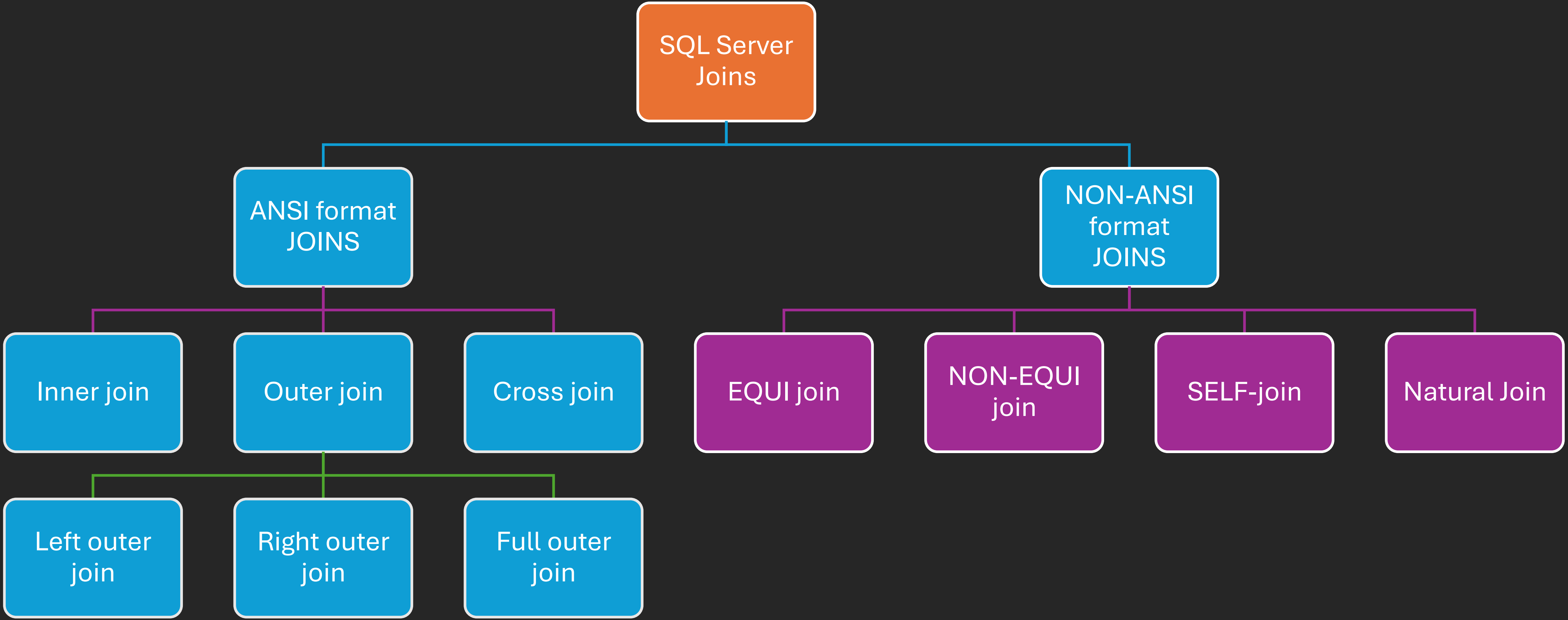
JOIN

- The SQL Server Joins are used to retrieve the data from two or more related tables.
- In general, tables are related to each other using the primary key and foreign key relationship but it is not mandatory.
- The tables involved in the joins must have a common field. And based on that common field the SQL Server JOINS retrieves the records.



SQL Server Joins

ANSI format JOINS

NON-ANSI format JOINS

Inner join

Outer join

Cross join

EQUI join

NON-EQUI join

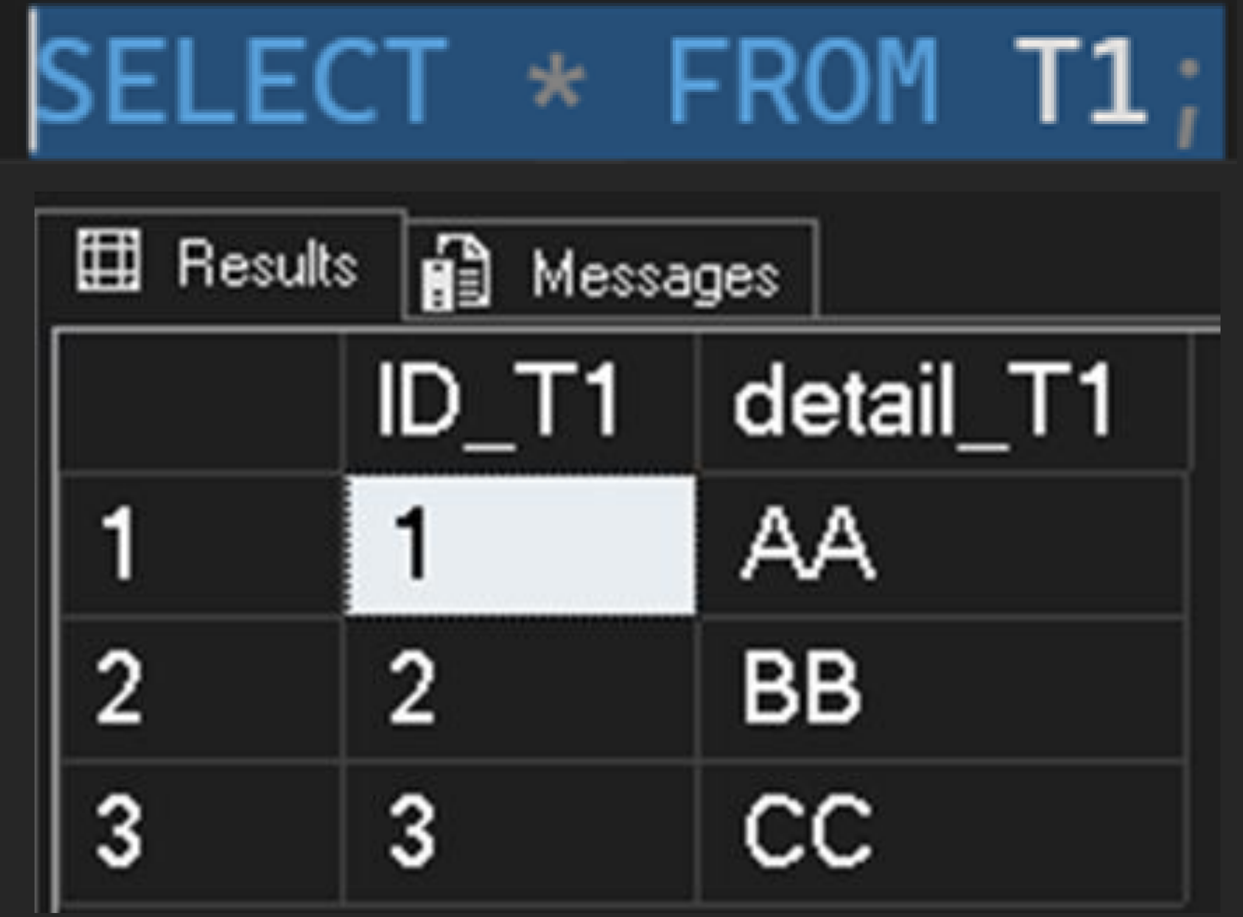
SELF-join

Natural Join

Left outer join

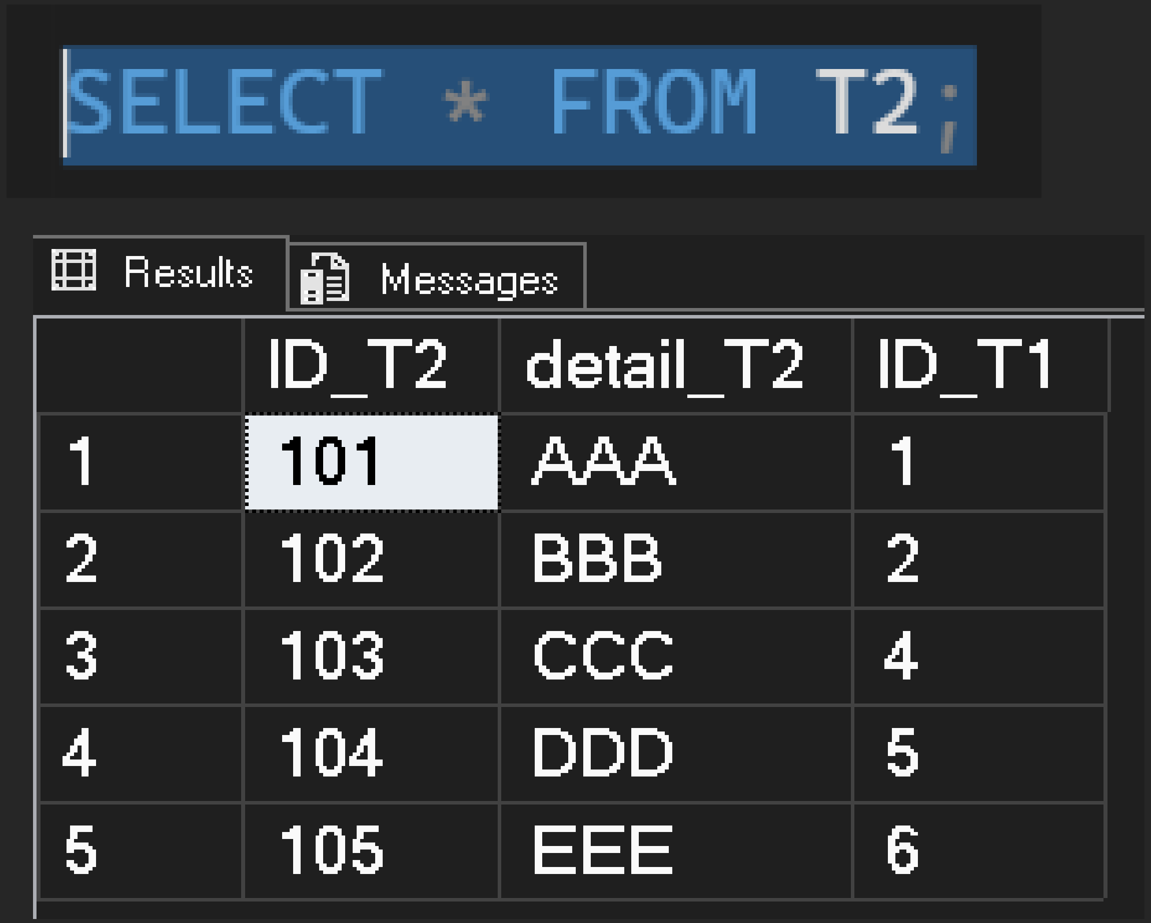
Right outer join

Full outer join



```
SELECT * FROM T1;
```

|  | ID_T1 | detail_T1 |
|---|---|---|
| 1 | 1 | AA |
| 2 | 2 | BB |
| 3 | 3 | CC |



```
SELECT * FROM T2;
```

|  | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|
| 1 | 101 | AAA | 1 |
| 2 | 102 | BBB | 2 |
| 3 | 103 | CCC | 4 |
| 4 | 104 | DDD | 5 |
| 5 | 105 | EEE | 6 |

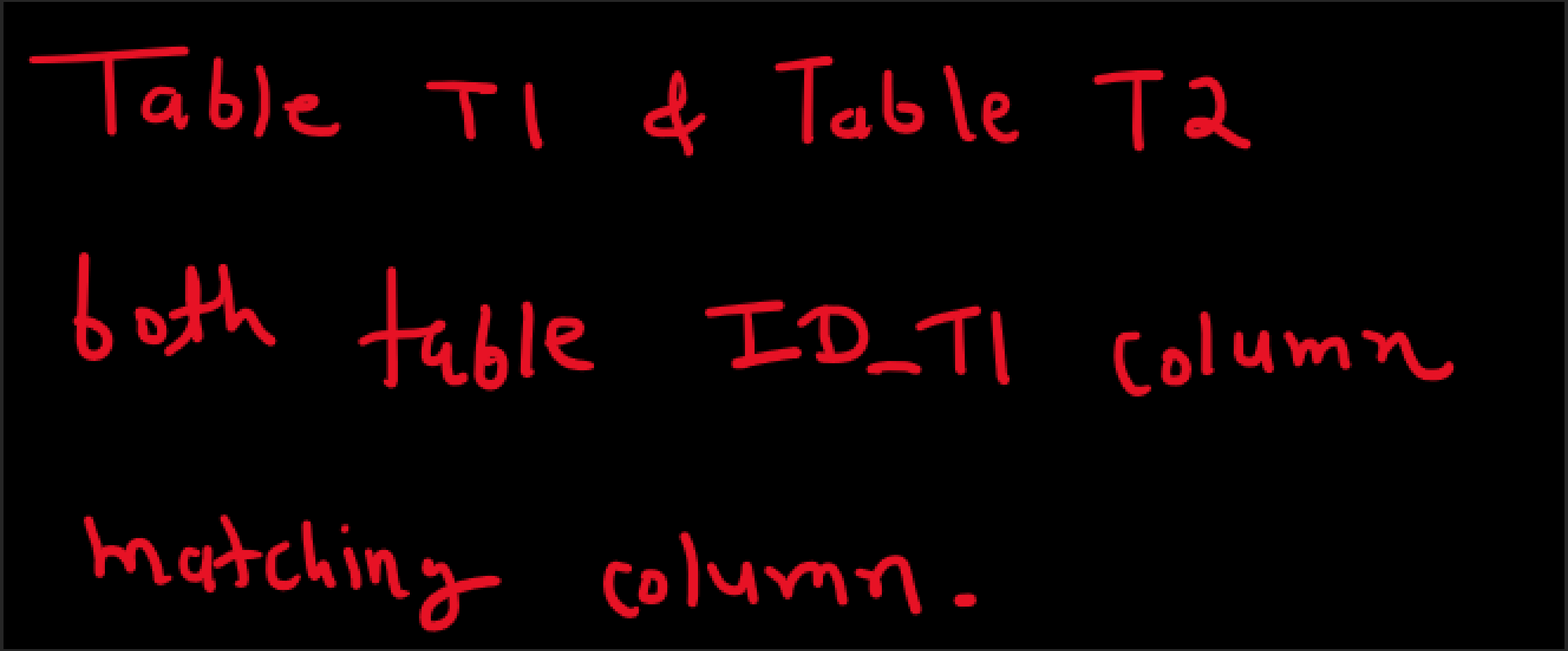


Table T1 & Table T2

both table ID_T1 column matching column.

# INNER JOIN (or JOIN)

Table-1

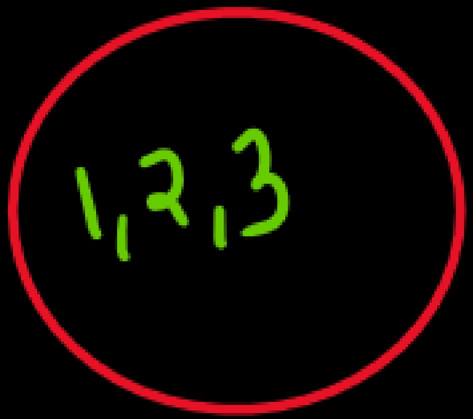
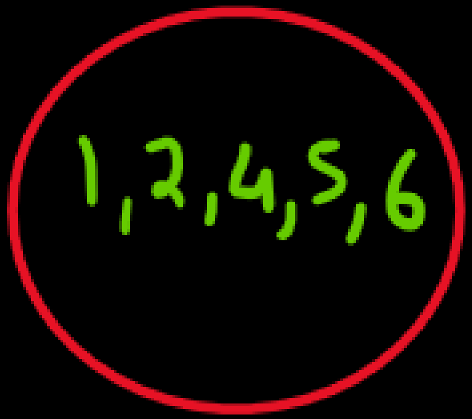
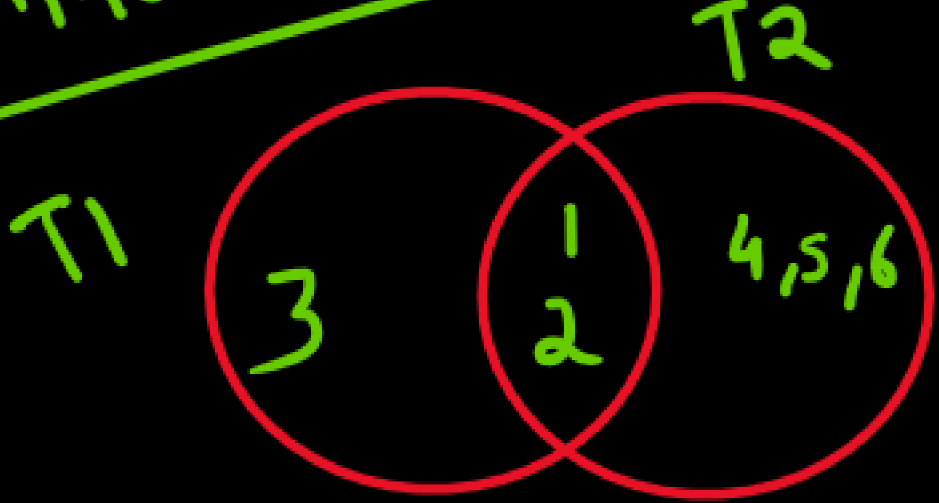
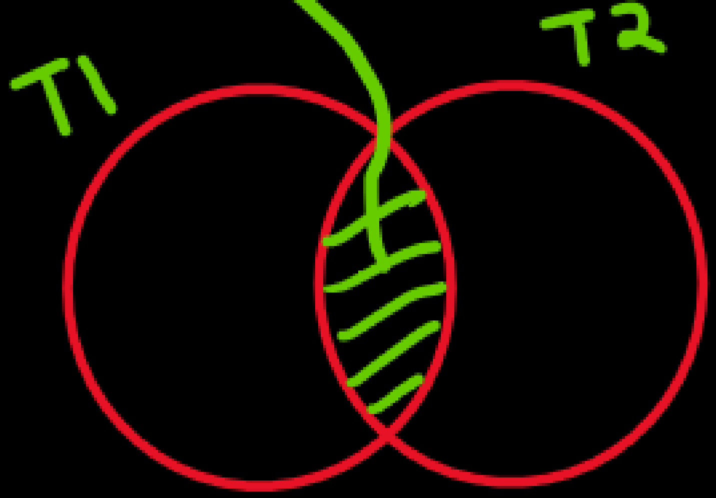


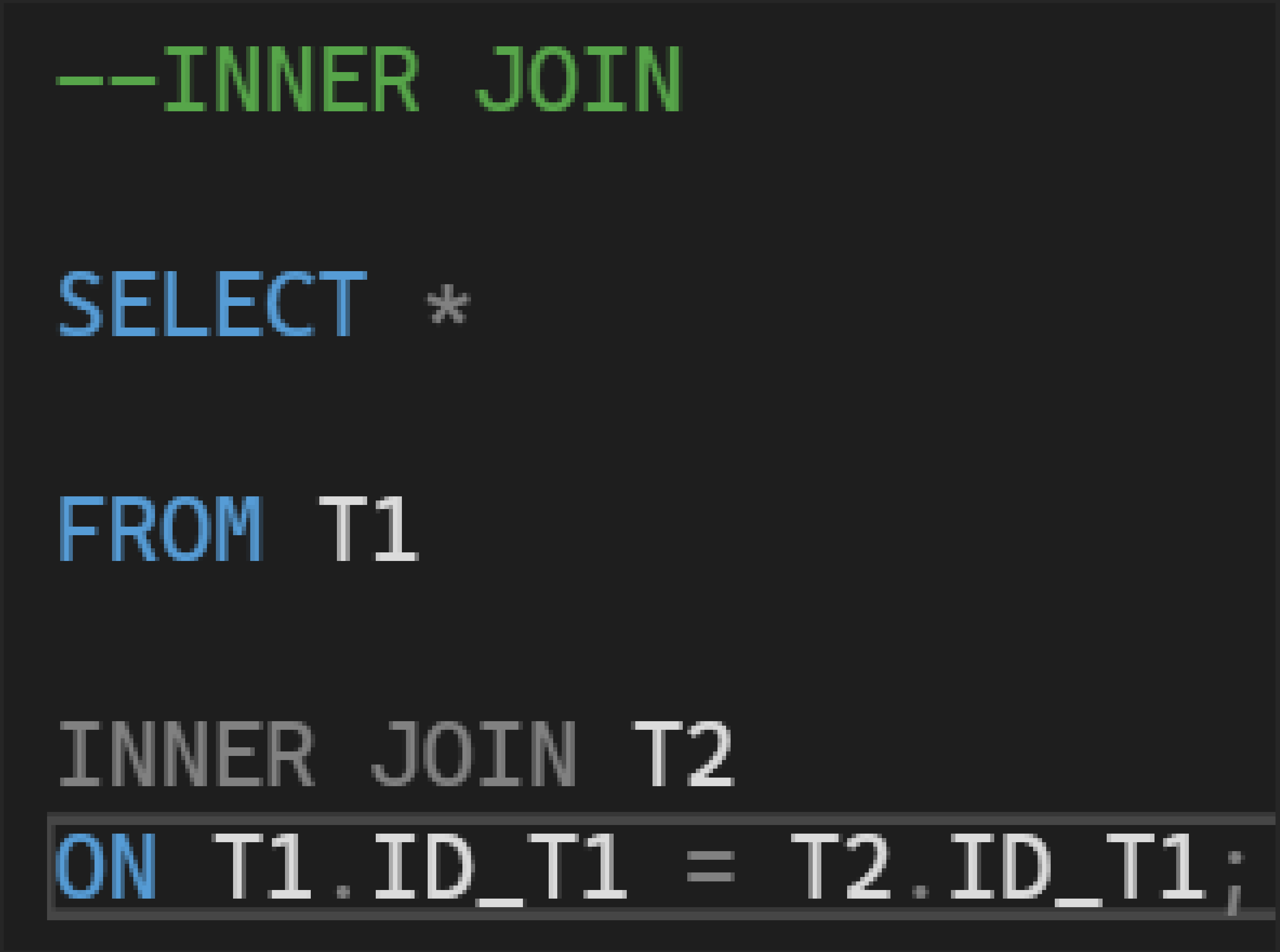
Table-2



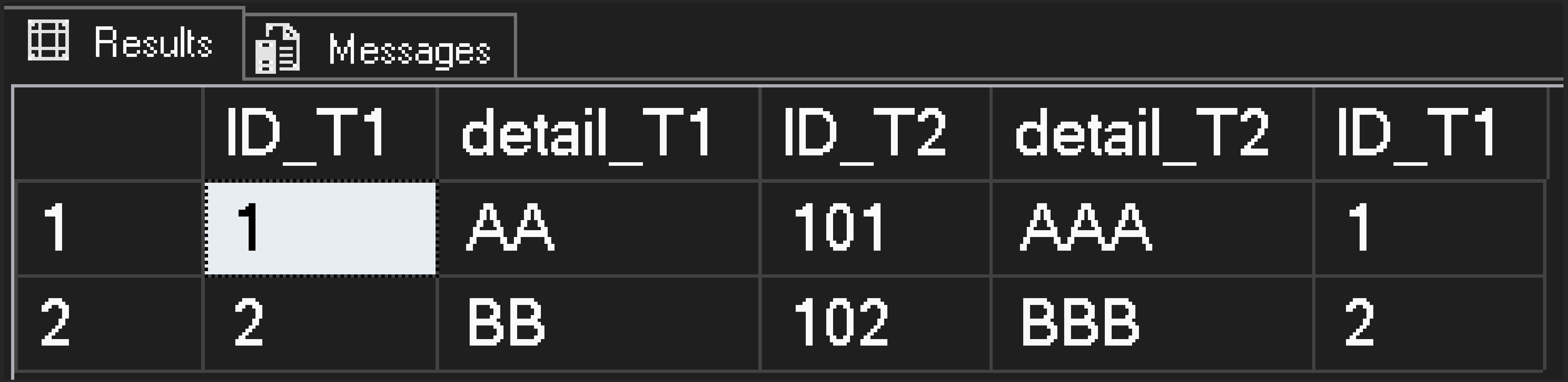
Inner Join







```sql
--INNER JOIN

SELECT *

FROM T1

INNER JOIN T2
ON T1.ID_T1 = T2.ID_T1;
```



| | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 2 | BB | 102 | BBB | 2 |

# LEFT JOIN (or LEFT OUTER JOIN)

Table-1

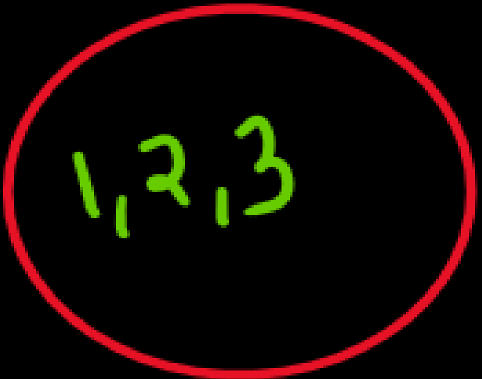
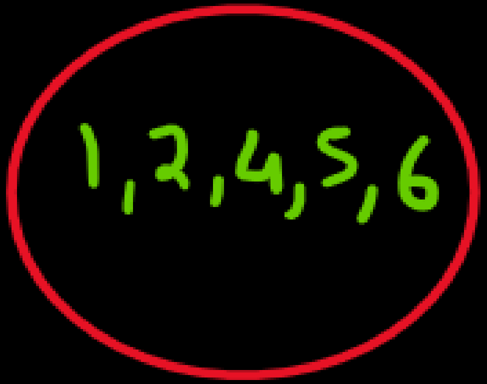
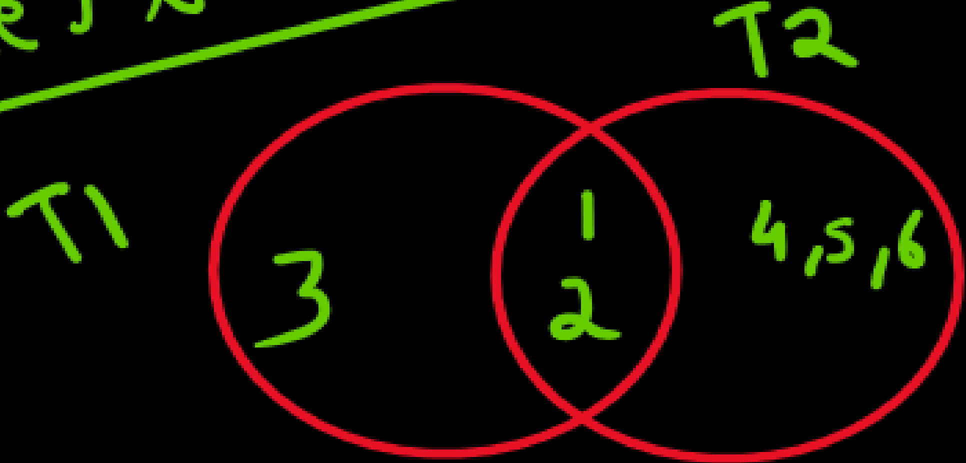
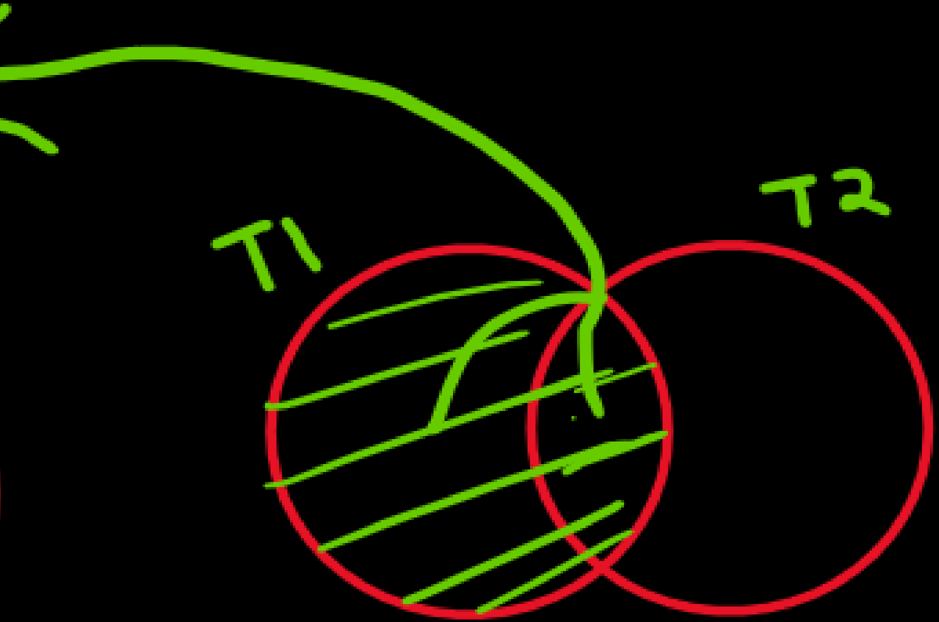


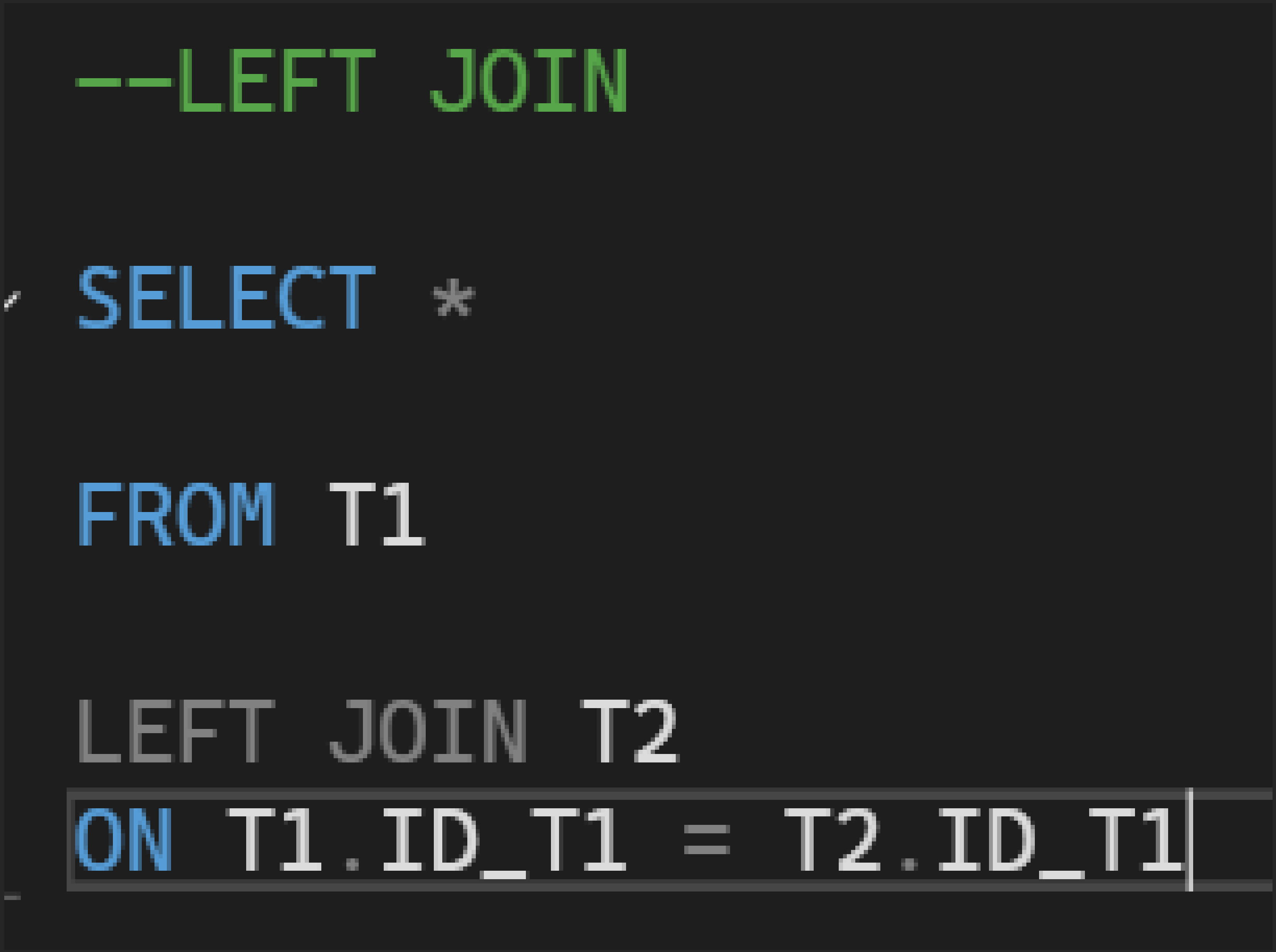
Table-2



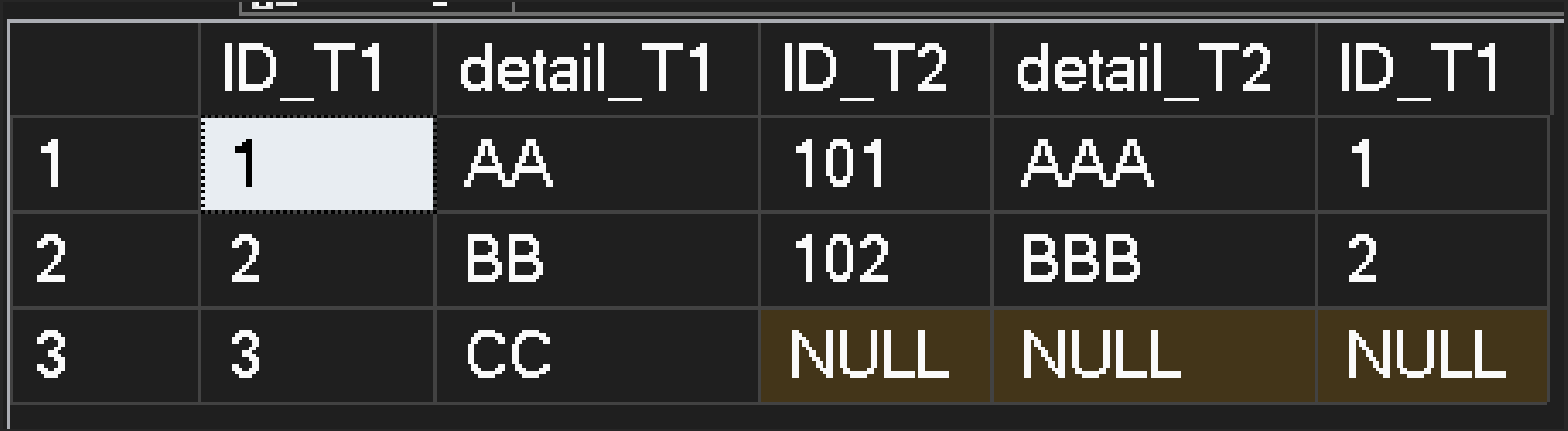
# Left Join







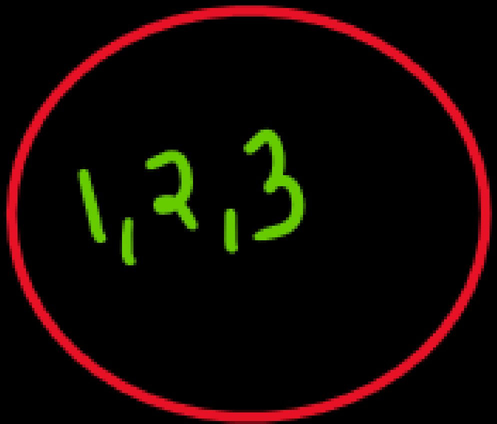
```
--LEFT JOIN

SELECT *

FROM T1

LEFT JOIN T2
ON T1.ID_T1 = T2.ID_T1
```



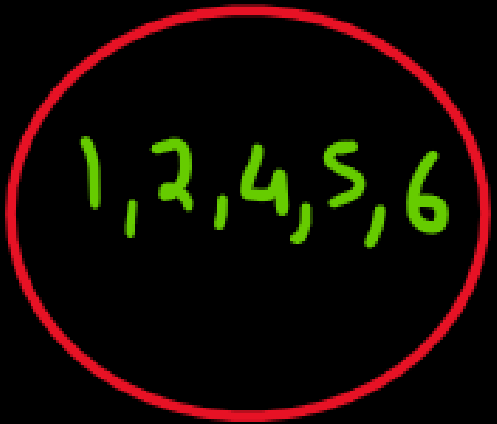
|  | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 2 | BB | 102 | BBB | 2 |
| 3 | 3 | CC | NULL | NULL | NULL |

# RIGHT JOIN (or RIGHT OUTER JOIN)

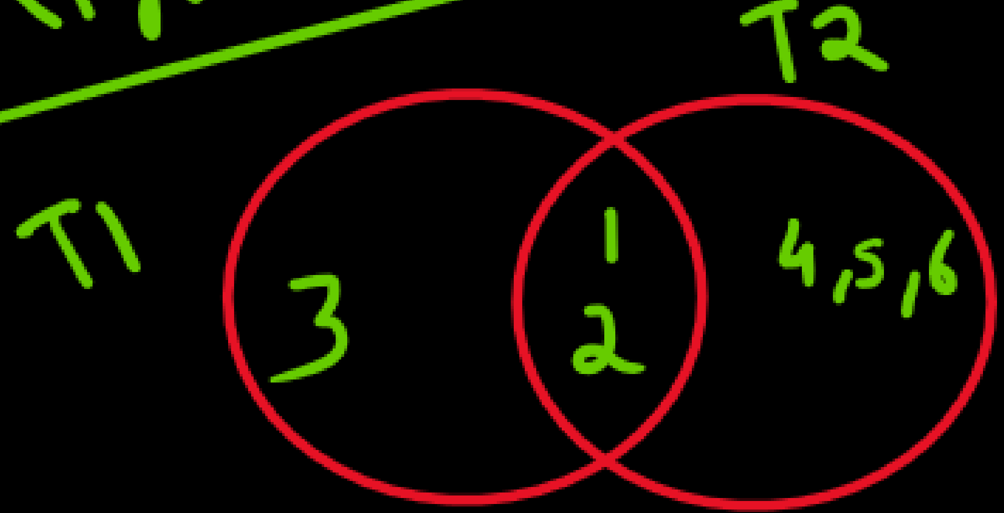
## Table-1

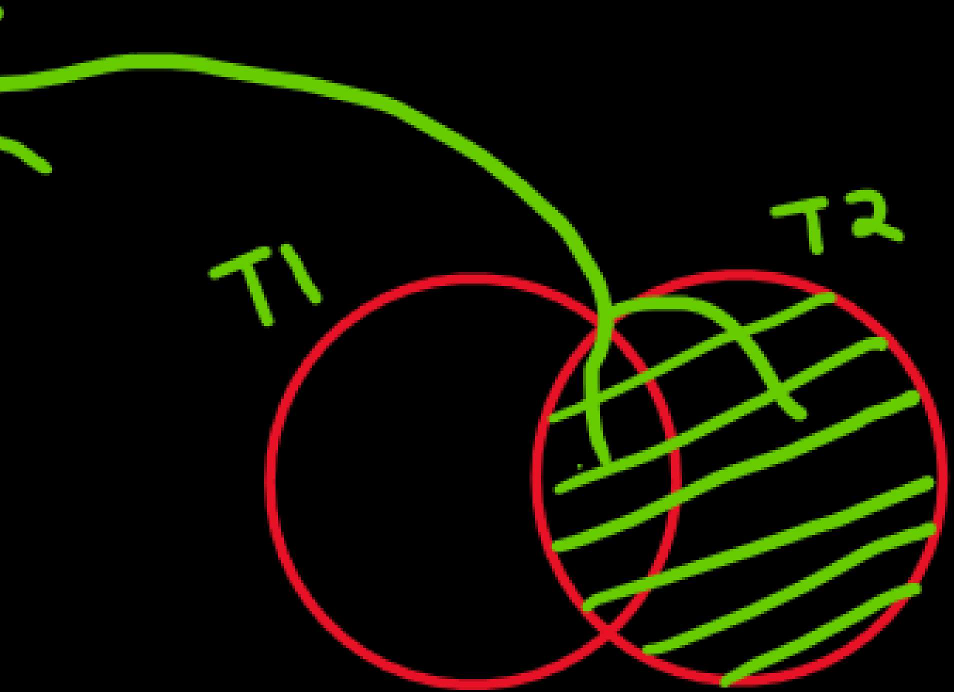


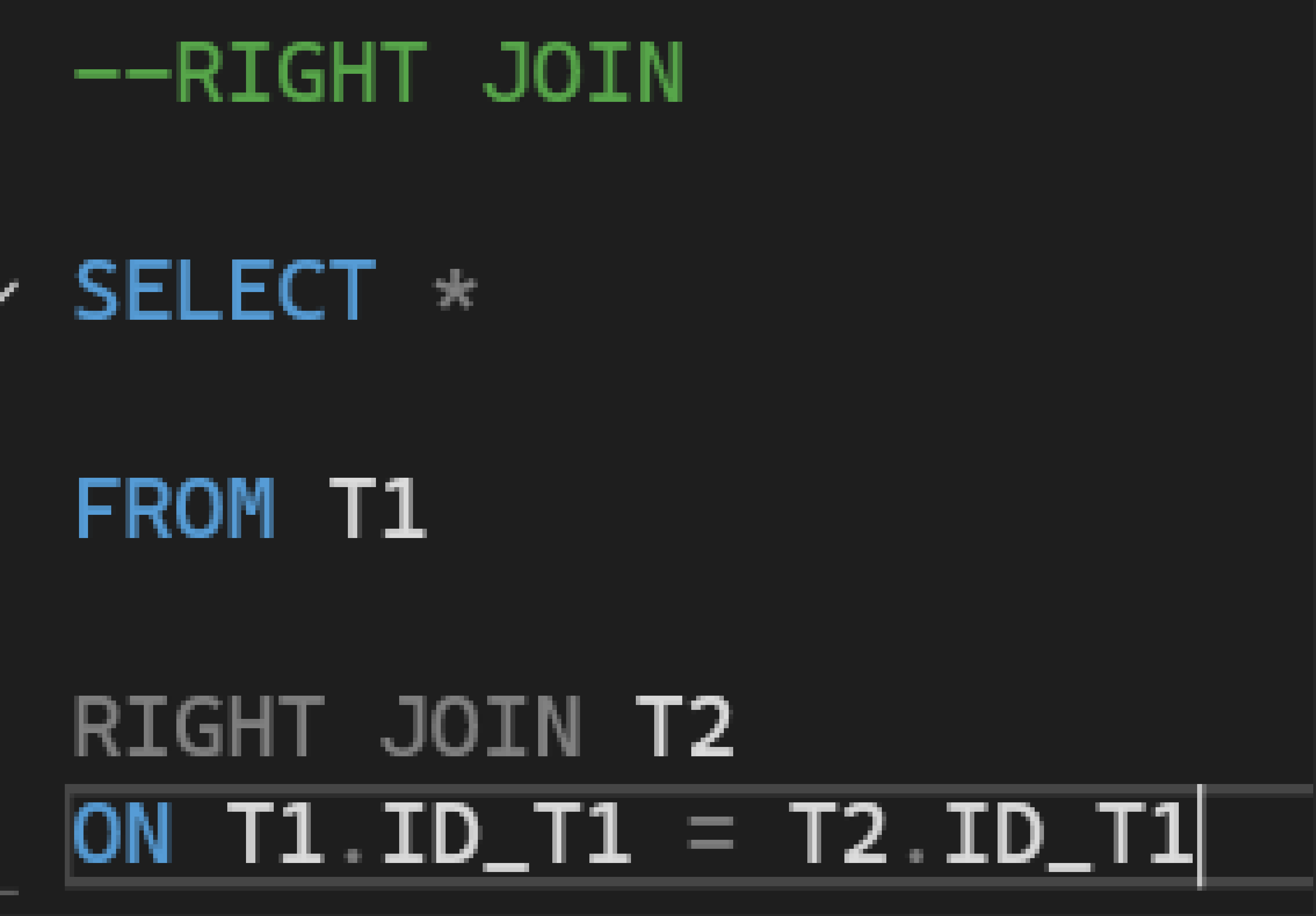
## Table-2



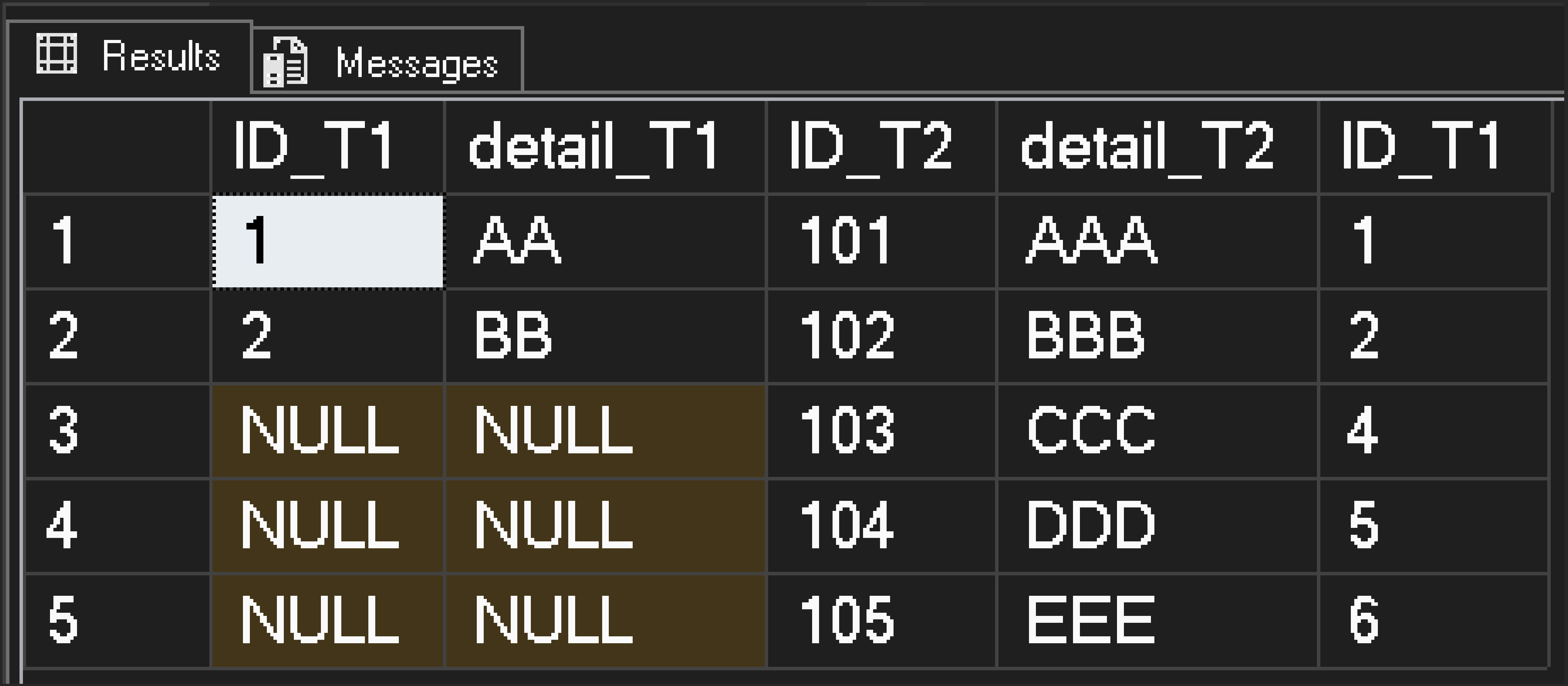
## Right Join







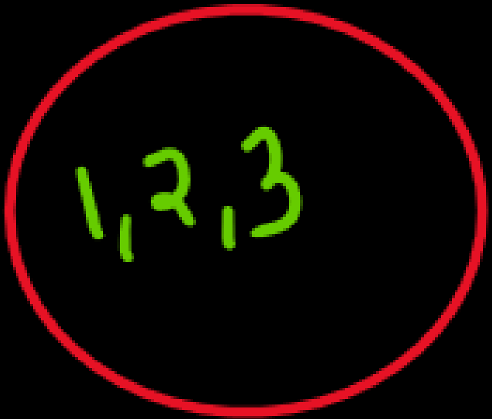
```
--RIGHT JOIN

SELECT *

FROM T1

RIGHT JOIN T2
ON T1.ID_T1 = T2.ID_T1
```



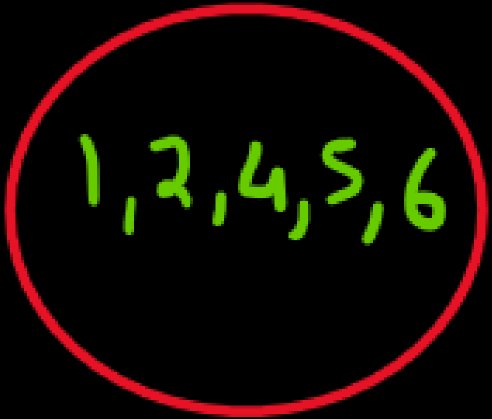
| | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 2 | BB | 102 | BBB | 2 |
| 3 | NULL | NULL | 103 | CCC | 4 |
| 4 | NULL | NULL | 104 | DDD | 5 |
| 5 | NULL | NULL | 105 | EEE | 6 |

# FULL JOIN (or FULL OUTER JOIN)

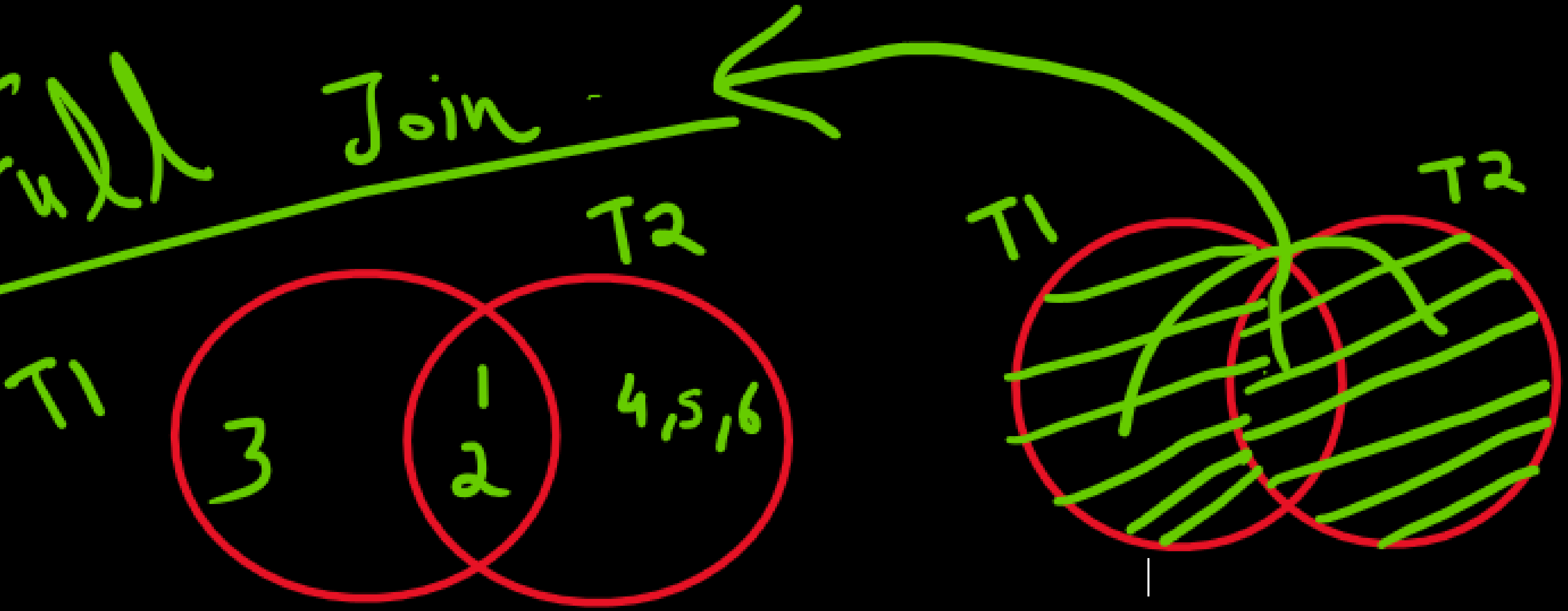
## Table-1

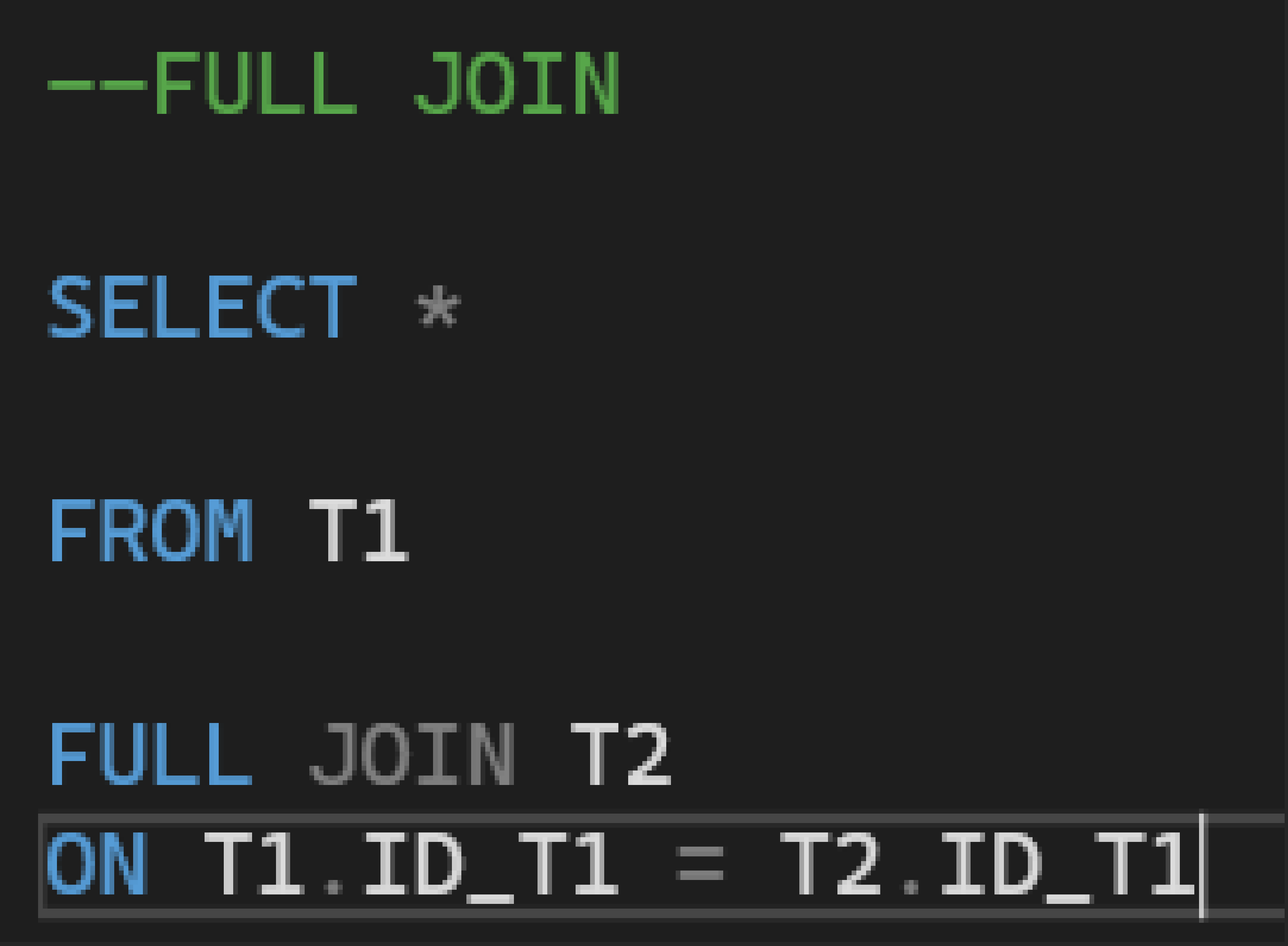


## Table-2

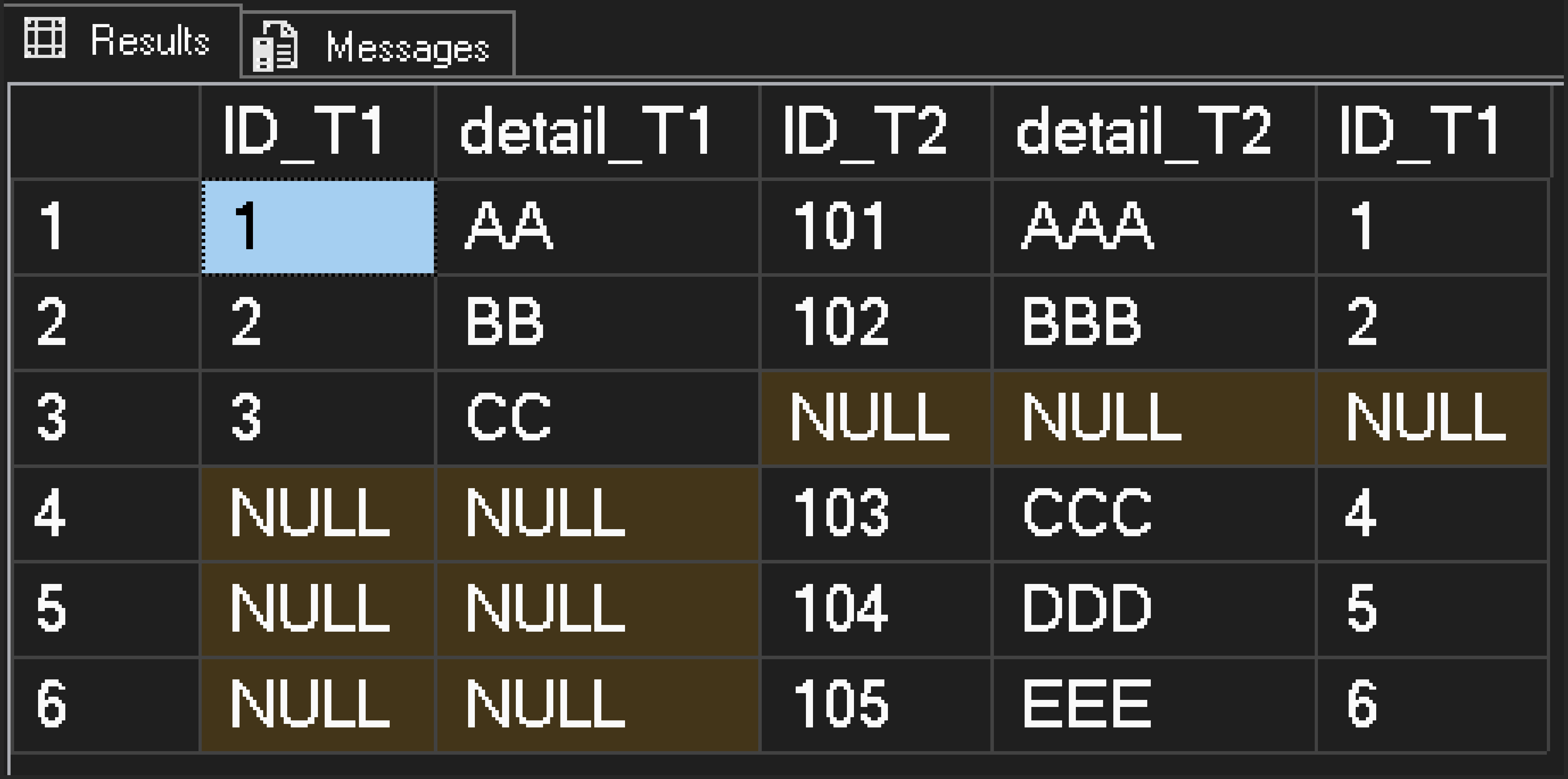


## Full Join



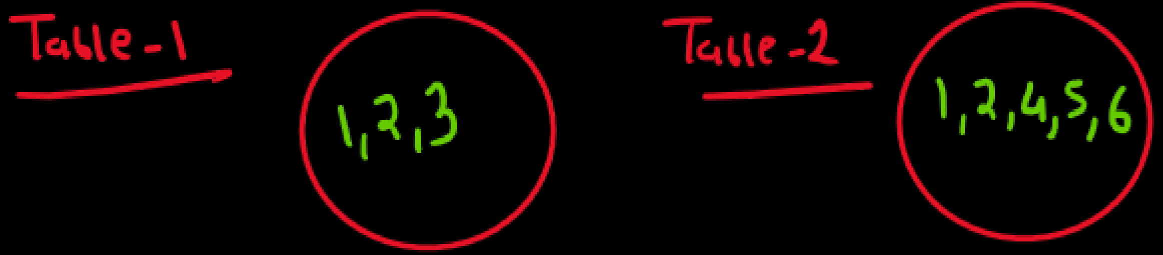


```
--FULL JOIN

SELECT *

FROM T1

FULL JOIN T2
ON T1.ID_T1 = T2.ID_T1
```

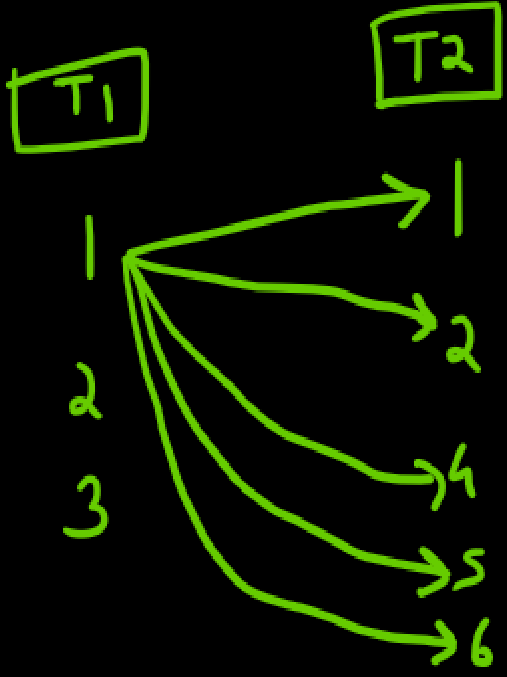


| | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 2 | BB | 102 | BBB | 2 |
| 3 | 3 | CC | NULL | NULL | NULL |
| 4 | NULL | NULL | 103 | CCC | 4 |
| 5 | NULL | NULL | 104 | DDD | 5 |
| 6 | NULL | NULL | 105 | EEE | 6 |

# CROSS JOIN



Cross Join



T1→ m row

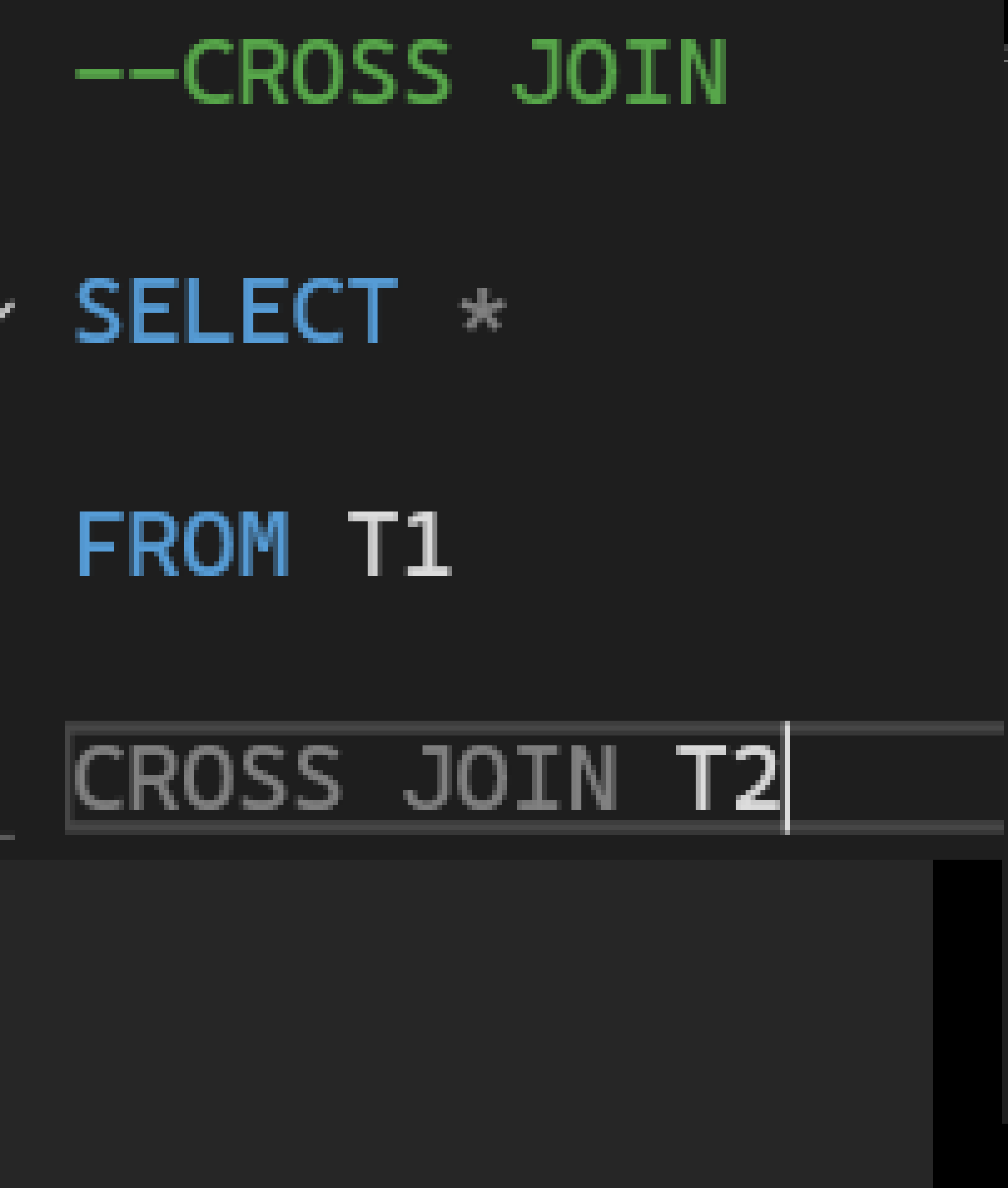
T2→ n row

Cross Join result

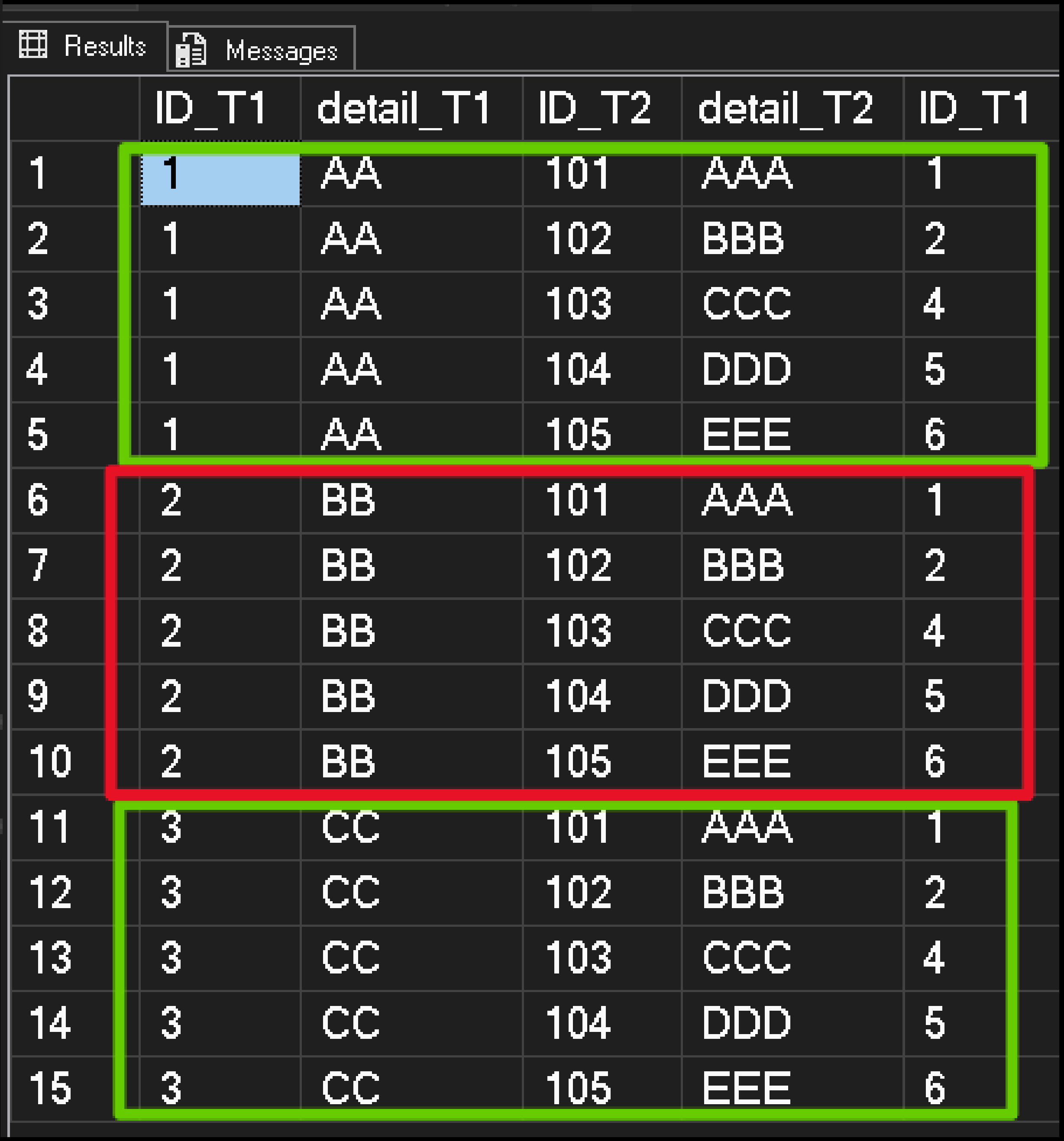
$\boxed{m*n \text{ rows}}$

Here
T1→ m=3 & T2→n=5

Cross Join→ 3*5 = 15 row



```
--CROSS JOIN

SELECT *

FROM T1

CROSS JOIN T2
```

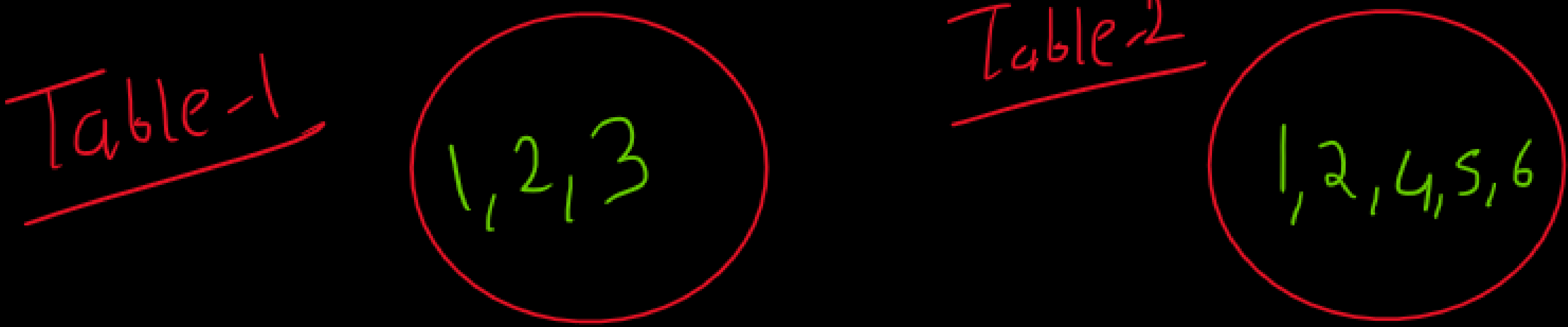


| | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 1 | AA | 102 | BBB | 2 |
| 3 | 1 | AA | 103 | CCC | 4 |
| 4 | 1 | AA | 104 | DDD | 5 |
| 5 | 1 | AA | 105 | EEE | 6 |
| 6 | 2 | BB | 101 | AAA | 1 |
| 7 | 2 | BB | 102 | BBB | 2 |
| 8 | 2 | BB | 103 | CCC | 4 |
| 9 | 2 | BB | 104 | DDD | 5 |
| 10 | 2 | BB | 105 | EEE | 6 |
| 11 | 3 | CC | 101 | AAA | 1 |
| 12 | 3 | CC | 102 | BBB | 2 |
| 13 | 3 | CC | 103 | CCC | 4 |
| 14 | 3 | CC | 104 | DDD | 5 |
| 15 | 3 | CC | 105 | EEE | 6 |

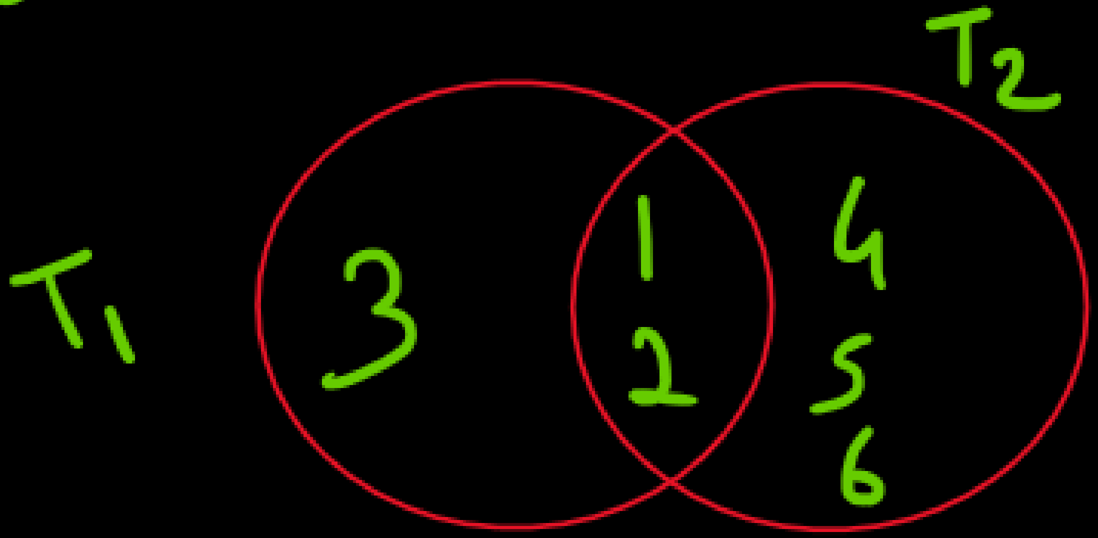
# LEFT ANTI JOIN (T1 – T2)

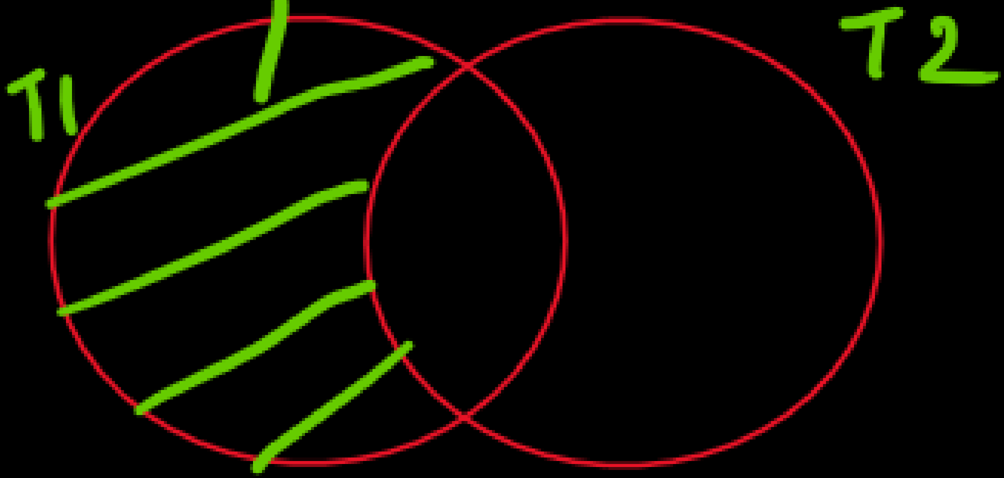
(or Only T1)

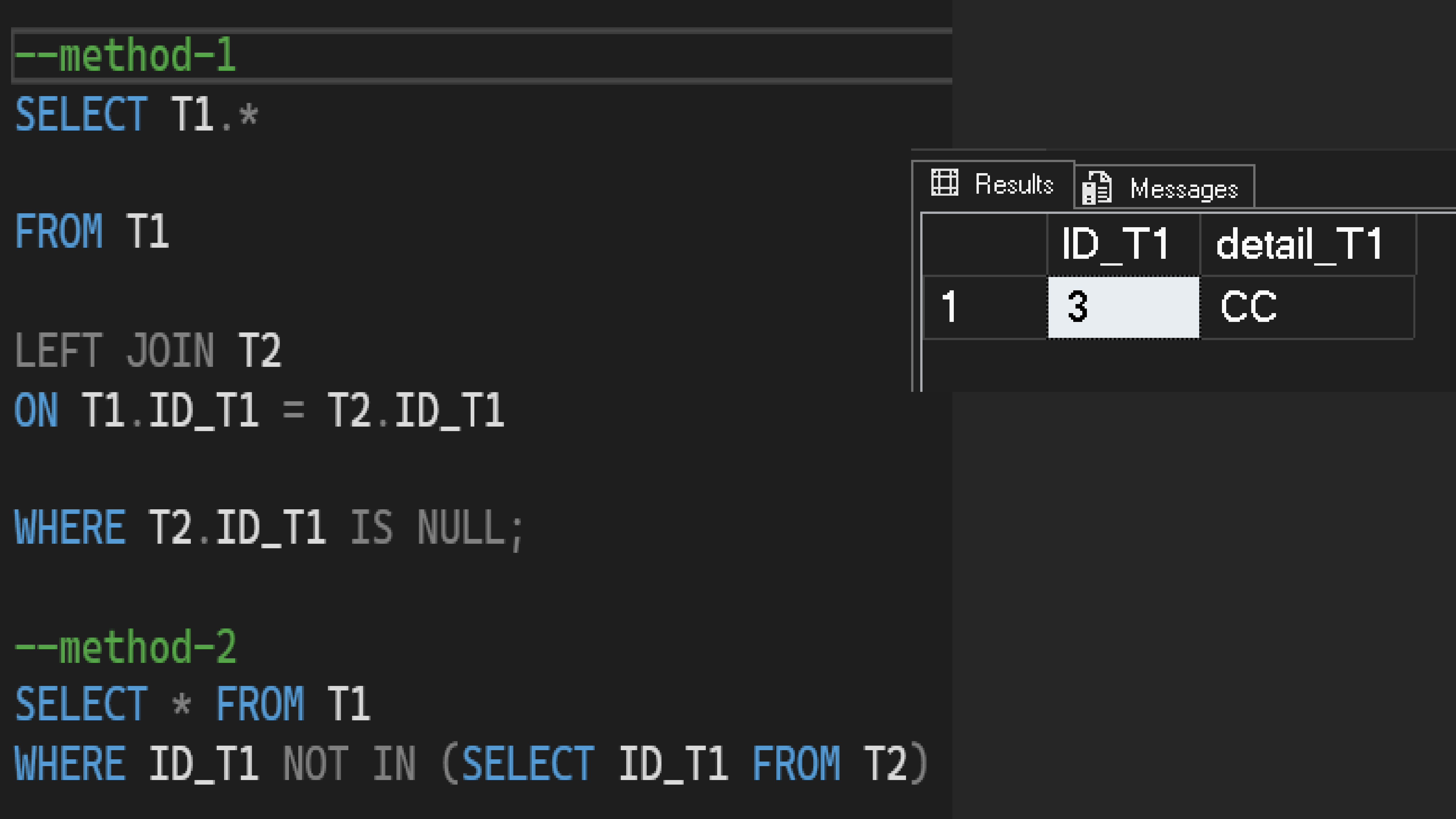
(or Left Join Excluding Inner Join)



Left Anti Join (only T1)







```
--method-1
SELECT T1.*

FROM T1

LEFT JOIN T2
ON T1.ID_T1 = T2.ID_T1

WHERE T2.ID_T1 IS NULL;

--method-2
SELECT * FROM T1
WHERE ID_T1 NOT IN (SELECT ID_T1 FROM T2)
```

|  | ID_T1 | detail_T1 |
|---|---|---|
| 1 | 3 | CC |

# RIGHT ANTI JOIN (T1 – T2)

(or Only T2)

(or Right Join Excluding Inner Join)

Table-1

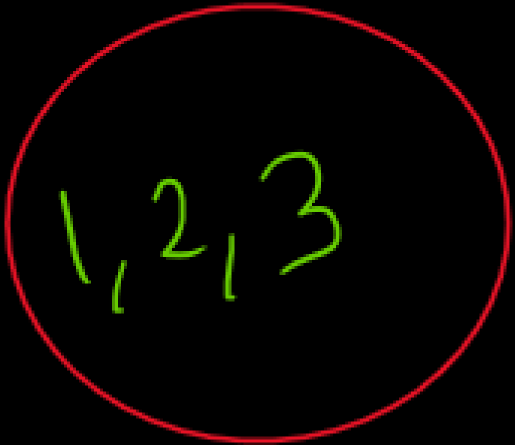
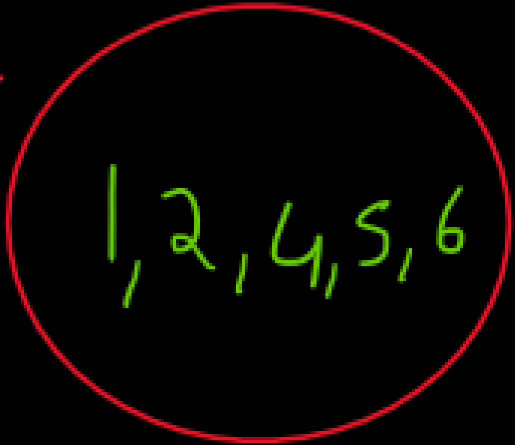
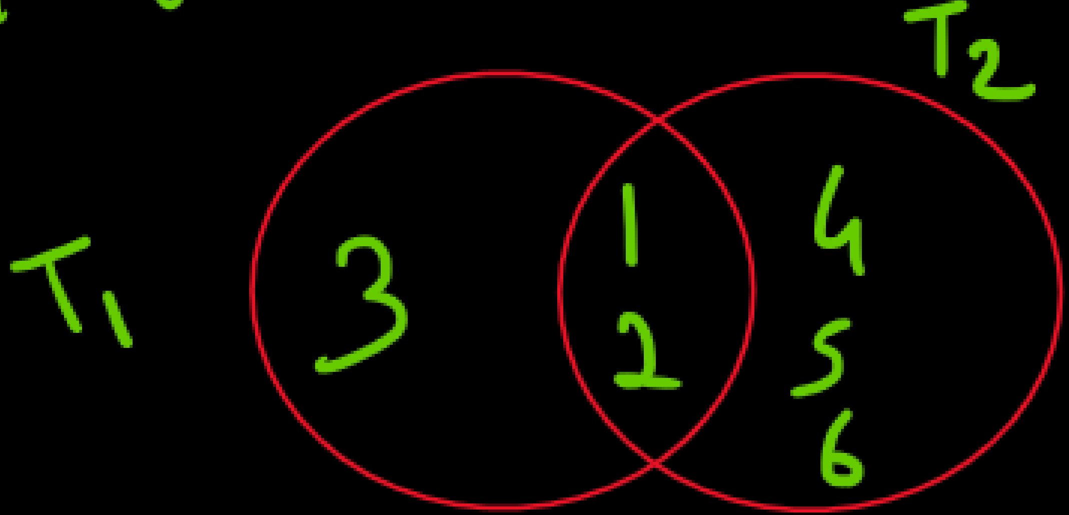
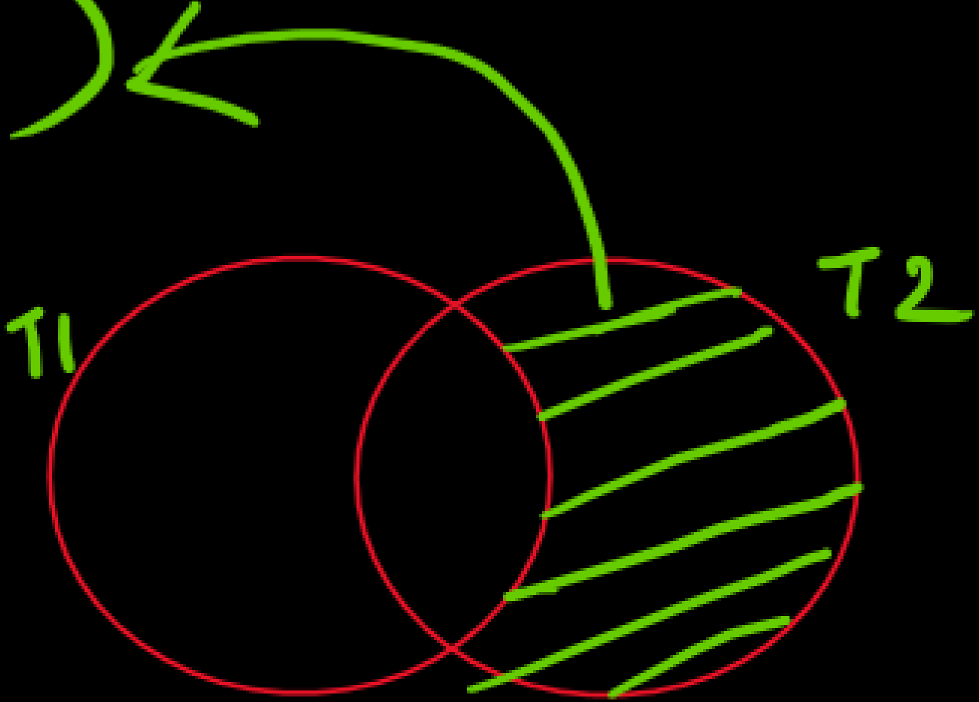


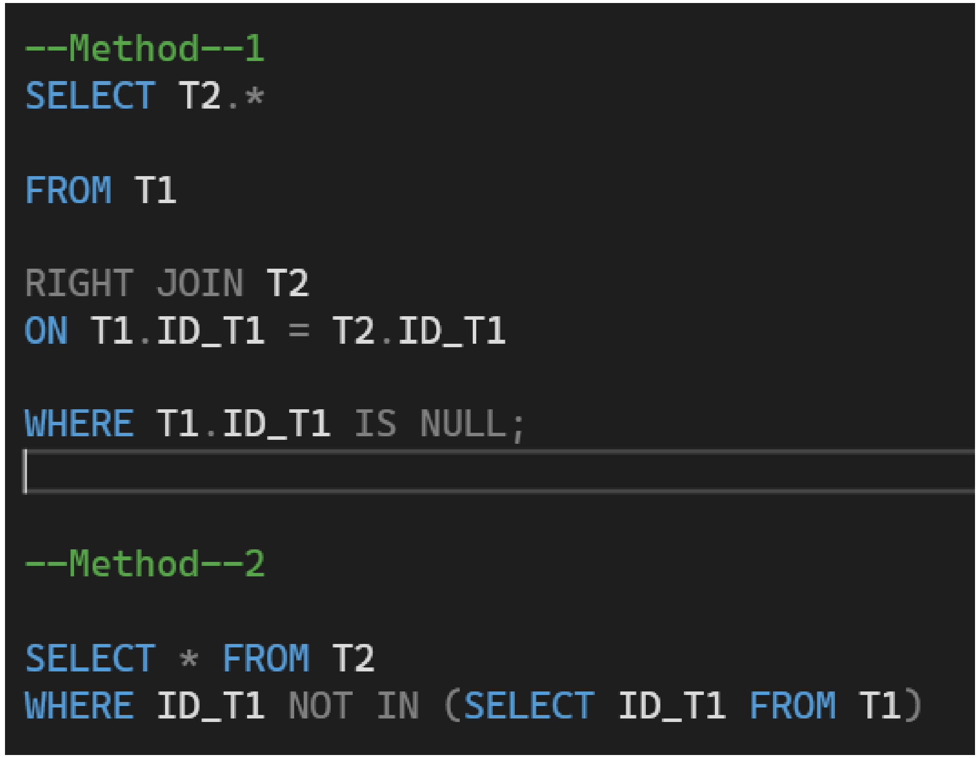
Table-2



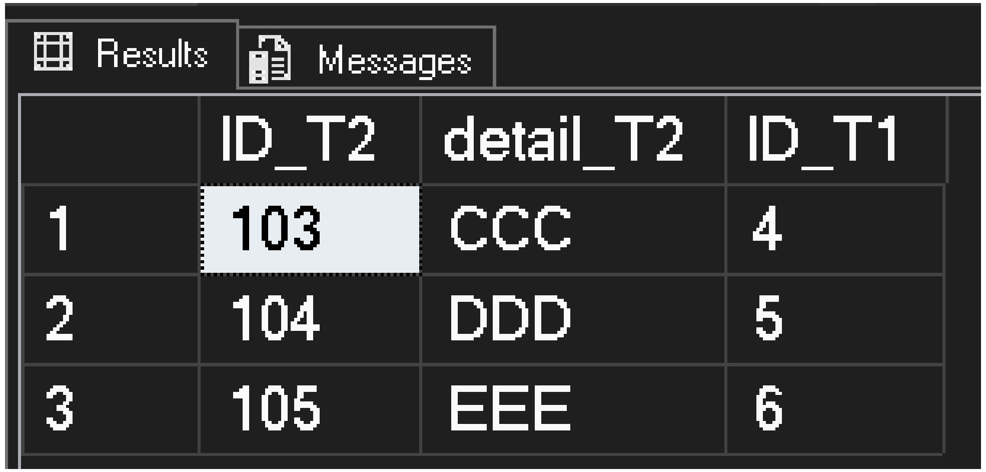
## Right Anti Join (only T2)





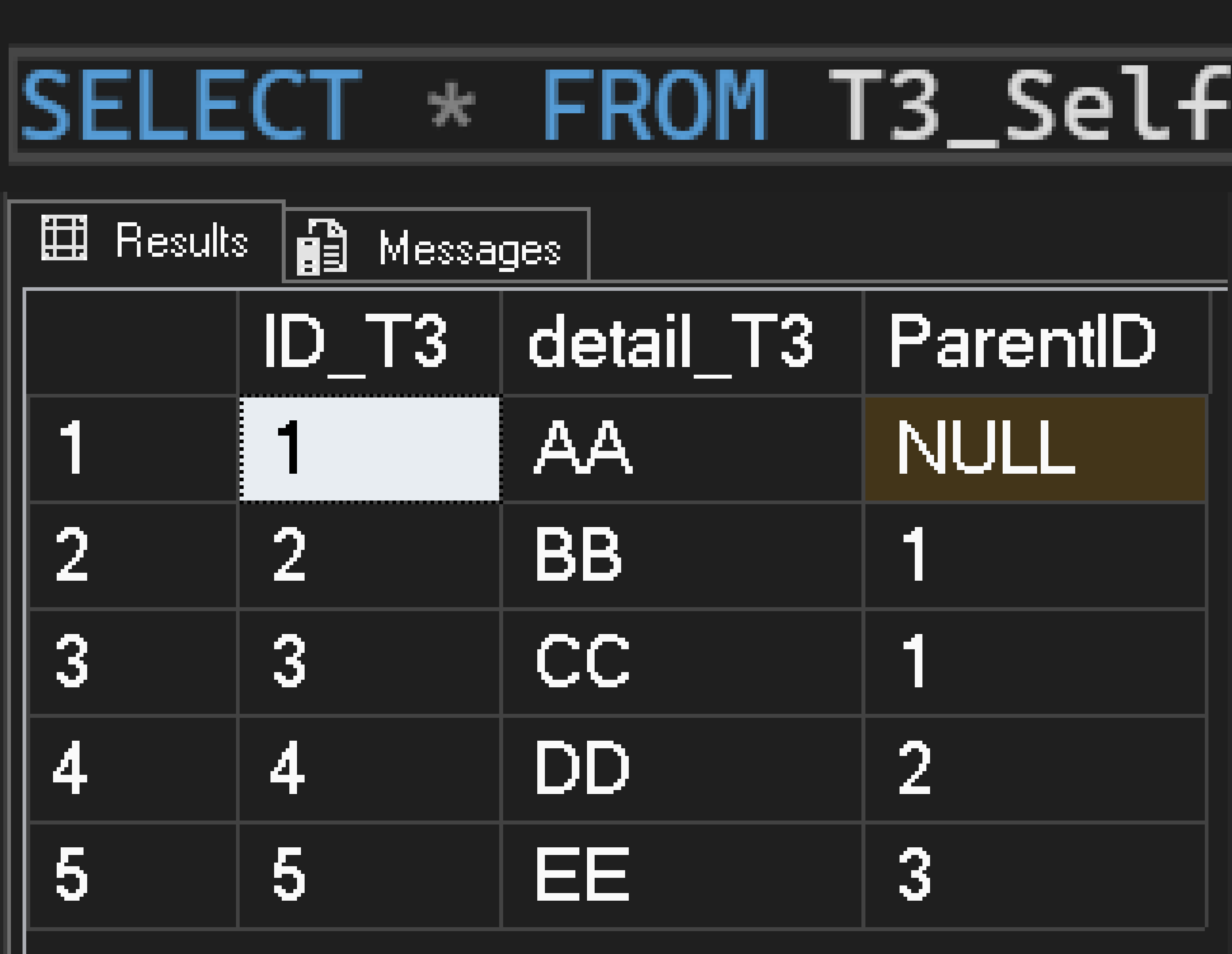


```
--Method--1
SELECT T2.*

FROM T1

RIGHT JOIN T2
ON T1.ID_T1 = T2.ID_T1

WHERE T1.ID_T1 IS NULL;


--Method--2

SELECT * FROM T2
WHERE ID_T1 NOT IN (SELECT ID_T1 FROM T1)
```



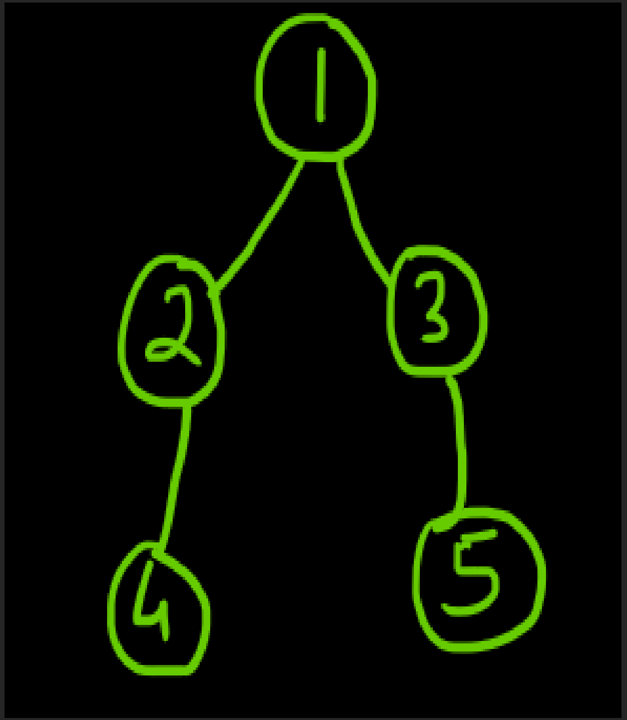
|  | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|
| 1 | 103 | CCC | 4 |
| 2 | 104 | DDD | 5 |
| 3 | 105 | EEE | 6 |

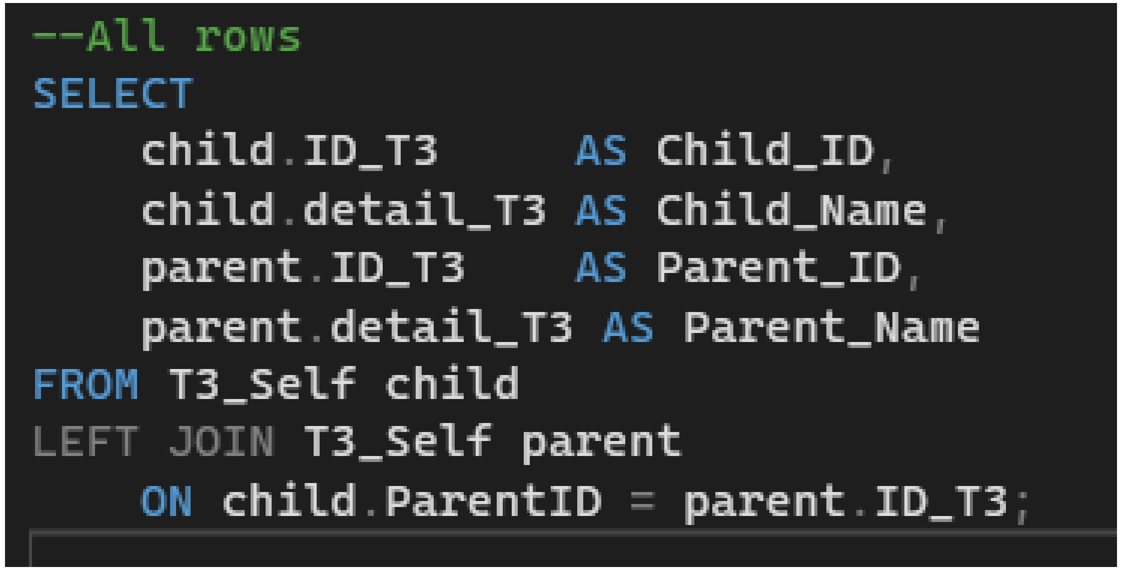
# SELF JOIN



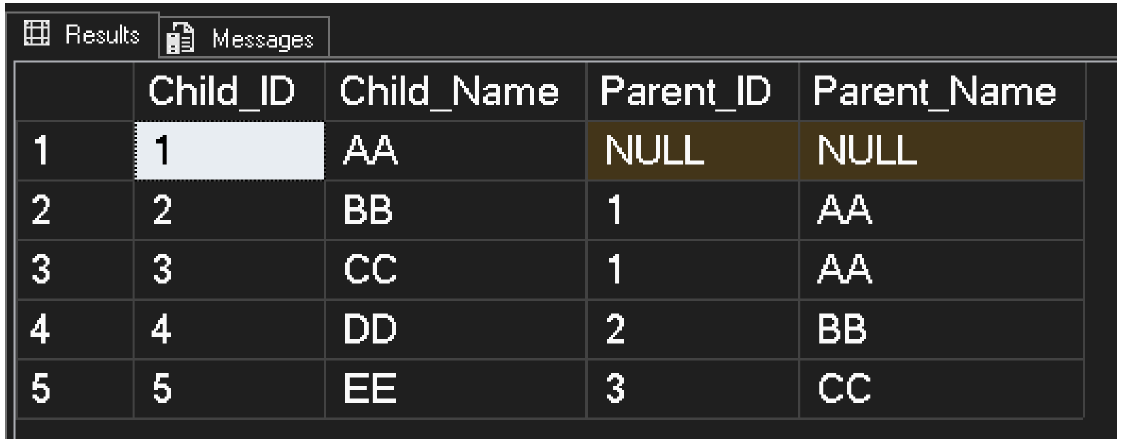
```
SELECT * FROM T3_Self;
```

| | ID_T3 | detail_T3 | ParentID |
|---|---|---|---|
| 1 | 1 | AA | NULL |
| 2 | 2 | BB | 1 |
| 3 | 3 | CC | 1 |
| 4 | 4 | DD | 2 |
| 5 | 5 | EE | 3 |

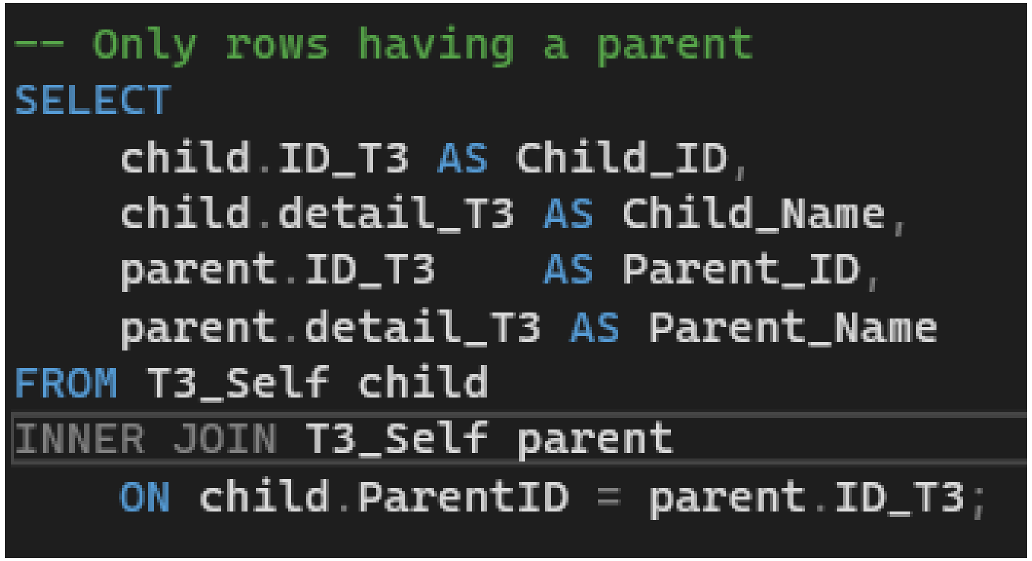




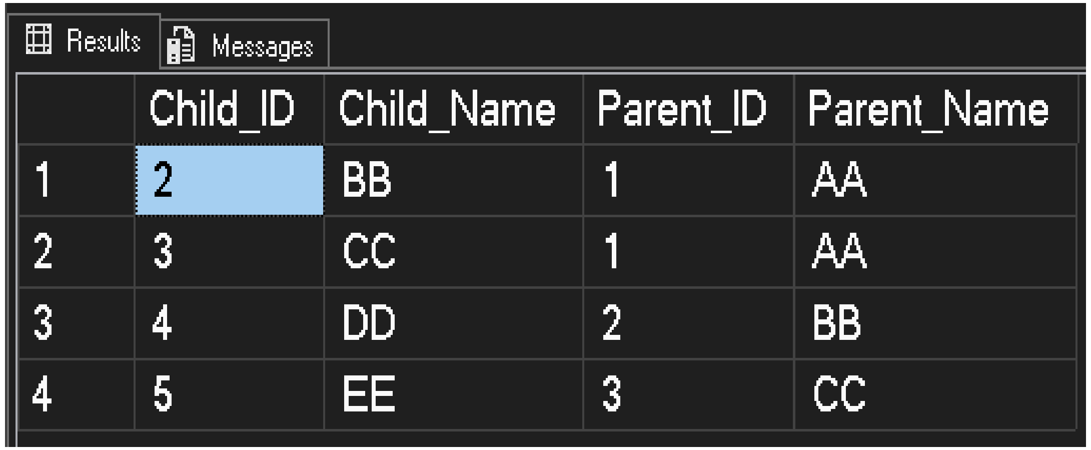
```sql
--All rows
SELECT
    child.ID_T3      AS Child_ID,
    child.detail_T3  AS Child_Name,
    parent.ID_T3     AS Parent_ID,
    parent.detail_T3 AS Parent_Name
FROM T3_Self child
LEFT JOIN T3_Self parent
    ON child.ParentID = parent.ID_T3;
```



| | Child_ID | Child_Name | Parent_ID | Parent_Name |
|---|---|---|---|---|
| 1 | 1 | AA | NULL | NULL |
| 2 | 2 | BB | 1 | AA |
| 3 | 3 | CC | 1 | AA |
| 4 | 4 | DD | 2 | BB |
| 5 | 5 | EE | 3 | CC |



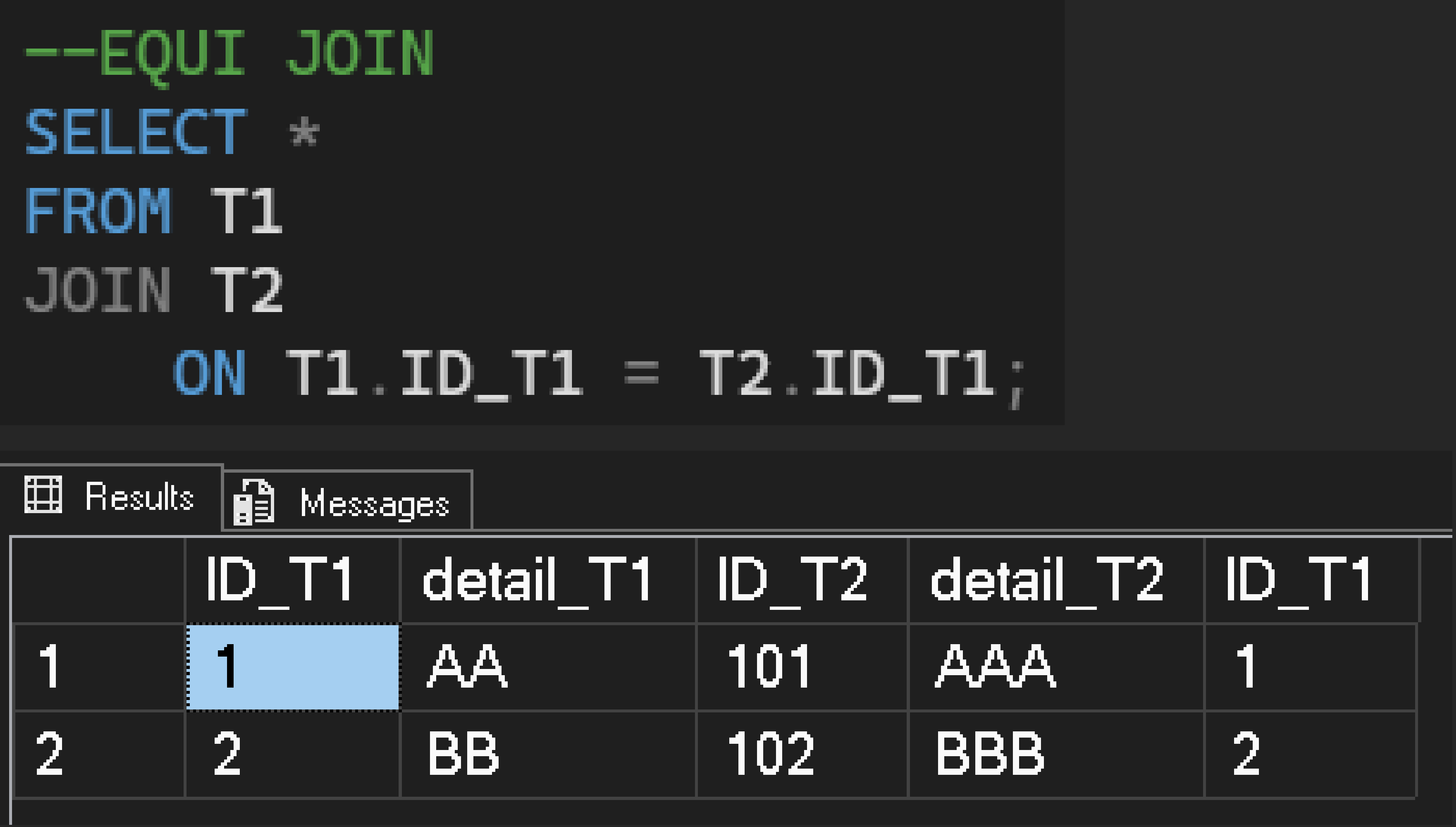
```sql
-- Only rows having a parent
SELECT
    child.ID_T3 AS Child_ID,
    child.detail_T3 AS Child_Name,
    parent.ID_T3     AS Parent_ID,
    parent.detail_T3 AS Parent_Name
FROM T3_Self child
INNER JOIN T3_Self parent
    ON child.ParentID = parent.ID_T3;
```



|  | Child_ID | Child_Name | Parent_ID | Parent_Name |
|---|---|---|---|---|
| 1 | 2 | BB | 1 | AA |
| 2 | 3 | CC | 1 | AA |
| 3 | 4 | DD | 2 | BB |
| 4 | 5 | EE | 3 | CC |

# EQUI JOIN

Join with equality condition between common columns.

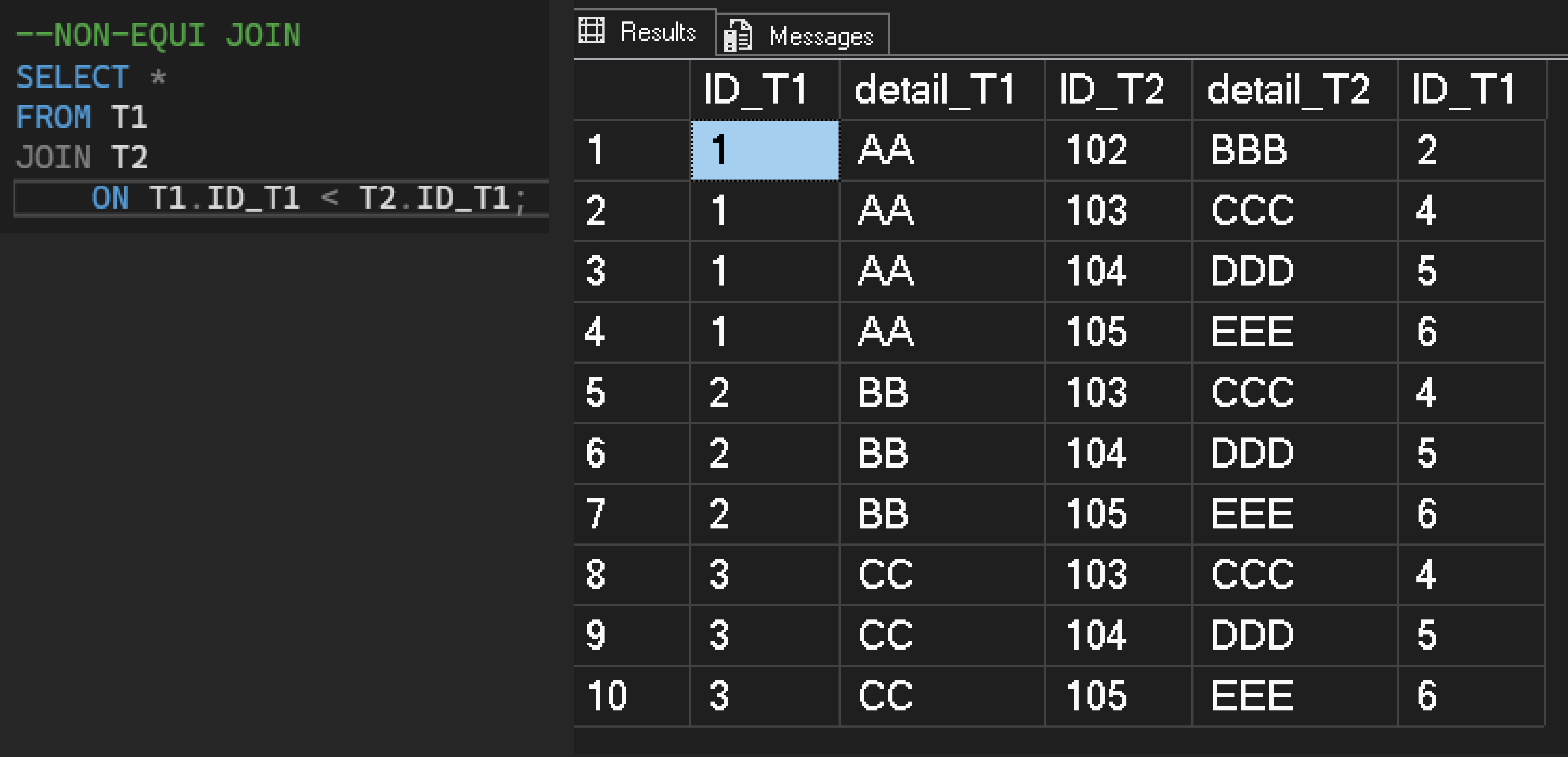


```sql
--EQUI JOIN
SELECT *
FROM T1
JOIN T2
    ON T1.ID_T1 = T2.ID_T1;
```

|   | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 2 | BB | 102 | BBB | 2 |

# NON-EQUI JOIN

Join using condition **other than "="** (like <, >, <=, etc.)



```sql
--NON-EQUI JOIN
SELECT *
FROM T1
JOIN T2
    ON T1.ID_T1 < T2.ID_T1;
```

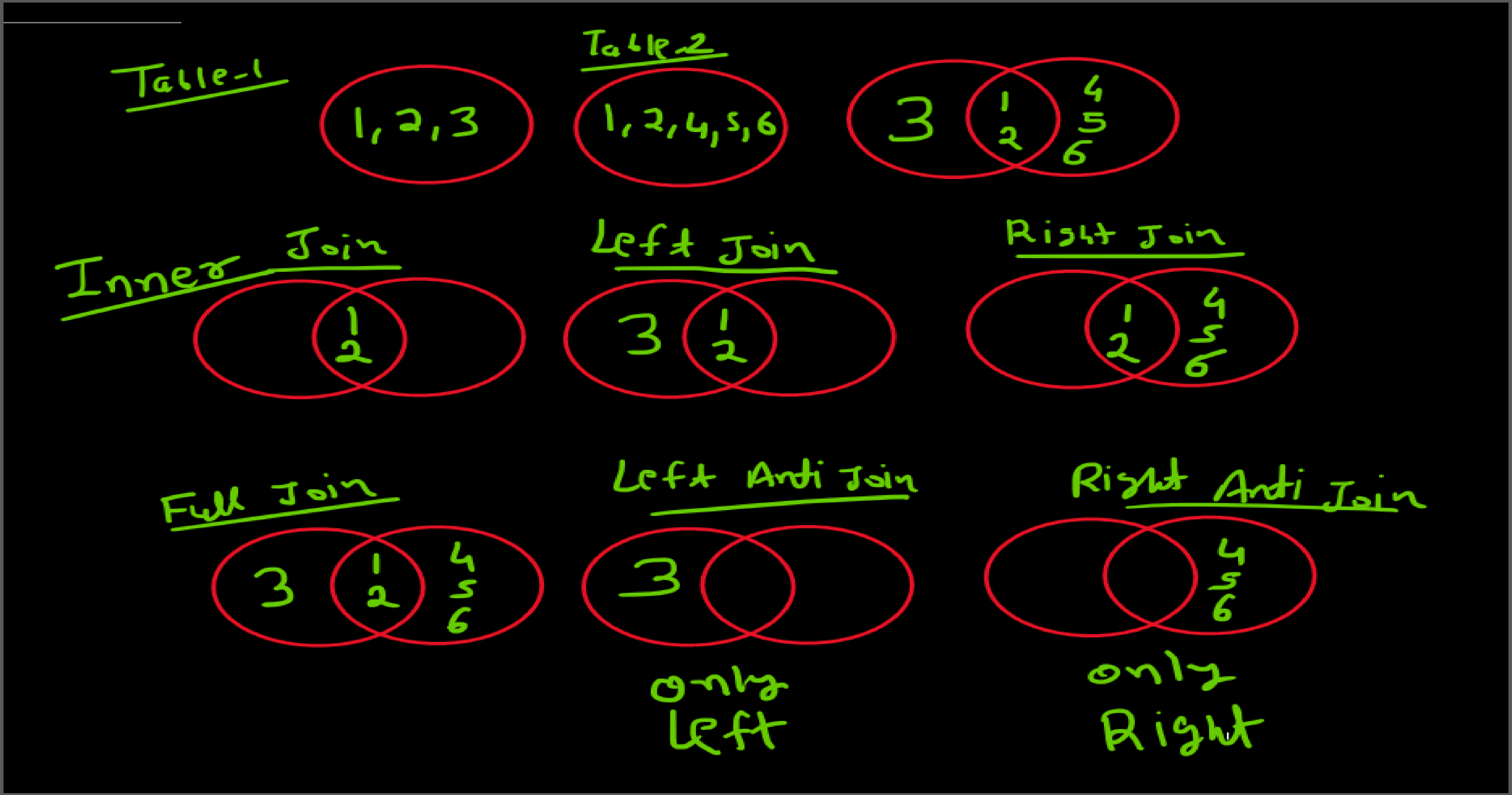
|    | ID_T1 | detail_T1 | ID_T2 | detail_T2 | ID_T1 |
|----|-------|-----------|-------|-----------|-------|
| 1  | 1     | AA        | 102   | BBB       | 2     |
| 2  | 1     | AA        | 103   | CCC       | 4     |
| 3  | 1     | AA        | 104   | DDD       | 5     |
| 4  | 1     | AA        | 105   | EEE       | 6     |
| 5  | 2     | BB        | 103   | CCC       | 4     |
| 6  | 2     | BB        | 104   | DDD       | 5     |
| 7  | 2     | BB        | 105   | EEE       | 6     |
| 8  | 3     | CC        | 103   | CCC       | 4     |
| 9  | 3     | CC        | 104   | DDD       | 5     |
| 10 | 3     | CC        | 105   | EEE       | 6     |

# NATURAL JOIN

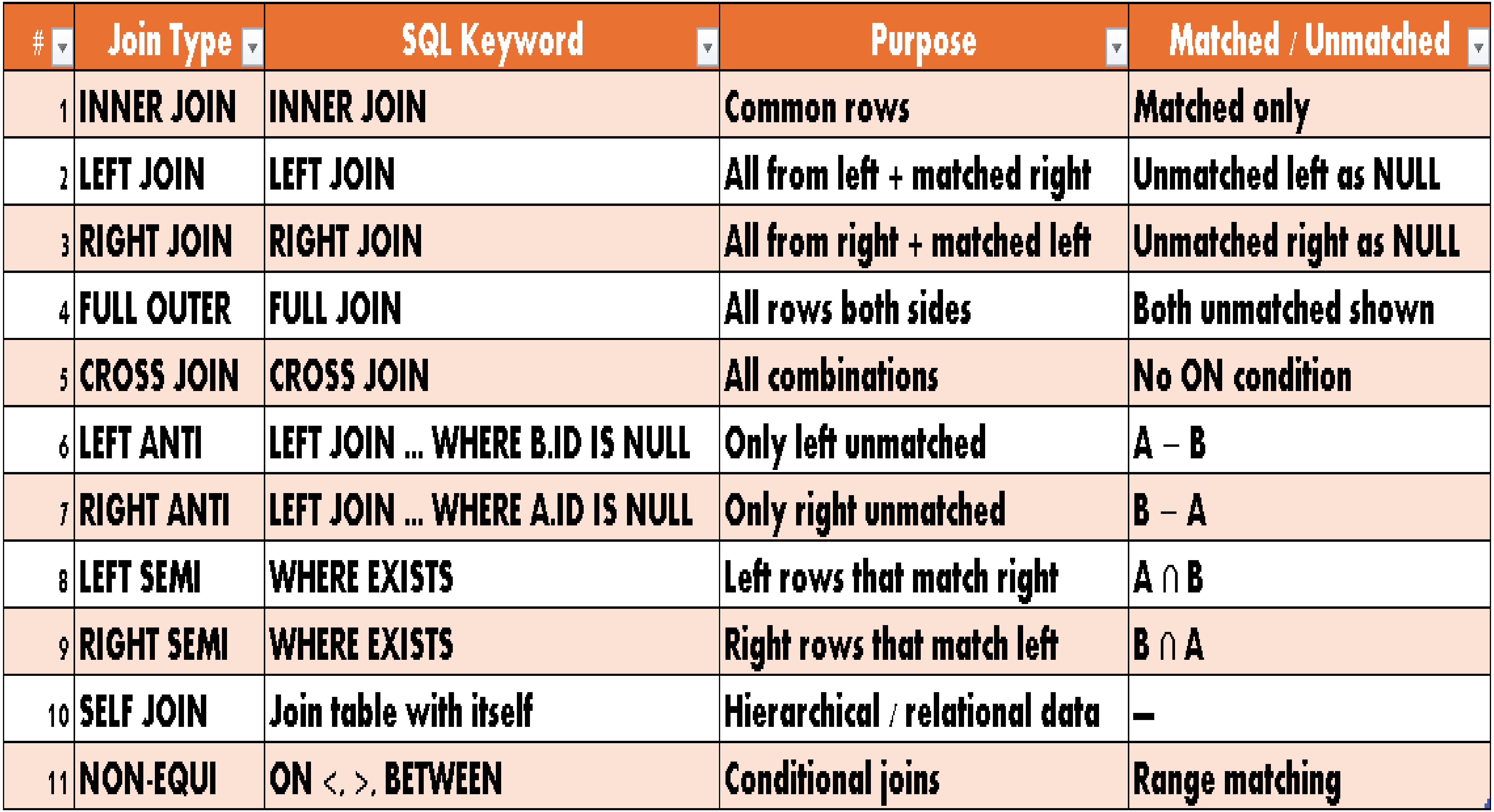
SQL Server ❌ doesn’t have NATURAL JOIN keyword,

but logically it’s an **EQUI JOIN on same column names.**

# Summary

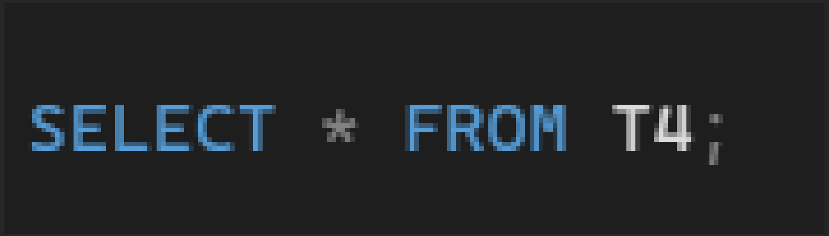


Talle-1
1, 2, 3
Talle-2
1, 2, 4, 5, 6
3
1
2
4
5
6
Inner Join
1
2
Left Join
3
1
2
Right Join
1
2
4
5
6
Full Join
3
1
2
4
5
6
Left Anti Join
3
only
left
Right Anti Join
4
5
6
only
Right

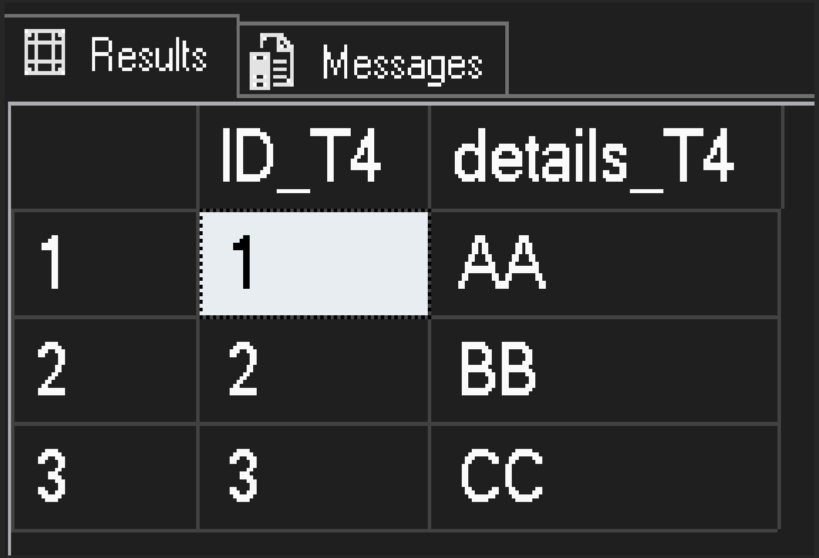


| # | Join Type | SQL Keyword | Purpose | Matched / Unmatched |
|---|---|---|---|---|
| 1 | INNER JOIN | INNER JOIN | Common rows | Matched only |
| 2 | LEFT JOIN | LEFT JOIN | All from left + matched right | Unmatched left as NULL |
| 3 | RIGHT JOIN | RIGHT JOIN | All from right + matched left | Unmatched right as NULL |
| 4 | FULL OUTER | FULL JOIN | All rows both sides | Both unmatched shown |
| 5 | CROSS JOIN | CROSS JOIN | All combinations | No ON condition |
| 6 | LEFT ANTI | LEFT JOIN ... WHERE B.ID IS NULL | Only left unmatched | A – B |
| 7 | RIGHT ANTI | LEFT JOIN ... WHERE A.ID IS NULL | Only right unmatched | B – A |
| 8 | LEFT SEMI | WHERE EXISTS | Left rows that match right | A ∩ B |
| 9 | RIGHT SEMI | WHERE EXISTS | Right rows that match left | B ∩ A |
| 10 | SELF JOIN | Join table with itself | Hierarchical / relational data | – |
| 11 | NON-EQUI | ON <, >, BETWEEN | Conditional joins | Range matching |

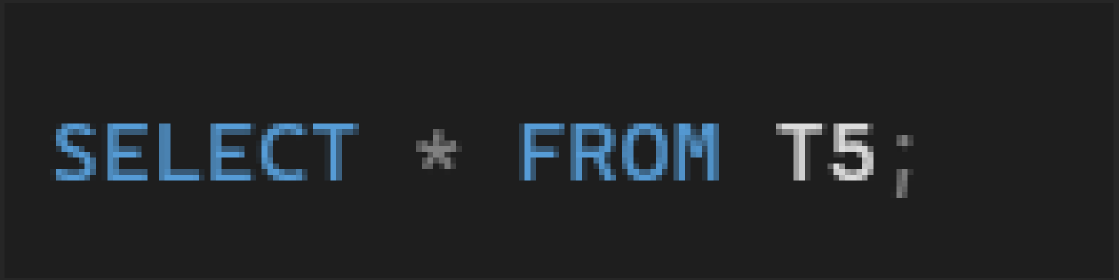
# New Table for semi join



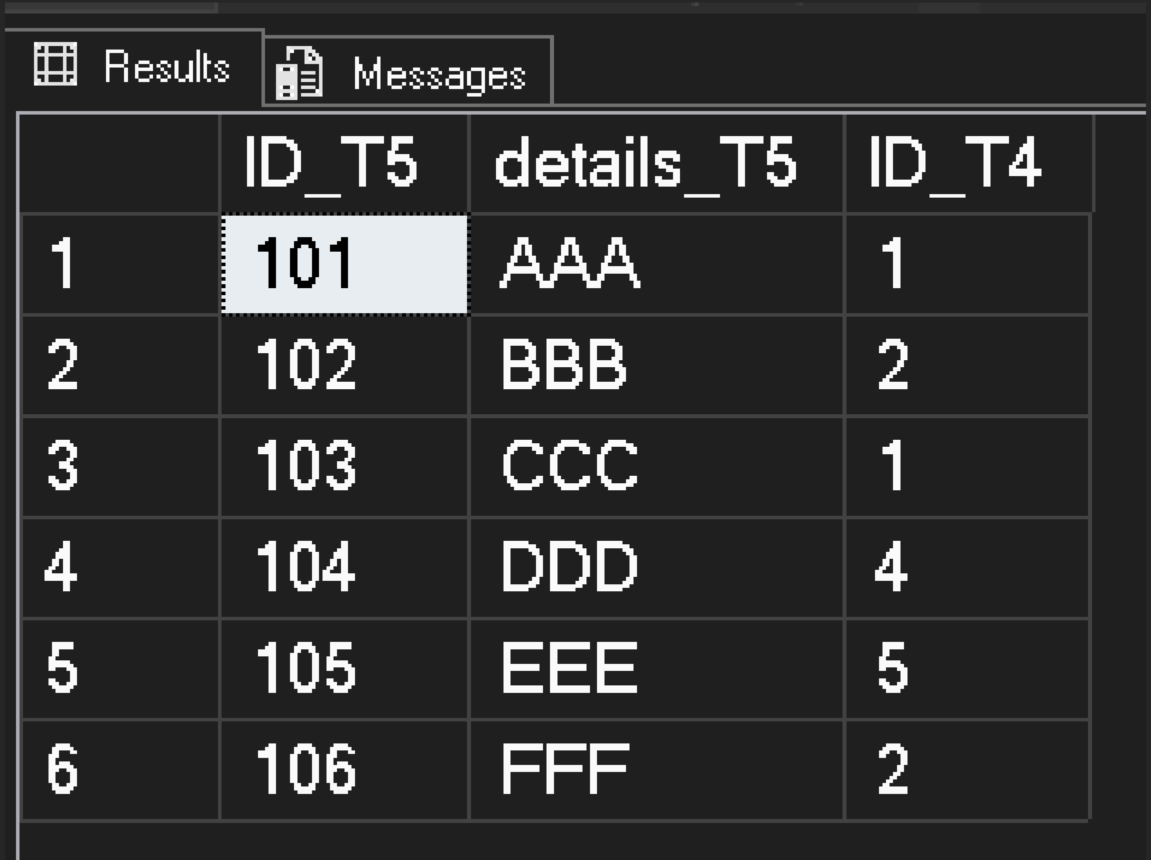
```
SELECT * FROM T4;
```



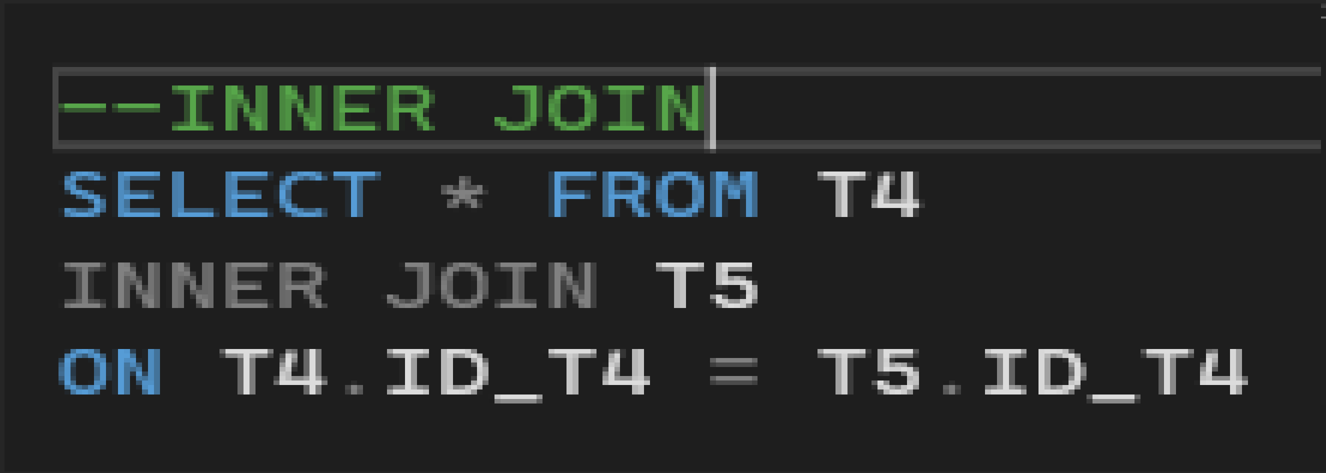
| | ID_T4 | details_T4 |
|---|---|---|
| 1 | 1 | AA |
| 2 | 2 | BB |
| 3 | 3 | CC |



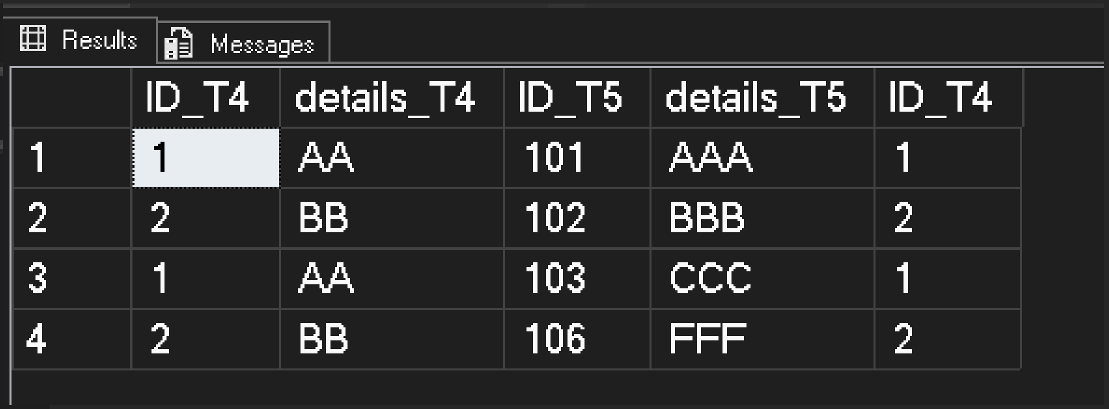
```
SELECT * FROM T5;
```



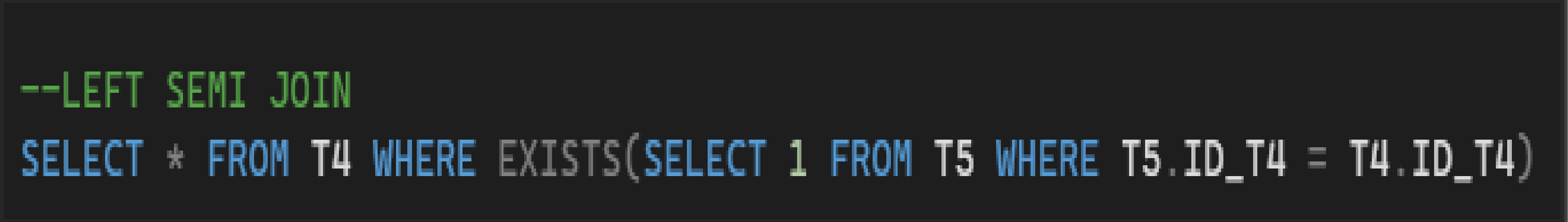
| | ID_T5 | details_T5 | ID_T4 |
|---|---|---|---|
| 1 | 101 | AAA | 1 |
| 2 | 102 | BBB | 2 |
| 3 | 103 | CCC | 1 |
| 4 | 104 | DDD | 4 |
| 5 | 105 | EEE | 5 |
| 6 | 106 | FFF | 2 |



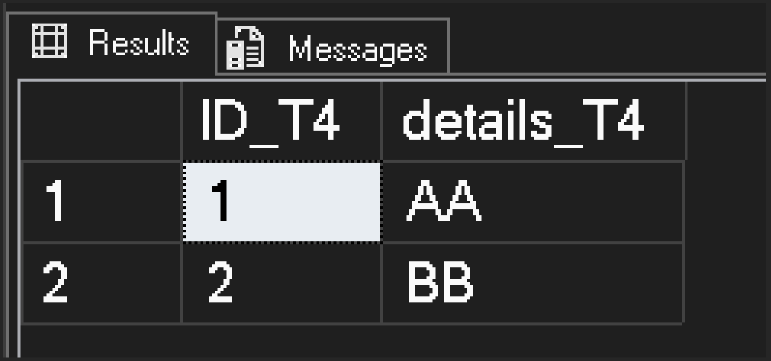
```
--INNER JOIN
SELECT * FROM T4
INNER JOIN T5
ON T4.ID_T4 = T5.ID_T4
```



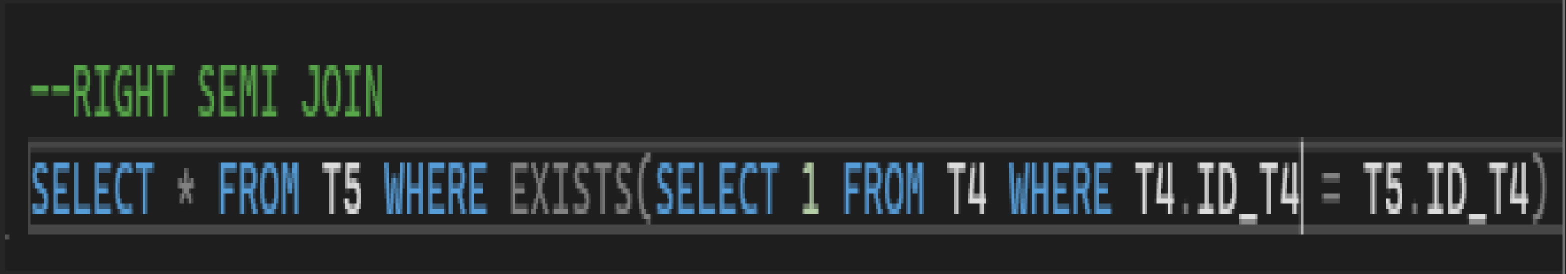
| | ID_T4 | details_T4 | ID_T5 | details_T5 | ID_T4 |
|---|---|---|---|---|---|
| 1 | 1 | AA | 101 | AAA | 1 |
| 2 | 2 | BB | 102 | BBB | 2 |
| 3 | 1 | AA | 103 | CCC | 1 |
| 4 | 2 | BB | 106 | FFF | 2 |



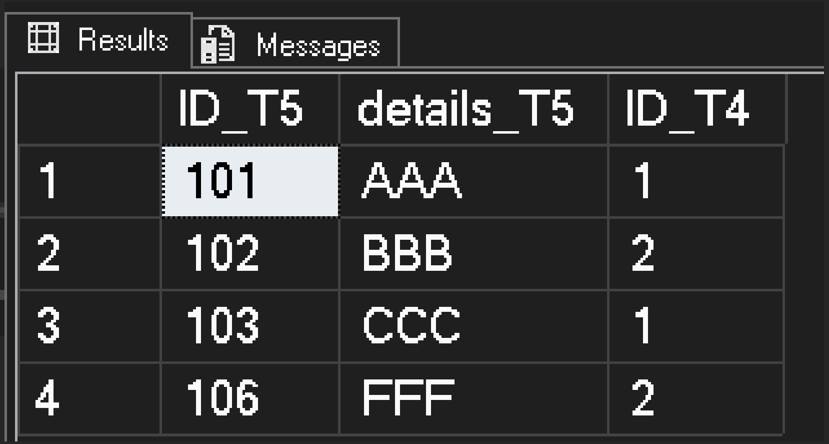
```
--LEFT SEMI JOIN
SELECT * FROM T4 WHERE EXISTS(SELECT 1 FROM T5 WHERE T5.ID_T4 = T4.ID_T4)
```



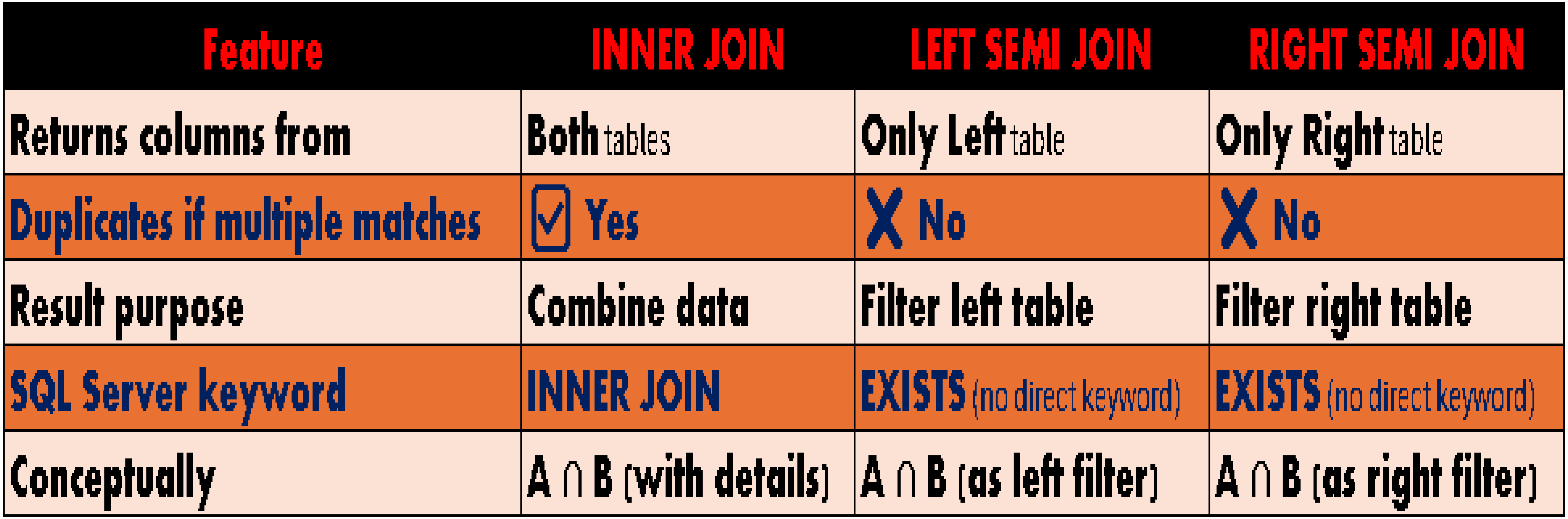
| | ID_T4 | details_T4 |
|---|---|---|
| 1 | 1 | AA |
| 2 | 2 | BB |



```
--RIGHT SEMI JOIN
SELECT * FROM T5 WHERE EXISTS(SELECT 1 FROM T4 WHERE T4.ID_T4 = T5.ID_T4)
```



| | ID_T5 | details_T5 | ID_T4 |
|---|---|---|---|
| 1 | 101 | AAA | 1 |
| 2 | 102 | BBB | 2 |
| 3 | 103 | CCC | 1 |
| 4 | 106 | FFF | 2 |



| Feature | INNER JOIN | LEFT SEMI JOIN | RIGHT SEMI JOIN |
|---|---|---|---|
| Returns columns from | **Both** tables | **Only Left** table | **Only Right** table |
| Duplicates if multiple matches | ☑ Yes | ✗ No | ✗ No |
| Result purpose | Combine data | Filter left table | Filter right table |
| SQL Server keyword | INNER JOIN | **EXISTS** (no direct keyword) | **EXISTS** (no direct keyword) |
| Conceptually | A ∩ B (with details) | A ∩ B (as left filter) | A ∩ B (as right filter) |