# **CS 344:** DATABASE MANAGEMENT SYSTEMS

Lecture Notes for 22<sup>ND</sup> August

# **Topic:** SQL (Continued)

# Submitted by:

Sidharth C. Pardeshi 09010162 Sanket Purwar 09010135

## **Other SQL Features:**

## 1. Order by:

ORDER BY clause can be used to specify a sort order.

Syntax [A]: SELECT column\_name(s) FROM table\_name ORDER BY column\_name(s) ASC|DESC

## **Explanation:**

In the above query, the tuples from "table\_name" are obtained in a sorted order (which can be either ascending or descending).

## 2. NULL Values:

A null value if a special value used to denote that the value for a field is unavailable or inapplicable. If one wants to specify that a particular attribute cannot contain NULL values, then that attribute is declared as follows [B]:

attribute char(10) NOT NULL

## 3. Unique/ Not Unique:

In SQL we can declare that a subset of the columns of a table constitute a key by using the UNIQUE constraint.

Syntax [C]: CREATE TABLE Students (sid CHAR(20), name CHAR(30), login CHAR(20), age INTEGER, gpa REAL, UNIQUE (name, age), CONSTRAINT StudentsKey PRIMARY KEY (sid) )

## 4. With:

The SQL With clause allows one to give a sub-query block a name, which can be referenced in several places withing the main SQL query.

### Syntax [D]:

Insert Into new\_table Select \* From (WITH alias\_name AS ( Select a,b,c... From old\_table Select \* From alias\_name))

### 5. Views:

A view is a table whose rows are not explicitly stored in the database but are computed as needed from a view definition.

Syntax [E]: CREATE VIEW B-Students (name, sid, course) AS SELECT S.sname, S.sid, E.cid FROM Students S, Enrolled E WHERE S.sid = E.sid AND E.grade = `B'

#### **Explanation:**

The view B-Students has three fields called name, sid, and course with the same domains as the fields sname and sid in Students and cid in Enrolled.

## Exercise 4.2 [F]

## **SCHEMA:**

Employee (<u>ename</u>, street, city) Works (<u>ename</u>, cname, salary) Company (<u>cname</u>, city) Manages (<u>ename</u>, mname)

#### 1. Find the name and city of the employees who work for company 'X'.

#### Solution:

Step 1: Take a natural join of Employee and Works ( $R_1$ = Employee |**X**| Works) Step 2: Select those tuples which have cname='X' ( $R_2$ =  $\sigma_{cname='X'}$   $R_1$ ) Step 3: Project ename, city from  $R_2$  ( $\pi_{ename,city}$   $R_2$ )

## SQL equivalent:

Select enam, city From employee Natural Join Works Where cname='X'

2. Find the name and city of the employees who work for company 'X' and have a salary of more than Rs. 10,000.

#### Solution:

Step 1: Take a natural join of Employee and Works ( $R_1$ = Employee |**X**| Works) Step 2: Select those tuples which have cname='X' and salary>=Rs. 10,000

 $(R_2 = \sigma_{\text{(cname='X')} \land (\text{salary} >= 10,000)} R_1)$ Step 3: Project ename, city from R<sub>2</sub> (  $\pi_{\text{ename,city}} R_2$ )

## SQL equivalent:

Select enam, city From employee Natural Join Works Where cname='X' and salary>=10,000

#### 3. Find all employee not working for company 'X'.

#### Solution:

Step 1: Project ename from Employee table. ( $R_1 = \pi_{ename}$  Employee)

Step 2: Select those tuples from works where cname='X'. ( $R_2 = \sigma_{\text{cname='X'}}$  Works)

Step 3: Project ename from  $R_2$  ( $R_3 = \pi_{ename,city} R_2$ )

Step 4: Subtract the relation  $R_3$  from  $R_1$ . ( $R_1$ - $R_3$ )

SQL equivalent1:		
Select ename		
From employee		
MINUS		
Select ename		
From Works		
Where cname='X'		

SQL equivalent2: Select ename From employee Where ename NOT IN Select ename From Works Where cname='X'

#### 4. Find the employees who earn more than any employee of company 'X'.

#### Solution:

Step 1: Find the maximum salary in company 'X' from Works table. (If max(salary represents this salary, then R<sub>1</sub>= σ<sub>(cname='X')^(salary=max(salary))</sub> Works)
Step 2: Project the max(salary) (R<sub>2</sub>=π<sub>salary</sub> R<sub>1</sub>)

Step 3: Select those tuples from Works which have salary greater than R2.

	<b>SQL equivalent:</b> R <sub>1</sub> = <b>Select</b> MAX(salary) <b>From</b> Works <b>Where</b> cname='X'	
<b>Select</b> * <b>From</b> Works <b>Where</b> Salary > ALL R1	OR	<b>Select</b> * <b>From</b> Works <b>Where</b> Salary > ANY R1

## 5. Find the company who has most number of employees.

#### Solution:

- Step 1: First group the tuples in Works by cname. Then select cname and count(ename) (call it  $R_1$ )
- Step 2: Select cname from  $R_1$  where the number of employee (count(ename)) is the maximum.

## SQL equivalent:

R<sub>1</sub> = Select cname, count(ename) From Works Group by cname Select cname From R1 Where count(ename)>= ALL

Select Max(count(ename))

From R1.

## A few definitions:

1. Functional Dependence [G]:

Functional dependence generalises the concept of a key. If R is a relation schema, then we say that an instance r of R satisfies the functional dependence  $X \rightarrow Y$  if the following holds for every pair of tuples  $t_1$  and  $t_2$  in r: If  $t_1 \cdot X = t_2 \cdot X$  then  $t_1 \cdot Y = t_2 \cdot Y$ 

- 2. Armstrong's Axioms [H]:
  - a. **Reflexivity:** If  $X \supseteq Y$ , then  $X \rightarrow Y$
  - b. Augmentation: If  $X \rightarrow Y$ , then  $XZ \rightarrow YZ$  for any Z.
  - c. **Transitivity:** If  $X \rightarrow Y$  and  $Y \rightarrow Z$ , then  $X \rightarrow Z$

## **Reference:**

- [A] w3schools.com, "SQL Order By keyword" http://www.w3schools.com/sql/sql\_orderby.asp
- [B] Ramakrishnan, Gehrke, Database Management Systems, Page 67
- [C] Ramakrishnan, Gehrke, Database Management Systems, "Specifying Key Constraints in SQL", Page 67
- [D] S. R. Obbayi, "SQL's With Clause" http://www.brighthub.com/internet/web-development/articles/91893.aspx
- [E] Ramakrishnan, Gehrke, Database Management Systems, "Introduction to Views", Page 67
- [F] Silberschatz, Korth and Sudarshan, Database Systems Concepts, Exercise 4.2, Page 149
- [G] Ramakrishnan, Gehrke, Database Management Systems, "Functional Dependencies", Page 611
- [H] Ramakrishnan, Gehrke, Database Management Systems, "Closure of a set of FDs", Page 612