

# Databases and Information Systems

## CS303

SQL

11-08-2023

# Recap

- SQL
  - Create / Delete : Databases and Tables
  - SELECT queries

# Ordering the displayed results

- `SELECT name`  
`FROM instructor`  
`WHERE dept name = 'Comp. Sci.' ;`

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

`instructor`

# Ordering the displayed results

- `SELECT name`  
`FROM instructor`  
`WHERE dept name = 'Comp. Sci.' ;`  
`ORDER by name;`
- `SELECT *`  
`FROM instructor`  
`ORDER BY salary DESC, name ASC;`

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

`instructor`

# WHERE - BETWEEN

- Find the names of instructors with salary amounts between 70,000 and 90,000

- SELECT name  
FROM instructor  
WHERE salary <= 90000 AND salary >= 70000;

- SELECT name  
FROM instructor  
WHERE salary BETWEEN 70000 AND 90000;

- For the negation:

SELECT name  
FROM instructor  
WHERE salary NOT BETWEEN 70000 AND 90000;

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

instructor

# WHERE - Tuple comparison

- Find the instructor names in Biology department names who have taught some course and also retrieve the courses ids' that they have taught.
- SELECT name, course\_id  
FROM instructor, teaches  
WHERE instructor.ID= teaches.ID AND dept\_name = 'Biology';
- SELECT name, course\_id  
FROM instructor, teaches  
WHERE (instructor.ID, dept\_name) = (teaches.ID, 'Biology');

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

instructor

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

teaches

# SET operations

- `SELECT course_id`  
`FROM section`  
`WHERE semester = 'Fall' AND year= 2022;`
- `SELECT course_id`  
`FROM section`  
`WHERE semester = 'Spring' AND year= 2022;`

*course\_id*

CS-101  
CS-347  
PHY-101

*course\_id*

CS-101  
CS-315  
CS-319  
CS-319  
FIN-201  
HIS-351  
MU-199

# SET operations : Union

- `SELECT course_id`  
`FROM section`  
`WHERE semester = 'Fall' and year= 2022;`
- `SELECT course_id`  
`FROM section`  
`WHERE semester = 'Spring' and year= 2022;`
- `(SELECT course_id`  
`FROM section`  
`WHERE semester = 'Fall' and year= 2022 )`  
`UNION`  
`(SELECT course_id`  
`FROM section`  
`WHERE semester = 'Spring' and year= 2022 ) ;`
- UNION eliminates duplicates  
If you want duplicates use UNION ALL

*course\_id*

CS-101  
CS-347  
PHY-101

*course\_id*

CS-101  
CS-315  
CS-319  
CS-319  
FIN-201  
HIS-351  
MU-199

*course\_id*

CS-101  
CS-315  
CS-319  
CS-347  
FIN-201  
HIS-351  
MU-199  
PHY-101



# SET operations : Intersection

- SELECT course id  
FROM section  
WHERE semester = 'Fall' and year= 2022;
- SELECT course id  
FROM section  
WHERE semester = 'Spring' and year= 2022;
- (SELECT course id  
FROM section  
WHERE semester = 'Fall' and year= 2022 )  
INTERSECTION  
(SELECT course id  
FROM section  
WHERE semester = 'Spring' and year= 2022 ) ;
- INTERSECTION eliminates duplicates  
If you want duplicates use INTERSECTION ALL

*course\_id*

CS-101  
CS-347  
PHY-101

*course\_id*

CS-101  
CS-315  
CS-319  
CS-319  
FIN-201  
HIS-351  
MU-199

*course\_id*

CS-101

# SET operations : SET DIFFERENCE

- SELECT course id  
FROM section  
WHERE semester = 'Fall' and year= 2022;
- SELECT course id  
FROM section  
WHERE semester = 'Spring' and year= 2022;
- (SELECT course id  
FROM section  
WHERE semester = 'Fall' and year= 2022 )  
EXCEPT  
(SELECT course id  
FROM section  
WHERE semester = 'Spring' and year= 2022 ) ;
- EXCEPT eliminates duplicates  
If you want duplicates use EXCEPT ALL

*course\_id*

CS-101  
CS-347  
PHY-101

*course\_id*

CS-101  
CS-315  
CS-319  
CS-319  
FIN-201  
HIS-351  
MU-199

*course\_id*

CS-347  
PHY-101

# NULL values

- What happens to NULL in arithmetic operations?

`SELECT r.A * 0.3 FROM r;`

- WHAT happens to comparisons?

`SELECT * FROM r WHERE r.A < 10 ;`

- `5 < NULL` (cannot be true; cannot be false)
- `NULL` evaluates to special value 'unknown'
- `Truth tables` are defined accordingly

# NULL values

- `SELECT name`  
`FROM instructor`  
`WHERE salary IS NULL;`
- `SELECT name`  
`FROM instructor`  
`WHERE salary IS NOT NULL;`

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

`instructor`

# Aggregate functions

- Average: avg
- Minimum: min
- Maximum: max
- Total: sum
- Count: count

- `SELECT avg(salary)`  
`FROM instructor`  
`WHERE dept_name = 'Comp. Sci.' ;`

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

`instructor`

# Aggregate functions

- Average: avg
- Minimum: min
- Maximum: max
- Total: sum
- Count: count

- `SELECT count(*)`  
`FROM instructor`  
`WHERE dept_name = 'Comp. Sci.' ;`

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

`instructor`

# Aggregate functions

- Average: avg
  - Minimum: min
  - Maximum: max
  - Total: sum
  - Count: count
- 
- `SELECT count( DISTINCT dept_name)`  
`FROM instructor ;`

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

`instructor`

# Aggregate functions

- Average: avg
  - Minimum: min
  - Maximum: max
  - Total: sum
  - Count: count
- 
- What happens to aggregation when there are **NULL** values?



# GROUP BY

- 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

result

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

instructor

# GROUP BY

- While using GROUP BY, the SELECT can only have attributes that are grouped by and aggregate operators.
- SELECT dept\_name, 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

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

instructor

# GROUP BY

- Find the number of instructors in each department who teach a course in the Spring 2010 semester.

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

*instructor*

<i>ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

*teaches*

# GROUP BY - HAVING

- Retrieve names and the average salary of departments for all the departments that have average salary > 40000
- 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

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

instructor

# GROUP BY - HAVING

- Retrieve names and the average salary of departments for all the departments that have average salary > 40000
- SELECT dept\_name, avg (salary) as avg\_salary  
FROM instructor  
GROUP BY dept\_name  
HAVING avg(salary)>40000;

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

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

instructor

# GROUP BY - HAVING

- For each course section offered in 2009, find the average total credits (tot\_cred) of all students enrolled in the section, if the section had at least 2 students

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

student

ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	null

takes

# GROUP BY - HAVING queries evaluation steps

- `SELECT A1, A2 .... An`  
`FROM r1, r2 ... rm`  
`WHERE P1`  
`GROUP BY .....`  
`HAVING P2;`
- **FROM** gives all the relevant relations to be considered
- If there is a **WHERE** condition, apply it and keep only the tuples that satisfy the condition
- Group the remaining relation based on the **GROUP BY** condition
- Remove the groups that do not satisfy the **HAVING** condition
- **SELECT** the remaining groups and retrieve the attributes / aggregate functions