Recursive Queries in SQL

- An example of a **recursive relationship** between tuples of the same type is the relationship between an employee and a supervisor.
- This relationship is described by the foreign key Super_ssn of the EMPLOYEE relation
- An example of a **recursive operation** is to retrieve all supervisees of a supervisory employee e at all levels—that is, all employees e' directly supervised by e, all employees e'' directly supervised by each employee e', all employees e''' directly supervised by each employee e'', and so on. Thus the CEO would have each employee in the company as a supervisee in the resulting table. Example shows such table SUP_EMP with 2 columns (Supervisor,Supervisee(any level)):

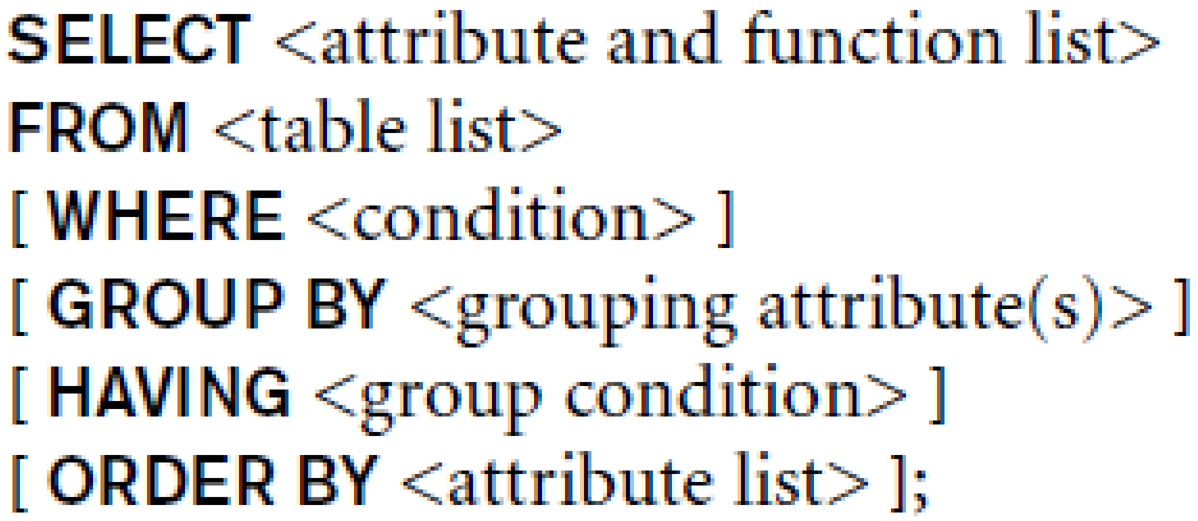
Slide Credits: Ramez Elmasri and Shamkant B. Navathe

An EXAMPLE of RECURSIVE Query

-
  ```
  Q29:   WITH RECURSIVE SUP_EMP (SupSsn, EmpSsn) AS
            SELECT      SupervisorSsn, Ssn
             FROM EMPLOYEE
              UNION
             SELECT      E.Ssn, S.SupSsn
             FROM EMPLOYEE AS E, SUP_EMP AS S
             WHERE       E.SupervisorSsn = S.EmpSsn)
             SELECT                *
             FROM SUP_EMP;
  ```
- The above query starts with an empty SUP_EMP and successively builds SUP_EMP table by computing immediate supervisees first, then second level supervisees, etc. until a **fixed point** is reached and no more supervisees can be added

Slide Credits: Ramez Elmasri and Shamkant B. Navathe

EXPANDED Block Structure of SQL Queries



```
SELECT <attribute and function list>
FROM <table list>
[ WHERE <condition> ]
[ GROUP BY <grouping attribute(s)> ]
[ HAVING <group condition> ]
[ ORDER BY <attribute list> ];
```

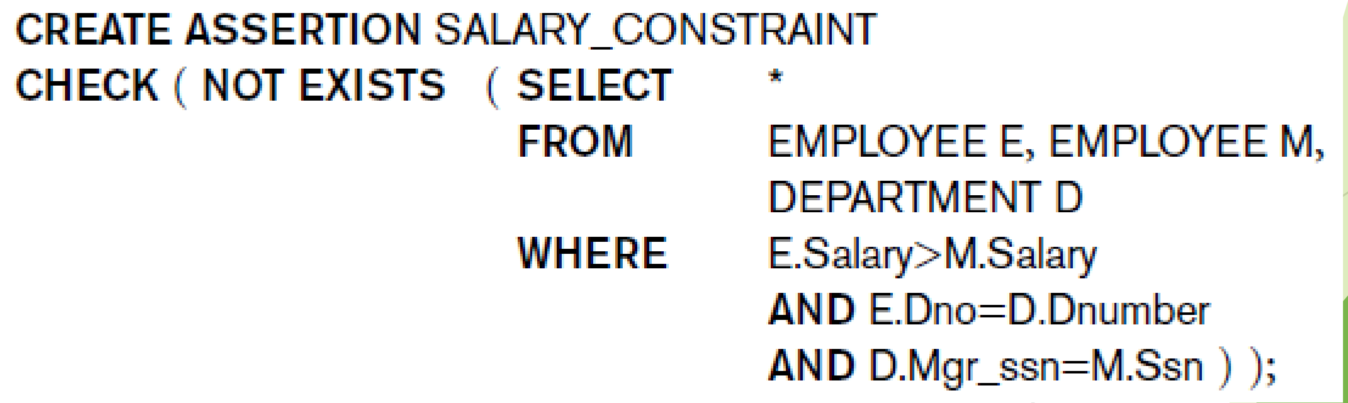
Specifying Constraints as Assertions and Actions as Triggers

- Semantic Constraints: The following are beyond the scope of the EER and relational model
- **`CREATE ASSERTION`**
  - Specify additional types of constraints outside scope of built-in relational model constraints
- **`CREATE TRIGGER`**
  - Specify automatic actions that database system will perform when certain events and conditions occur

Slide Credits: Ramez Elmasri and Shamkant B. Navathe

Specifying General Constraints as Assertions in SQL

- **`CREATE ASSERTION`**
  - Specify a query that selects any tuples that violate the desired condition
  - Use only in cases where it goes beyond a simple `CHECK` which applies to individual attributes and domains



```
CREATE ASSERTION SALARY_CONSTRAINT
CHECK ( NOT EXISTS   ( SELECT     *
                       FROM       EMPLOYEE E, EMPLOYEE M,
                                  DEPARTMENT D
                       WHERE      E.Salary>M.Salary
                                  AND E.Dno=D.Dnumber
                                  AND D.Mgr_ssn=M.Ssn ) );
```

Slide Credits: Ramez Elmasri and Shamkant B. Navathe

Introduction to Triggers in SQL

- `CREATE TRIGGER` statement
  - Used to monitor the database
- Typical trigger has three components which make it a rule for an “active database “ (more on active databases) :
  - **Event(s)**
  - **Condition**
  - **Action**

Slide Credits: Ramez Elmasri and Shamkant B. Navathe

USE OF TRIGGERS

- AN EXAMPLE with standard Syntax.(Note : other SQL implementations like PostgreSQL use a different syntax.)

```
R5:
CREATE TRIGGER SALARY_VIOLATION
BEFORE INSERT OR UPDATE OF Salary, Supervisor_ssn ON EMPLOYEE

FOR EACH ROW
WHEN (NEW.SALARY > ( SELECT Salary FROM EMPLOYEE
                     WHERE Ssn = NEW. Supervisor_Ssn))
INFORM_SUPERVISOR (NEW.Supervisor.Ssn, New.Ssn)
```

Sample Example: http://www.mysqltutorial.org/create-the-first-trigger-in-mysql.aspx

Slide Credits: Ramez Elmasri and Shamkant B. Navathe