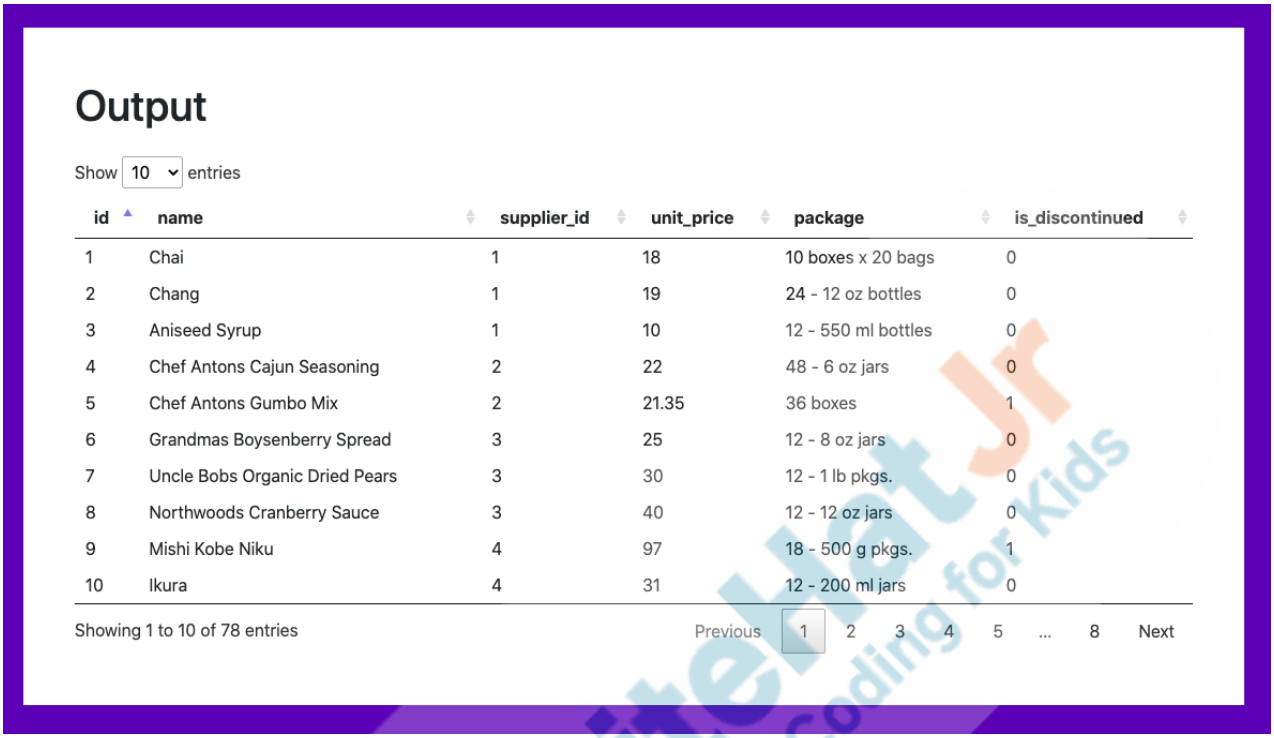
Topic	JOIN Statements	
Class Description	**Students will learn about JOIN statements in SQL**	
Class	**C-232**	
Class time	**45 mins**	
Goal	• Understand about Join SQL Statements • Differences b/w different join statements	
Resources Required	• Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Visual Studio Code • Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Visual Studio Code	
Class structure	**Warm-Up** **Teacher-led Activity 1** **Student-led Activity 1** **Wrap-Up**	**10 mins** **15 mins** **15 mins** **5 mins**

WARM-UP SESSION - 10mins	
Teacher Action	**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?	**ESR**: Hi, thanks, yes, I am excited about it!

In the last session, we learned about DBMS, and we worked on SQL statements, select, where and insert and many more Any doubts from the last session? *The teacher clarifies doubts (if any)* *So what do you think now we all know about SQL statements or still we need to learn more about this!* *Yes,* *Still, many things need to learn so today, we are using more sql statements* *Let's do that!*	**ESR**: Varied!
Q&A Session	
Question	**Answer**
What would be the statement, if you wanted to select all the emails from a table called users? A. select email from users; B. select * from users; C. select users from emails; D. select * from emails;	**A**
What would be the statement, if you wanted to select all the users from Australia or New Zealand? A. select * from users where country=Australia or country=New Zealand; B. select * from users where country='Australia' and	**C**

country='New Zealand'; C. select * from users where country='Australia' or country='New Zealand'; D. select * from users where country=Australia and country=New Zealand;	
TEACHER-LED ACTIVITY - 15mins	
Teacher Initiates Screen Share	
ACTIVITY • **JOIN statements** • **Differences b/w inner join, outer join, left join and right join**	
Teacher Action	**Student Action**
In the last class, we learnt about select statements in SQL. Today, we are going to learn about the Join statements. What do you understand by the word "join", and how do you think it would be used in SQL? Join keyword is usually used to join things. What could be the things in SQL? That's right! It is used to join 2 or more tables together! Actually, it is used to join the rows of 2 or more tables based on the common fields it has. Let's understand them in deep but first, there are 4 types of join statements - 1. **Inner Join -** The inner join keyword selects records that have matching values in both tables 2. **Full Join -** This keyword returns all records that match the records in the left (table1) and right (table2) tables. Full Join and Full outer join both are same 3. **Left Join -** A Left Join returns all records from the	**ESR** Varied! **ESR:** Tables!

left table (table1) and the matching records from the right table (table2). If there is no match, the right side will show 0 records. 4. **Right Join -** This keyword returns all records from the right table (table2), and all matching records from the left table (table1). If there is no match, 0 records will appear from the left side. Now, with this knowledge, you might be wondering, what does this actually mean? Let's take a look at an example for all the join statements! For that, let's open our SQL Editor from Student Activity 1 *Teacher refers to Teacher Activity 1*	*Student refers to Student Activity 1*
Before we begin, let's take a look at all the tables and the data that it holds, so that we understand our data better! In the last class, we looked at **customers** and **suppliers**, but there are 3 other tables - 1. **company_products** 2. **company_orders** 3. **order_items** Let's query them one by one and see the columns that it has! *Teacher queries the database for **company_products*** **SELECT * from company_products;**	



Output

Show 10 entries

id	name	supplier_id	unit_price	package	is_discontinued
1	Chai	1	18	10 boxes x 20 bags	0
2	Chang	1	19	24 - 12 oz bottles	0
3	Aniseed Syrup	1	10	12 - 550 ml bottles	0
4	Chef Antons Cajun Seasoning	2	22	48 - 6 oz jars	0
5	Chef Antons Gumbo Mix	2	21.35	36 boxes	1
6	Grandmas Boysenberry Spread	3	25	12 - 8 oz jars	0
7	Uncle Bobs Organic Dried Pears	3	30	12 - 1 lb pkgs.	0
8	Northwoods Cranberry Sauce	3	40	12 - 12 oz jars	0
9	Mishi Kobe Niku	4	97	18 - 500 g pkgs.	1
10	Ikura	4	31	12 - 200 ml jars	0

Showing 1 to 10 of 78 entries Previous 1 2 3 4 5 ... 8 Next

Here, we can see the following columns -

1. id
2. name
3. supplier_id
4. unit_price
5. package
6. is_discontinued

Here, all the columns make sense, but do take a look at the supplied_id. Do you know what it means?

ESR:
It means that this table has a relation with the suppliers table!

That's right! Let's take a look at the ***company_orders*** table now!

Teacher queries the database for ***company_orders***

SELECT * from company_orders;

Output

Show 10 entries

id	date	customer_id	total_amount	order_number
1	Wed, 04 Jul 2012 00:00:00 GMT	85	440	542378
2	Thu, 05 Jul 2012 00:00:00 GMT	79	1863.4	542379
3	Sun, 08 Jul 2012 00:00:00 GMT	34	1813	542380
4	Sun, 08 Jul 2012 00:00:00 GMT	84	670.8	542381
5	Mon, 09 Jul 2012 00:00:00 GMT	76	3730	542382
6	Tue, 10 Jul 2012 00:00:00 GMT	34	1444.8	542383
7	Wed, 11 Jul 2012 00:00:00 GMT	14	625.2	542384
8	Thu, 12 Jul 2012 00:00:00 GMT	68	2490.5	542385
9	Sun, 15 Jul 2012 00:00:00 GMT	88	517.8	542386
10	Mon, 16 Jul 2012 00:00:00 GMT	35	1119.9	542387

Showing 1 to 10 of 830 entries Previous 1 2 3 4 5 ... 83 Next

Here too, we can see that there is a column known as ***customer_id***, which means that this table has a relation with the customers table!

Now, to our final table - ***order_items***

Teacher queries the database for ***order_items***

SELECT * from order_items;

Output

Show 10 entries

id	order_id	product_id	unit_price	quantity
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	2	2	2	2
5	2	2	2	2
6	3	3	3	3
7	3	3	3	3
8	3	3	3	3
9	4	4	4	4
10	4	4	4	4

Showing 1 to 10 of 206 entries Previous 1 2 3 4 5 ... 21 Next

Here, in this table, can you tell me with which tables does this have a relation?	**ESR:** *company_products and company_orders!*
That's right! Awesome! So, now we know that we have 5 tables - 1. Customers 2. Suppliers 3. Company Products with a relation with suppliers 4. Company Orders with a relation with customers 5. Order items with a relation to both company products and company orders!	
Sounds complex?	**ESR:** Yes!
Well, trust me! This is nothing. There are even more complex databases out there!	

Now, what if you want to get all the customer's name, along with all the amounts of orders they have placed with dates! How would you do that? Let's think about it for a minute. We need 3 things - 1. Customer's Name 2. Total Amount of order they placed 3. Date on which they place the order If we think about it, we can find the name of the customer in the **customers** table and the rest of the two things in the **company_orders** table. Now, looking at it, it seems understood that this situation here requires us to use a JOIN statement. Let's do that. I'm going to type a statement, and then let's go through it together! *Teacher executes the following query in the editor*	**ESR:** Varied *Student observes*

SELECT customers.first_name, customers.last_name, company_orders.date, company_orders.total_amount **FROM** customers **INNER JOIN** company_orders **ON** customers.id=company_orders.customer_id;

Executing the above query on the Editor gives the following output -

Output

Show 10 entries

first_name	last_name	date	total_amount
Alejandra	Camino	Tue, 14 Aug 2012 00:00:00 GMT	86.5
Alejandra	Camino	Wed, 15 Aug 2012 00:00:00 GMT	155.4
Alejandra	Camino	Sun, 16 Sep 2012 00:00:00 GMT	498.5
Alejandra	Camino	Sun, 02 Mar 2014 00:00:00 GMT	365.89
Alejandra	Camino	Wed, 09 Apr 2014 00:00:00 GMT	361
Alexander	Feuer	Thu, 09 Aug 2012 00:00:00 GMT	1200.8
Alexander	Feuer	Thu, 20 Jun 2013 00:00:00 GMT	2147.4
Alexander	Feuer	Wed, 09 Oct 2013 00:00:00 GMT	114
Alexander	Feuer	Mon, 16 Dec 2013 00:00:00 GMT	1335
Alexander	Feuer	Wed, 12 Mar 2014 00:00:00 GMT	245

Showing 1 to 10 of 830 entries Previous 1 2 3 4 5 ... 83 Next

Great! Now, let's study this statement once!

First, let's understand it's structure! It is structured in the following way -

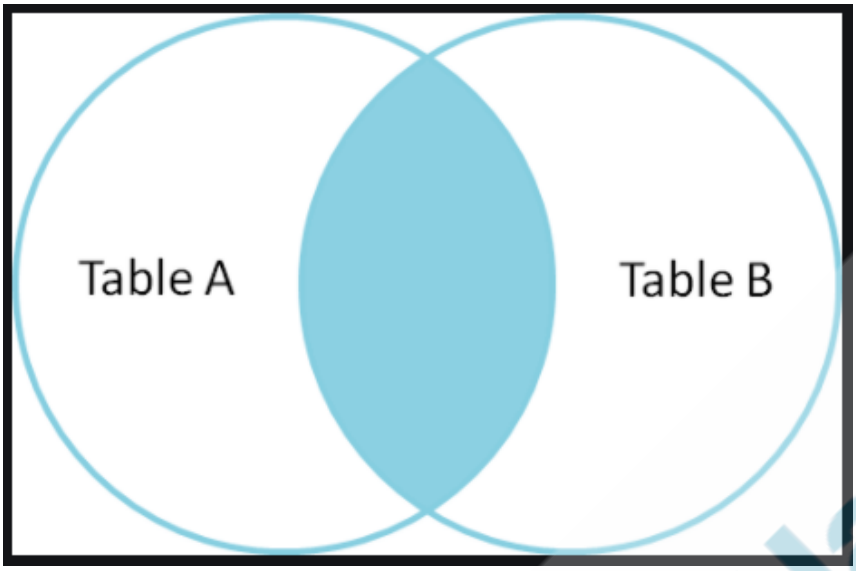
SELECT table1.column1, table1.column2, table2.column1, table2.column2… **FROM** table1 **INNER JOIN** table2 **ON** table1.matching_column=table2.matching_column;

Now in the statement above, we are again using the ***SELECT*** statement, but this time, in a different way. We are selecting columns from tables that we want to display with the help of ***table.column*** structure, where we are using the name of the tables and columns.

Do note that using an asterisk (*) here would get all the columns from all the tables joined together.

Next, again we are using the ***FROM*** keyword to select the

table that we want to query. Then, we are using ***INNER JOIN***, along with the name of the second table we want to join it with. Here, ***INNER JOIN*** could have also been ***LEFT JOIN, RIGHT JOIN*** and ***FULL JOIN*** too. Next, we are using a keyword called ***ON***, which means, "On what do you want to join the two tables?". We provide a condition of 2 columns, one from each table, that we expect to be equal to join the data of their rows together. It's not that difficult, right?	**ESR:** Yes!
Great! But what does INNER JOIN really mean? In context of the statement that we have written above, it means that we only want to JOIN those table rows which satisfy ***customers.id=company_orders.customer_id*** statement. The INNER JOIN is something like this -	

 Here, we can clearly observe that it will only take those rows and join them, that intersect, and that have something in common (customer ID in our case). No other data will be displayed! With this, can you guess what the other JOINS mean? What does FULL, LEFT and RIGHT JOIN mean?	**ESR:** Varied!
Well, let's consider some situations. In our statement, we are INNER Joining the **customers** table with the **company_orders** table. Here, the **customers** table is on the ***LEFT*** side of the ***JOIN*** keyword and ***company_orders*** is on the ***RIGHT*** side of the ***JOIN*** statement.	
Now let's say, there are some customers, who have not ordered anything yet, but we still want to fetch their data as well. Since no row for their order in the ***company_orders*** table would appear, their data would not be displayed in an ***INNER JOIN*** because it strictly requires the statement ***customers.id=company_orders.customer_id*** to be true.	

If we were to query in a way that it displays those customer's data as well, who didn't make any order, which join would we use?

ESR:
Left Join!

Why would we use a Left Join?

ESR:
Then it will display everything from the INNER JOIN, but would also display those customers' data who didn't order anything!

That's correct! Currently with Inner Join, we have about 830 rows!

Now, let's tweak the statement a little and use ***LEFT JOIN*** instead!

Teacher executes the following query in the editor

Student observes

SELECT customers.first_name, customers.last_name, company_orders.date, company_orders.total_amount **FROM** customers **LEFT JOIN** company_orders **ON** customers.id=company_orders.customer_id;

Executing the above query on the Editor gives the following output -

Output

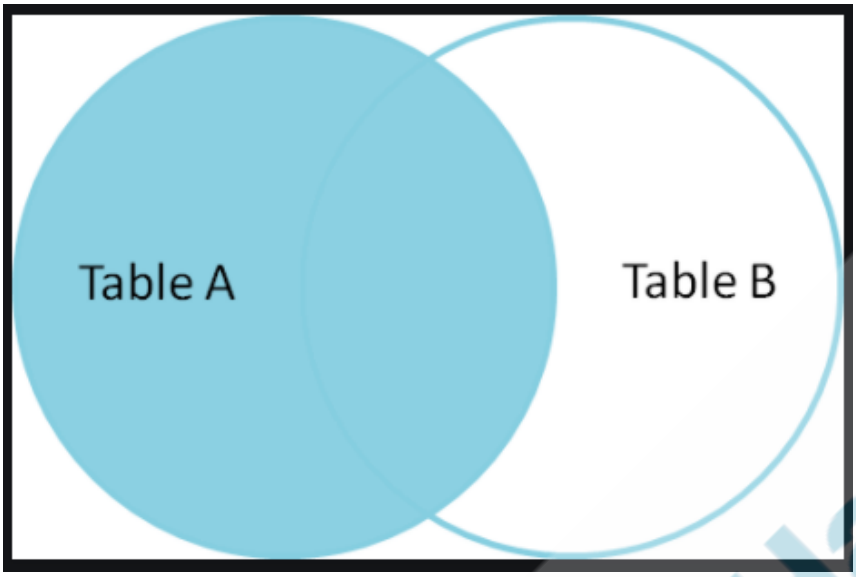
Show 10 entries

first_name	last_name	date	total_amount
Diego	Roel	null	null
Marie	Bertrand	null	null
Patricia	McKenna	Mon, 11 Nov 2013 00:00:00 GMT	997
Karl	Jablonski	Tue, 08 Oct 2013 00:00:00 GMT	996
Palle	Ibsen	Wed, 02 Apr 2014 00:00:00 GMT	990
Jose	Pavarotti	Tue, 11 Feb 2014 00:00:00 GMT	988.4
Martín	Sommer	Wed, 10 Oct 2012 00:00:00 GMT	982
Fran	Wilson	Mon, 03 Feb 2014 00:00:00 GMT	98.4
Paula	Wilson	Sun, 16 Jun 2013 00:00:00 GMT	977.5
Miguel	AngelPaolino	Mon, 20 Jan 2014 00:00:00 GMT	975

Showing 1 to 10 of 832 entries Previous 1 2 3 4 5 ... 84 Next

Awesome! Now, here we can observe that this time, we have about 832 rows, with 2 new customers that have not placed any order!

Left Join therefore looks something like this -



It's counterpart, however, which is RIGHT JOIN, is not applicable in our use case, since there cannot be any orders that do not belong to a customer!

Full Join is again, just all the data from both the tables, with as many rows that could be joined together and all the other data.

STUDENT-LED ACTIVITY - 15 mins

- **Ask the student to press the ESC key to come back to the panel.**
- **Guide the student to start Screen Share.**
- **The teacher gets into Full Screen.**

ACTIVITY

- **Practice Inner, Left, Right and Full Join statements!**

Teacher Action	Student Action
By now, I'm sure you have a very fair idea of what JOIN statements are, why they are used and what is the difference between inner join, full join, left join and right join.	

Now, it's time for you to practice! We have a table called ***suppliers***, which has a relation with ***company_products*** Can you find the name of the supplier company, their contact's name and phone number, along with the name of the product? Make sure that it is not a discontinued product! What do you think should be the first step?	**ESR:** To identify what data is available in which table
Great! Can you query the suppliers table? *Teacher guides the student to query all the suppliers*	**ESR:** Yes! *Student queries the suppliers*

```
1 select * from suppliers;
```

Gives the Output -

Output

Show 10 entries

id	company_name	contact_name	city	country	phone	fax
1	Exotic Liquids	Charlotte Cooper	London	UK	(171) 555-2222	null
2	New Orleans Cajun Delights	Shelley Burke	New Orleans	USA	(100) 555-4822	null
3	Grandma Kellys Homestead	Regina Murphy	Ann Arbor	USA	(313) 555-5735	null
4	Tokyo Traders	Yoshi Nagase	Tokyo	Japan	(03) 3555-5011	null
5	Cooperativa de Quesos Las Cabras	Antonio del Valle Saavedra	Oviedo	Spain	(98) 598 76 54	null
6	Mayumis	Mayumi Ohno	Osaka	Japan	(06) 431-7877	null
7	Pavlova	Ltd.	Ian Devling	Melbourne	Australia	null
8	Specialty Biscuits	Ltd.	Peter Wilson	Manchester	UK	null
9	PB Knäckebröd AB	Lars Peterson	Göteborg	Sweden	031-987 65 43	null
10	Refrescos Americanas LTDA	Carlos Diaz	Sao Paulo	Brazil	(11) 555 4640	null

Showing 1 to 10 of 29 entries Previous 1 2 3 Next

Here, we can see that we have the following fields in this table only!

1. Company Name
2. Contact Name
3. Phone Number

Great! Next, let's checkout the **company_products** table -

Teacher guides the student to query all the suppliers

Student queries the company_products

Output

Show 10 entries

id	name	supplier_id	unit_price	package	is_discontinued
1	Chai	1	18	10 boxes x 20 bags	0
2	Chang	1	19	24 - 12 oz bottles	0
3	Aniseed Syrup	1	10	12 - 550 ml bottles	0
4	Chef Antons Cajun Seasoning	2	22	48 - 6 oz jars	0
5	Chef Antons Gumbo Mix	2	21.35	36 boxes	1
6	Grandmas Boysenberry Spread	3	25	12 - 8 oz jars	0
7	Uncle Bobs Organic Dried Pears	3	30	12 - 1 lb pkgs.	0
8	Northwoods Cranberry Sauce	3	40	12 - 12 oz jars	0
9	Mishi Kobe Niku	4	97	18 - 500 g pkgs.	1
10	Ikura	4	31	12 - 200 ml jars	0

Showing 1 to 10 of 78 entries Previous 1 2 3 4 5 ... 8 Next

Awesome! Here, we have the names of the products!

Also, we can observe that there is a column called **is_discontinued**, which has the value 0 or 1. What does this 0 mean?

ESR:
False

That's right! Therefore, all the products with value 1 are now discontinued. We do not want those products in our data!

I think now we have seen everything we needed to. Time to join the 2 tables!

Teacher guides the student in joining the two tables. Let the student try it themselves. If they are really stuck, only then give them hints/help.

Student writes the query to join the 2 tables to get the desired output

```sql
SELECT
    suppliers.company_name,
    suppliers.contact_name,
    suppliers.phone,
    company_products.name
FROM company_products
INNER JOIN suppliers
ON
    company_products.supplier_id=suppliers.id
AND
    company_products.is_discontinued=0;
```

With Output -

Output

Show 10 entries

company_name	contact_name	phone	name
Aux joyeux ecclésiastiques	Guylène Nodier	(1) 03.83.00.68	Côte de Blaye
Aux joyeux ecclésiastiques	Guylène Nodier	(1) 03.83.00.68	Chartreuse verte
Bigfoot Breweries	Cheryl Saylor	(503) 555-9931	Sasquatch Ale
Bigfoot Breweries	Cheryl Saylor	(503) 555-9931	Steeleye Stout
Bigfoot Breweries	Cheryl Saylor	(503) 555-9931	Laughing Lumberjack Lager
Cooperativa de Quesos Las Cabras	Antonio del Valle Saavedra	(98) 598 76 54	Queso Cabrales
Cooperativa de Quesos Las Cabras	Antonio del Valle Saavedra	(98) 598 76 54	Queso Manchego La Pastora
Escargots Nouveaux	Marie Delamare	85.57.00.07	Escargots de Bourgogne
Exotic Liquids	Charlotte Cooper	(171) 555-2222	Chai
Exotic Liquids	Charlotte Cooper	(171) 555-2222	Chang

Showing 1 to 10 of 70 entries Previous 1 2 3 4 5 6 7 Next

Awesome! Now try tweaking the Inner Join with Left, Right and Full.

You will notice that -

1. Inner Join has - 70 rows 2. Left Join has - 78 rows 3. Right Join has - 71 rows 4. Full Join has - 79 rows Can you explain why this is the case, based on the statement we constructed?	**ESR:** Varied!
• **INNER JOIN** - In this case, it only displays those products that have a supplier and are not discontinued. • **LEFT JOIN** - In this case, it displays all the data from the Inner Join, plus some extra rows for the products that either do not have a supplier or are discontinued. • **RIGHT JOIN -** In this case, it displays all the data from the Inner Join, plus some extra rows from the suppliers that do not supply any products anymore. • **FULL JOIN -** In this case, it displays all the rows from both the tables.	
With this, we now have clarity on how JOIN works! In the next class, we will learn about UNION statements!	
Teacher Guides Student to Stop Screen Share	
WRAP UP SESSION - 5 Mins	
Quiz time - Click on in-class quiz	
Question	**Answer**
Which join refers to the join records from the right table that has no matching key in the left table? A. Right Join B. Inner Join C. Left Join	**A**

D. Outer Join	
Which query is used that retrieves rows from one or more tables? A. Select B. Join C. Where D. None of the above	**B**
When do you apply the inner join statement? A. Matching values in both tables B. Both A and C C. Different values in both tables D. None of the above	**A**
End the quiz panel	
FEEDBACK • **Appreciate the students for their efforts in the class.** • **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**	
Teacher Action	**Student Action**
You get Hats off for your excellent work! In the next class we will learn about SQL Union	*Make sure you have given at least 2 Hats Off during the class for:* 
Project Discussion	

You were approached by a friend, who is trying to learn MySQL and is stuck on trying to find answers to simple questions like getting all the users who are from a particular state, or which neighborhood has the most number of users. Your task is to help your friend in trying to find these data attributes.	
Teacher Clicks 	
ADDITIONAL ACTIVITIES	
Additional Activities *Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? ○ Describe what happened. ○ The code I wrote. ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The student uses the markdown editor to write her/his reflections in the reflection journal.*

ACTIVITY LINKS

Activity Name	Description	Link
Teacher Activity1	SQL Editor	http://ec2-3-108-196-161.ap-south-1.compute.amazonaws.com/editor
Student Activity 1	SQL Editor	http://ec2-3-108-196-161.ap-south-1.compute.amazonaws.com/editor