIN Pilot p ON c.pid = p.pid
JOIN Flights f ON a.cruisingrange >= f.distance
```

```sql
WHERE f.frm = 'Los Angeles' AND f.to_city = 'Chicago';


--4  Find the distance and price for the flight used on the route Washington to Chicago:
SELECT distance, price
FROM Flights
WHERE frm = 'Washington' AND to_city = 'Chicago';
```

```sql
--q7

CREATE TABLE Flights (
    flno INT,
    frm VARCHAR(255),
    to_city VARCHAR(255),
    distance INT,
    price INT
);

CREATE TABLE Aircraft (
    aid INT,
    aname VARCHAR(255),
    cruisingrange INT
);

CREATE TABLE Certified (
    pid INT,
    aid INT
);

CREATE TABLE Pilot (
    pid INT,
    pname VARCHAR(255),
    salary INT
);

INSERT INTO Flights VALUES (1, 'Los Angeles', 'Chicago', 2000, 500),
INSERT INTO Flights VALUES (2, 'Washington', 'Chicago', 800, 300);

INSERT INTO Aircraft VALUES (101, 'Boeing 747', 1500);
INSERT INTO Aircraft VALUES (102, 'Airbus A320', 1200);

INSERT INTO Certified VALUES (1, 101);
INSERT INTO Certified VALUES (2, 102);

INSERT INTO Pilot VALUES (1, 'John', 90000);
INSERT INTO Pilot VALUES (2, 'Alice', 85000);


--1  For each pilot who is certified for more than three aircraft, find the pid:
SELECT pid
FROM Certified
GROUP BY pid
HAVING COUNT(aid) > 3;

--2 salary of all pilots certified for this aircraft:
SELECT c.aid, AVG(p.salary) AS average_salary
FROM Certified c
JOIN Aircraft a ON c.aid = a.aid
```

```sql
JOIN Pilot p ON c.pid = p.pid
WHERE a.cruisingrange > 1000
GROUP BY c.aid;

--3 from Chicago to Los Angeles:
SELECT aid
FROM Flights
JOIN Aircraft ON Flights.distance <= Aircraft.cruisingrange
WHERE frm = 'Chicago' AND to_city = 'Los Angeles';

--4 flight used on route Washington to Chicago:
SELECT distance, price
FROM Flights
WHERE frm = 'Washington' AND to_city = 'Chicago';
```

```sql
--q8
CREATE TABLE Emp (
    eid INT,
    ename VARCHAR(255),
    age INT,
    salary REAL
);

CREATE TABLE Works (
    eid INT,
    did INT,
    duration INT
);

CREATE TABLE Dept (
    did INT,
    budget REAL,
    managerid INT
);

INSERT INTO Emp VALUES (1, 'John', 25, 50000), (2, 'Alice', 35, 70000);
INSERT INTO Works VALUES (1, 1, 12), (2, 2, 18);
INSERT INTO Dept VALUES (1, 1200000, 1), (2, 800000, 2);

--1Print the names and ages of each employee who works in the Hardware department:
SELECT e.ename, e.age
FROM Emp e
JOIN Works w ON e.eid = w.eid
JOIN Dept d ON w.did = d.did
WHERE d.did = 1;

--2 manage only departments with budgets greater than $1,000,000:
SELECT DISTINCT managerid
FROM Dept d
WHERE NOT EXISTS (
    SELECT did
    FROM Dept
    WHERE managerid = d.managerid AND budget <= 1000000
);

--3 Software department for the maximum duration:
SELECT TOP 1 e.ename
```

```sql
FROM Emp e
JOIN Works w ON e.eid = w.eid
JOIN Dept d ON w.did = d.did
WHERE d.did = 2
ORDER BY w.duration DESC;

--4 manager's age is less than 30:
SELECT did
FROM Dept
WHERE managerid IN (SELECT eid FROM Emp WHERE age < 30);
```

```sql
--9
CREATE TABLE Emp (
    eid INT,
    ename VARCHAR(255),
    age INT,
    salary REAL
);

CREATE TABLE Works (
    eid INT,
    did INT,
    duration INT
);

CREATE TABLE Dept (
    did INT,
    budget REAL,
    managerid INT
);

INSERT INTO Emp VALUES (1, 'John', 25, 50000), (2, 'Alice', 35, 70000);
INSERT INTO Works VALUES (1, 1, 12), (2, 2, 18);
INSERT INTO Dept VALUES (1, 1200000, 1), (2, 800000, 2);

--1  Print the names and ages of each employee who works in the Software department:
SELECT e.ename, e.age
FROM Emp e
JOIN Works w ON e.eid = w.eid
JOIN Dept d ON w.did = d.did
WHERE d.did = 2;

--2 budget of all the departments that he or she works in:
SELECT e.ename
FROM Emp e
WHERE NOT EXISTS (
    SELECT d.did
    FROM Works w
    JOIN Dept d ON w.did = d.did
    WHERE w.eid = e.eid AND e.salary <= d.budget
);

--3 Hardware department for the maximum duration:
SELECT TOP 1 e.ename
FROM Emp e
JOIN Works w ON e.eid = w.eid
JOIN Dept d ON w.did = d.did
```

```sql
WHERE d.did = 1
ORDER BY w.duration DESC;

--4  salary is more than $10,000:
SELECT did
FROM Dept
WHERE managerid IN (SELECT eid FROM Emp WHERE salary > 10000);
```

```sql
--q10

CREATE TABLE Emp (
    eid INT,
    ename VARCHAR(255),
    age INT,
    salary REAL
);

CREATE TABLE Works (
    eid INT,
    did INT,
    duration INT
);

CREATE TABLE Dept (
    did INT,
    budget REAL,
    managerid INT
);

INSERT INTO Emp VALUES (1, 'John', 25, 50000), (2, 'Alice', 35, 70000);
INSERT INTO Works VALUES (1, 1, 12), (2, 2, 18);
INSERT INTO Dept VALUES (1, 1200000, 1), (2, 800000, 2);

--1 both the Hardware department and the Software department:
SELECT e.ename, e.age
FROM Emp e
JOIN Works w ON e.eid = w.eid
JOIN Dept d ON w.did = d.did
WHERE d.did IN (1, 2)
GROUP BY e.eid, e.ename, e.age
HAVING COUNT(DISTINCT d.did) = 2;

--2 more than 20 employees:
SELECT did
FROM Works
GROUP BY did
HAVING COUNT(eid) > 20;

--3 the departments with the largest budget:
SELECT e.ename
FROM Emp e
JOIN Dept d ON e.eid = d.managerid
WHERE d.budget = (SELECT MAX(budget) FROM Dept);

--4 departments managed by "Steven"
SELECT did
FROM Dept
WHERE managerid = (SELECT eid FROM Emp WHERE ename = 'Steven');
```