

# SQL CODING ASSESSMENT

## 2) Explain joins with examples.

- Join statement is used to combine data or rows from two or more tables based on a common field between them.
- The join clause allows us to **retrieve data from two or more related tables** into a meaningful result set.
- We can join the table using a **SELECT** statement and a **join condition**. It indicates how SQL Server can use data from one table to select rows from another table.
- 

### Different types of joins:

#### 1) Inner Join:

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left shows the database structure, including tables like 'Student' and 'Payment'. The 'Query Editor' in the center contains the following SQL code:

```
select sum(amount) as Total_Amount
from Payment;

-- Inner join
select * from student s
inner join payment p on p.StudentID=s.StudentID
```

The 'Results' pane at the bottom shows the output of the query, which is a table with 10 rows and 10 columns. The columns are: StudentID, Surname, lastname, DateOfBirth, Email, PhoneNumber, PaymentID, StudentID, Amount, and PaymentDate. The data represents a list of students and their corresponding payment records.

StudentID	Surname	lastname	DateOfBirth	Email	PhoneNumber	PaymentID	StudentID	Amount	PaymentDate
1	Vishal	Bachu	2000-09-17	vishal@gmail.com	9876543210	501	101	8900	2022-09-15
2	Nitin	Sarna	2007-06-02	sarna@gmail.com	9876500000	502	104	9000	2021-11-21
3	Arunthi	Mahra	2001-06-27	arunthi@gmail.com	9800043210	503	102	7500	2020-10-17
4	Archiya	Phapala	2000-06-16	archiya@gmail.com	9132320442	504	107	6200	2022-09-11
5	Brigha	Vole	2000-09-19	scile@gmail.com	9177864130	505	108	20000	2022-08-11
6	Sandhya	Vera	2000-06-11	vera@gmail.com	9701169824	506	109	12000	2023-07-12
7	Aryen	Sai	2000-12-19	sai@gmail.com	9870011122	507	110	43222	2022-05-28
8	Arunthi	Mahra	2001-06-27	arunthi@gmail.com	9800043210	508	102	8900	2021-06-28
9	Ashraf	Syed	2003-08-12	syed@gmail.com	9811143210	509	103	8900	2022-03-11
10	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238787	510	105	9050	2022-02-19

### Explanation:

In this query inner join joins student and payment atble and gives recors till condition p.studentid=s.studentid is true.

## 2) Left Join

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
inner join payment p on p.StudentID=s.StudentID

--left join
select * from student s
left join Payment p on p.StudentID=s.StudentID
```

The Results pane displays the output of the query, showing 11 rows. The columns are: StudentID, firstame, lastname, DateOfBirth, Email, PhoneNumber, PaymentID, StudentID, Amount, and PaymentDate. The data is as follows:

StudentID	firstame	lastname	DateOfBirth	Email	PhoneNumber	PaymentID	StudentID	Amount	PaymentDate
101	Vyethnari	Bacchu	2000-09-17	vyeth@gmail.com	9876543210	501	101	8900	2022-09-15
102	Avanthi	Mahra	2001-06-27	avanthi@gmail.com	9800043210	503	102	7500	2020-10-17
102	Avanthi	Mahra	2001-06-27	avanthi@gmail.com	9800043210	508	102	5600	2021-06-26
103	Ashraf	Syed	2007-06-02	syed@gmail.com	9811143210	509	103	8900	2022-03-11
104	Nithin	Soma	2007-06-02	soma@gmail.com	9876500000	502	104	9000	2021-11-21
105	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238787	510	105	9050	2022-02-19
106	Rani	Vulture	2000-11-13	rani@gmail.com	9912247371	NULL	NULL	NULL	NULL
107	Archana	Puppala	2000-06-16	archana@gmail.com	9182320242	504	107	6200	2022-08-11
108	Snigha	Vole	2000-09-19	voile@gmail.com	9177364130	505	108	29000	2022-08-11
109	Sandhya	Vara	2000-06-11	vara@gmail.com	9701168824	506	109	12000	2023-07-12
110	Ayan	Sai	2000-12-19	sai@gmail.com	9870011122	507	110	43222	2022-05-28

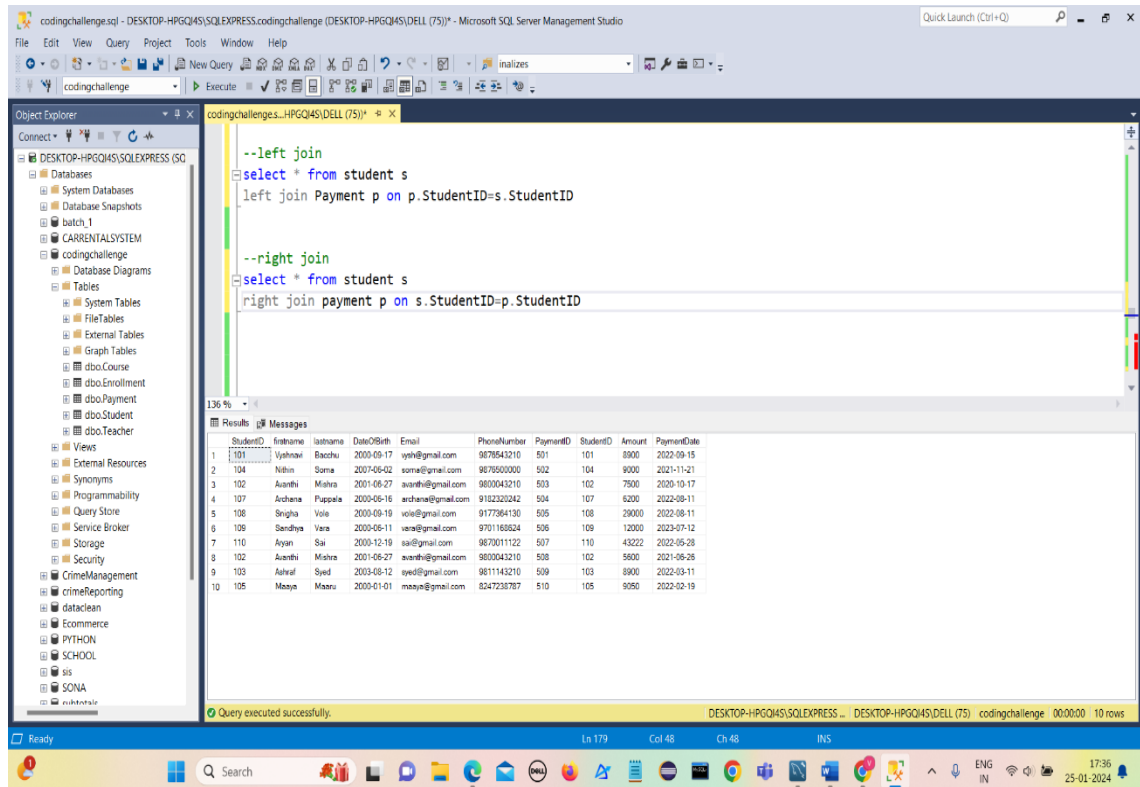
The status bar at the bottom indicates "Query executed successfully." and "11 rows".

### Explanation:

In this query left join is performed on student and payment table which gives all records from student table and matching records from payment table.

### 3) Right join

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--left join
select * from student s
left join Payment p on p.StudentID=s.StudentID

--right join
select * from student s
right join payment p on s.StudentID=p.StudentID
```

The Results pane displays the output of the right join query, showing 10 rows. The columns are: StudentID, FirstName, LastName, DateOfBirth, Email, PhoneNumber, PaymentID, StudentID, Amount, and PaymentDate.

StudentID	FirstName	LastName	DateOfBirth	Email	PhoneNumber	PaymentID	StudentID	Amount	PaymentDate
101	Niharika	Bachu	2000-09-17	niharika@gmail.com	9876543210	501	101	8900	2022-09-15
104	Nidhi	Sarna	2007-06-02	sarna@gmail.com	9876500000	502	104	9000	2021-11-21
102	Aarathi	Mahra	2001-06-27	aarathi@gmail.com	9800043210	503	102	7500	2020-10-17
107	Archana	Puppala	2000-06-16	archana@gmail.com	9182320242	504	107	6200	2022-08-11
108	Shrisha	Vole	2000-09-19	voile@gmail.com	9177364130	505	108	29000	2022-08-11
109	Bendhya	Vera	2000-06-11	vera@gmail.com	9701165824	506	109	12000	2023-07-12
110	Aryan	Sai	2000-12-19	sai@gmail.com	9870011122	507	110	43222	2022-05-28
102	Aarathi	Mahra	2001-06-27	aarathi@gmail.com	9800043210	508	102	5600	2021-06-26
103	Ashraf	Syed	2003-08-12	syed@gmail.com	9811143210	509	103	8900	2022-03-11
105	Mazya	Mazru	2000-01-01	mazya@gmail.com	8247238787	510	105	9050	2022-02-19

#### Explanation:

In this query right join is performed on student and payment table which gives all records from payment table and matching records from student table.

## 4) Cross join

A cross join is a type of join that returns the Cartesian product of rows from the tables in the join. It combines each row from the first table with each row from the second table

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
left join Payment p on p.StudentID=s.StudentID

--right join
select * from student s
right join payment p on s.StudentID=p.StudentID

--cross join
select * from Student
cross join payment
```

The Results pane displays the output of the cross join query, showing a Cartesian product of the Student and Payment tables. The results are as follows:

StudentID	FirstName	LastName	DateOfBirth	Email	PhoneNumber	PaymentID	StudentID	Amount	PaymentDate
1	Vijetha	Beochu	2000-09-17	vijeth@gmail.com	9876543210	501	101	8900	2022-09-15
2	Aarshi	Mahra	2001-06-27	aarshi@gmail.com	9800043210	501	101	8900	2022-09-15
3	Ashraf	Syed	2003-08-12	syed@gmail.com	981143210	501	101	8900	2022-09-15
4	Nithin	Soma	2007-06-02	soma@gmail.com	9876500000	501	101	8900	2022-09-15
5	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238767	501	101	8900	2022-09-15
6	Rani	Vulture	2000-11-13	rani@gmail.com	9912247371	501	101	8900	2022-09-15
7	Archana	Puppala	2000-06-16	archana@gmail.com	9182320242	501	101	8900	2022-09-15
8	Snigha	Vole	2000-09-19	vole@gmail.com	9177364130	501	101	8900	2022-09-15
9	Sandhya	Vera	2000-06-11	vera@gmail.com	970168624	501	101	8900	2022-09-15
10	Ajaya	Bai	2000-12-19	bai@gmail.com	9870011122	501	101	8900	2022-09-15
11	Vijetha	Beochu	2000-09-17	vijeth@gmail.com	9876543210	502	104	9000	2021-11-21
12	Aarshi	Mahra	2001-06-27	aarshi@gmail.com	9800043210	502	104	9000	2021-11-21
13	Ashraf	Syed	2003-08-12	syed@gmail.com	981143210	502	104	9000	2021-11-21
14	Nithin	Soma	2007-06-02	soma@gmail.com	9876500000	502	104	9000	2021-11-21
15	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238767	502	104	9000	2021-11-21
16	Rani	Vulture	2000-11-13	rani@gmail.com	9912247371	502	104	9000	2021-11-21
17	Archana	Puppala	2000-06-16	archana@gmail.com	9182320242	502	104	9000	2021-11-21

### Explanation:

In this query cross join is performed on student and payment table which gives cartesian product of rows from student and payment table. It combines each row from the student table with each row from the payment table.

## 5) Full join

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--cross join
select * from student
cross join payment

-- full join
select * from student s
full join payment p on p.StudentID=s.StudentID

-- self join
select * from students s
```

The Results pane displays the output of the full join query, showing the Cartesian product of the Student and Payment tables. The results are as follows:

StudentID	FirstName	LastName	DateOfBirth	Email	PhoneNumber	PaymentID	StudentID	Amount	PaymentDate
1	Vijetha	Beochu	2000-09-17	vijeth@gmail.com	9876543210	501	101	8900	2022-09-15
2	Aarshi	Mahra	2001-06-27	aarshi@gmail.com	9800043210	503	102	7500	2020-10-17
3	Aarshi	Mahra	2001-06-27	aarshi@gmail.com	9800043210	508	102	8900	2021-06-26
4	Ashraf	Syed	2003-08-12	syed@gmail.com	981143210	509	103	8900	2022-03-11
5	Nithin	Soma	2007-06-02	soma@gmail.com	9876500000	502	104	9000	2021-11-21
6	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238767	510	105	9050	2022-02-19
7	Rani	Vulture	2000-11-13	rani@gmail.com	9912247371	NULL	NULL	NULL	NULL
8	Archana	Puppala	2000-06-16	archana@gmail.com	9182320242	504	107	6200	2022-08-11
9	Snigha	Vole	2000-09-19	vole@gmail.com	9177364130	505	108	29000	2022-08-11
10	Sandhya	Vera	2000-06-11	vera@gmail.com	970168624	506	109	12000	2023-07-12
11	Ajaya	Bai	2000-12-19	bai@gmail.com	9870011122	507	110	43222	2022-05-28

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values

### Explanation:

This full join joins student and payment table and combines the results of both left and right join.

## 6) Self join:

A self join is a regular join that is used to join a table with itself. It basically allows us to combine the rows from the same table based on some specific conditions.

The screenshot shows the Microsoft SQL Server Enterprise Edition interface. The 'Object Explorer' on the left displays the database structure for 'codingchallenge'. The 'Query Editor' in the center contains the following SQL script:

```
select * from student s
full join payment p on p.StudentID=s.StudentID

-- self join
select * from student s
join student s1 on s.StudentID=s1.StudentID
```

The 'Results' pane at the bottom displays the output of the self-join query, showing 10 rows of data. Each row contains two identical sets of student information, as the table is joined to itself.

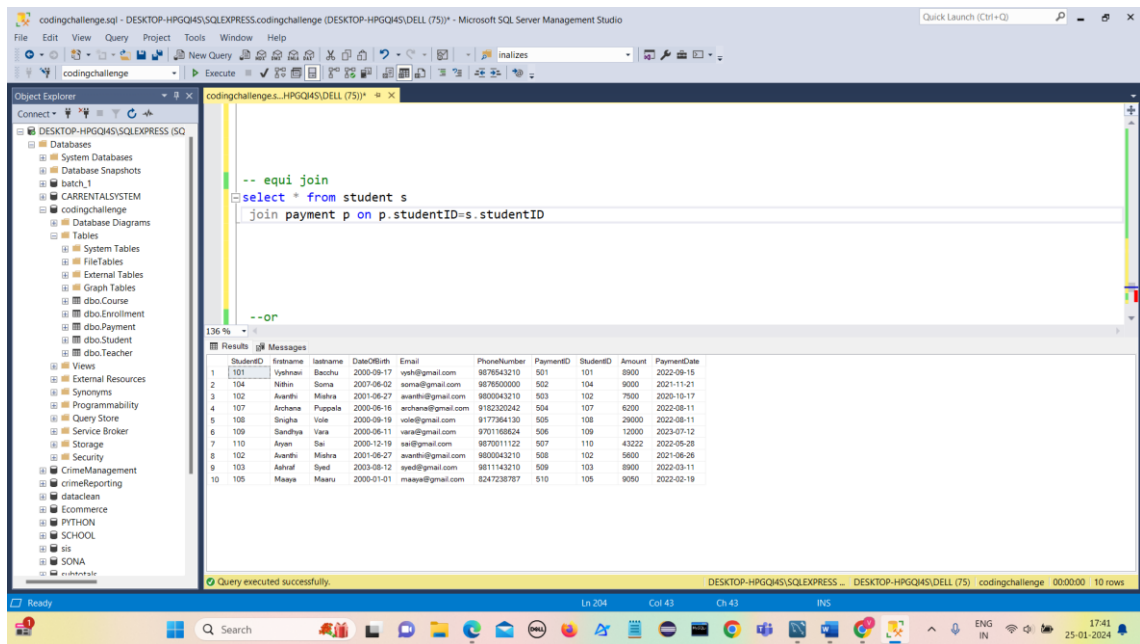
StudentID	firstname	lastname	DateOfBirth	Email	PhoneNumber	StudentID	firstname	lastname	DateOfBirth	Email	PhoneNumber
1	Vyethnnei	Becchu	2000-09-17	vyeth@gmail.com	9876543210	101	Vyethnnei	Becchu	2000-09-17	vyeth@gmail.com	9876543210
2	Awarthi	Mahra	2001-06-27	awarth@gmail.com	9800043210	102	Awarthi	Mahra	2001-06-27	awarth@gmail.com	9800043210
3	Ahnaf	Syed	2003-08-12	syed@gmail.com	9811143210	103	Ahnaf	Syed	2003-08-12	syed@gmail.com	9811143210
4	Nithin	Soma	2007-06-02	soma@gmail.com	9876500000	104	Nithin	Soma	2007-06-02	soma@gmail.com	9876500000
5	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238787	105	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238787
6	Ravi	Vulture	2000-11-13	ravi@gmail.com	9912247371	106	Ravi	Vulture	2000-11-13	ravi@gmail.com	9912247371
7	Archana	Pappala	2000-06-16	archana@gmail.com	9182302042	107	Archana	Pappala	2000-06-16	archana@gmail.com	9182302042
8	Singha	Voie	2000-09-19	voie@gmail.com	9177364130	108	Singha	Voie	2000-09-19	voie@gmail.com	9177364130
9	Sandhya	Vara	2000-06-11	vara@gmail.com	9701168624	109	Sandhya	Vara	2000-06-11	vara@gmail.com	9701168624
10	Ayan	Sai	2000-12-19	sai@gmail.com	9870011122	110	Ayan	Sai	2000-12-19	sai@gmail.com	9870011122

### Explanation:

In this query student table is joined with itself based on a condition.

## 7) Equi join:

EQUI JOIN creates a JOIN for equality or matching column(s) values of the relative tables. EQUI JOIN also create JOIN by using JOIN with ON and then providing the names of the columns with their relative tables to check equality using equal sign (=).

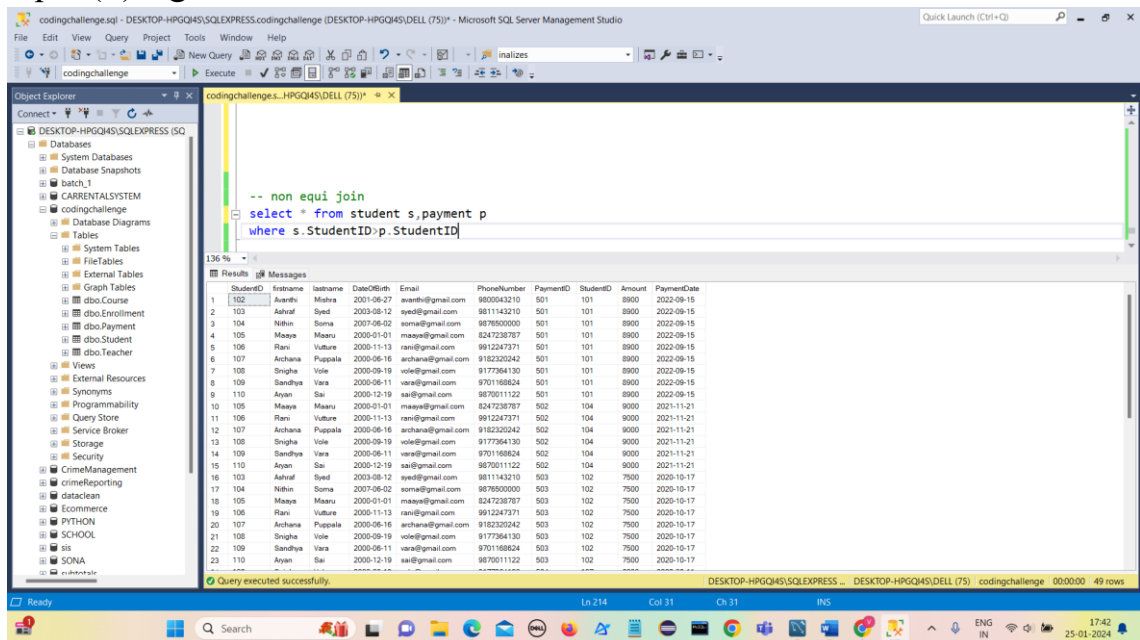


## Explanation:

In this query student and payment table are joined based on a equality condition .The queries are returned when the studentid s are same.

## 8) Non-Equi Join:

NON EQUI JOIN performs a JOIN using comparison operator other than equal(=) sign like >, =, <= with conditions.



**Explanation:**

In this query student and payment table are combined with an operator > .It returns queries when studentid in student table is greater than that in payment table.

## 9) Natural join

A natural join returns all rows by matching values in common columns having same name and data type of columns and that column should be present in both tables.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
select * from student s,payment p
where s.StudentID>p.StudentID

-- 9) Natural join
select * from student
natural join payment;
```

The Results pane displays the output of the natural join query, showing 13 rows of data. The columns are StudentID, FirstName, LastName, DateOfBirth, Email, PhoneNumber, PaymentID, StudentID, Amount, and PaymentDate.

StudentID	FirstName	LastName	DateOfBirth	Email	PhoneNumber	PaymentID	StudentID	Amount	PaymentDate
102	Aashvi	Mathra	2001-06-27	aashvi@gmail.com	9800043210	501	101	8900	2022-09-15
103	Ashraf	Syed	2003-08-12	syed@gmail.com	9811143210	501	101	8900	2022-09-15
104	Nitin	Soma	2007-06-02	soma@gmail.com	9878500000	501	101	8900	2022-09-15
105	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238787	501	101	8900	2022-09-15
106	Ravi	Vulture	2000-11-13	ravi@gmail.com	9912247371	501	101	8900	2022-09-15
107	Archana	Puppala	2000-06-16	archana@gmail.com	9182302042	501	101	8900	2022-09-15
108	Snigha	Vole	2000-09-19	vole@gmail.com	9177364130	501	101	8900	2022-09-15
109	Sandhya	Vara	2000-08-11	vare@gmail.com	970168824	501	101	8900	2022-09-15
110	Apoor	Sai	2000-12-19	sai@gmail.com	9870011122	501	101	8900	2022-09-15
105	Maaya	Maaru	2000-01-01	maaya@gmail.com	8247238787	502	104	9000	2021-11-21
106	Ravi	Vulture	2000-11-13	ravi@gmail.com	9912247371	502	104	9000	2021-11-21
107	Archana	Puppala	2000-06-16	archana@gmail.com	9182302042	502	104	9000	2021-11-21
108	Snigha	Vole	2000-09-19	vole@gmail.com	9177364130	502	104	9000	2021-11-21

**Explanation:**

In this query student and payment table are joined if they have same name and same data type.