-- Creating the Worker table

CREATE TABLE Worker (

WORKER\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT, -- Unique identifier for each worker

FIRST\_NAME CHAR(25), -- First name of the worker

LAST\_NAME CHAR(25), -- Last name of the worker

SALARY INT, -- Salary of the worker

JOINING\_DATE DATETIME, -- The date and time the worker joined

DEPARTMENT CHAR(25) -- Department in which the worker works

);

-- Inserting sample data into the Worker table

INSERT INTO Worker

(WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE, DEPARTMENT) VALUES

(1, 'Monika', 'Arora', 100000, '2014-02-20 09:00:00', 'HR'),

(2, 'Niharika', 'Verma', 80000, '2014-06-11 09:00:00', 'Admin'),

(3, 'Vishal', 'Singhal', 300000, '2014-02-20 09:00:00', 'HR'),

(4, 'Amitabh', 'Singh', 500000, '2014-02-20 09:00:00', 'Admin'),

(5, 'Vivek', 'Bhati', 500000, '2014-06-11 09:00:00', 'Admin'),

(6, 'Vipul', 'Diwan', 200000, '2014-06-11 09:00:00', 'Account'),

(7, 'Satish', 'Kumar', 75000, '2014-01-20 09:00:00', 'Account'),

(8, 'Geetika', 'Chauhan', 90000, '2014-04-11 09:00:00', 'Admin');

-- Creating the Bonus table

CREATE TABLE Bonus (

WORKER\_REF\_ID INT, -- References WORKER\_ID from the Worker table

BONUS\_AMOUNT INT, -- Bonus amount received by the worker

BONUS\_DATE DATETIME, -- Date when the bonus was given

FOREIGN KEY (WORKER\_REF\_ID) -- Establishing relationship with the Worker table

REFERENCES Worker(WORKER\_ID)

ON DELETE CASCADE -- If a worker is deleted, their bonus data is also removed

);

-- Inserting sample data into the Bonus table

INSERT INTO Bonus

(WORKER\_REF\_ID, BONUS\_AMOUNT, BONUS\_DATE) VALUES

(1, 5000, '2016-02-20 00:00:00'),

(2, 3000, '2016-06-11 00:00:00'),

(3, 4000, '2016-02-20 00:00:00'),

(1, 4500, '2016-02-20 00:00:00'),

(2, 3500, '2016-06-11 00:00:00');

-- Creating the Title table

CREATE TABLE Title (

WORKER\_REF\_ID INT, -- References WORKER\_ID from the Worker table

WORKER\_TITLE CHAR(25), -- Title/Designation of the worker

AFFECTED\_FROM DATETIME, -- The date when the title became effective

FOREIGN KEY (WORKER\_REF\_ID) -- Establishing relationship with the Worker table

REFERENCES Worker(WORKER\_ID)

ON DELETE CASCADE -- If a worker is deleted, their title data is also removed

);

-- Inserting sample data into the Title table

INSERT INTO Title

(WORKER\_REF\_ID, WORKER\_TITLE, AFFECTED\_FROM) VALUES

(1, 'Manager', '2016-02-20 00:00:00'),

(2, 'Executive', '2016-06-11 00:00:00'),

(8, 'Executive', '2016-06-11 00:00:00'),

(5, 'Manager', '2016-06-11 00:00:00'),

(4, 'Asst. Manager', '2016-06-11 00:00:00'),

(7, 'Executive', '2016-06-11 00:00:00'),

(6, 'Lead', '2016-06-11 00:00:00'),

(3, 'Lead', '2016-06-11 00:00:00');

-- Query 1: Fetch all workers, ordered by FIRST\_NAME in ascending order

SELECT \* FROM Worker

ORDER BY FIRST\_NAME ASC;

-- Query 2: Fetch all workers, ordered by FIRST\_NAME in ascending order

-- and DEPARTMENT in descending order when FIRST\_NAME is the same

SELECT \* FROM Worker

ORDER BY FIRST\_NAME ASC, DEPARTMENT DESC;

-- Query 3: Fetch details of workers excluding those with FIRST\_NAME 'Vipul' or 'Satish'

SELECT \* FROM Worker

WHERE FIRST\_NAME NOT IN ('Vipul', 'Satish');

-- Query 4: Fetch details of workers whose FIRST\_NAME ends with 'h' and contains exactly six characters

SELECT \* FROM Worker

WHERE FIRST\_NAME LIKE '\_\_\_\_\_h' AND LENGTH(FIRST\_NAME) = 6;

-- Query 5: Count the number of employees working in the 'Admin' department

SELECT COUNT(\*) AS Admin\_Employee\_Count

FROM Worker

WHERE DEPARTMENT = 'Admin';

-- Query 6: Fetch worker names (FIRST\_NAME, LAST\_NAME) with salaries between 50,000 and 100,000

SELECT FIRST\_NAME, LAST\_NAME

FROM Worker

WHERE SALARY >= 50000 AND SALARY <= 100000;

-- Query 7: Fetch the number of workers in each department and sort the results in descending order

SELECT DEPARTMENT, COUNT(\*) AS Worker\_Count

FROM Worker

GROUP BY DEPARTMENT

ORDER BY Worker\_Count DESC;

-- Query: Fetch departments that have less than three employees

SELECT DEPARTMENT, COUNT(\*) AS Worker\_Count

FROM Worker

GROUP BY DEPARTMENT

HAVING COUNT(\*) < 3;