```
-- =====================================
-- SECTION A: EMP(EID, EMP_NAME, SALARY, DESIGNATION, DEPTNO)
-- =====================================
CREATE TABLE EMP (
  EID NUMBER PRIMARY KEY,
  EMP_NAME VARCHAR2(50),
  SALARY NUMBER,
  DESIGNATION VARCHAR2(30),
  DEPTNO NUMBER
);

INSERT INTO EMP VALUES (101, 'Alice', 4000, 'CLERK', 10);
INSERT INTO EMP VALUES (102, 'Bob', 2800, 'CLERK', 20);
INSERT INTO EMP VALUES (103, 'Carol', 3500, 'MANAGER', 10);
INSERT INTO EMP VALUES (104, 'David', 2900, 'CLERK', 30);
INSERT INTO EMP VALUES (105, 'Steve', 5000, 'CLERK', 10);

-- Answer 1:
SELECT EMP_NAME FROM EMP WHERE DEPTNO = 10;
-- Answer 2:
SELECT EMP_NAME FROM EMP WHERE DESIGNATION = 'CLERK' AND SALARY > 3000;
-- Answer 3:
SELECT EMP_NAME FROM EMP WHERE LENGTH(EMP_NAME) = 5;
-- Answer 4:
SELECT * FROM EMP WHERE ROWNUM <= 5;
-- Answer 5:
SELECT * FROM EMP E1 WHERE SALARY > (SELECT AVG(SALARY) FROM EMP E2);
```

```
-- =====================================
-- SECTION B: EMP_B, COMPANY, MANAGER
-- =====================================
CREATE TABLE EMP_B (
  EMP_NAME VARCHAR2(50),
  EMP_CITY VARCHAR2(50),
  SALARY NUMBER,
  COMPANY_NAME VARCHAR2(50),
  MANAGER_NAME VARCHAR2(50),
  STREET VARCHAR2(50)
);

CREATE TABLE COMPANY (
  COMPANY_NAME VARCHAR2(50),
  COMPANY_CITY VARCHAR2(50)
);

CREATE TABLE MANAGER (
  MANAGER_NAME VARCHAR2(50),
  STREET VARCHAR2(50),
  CITY VARCHAR2(50)
);

INSERT INTO EMP_B VALUES ('John', 'New York', 4000, 'IBM', 'Mike', '5th Ave');
INSERT INTO EMP_B VALUES ('Sara', 'San Jose', 4500, 'Apple', 'Linda', 'Main St');
INSERT INTO EMP_B VALUES ('Ravi', 'Bangalore', 3000, 'IBM', 'Mike', 'MG Road');
INSERT INTO EMP_B VALUES ('Nina', 'New York', 4800, 'IBM', 'Mike', '5th Ave');
INSERT INTO EMP_B VALUES ('Alex', 'San Jose', 3500, 'Apple', 'Linda', 'Main St');

INSERT INTO COMPANY VALUES ('IBM', 'New York');
```

```sql
INSERT INTO COMPANY VALUES ('Apple', 'San Jose');

INSERT INTO COMPANY VALUES ('First Bank Corporation', 'Chicago');


INSERT INTO MANAGER VALUES ('Mike', '5th Ave', 'New York');

INSERT INTO MANAGER VALUES ('Linda', 'Main St', 'San Jose');


-- Answer 1:

SELECT EMP_NAME FROM EMP_B WHERE COMPANY_NAME = 'IBM';

-- Answer 2:

SELECT EMP_NAME FROM EMP_B E JOIN COMPANY C ON E.COMPANY_NAME = C.COMPANY_NAME WHERE E.EMP_CITY = C.COMPANY_CITY;

-- Answer 3:

SELECT EMP_NAME FROM EMP_B E JOIN MANAGER M ON E.MANAGER_NAME = M.MANAGER_NAME WHERE E.EMP_CITY = M.CITY AND E.STREET = M.STREET;

-- Answer 4:

SELECT EMP_NAME FROM EMP_B E WHERE SALARY > (SELECT AVG(SALARY) FROM EMP_B WHERE COMPANY_NAME = E.COMPANY_NAME);

-- Answer 5:

SELECT COMPANY_NAME FROM (

  SELECT COMPANY_NAME, SUM(SALARY) AS PAYROLL FROM EMP_B GROUP BY COMPANY_NAME ORDER BY PAYROLL ASC

) WHERE ROWNUM = 1;
```

```
-- ========================================
-- SECTION C: STUDENT, INSTRUCTOR
-- ========================================
CREATE TABLE STUDENT (
  STUD_ID NUMBER PRIMARY KEY,
  STUD_NAME VARCHAR2(50),
  COURSE VARCHAR2(50)
);

CREATE TABLE INSTRUCTOR (
  INSTRUCTOR_ID NUMBER PRIMARY KEY,
  NAME VARCHAR2(50),
  DEPARTMENT VARCHAR2(50),
  SALARY NUMBER
);

INSERT INTO STUDENT VALUES (1, 'Arun', 'Comp.Sci');
INSERT INTO STUDENT VALUES (2, 'Meena', 'Math');
INSERT INTO STUDENT VALUES (3, 'Ravi', 'Comp.Sci');
INSERT INTO STUDENT VALUES (4, 'Divya', 'Physics');
INSERT INTO STUDENT VALUES (5, 'Tom', 'Comp.Sci');

INSERT INTO INSTRUCTOR VALUES (1, 'Dr. Rao', 'Comp.Sci', 80000);
INSERT INTO INSTRUCTOR VALUES (2, 'Dr. Smith', 'Math', 70000);
INSERT INTO INSTRUCTOR VALUES (3, 'Dr. Lin', 'Physics', 72000);
INSERT INTO INSTRUCTOR VALUES (4, 'Dr. Raj', 'Chemistry', 65000);
INSERT INTO INSTRUCTOR VALUES (5, 'Dr. Zoe', 'Biology', 77000);

-- Answer 1:
SELECT DISTINCT STUD_NAME FROM STUDENT WHERE COURSE = 'Comp.Sci';
```

```
-- Answer 2:

-- Skipping course offerings logic due to lack of related table in original structure

-- Answer 3:

SELECT DEPARTMENT, MAX(SALARY) FROM INSTRUCTOR GROUP BY DEPARTMENT;

-- Answer 4:

SELECT MIN(MAX_SALARY) FROM (

  SELECT MAX(SALARY) AS MAX_SALARY FROM INSTRUCTOR GROUP BY DEPARTMENT

);

-- Answer 5:

SELECT STUD_NAME FROM STUDENT WHERE LOWER(STUD_NAME) LIKE 's%';
```

```
-- ========================================
-- SECTION D: EMP_D, COMPANY_D, MANAGER_D
-- ========================================
CREATE TABLE EMP_D (
  EMP_NAME VARCHAR2(50),
  EMP_CITY VARCHAR2(50),
  SALARY NUMBER,
  COMPANY_NAME VARCHAR2(50),
  MANAGER_NAME VARCHAR2(50),
  STREET VARCHAR2(50)
);

CREATE TABLE COMPANY_D (
  COMPANY_NAME VARCHAR2(50),
  COMPANY_CITY VARCHAR2(50)
);

CREATE TABLE MANAGER_D (
  MANAGER_NAME VARCHAR2(50),
  STREET VARCHAR2(50),
  CITY VARCHAR2(50)
);

INSERT INTO EMP_D VALUES ('Tom', 'Chicago', 5000, 'First Bank Corporation', 'Jade', 'Lakeview');
INSERT INTO EMP_D VALUES ('Amy', 'New York', 4200, 'IBM', 'Ron', '5th Ave');
INSERT INTO EMP_D VALUES ('Rick', 'Chicago', 4700, 'First Bank Corporation', 'Jade', 'Lakeview');
INSERT INTO EMP_D VALUES ('Mia', 'San Francisco', 3900, 'Google', 'Zoe', 'Market St');
INSERT INTO EMP_D VALUES ('Leo', 'Chicago', 5100, 'First Bank Corporation', 'Jade', 'Lakeview');

INSERT INTO COMPANY_D VALUES ('First Bank Corporation', 'Chicago');
```

```sql
INSERT INTO COMPANY_D VALUES ('IBM', 'New York');

INSERT INTO COMPANY_D VALUES ('Google', 'San Francisco');


INSERT INTO MANAGER_D VALUES ('Jade', 'Lakeview', 'Chicago');

INSERT INTO MANAGER_D VALUES ('Ron', '5th Ave', 'New York');

INSERT INTO MANAGER_D VALUES ('Zoe', 'Market St', 'San Francisco');


-- Answer 1:

SELECT EMP_NAME FROM EMP_D WHERE COMPANY_NAME = 'First Bank Corporation';

-- Answer 2:

SELECT EMP_NAME FROM EMP_D E JOIN COMPANY_D C ON E.COMPANY_NAME = C.COMPANY_NAME WHERE E.EMP_CITY = C.COMPANY_CITY;

-- Answer 3:

SELECT EMP_NAME FROM EMP_D E JOIN MANAGER_D M ON E.MANAGER_NAME = M.MANAGER_NAME WHERE E.EMP_CITY = M.CITY AND E.STREET = M.STREET;

-- Answer 4:

SELECT EMP_NAME FROM EMP_D E WHERE SALARY > (SELECT AVG(SALARY) FROM EMP_D WHERE COMPANY_NAME = E.COMPANY_NAME);

-- Answer 5:

SELECT COMPANY_NAME FROM (

  SELECT COMPANY_NAME, SUM(SALARY) AS PAYROLL FROM EMP_D GROUP BY COMPANY_NAME ORDER BY PAYROLL DESC

) WHERE ROWNUM = 1;
```

```
-- ====================================
-- SECTION E: EMP, PROJECT
-- ====================================
CREATE TABLE EMP_E (
  E_ID NUMBER,
  E_NAME VARCHAR2(50),
  EMP_DEPT VARCHAR2(50),
  PROJECT_ID NUMBER
);

CREATE TABLE PROJECT (
  PROJECT_ID NUMBER,
  PROJECT_NAME VARCHAR2(100),
  PROJECT_START_DATE DATE,
  PROJECT_END_DATE DATE
);

INSERT INTO EMP_E VALUES (101, 'Aliya', 'HR', 1);
INSERT INTO EMP_E VALUES (102, 'Sam', 'Finance', 1);
INSERT INTO EMP_E VALUES (103, 'Ron', 'IT', 2);
INSERT INTO EMP_E VALUES (104, 'Sara', 'IT', 3);
INSERT INTO EMP_E VALUES (105, 'Vicky', 'Finance', 3);

INSERT INTO PROJECT VALUES (1, 'Alpha', TO_DATE('2022-01-01','YYYY-MM-DD'), TO_DATE('2022-12-31','YYYY-MM-DD'));
INSERT INTO PROJECT VALUES (2, 'Beta', TO_DATE('2021-03-01','YYYY-MM-DD'), TO_DATE('2021-11-20','YYYY-MM-DD'));
INSERT INTO PROJECT VALUES (3, 'Gamma', TO_DATE('2020-01-01','YYYY-MM-DD'), TO_DATE('2022-05-10','YYYY-MM-DD'));

-- Answer 1:
```

```sql
SELECT PROJECT_ID, COUNT(DISTINCT EMP_DEPT) FROM EMP_E GROUP BY PROJECT_ID;

-- Answer 2:

SELECT COUNT(*) FROM PROJECT WHERE EXTRACT(YEAR FROM PROJECT_START_DATE) =
EXTRACT(YEAR FROM PROJECT_END_DATE);

-- Answer 3:

SELECT EMP_DEPT FROM EMP_E WHERE E_ID > 103;

-- Answer 4:

SELECT E_NAME FROM EMP_E WHERE LOWER(E_NAME) LIKE '%a';
```

```
- ======================================

-- SECTION F: EMP, MANAGER

-- ======================================

CREATE TABLE EMP_F (

  EMPID NUMBER,

  EMP_NAME VARCHAR2(50),

  EMP_SALARY NUMBER,

  CITY VARCHAR2(50),

  PROJECT VARCHAR2(50)

);


CREATE TABLE MANAGER_F (

  MANAGER_ID NUMBER,

  EMP_NAME VARCHAR2(50),

  PROJECT VARCHAR2(50)

);


INSERT INTO EMP_F VALUES (1, 'Ankit', 12000, 'Toronto', 'P1');

INSERT INTO EMP_F VALUES (2, 'Nora', 14000, 'Toronto', 'P1');

INSERT INTO EMP_F VALUES (3, 'Harsh', 11000, 'Montreal', 'P2');


INSERT INTO MANAGER_F VALUES (986, 'Raj', 'P1');

INSERT INTO MANAGER_F VALUES (321, 'Asha', 'P2');


-- Answer 1:

SELECT EMPID, EMP_NAME FROM EMP_F WHERE PROJECT IN (SELECT PROJECT FROM MANAGER_F
WHERE MANAGER_ID = 986);

-- Answer 2:

SELECT COUNT(*) FROM EMP_F WHERE PROJECT = 'P1';

-- Answer 3:
```

```sql
SELECT EMPID FROM EMP_F WHERE EMP_SALARY BETWEEN 9000 AND 15000;

-- Answer 4:

SELECT EMP_NAME FROM EMP_F WHERE CITY = 'Toronto' AND PROJECT IN (SELECT PROJECT FROM MANAGER_F WHERE MANAGER_ID = 321);

-- Answer 5:

SELECT MAX(EMP_SALARY), MIN(EMP_SALARY), AVG(EMP_SALARY) FROM EMP_F;
```

```
-- =====================================
-- SECTION G: CUSTOMER
-- =====================================
CREATE TABLE CUSTOMER (
  C_ID NUMBER,
  C_NAME VARCHAR2(50),
  C_CITY VARCHAR2(50)
);

INSERT INTO CUSTOMER VALUES (1, 'Bairistow', 'Delhi');
INSERT INTO CUSTOMER VALUES (2, 'Rahul', 'Chennai');
INSERT INTO CUSTOMER VALUES (12, 'Smith', 'New York');

-- Answer 1:
DELETE FROM CUSTOMER WHERE C_CITY = 'Chennai';
-- Answer 2:
ALTER TABLE CUSTOMER DROP COLUMN C_CITY;
-- Answer 3:
ALTER TABLE CUSTOMER ADD ADDRESS VARCHAR2(100);
-- Answer 4:
UPDATE CUSTOMER SET C_CITY = 'Oslo' WHERE C_NAME = 'Bairistow';
-- Answer 5:
SELECT * FROM CUSTOMER WHERE C_ID = 12;
-- Answer 6:
DROP TABLE CUSTOMER;
```

```
-- ====================================

-- SECTION H: All Types of Join Operations

-- ====================================

-- Inner Join

SELECT * FROM EMP_B INNER JOIN COMPANY ON EMP_B.COMPANY_NAME =
COMPANY.COMPANY_NAME;


-- Left Outer Join

SELECT * FROM EMP_B LEFT OUTER JOIN COMPANY ON EMP_B.COMPANY_NAME =
COMPANY.COMPANY_NAME;


-- Right Outer Join

SELECT * FROM EMP_B RIGHT OUTER JOIN COMPANY ON EMP_B.COMPANY_NAME =
COMPANY.COMPANY_NAME;


-- Full Outer Join

SELECT * FROM EMP_B FULL OUTER JOIN COMPANY ON EMP_B.COMPANY_NAME =
COMPANY.COMPANY_NAME;


-- Cross Join

SELECT * FROM EMP_B CROSS JOIN COMPANY;


-- Self Join (Using alias to join EMP_B to itself on MANAGER_NAME)

SELECT E1.EMP_NAME AS EMPLOYEE, E2.EMP_NAME AS MANAGED_BY

FROM EMP_B E1, EMP_B E2

WHERE E1.MANAGER_NAME = E2.EMP_NAME;
```

```
-- ======================================
-- SECTION I: Integrity Constraints
-- ======================================
-- Primary Key Constraint
CREATE TABLE CONSTRAINT_EXAMPLE1 (
  ID NUMBER PRIMARY KEY,
  NAME VARCHAR2(50)
);

-- Foreign Key Constraint
CREATE TABLE CONSTRAINT_EXAMPLE2 (
  REF_ID NUMBER,
  CONSTRAINT FK_REF_ID FOREIGN KEY (REF_ID) REFERENCES CONSTRAINT_EXAMPLE1(ID)
);

-- NOT NULL Constraint
CREATE TABLE CONSTRAINT_EXAMPLE3 (
  USER_ID NUMBER,
  USERNAME VARCHAR2(50) NOT NULL
);

-- UNIQUE Constraint
CREATE TABLE CONSTRAINT_EXAMPLE4 (
  EMAIL VARCHAR2(100) UNIQUE
);

-- CHECK Constraint
CREATE TABLE CONSTRAINT_EXAMPLE5 (
  AGE NUMBER CHECK (AGE >= 18)
);


-- DEFAULT Constraint
CREATE TABLE CONSTRAINT_EXAMPLE6 (
  STATUS VARCHAR2(10) DEFAULT 'ACTIVE'
);
```

```sql
-- ======================================
-- SECTION J: EMP(EID, EMP_NAME, SALARY, DESIGNATION)
-- ======================================
CREATE TABLE EMP_J (
  EID NUMBER PRIMARY KEY,
  EMP_NAME VARCHAR2(50),
  SALARY NUMBER,
  DESIGNATION VARCHAR2(30),
  DEPTNO NUMBER
);

INSERT INTO EMP_J VALUES (201, 'Ritesh', 3200, 'CLERK', 10);
INSERT INTO EMP_J VALUES (202, 'Aarti', 3400, 'MANAGER', 20);
INSERT INTO EMP_J VALUES (203, 'Simmi', 3100, 'CLERK', 10);
INSERT INTO EMP_J VALUES (204, 'Ajay', 3700, 'CLERK', 30);
INSERT INTO EMP_J VALUES (205, 'Sunny', 5000, 'MANAGER', 10);

-- 1. DISPLAY THE NAMES OF ALL EMPLOYEES WHO ARE WORKING IN DEPARTMENT NUMBER 10
SELECT EMP_NAME FROM EMP_J WHERE DEPTNO = 10;

-- 2. DISPLAY THE NAMES OF ALL EMPLOYEES WORKING AS CLERKS AND DRAWING A SALARY MORE THAN 3000
SELECT EMP_NAME FROM EMP_J WHERE DESIGNATION = 'CLERK' AND SALARY > 3000;

-- 3. DISPLAY THE NAMES OF EMPLOYEES WHOSE NAME IS EXACTLY FIVE CHARACTERS IN LENGTH
SELECT EMP_NAME FROM EMP_J WHERE LENGTH(EMP_NAME) = 5;

-- 4. WRITE QUERY THAT WILL RETURN THE TOP N RECORDS (USING ROWNUM, since Oracle 10g does not support TOP)
```

```
SELECT * FROM EMP_J WHERE ROWNUM <= 3;


-- 5. FIND ALL EMPLOYEES WHO EARN MORE THAN THE AVERAGE SALARY OF ALL EMPLOYEES

SELECT * FROM EMP_J WHERE SALARY > (SELECT AVG(SALARY) FROM EMP_J);
```

```
-- ====================================
-- SECTION K: EMPLOYEE(E_ID, E_NAME, SALARY, DEPARTMENT)
-- ====================================
CREATE TABLE EMP_K (
  E_ID NUMBER PRIMARY KEY,
  E_NAME VARCHAR2(50),
  SALARY NUMBER,
  DEPARTMENT VARCHAR2(50)
);

INSERT INTO EMP_K VALUES (1, 'Kunal', 3500, 'HR');
INSERT INTO EMP_K VALUES (2, 'Aarti', 4200, 'IT');
INSERT INTO EMP_K VALUES (3, 'Ravi', 3000, 'SALES');
INSERT INTO EMP_K VALUES (4, 'Divya', 2800, 'HR');
INSERT INTO EMP_K VALUES (5, 'Mona', 3600, 'IT');
INSERT INTO EMP_K VALUES (6, 'Lalit', 3900, 'HR');
INSERT INTO EMP_K VALUES (7, 'Sunil', 2500, 'IT');

-- 1. WRITE AN SQL QUERY TO FETCH THE FIRST 5 RECORDS FROM A TABLE
SELECT * FROM EMP_K WHERE ROWNUM <= 5;

-- 2. WRITE AN SQL QUERY TO FETCH THREE MIN SALARIES FROM A TABLE
SELECT DISTINCT SALARY FROM EMP_K ORDER BY SALARY ASC FETCH FIRST 3 ROWS ONLY;

-- 3. DISPLAY THE MAXIMUM SALARY BEING PAID TO CLERK
-- NOTE: As there is no DESIGNATION column, we assume CLERK is not included here
-- (Skip or modify schema to include DESIGNATION if needed)

-- 4. DISPLAY DEPT NUMBERS AND TOTAL NUMBER OF EMPLOYEES WITHIN EACH GROUP
SELECT DEPARTMENT, COUNT(*) AS TOTAL_EMPLOYEES FROM EMP_K GROUP BY DEPARTMENT;
```

```
-- 5. DISPLAY THE DEPARTMENT NUMBERS WITH MORE THAN THREE EMPLOYEES IN EACH DEPT

SELECT DEPARTMENT FROM EMP_K GROUP BY DEPARTMENT HAVING COUNT(*) > 3;
```

```
-- ======================================
-- SECTION L: EMPLOYEE(E_ID, E_NAME, SALARY, DEPARTMENT)
-- ======================================
CREATE TABLE EMP_L (
  E_ID NUMBER PRIMARY KEY,
  E_NAME VARCHAR2(50),
  SALARY NUMBER,
  DEPARTMENT VARCHAR2(50),
  DESIGNATION VARCHAR2(30)
);

INSERT INTO EMP_L VALUES (1, 'Neeta', 2800, 'HR', 'CLERK');
INSERT INTO EMP_L VALUES (2, 'Manoj', 3100, 'IT', 'MANAGER');
INSERT INTO EMP_L VALUES (3, 'Rajeev', 2700, 'SALES', 'CLERK');
INSERT INTO EMP_L VALUES (4, 'Sneha', 2900, 'HR', 'CLERK');
INSERT INTO EMP_L VALUES (5, 'Preeti', 3500, 'HR', 'CLERK');
INSERT INTO EMP_L VALUES (6, 'Vikram', 4000, 'IT', 'CLERK');
INSERT INTO EMP_L VALUES (7, 'Anu', 3600, 'SALES', 'CLERK');

-- 1. WRITE AN SQL QUERY TO FETCH THE FIRST 5 RECORDS FROM A TABLE
SELECT * FROM EMP_L WHERE ROWNUM <= 5;

-- 2. WRITE AN SQL QUERY TO FETCH THREE MIN SALARIES FROM A TABLE
SELECT DISTINCT SALARY FROM EMP_L ORDER BY SALARY ASC FETCH FIRST 3 ROWS ONLY;

-- 3. DISPLAY THE MAXIMUM SALARY BEING PAID TO CLERK
SELECT MAX(SALARY) FROM EMP_L WHERE DESIGNATION = 'CLERK';

-- 4. DISPLAY DEPT NUMBERS AND TOTAL NUMBER OF EMPLOYEES WITHIN EACH GROUP
SELECT DEPARTMENT, COUNT(*) AS TOTAL_EMPLOYEES FROM EMP_L GROUP BY DEPARTMENT;
```

```
-- 5. DISPLAY THE DEPARTMENT NUMBERS WITH MORE THAN THREE EMPLOYEES IN EACH DEPT

SELECT DEPARTMENT FROM EMP_L GROUP BY DEPARTMENT HAVING COUNT(*) > 3;
```

```
-- ========================================
-- SECTION M & N
-- ========================================
-- SECTION N: Products Table
CREATE TABLE Products (
  product_id INT PRIMARY KEY,
  product_name VARCHAR(100),
  category VARCHAR(50),
  unit_price DECIMAL(10, 2)
);

INSERT INTO Products (product_id, product_name, category, unit_price) VALUES
(101, 'Laptop', 'Electronics', 500.00),
(102, 'Smartphone', 'Electronics', 300.00),
(103, 'Headphones', 'Electronics', 30.00),
(104, 'Keyboard', 'Electronics', 20.00),
(105, 'Mouse', 'Electronics', 15.00);

-- Create Sales Table for Section M queries
CREATE TABLE Sales (
  sale_id INT PRIMARY KEY,
  product_id INT,
  quantity INT,
  total_price DECIMAL(10, 2),
  sale_date DATE,
  FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

INSERT INTO Sales VALUES (1, 101, 2, 1000.00, TO_DATE('2024-01-03', 'YYYY-MM-DD'));
INSERT INTO Sales VALUES (2, 102, 1, 300.00, TO_DATE('2024-01-03', 'YYYY-MM-DD'));
INSERT INTO Sales VALUES (3, 103, 3, 90.00, TO_DATE('2024-01-02', 'YYYY-MM-DD'));
```

```sql
INSERT INTO Sales VALUES (4, 104, 2, 40.00, TO_DATE('2024-01-01', 'YYYY-MM-DD'));
INSERT INTO Sales VALUES (5, 105, 1, 15.00, TO_DATE('2024-01-03', 'YYYY-MM-DD'));

-- 1. Retrieve the product_name and unit_price from the Products table
SELECT product_name, unit_price FROM Products;

-- 2. Filter the Products table to show only products in the 'Electronics' category
SELECT * FROM Products WHERE category = 'Electronics';

-- 3. Retrieve the sale_id and total_price from the Sales table for sales made on January 3, 2024
SELECT sale_id, total_price FROM Sales WHERE sale_date = TO_DATE('2024-01-03', 'YYYY-MM-DD');

-- 4. Calculate the total revenue generated from all sales in the Sales table
SELECT SUM(total_price) AS total_revenue FROM Sales;

-- 5. Count Sales Per Day from the Sales table
SELECT sale_date, COUNT(*) AS sales_count FROM Sales GROUP BY sale_date;

-- 6. Retrieve product_name and unit_price from the Products table with the Highest Unit Price
SELECT product_name, unit_price FROM Products WHERE unit_price = (SELECT MAX(unit_price) FROM Products);

-- 7. Retrieve the product_name and unit_price from the Products table, ordering the results by unit_price in descending order
SELECT product_name, unit_price FROM Products ORDER BY unit_price DESC;
```