L07: SQL: Advanced & Practice

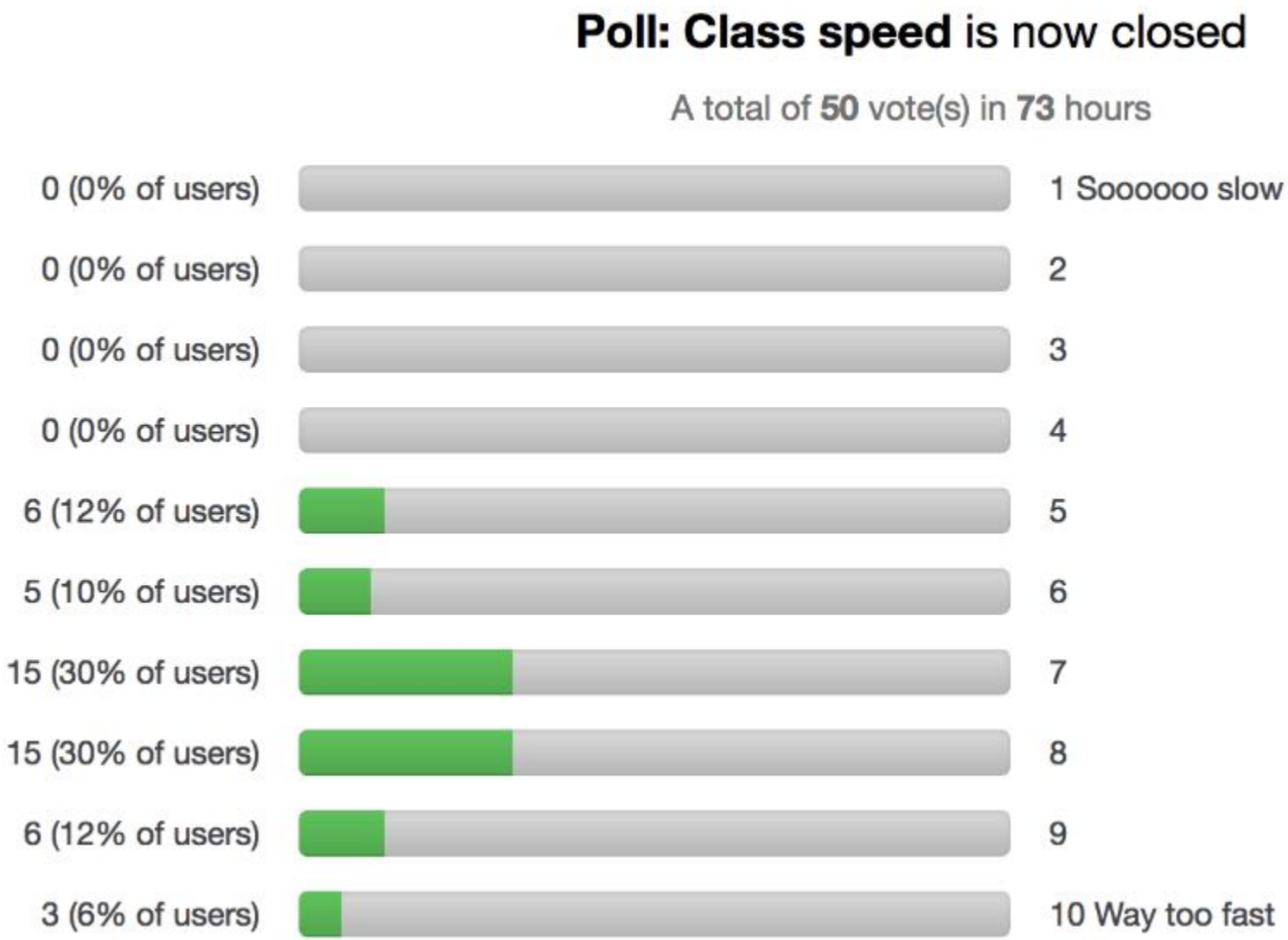
CS3200 Database design (sp18 s2)

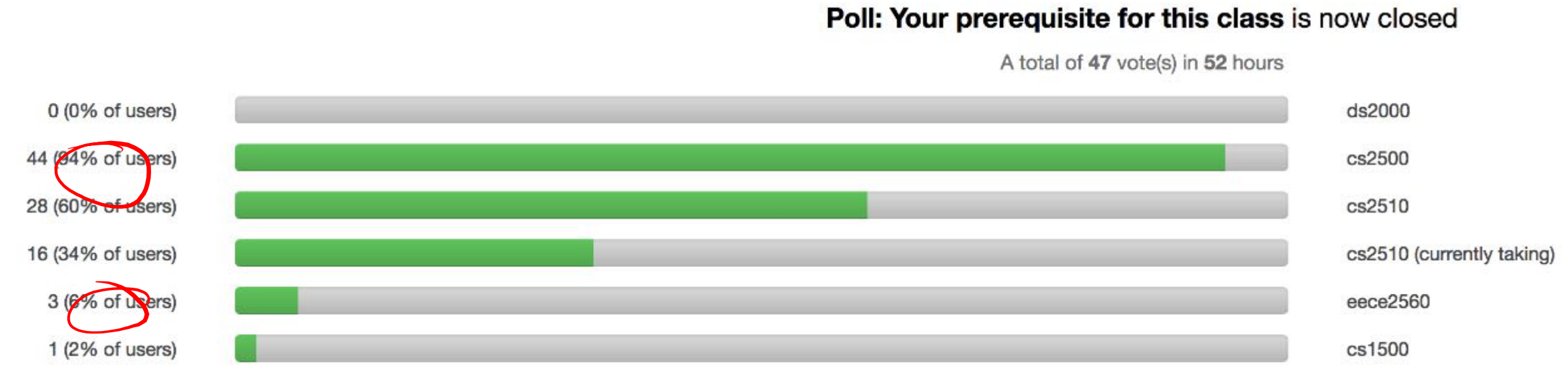
1/11/2018

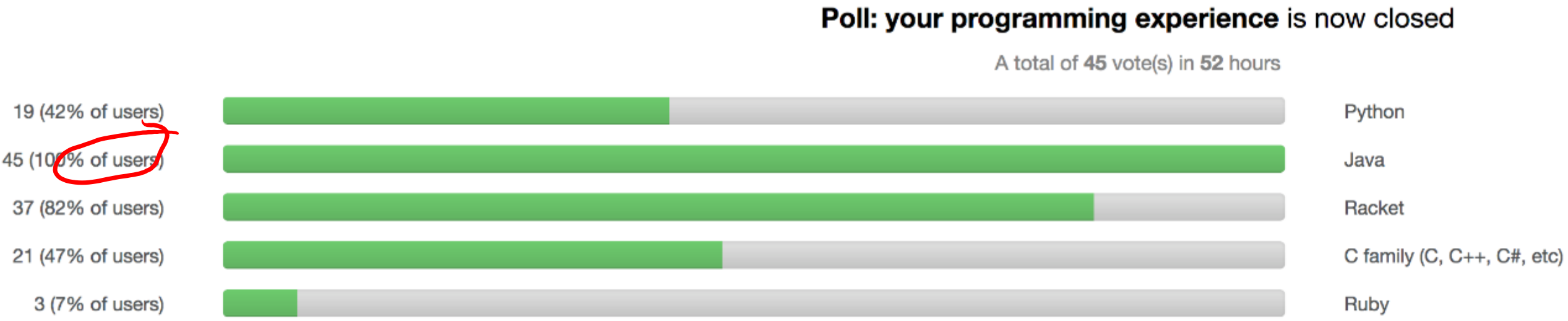
Announcements!

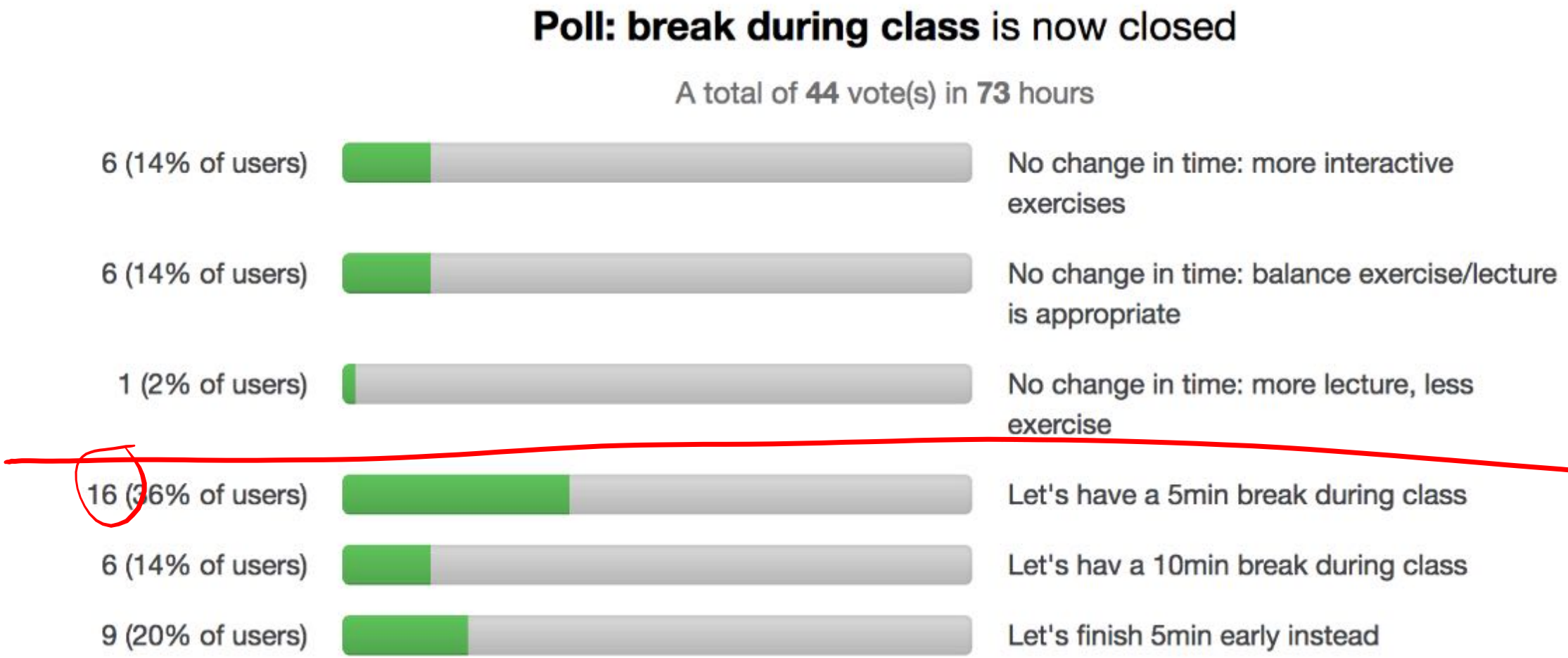
- Please pick up your name card
  - always bring it
  - move closer to the front
- HW3 will be again SQL
  - individual submission, but discussion in teams
- Exam on laptop in class
  - practice exam in class next week Thursday
- Polls

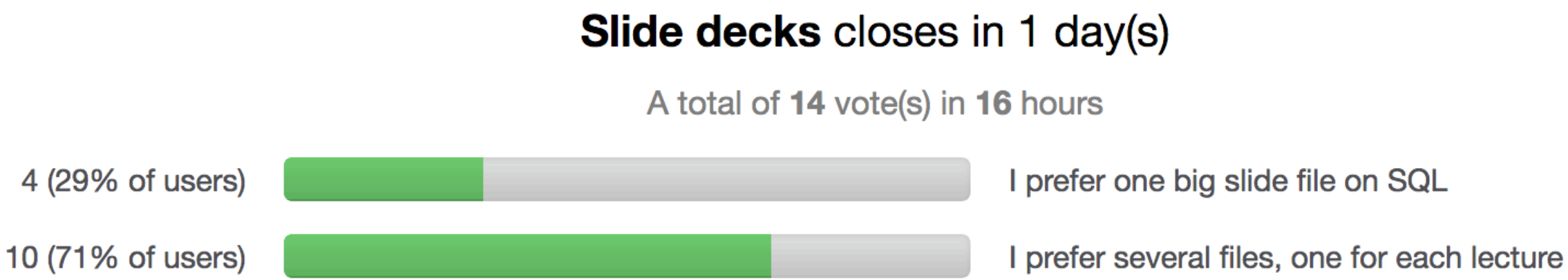
- Outline today:
  - HW1 together
  - outer joins, nulls


Poll: Class speed is now closed
A total of 50 vote(s) in 73 hours
0 (0% of users)
1 Soooooo slow
0 (0% of users)
2
0 (0% of users)
3
0 (0% of users)
4
6 (12% of users)
5
5 (10% of users)
6
15 (30% of users)
7
15 (30% of users)
8
6 (12% of users)
9
3 (6% of users)
10 Way too fast


Poll: Your prerequisite for this class is now closed
A total of 47 vote(s) in 52 hours
0 (0% of users)
ds2000
44 (94% of users)
cs2500
28 (60% of users)
cs2510
16 (34% of users)
cs2510 (currently taking)
3 (6% of users)
eece2560
1 (2% of users)
cs1500


Poll: your programming experience is now closed
A total of 45 vote(s) in 52 hours
19 (42% of users)
Python
45 (100% of users)
Java
37 (82% of users)
Racket
21 (47% of users)
C family (C, C++, C#, etc)
3 (7% of users)
Ruby


Poll: break during class is now closed
A total of 44 vote(s) in 73 hours
6 (14% of users)
No change in time: more interactive exercises
6 (14% of users)
No change in time: balance exercise/lecture is appropriate
1 (2% of users)
No change in time: more lecture, less exercise
16 (36% of users)
Let's have a 5min break during class
6 (14% of users)
Let's hav a 10min break during class
9 (20% of users)
Let's finish 5min early instead


Slide decks closes in 1 day(s)
A total of **14** vote(s) in **16** hours
4 (29% of users)
I prefer one big slide file on SQL
10 (71% of users)
I prefer several files, one for each lecture

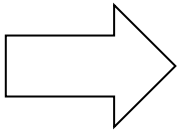
More on WITH

Recall: Witnesses: with aggregates per group (8/8)

Second: How to get the product that is sold with max sales?

Purchase

Product	Price	Quantity
Bagel	3	20
Bagel	2	20
Banana	1	50
Banana	2	10
Banana	4	10



Product	sales
Banana	70

```
SELECT    product, sum(quantity) as sales
FROM      Purchase
GROUP BY product
HAVING    sum(quantity) =
          (SELECT max (Q)
          FROM
                (SELECT    sum(quantity) Q
                FROM       Purchase
                GROUP BY product) X
          )
```

WITH clause


```
WITH X AS
        (SELECT   product, SUM(quantity) sales
        FROM      Purchase
        GROUP BY product)
```

```
SELECT    product, sum(quantity) as sales
FROM      Purchase
GROUP BY product
HAVING    sum(quantity) =
          (SELECT max (Q)
          FROM   (SELECT    sum(quantity) Q
                  FROM      Purchase
                  GROUP BY product) X )
```

WITH clause


```
WITH X AS
            (SELECT   product, SUM(quantity) sales
            FROM      Purchase
            GROUP BY product)
SELECT      *
FROM        X
WHERE
```

```
SELECT      product, sum(quantity) as sales
FROM        Purchase
GROUP BY product
HAVING      sum(quantity) =
            (SELECT max (Q)
            FROM    (SELECT     sum(quantity) Q
                     FROM       Purchase
                     GROUP BY product) X )
```

WITH clause


```
WITH X AS
            (SELECT   product, SUM(quantity) sales
            FROM      Purchase
            GROUP BY product)
SELECT      *
FROM        X
WHERE       sales =
            (SELECT   MAX (sales)
            FROM      X)
```

```
SELECT      product, sum(quantity) as sales
FROM        Purchase
GROUP BY product
HAVING      sum(quantity) =
            (SELECT max (Q)
            FROM
                  (SELECT     sum(quantity) Q
                  FROM        Purchase
                  GROUP BY product) X
            )
```

WITH clause


```
WITH X AS
            (SELECT   product, SUM(quantity) sales
            FROM      Purchase
            GROUP BY product)
Y AS
            (SELECT   MAX (sales) maxs
            FROM      X)
SELECT      *
FROM        X
WHERE       sales = (SELECT maxs FROM Y))
```

```
SELECT      product, sum(quantity) as sales
FROM        Purchase
GROUP BY product
HAVING      sum(quantity) =
            (SELECT max (Q)
            FROM    (SELECT     sum(quantity) Q
                     FROM       Purchase
                     GROUP BY product) X )
```

Inner Joins vs. Outer Joins

Illustration

361

English

eText	eid
One	1
Two	2
Three	3
Four	4
Five	5
Six	6

French

fid	fText
1	Un
3	Trois
4	Quatre
5	Cinq
6	Siz
7	Sept
8	Huit

An "inner join":

```
SELECT *
FROM   English, French
WHERE  eid = fid
```

etext	eid	fid	ftext
One	1	1	Un
Three	3	3	Trois
Four	4	4	Quatre
Five	5	5	Cinq
Six	6	6	Siz

Same as:

```
SELECT *
FROM   English JOIN French
ON     eid = fid
```

"JOIN"
same as
"INNER JOIN"

Illustration

English

eText	eid
One	1
Two	2
Three	3
Four	4
Five	5
Six	6

French

fid	fText
1	Un
3	Trois
4	Quatre
5	Cinq
6	Siz
7	Sept
8	Huit

"FULL JOIN"
same as
"FULL OUTER JOIN"

```
SELECT *
FROM   English FULL JOIN French
ON     English.eid = French.fid
```

etext	eid	fid	ftext
One	1	1	Un
Two	2	NULL	NULL
Three	3	3	Trois
Four	4	4	Quatre
Five	5	5	Cinq
Six	6	6	Siz
NULL	NULL	7	Sept
NULL	NULL	8	Huit

~~
```
SELECT *
FROM   English JOIN French
ON     eid = fid
```
~~

SQLite does not support "FULL OUTER JOIN"s ☹ (but "LEFT JOIN")

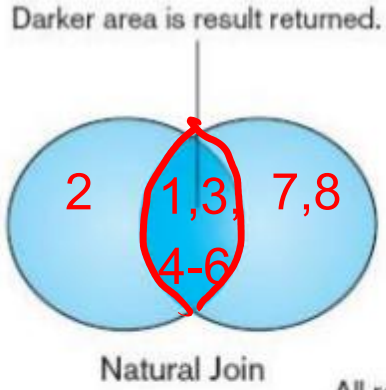
Illustration

English

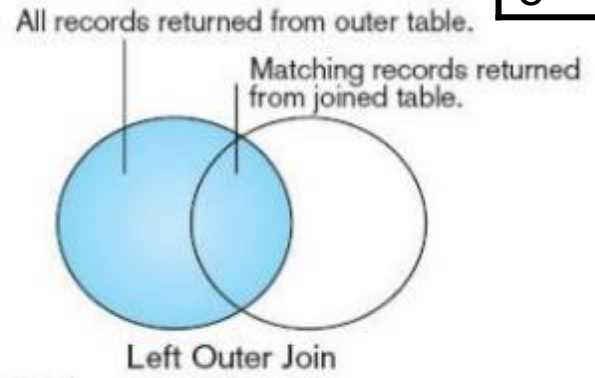
eText	eid
One	1
Two	2
Three	3
Four	4
Five	5
Six	6

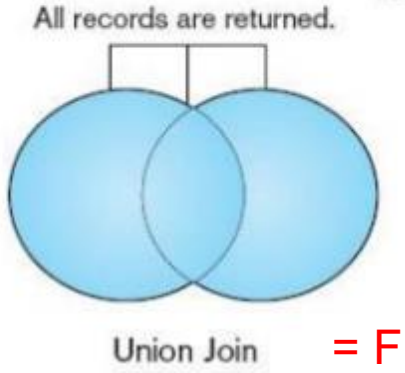
French

fid	fText
1	Un
3	Trois
4	Quatre
5	Cinq
6	Siz
7	Sept
8	Huit



= (INNER) JOIN

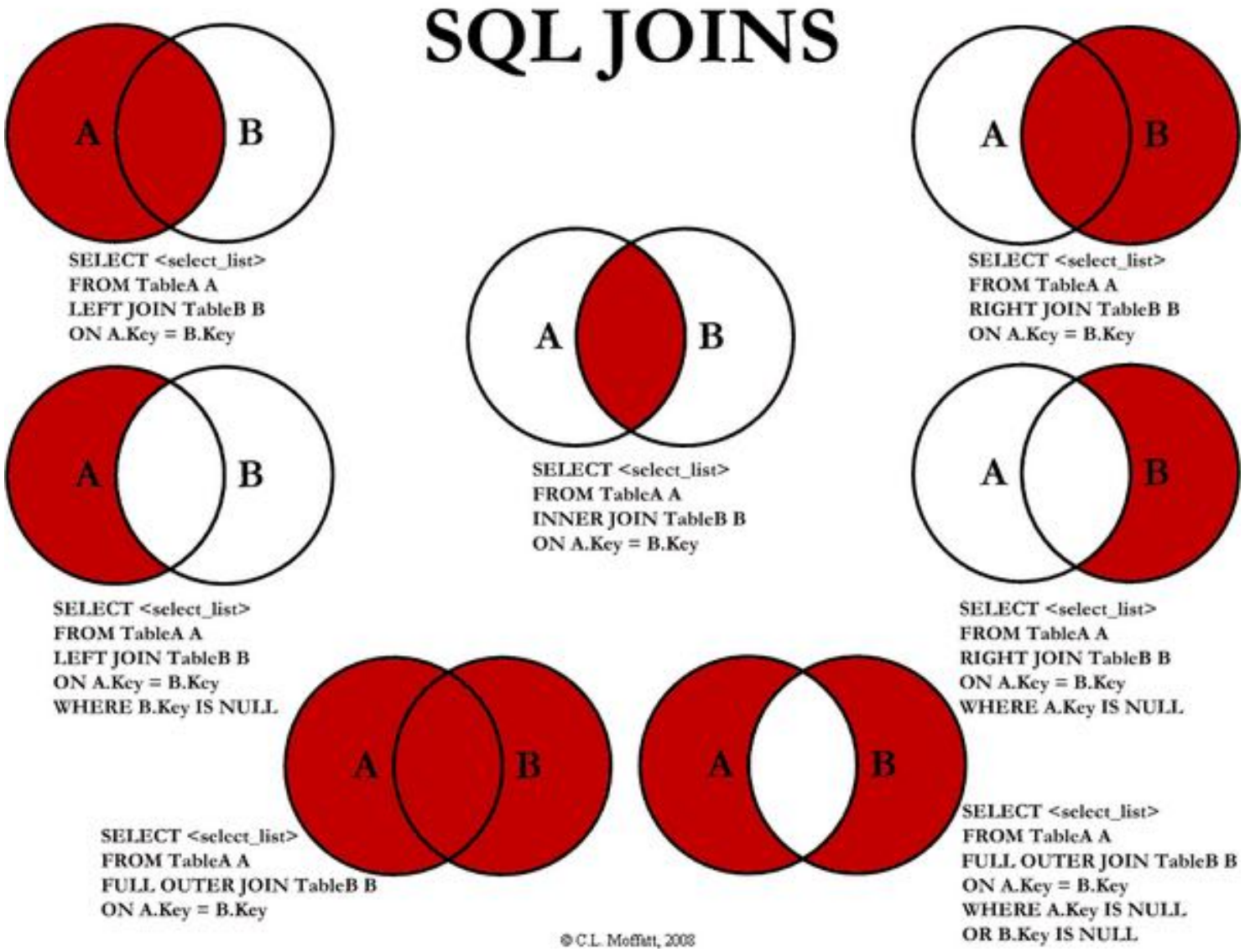




= FULL (OUTER) JOIN

Source: Fig. 7-2, Hoffer et al., Modern Database Management, 10ed ed, 2011.

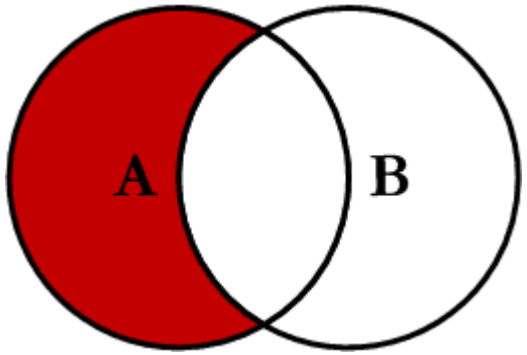
Detailed Illustration with Examples (follow the link)



Check this web page for illustrating examples

Source: http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins

Let's practice



English

eText	eid
One	1
Two	2
Three	3
Four	4
Five	5
Six	6

French

fid	fText
1	Un
3	Trois
4	Quatre
5	Cinq
6	Siz
7	Sept
8	Huit

```
SELECT <select_list>
FROM A
LEFT JOIN B
ON A.key = B.key
WHERE B.key IS NULL
```

```
SELECT *
FROM English
WHERE eid NOT IN
   (SELECT fid
   FROM French)
```

```
SELECT eText, eid
FROM English
LEFT JOIN French
ON eid = fid
WHERE fid IS NULL
```

Source: http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins

Missing sales

Item(name, category)
Purchase2(iName, store, month)

334

An "inner join":

```
SELECT  Item.name, Purchase2.store
FROM    Item, Purchase2
WHERE   Item.name = Purchase2.iName
```

Same as:

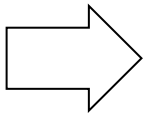
```
SELECT  Item.name, Purchase2.store
FROM    Item JOIN Purchase2 ON
        Item.name = Purchase2.iName
```

Item

Name	Category
Gizmo	Gadget
Camera	Photo
OneClick	Photo

Purchase2

iName	Store	Month
Gizmo	Wiz	8
Camera	Ritz	8
Camera	Wiz	9



Result

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Missing sales

Item(name, category)
Purchase2(iName, store, month)

An "inner join":

```
SELECT  Item.name, Purchase2.store
FROM    Item, Purchase2
WHERE   Item.name = Purchase2.iName
```

Same as:

```
SELECT  Item.name, Purchase2.store
FROM    Item INNER JOIN Purchase2 ON
        Item.name = Purchase2.iName
```

"INNER JOIN" same as "JOIN"

Item

Name	Category
Gizmo	Gadget
Camera	Photo
OneClick	Photo

Purchase2

iName	Store	Month
Gizmo	Wiz	8
Camera	Ritz	8
Camera	Wiz	9

Result

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Products that never sold will be lost ☹

Outer Joins

Item(<u>name</u>, category)
Purchase2(iName, store, month)

"LEFT OUTER JOIN"
same as
"LEFT JOIN"

If we want to include the never-sold products, then we need an "outer join":

```
SELECT  Item.name, Purchase2.store
FROM    Item LEFT OUTER JOIN Purchase2 ON
        Item.name = Purchase2.iName
```

Item

Name	Category
Gizmo	Gadget
Camera	Photo
OneClick	Photo

Purchase2

iName	Store	Month
Gizmo	Wiz	8
Camera	Ritz	8
Camera	Wiz	9

Result

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz
OneClick	NULL

Now we include those products ☺

Outer Joins

Item(name, category)
Purchase2(iName, store, month)

334

Explanation: the filter ("month = 9") applies of the result of the outer join. Any tuple that has NULL as month, does not pass the filter

Same question, but now only for those sold in month = 9:

```
SELECT  Item.name, Purchase2.store
FROM    Item LEFT OUTER JOIN Purchase2 ON
        Item.name = Purchase2.iName
WHERE   month = 9
```

Item

Name	Category
Gizmo	Gadget
Camera	Photo
OneClick	Photo

Purchase2

iName	Store	Month
Gizmo	Wiz	8
Camera	Ritz	8
Camera	Wiz	9

Result

Name	Store
Camera	Wiz

The products disappeared *despite* outer join ☹

Outer Joins

Item(name, category)
Purchase2(iName, store, month)

Explanation: now the filter ("month = 9") applies to the right side of the left join *before* joining. NULLs are appended only after filter, during join

Same question, but now only for those sold in month = 9:

```
SELECT  Item.name, Purchase2.store
FROM    Item LEFT OUTER JOIN Purchase2 ON
        (Item.name = Purchase2.iName
AND     month = 9)
```

parenthesis not required, and just for illustration

Item

Name	Category
Gizmo	Gadget
Camera	Photo
OneClick	Photo

Purchase2

iName	Store	Month
Gizmo	Wiz	8
Camera	Ritz	8
Camera	Wiz	9

Result

Name	Store
Camera	Wiz
Gizmo	NULL
OneClick	NULL

Now they are back again ☺

Empty Group Problem

Item(name, category)
Purchase2(iName, store, month)

Compute, for each product, the total number of sales in Sept (= month 9)

```
SELECT      name, count(*)
FROM        Item, Purchase2
WHERE       name = iName
    and     month = 9
GROUP BY    name
```

What's wrong?

Empty Group Problem

Item(name, category)
Purchase2(iName, store, month)

334

Compute, for each product, the total number of sales in Sept (= month 9)

We need to use an attribute from "Purchase2" to get the correct 0 count. Try "name" from "Item".

```
SELECT    name, count(store)
FROM      Item LEFT JOIN Purchase2 ON
          name = iName
    and   month = 9
GROUP BY  name
```

Now we also get the products with 0 sales

Empty Group Problem

Item(name, category)
Purchase2(iName, store, month)

334

Compute, for each product, the total number of sales in Sept (= month 9)

We need to use an attribute from "Purchase2" to get the correct 0 count. Try "name" from "Item".

```
SELECT    x.name, count(y.store)
FROM      Item x LEFT OUTER JOIN Purchase2 y ON
          x.name = y.iName
    and   y.month = 9
GROUP BY  x.name
```

Now we also get the products with 0 sales

Outer Joins: summary

- Left (outer) join:
  - Include the left tuple even if there's no match
- Right (outer) join:
  - Include the right tuple even if there's no match
- Full (outer) join:
  - Include both left and right tuples even if there's no match

- (inner) join:
  - Include only the matches

Processing Multiple Tables–Joins

- **Join**: a relational operation that causes two or more tables with a common domain to be combined into a single table or view
- **Equi-join**: a join in which the joining condition is based on equality between values in the common columns; common columns appear redundantly in the result table
- A **Theta-join** allows for arbitrary comparison relationships (e.g., ≥). An equijoin is a theta join using the equality operator.
- **Natural join**: an equi-join in which one of the duplicate columns is eliminated in the result table

The common columns in joined tables are usually the primary key of the dominant table and the foreign key of the dependent table in 1:M relationships

Processing Multiple Tables–Joins

- **Left Outer join**: a join in which rows from the left table that do not have matching values in common columns are nonetheless included in the result table (as opposed to inner join, in which rows must have matching values in order to appear in the result table)

- Union join ("**Full outer join**"): includes all columns from each table in the join, and an instance for each row of each table

More Practice

1. What does this query return?

Actor

id	name	gender
1	Alice	f
2	Bob	m

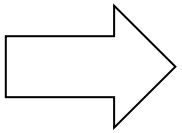
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  movie.name
FROM    movie, casts, actor
WHERE   casts.aid = actor.id
  AND   movie.id = casts.mid
  AND   actor.id NOT IN
        (SELECT  a2.id
        FROM     actor a2
        WHERE    a2.gender = 'm')
```



?

1. What does this query return?

Actor

id	name	gender
1	Alice	f
2	Bob	m

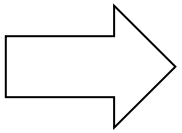
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  movie.name
FROM    movie, casts, actor
WHERE   casts.aid = actor.id
  AND   movie.id = casts.mid
  AND   actor.id NOT IN
        (SELECT  a2.id
        FROM     actor a2
        WHERE    a2.gender = 'm')
```



name
Kill Bill

1. The "Universal Relation"

Select *

id	name	gender	aid	mid	role	id	name
1	Alice	f	1	1	role 1	1	Kill Bill
2	Bob	m	2	1	role 2	1	Kill Bill
2	Bob	m	2	1	role 3	1	Kill Bill

```
SELECT *
FROM   movie, casts, actor
WHERE  casts.aid = actor.id
  AND  movie.id = casts.mid
```

```
SELECT a2.id
FROM   actor a2
WHERE  a2.gender = 'm'
```

id
2

More information about the "universal relation": https://en.wikipedia.org/wiki/Universal_relation_assumption

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name
FROM
WHERE
GROUP BY
```

?

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

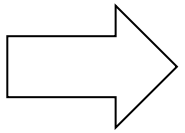
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
GROUP BY m.name
```



name	(no name)
Kill Bill	4

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

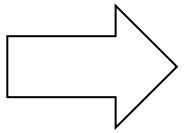
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
GROUP BY m.id
```



?

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

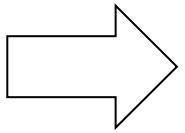
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
GROUP BY m.id
```



?

Error on some databases!
(E.g. Microsoft SQLserver)

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

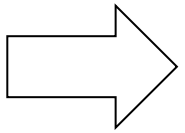
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
GROUP BY m.id
```



name	(no name)
Kill Bill	3
Kill Bill	1

?

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

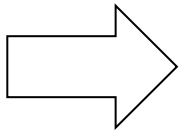
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
GROUP BY m.id, m.name
```



name	(no name)
Kill Bill	3
Kill Bill	1

2. How to get the number of casts for each movie?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

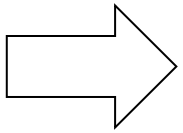
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(distinct c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
GROUP BY m.id, m.name
```



name	(no name)
Kill Bill	2
Kill Bill	1

3. How to get the number of casts for '%Bill%'?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(distinct c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid

GROUP BY m.id, m.name
```

?

3. How to get the number of casts for '%Bill%'?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  m.name, count(distinct c.aid)
FROM    movie m, casts c
WHERE   m.id = c.mid
   AND  m.name like '%Bill%'
GROUP BY m.id, m.name
```

4. How to get the number of casts for each actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT
FROM
WHERE
GROUP BY
```

?

name	(no name)
Alice	1
Bob	2
Charly	1

4. How to get the number of casts for each actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

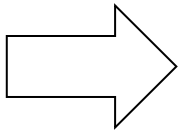
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  a.name, count(*)
FROM    actor a, casts c
WHERE   a.id = c.aid
GROUP BY a.id, a.name
```



name	(no name)
Alice	1
Bob	2
Charly	1

4. How to get the number of casts for each actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  a.name, count(distinct c.aid)
FROM    actor a, casts c
WHERE   a.id = c.aid
GROUP BY a.id, a.name
```

⇨ ?

4. How to get the number of casts for each actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  a.name, count(distinct c.aid)
FROM    actor a, casts c
WHERE   a.id = c.aid
GROUP BY a.id, a.name
```

name	(no name)
Alice	1
Bob	1
Charly	1

Will always show 1 (since there is only one distinct aid per group grouped by aid *by def.*)

4. How to get the number of movies for each actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT
FROM
WHERE
GROUP BY
```

?

name	(no name)
Alice	1
Bob	1
Charly	1

4. How to get the number of movies for each actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

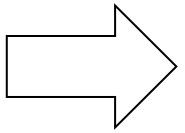
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  a.name, count(distinct c.mid)
FROM    actor a, casts c
WHERE   a.id = c.aid
GROUP BY a.id, a.name
```



name	(no name)
Alice	1
Bob	1
Charly	1

4. How to get the number of casts for each male actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

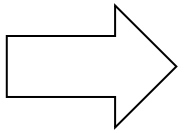
aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT
FROM
WHERE
      AND
GROUP BY
```

?



name	(no name)
Bob	2
Charly	1

4. How to get the number of casts for each male actor?

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

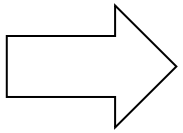
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  a.name, count(distinct a.id)
FROM    actor a, casts c
WHERE   a.id = c.aid
        AND a.gender = 'm'
GROUP BY a.id, a.name
```



name	(no name)
Bob	2
Charly	1

5. How to get male actors with number of casts> 1

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

Casts

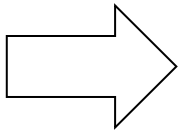
aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT
FROM
WHERE
      AND
GROUP BY
```

?



name	(no name)
Bob	2

5. How to get male actors with number of casts> 1

Actor

id	name	gender
1	Alice	f
2	Bob	m
3	Charly	m

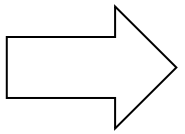
Casts

aid	mid	role
1	1	role 1
2	1	role 2
2	1	role 3
3	2	role 4

Movie

id	name
1	Kill Bill
2	Kill Bill

```
SELECT  a.name, count(*)
FROM    actor a, casts c
WHERE   a.id = c.aid
        AND a.gender = 'm'
GROUP BY a.id, a.name
HAVING count(*) > 1
```



name	(no name)
Bob	2

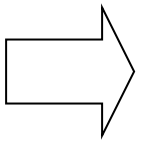
SQL injection

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT  PName
FROM    Product
WHERE   category='Gadgets'
```



PName
Gizmo
Powergizmo

Parameterized SQL Query

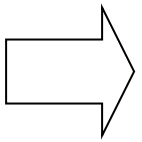
Varies between DBMSs. The following is the semantics for SQL server. You do not need to know that for exams

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

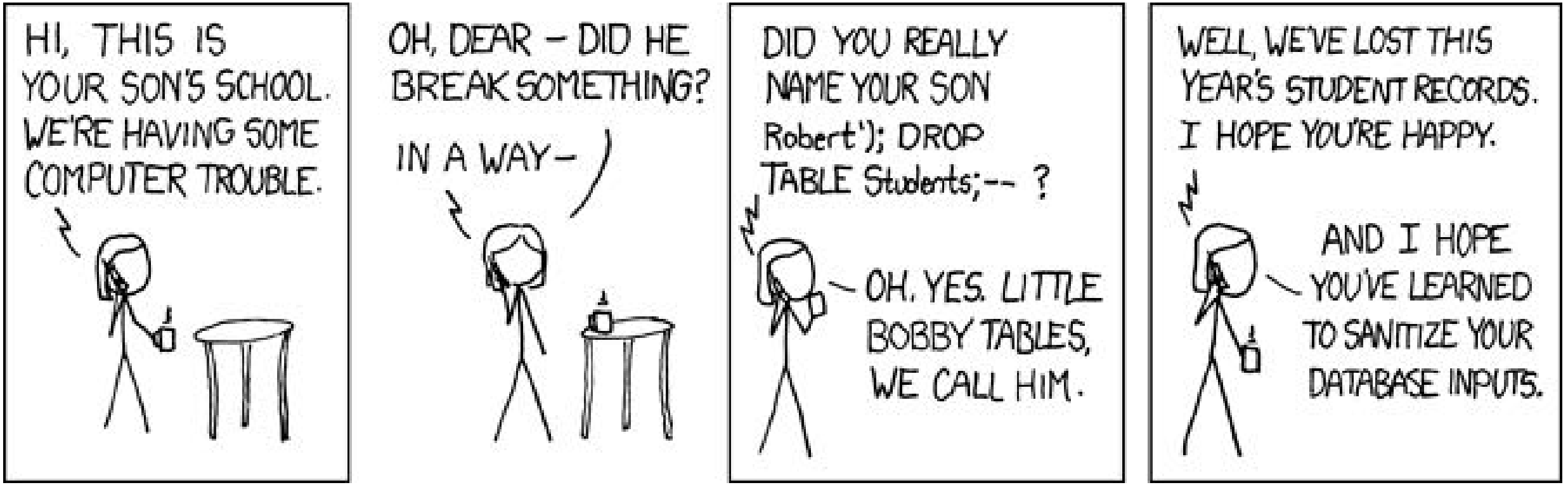
```
DECLARE @num VARCHAR(50)
SET @num = 'Gadgets'

SELECT  PName
FROM    Product
WHERE   category=@num
```



PName
Gizmo
Powergizmo

What happened here?



What does this SQL do:

```
Robert'); DROP
TABLE STUDENTS; --
```

Source: http://xkcd.com/327/, http://stackoverflow.com/questions/332365/xkcd-sql-injection-please-explain

It's called SQL injection: Version 1

Let's say the name was used in a variable, `$Name`. You then run this query:

```
INSERT INTO Students VALUES ( '$Name' )
```

What you get is:

```
INSERT INTO Students VALUES ( 'Robert' );  DROP TABLE STUDENTS; --')
```

The `--` only comments the remainder of the line.

Source: http://xkcd.com/327/, http://stackoverflow.com/questions/332365/xkcd-sql-injection-please-explain

It's called SQL injection: Version 2

It drops the students table.

The original query in the school's program probably looks something like

```
var query = "SELECT * FROM Students WHERE (Name = '" + tbName.Text + "')";
```

This is the naive way to add user text to a query, and is *very bad*. So bad, one might even say *evil*. Since the student's name is `Robert'); DROP TABLE STUDENTS; --` the resulting query (after concatenation) is

```
SELECT * FROM Students WHERE (Name = 'Robert'); DROP TABLE Students; --')
```

which, in plain English, roughly translates to the two queries:

> Get everything from the Students table where the student's name is Robert.

and

> Delete the Students table and ignore everything else I say from this point on ') and any other query-breaking junk.

Source: http://xkcd.com/327/, http://stackoverflow.com/questions/332365/xkcd-sql-injection-please-explain

SQL injection

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

```
' or '1'='1
' or '1'='1' -- '
' or '1'='1' ({ '
' or '1'='1' /* '
```

```
SELECT * FROM users WHERE name = '' OR '1'='1';
SELECT * FROM users WHERE name = '' OR '1'='1' -- ';
```

Source: http://en.wikipedia.org/wiki/SQL_injection

TJX Credit Card Numbers Theft

- In 2006, a $17.5 billion Fortune 500 firm
- Unauthorized intrusion resulting in lost of credit/debit card information
  - From May 2006 to January 2007?
  - From July 2005 to December 2006?
- TJX's overall losses: $1.35 billion – $4.5 billion