Keys, Join, and Union

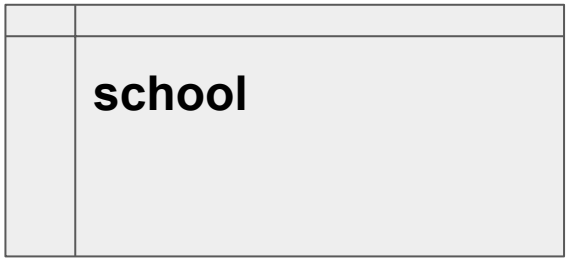
Module 2: 03

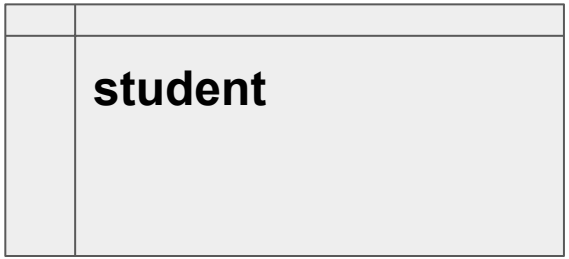
Today’s Objectives

1. Keys
2. Cardinality
3. Joins
4. Unions

The problems…

How to we tell which students have which classes?
How can we tell what lockers are assigned to students taking Algebra?





Primary Keys

Primary Keys are used to uniquely identify a row on a table.

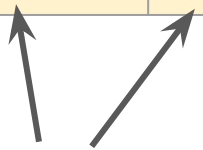
- **Natural Key** - a unique value *in the data* that can be used to identify a specific row on the table. (example a student id)

- **Surrogate Key** - a *generated* unique value that is used when no natural key is available (example a sequentially generated number)

- **Composite Key** - when a *2 or more columns are used as the key* to identify a unique row on a table. (example a card suit and rank)

Natural Composite Primary Key

suit	value	times_played
Hearts	Ace	5
Diamonds	Three	2
Hearts	Jack	4
Spades	Ace	1



primary key - composite natural key. One column is not enough to identify a unique value, but together they form a unique key

Surrogate Primary Key

id	first_name	last_name
1	Jack	Burton
2	Gracie	Law
3	Eddie	Lee

primary key - surrogate key. There is no value in the data that identifies a unique value, so a unique value is generated for each row.

Foreign Key

A **foreign key** Exist in other tables to reference a unique related row in the source table. Used to create relationships between tables.

- Usually References a **primary key**, but can reference any column that contains a unique value that can be used to identify a specific row.
- Can reference a composite key, but all columns that make up the primary key on the source table must be referenced on the table.

Country

code	name
USA	United States
GBR	Great Britain
CAN	Canada

City

id	countrycode	name
1	**USA**	United States
2	GBR	Great Britain
3	CAN	Canada

The country table's **primary key**, code, is a **foreign key** on the City table to reference what country a city is in.

Cardinality

Cardinality is a way of defining the relationship of data between tables.

Degrees of Cardinality

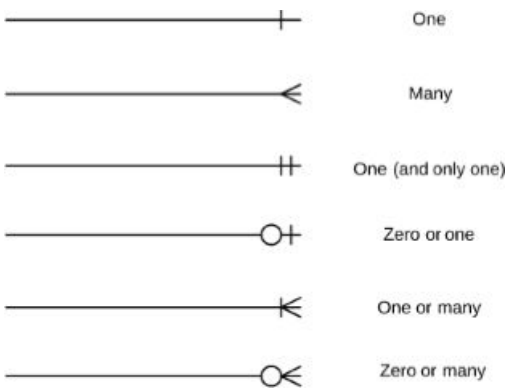
One-to-One (1:1) - One entity of data on a table relates to a single entity of data on another table.

One-to-Many (1:N) - One entity of data on a table relates to a multiple entities of data on another table.

Many-to-Many (M:N) - Multiple entities of data on a table relates to a multiple entities of data on another table.

Cardinality

- Describes relationship between two tables
- Relationship between a row in one table and a row of another table.
- Options are one or many
- 1 to 1, 1 to M, M to M



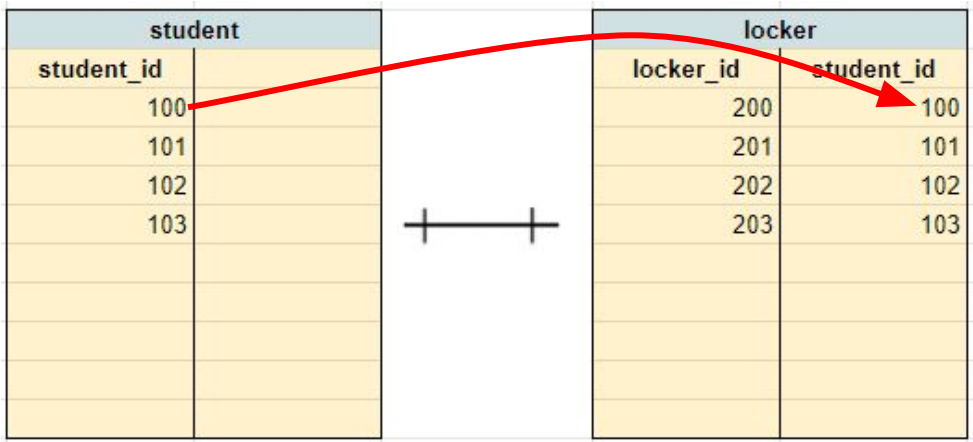
One-to-One (1:1) Cardinality

An entity on one table relates to a single entity on a second table.

A student can have one locker and each locker can only have one student assigned to it

Requires no extra tables, a foreign key can be used either way without duplicating data.

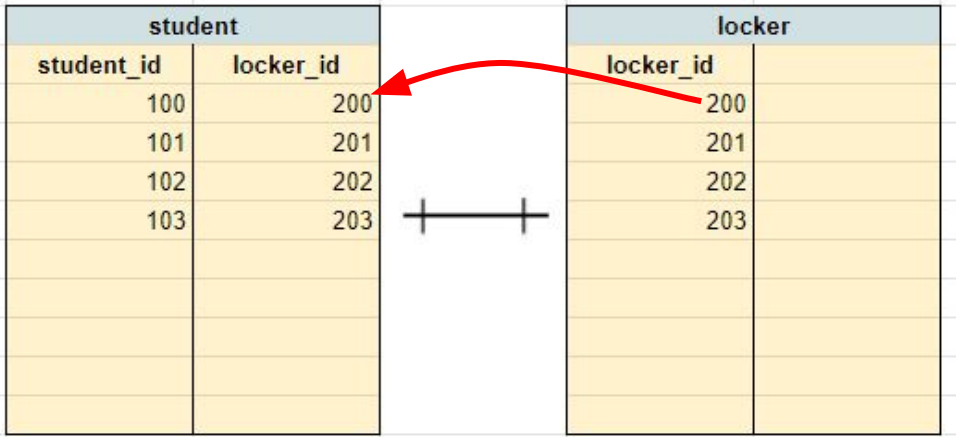
One-to-One (1:1) Cardinality



The student_id is added as a foreign key on the locker table

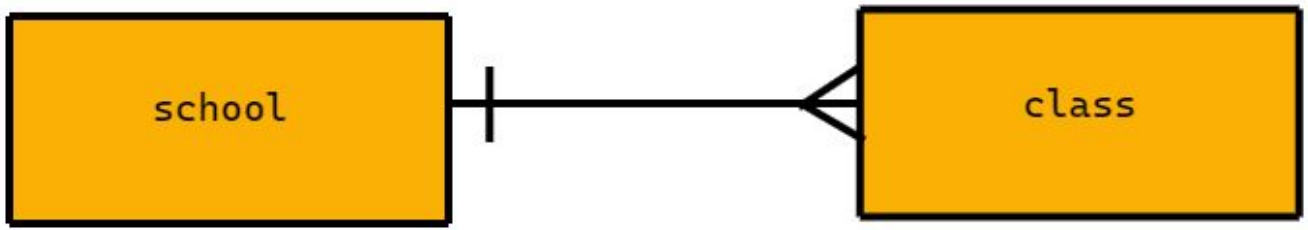
OR

The locker_id could be added as a foreign key on the student table



One-to-Many (1:N) Cardinality

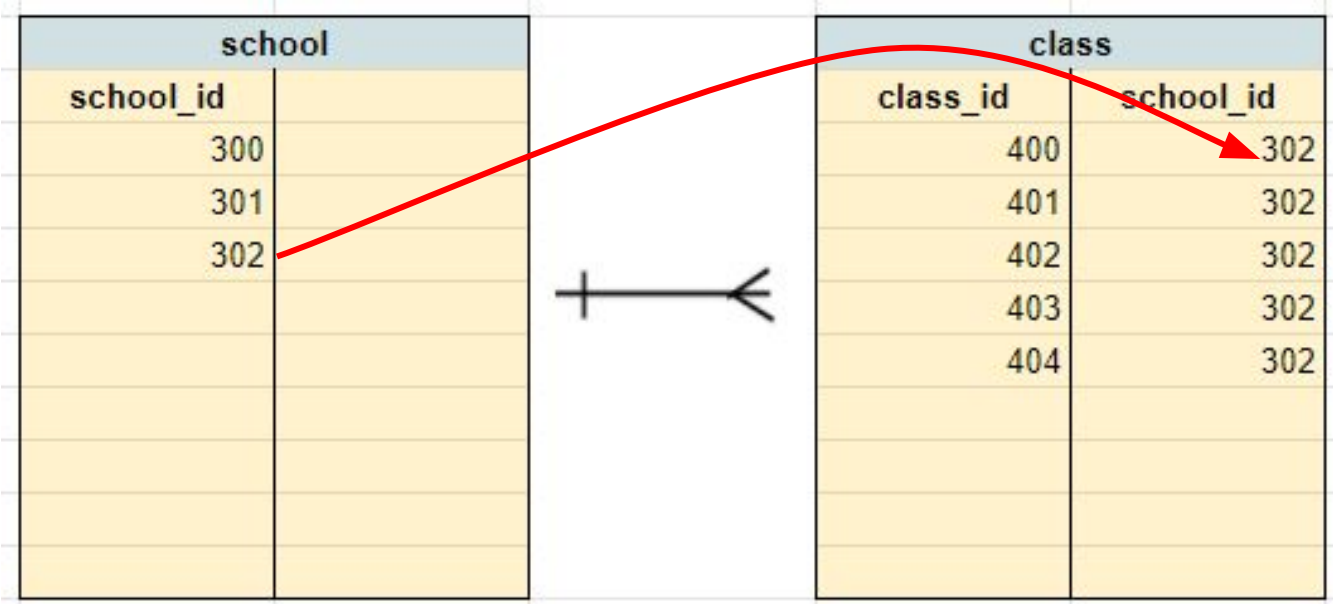
An entity on one table can relate to multiple entities on a second table.



A School can have multiple classes, but a class can only be related to one school. .

Requires no extra table, the table on the 1 side’s primary key can be added to the table with the M relationships without duplication of data

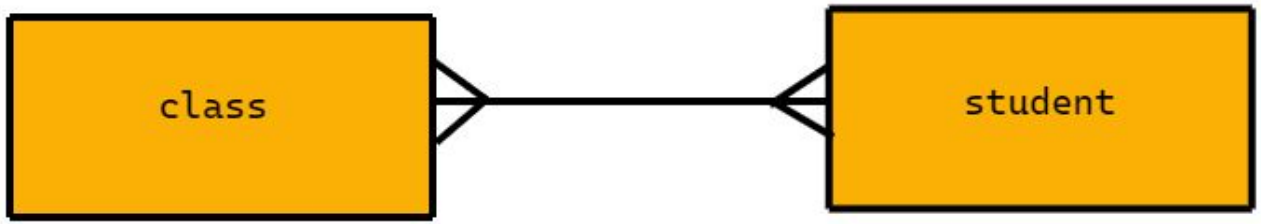
One-to-Many (1:N) Cardinality



Since one school may have many classes and each class may only have one school, the school id is added the class table.

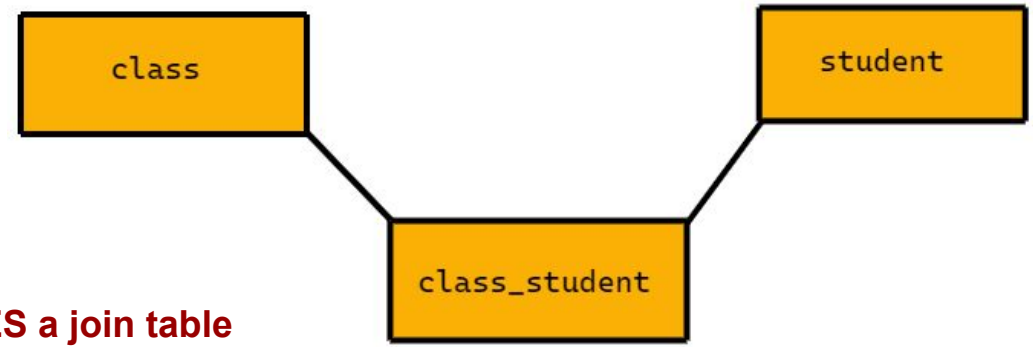
Many-to-Many (M:N) Cardinality

Each entity on a table can relate to multiple entities on a second table.



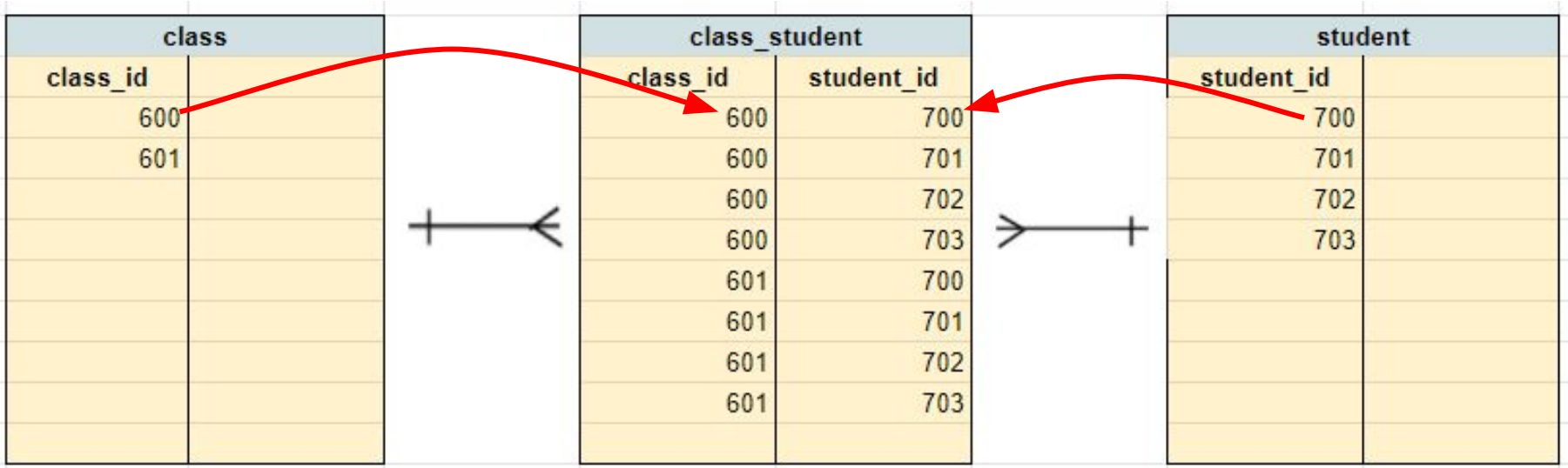
A class can have multiple students, and a student may have multiple classes.

Requires a join table that contains just the primary keys of both tables to avoid duplicating data



A Many-to-Many Relationship REQUIRES a join table

Many-to-Many (M:N) Cardinality

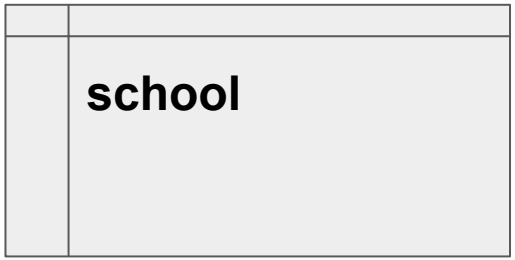


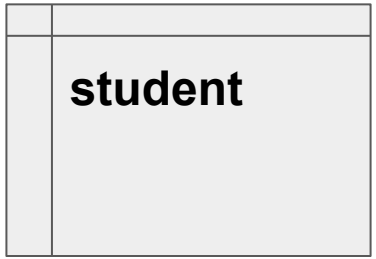
It is not possible to model a Many-to-Many relationship with only 2 tables since adding the id of the other onto either table would cause duplicate data.

To model a Many-to-Many relationship a third Join Table must be used to create the relationship between the ids without duplicating either the class or the student data.

The Solution

How to we tell which students have which classes?
How can we tell what lockers are assigned to students taking Algebra?

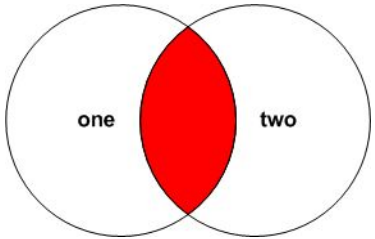




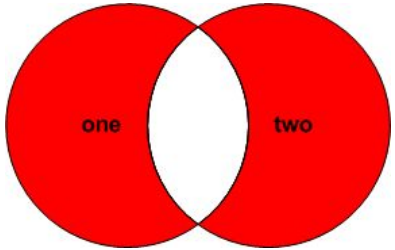
Joins

- SQL JOINs allow us to create queries that produce data from one or more tables.
- Related records are "joined" into a single result.
- *Joins are referred to as **INNER** and **OUTER**.*

INNER

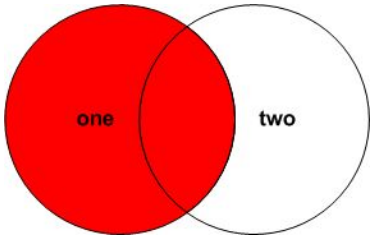


OUTER

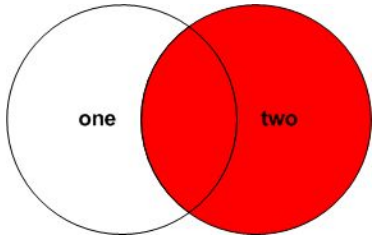


- *The tables involved in a JOIN are referred to as **LEFT** and **RIGHT**.*

LEFT



RIGHT



Tables

Table one	
number	**description**
100	ONE - 100
101	ONE - 101
102	ONE - 102
103	ONE - 103
104	ONE - 104
105	ONE - 105
990	ONE-BOTH - 990
991	ONE-BOTH - 991
992	ONE-BOTH - 992
993	ONE-BOTH - 993
994	ONE-BOTH - 994
995	ONE-BOTH - 995

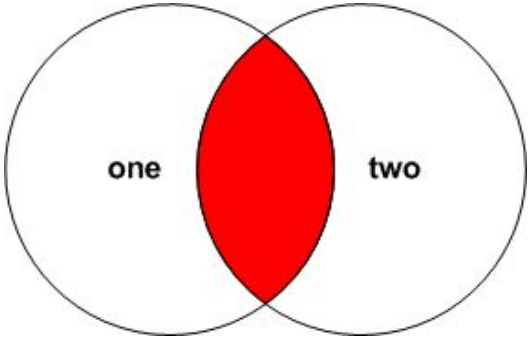
Table two	
number	**description**
200	TWO - 200
201	TWO - 201
202	TWO - 202
203	TWO - 203
204	TWO - 204
205	TWO - 205
990	TWO-BOTH - 990
991	TWO-BOTH - 991
992	TWO-BOTH - 992
993	TWO-BOTH - 993
994	TWO-BOTH - 994
995	TWO-BOTH - 995

Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

Inner Join

(Default)

```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
JOIN two ON one.number = two.number
```

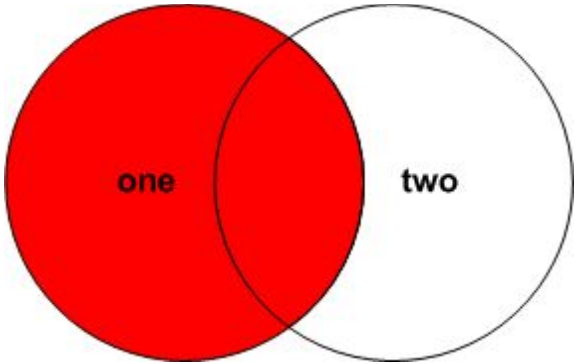


Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	**ONE-BOTH - 990**	**990**	**TWO-BOTH - 990**
991	**ONE-BOTH - 991**	**991**	**TWO-BOTH - 991**
992	**ONE-BOTH - 992**	**992**	**TWO-BOTH - 992**
993	**ONE-BOTH - 993**	**993**	**TWO-BOTH - 993**
994	**ONE-BOTH - 994**	**994**	**TWO-BOTH - 994**
995	**ONE-BOTH - 995**	**995**	**TWO-BOTH - 995**
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

Left Join

(Left Outer Join)

```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
LEFT JOIN two ON one.number = two.number
```

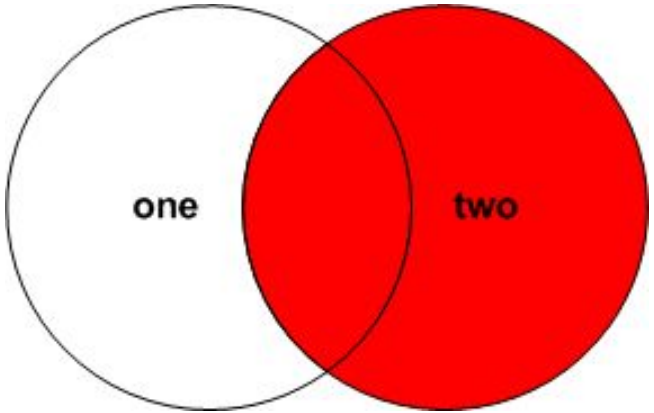


Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	**ONE - 100**	**null**	**null**
101	**ONE - 101**	**null**	**null**
102	**ONE - 102**	**null**	**null**
103	**ONE - 103**	**null**	**null**
104	**ONE - 104**	**null**	**null**
105	**ONE - 105**	**null**	**null**
990	**ONE-BOTH - 990**	**990**	**TWO-BOTH - 990**
991	**ONE-BOTH - 991**	**991**	**TWO-BOTH - 991**
992	**ONE-BOTH - 992**	**992**	**TWO-BOTH - 992**
993	**ONE-BOTH - 993**	**993**	**TWO-BOTH - 993**
994	**ONE-BOTH - 994**	**994**	**TWO-BOTH - 994**
995	**ONE-BOTH - 995**	**995**	**TWO-BOTH - 995**
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

Right Join

(Right Outer Join)

```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
RIGHT JOIN two ON one.number = two.number
```



Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	**ONE-BOTH - 990**	**990**	**TWO-BOTH - 990**
991	**ONE-BOTH - 991**	**991**	**TWO-BOTH - 991**
992	**ONE-BOTH - 992**	**992**	**TWO-BOTH - 992**
993	**ONE-BOTH - 993**	**993**	**TWO-BOTH - 993**
994	**ONE-BOTH - 994**	**994**	**TWO-BOTH - 994**
995	**ONE-BOTH - 995**	**995**	**TWO-BOTH - 995**
null	**null**	**200**	**TWO - 200**
null	**null**	**201**	**TWO - 201**
null	**null**	**202**	**TWO - 202**
null	**null**	**203**	**TWO - 203**
null	**null**	**204**	**TWO - 204**
null	**null**	**205**	**TWO - 205**

Popular Interview Question...

Question: What is the difference between a **LEFT JOIN** and a **LEFT OUTER JOIN**?

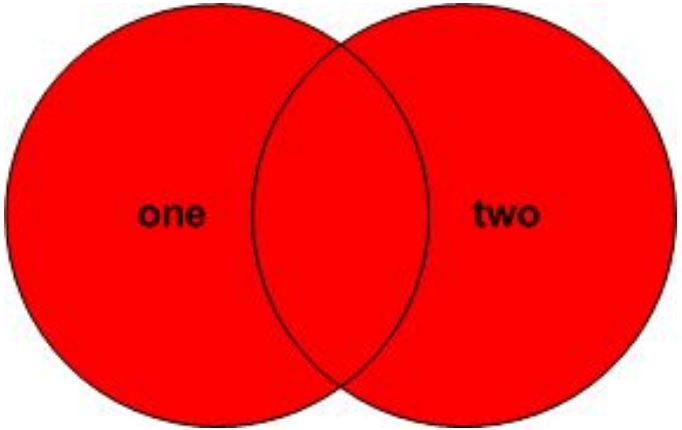
Answer: *Nothing, they are the same!*

Question: What is the difference between a **RIGHT JOIN** and **RIGHT OUTER JOIN**?

Answer: *Nothing, they are the same!*

Full Outer Join

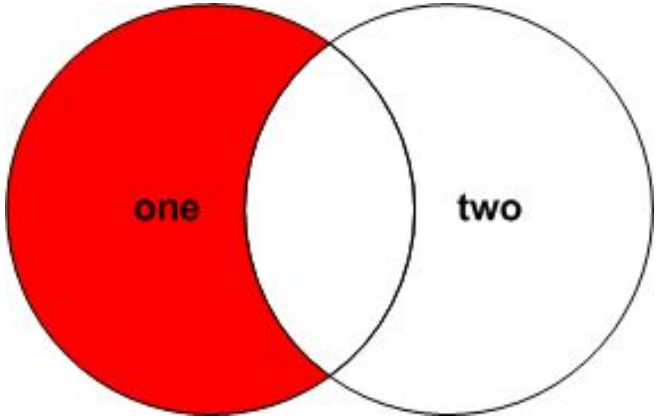
```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
FULL OUTER JOIN two ON one.number = two.number
```



Tables joined on number			
one.number	one.description	two.number	two.description
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

Only the Left Table Values
(Unnamed)

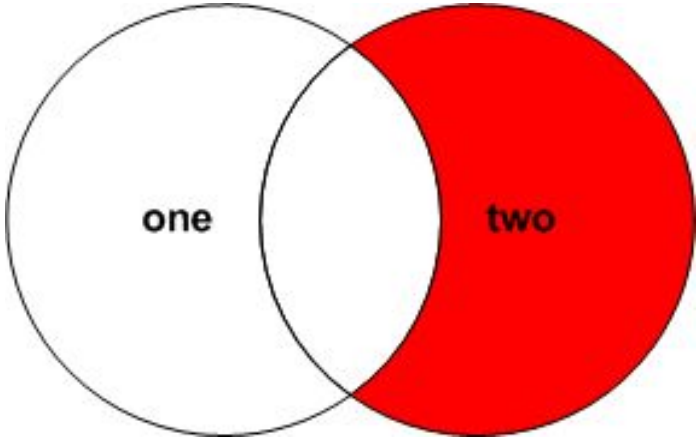
```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
LEFT JOIN two ON one.number = two.number
WHERE two.number IS NULL
```



Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	**ONE - 100**	**null**	**null**
101	**ONE - 101**	**null**	**null**
102	**ONE - 102**	**null**	**null**
103	**ONE - 103**	**null**	**null**
104	**ONE - 104**	**null**	**null**
105	**ONE - 105**	**null**	**null**
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

Only the Right Table Values
(Unnamed)

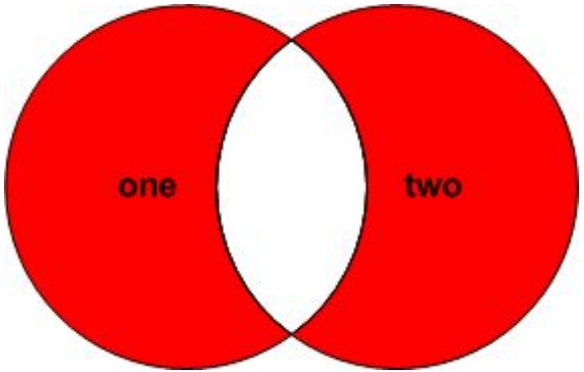
```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
RIGHT JOIN two ON one.number = two.number
WHERE one.number IS NULL
```



Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	**null**	**200**	**TWO - 200**
null	**null**	**201**	**TWO - 201**
null	**null**	**202**	**TWO - 202**
null	**null**	**203**	**TWO - 203**
null	**null**	**204**	**TWO - 204**
null	**null**	**205**	**TWO - 205**

In the LEFT or RIGHT, but not both (Unnamed)

```
SELECT one.number AS one_number,
one.description as one_description, two.number
as two_number, two.description as
two_description
FROM one
FULL OUTER JOIN two ON one.number = two.number
WHERE one.number IS NULL OR two.number IS NULL
```



Tables joined on number			
one.number	**one.description**	**two.number**	**two.description**
100	**ONE - 100**	**null**	**null**
101	**ONE - 101**	**null**	**null**
102	**ONE - 102**	**null**	**null**
103	**ONE - 103**	**null**	**null**
104	**ONE - 104**	**null**	**null**
105	**ONE - 105**	**null**	**null**
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	**null**	**200**	**TWO - 200**
null	**null**	**201**	**TWO - 201**
null	**null**	**202**	**TWO - 202**
null	**null**	**203**	**TWO - 203**
null	**null**	**204**	**TWO - 204**
null	**null**	**205**	**TWO - 205**

Union

- A SQL UNION combines the results of two or more queries into a single result set.
- **The number of columns involved must match exactly and data types must be identical.**
- **Duplicate rows are removed.**

A good example for this is a database that might have faculty and students separated into different tables but we want to return all people who attend or work at a school.

```
SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions]
UNION
SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions]
```