



Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

# Database Management Systems

## Module 10: Introduction to SQL/3

Partha Pratim Das

Department of Computer Science and Engineering  
Indian Institute of Technology, Kharagpur

*ppd@cse.iitkgp.ac.in*



## Module 10

Partha Pratim  
Das

### Objectives & Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Completed the understanding of basic query structure



## Module 10

Partha Pratim  
Das

### Objectives & Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- To familiarize with set operations, null values and aggregation



## Module 10

Partha Pratim  
Das

### Objectives & Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Set Operations: union, intersect, except
- Null Values
- Aggregate Functions: avg, min, max, sum, and count
  - Group By
  - Having
  - Null Values



## Module 10

Partha Pratim  
Das

Objectives &  
Outline

**Set Operations**

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

# Set Operations



# Set Operations

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Find courses that ran in Fall 2009 or in Spring 2010  
(**select** *course\_id* **from** *section* **where** *sem* = 'Fall' **and** *year* = 2009)  
**union**  
(**select** *course\_id* **from** *section* **where** *sem* = 'Spring' **and** *year* = 2010)
- Find courses that ran in Fall 2009 and in Spring 2010  
(**select** *course\_id* **from** *section* **where** *sem* = 'Fall' **and** *year* = 2009)  
**intersect**  
(**select** *course\_id* **from** *section* **where** *sem* = 'Spring' **and** *year* = 2010)
- Find courses that ran in Fall 2009 but not in Spring 2010  
(**select** *course\_id* **from** *section* **where** *sem* = 'Fall' **and** *year* = 2009)  
**except**  
(**select** *course\_id* **from** *section* **where** *sem* = 'Spring' **and** *year* = 2010)



# Set Operations (2)

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Find the salaries of all instructors that are less than the largest salary

```
select distinct T.salary  
from instructor as T, instructor as S  
where T.salary < S.salary
```

- Find all the salaries of all instructors

```
select distinct salary  
from instructor
```

- Find the largest salary of all instructors

```
(select "second query" )  
except  
(select "first query")
```



# Set Operations (3)

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Set operations **union**, **intersect**, and **except**
  - Each of the above operations automatically eliminates duplicates
- To retain all duplicates use the corresponding multiset versions **union all**, **intersect all**, and **except all**.
- Suppose a tuple occurs  $m$  times in  $r$  and  $n$  times in  $s$ , then, it occurs:
  - $m + n$  times in  $r$  **union all**  $s$
  - $\min(m, n)$  times in  $r$  **intersect all**  $s$
  - $\max(0, m - n)$  times in  $r$  **except all**  $s$





## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

**Null Values**

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

# Null Values



# Null Values

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- It is possible for tuples to have a null value, denoted by *null*, for some of their attributes
- *null* signifies an unknown value or that a value does not exist
- The result of any arithmetic expression involving *null* is *null*
  - Example:  $5 + \text{null}$  returns null
- The predicate **is null** can be used to check for null values
  - Example: Find all instructors whose salary is null  

```
select name
from instructor
where salary is null
```
- It is not possible to test for **null** values with comparison operators, such as  $=$ ,  $<$ , or  $<>$   
We need to use the **is null** and **is not null** operators instead



# Null Values (2): Three Valued Logic

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Three values – **true**, **false**, **unknown**
- Any comparison with *null* returns *unknown*
  - Example:  $5 < null$  or  $null <> null$  or  $null = null$
- Three-valued logic using the value unknown:
  - OR: (*unknown* **or** *true*) = *true*,  
(*unknown* **or** *false*) = *unknown*  
(*unknown* **or** *unknown*) = *unknown*
  - AND: (*true* **and** *unknown*) = *unknown*,  
(*false* **and** *unknown*) = *false*,  
(*unknown* **and** *unknown*) = *unknown*
  - NOT: (**not** *unknown*) = *unknown*
  - “*P* is **unknown**” evaluates to *true* if predicate *P* evaluates to *unknown*
- Result of **where** clause predicate is treated as *false* if it evaluates to *unknown*



## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

**Aggregate  
Functions**

Group By

Having

Null Values

Module Summary

# Aggregate Functions



# Aggregate Functions

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- These functions operate on the multiset of values of a column of a relation, and return a value
  - avg**: average value
  - min**: minimum value
  - max**: maximum value
  - sum**: sum of values
  - count**: number of values



# Aggregate Functions (2)

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Find the average salary of instructors in the Computer Science department  
**select avg (salary)**  
**from instructor**  
**where dept\_name = 'Comp. Sci';**
- Find the total number of instructors who teach a course in the Spring 2010 semester  
**select count (distinct ID)**  
**from teaches**  
**where semester = 'Spring' and year = 2010;**
- Find the number of tuples in the *course* relation  
**select count (\*)**  
**from courses;**



# Aggregate Functions (3): Group By

## Module 10

Partha Pratim  
DasObjectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Find the average salary of instructors in each department

```
select dept_name, avg(salary) as avg_salary  
from instructor  
group by dept_name;
```

ID	name	dept_name	salary
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

dept_name	avg_salary
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000



# Aggregate Functions (4): Group By

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Attributes in **select** clause outside of aggregate functions must appear in **group by** list  
*/\* erroneous query \*/*  
**select** *dept\_name, ID, avg(salary)*  
**from** *instructor*  
**group by** *dept\_name;*





# Aggregate Functions (5): Having Clause

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

**Having**

Null Values

Module Summary

- Find the names and average salaries of all departments whose average salary is greater than 42000

```
select dept_name, ID, avg(salary)
from instructor
group by dept_name
having avg(salary) > 42000;
```

Note: predicates in the **having** clause are applied after the formation of groups whereas predicates in the **where** clause are applied before forming groups



# Null Values and Aggregates

## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Total all salaries  
**select sum (salary)**  
**from instructor;**
  - Above statement ignores null amounts
  - Result is *null* if there is no non-null amount
- All aggregate operations except **count(\*)** ignore tuples with null values on the aggregated attributes
- What if collection has only null values?
  - count returns 0
  - all other aggregates return null



## Module 10

Partha Pratim  
Das

Objectives &  
Outline

Set Operations

Null Values

Three Valued Logic

Aggregate  
Functions

Group By

Having

Null Values

Module Summary

- Completed the understanding of set operations, null values, and aggregation

**Slides used in this presentation are borrowed from <http://db-book.com/> with kind permission of the authors.**

**Edited and new slides are marked with “PPD”.**