CS 220
Database Systems

Fall 2019

Lecture 5

Comparisons Involving NULL

NULL is used to represent a missing value, there can be different interpretations

- Unknown value
  e.g if date of birth is not known
- Unavailable or Withheld value
  e.g A person has a home phone but does not want it to be listed
- Not Applicable Attribute
  e.g. LastCollegeDegree would be NULL for a person who has no college degrees because it does not apply to that person

→ SQL uses the comparison operators IS or IS NOT.

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

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

View

- A view is a virtual table based on the result-set of an SQL statement
- A SQL statement that is stored in the database with an associated name.
- A view contains rows and columns, just like a real table.
- The fields in a view are fields from one or more real tables in the database.
- We can think of a view as a way of specifying a table that we need to reference frequently, even though it may not exist physically

Storing a View

A view may either be:

- Virtual - produced by a SQL query on demand.
  - **Query modification**: involves modifying or transforming the view query into a query on the underlying base tables
  - Time-consuming to execute, if multiple view queries are going to be applied to the same view within a short period of time.
- Materialized View - the view is stored as a derived table
  - Involves physically creating a temporary or permanent view table when the view is first queried or created and keeping that table on the assumption that other queries on the view will follow
  - Updated when the source tables are updated

Creating a View

A **SQL statement** that is stored in the database with an associated name.

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

- Select statement can be any SQL query and is called the defining query of the view.
- It is possible to rename the columns when defining the view, otherwise they default to the names produced by the query.

Examples

Create a view that has only the employees of department 'D2':

```
CREATE VIEW empD2
AS SELECT * FROM emp WHERE dno = 'D2';
```

Create a view that only shows the employee number, title, and name:

```
CREATE VIEW staff (Number,Name,Title)
AS SELECT eno,ename,title FROM emp;
```

- The first example is a *horizontal view* because it only contains a subset of the rows.
- The second example is a *vertical view* because it only contains a subset of the columns.

Other Operations on View

- Updating a View

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

- Removing a View

```
DROP VIEW view_name;
```

DROP VIEW viewName [*RESTRICT*|*CASCADE*]

RESTRICT will not delete a view if other views are dependent on it.
CASCADE deletes the view and all dependent views.

Views - Points to Remember

- A user can always issue a retrieval query against any view.
- INSERT, DELETE, or UPDATE command on a view is not possible in many cases.
- An update on a view defined on a single table without any aggregate functions can be mapped to an update on the underlying base table under certain conditions.
- For a view involving joins, an update operation may be mapped to update operations on the underlying base relations in multiple ways. Hence, it is often not possible for the DBMS to determine which of the updates is intended.