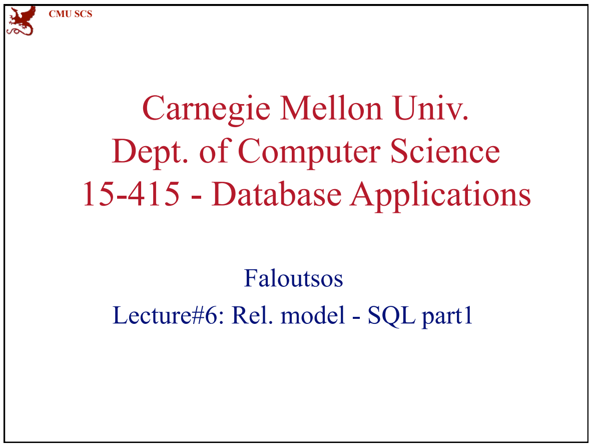
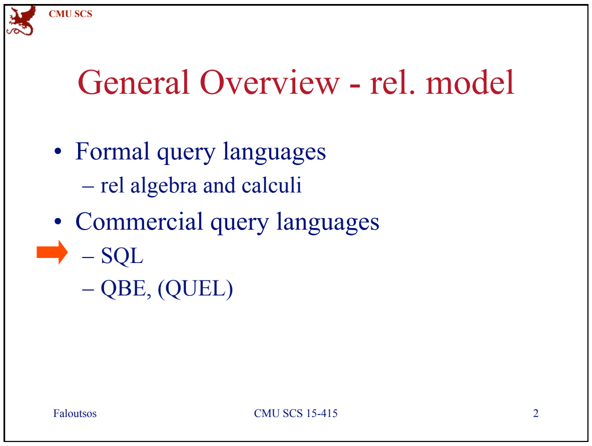


Carnegie Mellon Univ.
Dept. of Computer Science
15-415 - Database Applications

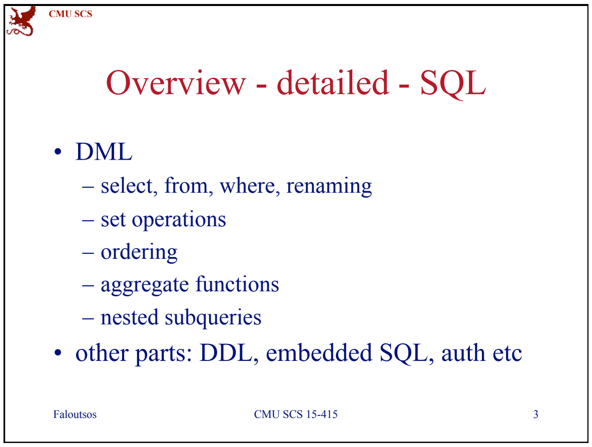
Faloutsos

Lecture#6: Rel. model - SQL part1



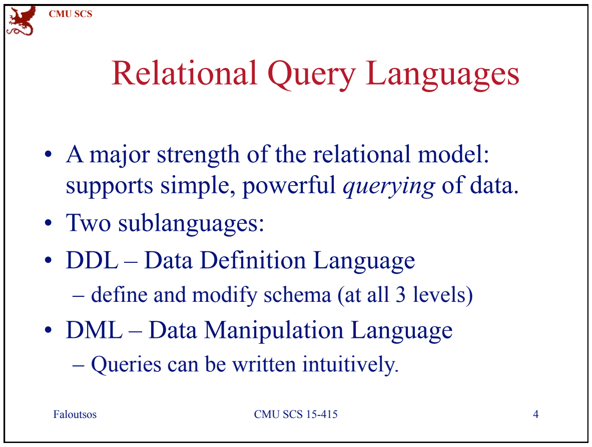
General Overview - rel. model

- Formal query languages
  - rel algebra and calculi
- Commercial query languages
  - ➡ SQL
  - QBE, (QUEL)



Overview - detailed - SQL

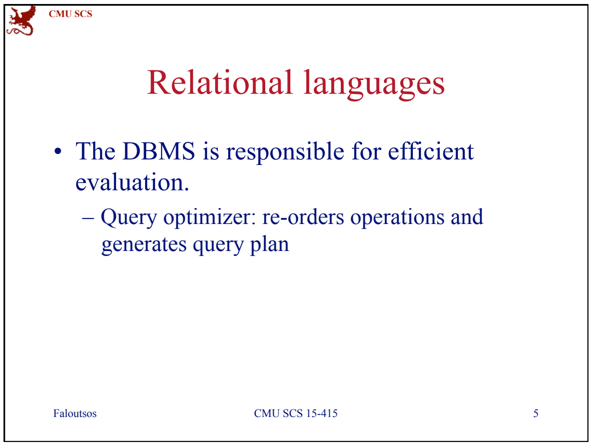
- DML
  - select, from, where, renaming
  - set operations
  - ordering
  - aggregate functions
  - nested subqueries
- other parts: DDL, embedded SQL, auth etc



CMU SCS

Relational Query Languages

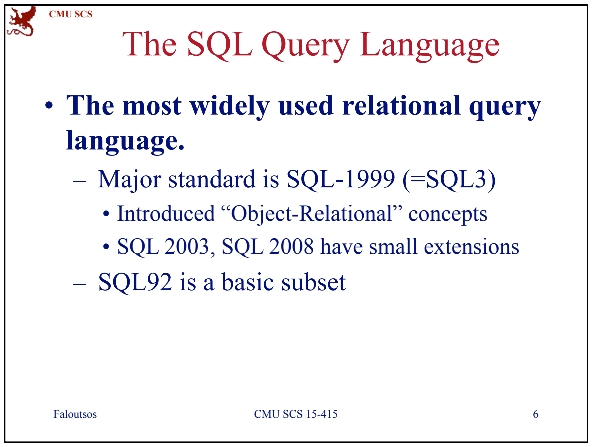
- A major strength of the relational model: supports simple, powerful *querying* of data.
- Two sublanguages:
- DDL – Data Definition Language
  - define and modify schema (at all 3 levels)
- DML – Data Manipulation Language
  - Queries can be written intuitively.



CMU SCS

Relational languages

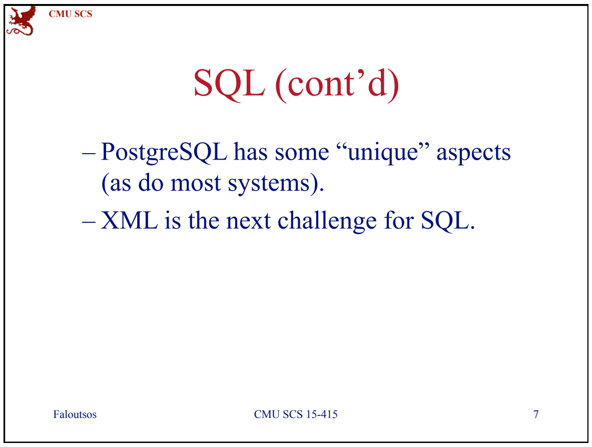
- The DBMS is responsible for efficient evaluation.
  - Query optimizer: re-orders operations and generates query plan



CMU SCS

The SQL Query Language

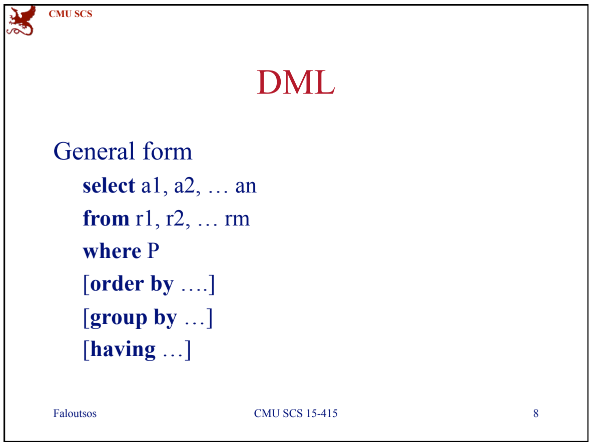
- **The most widely used relational query language.**
  - Major standard is SQL-1999 (=SQL3)
    - Introduced “Object-Relational” concepts
    - SQL 2003, SQL 2008 have small extensions
  - SQL92 is a basic subset



CMU SCS

SQL (cont'd)

- PostgreSQL has some "unique" aspects (as do most systems).
- XML is the next challenge for SQL.

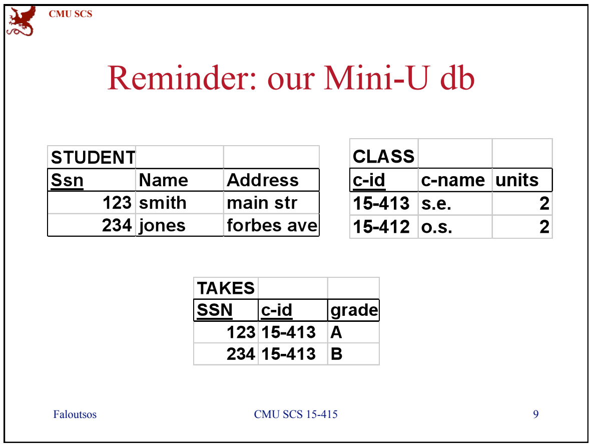


CMU SCS

DML

General form

select a1, a2, … an
from r1, r2, … rm
where P
[**order by** ….]
[**group by** …]
[**having** …]



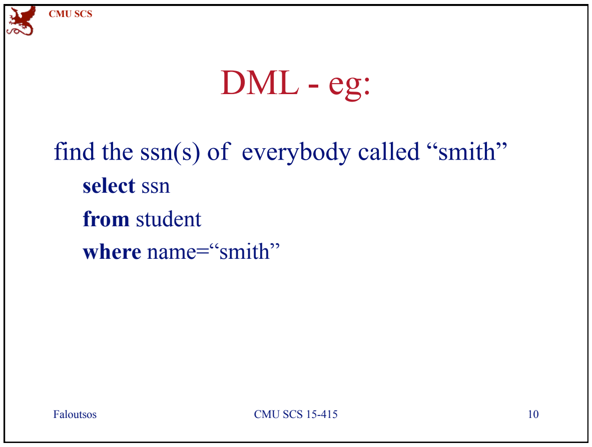
CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

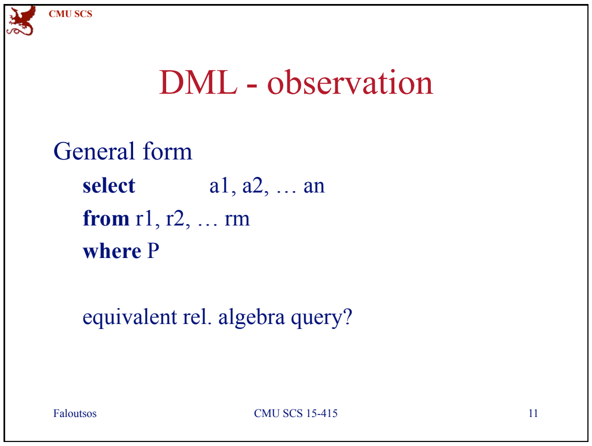
TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B



DML - eg:

find the ssn(s) of everybody called “smith”

select ssn
from student
where name=“smith”

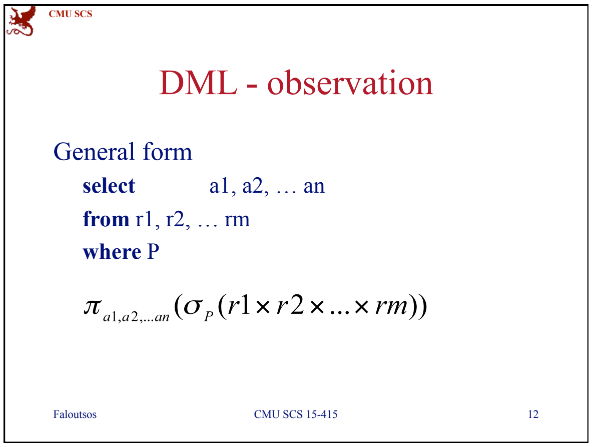


DML - observation

General form

select a1, a2, … an
from r1, r2, … rm
where P

equivalent rel. algebra query?

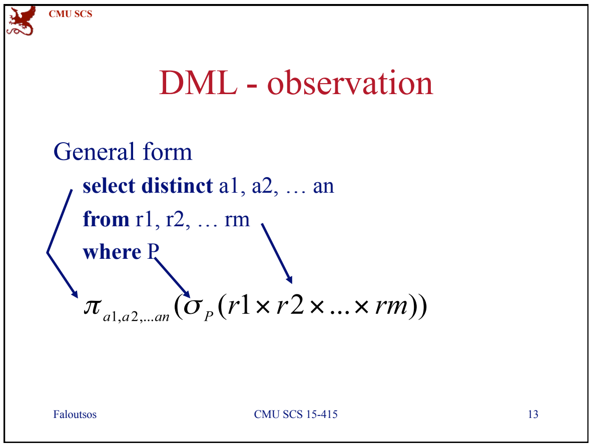


DML - observation

General form

select a1, a2, … an
from r1, r2, … rm
where P

$$\pi_{a1,a2,...an}(\sigma_P(r1 \times r2 \times ... \times rm))$$



CMU SCS

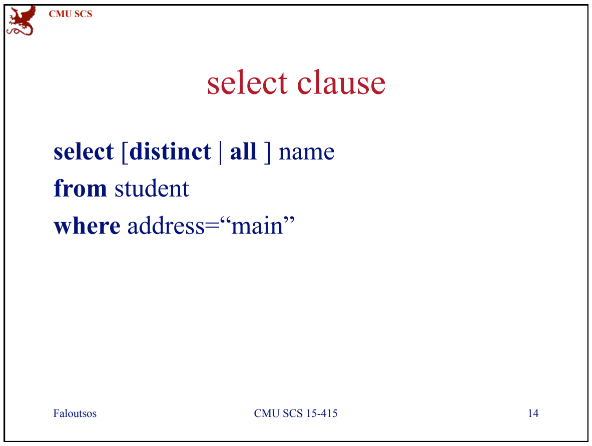
DML - observation

General form

select distinct a1, a2, … an
from r1, r2, … rm
where P

$$\pi_{a1,a2,...an}(\sigma_P(r1 \times r2 \times ... \times rm))$$

Faloutsos CMU SCS 15-415 13

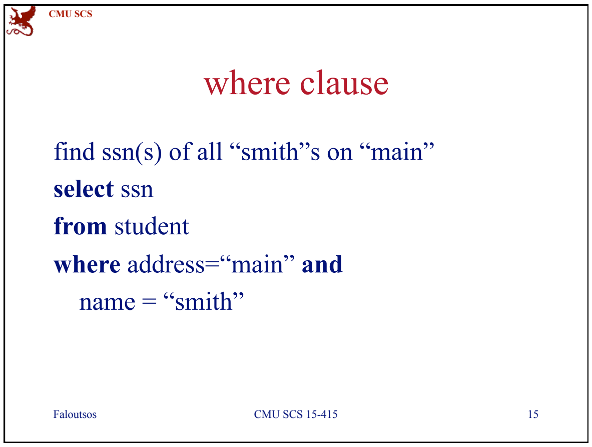


CMU SCS

select clause

select [**distinct** | **all**] name
from student
where address="main"

Faloutsos CMU SCS 15-415 14



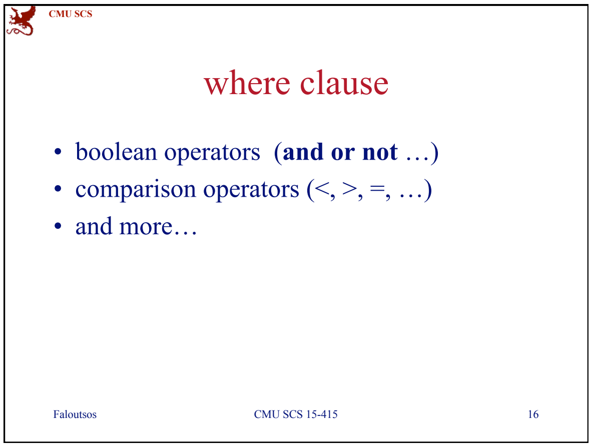
CMU SCS

where clause

find ssn(s) of all "smith"s on "main"

select ssn
from student
where address="main" **and**
name = "smith"

Faloutsos CMU SCS 15-415 15

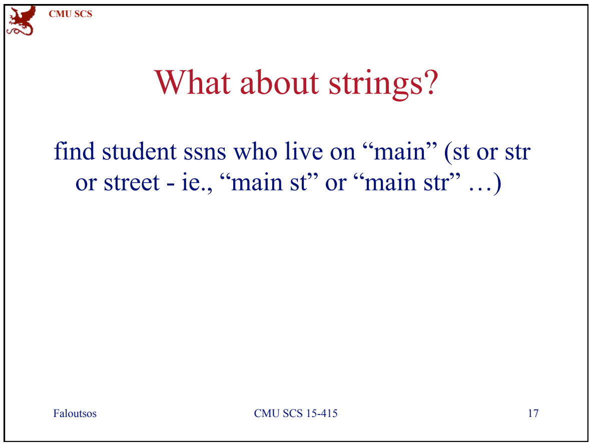


CMU SCS

where clause

- boolean operators (**and or not** …)
- comparison operators (<, >, =, …)
- and more…

Faloutsos CMU SCS 15-415 16

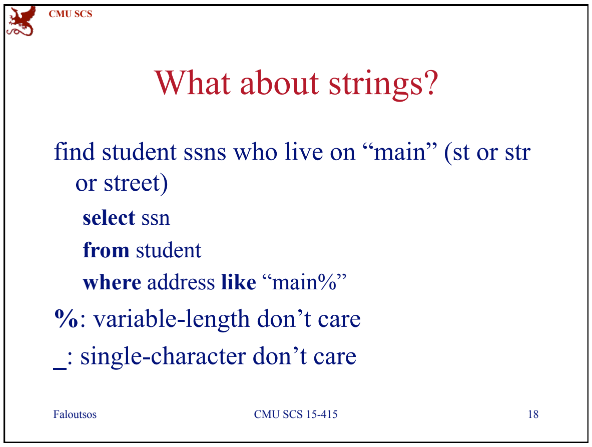


CMU SCS

What about strings?

find student ssns who live on “main” (st or str or street - ie., “main st” or “main str” …)

Faloutsos CMU SCS 15-415 17



CMU SCS

What about strings?

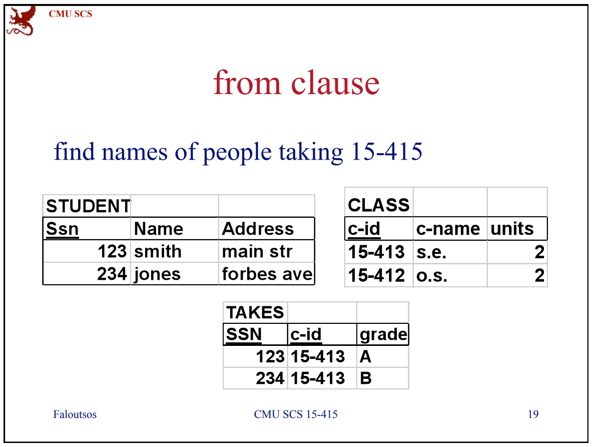
find student ssns who live on “main” (st or str or street)

```
select ssn
from student
where address like “main%”
```

%: variable-length don’t care

_: single-character don’t care

Faloutsos CMU SCS 15-415 18



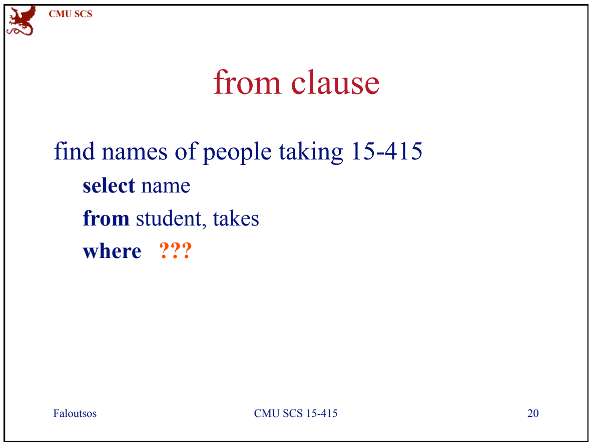
from clause

find names of people taking 15-415

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B



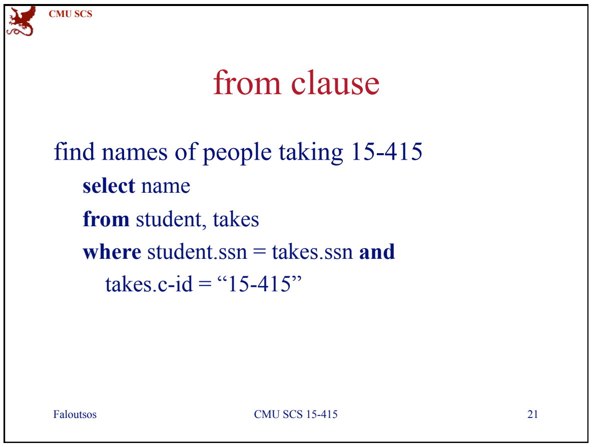
from clause

find names of people taking 15-415

select name

from student, takes

where ???



from clause

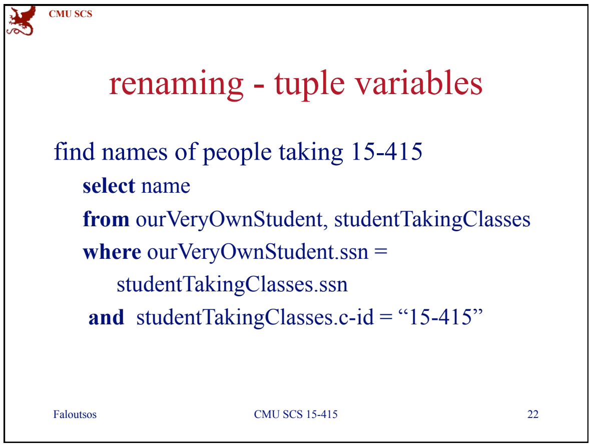
find names of people taking 15-415

select name

from student, takes

where student.ssn = takes.ssn **and**

takes.c-id = “15-415”

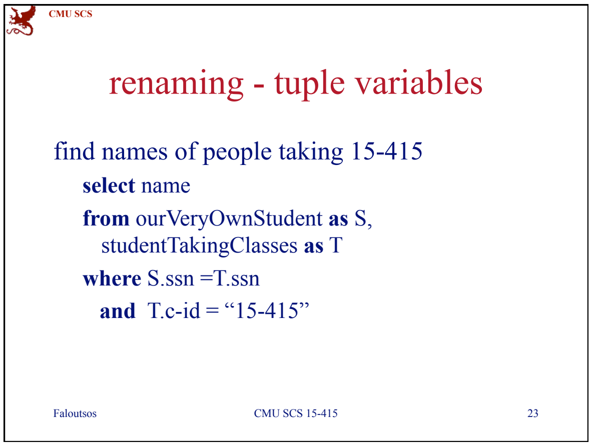


CMU SCS

renaming - tuple variables

find names of people taking 15-415

```
select name
from ourVeryOwnStudent, studentTakingClasses
where ourVeryOwnStudent.ssn =
     studentTakingClasses.ssn
 and  studentTakingClasses.c-id = “15-415”
```

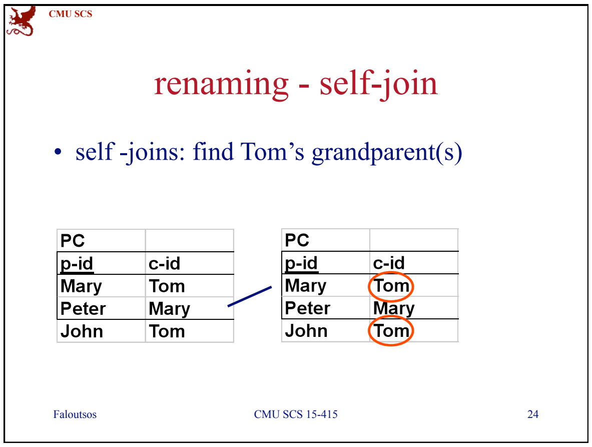


CMU SCS

renaming - tuple variables

find names of people taking 15-415

```
select name
from ourVeryOwnStudent as S,
   studentTakingClasses as T
where S.ssn =T.ssn
   and  T.c-id = “15-415”
```



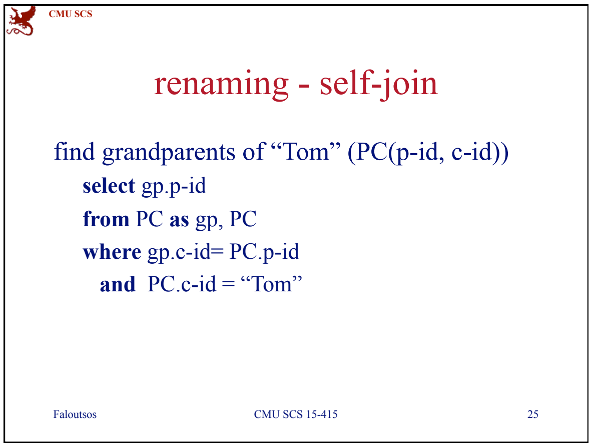
CMU SCS

renaming - self-join

- self -joins: find Tom’s grandparent(s)

PC	
p-id	c-id
Mary	Tom
Peter	Mary
John	Tom

PC	
p-id	c-id
Mary	Tom
Peter	Mary
John	Tom

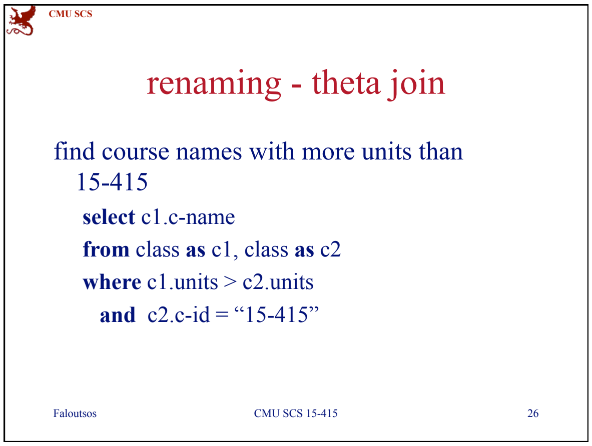


CMU SCS

renaming - self-join

find grandparents of “Tom” (PC(p-id, c-id))

```
select gp.p-id
from PC as gp, PC
where gp.c-id= PC.p-id
  and  PC.c-id = “Tom”
```

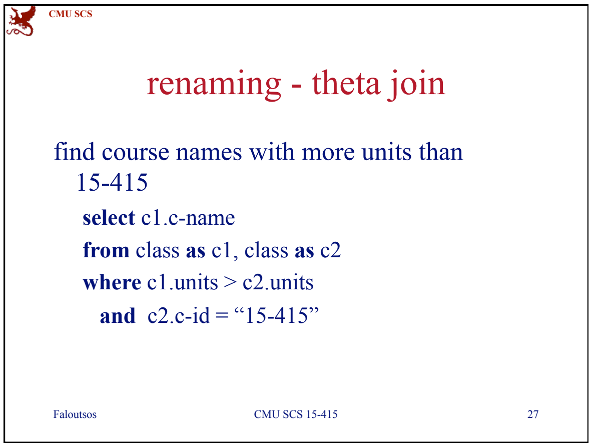


CMU SCS

renaming - theta join

find course names with more units than 15-415

```
select c1.c-name
from class as c1, class as c2
where c1.units > c2.units
  and  c2.c-id = “15-415”
```

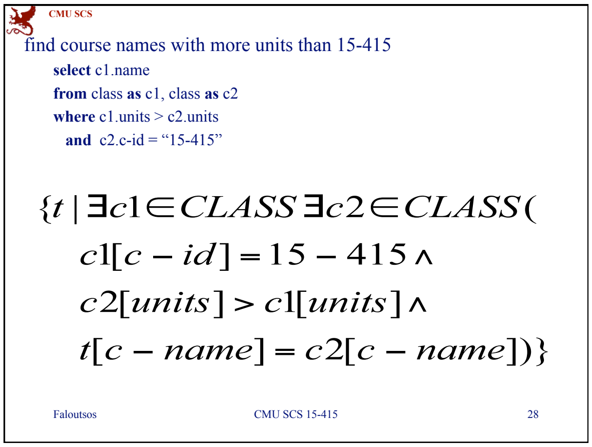


CMU SCS

renaming - theta join

find course names with more units than 15-415

```
select c1.c-name
from class as c1, class as c2
where c1.units > c2.units
  and  c2.c-id = “15-415”
```

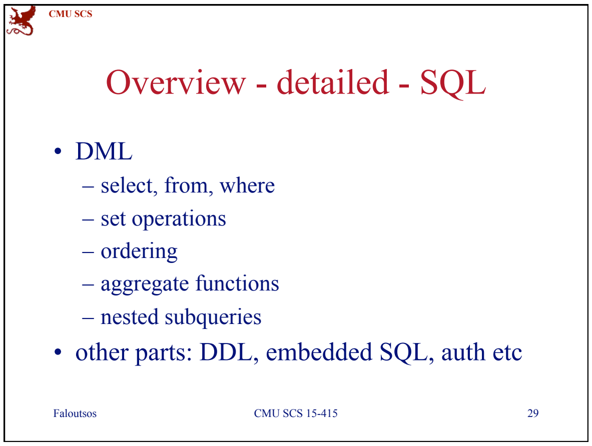


CMU SCS

find course names with more units than 15-415

```
select c1.name
from class as c1, class as c2
where c1.units > c2.units
  and  c2.c-id = “15-415”
```

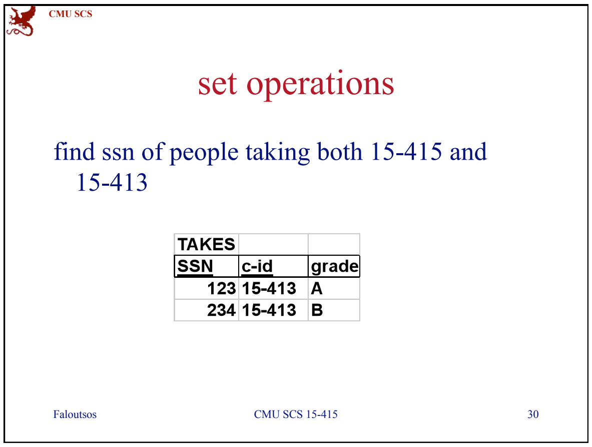
$$\{t \mid \exists c1 \in CLASS \, \exists c2 \in CLASS (\\ c1[c-id] = 15-415 \wedge \\ c2[units] > c1[units] \wedge \\ t[c-name] = c2[c-name])\}$$



CMU SCS

Overview - detailed - SQL

- DML
  - select, from, where
  - set operations
  - ordering
  - aggregate functions
  - nested subqueries
- other parts: DDL, embedded SQL, auth etc

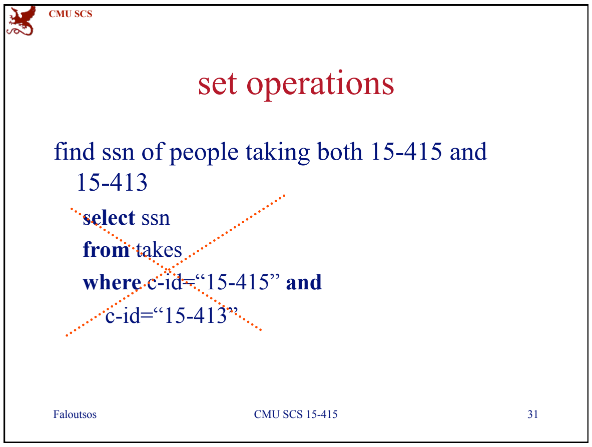


CMU SCS

set operations

find ssn of people taking both 15-415 and 15-413

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

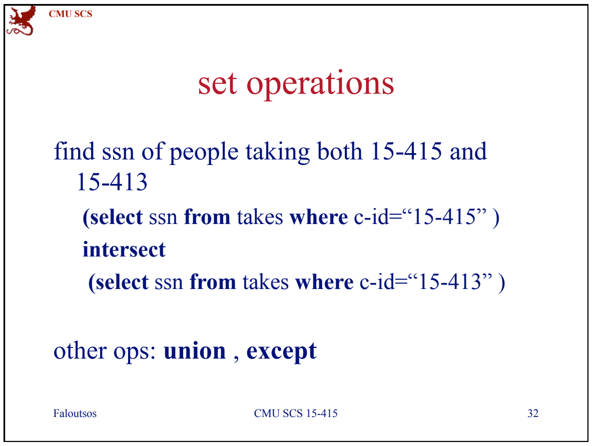


CMU SCS

set operations

find ssn of people taking both 15-415 and 15-413

~~**select** ssn~~
~~**from** takes~~
~~**where** c-id="15-415" **and**~~
~~c-id="15-413"~~



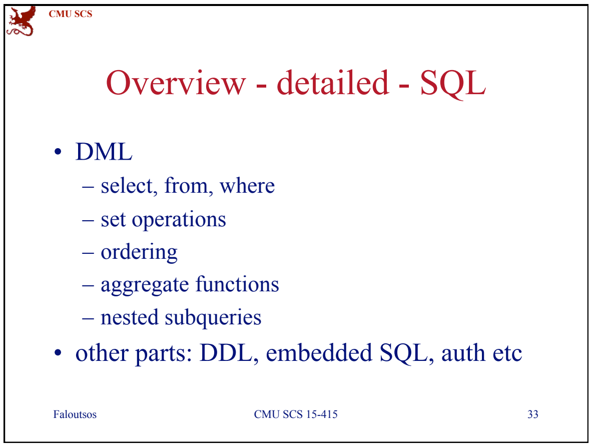
CMU SCS

set operations

find ssn of people taking both 15-415 and 15-413

(**select** ssn **from** takes **where** c-id="15-415")
intersect
(**select** ssn **from** takes **where** c-id="15-413")

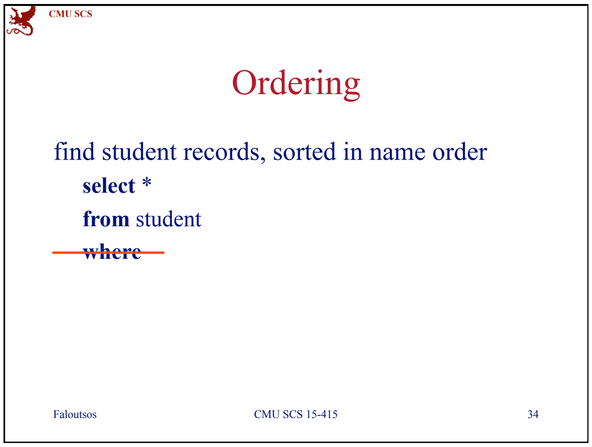
other ops: **union** , **except**



CMU SCS

Overview - detailed - SQL

- DML
  - select, from, where
  - set operations
  - ordering
  - aggregate functions
  - nested subqueries
- other parts: DDL, embedded SQL, auth etc



CMU SCS

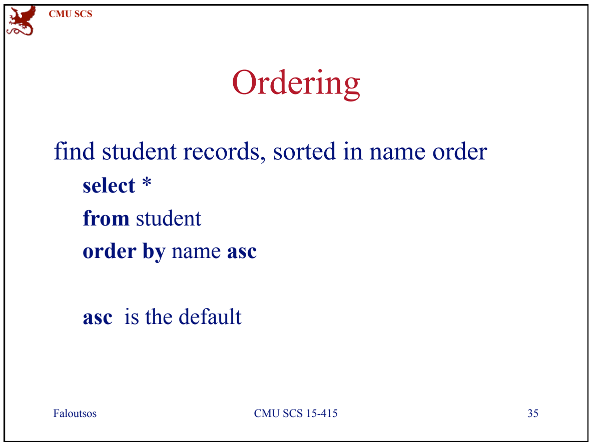
Ordering

find student records, sorted in name order

select *

from student

~~**where**~~



CMU SCS

Ordering

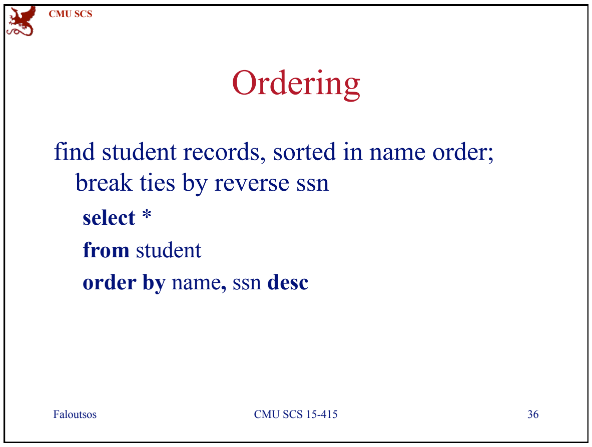
find student records, sorted in name order

select *

from student

order by name **asc**

asc is the default



CMU SCS

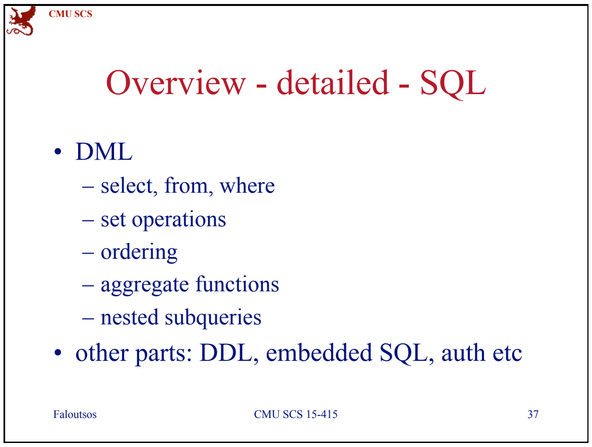
Ordering

find student records, sorted in name order; break ties by reverse ssn

select *

from student

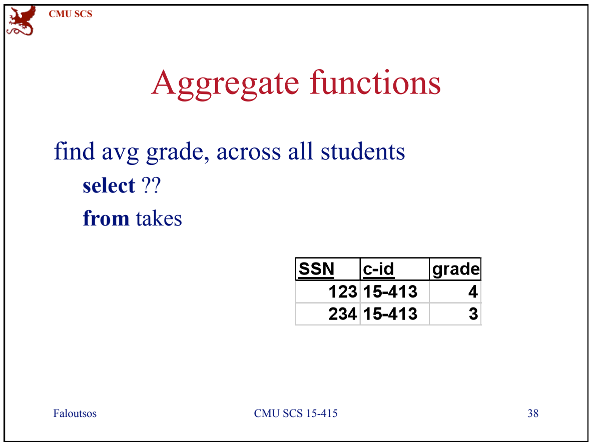
order by name, ssn **desc**



CMU SCS

Overview - detailed - SQL

- DML
  - select, from, where
  - set operations
  - ordering
  - aggregate functions
  - nested subqueries
- other parts: DDL, embedded SQL, auth etc



CMU SCS

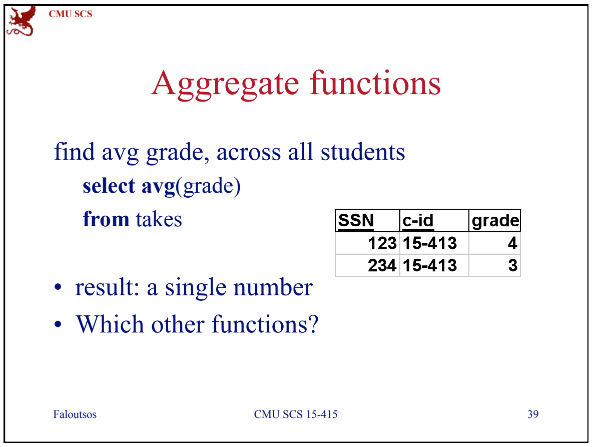
Aggregate functions

find avg grade, across all students

select ??

from takes

SSN	c-id	grade
123	15-413	4
234	15-413	3



CMU SCS

Aggregate functions

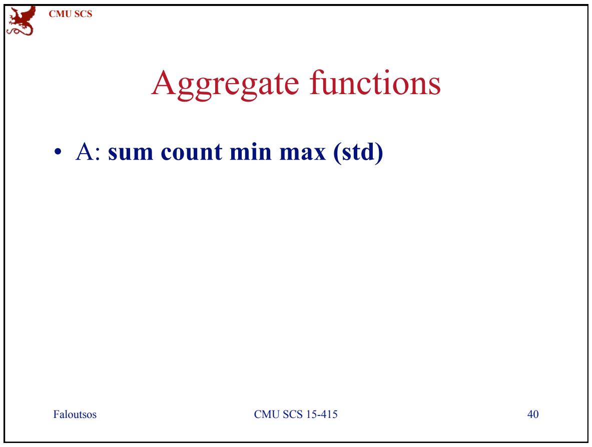
find avg grade, across all students

select avg(grade)

from takes

SSN	c-id	grade
123	15-413	4
234	15-413	3

- result: a single number
- Which other functions?

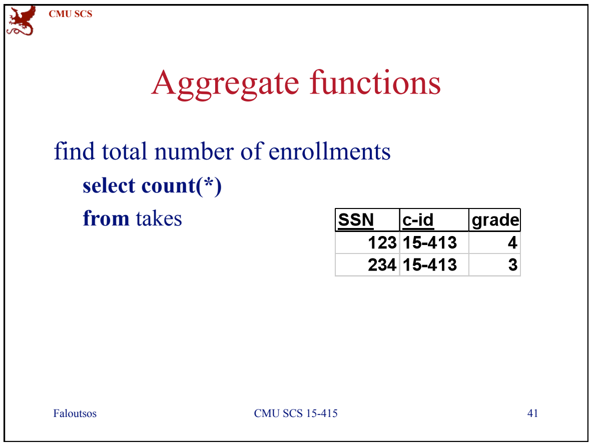


CMU SCS

Aggregate functions

- A: **sum count min max (std)**

Faloutsos CMU SCS 15-415 40



CMU SCS

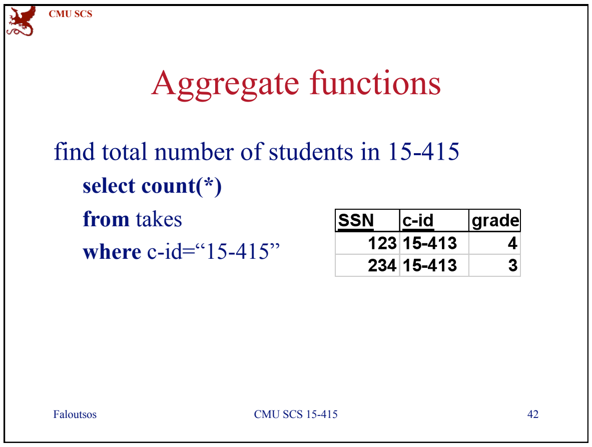
Aggregate functions

find total number of enrollments

select count(*)
from takes

SSN	c-id	grade
123	15-413	4
234	15-413	3

Faloutsos CMU SCS 15-415 41



CMU SCS

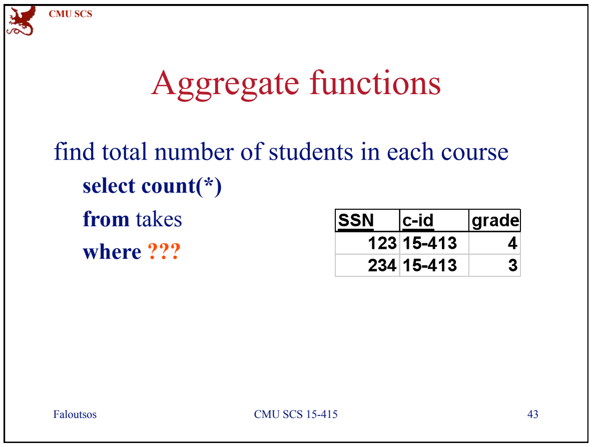
Aggregate functions

find total number of students in 15-415

select count(*)
from takes
where c-id="15-415"

SSN	c-id	grade
123	15-413	4
234	15-413	3

Faloutsos CMU SCS 15-415 42



CMU SCS

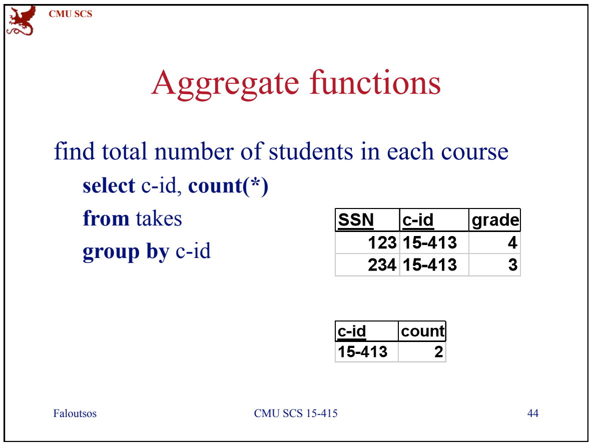
Aggregate functions

find total number of students in each course

select count(*)
from takes
where ???

SSN	c-id	grade
123	15-413	4
234	15-413	3

Faloutsos CMU SCS 15-415 43



CMU SCS

Aggregate functions

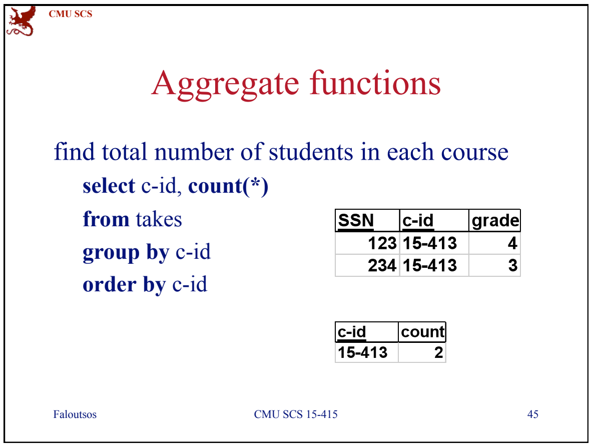
find total number of students in each course

select c-id, **count(*)**
from takes
group by c-id

SSN	c-id	grade
123	15-413	4
234	15-413	3

c-id	count
15-413	2

Faloutsos CMU SCS 15-415 44



CMU SCS

Aggregate functions

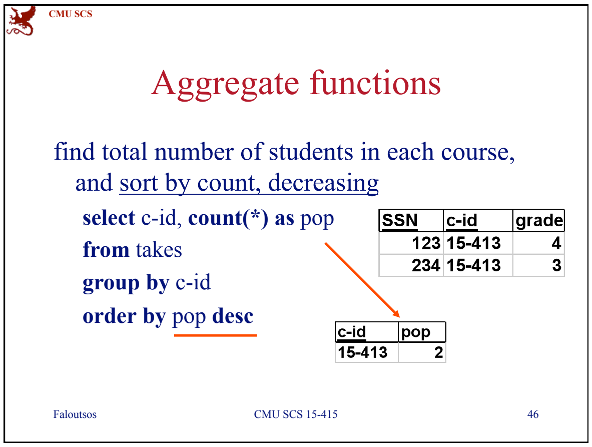
find total number of students in each course

select c-id, **count(*)**
from takes
group by c-id
order by c-id

SSN	c-id	grade
123	15-413	4
234	15-413	3

c-id	count
15-413	2

Faloutsos CMU SCS 15-415 45



CMU SCS

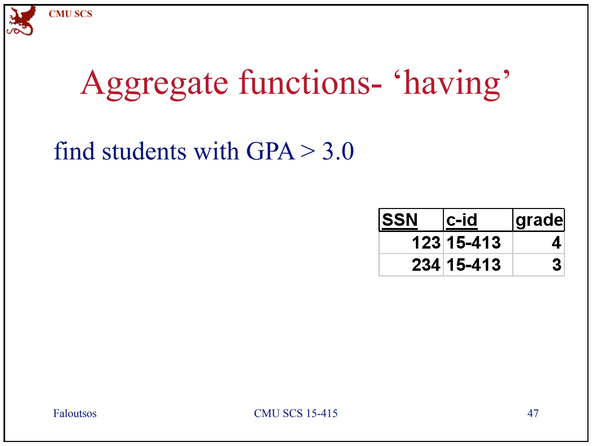
Aggregate functions

find total number of students in each course, and sort by count, decreasing

```
select c-id, count(*) as pop
from takes
group by c-id
order by pop desc
```

SSN	c-id	grade
123	15-413	4
234	15-413	3

c-id	pop
15-413	2

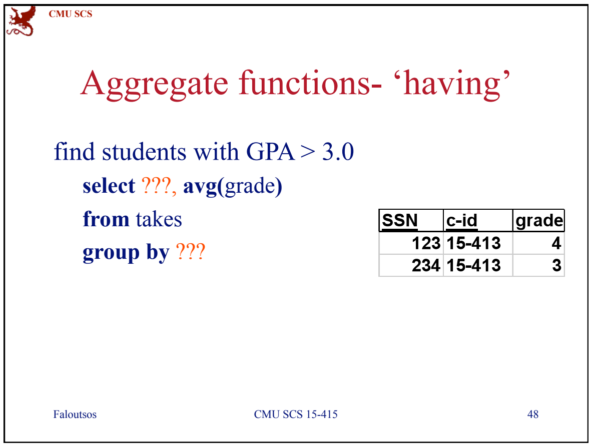


CMU SCS

Aggregate functions- ‘having’

find students with GPA > 3.0

SSN	c-id	grade
123	15-413	4
234	15-413	3



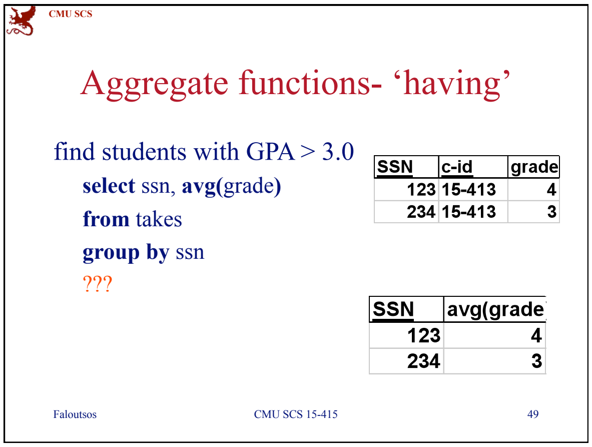
CMU SCS

Aggregate functions- ‘having’

find students with GPA > 3.0

```
select ???, avg(grade)
from takes
group by ???
```

SSN	c-id	grade
123	15-413	4
234	15-413	3



CMU SCS

Aggregate functions- ‘having’

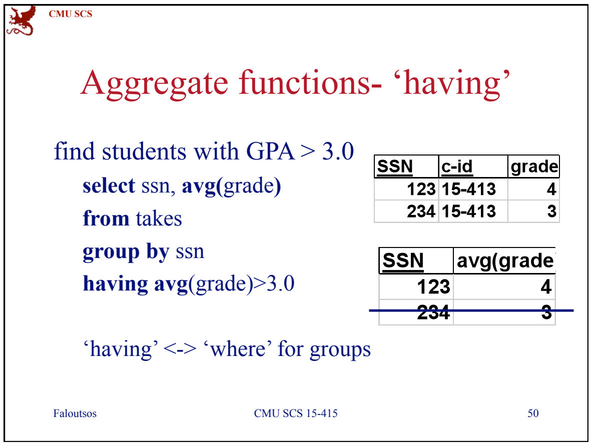
find students with GPA > 3.0

select ssn, **avg**(grade**)**
from takes
group by ssn
???

SSN	c-id	grade
123	15-413	4
234	15-413	3

SSN	avg(grade
123	4
234	3

Faloutsos CMU SCS 15-415 49



CMU SCS

Aggregate functions- ‘having’

find students with GPA > 3.0

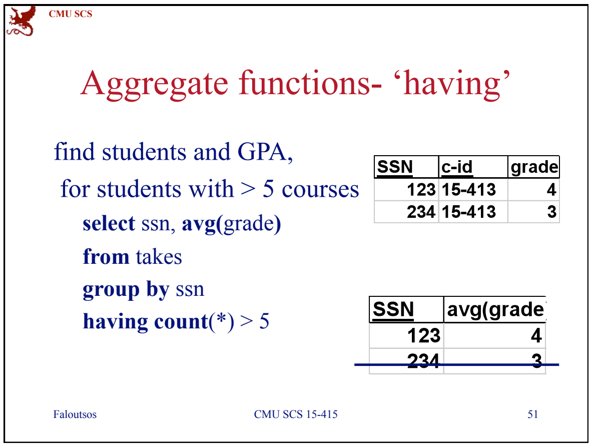
select ssn, **avg**(grade**)**
from takes
group by ssn
having avg(grade)>3.0

SSN	c-id	grade
123	15-413	4
234	15-413	3

SSN	avg(grade
123	4
~~234~~	~~3~~

‘having’ <-> ‘where’ for groups

Faloutsos CMU SCS 15-415 50



CMU SCS

Aggregate functions- ‘having’

find students and GPA,

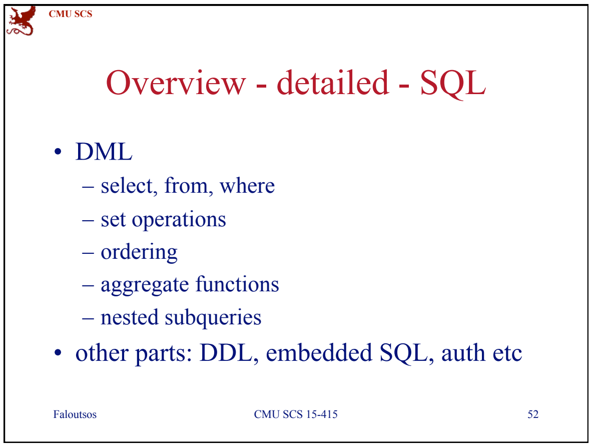
for students with > 5 courses

select ssn, **avg**(grade**)**
from takes
group by ssn
having count(*) > 5

SSN	c-id	grade
123	15-413	4
234	15-413	3

SSN	avg(grade
123	4
~~234~~	~~3~~

Faloutsos CMU SCS 15-415 51



CMU SCS

Overview - detailed - SQL

- DML
  - select, from, where
  - set operations
  - ordering
  - aggregate functions
  - nested subqueries
- other parts: DDL, embedded SQL, auth etc