Example 04:
Database Management System

Consider the following database with four relations.

employee(employee-name, street, city) works(employee-name, company-name, salary)	company(company-name, city) manages(employee-name, manager-name)

1. Insert the following data into those relations

Employee		
employee-name	Street	city
Arif	51 upashahar	Rajshahi
Sumon	52 east	Moynamati
Sagor	Neemgachhi	Sirajgong
Abdul	Binodpur	Rajshahi
Himesh	Nazrul avenue	Dhaka
Amirul	Chawk bazar	Sylhet
Sajib	99 north	Chittagong

company	
compny-name	city
Agrani	Rajshahi
Sonali	Sylhet
Janata	Dhaka

Works		
employee-name	company-name	salary
Sumon	Agrani	12000
Abdul	Sonali	13000
Himesh	Agrani	6000
Amirul	Sonali	20000
Sagor	Sonali	8000
Arif	Janata	13000
Sajib	Janata	9000

Manages	
employee-name	manager-name
Amirul	Amirul
Abdul	Amirul
Sagor	Amirul
Sumon	Sumon
Himesh	Sumon
Arif	Arif
Sajib	Arif

Ans:

```
 CREATE TABLE Employee (
    employee_name VARCHAR(100) PRIMARY KEY,
    street VARCHAR(255),
    city VARCHAR(100)
);

CREATE TABLE Company (
    company_name VARCHAR(100) PRIMARY KEY,
    city VARCHAR(100)
);

CREATE TABLE Works (
    employee_name VARCHAR(100) PRIMARY KEY,  -- Primary Key: employee_name
    company_name VARCHAR(100),
    salary DECIMAL(10, 2)
);

CREATE TABLE Manages (
    employee_name VARCHAR(100) PRIMARY KEY,
    manager_name VARCHAR(100)

);
```

/*ALTER TABLE Employee
ADD COLUMN created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP; */ for updating any information of the table.

```
INSERT INTO Employee (employee_name, street, city)
VALUES
('Arif', '51 upashahar', 'Rajshahi'),
('Sumon', '52 east', 'Moynamati'),
('Sagor', 'Neemgachhi', 'Sirajgong'),
('Abdul', 'Binodpur', 'Rajshahi'),
('Himesh', 'Nazrul avenue', 'Dhaka'),
('Amirul', 'Chawk bazar', 'Sylhet'),
('Sajib', '99 north', 'Chittagong');

INSERT INTO Company (company_name, city)
VALUES
('Agrani', 'Rajshahi'),
('Sonali', 'Sylhet'),
('Janata', 'Dhaka');

INSERT INTO Works (employee_name, company_name, salary)
VALUES
('Sumon', 'Agrani', 12000),
('Abdul', 'Sonali', 13000),
('Himesh', 'Agrani', 6000),
('Amirul', 'Sonali', 20000),
('Sagor', 'Sonali', 8000),
('Arif', 'Janata', 13000),
('Sajib', 'Janata', 9000);

INSERT INTO Manages (employee_name, manager_name)
VALUES
('Amirul', 'Amirul'),
('Abdul', 'Amirul'),
('Sagor', 'Amirul'),
('Sumon', 'Sumon'),
('Himesh', 'Sumon'),
('Arif', 'Arif'),
('Sajib', 'Arif');
```

2. Find the names of all employee who work for “Sonali”.

```
SELECT employee_name
FROM works
WHERE company_name = 'sonali';
```

3. **Find the names, streets and cities residence of all employees who work for "Agrani".**

```
SELECT e.employee_name, e.street, e.city
FROM employee e
JOIN works w ON e.employee_name = w.employee_name
WHERE company_name = 'Agrani';
```

4. **Find the names, streets and cities residence of all employees who work for "Sonali" and earn more than 1,20,000 per annum.**

```
SELECT e.employee_name, e.street, e.city
FROM employee e
JOIN works w ON e.employee_name = w.employee_name
WHERE w.company_name = 'Agrani' AND w.salary > 10000;
```

5. **Find all employees in the database who live in the same cities as the companies for which they work.**

```
SELECT e.employee_name
FROM employee e
JOIN works w ON e.employee_name = w.employee_name
JOIN company c ON w.company_name = c.company_name
WHERE e.city = c.city;
```

6. **Find all employees in the database who live in the same cities and on the same streets as do their managers.**

```
SELECT e.employee_name
FROM employee e
JOIN manages m ON e.employee_name = m.employee_name
JOIN employee m_emp ON m.manager_name = m_emp.employee_name
WHERE e.city = m_emp.city AND e.street = m_emp.street;
```

7. **Find all employees in the database who do not work for "Sonali" Bank.**

```
SELECT w.employee_name
FROM works w
WHERE company_name != 'sonali';
```

8. **Find all employees in the database who earn more than each employee of "Janata" Bank**

```
SELECT w.employee_name
FROM works w
```

```
WHERE w.salary > ALL( SELECT w.salary
FROM works w WHERE company_name = 'janata');
```

Or,

```
SELECT *
FROM works
WHERE works.salary>(
SELECT MAX(works.salary)
FROM works
WHERE works.company_name = 'Janata' )
```

9. Find all employees who earn more than the average salary of all employees of their companies.

```
SELECT employee_name
FROM works w1
WHERE salary > (SELECT AVG(salary) FROM works w2 WHERE w1.company_name =
w2.company_name);
```

10. Find the company that has the most employees.

```
SELECT company_name
FROM works w
GROUP BY company_name /* The GROUP BY clause groups the rows that have
                                    the same company_name.*/
ORDER BY COUNT(employee_name) DESC LIMIT 1;/*DESC means in descending
                order, so companies with more employees will appear first*?
/*limit 1:
This limits the result to only one row. Since the result is already ordered in descending
order, the first row will represent the company with the most employees.*/
```

11. Find the company that has the smallest payroll. //(jar beton kom)

```
SELECT company_name
FROM works w
GROUP BY company_name
ORDER by SUM(w.salary) ASC LIMIT 1;
```

Or.

```
SELECT company_name, SUM(salary) AS total_salary
FROM works
GROUP BY company_name
ORDER BY total_salary ASC
LIMIT 1;
```

12. Find those companies whose employees earn a higher salary, on average, than the average salary at "Agrani" Bank.

```
WITH AgraniAvg AS (
   SELECT AVG(salary) AS avg_salary
   FROM works
   WHERE company_name = 'Agrani'
)
SELECT w.company_name
FROM works w
GROUP BY w.company_name
HAVING AVG(w.salary) > (SELECT avg_salary FROM AgraniAvg);
```

Or

```
SELECT company_name
FROM works
GROUP BY company_name
HAVING AVG(salary) > (SELECT AVG(salary) FROM works WHERE
company_name = 'Agrani');
```

13. Modify the database so that "Arif" now lives in Natore.

```
UPDATE employee
SET city = 'nator'
WHERE employee_name = 'Arif';
```

14. Give all employees of "Agrani" Bank 1 10 percent raise.(beton barche**)**

```
UPDATE works
SET salary = salary * 1.10
WHERE company_name = 'Agrani';
```

15. Give all managers of "Agrani" Bank a 10 percent salary raise.

```
UPDATE works
SET salary = salary * 1.10
WHERE employee_name IN (SELECT m.manager_name FROM manages m JOIN works
w ON m.manager_name = w.employee_name WHERE w.company_name = 'Agrani');
```

Or,

```
UPDATE works
SET salary = salary-200
WHERE employee_name IN (
   SELECT manager_name
   FROM (
```

```
    SELECT DISTINCT manager_name
    FROM manages
    NATURAL JOIN works
    WHERE company_name = 'Agrani'
  ) AS temp_table
);
```

16. Give all managers a 10 percent salary raise unless salary becomes greater than 19,000; in such cases, give only a 3 percent salary raise.

```
UPDATE works
SET salary = CASE
        WHEN salary * 1.10 > 19000 THEN salary * 1.03
        ELSE salary * 1.10
END
WHERE employee_name IN (SELECT manager_name FROM manages);
```

17. Delete all tuples in the works relation for employees of "Janata" Bank.

```
DELETE FROM works
WHERE company_name = 'Janata';
```

18. Define a view consisting of manager-name and average salary of all employees who work for that manager. Now try to modify that view.

```
CREATE VIEW manager_avg_salary AS
SELECT m.manager_name, AVG(w.salary) AS avg_salary
FROM works w
JOIN manages m ON w.employee_name = m.employee_name
GROUP BY m.manager_name;
```

And:

```
-- Example: Trying to update the salary in the view (may not work, depending on the DBMS)
UPDATE manager_avg_salary
SET avg_salary = avg_salary * 1.10;
```