

# Basisopdrachten in SQL

Opdracht	Syntaxis	Beschrijving
ALTER TABLE ADD column	<b>ALTER TABLE table_name ADD column_name datatype;</b>	Het wordt gebruikt om kolommen toe te voegen aan een tabel in een database
ALTER TABLE ADD constraint	<b>ALTER TABLE table_name ADD CONSTRAINT constraint_name PRIMARY KEY (c1,c2); ALTER TABLE table_name ADD CONSTRAINT constraint_name FOREIGN KEY (c1,c2) REFERENCES table_name (c1,c2);</b>	Het wordt gebruikt om een Primaire sleutel op een tabel te plaatsen.  Het wordt gebruikt om een Verwijzende sleutel op een tabel te plaatsen.
AND	<b>SELECT column_name(s) FROM table_name WHERE column_1 = value_1 AND column_2 = value_2;</b>	Het is een operator die wordt gebruikt om twee voorwaarden te combineren
AS	<b>SELECT column_name AS 'Alias' FROM table_name;</b>	Het is een trefwoord in SQL dat wordt gebruikt om een kolom of tabel te hernoemen met een aliasnaam
AVG	<b>SELECT AVG(column_name) FROM table_name;</b>	Het wordt gebruikt om een numerieke kolom samen te voegen en het gemiddelde te retourneren
BETWEEN	<b>SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value_1 AND value_2;</b>	Het is een bewerking die wordt gebruikt om het resultaat binnen een bepaald bereik te filteren
CASE	<b>SELECT column_name, CASE WHEN condition THEN 'Result_1' WHEN condition THEN 'Result_2' ELSE 'Result_3' END FROM table_name;</b>	Het is een instructie die wordt gebruikt om verschillende uitvoer binnen een SELECT-instructie te creëren
COUNT	<b>SELECT COUNT(column_name) FROM table_name;</b>	Het is een functie die de naam van een kolom als argument neemt en het aantal rijen telt wanneer de kolom niet NULL is
CREATE DATABASE	<b>CREATE DATABASE database_name;</b>	Het wordt gebruikt om een nieuwe database aan te maken
CREATE TABLE	<b>CREATE TABLE table_name ( column_1 datatype, column_2 datatype, column_3 datatype);</b>	Het wordt gebruikt om een nieuwe tabel in een database te maken en de naam van de tabel en de kolommen erin op te geven
CREATE VIEW	<b>CREATE VIEW view_name AS SELECT column_name(2) FROM t1 WHERE column_name operator value;</b>	Het wordt gebruikt om een view aan te maken op kolommen van een of meerdere tabellen.
DELETE	<b>DELETE FROM table_name</b>	Het wordt gebruikt om de rijen uit een tabel te verwijderen

	<b>WHERE some_column = some_value;</b>	
<b>DROP DATABASE</b>	<b>DROP DATABASE database_name;</b>	Het wordt gebruikt om een database te verwijderen.
<b>DROP TABLE</b>	<b>DROP TABLE table_name;</b>	Het wordt gebruikt om een tabel te verwijderen.
<b>GROUP BY</b>	<b>SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name;</b>	Het is een clause in SQL die wordt gebruikt voor statistische functies in samenwerking met de SELECT-instructie
<b>HAVING</b>	<b>SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name HAVING COUNT(*) &gt; value;</b>	Het wordt gebruikt in SQL omdat het sleutelwoord WHERE niet kan worden gebruikt in aggregatiefuncties
<b>IN</b>	<b>SELECT column_name(s) FROM table_name WHERE column_name IN ( value_1, value_2,...);</b>	Het is een bewerking die wordt gebruikt om het resultaat binnen een bepaalde reeks te filteren
<b>INNER JOIN</b>	<b>SELECT column_name(s) FROM table_1 JOIN table_2 ON table_1.column_name = table_2.column_name;</b>	Het wordt gebruikt om rijen uit verschillende tabellen te combineren als de Join-voorwaarde TRUE wordt
<b>INSERT</b>	<b>INSERT INTO table_name (column_1, column_2, column_3) VALUES (value_1, 'value_2', value_3);</b>	Het wordt gebruikt om nieuwe rijen aan een tabel toe te voegen
<b>IS NULL/ IS NOT NULL</b>	<b>SELECT column_name(s) FROM table_name WHERE column_name IS NULL;</b>	Het is een operator die wordt gebruikt met de WHERE-component om te controleren op lege waarden
<b>LIKE</b>	<b>SELECT column_name(s) FROM table_name WHERE column_name LIKE 'pattern';</b>	Het is een speciale operator die wordt gebruikt met de WHERE-component om naar een specifiek patroon in een kolom te zoeken
<b>LIMIT</b>	<b>SELECT column_name(s) FROM table_name LIMIT number;</b>	Het is een clause om het maximale aantal rijen op te geven dat de resultaten set moet hebben
<b>MAX</b>	<b>SELECT MAX(column_name) FROM table_name;</b>	Het is een functie die een aantal kolommen als argument neemt en de grootste waarde daarvan retourneert
<b>MIN</b>	<b>SELECT MIN(column_name) FROM table_name;</b>	Het is een functie die een aantal kolommen als argument neemt en de kleinste waarde ervan retourneert
<b>OR</b>	<b>SELECT column_name FROM table_name WHERE column_name = value_1 OR column_name = value_2;</b>	Het is een operator die wordt gebruikt om de resultaatset te filteren zodat deze alleen de rijen bevat waarin een van beide voorwaarden WAAR is

ORDER BY	<b>SELECT column_name FROM table_name ORDER BY column_name ASC   DESC;</b>	Het is een clause die wordt gebruikt om de resultaatset op een bepaalde kolom te sorteren, hetzij numeriek of alfabetisch
OUTER JOIN	<b>SELECT column_name(s) FROM table_1 LEFT JOIN table_2 ON table_1.column_name = table_2.column_name;</b>	Het is uitgegeven om rijen uit verschillende tabellen te combineren, zelfs als de voorwaarde NIET WAAR is
ROUND	<b>SELECT ROUND(column_name, integer) FROM table_name;</b>	Het is een functie die de kolomnaam en een geheel getal als argument neemt en de waarden in een kolom afrondt op het aantal decimalen gespecificeerd door een geheel getal
SELECT	<b>SELECT column_name FROM table_name;</b>	Het is een instructie die wordt gebruikt om gegevens uit een database op te halen
SELECT DISTINCT	<b>SELECT DISTINCT column_name FROM table_name;</b>	Het wordt gebruikt om aan te geven dat de instructie een query is die unieke waarden in opgegeven kolommen retourneert
SUM	<b>SELECT SUM(column_name) FROM table_name;</b>	Het is een functie die wordt gebruikt om de som van waarden uit een bepaalde kolom te retourneren
Sub-query	<b>SELECT column_name(s) FROM table_name WHERE column_name IN (SELECT column_name FROM table_name);</b>	Het resultaat uit de sub-query wordt gebruikt in de WHERE van de omliggende query.
UPDATE	<b>UPDATE table_name SET some_column = some_value WHERE some_column = some_value;</b>	Het wordt gebruikt om rijen in een tabel te bewerken
USE	<b>USE database_name;</b>	Het wordt gebruikt om een database courant te maken.
WHERE	<b>SELECT column_name(s) FROM table_name WHERE column_name operator value;</b>	Het is een clause die wordt gebruikt om de resultaatset te filteren om de rijen op te nemen waarvan de voorwaarde TRUE is
WITH	<b>WITH temporary_name AS (SELECT * FROM table_name) SELECT * FROM temporary_name WHERE column_name operator value;</b>	Het wordt gebruikt om het resultaat van een bepaalde query op te slaan in een tijdelijke tabel met behulp van een alias

# Opdrachten en syntaxis voor het opvragen van gegevens uit een enkele tabel en meerdere tabellen:

Enkele tabel	Meerdere tabellen
<b>SELECT column_name(s)</b> <b>FROM t1</b> <b>WHERE column_name1 IN (SELECT column_name2 FROM t1);</b> Om met een subquery gegevens uit tabel t1 te selecteren waarbij de kolom1 gelijk is aan de waarde uit kolom2 uit dezelfde tabel t1.	<b>SELECT column_name(s)</b> <b>FROM t1</b> <b>WHERE column_name IN (SELECT column_name FROM t2);</b> Om met een subquery gegevens uit tabel t1 te selecteren waarvoor geldt dat de kolom gelijk is aan de kolomwaarde uit de selectie uit t2.
<b>SELECT c1</b> <b>FROM t;</b> Om de gegevens in kolom c1 uit tabel t te selecteren	<b>SELECT c1, c2</b> <b>FROM t1</b> <b>INNER JOIN t2 on condition;</b> Selecteer kolom c1 en c2 uit tabel t1 en voer een inner join uit tussen t1 en t2
<b>SELECT *</b> <b>FROM t;</b> Om alle rijen en kolommen uit tabel t te selecteren	<b>SELECT c1, c2</b> <b>FROM t1</b> <b>LEFT JOIN t2 on condition;</b> Selecteer kolom c1 en c2 uit tabel t1 en voer een linkse join uit tussen t1 en t2
<b>SELECT c1</b> <b>FROM t</b> <b>WHERE c1 = 'test';</b> Om gegevens in kolom c1 uit tabel t te selecteren, waarbij c1=test	<b>SELECT c1, c2</b> <b>FROM t1</b> <b>RIGHT JOIN t2 on condition;</b> Selecteer kolom c1 en c2 uit tabel t1 en voer een rechter join uit tussen t1 en t2
<b>SELECT c1</b> <b>FROM t</b> <b>ORDER BY c1 ASC (DESC);</b> Gegevens in kolom c1 uit tabel t selecteren in oplopende of aflopende volgorde	<b>SELECT c1, c2</b> <b>FROM t1</b> <b>FULL OUTER JOIN t2 on condition;</b> Selecteer kolom c1 en c2 uit tabel t1 en voer een volledige outer join uit tussen t1 en t2
<b>SELECT c1</b> <b>FROM t</b> <b>ORDER BY c1 LIMIT n OFFSET</b> <b>Offset;</b> Om de offset van rijen over te slaan en de volgende n rijen te retourneren	<b>SELECT c1, c2</b> <b>FROM t1</b> <b>CROSS JOIN t2;</b> Selecteer kolom c1 en c2 uit tabel t1 en produceer een Cartesiaans product van rijen in een tabel
<b>SELECT c1, aggregate(c2)</b> <b>FROM t</b> <b>GROUP BY c1;</b> Om rijen te groeperen met behulp van een aggregatiefunctie	<b>SELECT c1, c2</b> <b>FROM t1, t2;</b> Selecteer kolom c1 en c2 uit tabel t1 en produceer een Cartesiaans product van rijen in een tabel
<b>SELECT c1, aggregate(c2)</b> <b>FROM t</b> <b>GROUP BY c1 HAVING condition;</b> Groepeer rijen met behulp van een aggregatiefunctie en filter deze groepen met behulp van de 'HAVING'-clausule	<b>SELECT c1, c2</b> <b>FROM t1 A</b> <b>INNER JOIN t2 B on condition;</b> Selecteer kolom c1 en c2 uit tabel t1 en koppel deze aan zichzelf met behulp van de INNER JOIN-clausule

