SQL Joins

& Other Data Processing Tips

Anran Xing and Michelle Tin
Office of Institutional Research and Assessment
The University of Alabama

THE UNIVERSITY OF
ALABAMA®

FROM
SQL
Database
Inner
Table
Outer
Examples
Join
WHERE
Left
MySQL
Right
ON
SELECT

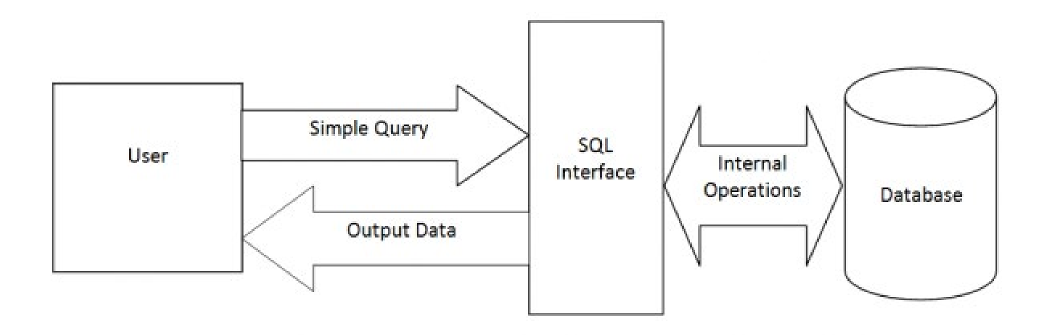
What is SQL?

"ess-q-ell" or "see-qwell"

SQL stands for **S**tructured **Q**uery **L**anguage

– SQL is the standard language for relational database management (American National Standards Institute, ANSI)

An interface which helps you communicate with your system's database using queries.

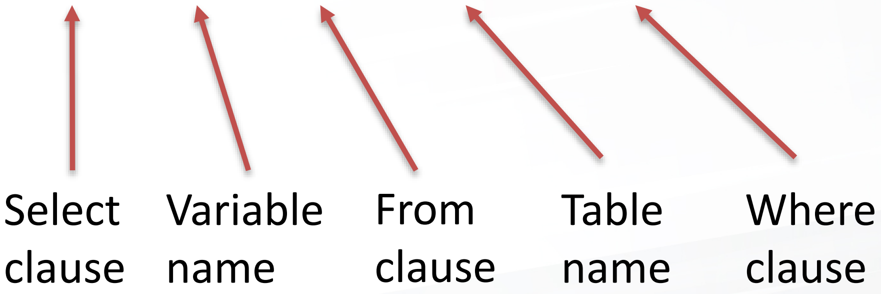


Basic SQL Queries

Students

Name
Mary
Jack
Ben

Select name **from** students **where** name='Mary'



Select clause

Variable name

From clause

Table name

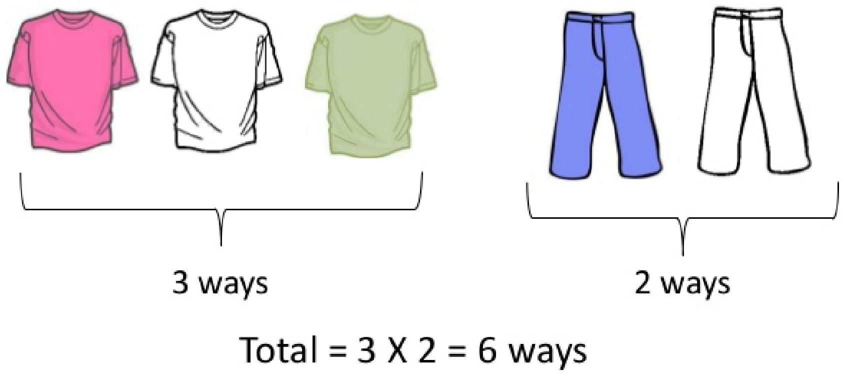
Where clause

SQL
Queries
Sub Queries
JOIN clause
meme-arsenal.ru

AN SQL QUERY GOES INTO A BAR, WALKS UP TO TWO TABLES AND ASKS...
'CAN I JOIN YOU?'

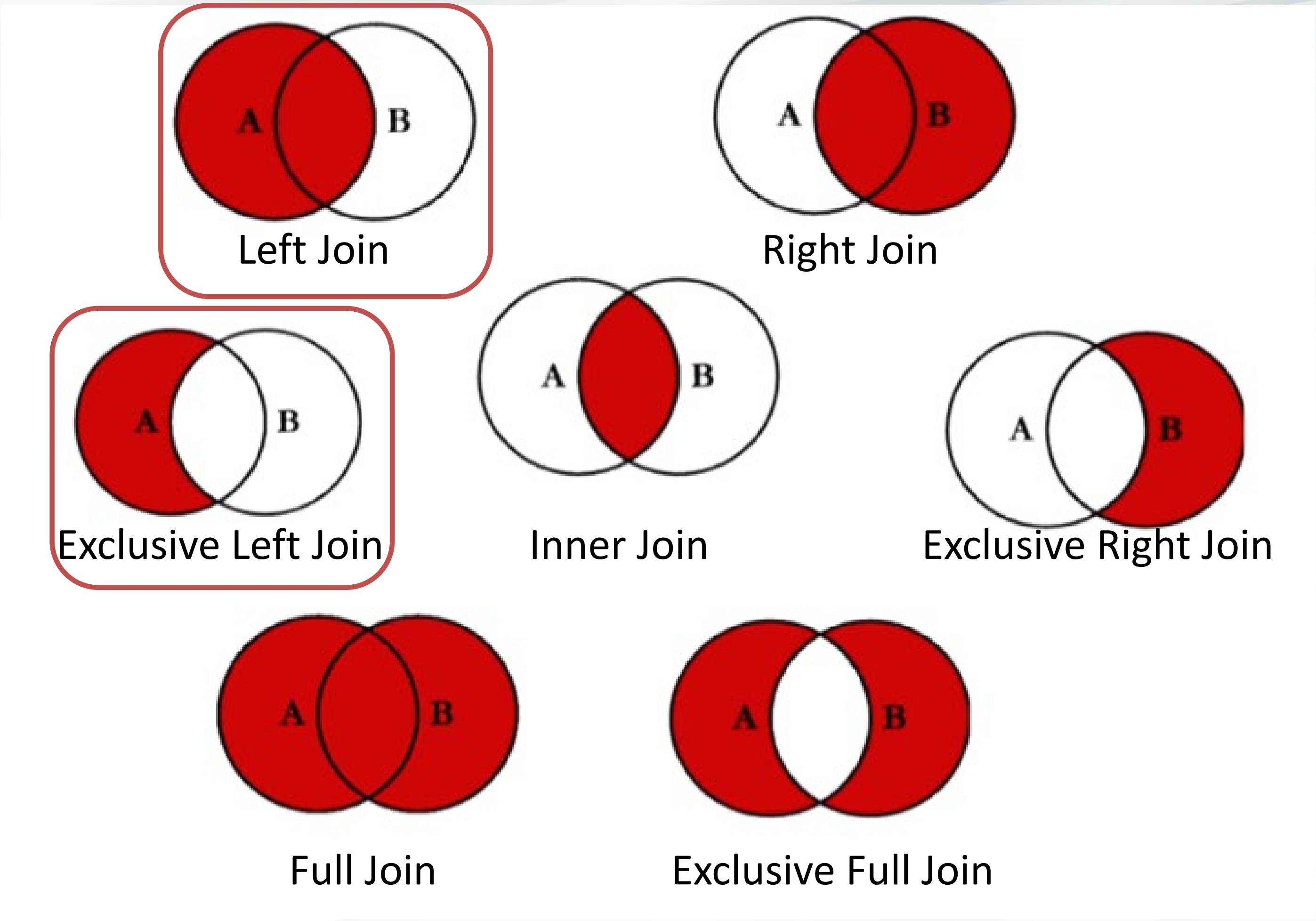
What are SQL Joins?

- SQL join clause - a join operation in relational algebra
  - Cartesian cross-products, combinations and permutations



- A join combines columns from one or more tables by using variables common to each

THE UNIVERSITY OF ALABAMA®



Left Join

Right Join

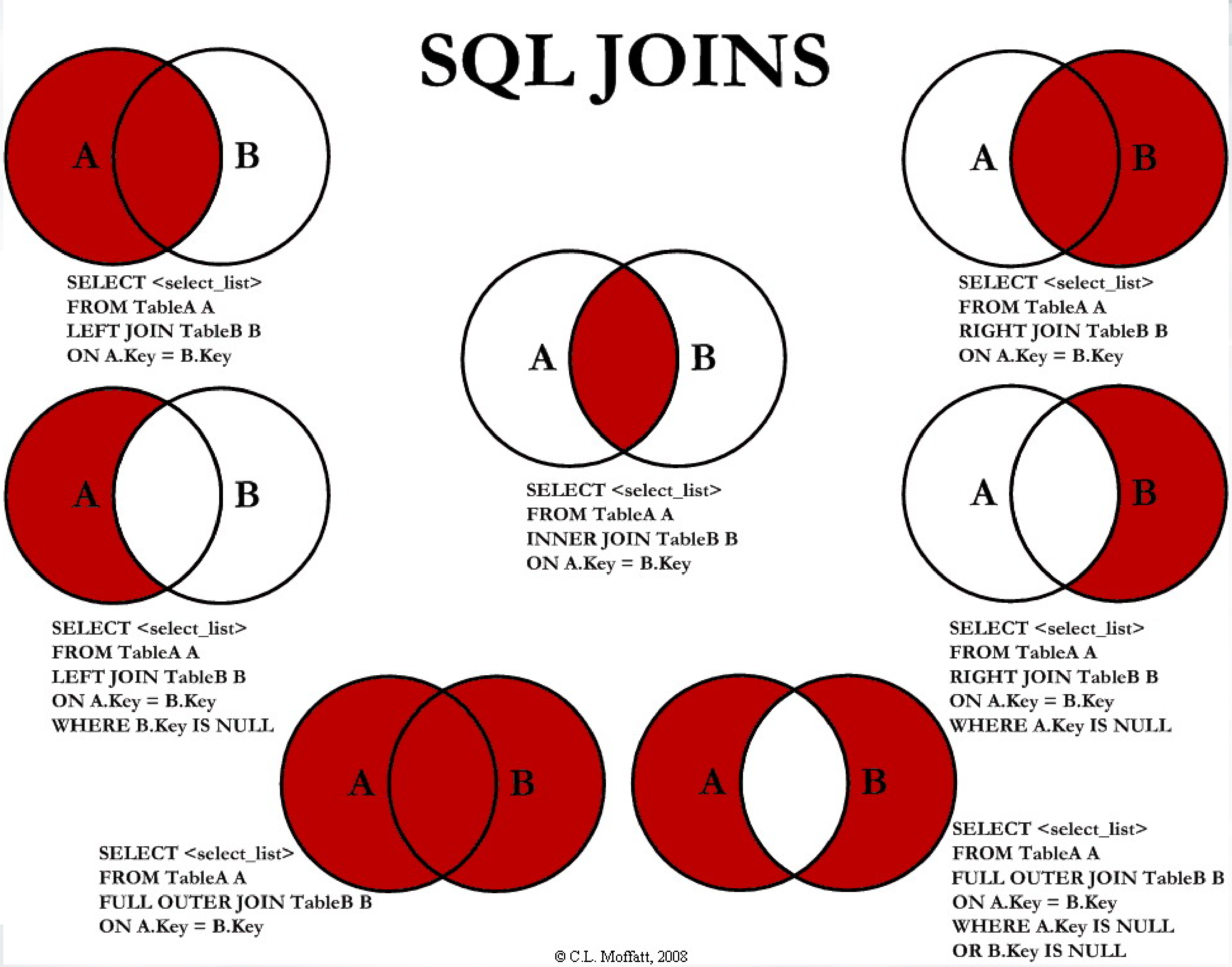
Exclusive Left Join

Inner Join

Exclusive Right Join

Full Join

Exclusive Full Join



SQL JOINS

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

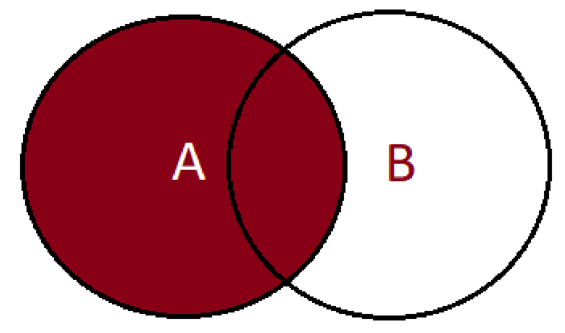
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

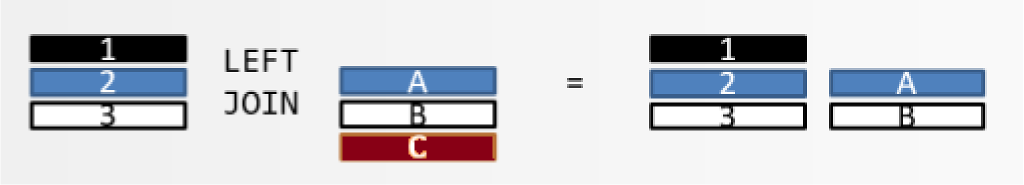
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Left (Outer) Join Definition

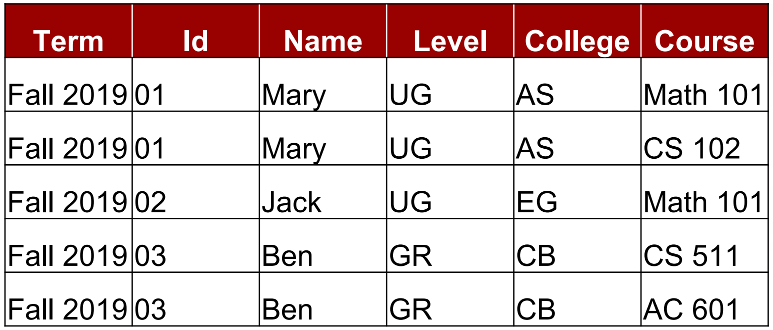


The SQL Left Join returns all the rows or records present in the left table and matching rows from the right table or NULL in case of no matching value.



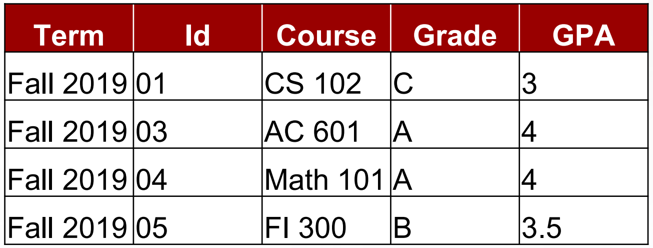
Sample Dataset

Table: Courses



Term	Id	Name	Level	College	Course
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

Table: Grades



Term	Id	Course	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

THE UNIVERSITY OF ALABAMA®

Basic Left Join

Courses

Term	*Id*	Name	Level	College	Course
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

Grades

Term	*Id*	Course	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

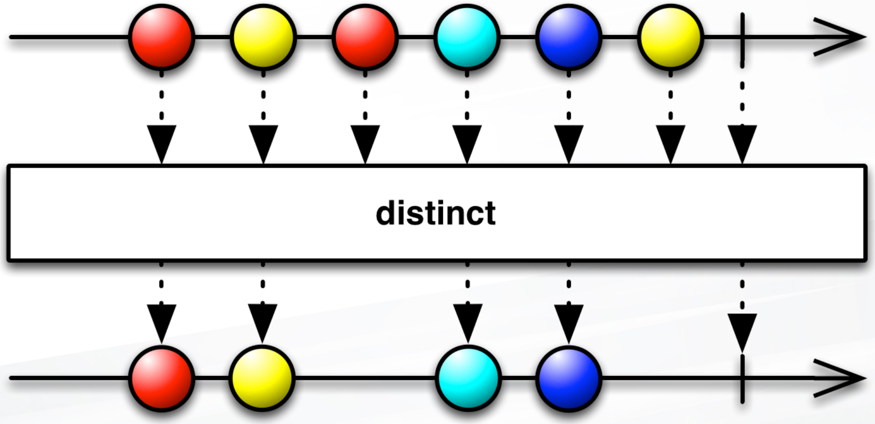
Id	Name	Grade
01	Mary	C
01	Mary	C
02	Jack	
03	Ben	A
03	Ben	A

```
select  Courses.id, Courses.name, Grades.grade
from Courses
     left join Grades on Courses.id = Grades.id
```

Id → key

Distinct

The SQL DISTINCT argument is used in conjunction with the SELECT statement to eliminate all the duplicate records and fetching only unique records.



THE UNIVERSITY OF ALABAMA®

Basic Left Join with Distinct

Courses

Term	*Id*	Name	Level	College	Course
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

Grades

Term	*Id*	Course	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

Id	Name	Grade
01	Mary	C
02	Jack	
03	Ben	A

```
select  DISTINCT  Courses.id, Courses.name, Grades.grade
from Courses
         left join Grades on Courses.id = Grades.id
```

Id → key

THE UNIVERSITY OF ALABAMA®

Basic Left Join with Multiple Keys

Courses

Term	*Id*	Name	Level	College	*Course*
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

Grades

Term	*Id*	*Course*	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

Id	Name	Course	Grade
01	Mary	CS 102	C
01	Mary	Math 101	
02	Jack	Math 101	
03	Ben	AC 601	A
03	Ben	CS 511	

```
select DISTINCT Courses.id, Courses.name, Courses.course, Grades.grade
from Courses
        left join Grades on Courses.id = Grades.id
                        and Courses.course = Grades.course
```

Id and course → key

SQL Aliases

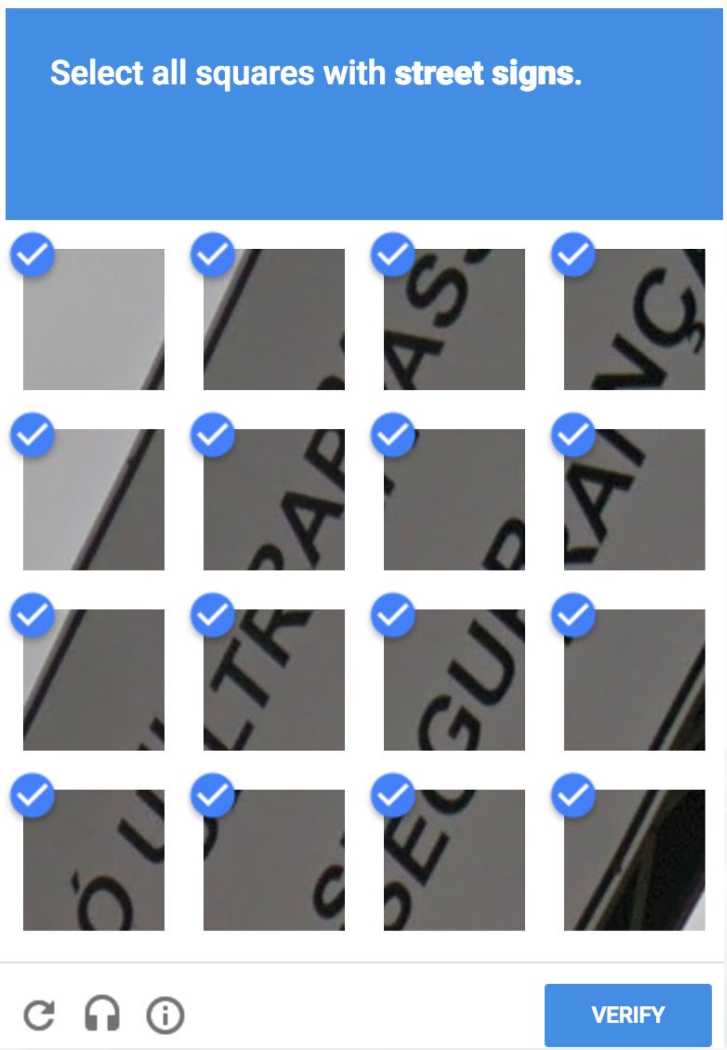
If your table names are too long…

Ex: *student_name_race_sex_state_by_term_level …*

- SQL aliases are used to give a **table**, or a **variable** in a table, a temporary name.
- Aliases are often used to rename variables or to make them more readable.
- An alias only exists for the duration of the query.

```
select distinct A.Term as semester
from courses as A
```

Select All Variables



Select all squares with **street signs.**

Left Join Example

Left Join Multiple Tables

Courses

Term	Id	Name	Level	College	Course
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

Grades

Term	Id	Course	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

Colleges

college_code	college_name
AS	Arts & Sciences
CB	Culverhouse College of Business
EG	Engineering

Levels

level_code	level_decode
UG	Undergraduate
GR	Graduate

Left Join Example

Left Join Multiple Tables

Term	Id	Name	Level	College	Course	St_id	Grade	school	level
Fall 2019	01	Mary	UG	AS	CS 102	01	C	Arts & Sciences	Undergraduate
Fall 2019	01	Mary	UG	AS	Math 101			Arts & Sciences	Undergraduate
Fall 2019	02	Jack	UG	EG	Math 101			Engineering	Undergraduate
Fall 2019	03	Ben	GR	CB	AC 601	03	A	Culverhouse College of Business	Graduate
Fall 2019	03	Ben	GR	CB	CS 511			Culverhouse College of Business	Graduate

```
select distinct A.*, B.id as St_id, B.grade, college_name as school, level_decode as level
from courses A
        left join grades as B on A.id=B.id and A.course=B.course
        left join colleges on college=college_code
        left join levels on level=level_code
where term=‘Fall 2019’
```

THE UNIVERSITY OF
ALABAMA®

Left Join Examples

Apply conditions with ON vs WHERE statements

- In SQL, the 'WHERE' and 'ON' clauses are both conditional statements
  - the ‘WHERE’ clause is used in select statements for specifying the conditions
  - the 'ON’ clause is used in joins
    - it verifies or checks if the records are matched in the target and source tables before the tables are joined

Left Join Examples

Filter in the ON clause

Courses

Term	*Id*	Name	Level	College	*Course*
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

Grades

Term	*Id*	*Course*	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

Name	Course	Grade
Ben	AC 601	A
Ben	CS 511	
Jack	Math 101	
Mary	CS 102	
Mary	Math 101	

```
select distinct A.name, A.course, B.grade
from courses A
left join grades B on  A.id=B.id
                       and A.course=B.course
                       and grade='A'
```

Left Join Examples

Filter in the WHERE clause

Courses

Term	*Id*	Name	Level	College	*Course*
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	01	Mary	UG	AS	CS 102
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511
Fall 2019	03	Ben	GR	CB	AC 601

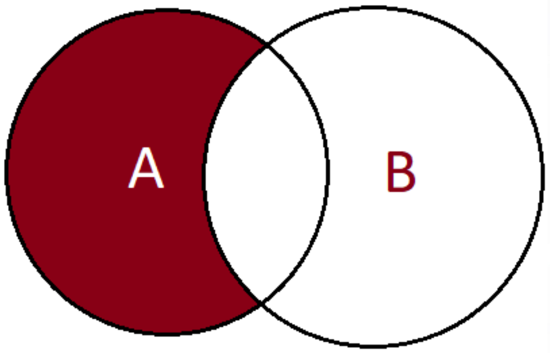
Grades

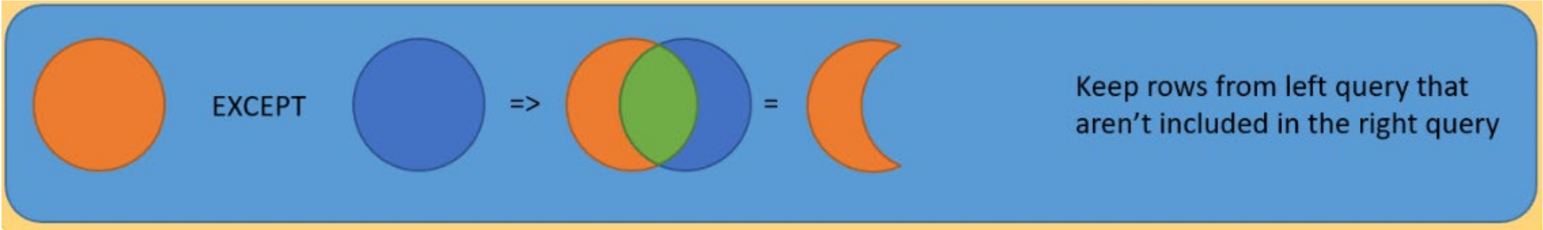
Term	*Id*	*Course*	Grade	GPA
Fall 2019	01	CS 102	C	3
Fall 2019	03	AC 601	A	4
Fall 2019	04	Math 101	A	4
Fall 2019	05	FI 300	B	3.5

Name	Course	Grade
Ben	AC 601	A
Ben	CS 511	
Jack	Math 101	
Mary	CS 102	C
Mary	Math 101	

```
select distinct A.name, A.course, B.grade
from courses A
left join grades B on  A.id=B.id
                        and A.course=B.course
where grade='A'
```

Exclusive Left Join/Except





Keep rows from left query that aren't included in the right query

Exclusive Left Join/Except Example

Left exclusive join using where statement

```
select distinct A.*
from courses A
        left join grades B on A.id=B.id
                                    and A.course=B.course
where B.id is null
```

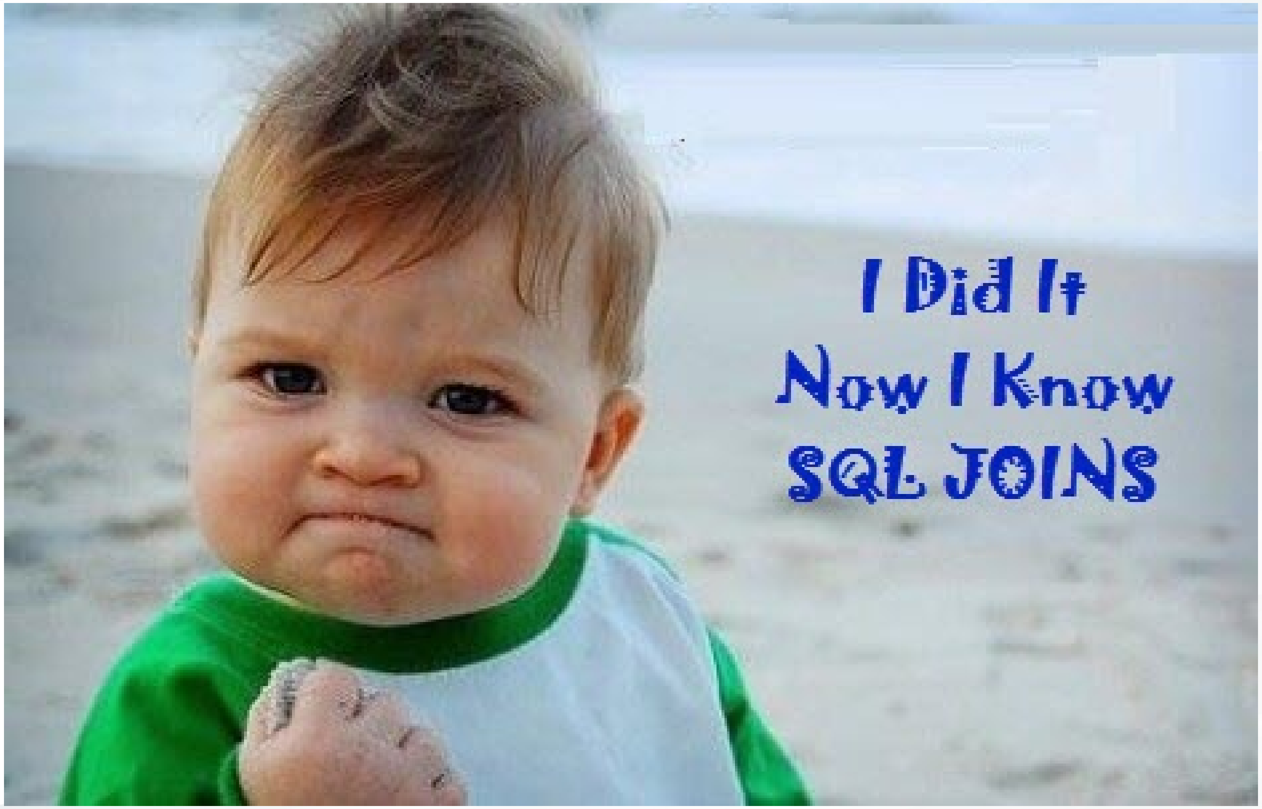
Term	Id	Name	Level	College	Course
Fall 2019	01	Mary	UG	AS	Math 101
Fall 2019	02	Jack	UG	EG	Math 101
Fall 2019	03	Ben	GR	CB	CS 511

Except function

```
select distinct term, id, course
from courses
except
        select distinct term, id, course
        from grades
```

Term	Id	Course
Fall 2019	01	Math 101
Fall 2019	02	Math 101
Fall 2019	03	CS 511

Questions?



- Anran Xing, Institutional Research Analyst
  - axing@ua.edu
- Michelle Tin, Institutional Research Analyst
  - mttin@ua.edu