Fundamentals of Database Systems

Chapter 5. More SQL: Complex Queries, Triggers, Views, and Schema Modification

Myungsun Kim

Information_Security@The_University_of_Suwon

2017, Fall

Contents

Note

Our main focus is §5.1; thus, study for the remaining sections from §5.2 to §5.4 on your own

§5.1 More Complex SQL Retrieval Queries

§5.1.1 Comparisons Involving NULL & Three-Valued Logic

- Three meanings of NULL: Unknown value, Unavailable value, Unavailable attribute
- Three-Valued Logic

AND	T	F	⊥
T	T	F	⊥
F	F	F	F
⊥	⊥	F	⊥

OR	T	F	⊥
T	T	T	T
F	T	F	⊥
⊥	T	⊥	⊥

NOT	
T	F
F	T
⊥	⊥

- Two NULL comparison operators: `IS` and `IS NOT`
- Query #18. Retrieve the names of all employees who do not have supervisors

⇒
```
SELECT Fname FROM EMPLOYEE
 WHERE Super_ssn IS NULL
```

§5.1 More Complex SQL Retrieval Queries

§5.1.2 Nested Queries, Tuples, and Set/Multiset Comparisons

- Nested Query

$$\underbrace{\texttt{SELECT-FROM-WHERE}}_{\text{outer query}}(\underbrace{\texttt{SELECT-FROM-WHERE}}_{\text{inner query}}(\cdots))$$

- In SQL: `IN`, `ANY`, and `ALL` operators

The `IN` Operator

- Syntax:

```
SELECT-FROM-WHERE (atttibute_list) IN
                  (SELECT atttibute_list FROM-WHERE);
```

- Inner queries return a table but not a relation
- `IN` is equivalent to `=ANY`

§5.1 More Complex SQL Retrieval Queries

The IN Operator (continued)

- Example: Retrieve the SSNs of all employees who have the same project and working hours as John Smith

⇒
```
SELECT DISTINCT Essn
FROM   WORKS_ON
WHERE  (Pno, Hours) IN
       (SELECT W.Pno, W.Hours
        FROM   WORKS_ON W, EMPLOYEE E
        WHERE  E.Fname="John" AND E.Lname="Smith"
        AND    W.Essn=E.Ssn);
```

§5.1 More Complex SQL Retrieval Queries

The ANY Operator

- Syntax:

```
SELECT-FROM-WHERE (atttibute) [=|>|>=|<|<=|<>]ANY
                  (SELECT atttibute FROM-WHERE);
```

- Example. Find all female employees such that there exist a male worker whose salary is less than her salary

⇒
```
SELECT DISTINCT Ssn
FROM   EMPLOYEE
WHERE  Sex='F'
  AND  Salary >ANY (SELECT Salary
                    FROM   EMPLOYEE
                    WHERE  Sex='M');
```

§5.1 More Complex SQL Retrieval Queries

The ALL Operator

- Syntax:

```
SELECT-FROM-WHERE (atttibute) [=|>|>=|<|<=|<>]ALL
                  (SELECT atttibute FROM-WHERE);
```

- Example. Find the names of employees whose salary is greater than or equal to the salary of all employees in the "Research" department

```
⇒ SELECT DISTINCT Fname, Lname
  FROM   EMPLOYEE
  WHERE  Salary >=All
        (SELECT E.Salary
         FROM   EMPLOYEE E, DEPARTMENT D
         WHERE  D.Dname="Research"
           AND  E.Dno=D.Dnumber);
```

§5.1 More Complex SQL Retrieval Queries

§5.1.3 Correlated Nested Queries

- Correlated queries

```
SELECT t.col1 FROM T1 t
WHERE t.col1 IN (SELECT s.col1 FROM T2 s
                 WHERE t.col1=s.col1 AND ...);
```

- A condition in the `WHERE` clause of a nested query references some attribute of a relation declared in the outer query
- To avoid potential errors and ambiguities,

$\Rightarrow$ (If possible) Trying to express as a single block query

§5.1 More Complex SQL Retrieval Queries

§5.1.3 Correlated Nested Queries

- Query #16. Find the name of each employee who has a dependent with the same first name and is the same sex as the employee

Correlated query

```
SELECT E.Fname,E.Lname
FROM   EMPLOYEE E
WHERE  E.Ssn IN
      (SELECT D.Essn
       FROM   DEPENDENT D
       WHERE  E.Fname=D.Dependent_name
         AND  E.Sex=D.Sex);
```

Single-block query

```
SELECT E.Fname,E.Lname
FROM   EMPLOYEE E,DEPENDENT D
WHERE  E.Ssn=D.Essn
  AND  E.Sex=D.Sex
  AND  E.Fname=
             D.Dependent_name;
```

✎ Two queries give the same result; but the second one is a better choice

§5.1 More Complex SQL Retrieval Queries

§5.1.4 The EXISTS and UNIQUE Functions in SQL

1. The EXISTS/NOT EXISTS Function
   - To check if the result of a correlated nested query is empty or not
   - Syntax
     `Boolean EXISTS(SELECT-FROM-WHERE)`
   - Return TRUE if there is at least one tuple in the result of nested SELECT-FROM-WHERE query; otherwise return FALSE
2. The UNIQUE Function
   - Syntax
     `Boolean UNIQUE(SELECT-FROM-WHERE)`
   - Return TRUE if there are no duplicates in the result of nested SELECT-FROM-WHERE query; otherwise return FALSE

§5.1 More Complex SQL Retrieval Queries

The EXISTS Function

- Query #16B. Find the name of each employee who has a dependent with the same first name and is the same sex as the employee

⇒
```
SELECT E.Fname,E.Lname
FROM   EMPLOYEE E
WHERE  EXISTS
      (SELECT *
       FROM   DEPENDENT D
       WHERE  E.Ssn=D.Essn
         AND  E.Fname=D.Dependent_name
         AND  E.Sex=D.Sex);
```

§5.1 More Complex SQL Retrieval Queries

The NOT EXISTS Function

- Query #6. Find the names of employees who have no dependents

```
⇒ SELECT E.Fname,E.Lname
  FROM   EMPLOYEE E
  WHERE  NOT EXISTS
         (SELECT *
          FROM   DEPENDENT D
          WHERE  E.Ssn=D.Essn);
```

§5.1 More Complex SQL Retrieval Queries

§5.1.5 Explicit Sets and Renaming of Attributes in SQL

- Use an explicit set of values in the WHERE clause

✎ Explicit sets in SQL should be enclosed in parentheses

- Query #17. Find the SSNs of all employees whose work on project numbers 1, 2, or, 3

⇒
```
SELECT DISTINCT Essn
FROM   WORKS_ON
WHERE  Pno IN (1, 2, 3);
```

§5.1 More Complex SQL Retrieval Queries

§5.1.6 Joined Tables in SQL and Outer Joins

- Joined table (or relation)
  - Supports the use of a table resulting from a join operation in the FROM clause of a query
  - Permissible join operations:
    JOIN, NATURAL JOIN, LEFT/RIGHT OUTER JOIN
- Query #1A. Find the name and address of every employee who works for the "Research" department

⇒
```
SELECT Fname, Lname, Address
FROM   (EMPLOYEE JOIN DEPARTMENT ON Dno=Dnumber)
WHERE  Dname="Research";
```

§5.1 More Complex SQL Retrieval Queries

§5.1.7 Aggregate Functions in SQL

- Aggregate functions
  - Summarize information from multiple tuples into a single-tuple summary
  - Buit-in aggregate functions: `COUNT`,`SUM`,`AVG`,`MIN/MAX`
- Query #22. Find the number of employees in the "Research" department

⇒
```
SELECT COUNT(*)
FROM   EMPLOYEE E, DEPARTMENT D
WHERE  D.Dname="Research" AND D.Number=E.Dno;
```

§5.1 More Complex SQL Retrieval Queries

§5.1.7 Aggregate Functions in SQL

- Query #5. Find the names of employees with two or more dependents

⇒
```
SELECT Fname, Lname
FROM   EMPLOYEE
WHERE  (SELECT COUNT(*)
        FROM   DEPENDENT
        WHERE  Ssn=Essn) >= 2;
```

- Query #20. Find the sum of the salaries of all employees, the maximum salary, and the average salary

⇒
```
SELECT SUM(Salary), MAX(Salary), AVG(Salary)
FROM   EMPLOYEE;
```

§5.1 More Complex SQL Retrieval Queries

§5.1.8 Grouping: The `GROUP BY` and `HAVING` Clauses

- Motivations
  - Apply the aggregate functions to **subgroups** of tuples in a relation
  - Here, all tuples in each group has the same values on some attributes
- Example
  - Find the average salary of employees in each department
- Process
  1. **Partition** a relation into non-overlapping groups of tuples
  2. Each group includes the tuples with the same value of some attributes–**grouping attributes**
  3. Apply the aggregate functions to each group

✎ Grouping attributes should appear in both the `GROUP BY` clause and the `SELECT` clause

§5.1 More Complex SQL Retrieval Queries

§5.1.8 Grouping: The GROUP BY and HAVING Clauses

- Query #24. For each department, retrieve the department number, the number of employees in the department, and their average salary

⇒
```
SELECT   Dno,COUNT(*),AVG(Salary)
FROM     EMPLOYEE
GROUP BY Dno;
```

- Query #25. For each project, retrieve the project number, the project name, and the # of employees working on that project

⇒
```
SELECT   Pnumber, Pname, COUNT(*)
FROM     PROJECT, WORKS_ON
WHERE    Pnumber=Pno
GROUP BY Pnumber,Pname;
```

§5.1 More Complex SQL Retrieval Queries

§5.1.8 Grouping: The GROUP BY and HAVING Clauses

- The HAVING clause
  - Provide a condition on the summary information regarding the group of tuples associated with each value of the grouping attributes
- Query #26. For each project on which more than two employees work, retrieve the project number, the project name, and the # of employees who work on that project

⇒
```
SELECT   Pnumber, Pname, COUNT(*)
FROM     PROJECT, WORKS_ON
WHERE    Pnumber=Pno
GROUP BY Pnumber,Pname
HAVING   COUNT(*) > 2;
```

§5.2 Specifying Constraints as Assertions and Actions as Triggers

Study yourself

[Image] Retrieved from
https://mnmmariam.files.wordpress.com/2015/01/a-cartoon-girl-reading-books.jpg

§5.3 Views in SQL

Study yourself

§5.4 Schema Change Statements in SQL

Study yourself

Wrap-up & Questions

정리

- Several complex SQL statements
- Declaring constraints
- View and Update schemas

********** Thanks & Question? **********

A state of RDB schema: COMPANY

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse