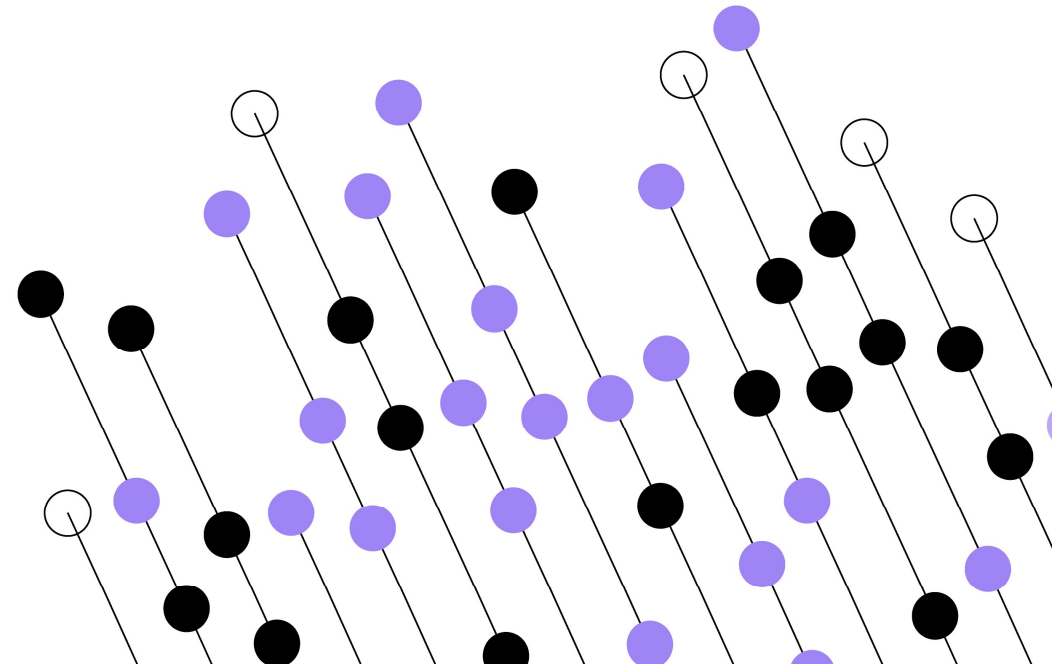


# SQL: Set Operators

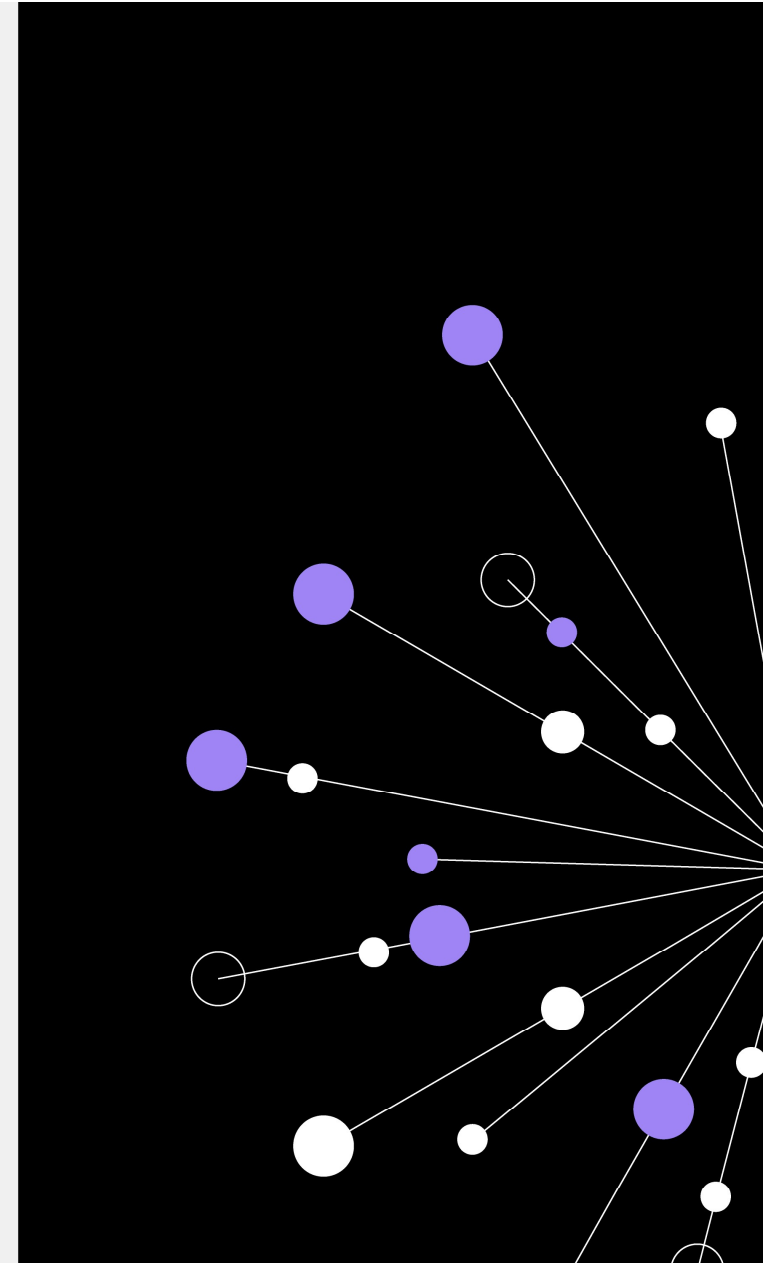


# SQL

## Contents

### Set Operators

- UNION
- UNION ALL
- INTERSECT
- EXCEPT



# Set operators

**Joins** allow us to work with multiple tables.

**Set Operators** allow us to work with multiple queries.

- **UNION**
- **UNION ALL**
- **INTERSECT**
- **EXCEPT**



# Union, union all

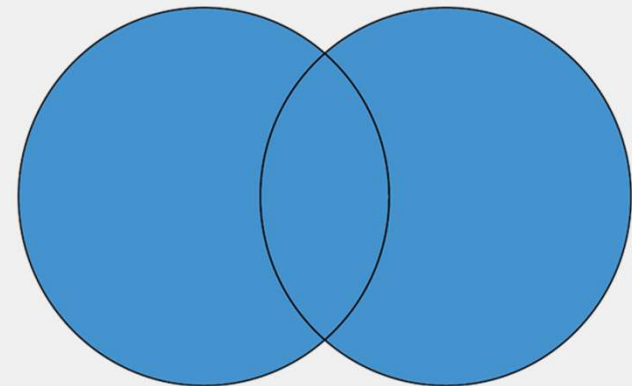
## Union

- Will return all the rows from two or more sets.
- Duplicate rows will be removed.
- The order the tables are used in does not matter.

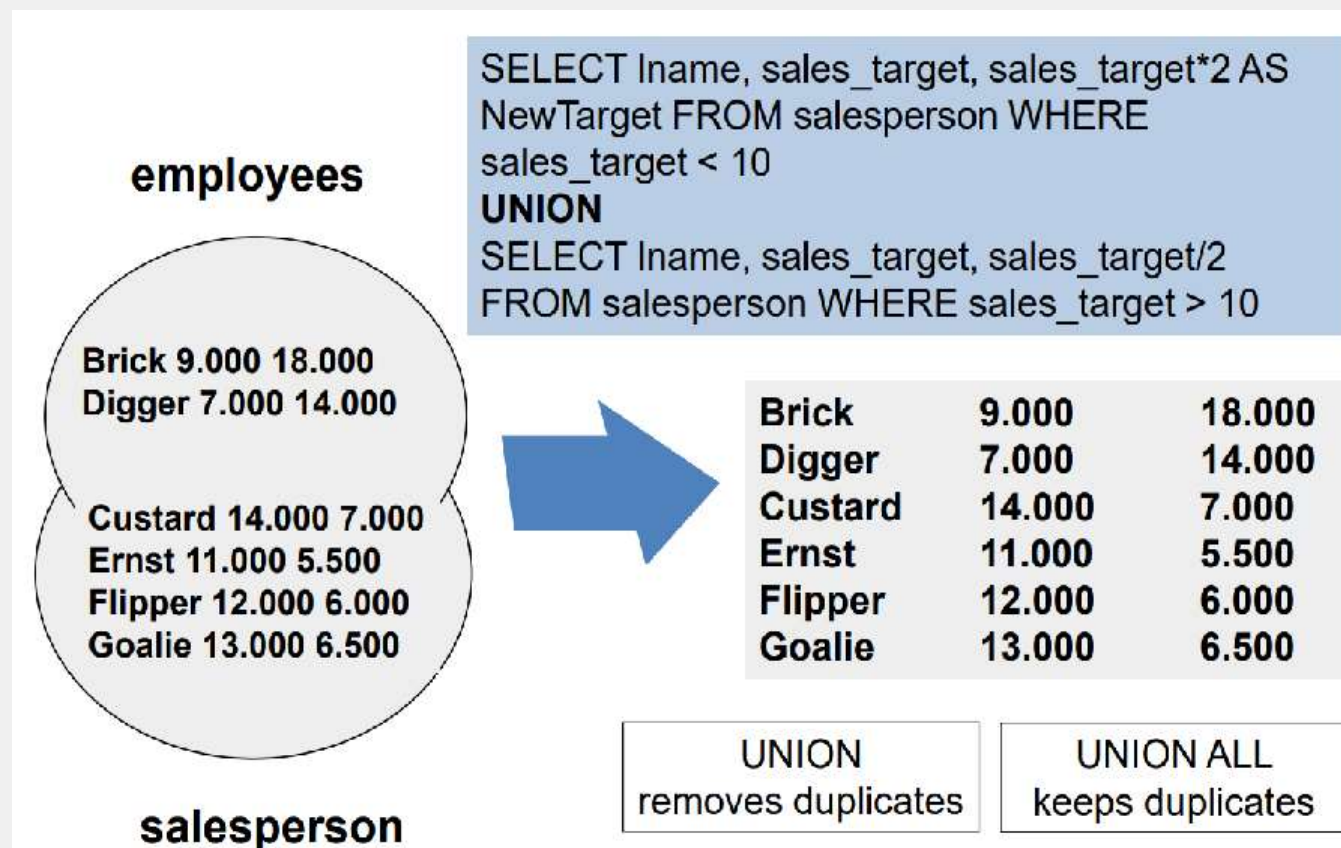
```
SELECT Columns FROM source1  
UNION [ALL]  
SELECT Columns FROM source2
```

## Union All

- Will return all the rows from two or more sets.
- Duplicate rows will not be removed.



## Union, union all example



Where are you likely to apply UNION or UNION ALL in your job?

# Intersect

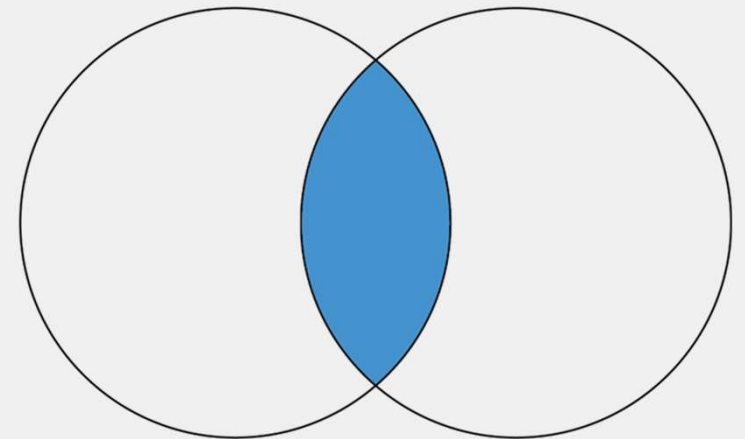
## Intersect

- Returns all rows that exist in both sets.
- The order the tables are used in does not matter.

## Notes:

- The sets included must have the same number of columns and each column must have compatible data types.
- The ORDER clause can only be used after the intersect has been performed.

```
SELECT Columns FROM source1  
INTERSECT  
SELECT Columns FROM source2
```



## Intersect example

| <u>Iname</u> | <u>sales target</u> |
|--------------|---------------------|
| Brick        | 9.0000              |
| Custard      | 14.0000             |
| Digger       | 7.0000              |
| Ernst        | 11.0000             |
| Flipper      | 12.0000             |
| Goalie       | 13.0000             |

| <u>Iname</u> | <u>sales target</u> |
|--------------|---------------------|
| Ernst        | 11.0000             |
| Shaw         | 15.0000             |
| Cassa        | 13.0000             |

```
SELECT Iname, sales_target
FROM salesperson
INTERSECT
SELECT Iname, sales_target
FROM salesperson_2
```

**salesperson**

**salesperson\_2**

| <u>Iname</u> | <u>sales target</u> |
|--------------|---------------------|
| Ernst        | 11.0000             |

## Try intersect

Using the Northwind database, display the products that are currently discontinued but have been ordered before 01/01/1997.

Where are you likely to apply INTERSECT in your job?





# Except

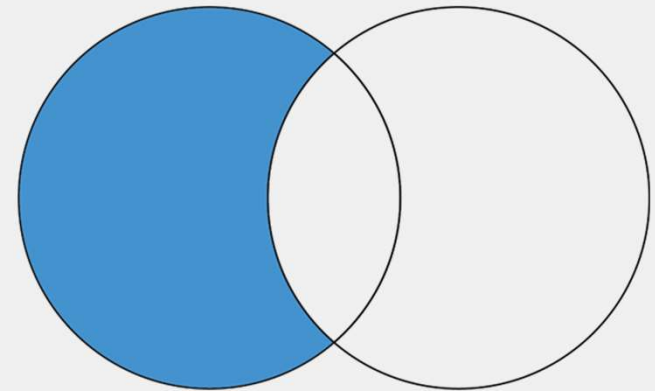
## Except

- Returns all rows that exist in set1 that do not exist in set2.
- The order the tables are used in does matter.

## Notes:

- The sets included must have the same number of columns and each column must have compatible data types.
- The ORDER clause can only be used after the intersect has been performed.

```
SELECT Columns FROM source1  
EXCEPT  
SELECT Columns FROM source2
```



## Except example

| <u>Iname</u> | <u>sales target</u> |
|--------------|---------------------|
| Brick        | 9.0000              |
| Custard      | 14.0000             |
| Digger       | 7.0000              |
| Ernst        | 11.0000             |
| Flipper      | 12.0000             |
| Goalie       | 13.0000             |

| <u>Iname</u> | <u>sales target</u> |
|--------------|---------------------|
| Ernst        | 11.0000             |
| Shaw         | 15.0000             |
| Cassa        | 13.0000             |

```
SELECT Iname, sales_target  
FROM salesperson_2  
EXCEPT  
SELECT Iname, sales_target  
FROM salesperson
```

**salesperson**

**salesperson\_2**

**Table order important!**

| <u>Iname</u> | <u>sales target</u> |
|--------------|---------------------|
| Shaw         | 15.0000             |
| Cassa        | 13.0000             |

## Try except

Using the Northwind database, display the products that are currently active (not discontinued) but have not been ordered before 01/01/1997.

Where are you likely to apply EXCEPT in your job?

