**SQL DATABASE UITLEG**

**QUERIES.**

SELECT queries: Om specifieke data terug te vinden in de database, maak je gebruik van de SELECT statement. Ofterwijl: SELECT queries.

*Je kan een tabel zien als een entiteit(iets wat bestaat) bijvoorbeeld: Honden. Elke kolom in de tabel heeft zijn eigen naam die deze entiteit als beste beschrijft.*

**Als je een specifieke kolom wil selecteren van een bepaalde tabel dan doe je dit zo:**

SELECT column, other column 🡨 dit geeft aan welke kolom(men) je wil hebben

FROM tabel; 🡨 dit geeft aan uit welke tabel de kolom komt.

**Als we alle kolommen met informatie van de tabel willen hebben dan doe je het zo:**

SELECT \* 🡨 het sterretje betekend: alles. Dus selecteer alles.

FROM tabel; 🡨 van de desbetreffende tabel.

*\*Note: Zorg dat wat je wil selecteren en intypt overeenkomt met de namen, anders krijg je een error.*

Als je superveel kolommen heb dan is het onhandig om deze allemaal te doorzoeken totdat je bij de benodigde informatie komt. Hierom is er een andere querie die je kan schrijven die aan bepaalde voorwaarden moet voldoen zodat de nodige informatie tevoorschijn komt.

Bijvoorbeeld:

SELECT kolom

FROM tabel

WHERE voorwaarde

AND/OR andere voorwaarde; 🡨 *\*note: het volgende is optional vgm*

Het volgende kan je gebruiken voor numerieke data: (\*note: vergeet niet wel de WHERE er neer te zetten)

= != < <= >= 🡨 Zelfde betekenis zoals in andere programmeertalen🡪 kolomnaam >= 4

Kolomnaam BETWEEN 1.5 AND 4

Kolomnaam NOT BETWEEN 5 AND 10

Kolomnaam IN 1, 2, 3 🡨 Integer, kijkt of de aangegeven cijfers voorkomen in de kolom.

Kolomnaam NOT IN 4, 12, 15 🡨 omgekeerde van wat hierboven staat.

Opdrachten:

1. Find the movie with a row **id** of 6 ✓
2. Find the movies released in the **year**s between 2000 and 2010 ✓
3. Find the movies **not** released in the **year**s between 2000 and 2010
4. Find the first 5 Pixar movies and their release **year**
5. SELECT title, year FROM movies

WHERE year = 6;

1. SELECT title, year FROM movies

WHERE year BETWEEN 2000 AND 2010;

1. SELECT title, year FROM movies

WHERE year < 2000 OR year > 2010;

1. SELECT title, year FROM movies

WHERE year <= 2003;

\*nummer 4. Rare gedachtegang, ik ging er van uit dat je er rekening mee moest houden dat je de namen niet kon zien en of de release datum. Kennelijk moest je eerst kijken wat de release datum is na de 5e film en anders onder die datum kunnen selecteren. Zo dus:

SELECT title, year FROM movies

WHERE year <= 2003;

DEEL1

OPDRACHT1

Create a query to list out the following columns from the **tblEvent**table, with the most recent first (for which you'll need to use an **ORDER BY**clause):

* The event name
* The event date

Hoe heb ik dit gedaan?:

SELECT Eventname, EventDate FROM tblEvent

ORDER BY EventDate DESC; 🡨DESC staat voor descanding, en zo krijg je dus de recentste datum te zien.

----------------------------------------------------------------------------------

Opdracht 2

Write a query to show the first 5 events (in date order) from the **tblEvent**table:

There are a few things to notice about this:

* You should give the columns aliases (**What**and **Details**in this case);
* Use SELECT TOP 5 to limit the results to 5 rows;
* Even though you're sorting by the event date, it doesn't have to be included in your results.

Hoe heb ik dit gedaan?:

SELECT TOP 5 EventName, EventDetails FROM tblEvent

ORDER BY EventDate;

\*noteer dat ik eventdate niet heb geselecteerd, hierdoor laat hij deze ook niet zien. Je heb wel de command order by gegeven en daarom laat hij hem wel zien op volgorde van datum

Als je een database runt word deze automatisch aangemaakt, je hoeft hem dus niet op te slaan. Het is ook zo dat je niet 2 databases met dezelfde naam kan hebben (vgm ook niet met dezelfde inhoud maar dat weet ik niet zeker)

* + Niet letten op de foutmelding die je krijgt nadat je bepaalde kolommen heb geselecteerd na SELECT voordat je from tabelnaam heb gedaan. Dit doet hij omdat je de tabelnaam nog niet heb getypt.

----------------------------------------------------------------------------------

OPDRACHT 3

Create a query to list out the id number and name of the last 3 categories from the **tblCategory**table in alphabetical order:

OPLOSSING:

SELECT TOP 3 CategoryID, CategoryNAME FROM tblCategory 🡨 Met TOP 3 gaf ik aan dat ik 3 waardes wou van de ID en de NAAM.

ORDER BY CategoryNAME DESC; 🡨 met order by CategoryNAME DESC gaf ik aan dat ik op basis van alfabetische volgorde het de meest recente wou selecteren. (meest recente is mbv DESC.)

----------------------------------------------------------------------------------

OPDRACHT 4

Create a query which uses two separate **SELECT** statements to show the first and last 2 events in date order from the **tblEvent**table:

*Run two****SELECT****commands one after the other to get these results.*

OPLOSSING:

SELECT TOP 2 Eventname, EventDate FROM tblEvent

ORDER BY Eventdate DESC;

SELECT TOP 2 Eventname, EventDate FROM tblEvent

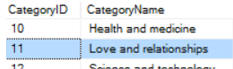
ORDER BY Eventdate ASC;

\*asc = de eerste en niet de recente.

DEEL2.

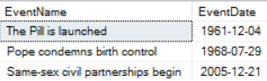
OPDRACHT 1.

In the **tblCategory** table, category number 11 is **Love and Relationships**:



*Category number 11 looks the most interesting!*

Write a query to list out all of the events from the **tblEvent**table in category number 11:



*Your query should return these 3 events.*

Bij deze opdracht was ik confused omdat je 2 verschillende tabellen had leek het wel.

Nummer 11 uit de tblCategory bevat 3 eventnamen. De bedoeling was dus dat je deze liet zien.

OPLOSSING:

SELECT Eventname, EventDate FROM tblEvent 🡨 als resultaat wil je de eventnaam en eventdate tezien krijgen

WHERE CategoryID = 11 🡨 waarvan (voorwaarde) de categoryID (waar ze zijn ingepakt als het ware) 11 is.

----------------------------------------------------------------------------------

OPDRACHT 2.

First show a list of all events which might have something to do water.  The Wise Owl interpretation of this is that one or more of the following is true:

* They take place in one of the island countries (8, 22, 30 and 35, corresponding to Japan, the Marshall Islands, Cuba and Sri Lanka respectively)
* Their **EventDetails**column contains the word **Water**(not the text **Water**, but the word)
* Their category is number 4 (**Natural World**)

This should return 11 rows.  Now add a criterion to show only those events which happened since 1970 (you may need to use parentheses to get this to give the correct answer).

In deze opdracht moet je een lijst laten zien met events die mogelijk iets te maken hebben met water, daarvoor moest de event een van de opties bevatten(or)

OPLOSSING:

SELECT \* FROM tblEvent🡨 Selecteer alles van tblEvent

WHERE (CountryID in (8, 22, 30, 35)🡨 Waar de countryID 8, 22, 30, 25 is

OR EventDetails LIKE '% water %' 🡨 OF die het woord water bevatten

OR CategoryID = 4) 🡨 OF waarvan de catergoryID 4 IS.

AND EventDate > '1970-01-01'; 🡨 EN(VEREISTE, Laat zien welke boven de datum van 1970 valt.)

\*Note, ik heb getest of als je de haakjes weghaalt bij de or statements of dan hetzelfde resultaat terugkomt, dit is dus niet zo.

Ik neem aan dat dit is omdat dit de or statements zijn en als de and er ook bij zit dan zou het dus: OR this maar moet wel voldoen aan dit. Zoiets.

----------------------------------------------------------------------------------

OPDRACHT 3

Je moest hier alle events die gebeurd zijn in de maand february laten zien.

Dit kan op twee manieren, mijn manier was:

SELECT Eventdate, Eventname FROM tblEvent

WHERE Eventdate > '2005-02-01'

AND EventDate < '2005-03-01';

De andere manier is:

SELECT Eventdate, Eventname FROM tblEvent

WHERE Eventdate between '2005-02-01' and '2005-02-28';

OPDRACHT 4. (LAATSTE OPDRACHT NIET VOLLEDIG UITGEKOMEN)

SELECT Eventname, Eventdetails FROM tblEvent

WHERE (CategoryID != 14

AND Eventdetails LIKE '%Train%');

SELECT EventName, EventDate FROM tblEvent

WHERE CountryID = 13

AND EventName NOT LIKE '%Space%' AND EventDetails NOT LIKE '%Space%';

SELECT Eventdetails FROM tblEvent

Where CategoryID= 5 OR CategoryID = 6

AND Eventdetails NOT LIKE '%Death%' OR Eventdetails NOT LIKE '%War%';

----------------------------------------------------------------------------------

OPDRACHT 4

Create a query which lists out all of the **tblEvent**events which include the word **Teletubbies**:

Teletubbies row

*You should find there's one event (which is probably one more than you were expecting).*

 Oplossing:

SELECT Eventname, EventDate FROM tblEvent

WHERE Eventname LIKE '%Teletubbies%';

\*note: je moet LIKE gebruiken en geen = als je doet aan pattern matching

Now add an **OR**condition to your query so that it lists out all events whose:

* Name includes **Teletubbies**; or
* Name includes **Pandy**.

This should give you two rows.

Oplossing:

SELECT Eventname, EventDate FROM tblEvent

WHERE Eventname LIKE '%Teletubbies%'

OR (Eventname LIKE '%teletubbies%'

OR Eventname LIKE '%Pandy%');

GEBRUIK MAKEN VAN CREATE TABLE 🡪 (SQL DDL COMMANDO’S)

Het DDL commando wordt gebruikt om een nieuwe tabel te creëren.

De tabel bestaat uit kolommen die de eigenschappen van de tabel beschrijft.

Deze kolommen hebben allemaal een eigen datatype.

Voorbeelden van deze datatypes zijn:

|  |  |
| --- | --- |
| **Datatype** | **Use** |
| INT | Gebruikt voor kolommen die cijfers(integers) willen opslaan. |
| FLOAT | Gebruikt voor kolommen die float gebruiken. Deze kan je gebruiken als je cijfers met decimalen wil toepassen bijvoorbeeld. |
| DOUBLE | Gebruikt voor kolommen die double waardes houden. Deze word meestal ook toegepast als je decimalen wil invoegen.  \*note; het verschil tussen een double en een float is is dat een double precieser is dan een float.. een double heeft 15 cijfers. |
| VARCHAR | Deze kan karakters en cijfers opslaan. Basically een string. |
| CHAR | Gebruikt voor kolommen die een karakter kan opslaan. Een letter of een cijfer. |
| DATE | Gebruikt voor kolommen die data kunnen opslaan. |
| TEXT | Gebruikt voor kolommen waarbij de tekst erg lang is. |

**UITLEG PRIMARY KEY EN FOREIGN KEY**

Je kan je tabellen niet aan elkaar linken (joinen) zonder een primary key en een foreign key te hebben.

Maar wat is een primary key en wat is een foreign key?

**Primary key:**

Je heb een tabel die bestaat uit kolommen.

De kolommen zeggen wat over de tabel.

Bijvoorbeeld:

Tabel; studenten

Kolommen; naam, studentnummer, klas, adres etc.

De primary key is de unieke rij in de tabel. De primary key is er namelijk bij elk object maar 1 van.

1 naam kan van meerdere personen zijn. Een studentnummer niet, deze is speciaal. Elk individu, maakt niet uit welke naam hij heeft, waar die woont en in welke klas die gene zit. Die gene beschikt over 1 uniek nummer. In dit geval is het studentennummer dus de primary key.

**Foreign key:**

het kan voorkomen dat je twee tabellen heb die allebei een zelfde kolom heb. Bijvoorbeeld:

tabel1; student

tabel2; school

kolommen tabel1; naam, studentnummer,klas,

kolommen tabel2; namen, plaats, klas, adres, studentnummers

stel je voor dat je door te zoeken op te naam uit tabel1 de school wil weten en de plaats/adres die erbij hoort. Dan heb je niet alleen tabel1 nodig van de student. Maar ook tabel2 school nodig.

Als je dit wil doen dan moet je ze dus joinen.

Een foreign key komt voor in de eerste tabel maar ook in de tweede. In dit geval bijvoorbeeld student nummer.

**INNER JOIN**

Als je eerst de kolommen aliassen wil geven dan moet je deze querie schrijven:

SELECT \*

FROM tblTest1 AS test1

Een SQL join clausule is onderdeel van de SQL query. Het voegt gegevens van twee of meer tabellen samen uit de database. Een join is dus een middel om de gegevens samen te voegen. Dit geld als ze gemeenschappelijke waarden hebben.

LEFT OUTER JOIN

Deze zal alleen waardes weergeven van de linkertabel. Als er geen overeenkomsten zijn met de rechter tabel dan zal de ontbrekende kant null weergeven.

RIGHT OUTER JOIN

Zelfde principe als left outer join maar dan met rechts.

Een full outer join geeft waardes terug van beide kanten. Als er geen overeenkomsten zijn dan geeft deze geen waardes terug.

Inner join kan je gebruiken om de kolommen van twee tabellen aan elkaar te linken.

Voorbeeld:

SELECT column\_name(s) 🡨 kolomnaam  
FROM table1 🡨 tabel waar de kolom in zit.  
INNER JOIN table2 🡨 tabel die gematched moet worden met tabel 1.ON table1.column\_name = table2.column\_name; de kolomnamen moeten overeenkomen. De kolomnaam is de naam die beiden tabellen bevatten.

Bijvoorbeeld doctor.id

**OPDRACHTEN INNER JOINS.**

Opdracht david en matt.

De opdracht is om alle doctors die in de episodes voorkwamen in 2010 te laten zien. Gegeven wat geshowed moet worden is: doctorname en title.

(naam van doktor en titel van de episode waar de doktor in voorkomt.)

\*note ; je hoeft de inner join niet te typen, je kan het namelijk ontwerpen 🡪 design query in editor.Je selecteerd eerst de tabellen en vervolgens klik je de kolommen aan die je uiteindelijk wil showen (select dus)

SELECT tblEpisode.Title AS titel, tblDoctor.DoctorName AS naam

FROM tblEpisode

INNER JOIN tblDoctor ON tblDoctor.DoctorId = tblEpisode.DoctorId

WHERE EpisodeDate BETWEEN '2010-01-01' AND '2010-12-31'

ORDER BY titel

SELECT 🡪 in dit geval selecteer je de titel uit de kolom episode en de naam van de dokter uit de tabel dokter. Ook geef je ze aliassen. Dit doe je meestal als de orginele namen te lang zijn. Deze aliassen kan je benoemen na de AS.

\*note; tabelnaam.kolom wat je wil selecteren.

FROM 🡪 in dit geval maakt het niet uit welke je eerst wil selecteren. Als je 2 kolommen heb kan je een van deze achter from zetten.

INNER JOIN 🡪 de andere tabel die overblijft die wil je ‘joinen’ met de ander.

ON 🡪 hier selecteer je de twee tabellen met achter de punt de overeenkomende kolom.

Bijvoorbeeld: from tbl.episode

Inner join tbl.doctor on tblEpisode.DoctorID = tblDoctor.DoctorID

Omdat de twee tabellen geconnect zijn met elkaar kan je bijvoorbeeld resultaten sorteren op een van de kolommen. Bijvoorbeeld de datum. Zoals je heb gezien in het voorbeeld.

WHERE EpisodeDate BETWEEN ‘2010-01-01’ AND ‘2010-12-31’ 🡨 selecteert dus alle episode datums in 2010

ORDER BY Title 🡨 samen met de vorige zin maakt dit het dus dat alle titels worden geselecteerd die in 2010 voorkomen.

**OPDRACHT BRR**

\*note ; je hoeft de inner join niet te typen, je kan het namelijk ontwerpen 🡪 design query in editor. Je selecteerd eerst de tabellen en vervolgens klik je de kolommen aan die je uiteindelijk wil showen (select dus)

-Query creeren waarin 2 tabellen met elkaar worden gelinkt.

- de query moet de events laten zien die op het continent antartica of de country russia zijn geweest

USE WorldEvents

SELECT tblEvent.EventDate, tblEvent.EventName, tblCountry.CountryName, tblContinent.ContinentName

FROM tblContinent

INNER JOIN tblCountry ON tblContinent.ContinentID = tblCountry.ContinentID

INNER JOIN tblEvent ON tblCountry.CountryID = tblEvent.CountryID

WHERE ContinentName = 'Antarctic' OR CountryName = 'RUSSIA'

Wat ik hier heb gedaan is als volgt:

USE Worldevents 🡪 zodat sql weet dat hij de querie moet linken met de database worldevents.

SELECT 🡪 hier staan de dingen die geshowed moeten worden. Namelijk in dit geval: de eventdatum, eventnaam, countrynaam en continentnaam.

Vervolgens is het de bedoeling dat je de 3 tabellen gaat linken met elkaar.

* D.m.v 2 inner joins.
* Het is een driehoek verhouding
* Het kan namelijk wel zo zijn dat de ene tabel een primary key heeft die ook in de andere tabel zit (als foreign key) maar niet in de derde tabel.
* In de tweede tabel die de foreign key van de eerste tabel bevat zit namelijk ook een primary key en die zit dan ook in de derde tabel als foreign key.

\*note: als je de inner join moet veranderen naar een outer join dan gaat dit zo: FULL OUTER JOIN

USE WorldEvents

SELECT tblCategory.CategoryName AS Naam, tblEvent.EventName AS EventN, tblEvent.EventDate AS EventD

FROM tblCategory

FULL OUTER JOIN tblEvent ON tblCategory.CategoryID = tblEvent.CategoryID

WHERE EventName is NULL

Met deze laat je de eventnamen zien die geen waardes hebben.

**OPDRACHT ENOUGH JOINTS FOR NOW:**

Write a query using inner joins to show all of the authors who have written episodes featuring the Daleks.

Schrijf een query waarin je inner joints gebruikt om alle auteurs te displayen die episodes geschreven hebben waar de daleks in mee spelen.

Dus je had drie dingen die je moet displayen:

* Naam auteurs
* Episode naam
* Enemy naam

Als je joint moet je op de volgende dingen letten:

* De tabellen die je displayed hebben allemaal een primary key. Als deze key voor komt in een van de andere tabellen waarvan jij een kolom wil laten zien dan moet je deze linken.
* Als jij 3 tabellen heb waarvan 1 primary key niet matched met een de rest van de tabellen dan innerjoin je een nieuwe tabel die wel deze foreignkey bevat.

*Zie onderstaande voorbeeld:*

SELECT tblEnemy.EnemyName, tblEpisode.Title, tblAuthor.AuthorName 🡨 je hebt hier 3 tabellen geselecteerd waarvan je gegevens uit kolommen wil weergeven

FROM tblEnemy

INNER JOIN tblEpisodeEnemy ON tblEnemy.EnemyId = tblEpisodeEnemy.EnemyId

INNER JOIN tblEpisode ON tblEpisodeEnemy.EpisodeId = tblEpisode.EpisodeId

INNER JOIN tblAuthor ON tblEpisode.AuthorId = tblAuthor.AuthorId

WHERE EnemyName LIKE '%Daleks%'

ORDER BY EpisodeDate

Je zou in dit geval author kunnen linken met episode door authorid

Maar tabel enemy kan je met geen van beiden linken. Hierom moet je dus een andere tabel laten inner joinen die wel de foreign key heeft. In dit geval is dat episodeEnemy.

Deze kan linken met enemy door de overeenkomende enemyID. En episodeEnemy kan je weer linken met Episode door de overeenkomende EpisodeID. En omdat je dus de episodeEnemy hebt kan je in dit geval, deze 4 tabellen linken.

Calculations with dates.

Opdracht:

Creer een querie die de events showed uit het jaar dat jij geboren bent.

Als je met datums gaat werken kan je gebruik maken van format of convert. Het geeft hetzelfde resultaat.

Dit houd in:

Je formatteerd de datum naar de volgorde die jij wilt

Bijvoorbeeld:

FORMAT(Eventdate, ‘dd/mm/yyyy’) AS UsingFormat

Uitkomst is

Usingformat = 02/09/2020

OF je convert de datum naar een varchar(deze kan nummers en cijfers vasthouden), Naamvankolom, stylenumber(in dit geval 103 want dit staat voor dd/mm/yyyy maar je kan ook een andere gewenste volgorde doen)

Voorbeeld:

CONVERT(varchar, EventDate, 103) AS UsingConvert

Stylenumbers:

|  |  |
| --- | --- |
| **Style** | **How it's displayed** |
| 101 | mm/dd/yyyy |
| 102 | yyyy.mm.dd |
| **103** | dd/mm/yyyy |
| 104 | dd.mm.yyyy |

Opdracht 2:

Het idee van deze opdracht is om alle events te showen die gebeurd zijn rond jou geboortedag.

Eerst moet je een querie maken die de verstreken dagen telt van de events die zijn gebeurd sinds jou geboortedag.

DATEDIFF geeft het verschil in dagen terug tussen twee datums.

DATEDIFF(interval, date1, date2)🡪 interval staat voor het gene wat je wil dat hij terug wil geven. Dagen, maanden of jaren.

Je kan vervolgens ABS gebruiken om van de negatieve getallen positieven te maken voor als je in de toekomst de getallen wil sorteren.

SELECT Abs(-243.5) AS AbsNum;

In dit geval word het:

DATEDIFF(DAY, EventDate, '2001/02/08') AS DaysOffset,

ABS(DATEDIFF(DAY, EventDate, '2001/02/08')) AS DaysDiffrence

Opdracht 3:

Schrijf een querie die de dag van de week plus het weeknummer toont.

De dag van de week kan je eruit halen met:

DATENAME ( datepart , date ) (de datepart is het gedeelte wat je terug wil krijgen en de datum is in dit geval de datum van de EventDate.)

DATENAME(WEEKDAY, EventDate) AS 'Day of week',

DAY(EventDate) AS 'Day number' (Hier haal je het nummer van de dag uit de EventDate)

QUERIE SUFFICES.

Doel van de opdracht: Volledige datum showen van elk event.

Omdat er verschil zit in de achtervoegsels van de dagen moeten we eerst iets anders fixen.

Bijvoorbeeld:

In het engels is eerste: 1st, met st erachter

Maar second (2th) met th erachter

Om dus te voorkomen dat het verkeerde achtervoegsel word gebruikt maken we gebruik van een case. Dit lijkt op de if statements en de switch statements in bijvoorbeeld c#

CASE

WHEN

Met de when wil je eerst een voorwaarde plaatsen. In dit geval is het wanneer de dagnummer van de Eventdate 1 is dan moet hij uiteindelijk dit laten zien:

THEN

Met behulp van CAST kan je een waarde converten naar een datatype naar keuze.

In dit geval willen we het in een varchar hebben want deze kan letters en cijfers waardes vasthouden.

Dan vervolgens wat tussen haakjes staat luid als volgt: DATENAME(weekday) 🡪 is de functie om de naam van de dag uit de week te krijgen. De volgende is de functie DAY en deze haalt het dagnummer uit de maand en zit in dit geval als het dus 1 is ST erachter. En dan nog het jaar uit de datum er krijgen en dan ben je klaar. En dit alles doet hij precies hetzelfde met alle waardes boven 1 alleen zet hij er dan th achter.

Vervolgens sluit je de case af met END en als alias benoem je je gemaakte datums met full date

Order by eventdate