ENSF 608: SQL

Dr. Emily Marasco

Fall 2021

Textbook: Fundamentals of Database Systems, 7th Ed., Elmasri & Navathe

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

More Complex SQL Retrieval Queries

- Additional features allow users to specify more complex retrievals from database:
  - Nested queries, joined tables, and outer joins (in the FROM clause), aggregate functions, and grouping

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Comparisons Involving NULL and Three-Valued Logic (1 of 3)

- Meanings of `NULL`
  - **Unknown value:** value exists and is not known, or it is not known if the value exists
  - **Unavailable or withheld value:** value exists but is purposely withheld
  - **Not applicable attribute:** attribute does not apply or is undefined for this particular tuple
- Each individual `NULL` value considered to be different from every other `NULL` value
- **NULL = NULL** comparison is avoided

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Comparisons Involving NULL and Three-Valued Logic (2 of 3)

Table 7.1 Logical Connectives in Three-Valued Logic

(a)	**AND**	TRUE	FALSE	UNKNOWN
	TRUE	TRUE	FALSE	UNKNOWN
	FALSE	FALSE	FALSE	FALSE
	UNKNOWN	UNKNOWN	FALSE	UNKNOWN
(b)	**OR**	TRUE	FALSE	UNKNOWN
	TRUE	TRUE	TRUE	TRUE
	FALSE	TRUE	FALSE	UNKNOWN
	UNKNOWN	TRUE	UNKNOWN	UNKNOWN
(C)	**NOT**			
	TRUE	FALSE		
	FALSE	TRUE		
	UNKNOWN	UNKNOWN		

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Comparisons Involving NULL and Three-Valued Logic (3 of 3)

- SQL allows queries that check whether an attribute value is `NULL`
- `IS` or `IS NOT NULL`

Query 18. Retrieve the names of all employees who do not have supervisors.

```
Q18:    SELECT    Fname, Lname
        FROM      EMPLOYEE
        WHERE     Super_ssn IS NULL;
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Nested Queries, Tuples, and Set/Multiset Comparisons

- **Nested queries**
  - Complete select-from-where blocks within WHERE clause of another query
  - **Outer query and nested subqueries**
- Comparison operator `IN`
  - Compares value *v* with a set (or multiset) of values *V*
  - Evaluates to `TRUE` if *v* is one of the elements in *V*

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Nested Queries (1 of 4)

```
Q4A:    SELECT    DISTINCT Pnumber
        FROM      PROJECT
        WHERE     Pnumber IN
                  ( SELECT      Pnumber
                    FROM        PROJECT, DEPARTMENT, EMPLOYEE
                    WHERE       Dnum=Dnumber AND
                                Mgr_ssn=Ssn AND Lname='Smith' )
                  OR
                  Pnumber IN
                  ( SELECT      Pno
                    FROM        WORKS_ON, EMPLOYEE
                    WHERE       Essn=Ssn AND Lname='Smith' );
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Nested Queries (2 of 4)

- Use tuples of values in comparisons
  - Place them within parentheses

```
SELECT      DISTINCT Essn
FROM        WORKS_ON
WHERE       (Pno, Hours) IN ( SELECT      Pno, Hours
                              FROM        WORKS_ON
                              WHERE       Essn='123456789' );
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Nested Queries (3 of 4)

- Use other comparison operators to compare a single value *v*
  - = `ANY` (or = `SOME`) operator
    - Returns `TRUE` if the value *v* is equal to some value in the set *V* and is hence equivalent to `IN`
  - Other operators that can be combined with `ANY` (or `SOME`): >, >=, <, <=, and <>
  - `ALL:` value must exceed all values from nested query

```
SELECT    Lname, Fname
FROM      EMPLOYEE
WHERE     Salary > ALL    ( SELECT    Salary
                            FROM      EMPLOYEE
                            WHERE     Dno=5 );
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Nested Queries (4 of 4)

- Avoid potential errors and ambiguities
  - Create tuple variables (aliases) for all tables referenced in SQL query

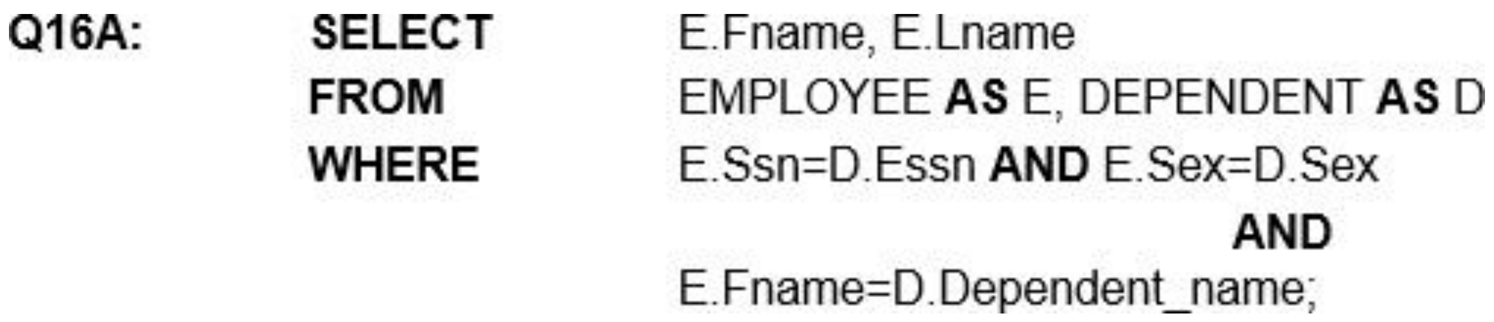
Query 16. Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee.

```
Q16:    SELECT    E.Fname, E.Lname
        FROM      EMPLOYEE AS E
        WHERE     E.Ssn IN  ( SELECT    Essn
                              FROM      DEPENDENT AS D
                              WHERE     E.Fname=D.Dependent_name
                                        AND E.Sex=D.Sex );
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Correlated Nested Queries

- **Queries that are nested using the = or IN comparison operator** can be collapsed into one single block: E.g., Q16 can be written as:



```
Q16A:    SELECT    E.Fname, E.Lname
         FROM      EMPLOYEE AS E, DEPENDENT AS D
         WHERE     E.Ssn=D.Essn AND E.Sex=D.Sex
                                        AND
                   E.Fname=D.Dependent_name;
```

- **Correlated** nested query
  - Evaluated once for each tuple in the outer query

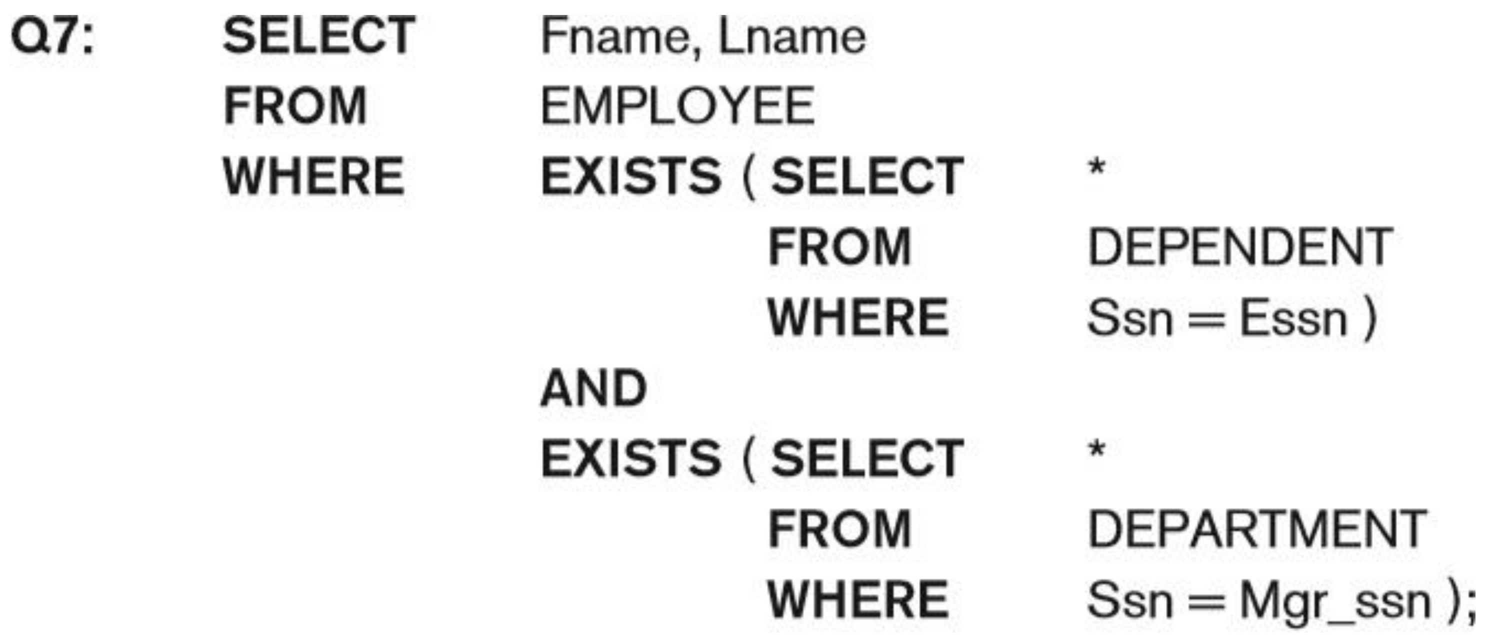
Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

The EXISTS and UNIQUE Functions in SQL for Correlating Queries

- `EXISTS` function
  - Check whether the result of a correlated nested query is empty or not. They are Boolean functions that return a TRUE or FALSE result.
- `EXISTS` and `NOT EXISTS`
  - Typically used in conjunction with a correlated nested query
- SQL function `UNIQUE(Q)`
  - Returns `True` if there are no duplicate tuples in the result of query Q

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

USE of EXISTS



```
Q7:     SELECT    Fname, Lname
        FROM      EMPLOYEE
        WHERE     EXISTS ( SELECT     *
                           FROM       DEPENDENT
                           WHERE      Ssn = Essn )
                  AND
                  EXISTS ( SELECT     *
                           FROM       DEPARTMENT
                           WHERE      Ssn = Mgr_ssn );
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

USE OF NOT EXISTS

To achieve the “for all” (universal quantifier- see Ch 8) effect, we use double negation this way in SQL:

Query: List first and last name of employees who work on **ALL projects controlled by Dno=5.**

```
SELECT      Fname, Lname
FROM        EMPLOYEE
WHERE       NOT EXISTS ( ( SELECT     Pnumber
                           FROM       PROJECT
                           WHERE      Dnum = 5)
                           EXCEPT     ( SELECT    Pno
                                        FROM      WORKS_ON
                                        WHERE     Ssn = Essn) );
```

The above is equivalent to double negation: List names of those employees for whom there does NOT exist a project managed by department no. 5 that they do NOT work on.

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.

Explicit Sets and Renaming of Attributes in SQL

- Can use explicit set of values in WHERE clause

```
Q17:    SELECT    DISTINCT Essn
        FROM      WORKS_ON
        WHERE     Pno IN (1, 2, 3);
```

- Use qualifier AS followed by desired new name
  - Rename any attribute that appears in the result of a query

```
Q8A:    SELECT    E.Lname AS Employee_name, S.Lname AS Supervisor_name
        FROM      EMPLOYEE AS E, EMPLOYEE AS S
        WHERE     E.Super_ssn=S.Ssn;
```

Copyright © 2020 Marasco, 2016 Pearson Education, Inc.