**6-4: Self-Joins and HIerarchical Queries**

D.suryakiran

192325057

CSA0560

-- Create the employees table

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

hire\_date DATE,

salary DECIMAL(10,2),

department\_id INT,

manager\_id INT

);

-- Insert data into the employees table (employees and their managers)

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (1, 'Lex', 'De Haan', '1990-01-01', 17000, 90, NULL); -- De Haan (Manager)

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (2, 'Nancy', 'Greenberg', '1994-01-01', 15000, 100, 1); -- Employee under De Haan

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (3, 'Daniel', 'Faviet', '1995-02-01', 12000, 90, 2); -- Employee under Nancy

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (4, 'John', 'Smith', '1992-05-05', 11000, 90, 1); -- Employee under De Haan

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (5, 'Jane', 'Doe', '1996-08-15', 10000, 100, 2); -- Employee under Nancy

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (6, 'King', 'Brown', '1989-03-03', 25000, 50, NULL); -- CEO with no manager

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (7, 'Mark', 'Taylor', '1997-07-07', 9000, 100, 2); -- Employee under Nancy

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (8, 'Alice', 'White', '1998-01-10', 10500, 100, 1); -- Employee under De Haan

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, salary, department\_id, manager\_id)

VALUES (9, 'Henry', 'Miller', '1997-04-20', 9500, 50, 6); -- Employee under King

-- Create the departments table

CREATE TABLE departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(50)

);

-- Insert data into the departments table

INSERT INTO departments (department\_id, department\_name)

VALUES (50, 'Executive');

INSERT INTO departments (department\_id, department\_name)

VALUES (90, 'IT');

INSERT INTO departments (department\_id, department\_name)

VALUES (100, 'Sales');

**1. Display the employee’s last name and employee number along with the manager’s last name and manager number**

SELECT e.last\_name AS Employee, e.employee\_id AS Emp#,

m.last\_name AS Manager, m.employee\_id AS Mgr#

FROM employees e

LEFT JOIN employees m ON e.manager\_id = m.employee\_id;

**2. Display all employees and their managers, even if the employee does not have a manager, sorted by employee last name**

SELECT e.last\_name AS Employee, e.employee\_id AS Emp#,

m.last\_name AS Manager, m.employee\_id AS Mgr#

FROM employees e

LEFT JOIN employees m ON e.manager\_id = m.employee\_id

ORDER BY e.last\_name;

**3. Display employees who were hired before their managers**

SELECT e.last\_name AS Employee, e.hire\_date AS "Emp Hired",

m.last\_name AS Manager, m.hire\_date AS "Mgr Hired"

FROM employees e

JOIN employees m ON e.manager\_id = m.employee\_id

WHERE e.hire\_date < m.hire\_date;

**4. Show the hierarchy for Lex De Haan's department**

SELECT e.last\_name, e.salary, e.department\_id

FROM employees e

START WITH e.last\_name = 'De Haan'

CONNECT BY PRIOR e.employee\_id = e.manager\_id;

**5. Corrected version of the START WITH and CONNECT BY statement for hierarchy starting with 'King'**

SELECT last\_name, department\_id, salary

FROM employees

START WITH last\_name = 'King'

CONNECT BY PRIOR employee\_id = manager\_id;

**6. Display the organization chart for the entire employee table with hierarchy**

SELECT LPAD('-', 2 \* (LEVEL - 1), '-') || last\_name AS Org\_Chart, employee\_id

FROM employees

START WITH manager\_id IS NULL

CONNECT BY PRIOR employee\_id = manager\_id;

**7. Exclude De Haan and his hierarchy in the organizational chart**

SELECT LPAD('-', 2 \* (LEVEL - 1), '-') || last\_name AS Org\_Chart, employee\_id

FROM employees

START WITH manager\_id IS NULL

CONNECT BY PRIOR employee\_id = manager\_id

AND employee\_id NOT IN (

SELECT employee\_id

FROM employees

START WITH last\_name = 'De Haan'

CONNECT BY PRIOR employee\_id = manager\_id

);