

## Uitleg INNER JOIN sweetshop

In het filmpje van Bob's Sweetshop ([https://www.youtube.com/watch?v=ia4eCxPPc\\_o](https://www.youtube.com/watch?v=ia4eCxPPc_o)) dat jullie eerder hebben gekeken worden 2 tabellen gebruikt, SWEET en CATEGORY. Deze tabellen worden gebruikt om de theorie voor het joinen uit te leggen.

Maak nu eerst de database sweetshop aan met deze 2 tabellen aan Gebruik hiervoor het script create\_sweetshop.sql uit Onderwijs Online. De inhoud van de tabellen kun je controleren met:

```
use sweetshop;  
select * from sweet;  
select * from category;
```

Tabel SWEET

	sweetid	sweetname	categoryid
►	25	Orange Toffee	1
	26	Tablet	1
	27	English Toffee	1
	28	Flumps	2
	29	Coconut Mallows	2
	30	Marshmallow Mushrooms	2
	31	Mallow Canbles	2
	33	Cola Cubes	4
	34	Lemon Drops	4
	35	Peanut Brittle	4
	36	Orange Beans	5
	37	Lime Beans	5

Tabel CATEGORY

	categoryid	categoryname
►	1	Toffee
	2	Marshmallow
	3	Fudge
	4	Hard sweet
	5	Jelly Candies
	6	Candy

We willen nu de een lijst maken met alle namen van de snoepjes met daarbij de naam van de categorie. Deze informatie staat in 2 tabellen. Als je informatie uit meer dan 1 tabel wil tonen, dan moet je een INNER JOIN gebruiken om deze tabellen te koppelen.

Het volgende statement

```
select sweetname, categoryname
from   sweet inner join category;
```

levert onderstaande output

	sweetname	categoryname
►	Orange Toffee	Toffee
	Orange Toffee	Marshmallow
	Orange Toffee	Fudge
	Orange Toffee	Hard sweet
	Orange Toffee	Jelly Candies
	Orange Toffee	Candy
	Tablet	Toffee
	Tablet	Marshmallow
	Tablet	Fudge
	Tablet	Hard sweet
	Tablet	Jelly Candies
	Tablet	Candy
	English Toffee	Toffee
	English Toffee	Marshmallow
	English Toffee	Fudge
	English Toffee	Hard sweet
	English Toffee	Jelly Candies
	English Toffee	Candy
	Flumps	Toffee
	Flumps	Marshmallow
	Flumps	Fudge
	Flumps	Hard sweet
	Flumps	Jelly Candies
	Flumps	Candy
	Coconut Mallows	Toffee
	Coconut Mallows	Marshmallow
	Coconut Mallows	Fudge

Dit is maar een deel van de output. Wat er nu gebeurt is dat elke rij uit de SWEET tabel gekoppeld wordt aan elke rij in de CATEGORY tabel. Je moet bij een join namelijk aangeven welke kolommen je gebruikt om de relatie te leggen. Zoals je weet wordt die relatie gelegd door de primary key CATEGORYID in CATEGORY en de foreign key CATEGORYID in de SWEET tabel.

Als je het statement aanpast naar

```
select sweetname, categoryname  
from   sweet inner join category  
       on sweet.categoryid=category.categoryid;
```

krijg je wel het goede resultaat:

	sweetname	categoryname
►	Orange Toffee	Toffee
	Tablet	Toffee
	English Toffee	Toffee
	Flumps	Marshmallow
	Coconut Mallows	Marshmallow
	Marshmallow Mushrooms	Marshmallow
	Mallow Canbles	Marshmallow
	Cola Cubes	Hard sweet
	Lemon Drops	Hard sweet
	Peanut Brittle	Hard sweet
	Orange Beans	Jelly Candies
	Lime Beans	Jelly Candies

In de ON clausule geef je welke kolommen een relatie hebben. De kolom CATEGORYID komt in beide tabellen, daarom moet je wel aangeven uit welke tabel de kolom gebruikt moet worden. Als je dat niet doet krijg je een foutmelding:

```
select sweetname, categoryname  
from   sweet inner join category  
       on categoryid=categoryid;
```

Error Code: 1052. Column 'categoryid' in on clause is ambiguous

Dit geldt niet alleen voor de ON clausule maar op alle plaatsen in het SELECT statement.

Stel we willen ook het ID van de categorie tonen, dan wordt het statement als volgt:

```
select sweetname, categoryname, category.categoryid
from   sweet inner join category
      on sweet.categoryid=category.categoryid;
```

	sweetname	categoryname	categoryid
►	Orange Toffee	Toffee	1
	Tablet	Toffee	1
	English Toffee	Toffee	1
	Flumps	Marshmallow	2
	Coconut Mallows	Marshmallow	2
	Marshmallow Mushrooms	Marshmallow	2
	Mallow Canbles	Marshmallow	2
	Cola Cubes	Hard sweet	4
	Lemon Drops	Hard sweet	4
	Peanut Brittle	Hard sweet	4
	Orange Beans	Jelly Candies	5
	Lime Beans	Jelly Candies	5

Je hebt al geleerd dat je een alias kan gebruiken bij een kolom. Stel je wil duidelijk aangeven dat het ID de PK is van de tabel CATEGORY, dan kun je dat als volgt doen:

```
select sweetname, categoryname, category.categoryid AS PK_category
from   sweet inner join category
      on sweet.categoryid=category.categoryid;
```

	sweetname	categoryname	PK_category
►	Orange Toffee	Toffee	1
	Tablet	Toffee	1
	English Toffee	Toffee	1
	Flumps	Marshmallow	2
	Coconut Mallows	Marshmallow	2
	Marshmallow Mushrooms	Marshmallow	2
	Mallow Canbles	Marshmallow	2
	Cola Cubes	Hard sweet	4
	Lemon Drops	Hard sweet	4
	Peanut Brittle	Hard sweet	4
	Orange Beans	Jelly Candies	5
	Lime Beans	Jelly Candies	5

Zo kun je ook voor de tabel een alias opgeven. Dit is handig als je hele lange of onduidelijke tabelnamen hebt. In sommige gevallen (die jullie later leren) is het zelfs noodzakelijk.

Het vorige voorbeeld, maar nu met een alias voor de tabellen:

```
select sweetname, categoryname, cat.categoryid AS PK_category
from   sweet swt inner join category cat
      on swt.categoryid=cat.categoryid;
```

Dit levert (zoals te verwachten is) hetzelfde resultaat:

	sweetname	categoryname	PK_category
►	Orange Toffee	Toffee	1
	Tablet	Toffee	1
	English Toffee	Toffee	1
	Flumps	Marshmallow	2
	Coconut Mallows	Marshmallow	2
	Marshmallow Mushrooms	Marshmallow	2
	Mallow Canbles	Marshmallow	2
	Cola Cubes	Hard sweet	4
	Lemon Drops	Hard sweet	4
	Peanut Brittle	Hard sweet	4
	Orange Beans	Jelly Candies	5
	Lime Beans	Jelly Candies	5

Je moet er wel aan denken dat als je eenmaal een tabelalias gebruikt, je dit overal moet doen. Onderstaand statement geeft namelijk een foutmelding:

```
select sweetname, categoryname, category.categoryid AS PK_category
from   sweet swt inner join category cat
      on swt.categoryid=cat.categoryid;
```

Error Code: 1054. Unknown column 'category.categoryid' in 'field list'