



CSc 484

# Database Management Systems

Ken Gamradt

Spring 2024

Select

## Select statement – review

- The order of the clauses in the SELECT statement **cannot** be changed

```
SELECT [DISTINCT | ALL] { * | [columnExpression [AS newName]] [,...]}  
FROM   TableName [alias] [,...]  
[WHERE condition]  
[GROUP BY columnList] [HAVING condition]  
[ORDER BY columnList]
```

## Select statement – review

- The sequence of processing in a **SELECT** statement:
  - **FROM**: specify the table(s) to be used
  - **WHERE**: filter the rows subject to some condition
  - **GROUP BY**: form the groups of rows with the same column value
  - **HAVING**: filter the groups subject to some condition
  - **SELECT**: specify which columns are to appear in the output
  - **ORDER BY**: specify the order of the output



## A little more about Aggregation

- If the **SELECT** list includes an aggregate function, and no **GROUP BY** clause is being used, then no column names can appear in the **SELECT** clause
  - Unless they are in an aggregate function

```
select name, AVG(salary)      -- illegal
from instructor
```

- When aggregation is used without the **GROUP BY** clause, the whole relation is treated as one group
  - PostgreSQL
    - Column "instructor.name" must appear in the **GROUP BY** clause or be used in an aggregate function

## Subquery – review

- One **SELECT** statement (inner select / subselect) can be embedded within another **SELECT** statement (outer statement) to determine the contents of the outer statement

```
select *  
  from instructor  
 where salary > (select AVG(salary) -- subselect always in ( )  
                from instructor);  -- usually used in WHERE,  
                                   -- HAVING clauses of outer select
```

- A subselect can also be used in INSERT, UPDATE, DELETE statements



# Relations

## instructor

id	name	dept_name	salary
abc Filter...	abc Filter...	abc Filter...	abc Filter...
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

## department

dept_name	building	budget
abc Filter...	abc Filter...	abc Filter...
Biology	Watson	90000.00
Comp. Sci.	Taylor	100000.00
Elec. Eng.	Taylor	85000.00
Finance	Painter	120000.00
History	Painter	50000.00
Music	Packard	80000.00
Physics	Watson	70000.00

# Relations

## instructor

id	name	dept_name	salary
<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

## teaches

id	course_id	sec_id	semester	year
<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017



# Subquery

- Usage 1: set membership
  - Find the ID and name of instructors whose department is located in the Taylor building

```
select id, name          -- outer select
  from instructor
 where dept_name in (
    select dept_name      -- inner select
      from department     -- 'Comp. Sci.' and 'Elec. Eng.'
     where building = 'Taylor');
```

- This should be a **JOIN**, not a subquery



# Subquery

- Usage 1: set membership
  - Find all `course_id` that were taught in Fall 2017 and that are also members of the set of `course_id` taught in Spring 2018

```
select course_id
  from section
 where semester = 'Fall' AND year = 2017
```

```
select course_id
  from section
 where semester = 'Spring' AND year = 2018
```

## course\_id

abc Filter...

CS-101

CS-347

PHY-101

## course\_id

abc Filter...

CS-101

CS-315

CS-319

CS-319

FIN-201





HIS-351

MU-199

# Subquery

- Usage 2: comparison
  - List all instructor whose salary is greater than the average salary of Biology

```
select ID, name, dept_name, salary
  from instructor
 where salary > (
    select AVG(salary) -- > 72000
      from instructor
     where dept_name = 'Biology');
```

id	name	dept_name	salary
 Filter...	 Filter...	 Filter...	 Filter...
12121	Wu	Finance	90000.00
22222	Einstein	Physics	95000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
76543	Singh	Finance	80000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00



# Subquery

- Usage 2: comparison and **ANY/ALL**
- **ANY** and **ALL** can be used with subqueries that produce a single column of numbers
- **ALL**: condition is true only if satisfied by all values produced by the subquery
  - Eg: ... > **ALL** (subquery) means greater than all
- **ANY**: condition is true if satisfied by any (1+) values produced by the subquery
  - Eg: ... > **ANY** (subquery) means greater than at least one
- If the subquery is empty
  - **ALL** condition returns true
  - **ANY** condition returns false
- ISO standard also allows **SOME** to be used in place of **ANY**

# Subquery

- Usage 2: comparison and **ANY/ALL**
  - List all instructor whose salary is larger than the salary of every instructor in the Comp. Sci. department

```
select ID, name
  from instructor
 where salary > ALL (
    select salary
      from instructor
     where dept_name = 'Comp. Sci.');
```

id	name
abc Filter...	abc Filter...
22222	Einstein



# Subquery

- Usage 2: comparison and **ANY/ALL**
  - List all instructor whose salary is larger than the salary of at least one instructor in the Biology department

```
select ID, name
  from instructor
 where salary > ANY (
    select salary
      from instructor
     where dept_name = 'Biology');
```

id	name
abc Filter...	abc Filter...
12121	Wu
22222	Einstein
33456	Gold
45565	Katz
76543	Singh
83821	Brandt
98345	Kim

## Subquery – rules for using

- The **ORDER BY** clause may not be used in a subquery
  - It may be used in the outer select statement
- When using for comparison, the subquery select list must consist of a single column name or expression
  - Except for subqueries that use the keyword **EXISTS**
- By default, column names in a subquery refer to the table name in the **FROM** clause of the subquery
- When the subquery as an operand in a comparison, the subquery must appear on the right-hand side



# Select

Example: find the **name**, **dept\_name**, **building** of all instructor who work in the Taylor building?

id	name	dept_name	salary
abc Filter...	abc Filter...	abc Filter...	abc Filter...
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

dept_name	building	budget
abc Filter...	abc Filter...	abc Filter...
Biology	Watson	90000.00
Comp. Sci.	Taylor	100000.00
Elec. Eng.	Taylor	85000.00
Finance	Painter	120000.00
History	Painter	50000.00
Music	Packard	80000.00
Physics	Watson	70000.00

## Select – join

- Join operation
  - Combine information from two tables by forming pairs of related rows from the two tables

```
select name, instructor.dept_name, building
  from instructor, department      -- relations that need to be accessed
 where instructor.dept_name = department.dept_name AND
        building = 'Taylor';      -- matching columns – join condition
                                   -- define cartesian product of relations
select name, I.dept_name, building -- apply aliases to relation names
  from instructor as I, department as D
 where I.dept_name = D.dept_name AND building = 'Taylor';
```



## Table alias

- Early versions of SQL did not include the keyword **AS**
  - Some implementations (Oracle, ...), do not permit the keyword **AS** in the **FROM** clause

```
select name, D.dept_name, building
  from instructor I, department D
 where I.dept_name = D.dept_name
```

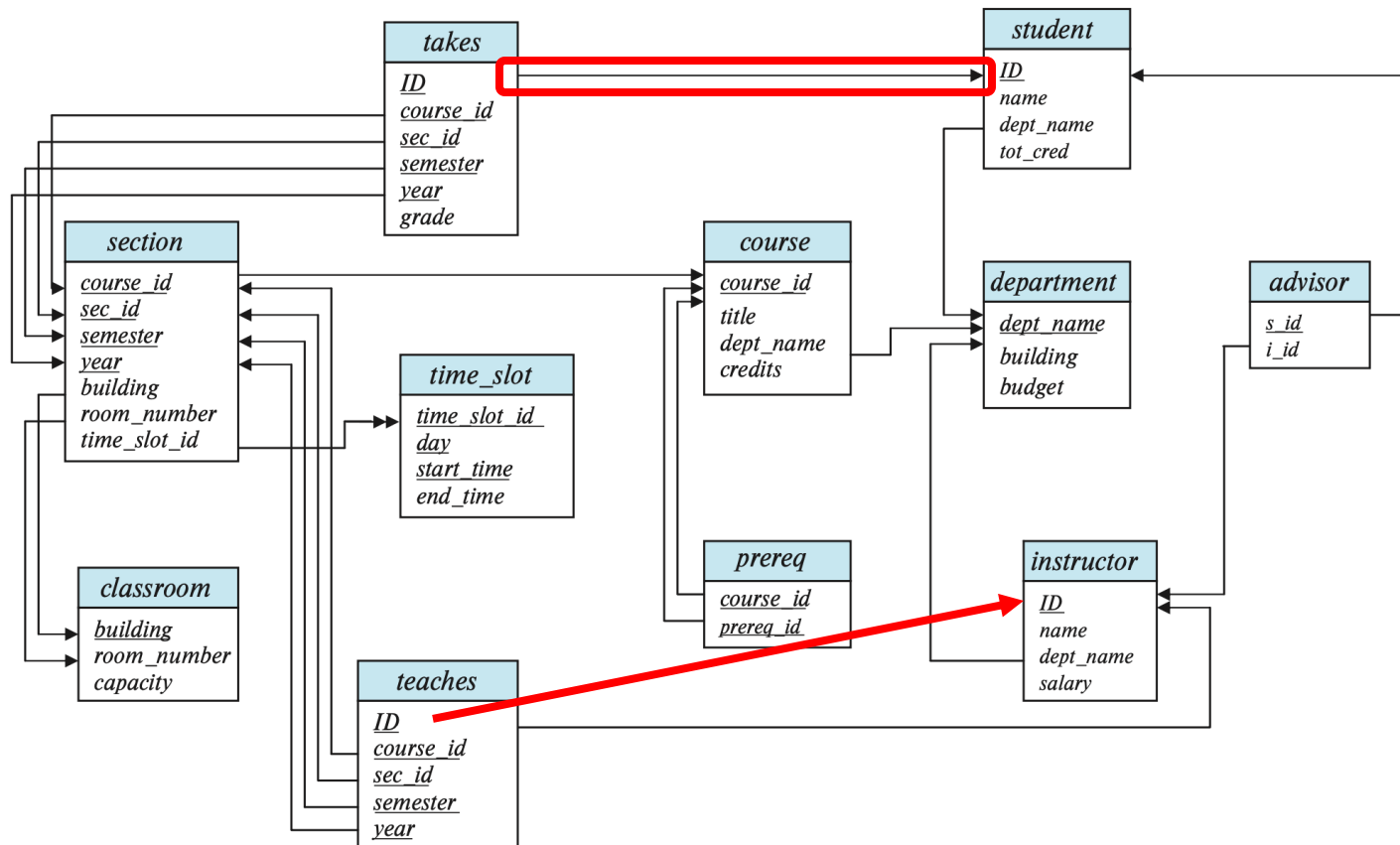
```
select name, D.dept_name, building
  from instructor as I, department as D
 where I.dept_name = D.dept_name
```

## Select – join condition

- The most common multi-table queries involve two tables that have foreign-key constraints
  - instructor and teaches
  - student and takes
- In most cases, the join condition is created using the primary key of one table and the foreign key of another table



# University schema diagram



## Select – join condition

- Overall process
  - Form the Cartesian product of the tables named in the **FROM** clause
  - If there is a **WHERE** clause
    - Apply the search condition to each row of the product table
      - Retaining those rows that satisfy the condition
    - For each retained row
      - Determine the value of each item in the **SELECT** list to produce a single row in the result table
  - If **SELECT DISTINCT** has been specified
    - Eliminate any duplicate rows from the result table
  - If there is an **ORDER BY** clause
    - Sort the result table as required



## Select – join condition

- Step 1:

*from instructor I, department D*

- Form the cartesian product of tables in the **FROM** clause

```
select name, D.dept_name, building
from instructor I, department D
where I.dept_name = D.dept_name
```

	ID	name	dept_name	salary	dept_name	building	budget
1	10101	Srinivasan	Comp. Sci.	65000.00	Biology	Watson	90000.00
2	12121	Wu	Finance	90000.00	Biology	Watson	90000.00
3	15151	Mozart	Music	40000.00	Biology	Watson	90000.00
4	22222	Einstein	Physics	95000.00	Biology	Watson	90000.00
5	32343	El Said	History	60000.00	Biology	Watson	90000.00
6	33456	Gold	Physics	87000.00	Biology	Watson	90000.00
7	45565	Katz	Comp. Sci.	75000.00	Biology	Watson	90000.00
8	58583	Califeri	History	62000.00	Biology	Watson	90000.00
9	76543	Singh	Finance	80000.00	Biology	Watson	90000.00
10	76766	Crick	Biology	72000.00	Biology	Watson	90000.00
11	83821	Brandt	Comp. Sci.	92000.00	Biology	Watson	90000.00
12	98345	Kim	Elec. Eng.	80000.00	Biology	Watson	90000.00
13	10101	Srinivasan	Comp. Sci.	65000.00	Comp. Sci.	Taylor	100000.00
14	12121	Wu	Finance	90000.00	Comp. Sci.	Taylor	100000.00
15	15151	Mozart	Music	40000.00	Comp. Sci.	Taylor	100000.00
16	22222	Einstein	Physics	95000.00	Comp. Sci.	Taylor	100000.00
17	32343	El Said	History	60000.00	Comp. Sci.	Taylor	100000.00
18	33456	Gold	Physics	87000.00	Comp. Sci.	Taylor	100000.00
19	45565	Katz	Comp. Sci.	75000.00	Comp. Sci.	Taylor	100000.00
20	58583	Califeri	History	62000.00	Comp. Sci.	Taylor	100000.00
21	76543	Singh	Finance	80000.00	Comp. Sci.	Taylor	100000.00
22	76766	Crick	Biology	72000.00	Comp. Sci.	Taylor	100000.00
23	83821	Brandt	Comp. Sci.	92000.00	Comp. Sci.	Taylor	100000.00
24	98345	Kim	Elec. Eng.	80000.00	Comp. Sci.	Taylor	100000.00
25	10101	Srinivasan	Comp. Sci.	65000.00	Elec. Eng.	Taylor	85000.00
26	12121	Wu	Finance	90000.00	Elec. Eng.	Taylor	85000.00



## Select – join condition

- Step 2:
  - *where I.dept\_name = D.dept\_name*
  - Process the **WHERE** clause which include the matching condition

```
select name, D.dept_name, building
from instructor I, department D
where I.dept_name = D.dept_name
```

	ID	name	dept_name	salary	dept_name	building	budget
1	10101	Srinivasan	Comp. Sci.	65000.00	Comp. Sci.	Taylor	100000.00
2	12121	Wu	Finance	90000.00	Finance	Painter	120000.00
3	15151	Mozart	Music	40000.00	Music	Packard	80000.00
4	22222	Einstein	Physics	95000.00	Physics	Watson	70000.00
5	32343	El Said	History	60000.00	History	Painter	50000.00
6	33456	Gold	Physics	87000.00	Physics	Watson	70000.00
7	45565	Katz	Comp. Sci.	75000.00	Comp. Sci.	Taylor	100000.00
8	58583	Califeri	History	62000.00	History	Painter	50000.00
9	76543	Singh	Finance	80000.00	Finance	Painter	120000.00
10	76766	Crick	Biology	72000.00	Biology	Watson	90000.00
11	83821	Brandt	Comp. Sci.	92000.00	Comp. Sci.	Taylor	100000.00
12	98345	Kim	Elec. Eng.	80000.00	Elec. Eng.	Taylor	85000.00



## Select – join condition

- Step 3:  
*select name, D.dept\_name,  
building*
  - Process the other clauses on the result relation from step 2 condition

```
select name, D.dept_name, building
from instructor I, department D
where I.dept_name = D.dept_name
```

name	dept_name	building
abc Filter...	abc Filter...	abc Filter...
Srinivasan	Comp. Sci.	Taylor
Wu	Finance	Painter
Mozart	Music	Packard
Einstein	Physics	Watson
El Said	History	Painter
Gold	Physics	Watson
Katz	Comp. Sci.	Taylor
Califieri	History	Painter
Singh	Finance	Painter
Crick	Biology	Watson
Brandt	Comp. Sci.	Taylor
Kim	Elec. Eng.	Taylor

# Join

teaches

id	course_id	sec_id	semester	year
<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

section

course_id	sec_id	semester	year	building	room_number	time_slot_id
<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...	<input type="text" value="abc"/> Filter...
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A



## Select – Join

```
select ID, T.course_id, building
  from teaches T, section S
 where
    T.course_id = S.course_id AND
    T.sec_id = S.sec_id AND
    T.semester = S.semester AND
    T.year = S.year
```

id	course_id	building
abc Filter...	abc Filter...	abc Filter...
10101	CS-101	Packard
10101	CS-315	Watson
10101	CS-347	Taylor
12121	FIN-201	Packard
15151	MU-199	Packard
22222	PHY-101	Watson
32343	HIS-351	Painter
45565	CS-101	Packard
45565	CS-319	Watson
76766	BIO-101	Painter
76766	BIO-301	Painter
83821	CS-190	Taylor
83821	CS-190	Taylor
83821	CS-319	Taylor
98345	EE-181	Taylor

# Acknowledgements

- WIKIPEDIA
  - <https://en.wikipedia.org/wiki/SQL>