

# Inner en Outer Joins

In deze les leer je wat een **Cartesian product** is en hoe je deze kunt visualiseren. Daarna leer je het verschil tussen de **inner**- en de verschillende **outer joins**.

## Join

Je kunt een join visualiseren.

Stel je maakt een join tussen de tabellen persoon en de tabel ziektemeldingen.

### Tabel persoon

| ID   | voornaam | achternaam    |
|------|----------|---------------|
| 1001 | Aagje    | de Groot      |
| 1002 | Amber    | Nieuwenhuzien |

### Tabel ziektemeldingen

| ID | persoon | datum_van  | datum_tot |
|----|---------|------------|-----------|
| 1  | 1002    | 14-06-2018 | 16-6-2018 |
| 2  | 1002    | 1-8-2018   | 6-8-2018  |
| 3  | 1003    | 2-8-2018   | 3-8-2018  |

## INNER JOIN

Nu maak je een query waarbij je de twee tabellen samenvoegd.

```
SELECT *  
FROM persoon, ziektemelding
```

```
SELECT *  
FROM persoon  
INNER JOIN ziektemelding
```

Deze queries zijn hetzelfde, hoewel de tweede is volgens de ANSI standaard en is 'hoe het hoort', omdat die duidelijker is.

Door deze query met een join wordt het **Cartesian product** bepaald:

| ID   | voornaam | achternaam        | ID | persoon | dataum_van | datum_tot |
|------|----------|-------------------|----|---------|------------|-----------|
| 1001 | Aagje    | de Groot          | 1  | 1002    | 14-06-2018 | 16-6-2018 |
| 1001 | Aagje    | de Groot          | 2  | 1002    | 1-8-2018   | 6-8-2018  |
| 1001 | Aagje    | de Groot          | 3  | 1003    | 2-8-2018   | 3-8-2018  |
| 1002 | Amber    | Nieuwenhuzie<br>n | 1  | 1002    | 14-06-2018 | 16-6-2018 |
| 1002 | Amber    | Nieuwenhuzie<br>n | 2  | 1002    | 1-8-2018   | 6-8-2018  |
| 1002 | Amber    | Nieuwenhuzie<br>n | 3  | 1003    | 2-8-2018   | 3-8-2018  |

Alle combinaties van alle rijen van de twee tabellen worden dus naast elkaar gezet. Hierna begint de selectie aan de hand van de WHERE clause van de query.

```
SELECT *
FROM persoon, ziektemelding
where persoon.id=ziektemelding.persoon
```

```
SELECT *
FROM persoon
INNER JOIN ziektemelding
on persoon.id=ziektemelding.persoon
```

Dit resulteert in het 'verwijderen' van de vier regels van het Cartesian product:

| ID              | voornaam | achternaam        | ID | persoon         | dataum_van | datum_tot |
|-----------------|----------|-------------------|----|-----------------|------------|-----------|
| <del>1001</del> | Aagje    | de Groot          | 1  | <del>1002</del> | 14-06-2018 | 16-6-2018 |
| <del>1001</del> | Aagje    | de Groot          | 2  | <del>1002</del> | 1-8-2018   | 6-8-2018  |
| <del>1001</del> | Aagje    | de Groot          | 3  | <del>1003</del> | 2-8-2018   | 3-8-2018  |
| 1002            | Amber    | Nieuwenhuzie<br>n | 1  | 1002            | 14-06-2018 | 16-6-2018 |
| 1002            | Amber    | Nieuwenhuzie<br>n | 2  | 1002            | 1-8-2018   | 6-8-2018  |
| <del>1002</del> | Amber    | Nieuwenhuzie<br>n | 3  | <del>1003</del> | 2-8-2018   | 3-8-2018  |

Het resultaat is dat je alleen de regels krijgt waarvan de primary key en foreign key hetzelfde zijn.

## LEFT OUTER JOIN

```
SELECT *  
FROM persoon  
LEFT OUTER JOIN ziektemelding  
on persoon.id=ziektemelding.persoon
```

| ID   | voornaam | achternaam        | ID | persoon | dataum_van | datum_tot |
|------|----------|-------------------|----|---------|------------|-----------|
| 1001 | Aagje    | de Groot          | 1  | 1002    | 14-06-2018 | 16-6-2018 |
| 1001 | Aagje    | de Groot          | 2  | 1002    | 1-8-2018   | 6-8-2018  |
| 1001 | Aagje    | de Groot          | 3  | 1003    | 2-8-2018   | 3-8-2018  |
| 1002 | Amber    | Nieuwenhuzie<br>n | 1  | 1002    | 14-06-2018 | 16-6-2018 |
| 1002 | Amber    | Nieuwenhuzie<br>n | 2  | 1002    | 1-8-2018   | 6-8-2018  |
| 1002 | Amber    | Nieuwenhuzie<br>n | 3  | 1003    | 2-8-2018   | 3-8-2018  |

Bij een **left** outer join wordt in de eerste stap hetzelfde gedaan als bij een inner join. Er wordt nu alleen gekeken of er er in de **linker** tabel regels zijn die niet één keer matchen met de rechter. Als dat zo is dan wordt deze regel uit de linker table dus toch afgedrukt. De kolommen uit de rechter kolom zijn er niet en zullen dus ook niet worden afgedrukt (je ziet NULL values).

Dus bij een left outer join worden alle regels uit de linker tabel (dat is de tabel na de select) ten minste één keer afgedrukt.

## RIGHT OUTER JOIN

```
SELECT *  
FROM persoon  
RIGHT OUTER JOIN ziektemelding  
on persoon.id=ziektemelding.persoon
```

| ID   | voornaam | achternaam | ID | persoon | dataum_van | datum_tot |
|------|----------|------------|----|---------|------------|-----------|
| 1001 | Aagje    | de Groot   | 1  | 1002    | 14-06-2018 | 16-6-2018 |

|      |       |                   |   |      |            |           |
|------|-------|-------------------|---|------|------------|-----------|
| 1001 | Aagje | de Groot          | 2 | 1002 | 1-8-2018   | 6-8-2018  |
| 1001 | Aagje | de Groot          | 3 | 1003 | 2-8-2018   | 3-8-2018  |
| 1002 | Amber | Nieuwenhuzie<br>n | 1 | 1002 | 14-06-2018 | 16-6-2018 |
| 1002 | Amber | Nieuwenhuzie<br>n | 2 | 1002 | 1-8-2018   | 6-8-2018  |
| 1002 | Amber | Nieuwenhuzie<br>n | 3 | 1003 | 2-8-2018   | 3-8-2018  |

Bij een **right** outer join wordt in de eerste stap hetzelfde gedaan als bij een inner join. Er wordt nu alleen gekeken of er er in de **rechter** tabel regels zijn die niet één keer matchen met de rechter. Als dat zo is dan wordt deze regel uit de rechter table dus toch afgedrukt. De kolommen uit de linker kolom zijn er niet en zullen dus ook niet worden afgedrukt (je ziet NULL values).

Dus bij een right outer join worden alle regels uit de rechter tabel (dat is de tabel na de on) ten minste één keer afgedrukt.

## FULL OUTER JOIN

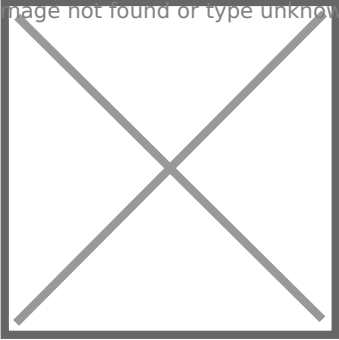
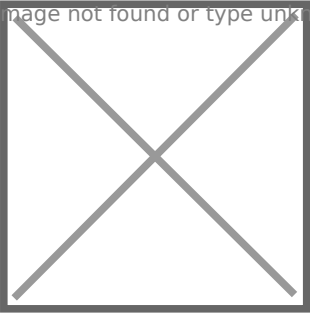
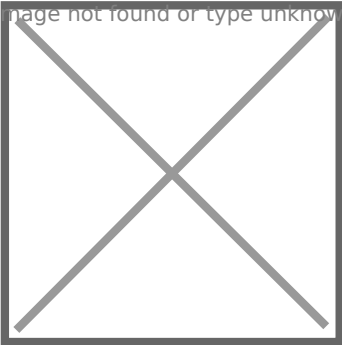
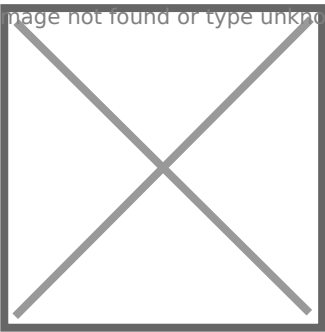
Bij een full outer join worden alle regels uit de linker en uit de rechter kolom ten minste één keer afgedrukt. Dit is een combinatie van de left- en right outer join. MariaDB en MySQL ondersteund geen full outer join. Deze kan worden uitgevoerd door de left- en right outer join te combineren met een union.

```
SELECT *
FROM persoon
LEFT OUTER JOIN ziektemelding
on persoon.id=ziektemelding.persoon

UNION

SELECT *
FROM persoon
RIGHT OUTER JOIN ziektemelding
on persoon.id=ziektemelding.persoon
```

## Samengevat

|  |   |
|--|---|
| <p>Image not found or type unknown</p>    | <pre>SELECT * FROM A <b>INNER JOIN</b> B ON ...</pre>   |
| <p>Image not found or type unknown</p>    | <pre>SELECT * FROM A <b>LEFT OUTER JOIN</b> B ON ...</pre>  |
| <p>Image not found or type unknown</p>   | <pre>SELECT * FROM A <b>RIGHT OUTER JOIN</b> B ON ...</pre>   |
| <p>Image not found or type unknown</p>  | <pre>SELECT * FROM A <b>FULL OUTER JOIN</b> B ON ...</pre> <p>(je kunt hier A en B ook omdraaien in de query)</p> |

Revision #8

Created 15 October 2019 07:49:17 by Max

Updated 22 January 2022 19:25:38 by Max