```
-- Customers Table
CREATE TABLE Customers (
    Customer_ID NUMBER PRIMARY KEY,
    Name VARCHAR2(100),
    Email VARCHAR2(100) UNIQUE,
    Phone VARCHAR2(15)
);

-- Products Table
CREATE TABLE Products (
    Product_ID NUMBER PRIMARY KEY,
    Product_Name VARCHAR2(100),
    Price NUMBER(10, 2),
    Stock_Quantity NUMBER
);


-- Orders Table
CREATE TABLE Orders (
    Order_ID NUMBER PRIMARY KEY,
    Customer_ID NUMBER,
    Order_Date DATE,
    CONSTRAINT fk_customer
        FOREIGN KEY (Customer_ID)
        REFERENCES Customers(Customer_ID)
);

-- Order_Items Table (junction table for many-to-many)
CREATE TABLE Order_Items (
    Order_ID NUMBER,
    Product_ID NUMBER,
    Quantity NUMBER,
    Total_Amount NUMBER(10, 2),
    PRIMARY KEY (Order_ID, Product_ID),
    CONSTRAINT fk_order
        FOREIGN KEY (Order_ID)
        REFERENCES Orders(Order_ID),
```

```
        FOREIGN KEY (Product_ID)
        REFERENCES Products(Product_ID)
);



-- 1. Table Creation (for example purposes)
CREATE TABLE Employees (
    EmployeeID INTEGER PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100) DEFAULT 'not_provided@example.com',
    Department VARCHAR(50),
    Salary NUMERIC,
    JoinDate DATE,
    Status VARCHAR(20) DEFAULT 'Active'
);


-- 2. IN SERT INTO (with and without NULLs)
INSERT INTO Employees (EmployeeID, FirstName, LastName, Email,
VALUES (3, 'Alice', 'Johnson', NULL, 'IT', 65000, TO_DATE('2022

INSERT INTO Employees (EmployeeID, FirstName, LastName, Departm
VALUES (2, 'Jane', 'Smith', 'HR', 55000, TO_DATE('2022-02-15',

INSERT INTO Employees (EmployeeID, FirstName, LastName, Email,
VALUES (1, 'John', 'Doe', 'john.doe@example.com', 'Finance', 60

SELECT *
FROM Employees;

-- 3. UPDATE (e.g., setting NULL email to default manually)
UPDATE Employees
SET Email = 'not_provided@example.com'
WHERE Email IS NULL;
```

```
UPDATE Employees
SET Salary = Salary + 5000
WHERE Department = 'IT';

-- 4. DELETE (e.g., based on condition)
DELETE FROM Employees
WHERE Salary < 60000;

INSERT INTO Employees (EmployeeID, FirstName, LastName, Email,
SELECT 4, 'Mark', 'Taylor', 'mark.taylor@example.com', 'IT', 70
FROM DUAL;

-- 7. INSERT specific columns only
INSERT INTO Employees (EmployeeID, FirstName)
VALUES (5, 'Priya');

-- 9. ON DELETE CASCADE (example setup)
CREATE TABLE Projects (
    ProjectID INTEGER PRIMARY KEY,
    ProjectName VARCHAR(100),
    EmployeeID INTEGER,
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) C
);



-- Assigning a project
INSERT INTO Projects (ProjectID, ProjectName, EmployeeID)
VALUES (1, 'AI Chatbot', 1);

-- Now deleting Employee with ID 1 will also delete related pro
DELETE FROM Employees
WHERE EmployeeID = 1;

SELECT * FROM employees;

-- Select specific columns
SELECT firstname, lastname, salary
```

```
--Use WHERE to filter rows
SELECT * FROM employees
WHERE department = 'Finance';

--Use WHERE with AND/OR
SELECT * FROM employees
WHERE department = 'Finance' AND salary > 60000;

-- Use LIKE to match a pattern
SELECT * FROM employees
WHERE firstname LIKE 'A%';

--Use BETWEEN for range
SELECT * FROM employees
WHERE joindate BETWEEN TO_DATE('2022-01-01', 'YYYY-MM-DD') AND
ORDER BY joindate;

SELECT * FROM employees
ORDER BY salary DESC;


-- Use LIMIT to restrict output
SELECT * FROM (
  SELECT * FROM employees
  ORDER BY joindate DESC
)
WHERE ROWNUM <= 5;

--Use aliasing
SELECT firstname AS fname, lastname AS lname
FROM employees;

-- Use DISTINCT to remove duplicates
SELECT DISTINCT department
FROM employees;
```

```sql
-- Using =
SELECT * FROM employees
WHERE department = 'IT';

-- Using IN
SELECT * FROM employees
WHERE department IN ('IT', 'Finance');
```