Joining Data

Mihaela Danci
Data Analyst

linkedin.com/in/mihaela-danci/

Overview

Joins syntax

Joins types

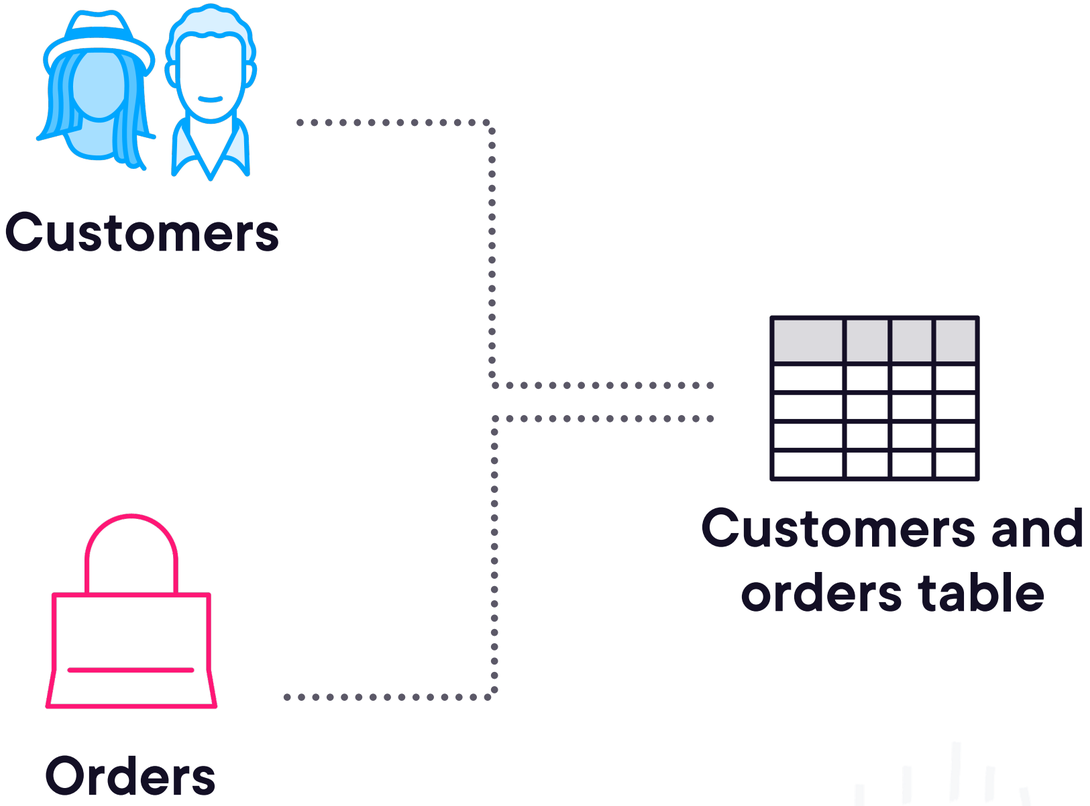
- Cross joins
- Inner joins
- Outer joins
- Self joins

Relational Databases

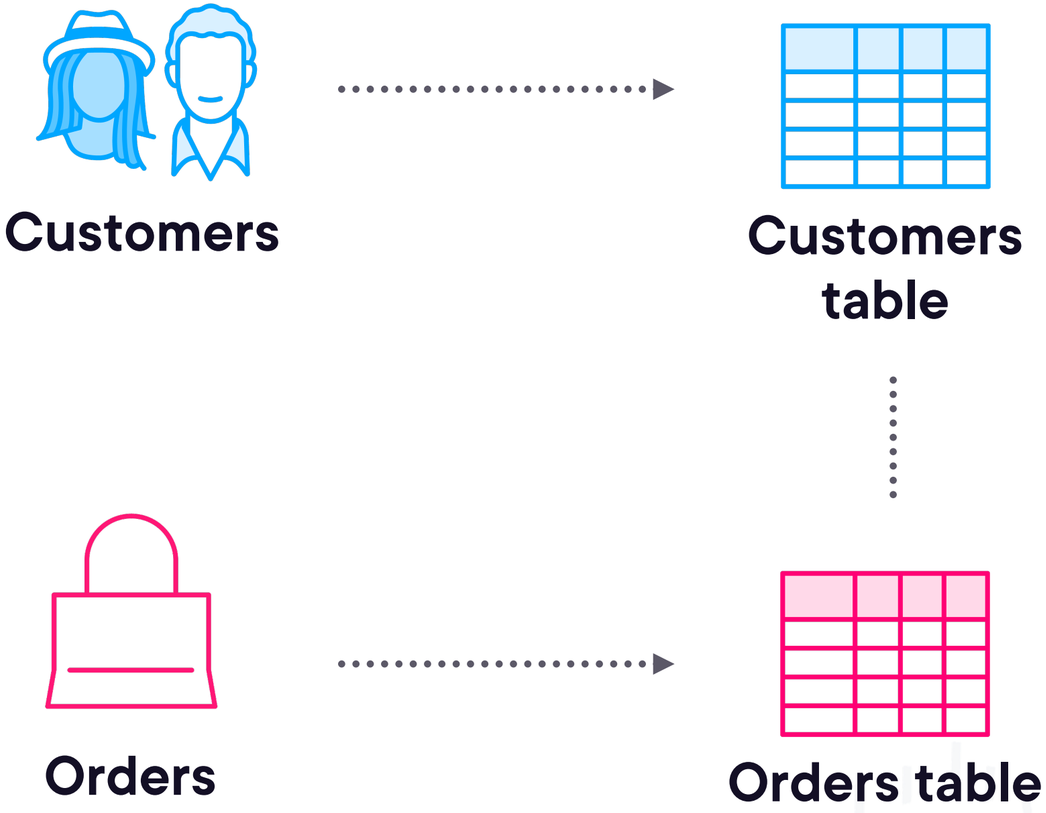
Data normalization

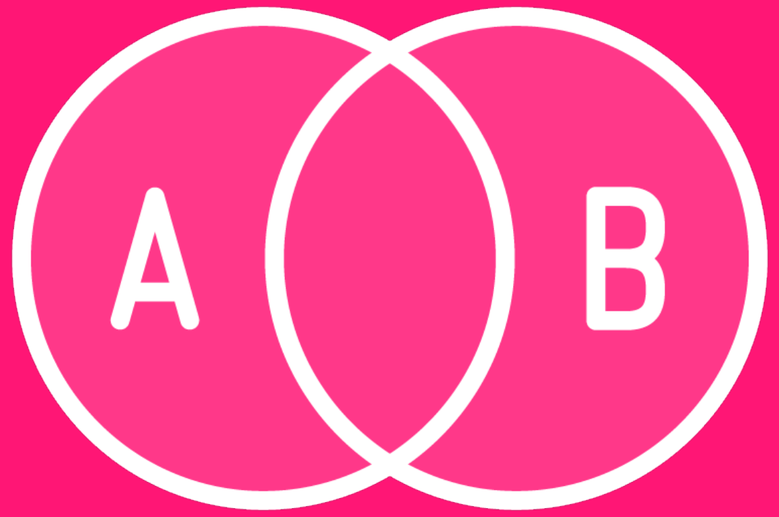
Minimize data duplication

Relational Databases



Relational Databases

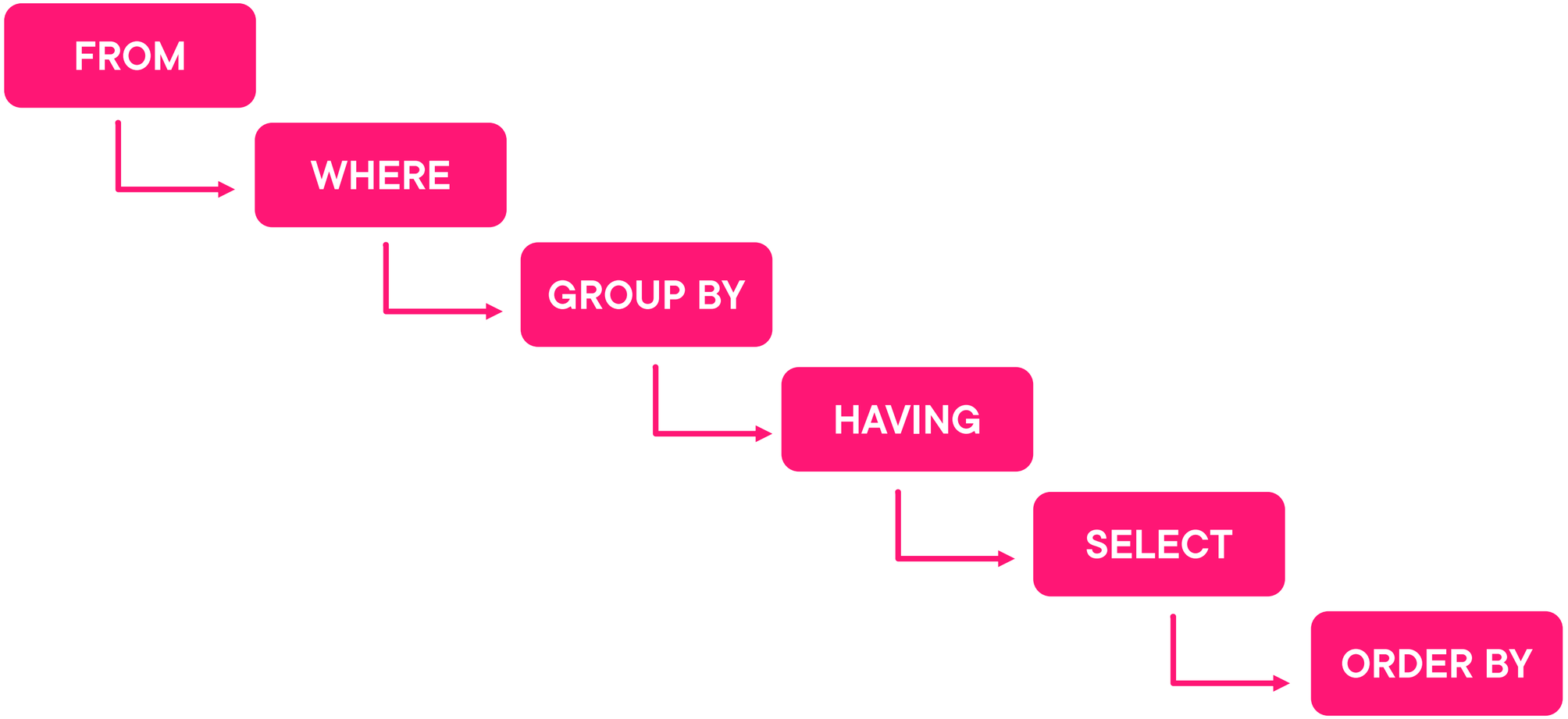




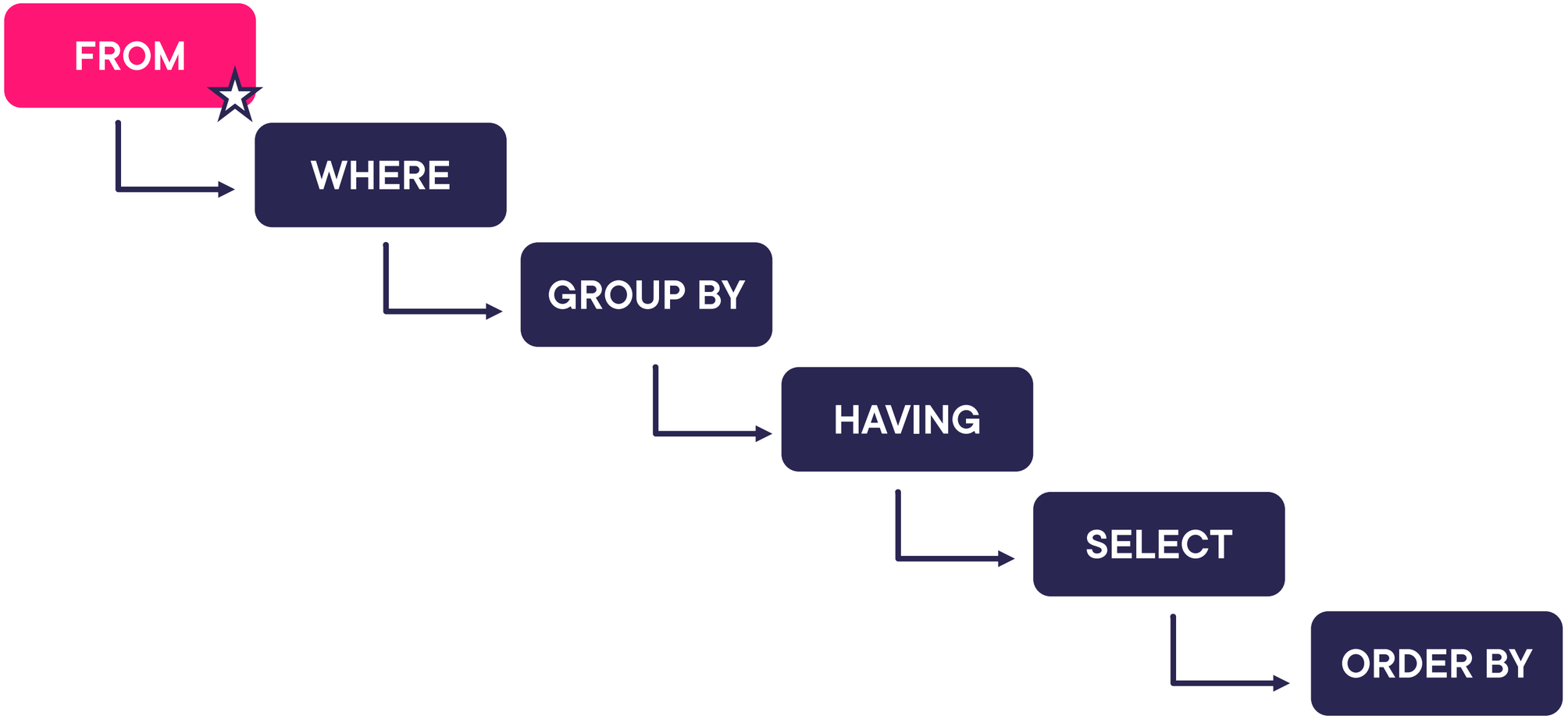
Combining data

Joins represent a method of combining data from multiple tables.

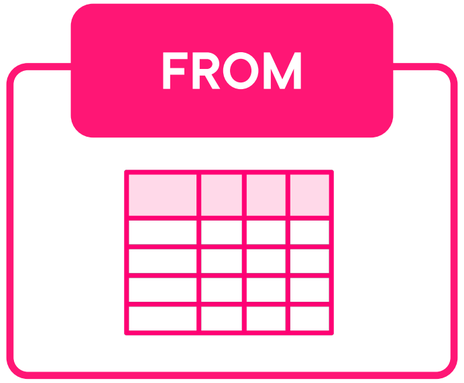
Logical Processing Order



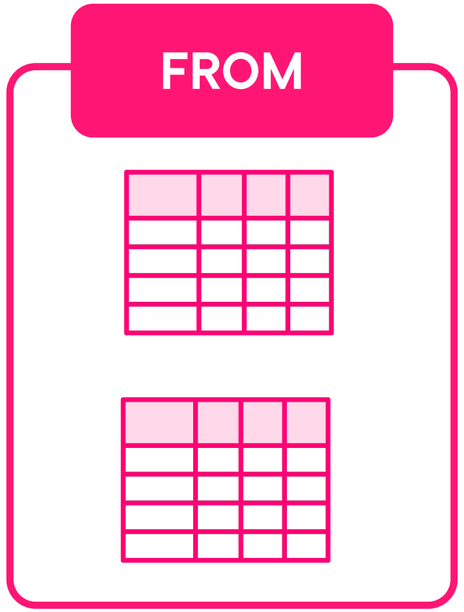
Logical Processing Order



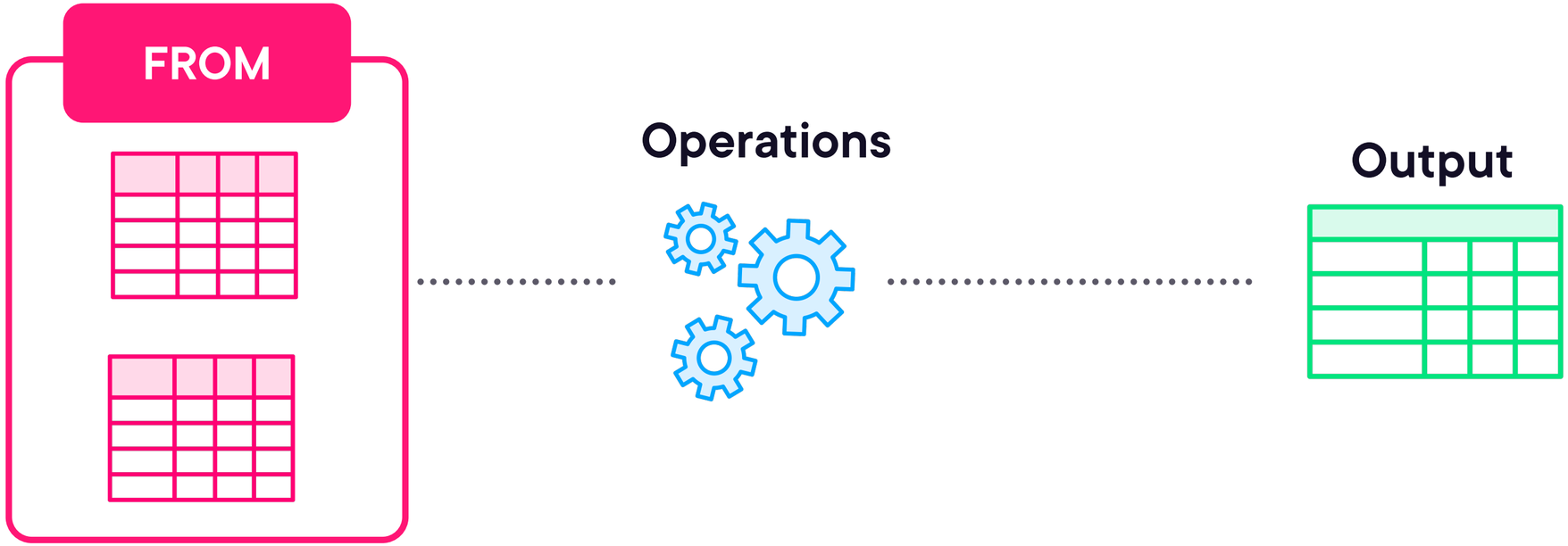
Logical Processing Order



Logical Processing Order



Logical Processing Order



The output is a set

Logical Query Processing Phases

The number of phases a join executes determines the type of join.

Cross Joins

Logical Query Processing Phases

Cross Join

Cross Join

Customer Key	Name
1	Tom
2	Mary

Order Key	Customer Key	Amount
10	1	100
20	2	200
30	1	50

Cross Join

Customer Key	Name
1	Tom
2	Mary

×

Order Key	Customer Key	Amount
10	1	100
20	2	200
30	1	50

=

Cross Join

Customer Key	Name
1	Tom
2	Mary

×

Order Key	Customer Key	Amount
10	1	100
20	2	200
30	1	50

=

Customer Key	Name	Order Key	Customer Key	Amount
1	Tom	10	1	100
1	Tom	20	2	200
1	Tom	30	1	50
2	Mary	10	1	100
2	Mary	20	2	200
2	Mary	30	1	50

Customer Key	Name
1	Tom
2	Mary

×

Order Key	Customer Key	Amount
10	1	100
20	2	200
30	1	50

=

Customer Key	Name	Order Key	Customer Key	Amount
1	Tom	10	1	100
1	Tom	20	2	200
1	Tom	30	1	50
2	Mary	10	1	100
2	Mary	20	2	200
2	Mary	30	1	50

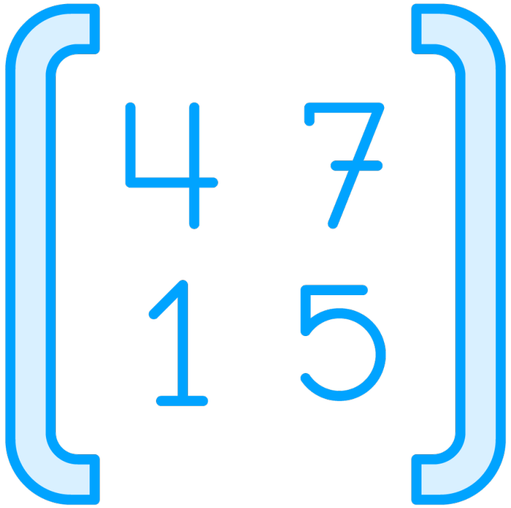
```sql
SELECT *
FROM Customers
    CROSS JOIN Orders;
```

```sql
-- standard syntax SQL-92

SELECT *
FROM Customers AS c
    CROSS JOIN Orders  AS o;
```

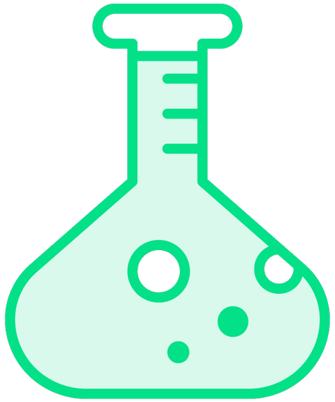
```sql
-- standard syntax SQL-89

SELECT *
FROM Customers AS c, Orders AS o;
```

When are cross joins used?

Usage



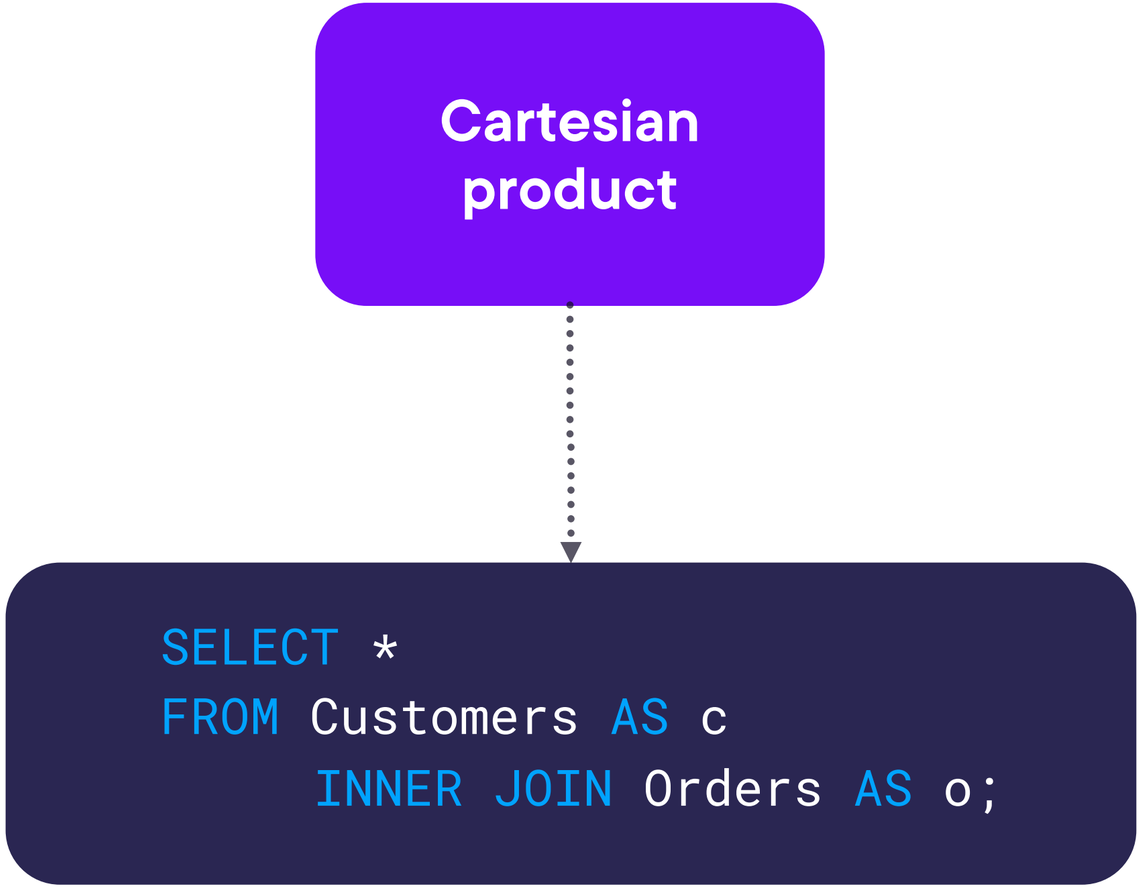
Create table of numbers



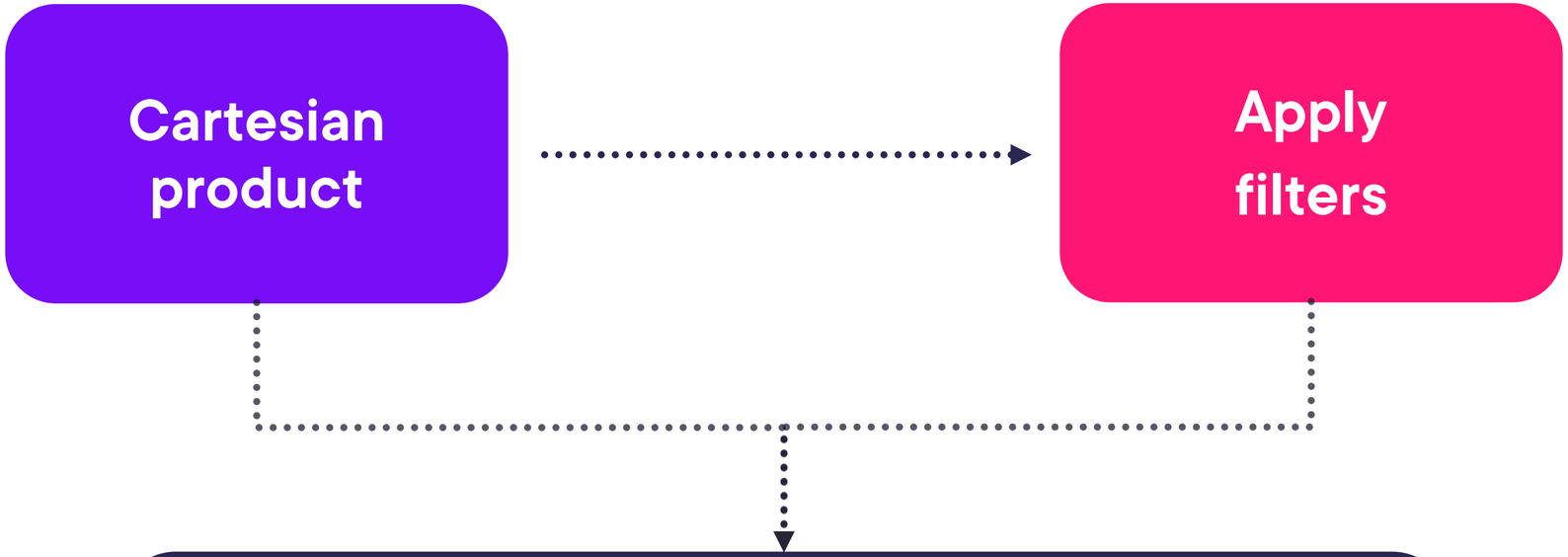
Create test data

Inner Joins

Logical Query Processing Phases

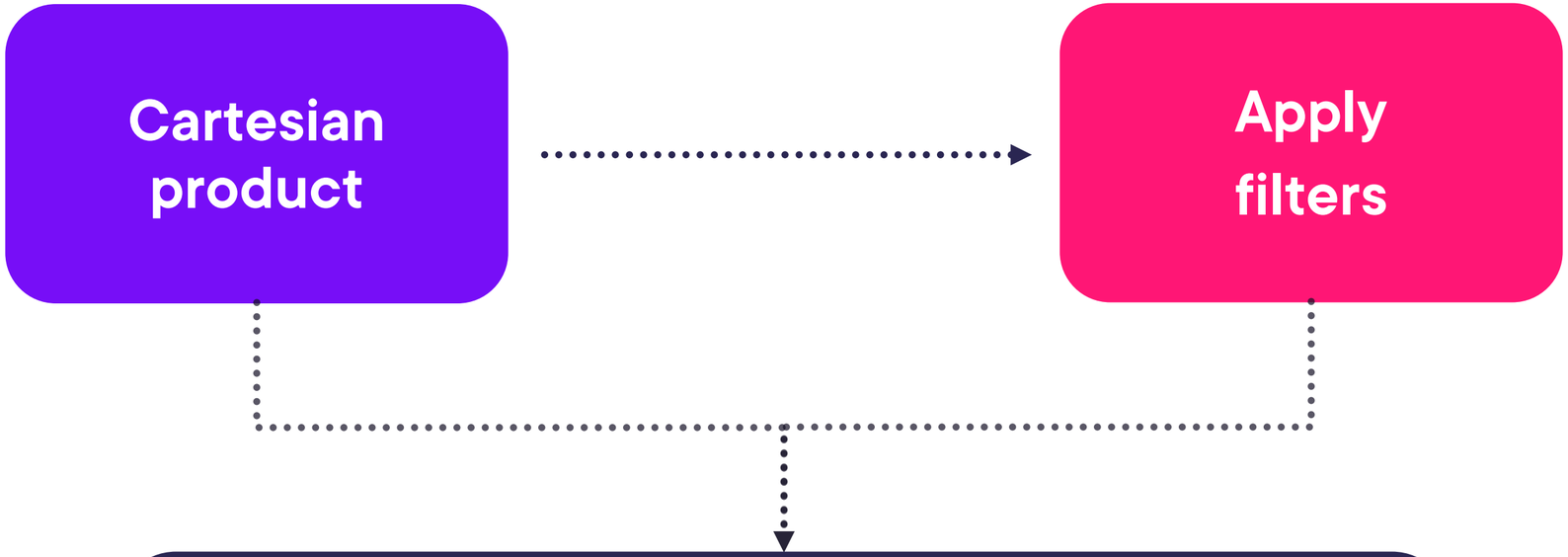


Logical Query Processing Phases




```sql
SELECT *
FROM Customers AS c
      INNER JOIN Orders  AS o
      ON c.CustomerKey = o.CustomerKey;
```

Logical Query Processing Phases




```sql
SELECT *
FROM Customers AS c
      INNER JOIN Orders  AS o
      ON c.CustomerKey = o.CustomerKey;
```

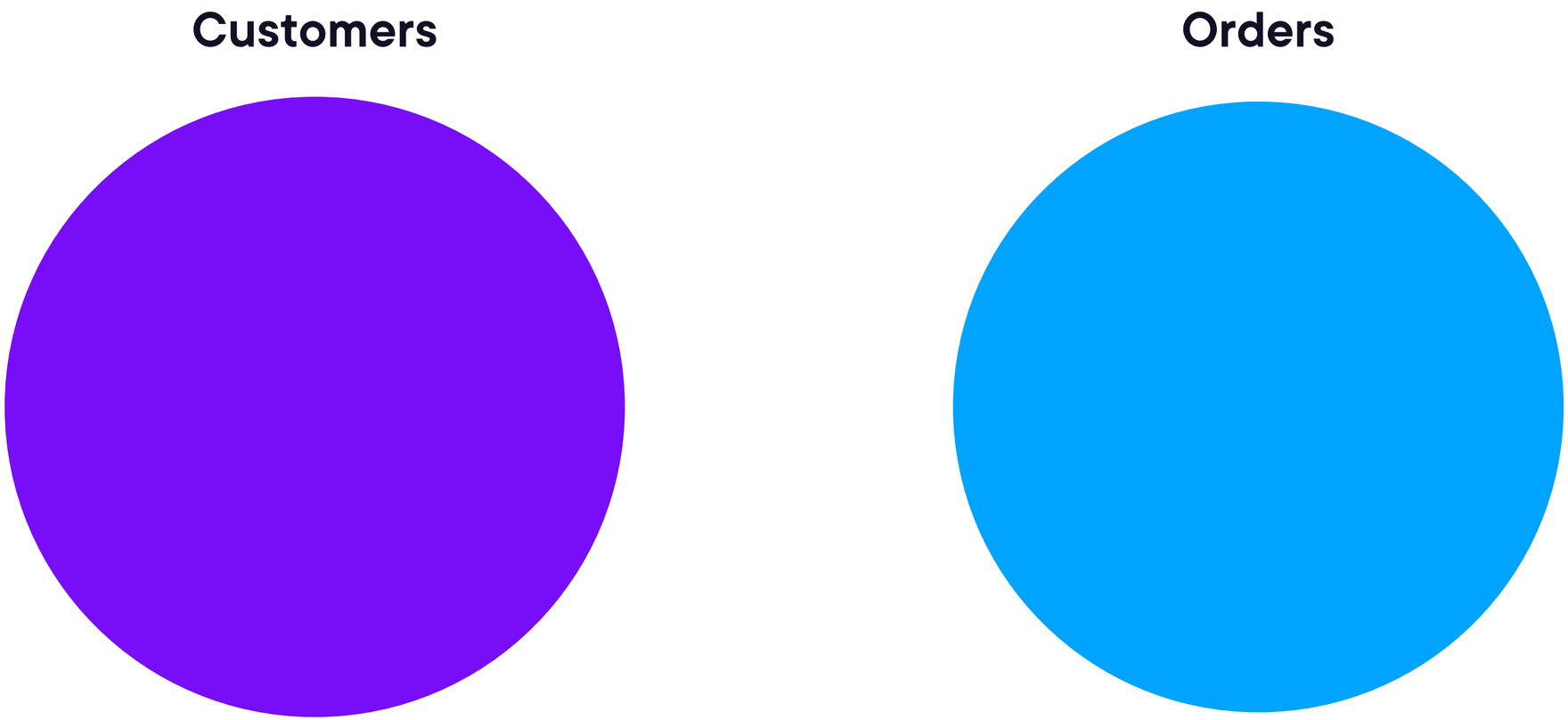
Inner Join

```sql
SELECT *
FROM Customers AS c
    INNER JOIN Orders  AS o
    ON c.CustomerKey = o.CustomerKey;
```

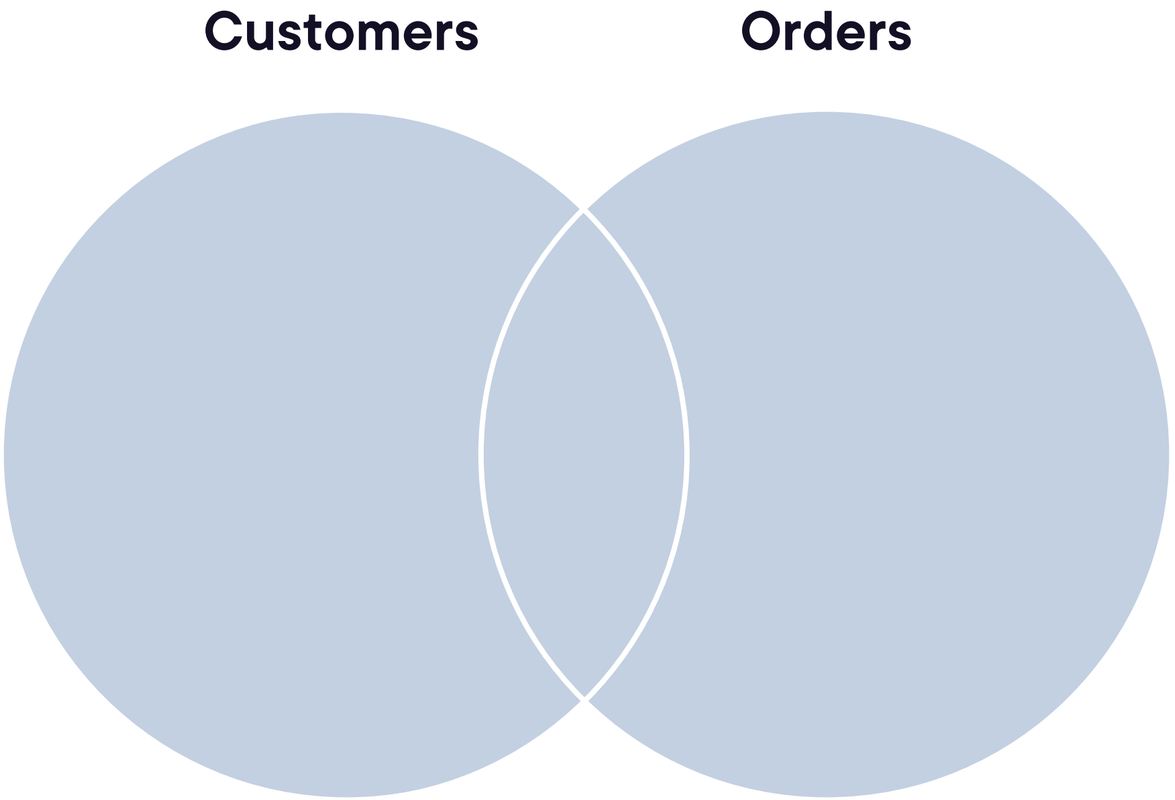
```sql
-- standard syntax SQL-92

SELECT *
FROM Customers AS c
    INNER JOIN Orders  AS o
    ON c.CustomerKey = o.CustomerKey;
```

```sql
-- standard syntax SQL-89

SELECT *
FROM Customers AS c, Orders AS o
WHERE c.CustomerKey = o.CustomerKey;
```

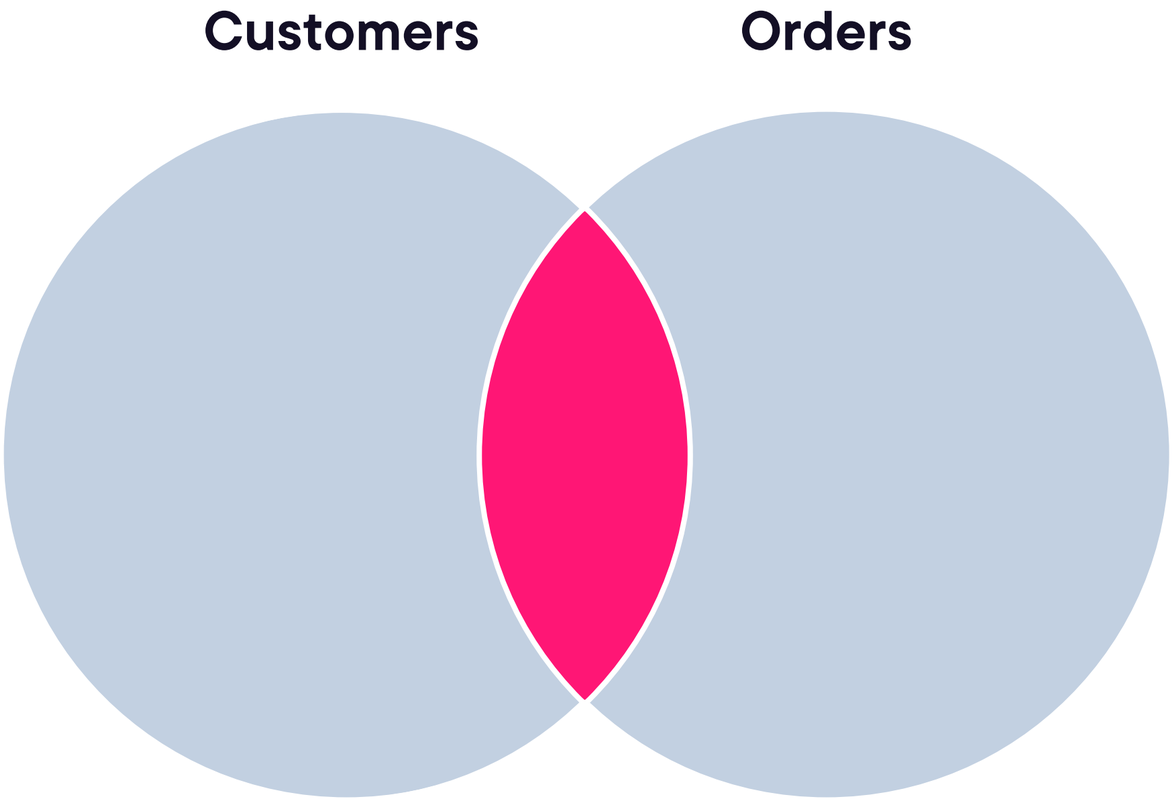
Inner Join



Inner Join



Inner Join



Inner Join

Table order

Multiple tables

Demo

Use inner joins

- Retrieve the list of customer names who placed orders

Outer Joins

Logical Query Processing Phases

Logical Query Processing Phases

Logical Query Processing Phases

Logical Query Processing Phases

The order in which we join the tables is important!

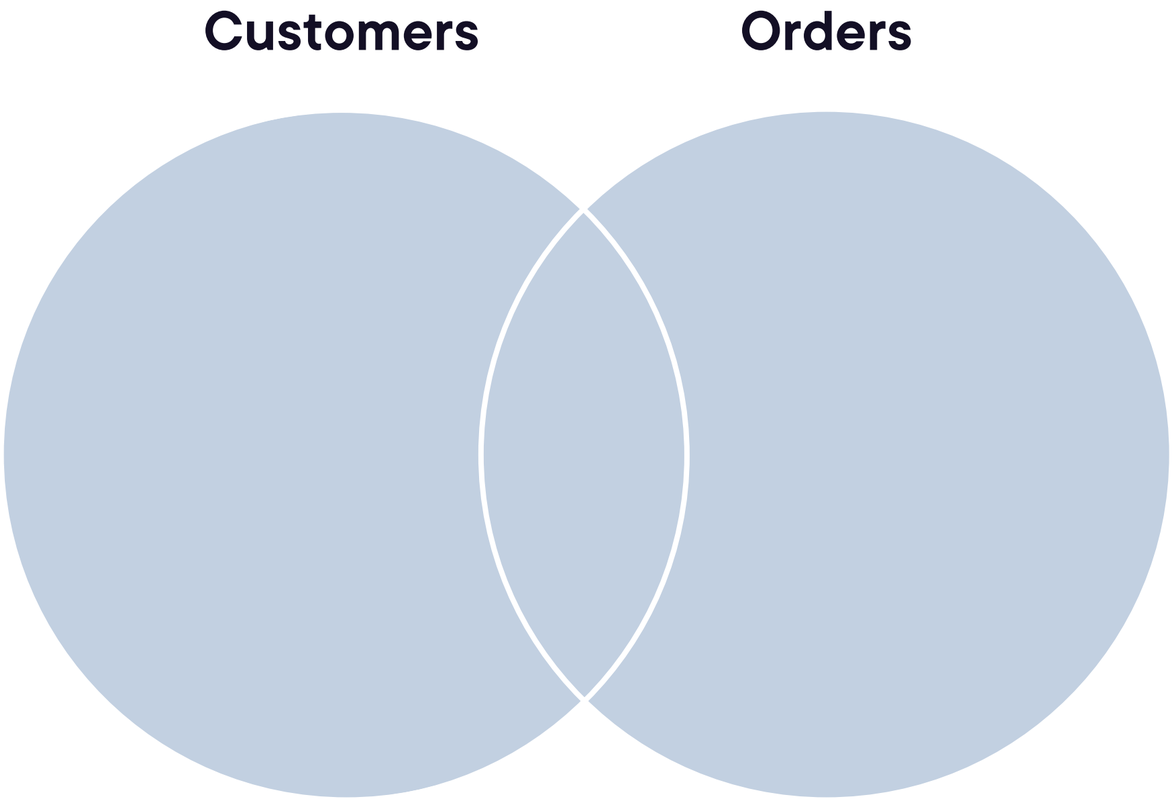
Types of Outer Join

Left outer join

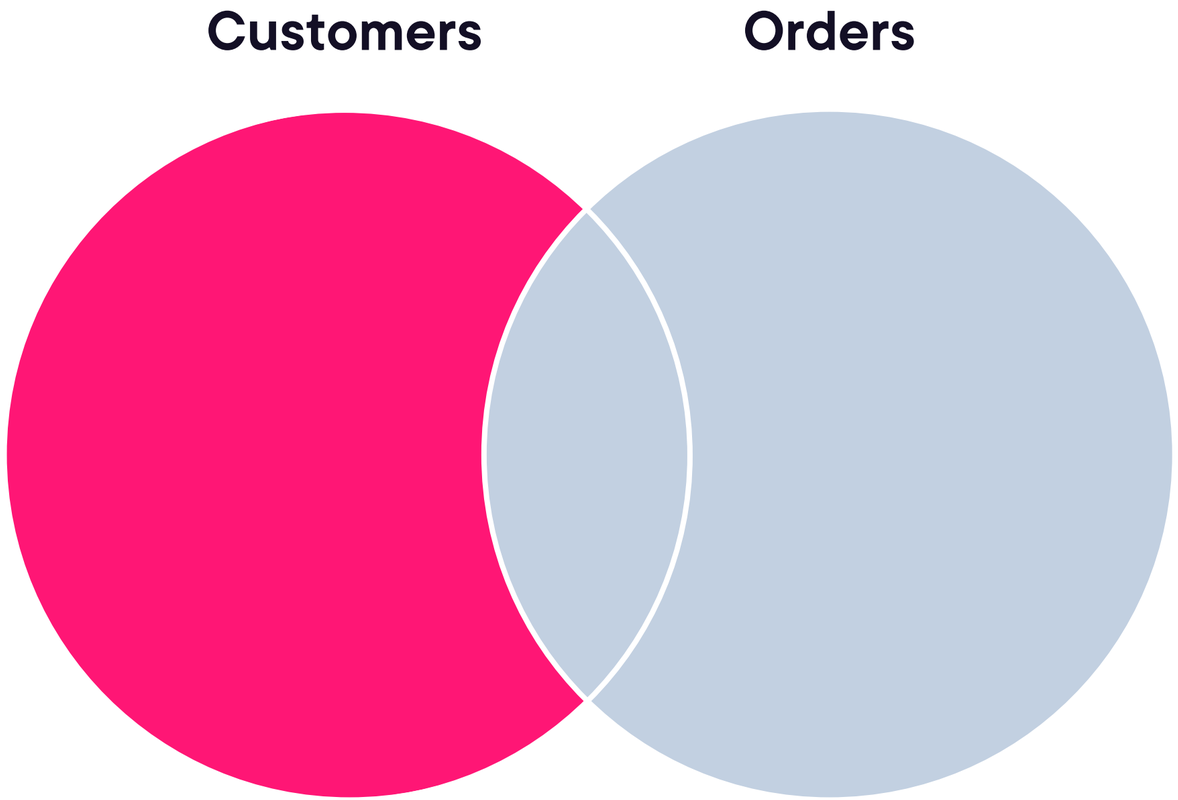
Right outer join

Full outer join

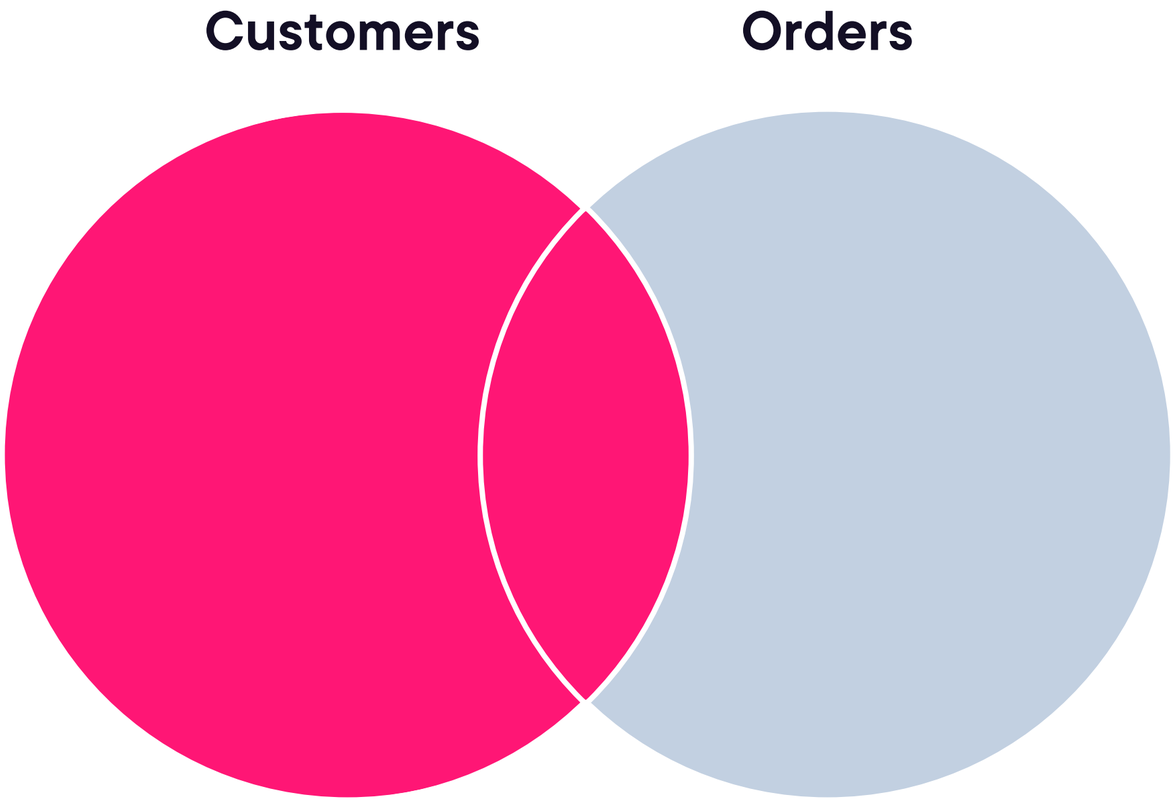
Left Outer Join



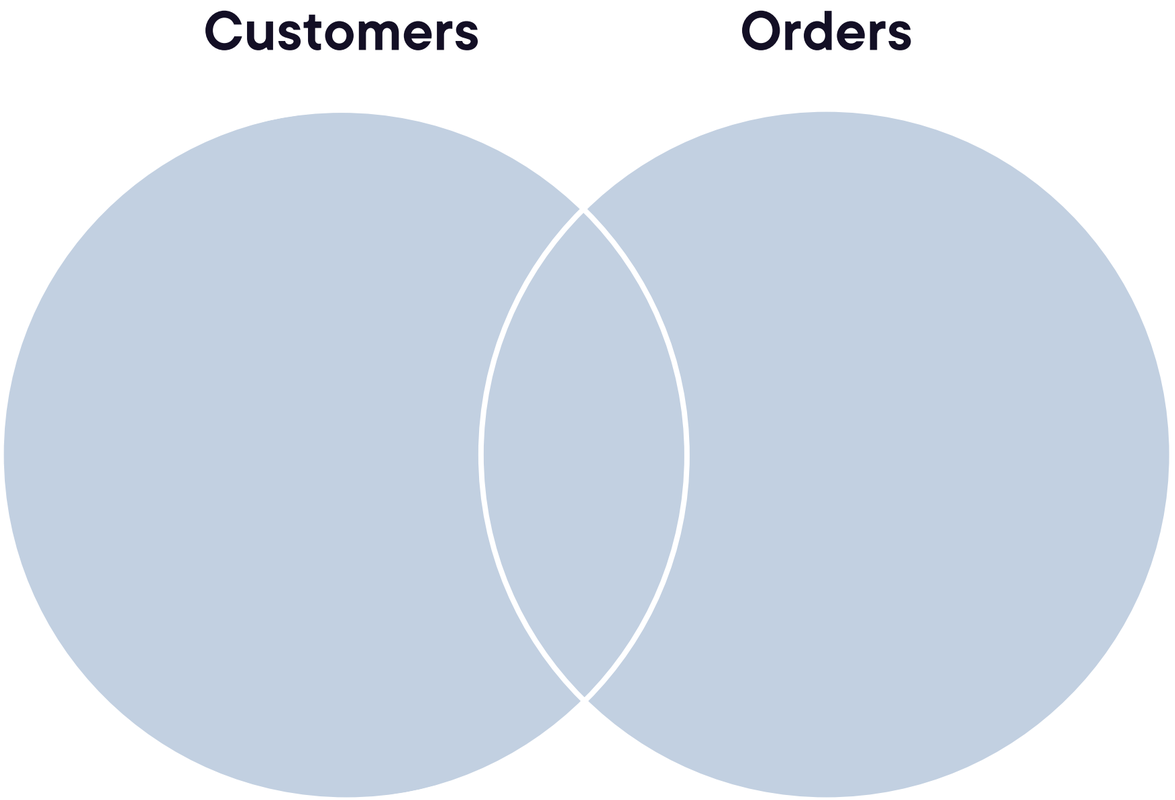
Left Outer Join



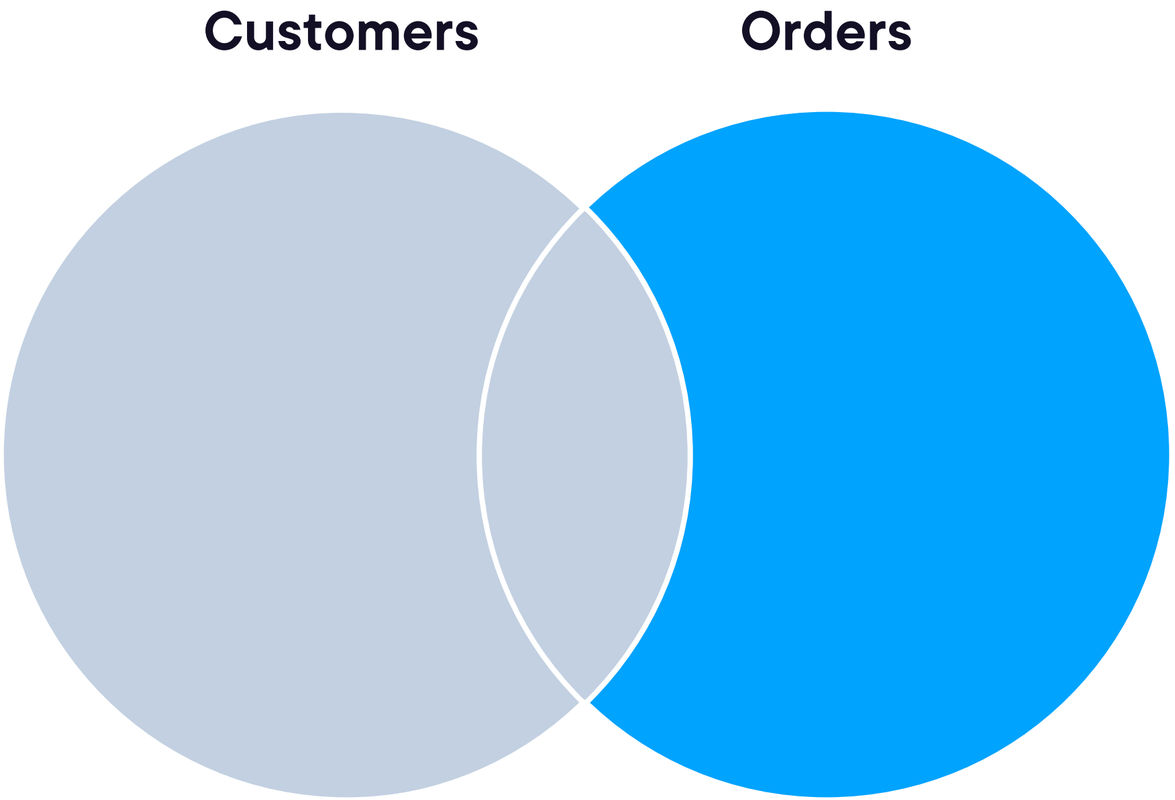
Left Outer Join



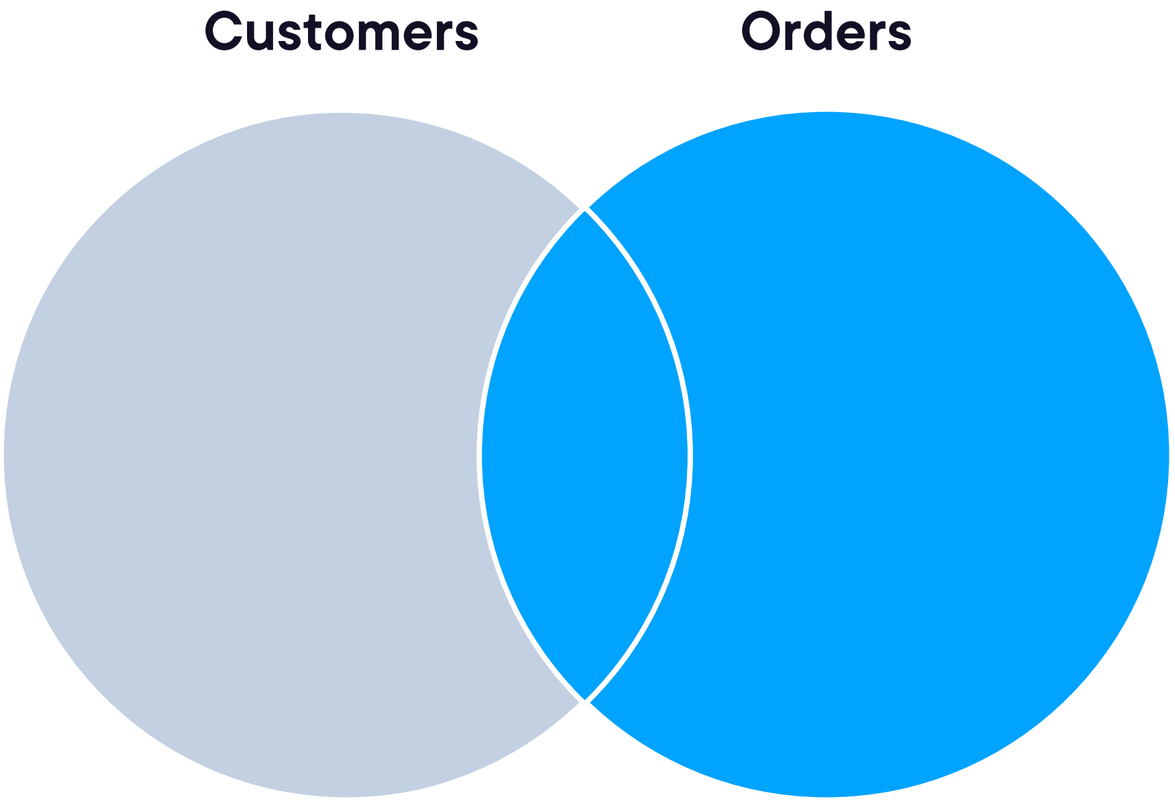
Right Outer Join



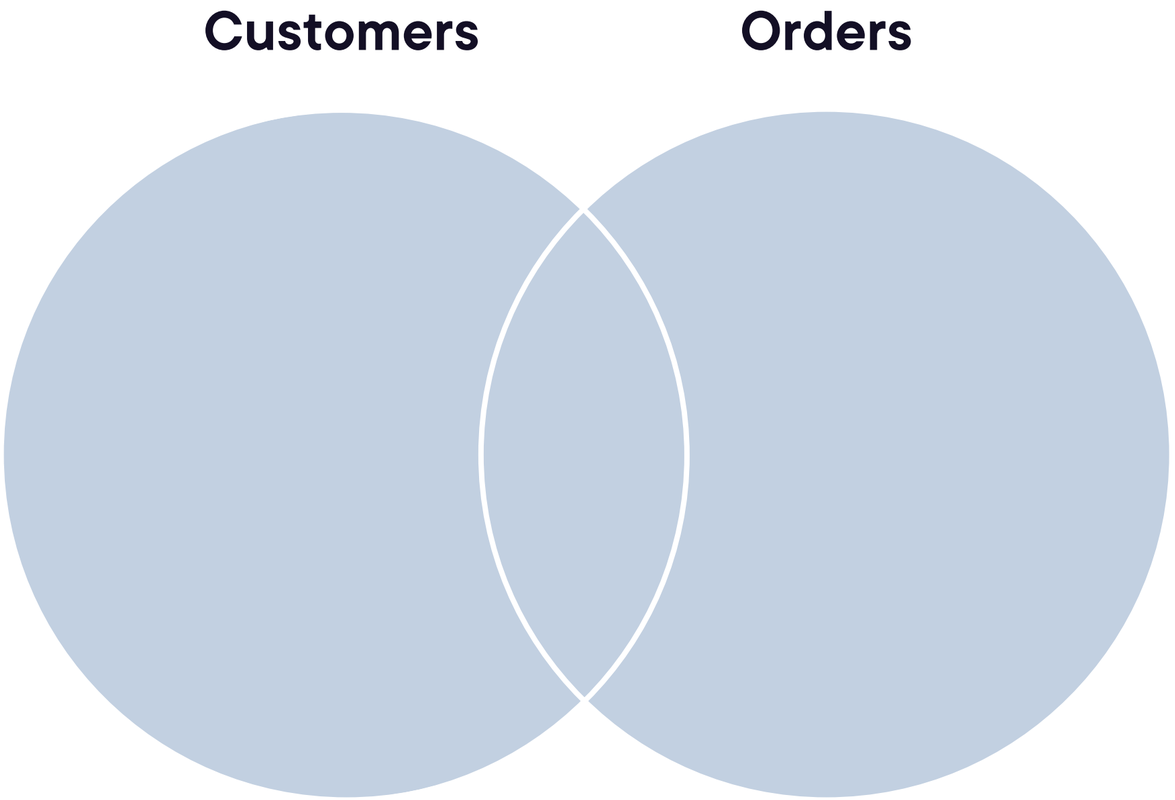
Right Outer Join



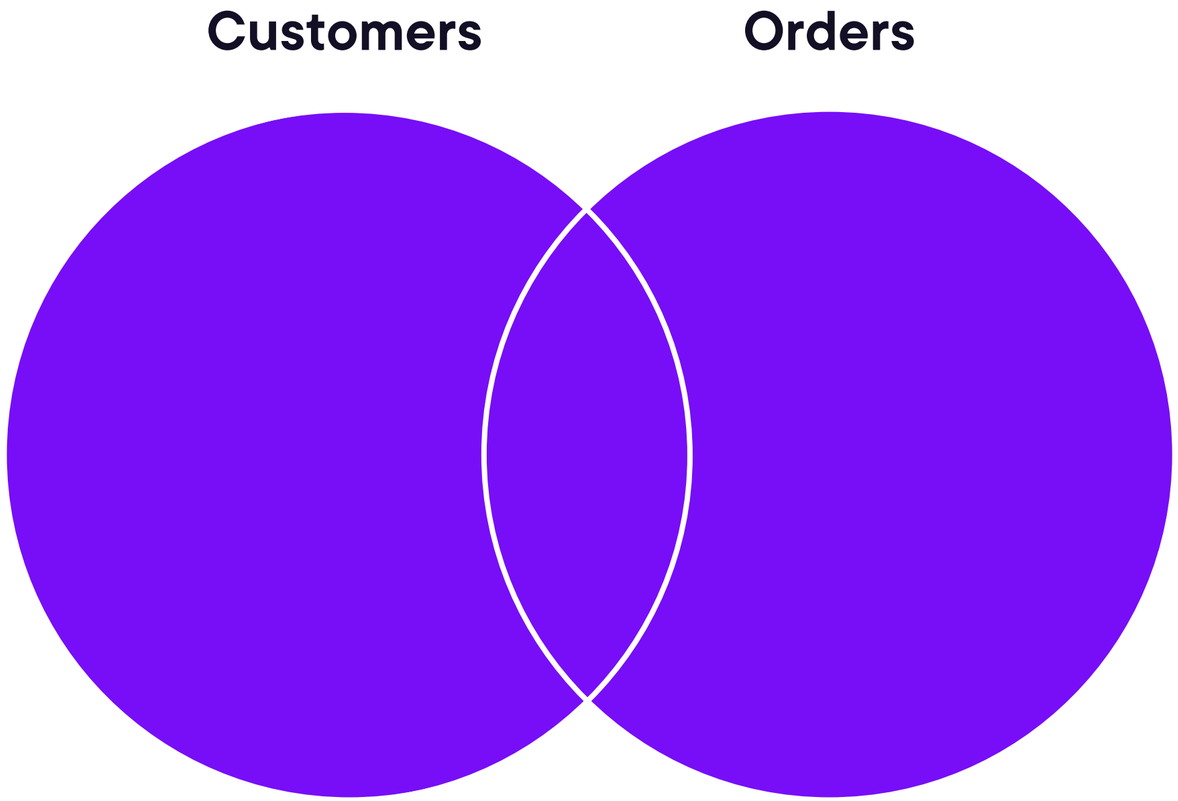
Right Outer Join



Full Outer Join



Full Outer Join



```sql
-- standard syntax SQL-92

SELECT *
FROM Customers AS c
    LEFT OUTER JOIN Orders  AS o
    ON c.CustomerKey = o.CustomerKey;
```

```sql
-- Outer keyword is optional

SELECT *
FROM Customers AS c
    LEFT JOIN Orders  AS o
    ON c.CustomerKey = o.CustomerKey;
```

Demo

Use outer joins

- Retrieve the list of customer who didn’t place any orders

Self Joins

Joins

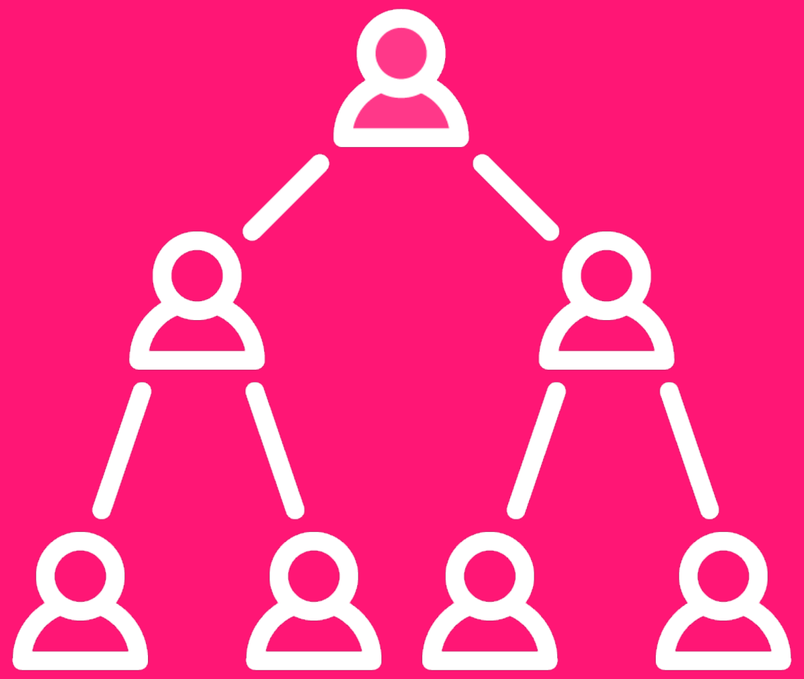
Cross joins

Inner joins

Outer joins

```sql
SELECT
FROM Customers AS c1
    CROSS JOIN  Customers AS c2
    ON c1.CustomerKey = c2.CustomerKey;
```

Aliasing



Most common use case

Working with human resources information.

Summary

Combining data

Logical phases

- Cartesian product
- Filtering data
- Adding outer rows

Join types

- Cross join
- Inner join
- Outer join

Up Next:

Cleaning Data