Database Query’s Document

Isotita

|  |  |
| --- | --- |
| Leereenheid: | Isotita |
| Versienummer: | 0.1 |
| Auteur(s): | Mohamad Alamoroush |
| Datum: | 01-06-2022 |

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Versie | Wie | Wijzigingen |
| 01-06-2022 | 0.1 | Mohamad | Database Query’s Document |
| 07-06-2022 | 0.2 | Mohamad | Uitgebreide query’s bijvoegen aan het document |

Inhoudsopgave

[Versiebeheer 2](#_Toc105103595)

Inleiding

Deze document maak ik om database query’s te schrijven met hun resultaat. Dat gebruik ik omdat ik veel van de query’s niet gebruik in mijn applicatie of omdat ze daar niet passen of omdat ze als extra kunnen gebruikt worden maar er is weinig tijd om dat toe te passen.

Database Query’s

Zoals eerder benoemd in mij databaseontwerp heb ik drie tabellen namelijk:

1. Company
2. Job
3. Applied

## Company-tabel

Ik begin met data uit company-tabel en dan voer ik query’s uit en dan toon ik het resultaat daarvan.

SELECT id, name, email, password, mobile\_number, address, city, zipcode

FROM company

ORDER BY id ASC;

Deze query zal zorgen dat ik alle geregistreerde bedrijven terugkrijg van de kleinste id getal tot het grootse.

Resultaat:



SELECT id, name, email, password, mobile\_number, address, city, zipcode

FROM company

ORDER BY id DESC;

Deze query zal zorgen dat ik alle geregistreerde bedrijven terugkrijg van de grootse id getal tot het kleinste.

Resultaat:



SELECT id, name, email, password, mobile\_number, address, city, zipcode

FROM company

WHERE id < 8;

Deze query geeft alle geregistreerde bedrijven terug waar de id kleiner is dan 8. Die zijn twee in totaal en dat is ook te tonen door het resultaat in het volgende figuur.



SELECT id, name, email, password, mobile\_number, address, city, zipcode

FROM company

ORDER BY id DESC

LIMIT 2;

Deze query zal zorgen dat ik alle geregistreerde bedrijven terugkrijg van de grootse id getal tot het kleinste. Wel met een limit van 2 records, dus ik verwacht bedrijven met ids 8 en 7 terug.

Resultaat:

 SELECT company.id, company.name, job.id, title, description

From company

INNER JOIN job

ON company.id = company\_id

GROUP BY job.id

Deze query zal alle bedrijven teruggeven met hun geplaatste vacatures. Bedrijven zonder vacatures komen hier niet bij

Resultaat:

Graphical user interface, text, application

Description automatically generated

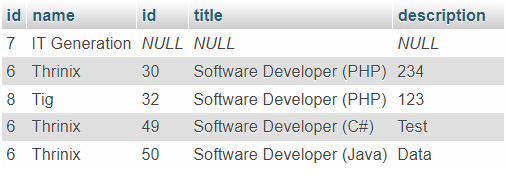
SELECT company.id, company.name, job.id, title, description

From company

LEFT JOIN job

ON company.id = company\_id

GROUP BY job.id

Deze query zal alle bedrijven teruggeven met hun geplaatste vacatures. Bedrijven zonder vacatures komen hier wel bij maar de vacaturegegevens staan dan op NULL

Resultaat:

SELECT company.id, company.name, job.id, title, description, COUNT(job\_id) AS totalAppliedPerJob From company

INNER JOIN job

ON company.id = company\_id

INNER JOIN applied

ON job.Id = job\_id

GROUP BY job.id;

Deze query haalt alle bedrijven die vacaturen hebben geplaatst met de aantal sollicitanten die gesolliciteerd hebben. Bedrijven met vacaturen waar niemand heeft gesolliciteerd komen hier niet bij.

Resultaat:

Graphical user interface, application

Description automatically generated

SELECT company.id, company.name, job.id, title, description, COUNT(job\_id) AS totalAppliedPerJob From company

INNER JOIN job

ON company.id = company\_id

LEFT JOIN applied

ON job.Id = job\_id

GROUP BY job.id;

Deze query haalt alle bedrijven die vacaturen hebben geplaatst. Dit keer zowel waar sollicitanten gesolliciteerd hebben als vacaturen waar niemand nog heeft gesolliciteerd.

Resultaat:

Graphical user interface, text, application

Description automatically generated

SELECT company.id, company.name, job.id, title, description, COUNT(job\_id) AS totalAppliedPerJob From company

LEFT JOIN job

ON company.id = company\_id

LEFT JOIN applied

ON job.Id = job\_id

GROUP BY job.id;

Dat query zorgt dat alle bedrijven worden opgehaald zowel bedrijven die vacaturen hebben geplaatst als bedrijven die nog geen vacaturen hebben geplaatst. Ten tweede haalt ze alle vacaturen waar sollicitanten hebben gesolliciteerd en vacaturen waar niemand heeft gesolliciteerd.

Resultaat:

Graphical user interface, text, application

Description automatically generated

## Job-tabel

SELECT id, company\_id, title, description, requirements, experience

FROM job

WHERE title LIKE "%S%"

Deze query zorgt dat alle vacaturen worden gehaald waar het karakter S zit in titel van een vacature. De twee percentage symbolen geven aan dat het karakter overal plek mag vinden.

Resultaat:

Graphical user interface, text, application

Description automatically generated

SELECT id, company\_id, title, description, requirements, experience

FROM job

WHERE title LIKE "%S"

Deze query zorgt dat alle vacaturen worden gehaald waar het karakter S zit in het laatste karakter van titel zit. Resultaat zou leeg zijn

Resultaat: 