Video 1: historia SQL : teoria de conjuntos y teoría relacional nace en los años 70 con IBM Sequel

Inicia system 38 1979

Sql ds 1081

Db2 1983

Relational company es oracle. Creo oralcol 1979, se convierte en estándar con la iso 2016 la ultima

Video 2:

Algebra relacional, es el conjunto de datos se refiere a una tabla no esta permitido una tupla, no hay un ultimo registro, no tiene un cuerpo, esta son las diferencia entre tabla.

Las operaciones a utilizas

Operadores unarios: solo requieren una relación o trabla para funcionar jemplo proyecion o selec

Comando where filtra el numero de tuplas que cumpla cierta condiciones.

Operadores binarias: necestina dos tablas diferentes,

Producto cartesiano es igual a una X, toma todos los elementos de cada tabla.

La unión: simboliza que se tendrán los elementos de una u otra tabla, siempre y cuando este en alguna de las dos cuenta como unión U.

La diferencia - : los elementos que están en una tabla pero no en la otra.

Video 3 instalar la BD de ejemplo

Video 4 la proyección es igual al comando select, que queremos que se muestre en las columnas

El select sirve para filtrar por columnas

El where sirve para filtrar por filas o registros o tuplas o renglones o rouws

Selec \*; sirve para seleccionar todas las columnas , no esta filtrando nada. Proyecta de manera visual. Sirve para que columnas queremos obtener de una tabla determinada

Ejemplos:

SELECT field AS alias; AS significa como

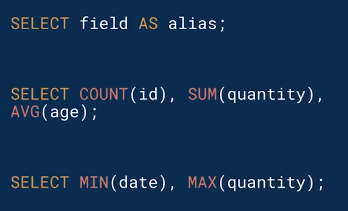
Selectr nombre de la columnas AS sirve para agregar un alias a la columna

COUNT: cuenta el número de elementos que existen en todos los datos

SUM: suma los valores de la columna

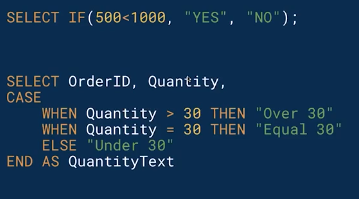
AVG : saca el valor promedio de toda la columna

MIN Y MAX sirve para saber el minio y el máximo de un grupo de elementos.



Estructuras de control

IF significa si se cumple esta condición haga esto si no haga esto



Cuando 500 sea menor a 1000. Ponga el letrero yes sino ponga letrero NO

CASE WHEN es cuando se tiene más dos opciones

EN EL CASO CUANDO LA COLUMNA CANTIDAD (QUANTITY) SEA MAYOR A 30 ENTONCES ESCRIBA POR ENCIMA DE 30

EN EL CASO CUANDO LA COLUMNA CANTIDAD (QUANTITY) SEA IGUAL 30 ENTONCES ESCRIBA POR IGUAL A 30

ELSE: por defaul si no escriba por debajo de 30.

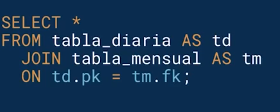
END AS qunatite text sirve para luego filtrar y/o ordenar

Video 5: FROM ORIGEN: DESDE O DE DONDE SE LLEGA LA INFORMACION

SELECT \* ( de que tabla es la fuente de datos).

FROM tabla\_diaria;

AS sirve para ponerle un alias a la ejemplo de tabla diara AS td



JOIN indica dentro del from, quiero los datos de la tabla que están relacionados , las relaciones con las tablas, join permite relacionar

Pk llave principal identificar único

Fk llave foránea o referencia a un id de la primera tabla

Querio los datos de la tabla diaria o td y quiero los datos que están en la tabla mensual o td.

En tabla diaria use la pk que va ser igual a la fk de la tabla mensual



DBLINK es una función para obtener los datos de una base que no está aquí que no está en local, y me permite conectarme a esa tabla.

Video 6: JOIN o Productos cartesianos: es una serie de elementos de un lado y multiplicar los elementos del otro lado (todos con todos) es una tabla caótica no es lo mas indicado.

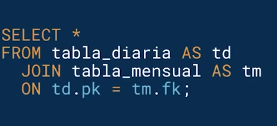
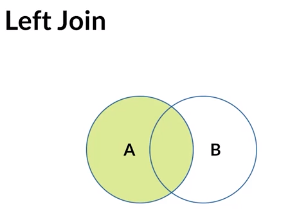
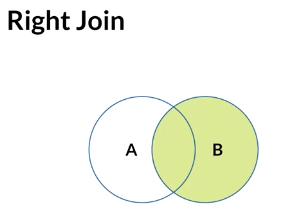


Diagrama de BEN



Left join sirve es para traer los elementos de la tabla uno junto con los elementos de la segunda tabla, pero no es importante que tengan su equivalente en la tabla A. En el ejemplo tabla A

Right join

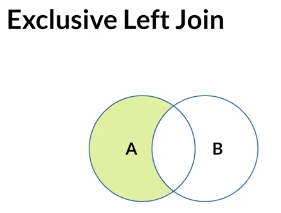


Traer los elementos de la tabal b sin importar si tiene equivalentes en la tabla a

EXCLUSIVE LEFT JOIN

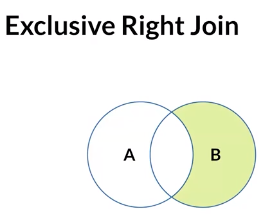
La clave es que estén seguro que no estén en la otra tabla

Trer todos los elementos de la tabla A , y asegurarme que no estén en la tabla B



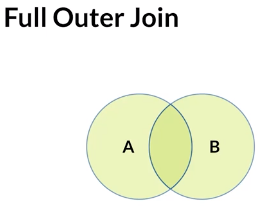
EXCLUSIVE RIGHT JOIN:

Tráigame todos los elementos de la tabla B , pero seguros que no estén en la tabla A



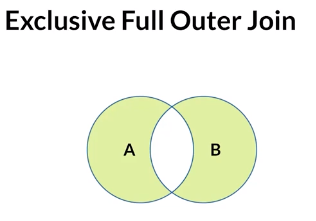
FULL OUTER JOIN

Traerme todo de la tabla A y multiplicame cada elemento de la tabla B es un caos completo.

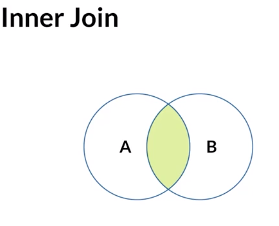


EXCLUSIVE FULL OUTER JOIN

Traer lo que existen en A y B pero no lo que tienen en común



INNER JOIN es el más común solamente lo que tenemos en común en ambas tablas.



Video 7: WHERE selección

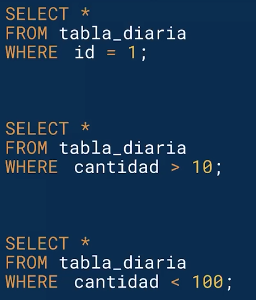
Where básico

Where id = 1

El id o llave principal es igual a 1 , las tuplas igual a 1 en su id

Vamos a seleccionar tuplas, o renglones o gros, o filas

Cuando el id es igual 1, me trae el renglón

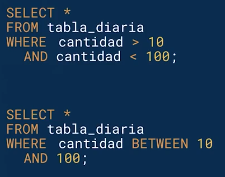


Where cantidad >10;

Todas que en la columna cantidad sea mayor a 10 nos filtra y nos trae.

Where cantidad <100; nos traerá menor a 100 nos filtra y trae.

Where donde la columna cantidad es >10 y menor a 100 las filtra y las presenta.



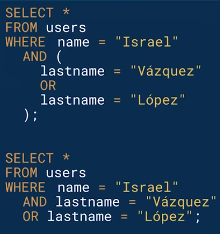
Where cantidad between 10 y 100;

rangos ,, en medio o entre datos entre 10 y 100

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

Or si se cumple una o la otra. Tráeme cierta información

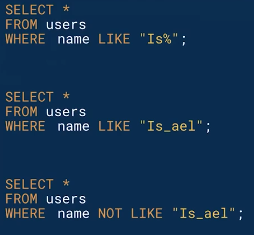
De la tabla usuarios traerme todos los llamados Israel de apellido Vásquez o López. Ojo con paréntesis.



De la tabla usuarios cuyo nombre sea Israel y apellido vasquez o su apelllido lopez, y te trea todos los Israel vasquez y luego todos los que se apellidan lopez

OPERADOR LIKE: parecedio o similar,

Traeme donde el nombre es parecido donde empiece con la palabra IS % significa y el resto de texto no me importa.



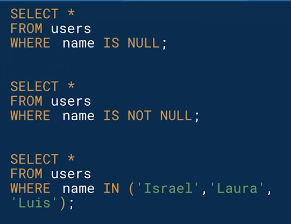
Is\_ ael: significa tráeme cual palabra que cumpla el inicio con IS y termine con AEL. Eso significa el guion bajo \_. Es el comodin.

NOT LIKE, tráeme todo las letreas que no se parezcan a IS\_AEL

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

NULL significa que el dato no está, que esta vacío.

IN: cuya columna nombre búscame los nombre que están en el (paréntesis).



Resumen

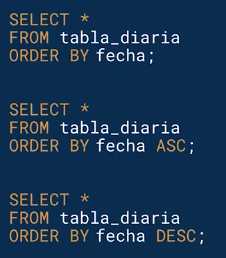
Proyección con selec

Origen con from y join

Where filtrar tuplas

Video 8: ORDENAMIENTO: order by (ordena por) criterios

Primer ejemplo organiza de menor a mayor con fechas o números o letras



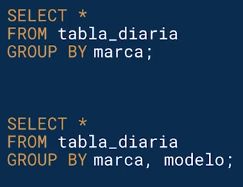
Con orden BY …..ASC: sirve para que los datos sean filtrados ascendentes

Con ORDER BY …….DESC: sirve para que filtre descendente de arriaba para abajo.

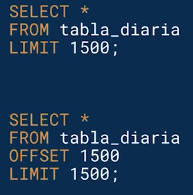
LOS INDICES: Sirve para búsquedas y ordenamientos sirve para hacer las extracciones muy rápidas

La escritura se queda lento, ya que cada elemento se debe reclasificar. Se llama indexar. Tener menos índices es mejor ya que lo deja lento, si se escribe la tabla mucho, no vale la pena realizar índices.

Video 9: COMANDOS AGREGACION GROUP BY: Esto es reducir los datos en tuplas sirve para estadísticas y sacar gráficas, sirve para agrupar o agrúpame por cierto campo. Va agrupando en varias bolsitas ciertos artículos ejemplo bolsa de Apple, bolsa ce hp, bolsa de hacer. Esto es el group by



LIMITANTES O LIMIT: significa limita mi consulta y sirve solo con números y solo tráeme ejemplo las 5 computadoras mas caras.. y con limit 5; así me trae las primeras 5 computadoras.

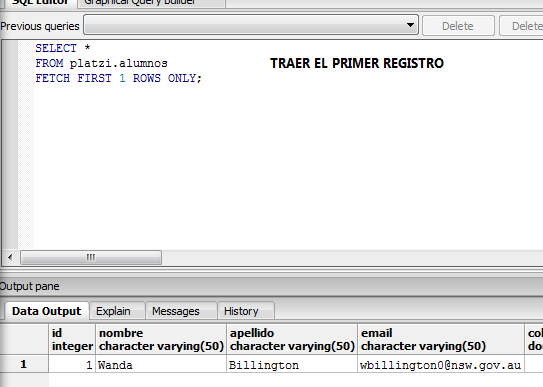


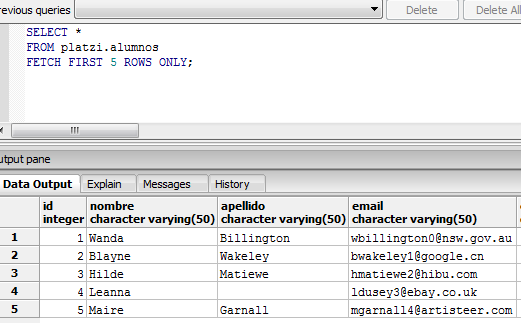
OFFSET quiere decir , de estos 1500, DESPUES tráigame los siguientes 1500.

Deseo saber traer los números del 10 al 15 ofset 10 limit 5

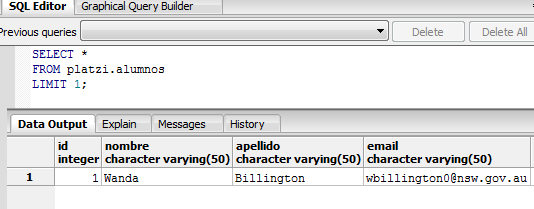
VIDEO 10:

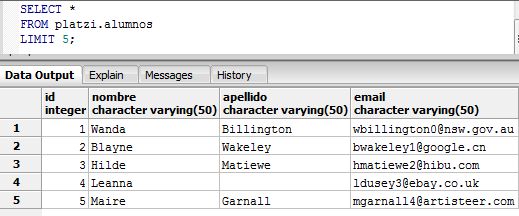
1. FETCH significa tráeme : EL PRIMER EJERCICIO Y EL PRIMER REGISTRO





2. LIMIT TAMBIEN SE PUEDE





1. ROW\_ NUMBER = FUNCIONES DE VENTANAS: sirve para que me traiga el número de registro Independientemente de cualquier criterio y le indico en que datos lo quiero medir

SELECT sirve para decile extráigame, extraer el siguiente dato

OVER () significa de toda la tabla

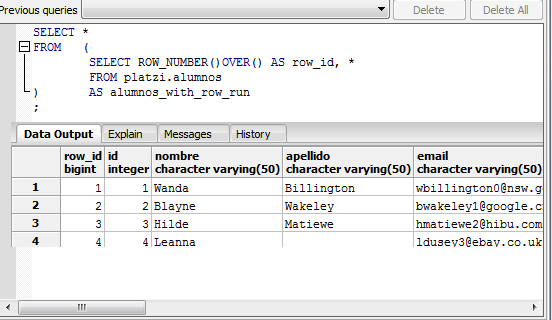
AS el alias se llamara row\_id (al principio) \* significa todo el resto de la tabla

FROM: de que tabla es que se saca la información

AS: esta consulta se llamará alumnos\_with\_row\_num

ROW significa registro

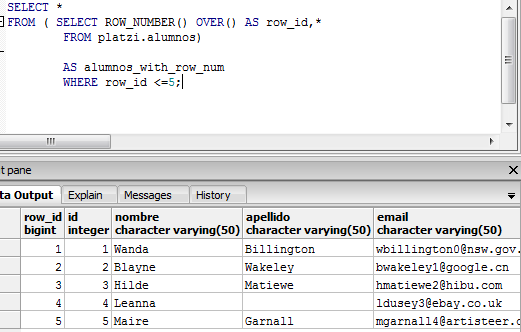
Si no se especifica nada trae toda la tabla

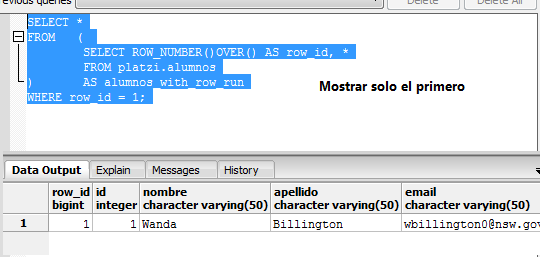


ROW\_ NUMBER es la función de ventana

WHERE row\_id = 1; sirve para mostrar solo el primer

SEGÚN EL NUMERO DE REGISTRO row\_ id me trae el registro solicitado ejemplo 10 me trae el registro 10





4. DISTINCT me trae solo una vez el registro

Video 12

5. SELECT ARRAYS (LISTAS):

Video 13

EXTRAC (YEAR FROM) Y/O (DATE\_PART) : sirve para extraer los campos fecha y hora

Video 14

REALIZAR FILTRADOS

Video 15

DOUBLE TROUBLE: encontrar los duplicados

Video 16

SELECTORES DE RANGOS:

Video 17

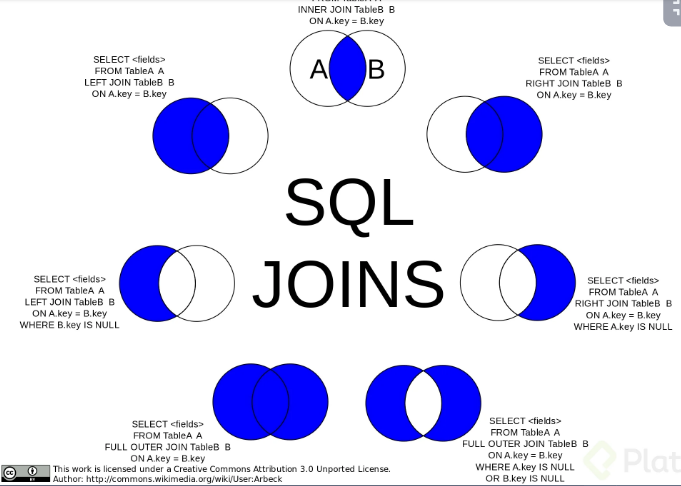
Como sacar los MINIMOS Y MAXIMOS

Video 18

SELFISH sirve para hacer un JOIN con la propia tabla

Video 19: LAS DIFERENCIAS: son necesarias para extraer cierto tipo de datos, Son los elementos que se encuentran en una tabla pero que no se encuentran en otra.

Video 20: TODAS LAS UNIONES O TODOS LOS JOINS:



1. LEF JOIN EXCLUSIVO: para ser exclusivo se debe anular el id de una tabla, para este ejemplo anulamos el id de carrera

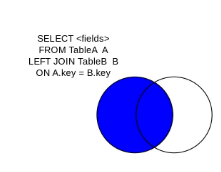
ON nos indica porque campo podemos unirlas las tablas, para este caso se realizara por carrera id tabla alumnos debe ser igual al campo id de la tabla carreras

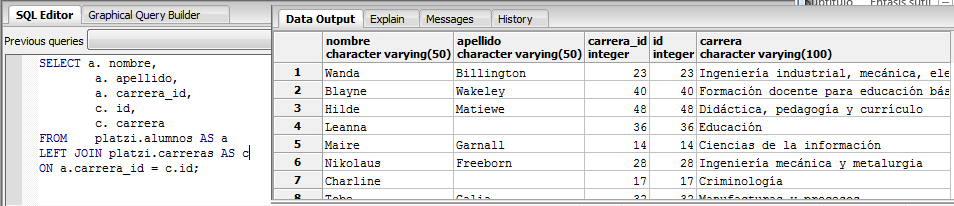


Nos muestra a los alumnos que cuentan con una carrera que ya no existe en la tabla carreras.

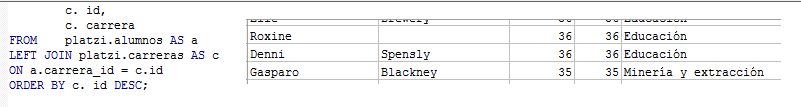
1. LEF JOIN INCLUSIVO o NORMAL:

Unimos la tabla platzi alumnos que es la tabla de la izquierda con la tabla carrera de la derecha a través de un LEF JOIN. De esta manera trae todas las carreras.





ORDENADO DECENDENTE

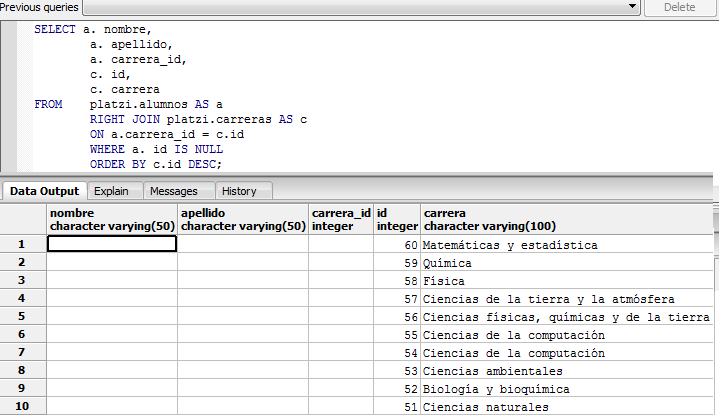


1. RIGHT JOIN: se cambia la palabra left por right y lo convierte, se traen las carreras sin importar un alumno si o no. Se traen todas las carreras con o sin alumno.

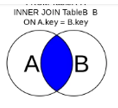
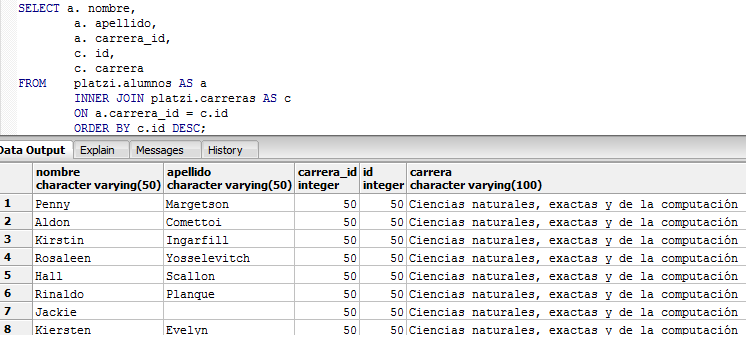


1. RIGHT JOIN EXCLUSIVO: se debe agregar la cláusula WHERE donde el id a sea null

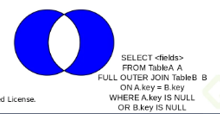
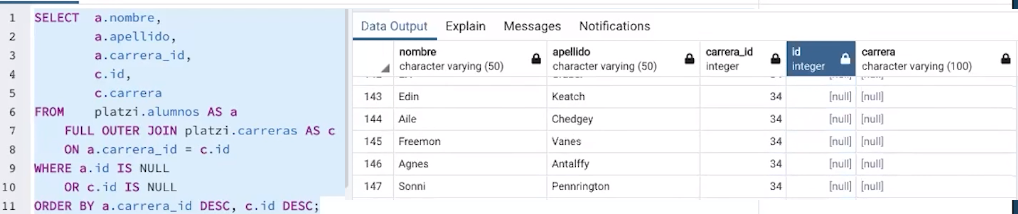
Es traer las carreras que no tiene alumnos, que existen a la derecha



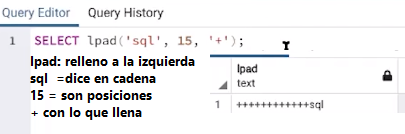
1. INNER JOIN: Es cuando no se especifica el JOIN, sirve para identificar la parte en común de ambas tablas. Pertenecen en las dos tablas. Es el mas común.



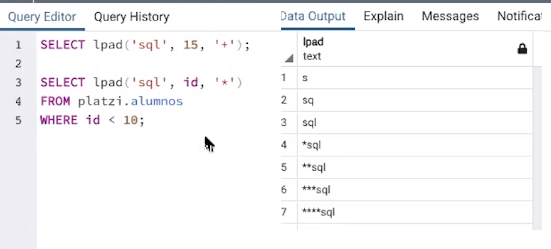
1. FULL OUTER JOIN: DIFERENCIA SIMETRICA: es la que sirve para identificar las que se encuentran A o se encuentra B pero no se encuentra en ambas.

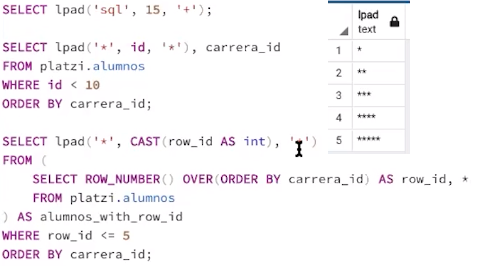
 

Video 21 TRIANGULANDO = METODO (lpad): sirve para hacer cadenas con longitudes que necesitemos muchas veces podemos llenarlas con espacios o guiones etc.

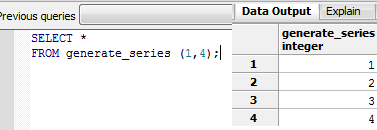


Triangulo creado con ID. Llega hasta 2.000 di

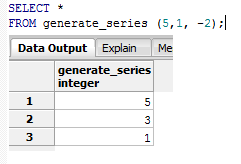




Video 23 GENERAR RANGOS: rangos básicos:

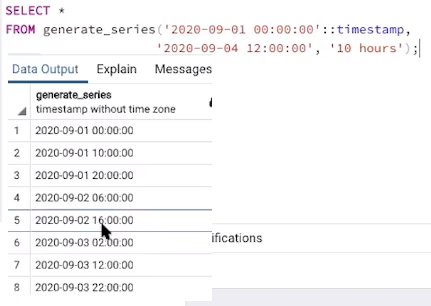
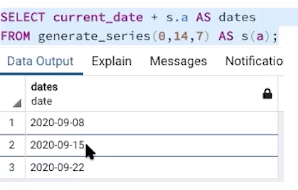


Rangos con delta o un paso (-2)

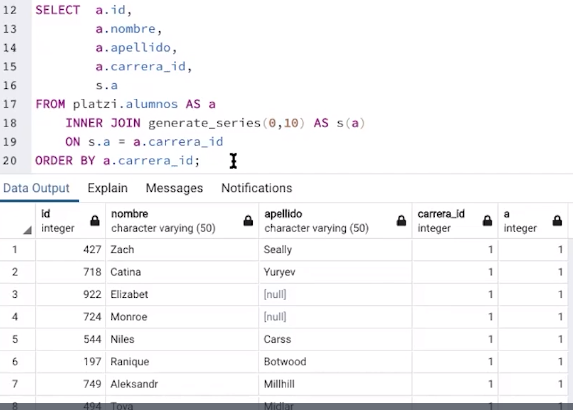


NOTA: cuando el primero es mayor que el segundo (5,1), genera un rango vacío, siempre se necesita un delta o un paso, que significa de cuanto en cuanto debe saltar.

RANGOS CON FECHAS



RANGOS CON DATOS: realizar rango para todos los carrera id entre el 1 y el 10.



Video 23: EXPRESIONES REGULARES: son forma contraída o resumida para hacer una validación de patrones compleja, filtra lo que se encuentre o no.

MAIL: son expresiones que permite filtrar los correos de mail

Video 24: BASE DE DATOS DISTRIBUIDAS: es una colección de una base de datos o una base de datos partida y distribuida geográficamente, pero conectada.

Ventajas. Desarrollo modular destinada para dividir la información y según los usuarios de México se guarda en México y mejora el tiempo de respuesta.

Desventajas : blindado para todos los sitios geográficos, si se trae la información por lo lejos es mas caro y mas demrado

HOMOGENEAS : sistemas operativos iguale en todo el mundo

HETEROGENEAS:

ARQUITECTURAS: cliente servidor

Par a par

Multimanejador de baso de datos diferentes base de datos

TOP dow se crea la base de datos desde cero

Botom up es construir la base de datos con base ya construidas

Fragmentación horizontal

Vertical

Mixta

REPILCACION

Completa

Parcial

Sin replicación sin sincronización

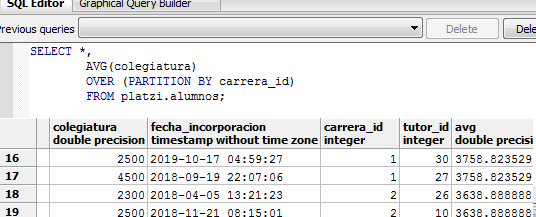
Video 25 QUERIES DISTRIBUIDOS

Video 26 SHARDING sirve para partir la base de datos es una pizza es una base de datos, se parte la data en ejemplo 8 pedazos sharding es un pedazo por geográfica

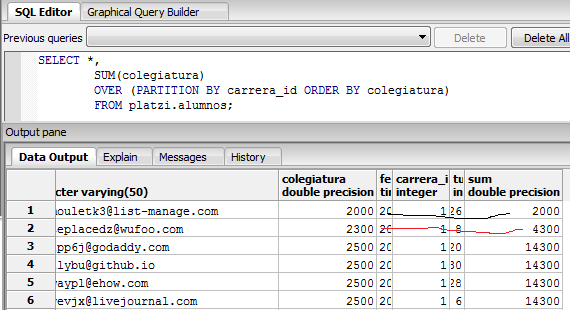
Videos26 WINDOS FUNTION: Es la relación entre una tupla o registro y el resto de registro en una sección de la tabla. Es la relación entre un registro y el resto de registro en una partición.

Sirven para ahorrar SELF JOIN (hacer una tabla unidad con ella misma).

Ejemplo: sacar el promedio de precio de la tabla alumnos partiendo de la columna carrera\_id. Esto sirve para saber el precio promedio por carrera ejemplo: carrera 1 promedio 3.758, carrera 2 3.638.

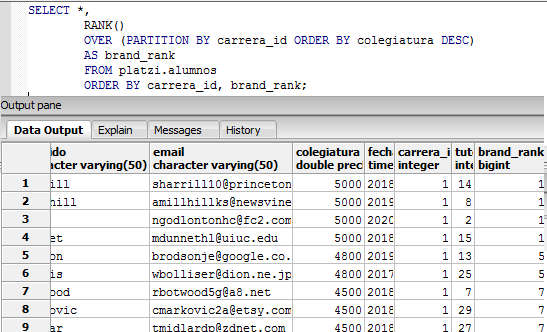


Ejemplo: sumar precio matricula, organizado por la columna carrera\_id y ordenado por precio matricula. Colegiatura es precio matricula.



RANKING: que lugar ocupa algo en la tabla. Organiza por carrera el precio mas alto y le colocar el numero dependiendo la posición llamada la columna Brand\_rank

Ejemplo: El precio numero uno corresponde a $5.000



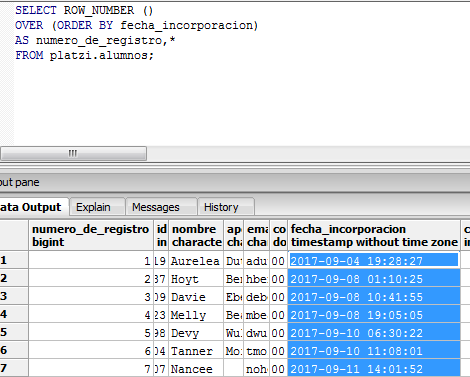
Video 27 PRINCIPALES WINDOW FUNCTIONS

1. ROW\_NUMBER: significa número del registro y quiere decir tráigame todos los registros.

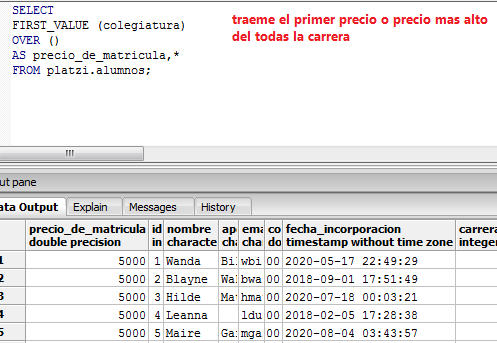
OVER () vació significa sin ninguna orden. (ORDER BY) significa ordenar por o filtrar por. (allí se le debe escribir por cual columna desea filtrar ejemplo por fecha de incorporación) (PARTITION BY significa agrupeme por )

AS significa el alias o nombre de la columna que estamos creando

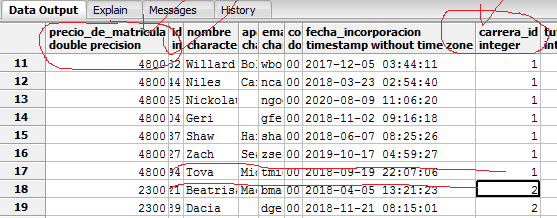
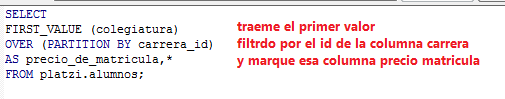
“\*” es \* al final significa añádale los datos del resto de la tabla.



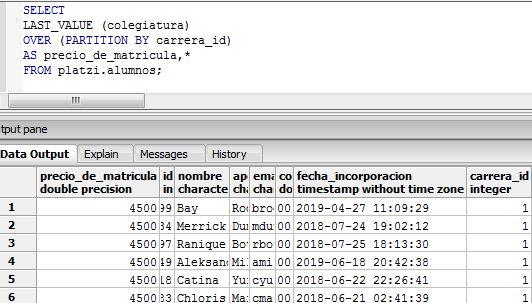
1. FIRST\_VALUE: Primer valor de una serie de datos



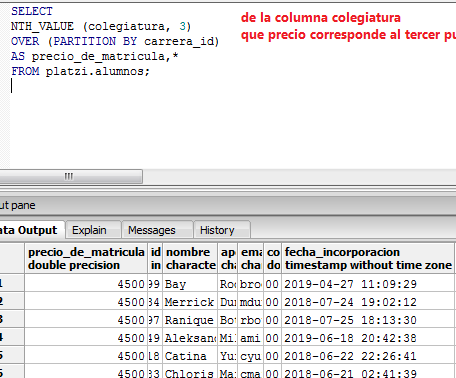
Ejemplo 2.



1. LAST\_VALUE Traer el ultimo valor

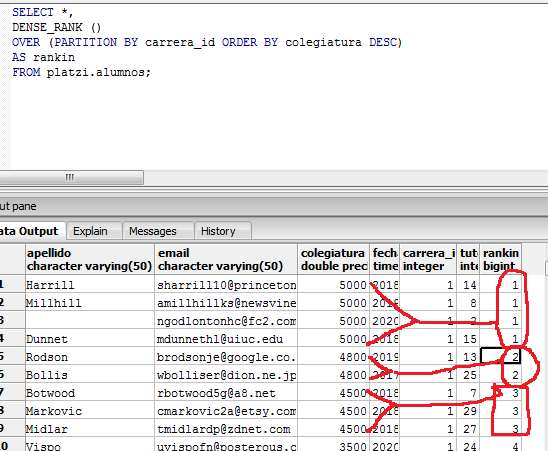


1. NTH\_VALUE: sirve para decirle que puesto debe traer, en este caso corresponde al tercer puesto.



1. DENSE\_RANK. Significa rango, es el lugar que ocupa de una competencia.

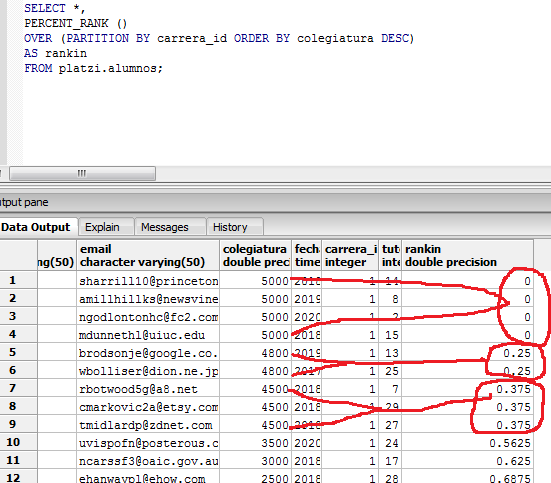
Con este código le decimos crea un columna organizada por la columna carrera y filtrada por el precio de matrícula de mayor a menor.



1. PERCENT\_RANK: sirve para organizar por porcentaje.

Qué lugar ocupa organizado por porcentaje.

La fórmula es (Rank -1) / (total rows -1)



Video 29 Futuro de SQL

Examen

Nombre de Sequel a squl derechos de autor

Unario proyección

Agregación group by

Join cartesiano full outer join

Order by ascendenteiang

Subqueries from falso

Date\_part falso

Operador :: cast

Operador \* los elementos en común entre los dos

Feljoin limitante group by falso

Simétrica contiene los elementos que no pertenecen a ambas tablas

Row-number por que es idependiente del orden de los datos

Expresiones regulares filtrar verdadero

Venta DB distribuidas desarrollo modular

Desventaja shardng baja elasticidad

Reducir window functions self join

Denser rank verdadero

Fetch firs 15 row limit 15 verdadero

Query distribuido de red se considera despreciable falso

Tabal remota dblink

Rango dinamico subquery

Max y limit / max sirve para subsets

convierte lef join where exclusive left join

Valor inicial valor final delta

ERROR

Comodin sustituir caracteres error \*? % \_

Limit 1 offset 105 error 105 104 111

Date parte error select y limit / select - where y limit



Video

ENTIDADES es algo parecido a un objeto es la representación de una caso de la vida real

ATRIBUTOS: son cosas que tienen las entidades

Doble circulo

Atributos compuestos de varios atributos

Los tributos

ENTIDADES = laptos

Atributos: color año modelo n° de serie disco duro método de entrada antigüedad pantalla



línea punteada es la llave: son propias no se pueden retirar

ENTIDAD FUERTE:

ENTIDAD DEBIL: son por identidad no se diferencia entre si solo por la clave de la fuerte. Son dependientes de la tabla fuerte por causa del id campo de identificación

Son débiles por existencia, se les agrega un id propio , sigue dependiendo de la fuerte. Ejmplo libro es fuerte ejemplar o copia del libro es débil

Video 4

DIAGRAMA ER: platziblog

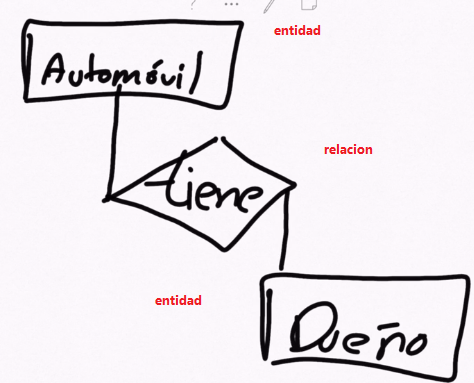
**ENTIDADES** son objetos del mundo real ejemplo identificados usuarios, comentarios, post, categorías.

**ATRIBUTOS:** es lo que define el objeto. Ejemplo posts. (títulos, fecha publicación, contendio, estatus, publicado o no, activo o no. Etiquetas es la manera de clasificar ejmplo ciencia, tecnoloiga, y se representan con doble circulo.

El atributo clave es el identificador ID.

Usuarios, login, pasword, apodo, email, ID(debe ser clave artificial)

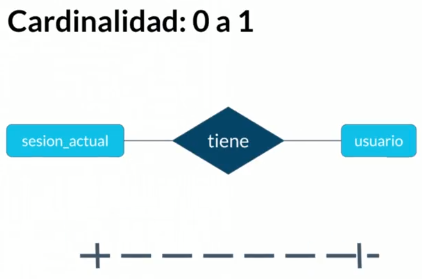
RELACIONES se representa con un rombo,



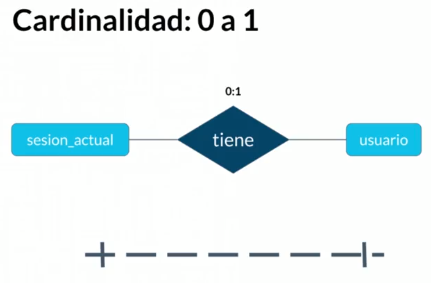
NOTA: cuando un atributo es multivaluado, eso quiere decir que tiene más de un atributo, y se convierte en una ENTIDAD SEPARADA.

CARDINALIAD: tiene que ver con números,

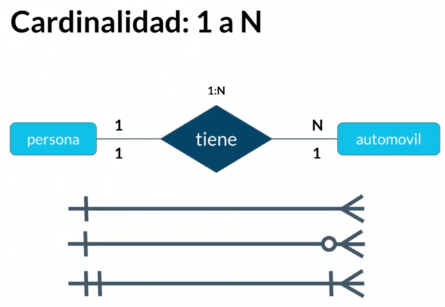
CARDINALIDAD 1 A 1: de un lado tengo un dato y del otro tengo solo uno también.



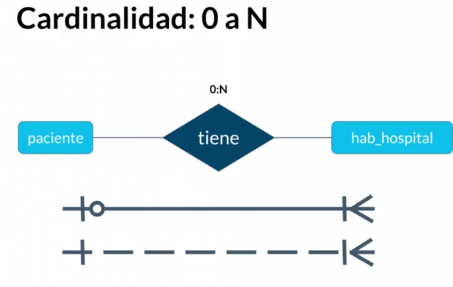
CARDINALIDAD 0 A 1 es llamada 1 a 1 opcional, puede existir que no haya nadie de un lado, tiene al menos uno.



CARDINALIDAD 1 A N (MUCHOS) en un extremo tenemos 1 persona, tiene muchos carros.



CARDINALIDAD 0 a muchos



CARDINALIDAD MUCHOS A MUCHOS

1 alumno pude tomar varias clases. O 1 clase tiene a varios alumnos.

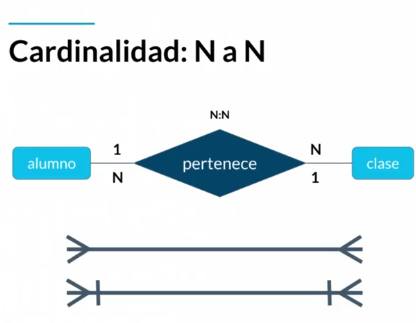
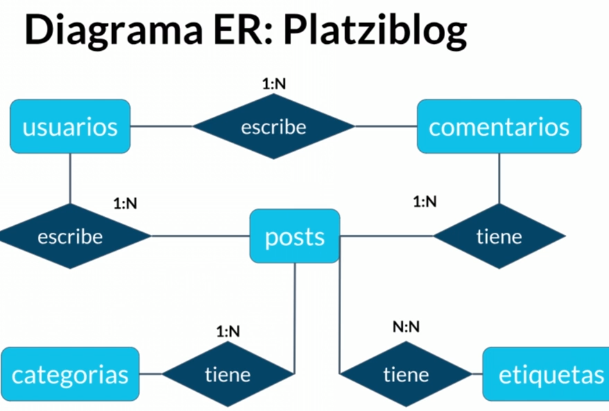


DIAGRAMA ENTIDAD RELACION es un mapa, debe ir antes de la creación de la base de datos.

TIPOS DE DATOS:

TEXTO char, permite cadenas caracteres, si son cadena de 8 caracteres y no se cambia

VARCHAR : permite cadenas caracteres, es dinamico reserva un mino espacio de memoria, se alarga y se encoge. LIMITE 255 caracteres.

TEXT: son cadenas superiores a 256 superior ejemplo 500 caracteres.

NUMERICO

INTEGER o ENTERO, numero sin punto decimal, sin fracción.

BIGINT: declarar número muy grandes

SMALLINT declara números de 99 y menos

DECIMAL / NUMERIC entradas primero entra el entero y luego el decimal

FECHA / HORA

DATE: año mes y día

TIME: hora del día

DATETIME: / TIMESTAMP: día y la hora para tener tiempo precios

LOGICO

BOOLEAN solo puede tener dos valores cierto o falso o 1 o 2.

CONSTRAINST (RESTRICCIONES)

NOT NULL: en esta columna no permitimos que este en blanco.

UNIQUE: es único, solo existe con ese valor en la table, ejemplo email,

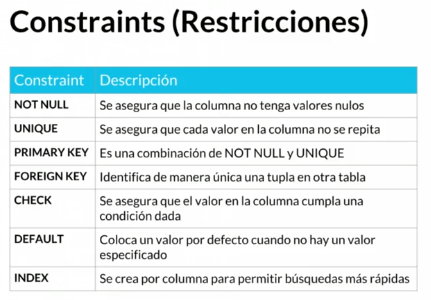
PRIMARY KEY: llave primaria, campo clave o campo llave, es not null y es unique, sirve para hacer la unión entre una tabla y se llama índice, sirve para hacer las relaciones.

FOREING KEY: llave foranea, cuando queremos juntar dos tablas la primary key de una tabla se añade como foreing key con la otra tabla. Es foránea por que viene de afuera.

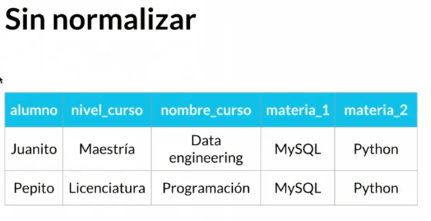
CHECK: permite definer las reglas que necesitamos ejemplo. solo deja entrar registro activos o inactivos.

DEFAULT: colocar un valor por defecto cuando no hay un valor especifico.

INDEX o INDICE: permite hacer la busqueda mas rapida en la base de datos, cada vez que añado un registro se lentiza la base de datos, solo sirve para sacar la base de datos y no metes registro.



Video 9 NORMALIZACION: son reglas para separar la información



1FN: permite atomizar o reducir los datos de manera que no crezca a la horizontal, sino a al

Vertical. No podemos tener campos repetidos.



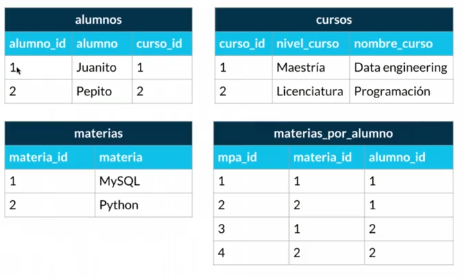
2FN: el campo de la clave debe depender (debe identificar el id de manera única un renglo)



3FN los campos que no son claves no deben tener dependencias.



4FN: Los campos multivaludos doble línea, se identifican con una clave única.



Video 10

Video 11

Video 12: instalamy sq

Vieo 13 : RDB Relation database managememnt systema SISTEMA MANEJADRO DE BASE DE DATOS RELACIONALES es un programa para utilizer la base de datos

My SQL, ORACLE, PG ADMIN.

OPEN SURCE POSSQL

Video 14: my sql para mac Video 15 instalar para

video 16 my sql worbench, cliente grafico crear base de datos.

clik derecho schema, y nombre + utfo permite manejar caracteres o asentos. Y dafaul

video 17: servicios claud o servicios administrados, son servicios que son manejados por terceros.

La empresa no administra la base de datos, aws. Claud platfon. Aushou

Video 18. SQL: hacer un lenguaje para consultar las bases de datos. STRUCTURED QUERY LANGUAJE es base relacional.

Video 19: DDL créate lenguaje data definition lenguaje, sirve para crear la estructura

Create sirve para crear , base de datos, tabla, una vista, un índice,

Alter. Sirve para modificar o alterar una tabla,

Drop sirve para borrar

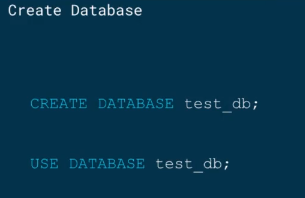
DDL manipula , los base de datos o esquimas,

table,

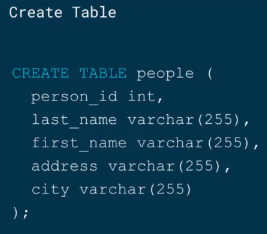
Viwe, son las vistas,

Comando CREATE objeto: DATABASE usar la BS USE DATABASE

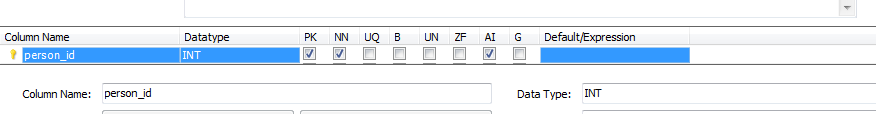
Crear base de datos:

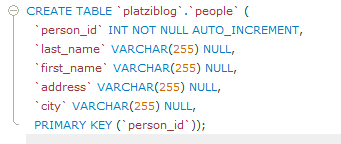


Crear tabla

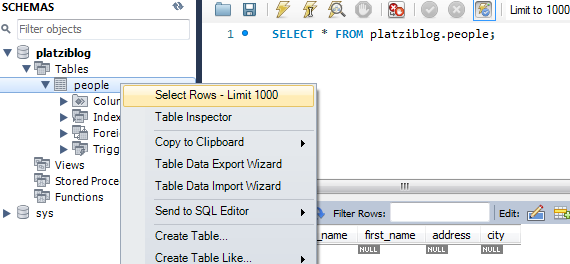


Crear columnas





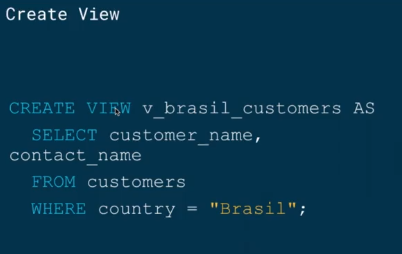
Primera consulta



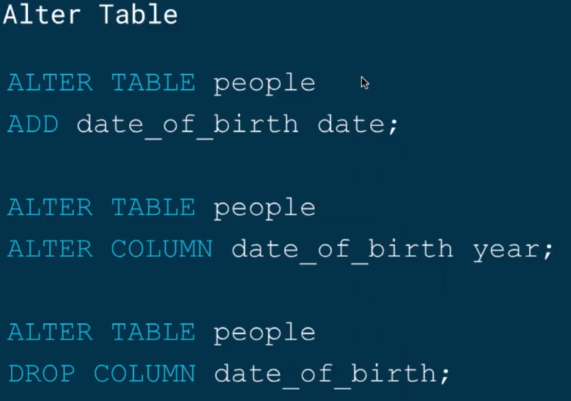
Video 20 vistas o view : sirven para tomar los datos para presentar y consultar

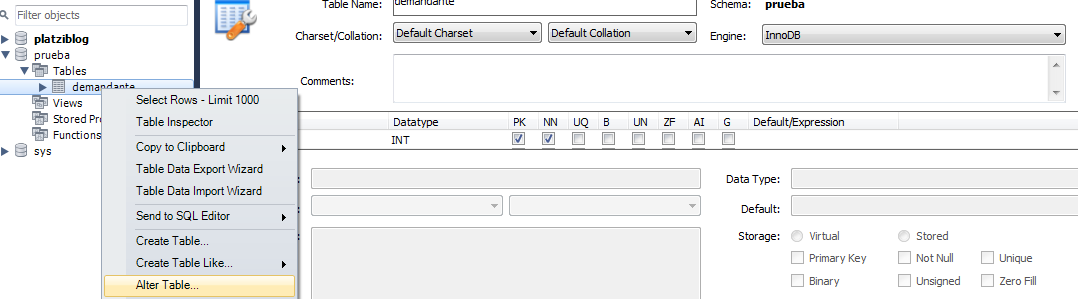
El nombre v\_ Brasil-customers es el inicio de la consulta.

Sirve para consultar los clientes de brazil

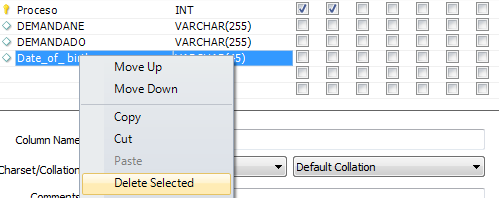


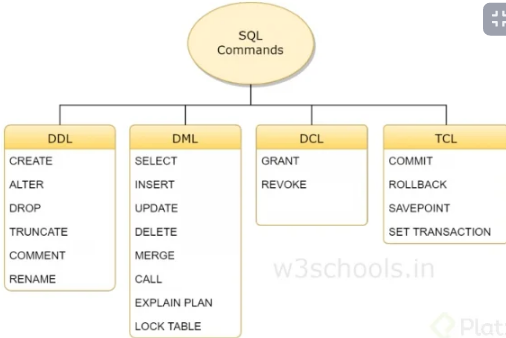
Alter para modificar





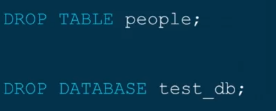
DROP BORRAR

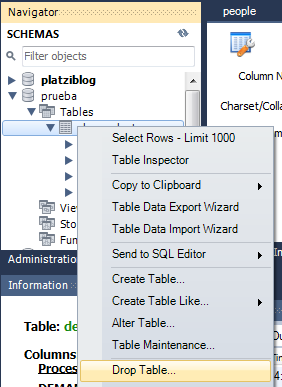




VIDEO 21 DROOP, sirve para borrar

Asi se borra tabla o base de datos





LENGUAJE DML: DATA MANIPULATION LANGUAJE : trata el contenido de la base de datos. Meteré actualizar extraer datos.

COMANDOS:

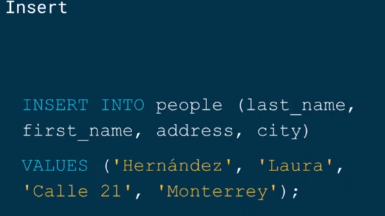
Insert: Update: Delete: Select:

INSERT: agrega un nuevo registro o renglones o tupla

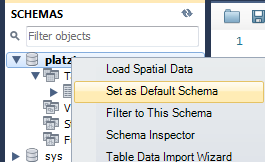
Insert inot luego a donde la tabla nombre de la tabla

(En paréntesis se escribren los campos que vamos a insertar)

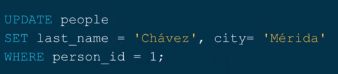
Values son los valores (ingresar los datos a insertar )



Sirve para marcar que la base de datos se identifique que está funcionando.



UPDATE: sirve actualizar o modificar los datos que ya se tiene, modifica el renglón



Esto quiere decir:

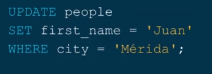
UPDATE significa De la tabla que vamos actualizar que es la tabla people

SET: significa que campo va a tener que valor / del campo last\_name cambiar por –chaves

Y del campo city cambia por Merida

WHERE donde la person\_id corresponde a 1;

EJEMPLO N° 2:



Esto quiere decir:

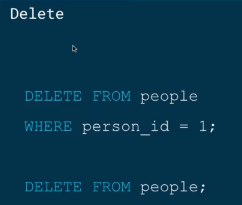
UPDATE significa De la tabla que vamos actualizar que es la tabla people

SET: significa que campo va a tener que valor / del campo first\_name cambiar por “Juan”

WHERE donde la city corresponda a Merida; (cambia todos los campos de frist\_name donde la ciudad dice Merida).

DELETE: sirve para borrar

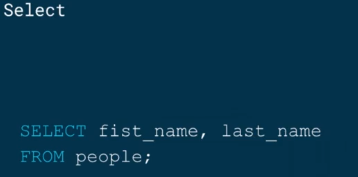
SI NO LE DECIMOS WHERE, borra toda la tabla.



SELECT: sirve para traer información de la base de datos

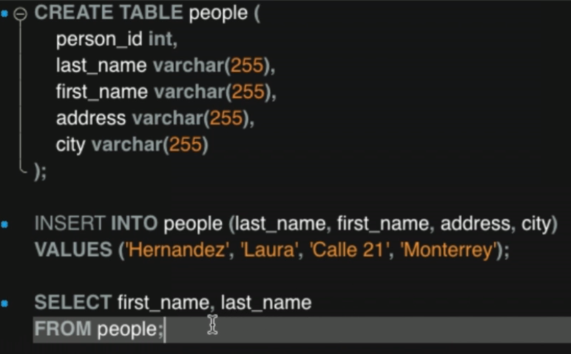
Select que campos quieres ver

From de que tabla quiere ver



VIDEO 23: el lenguaje SQL unifico la manera de preguntar al repositorio. El SQL es un lenguaje Universal

REPASO



VIDEO 24: crear base de datos

VIDEO 25: unir con llaves foráneas

VIDEO 26 crear base de datos

VIDEO 27: CONSULTAS o QUERY: extraer y presentar información permite

VIDEO 28: estructuras de QUERY: sentencia SELEC Y FROM

SELECT: permite traer los datos que queremos mostrar,

FROM: de donde tomamos los datos

WHERE: permite filtraer

GROUP BY: agrupar por

ORDER BY: ordenar el resultaro descendente

HAVING: filtrar el total

El \* significa todo

Video 29 como preguntar o hacer quares con SELECT ES PROYECTAR O MOSTRAR

Sirve para mostrar las columnas o los compos

El select con \* significa tráeme todo

FUNCIONES DE SELECT:

TRAER TODOS LOS CAMPOS

SELECT \*

FROM people;

TRAER SOLO LOS CAMPOS SOLICITADOS address y city

SELECT address , city

FROM people;

CAMBIAR EL NOMBRE DEL CAMPO o COLUMNA: Se le conoce como alias.

SELECT people\_id AS id\_persona, last\_name AS Apellido

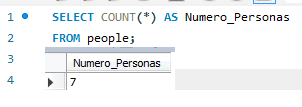
FROM people

COUNT sirve para contar todas las filas o registro que tiene la tabla

SELECT COUNT(\*)

FROM people;

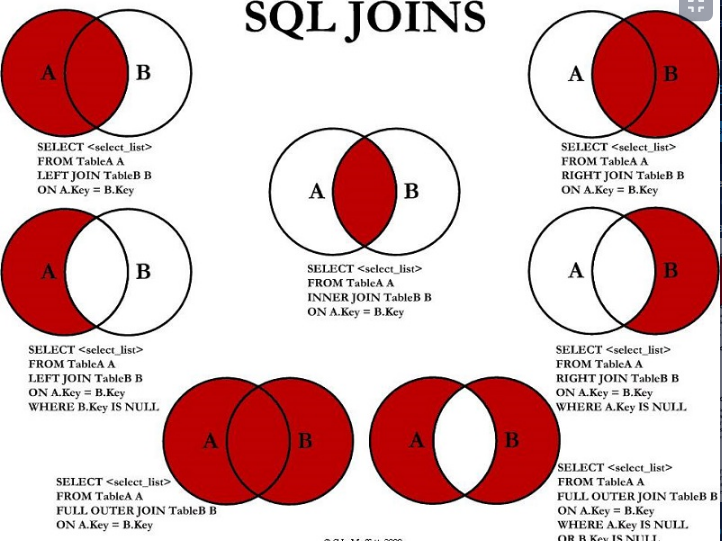
Con el alias AS le cambias el nombre a la columna que nos arroja el número de filas



Video 30: Ejercicio

Video 31: SENTENCIA FROM indica de donde quiero sacar la información, sirve para presentar la información como una sola tabla. Sirve para saber de que tabla tomamos la informacion

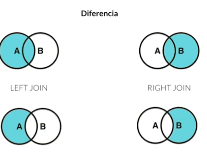
SENTENCIA JOIN significa juntar o unir, y son inseparables con FROM, sirve para unir tablas con la llave primaria con la llave foránea.



JOIN DIFERENCIAS: significa lo que te tenemos en un lado pero no está en el otro.

EL LEF JOIN NORMAL, significa unir todos los usuarios de la tabla B.

Tabla A y tabla B



INNER JOIN INTERSECCION donde se encuentran los dos círculos, es la general



UNION: trae todos los usuarios



DIFERENCIA SIMETRUICA:

Trae solo usuarios que no tengan relación.



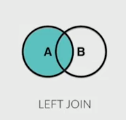
Video 32: Ejercicio. LEFT JOIN

Selecciona todo de la tabla usuarios, y unela de la tabla izquierda con la tabla derecha. Unela con las dos llaves.

SELECT \*

FROM usuarios

LEFT JOIN posts ON usuarios .id = POST.USUARIOS\_ID;



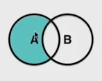
Mostrar todos los usuarios que no han escrito ningún post.

SELECT \*

FROM usuarios

LEFT JOIN posts ON usuarios.id = posts.usuarios\_id

WHERE posts.usuarios\_id IS NULL

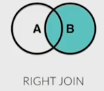


RIGHT JOIN

SELECT \*

FROM usuarios

RIGHT JOIN posts ON usuarios .id = POST.USUARIOS\_ID;



INTERSECCION INNER: significa (traer lo que está ligado) presentar datos ligados con relacionados con las 2 tablas

Este es el mas importante

SELECT \*

FROM usuarios

INNER JOIN posts ON usuarios .id = POST.USUARIOS\_ID;

UNION: traer todos los registros de las dos tablas

SELECT \*

FROM usuarios

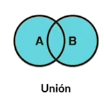
LEFT JOIN posts ON usuarios.id=posts.usuarios\_id

UNION

SELECT \*

FROM usuarios

RIGHT JOIN posts ON usuarios.id = posts.usuarios\_id;



DIFERENCIA SIMETRICA: traer lo que existe en A pero no en B y lo que existe en B pero no en A

SELECT \*

FROM usuarios

LEFT JOIN posts ON usuarios.id=posts.usuarios\_id

WHERE posts.usuarios\_id IS NULL

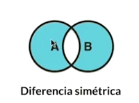
UNION

SELECT \*

FROM usuarios

RIGHT JOIN posts ON usuarios.id = posts.usuarios\_id;

WHERE posts.usuarios\_id IS NULL;



Video 33.

WHERE sirve para filtrar tuplas o registros

Presente los registros de mayores o iguales a 50. Sirve para filtrar por números

SELECT \*

FROM posts

WHERE id>=50;

Para filtrar con cadenas: se usa el operador =

A continuación se solicita proyectar todos los status activos de la base de datos.

SELECT \*

FROM posts

WHERE status = áctivos´;

También se puede traer con la sentencia no igual != (sirve con números y cadenas)

SELECT \*

FROM posts

WHERE status != áctivos´;

SENTENCIA LIKE:

Sirve para buscar una palabra parecida a la solicitada.

Filtrar la palabra %escandalo%, me

SELECT \*

FROM posts

WHERE titulo LIKE %escandalo% ;

SENTENCIA FECHA: filtrar por fechas

Presenta publicaciones menores a 01 enero 2025

SELECT \*

FROM posts

WHERE fecha\_publicacion>´2025-01-01´;

SENTENCIA BETWEEN sirve para filtrar entre fechas o id, y sirve para definir un rango.

SELECT \*

FROM posts

WHERE fecha\_publicacion BETWEEN ´2023-01-01´ AND ´2025-12-31´;

SELECT \*

FROM posts

WHERE id BETWEEN 50 AND 60;

SENTENCIA YEAR: sirve para filtrar en año entre un año y otro, genera un rango

SELECT \*

FROM posts

WHERE YEAR (fecha\_publicacion) BETWEEN ‘2023’ AND ‘2024’;

SENTENCIA MONTH, sirve para saber la fecha de publicación exacta o una fracción de fecha

SELECT \*

FROM posts

WHERE MONTH (fecha\_publicación) = ‘04’

Video 34 valores NULL y NO NULL

El valor null o nulo en una tabla es el valor favio , no mayo o menor que,

No null, quiere decir que es obligatorio el campo, se deb guarda campos.

Para consultar los valores de tipo null no es una número, no es una cadena, no es una fecha.

Para ver todos los registros donde el usuarios sin datos si emplea el siguiente código

SELECT \*

FROM posts

WHERE usuarios\_id IS NULL

O para presentar los not null filtro sobre filtro

SELECT\*

FROM posts

WHERE usuario\_id IS NOT NULL

AND status = ‘activo’

AND is <50

AND categoria\_id =2

Resumen

Select escoger la columna

From join de donde saco los datos o la fuente de datos a utilizar

Where filtrar datos por filas

Video 35 GRUP BY, significa agrupar, es indicarle a la base de datos que criterios deseo agrupar.

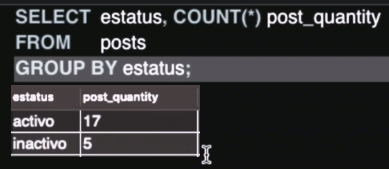
Quiero agrupar por categoría y agrupar por mes.

Sirve entregar informes, son cuadro resumen ejecutivo, no necestina ver el detalle.

SELECT estatus, COUNT (\*) post\_quantity

FROM posts

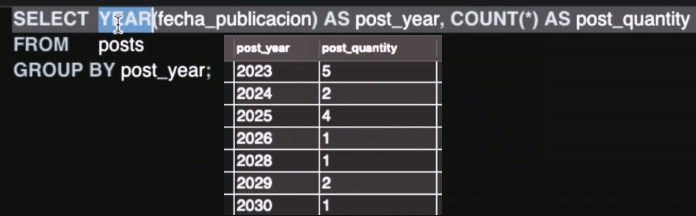
GROUP BY estatus;



SELECT YEAR ( fecha\_publicacion) AS post\_year, COUNT (\*) AS post\_quantity

FROM posts

GROUP BY post\_year;



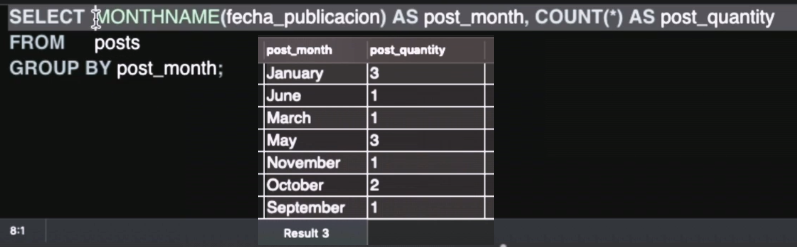
Agrupa por año, y presenta el número de posts que se publicaron en ese año.

MONTHNAME: significa nombre del mes.

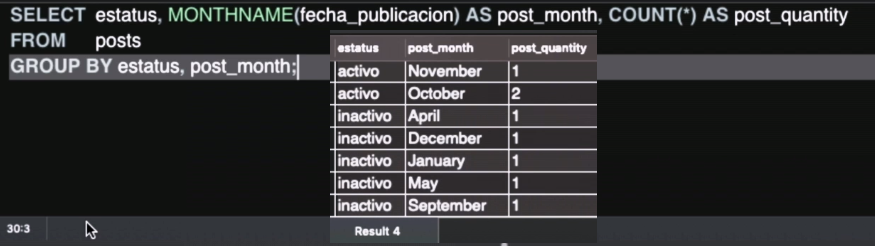
SELECT MONTHNAME (fecha\_publicación) AS post\_monthname, COUNT(\*) AS post\_quantity

FROM posts

GROUP BY post\_month;



PARA AGRUPAR POR VARIOS CRITERIOS



Marzo 29 / 23

Video 36:

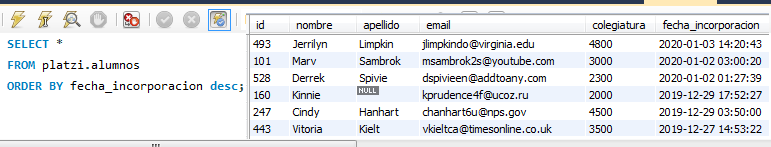
ORDER BY ordena por un criterio el campo (columna)

Ordenar fecha de mayor a menor

SELECT \*

FROM platzi.alumnos

ORDER BY fecha\_incorporacion desc; o ascendente ASC



ORDENAR POR CADENA:

Ordena el titulo por ASC y se puede también DESC (alfabético de Z hasta A)



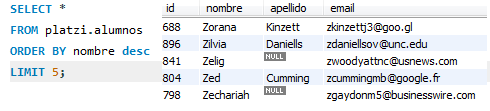
ORDER BY LIMIT: sirve para ordenar primero y presentar hasta un cierto número de datos. Es traer un top 10 o top 1000 o el top que quieras.

SELECT\*

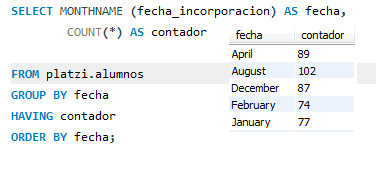
FROM platzi.alumnos

ORDER BY

LIMIT 10



SENTENCIA HAVING: son similares a where, y sirve para filtrar y traer por valores solicitados exactos.



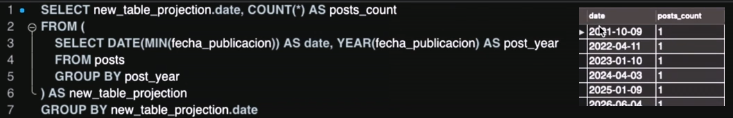
Video 37

SUB QUERY: dentro de un query se puede hacer otro query. Sirve para hacer una tabla y hacer un join con otra tabla, y existe solo en memoria. O hacer un join con otra tabla en base de datos.

Los querys aninados, se deben usar cuando un problema no se soluciona consultando una tabla, cuando se necesitan condiciones separadas.

Lo primero es hacer un query, luego lo conviertes en una tabla o valor, y luego lo conviertes en entrada del from o el where y de esta manera alimenta al segundo query.

Este proceso produce un producto cartesiano (es la multiplicación de todos los registros de una tabla por los registro de la otra tabla). Esto hace muy pesado el proceso del query.



NO es recomendable.

Video 38: TRANSFORMACION DE PREGUNTAS A QUERYS

SELECT quiere decir mostrar

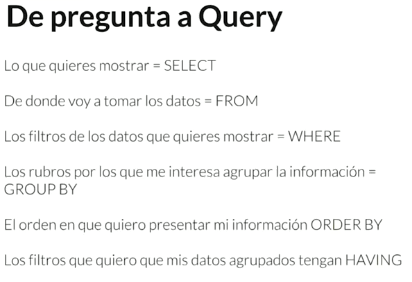
FROM de donde voy a tomar los datos

WHERE los filtros de datos a mostrar

GROUP BY los rubros por los que me interesa agrupar la información

ORDER BY es el orden por donde quiero presentar mi información (top 10)

HAVING filtrar nuevamente después del ORDER BY, es un segundo filtro.

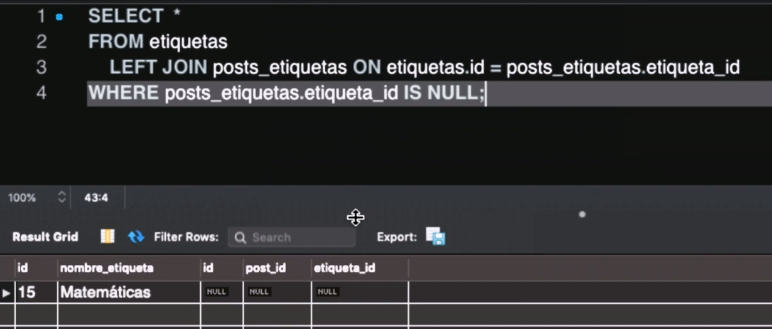


VIDEO 39 ejemplos: Cuantas etiquetas tiene un blogpost?

1. Que queremos mostrar o proyectar
2. Se conecta la unión de las tablas con ON y se crea la tabla transitiva
3. GROUP\_CONCAT agrupa el campo separado con comas, especifica



Ejemplo



Video 40

SELECT c.nombre\_categoria, COUNT (\*) AS cant\_posts

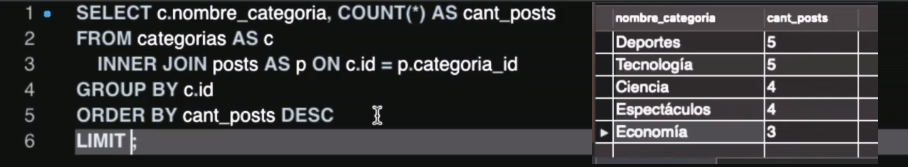
FROM categorias AS c

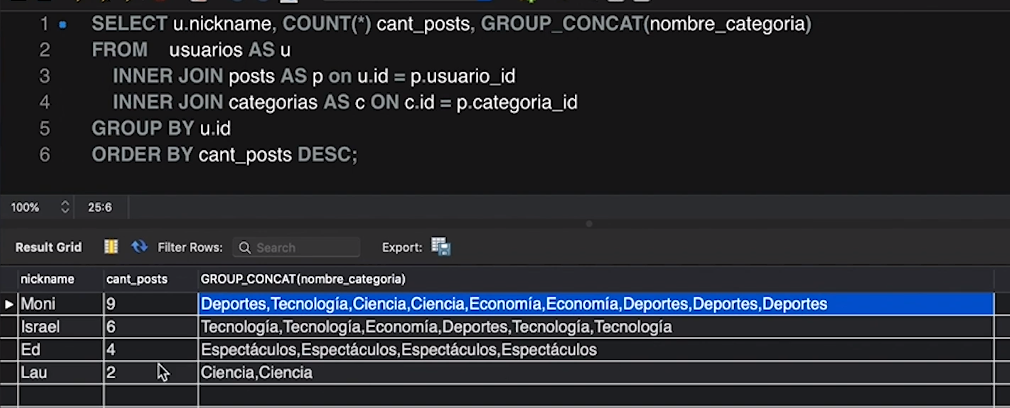
IINER JOIN posts AS p ON c.id = p.categoria\_id

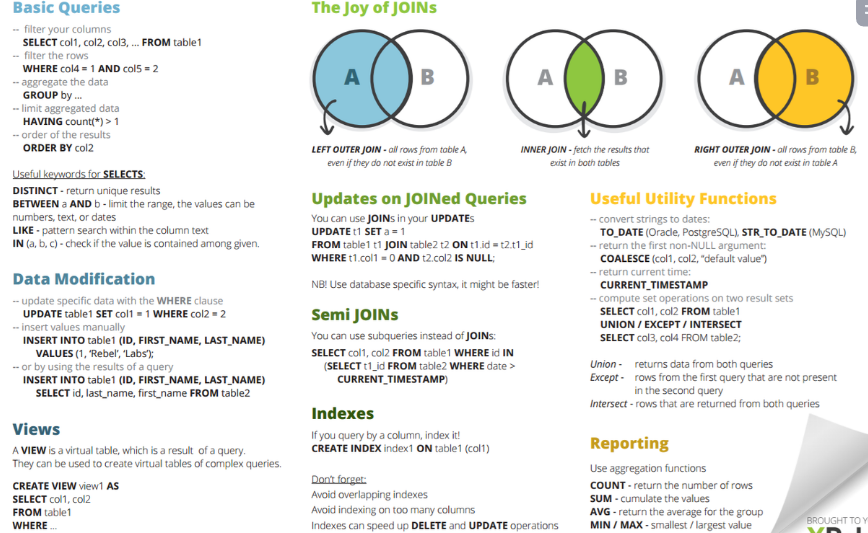
GROUP BY c.id

ORDER BY cant\_posts DESC

LIMIT;







Video 41 BASE DE DATOS NO RELACIONALES

Tipos:

Clave valor

Basadas Documentos

Basadas en grafos

En memoria

Optimizadas para búsqueda

CLAVE VALOR: sirve para almacenar datos rapidos, y gran cantidad de datos amarrado a un ID, algoritmos tipos diccionarios ejemplo Cassandra de Facebook, Dynamo

BASADAS EN DOCUMENTOS: los documentos son objetos de tipo JASON, son stander de la industria, están guardados como clave valor. Son los más utilizados ejemplos, MONGODB, FIRESTORE; son buenos para guardar datos de un juego por ejemplo, No es bueno para buscar por parámetros.

BASADAS EN GRAFOS: son nodos o entidades que tiene multiples relaciones, se usan en inteligencia artificial, NEO4J y TITAN

EN MEMORIA son muy rapidas, y pero tienen límite de capacidad, es de gran trabajo tener todo en sincronía. Ejemplo MEMCACHED REDIS.

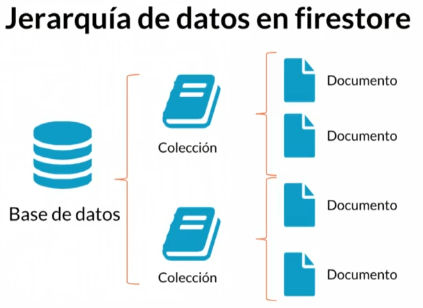
OPTIMIZADAS PARA BUSQUEDA, ayuda hacer queri muy difíciles en milisegundos. Ejemplo BIGQUERY, ELASTICSEARCH. Son utilizadas en bussines inteligen, machin learner.

VIDEO 42: BASADAS EN DOCUMENTOS servicio administrado (o servicios de nube (se delega a un operador clud la seguridad, redes, distribución de datos)).

FIRESTORES es una base de datos que utiliza google para crear aplicaciones web, o android, ioss, google interactua con firestore con interface web.

JERARQUIA DE DATOS EN FIRESTORE.

1. Base de datos
2. Colección (tablas)
3. Documentos (tuplas, es la información)



Video 43: es más parecido al mundo real la base de datos firestore.

TOP LEVEL COLLECTION : colecciones de nivel más alto.

Se crea base de datos web d google con firestore

Video 44:

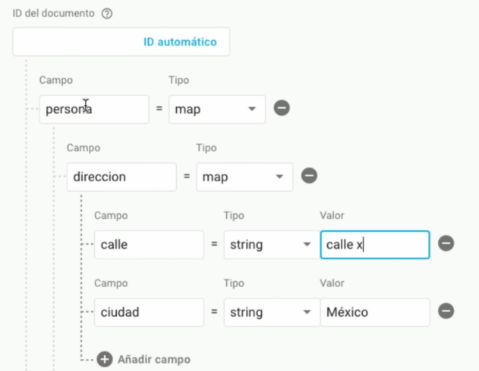
Guardar tipos de datos en firestore

String texto,

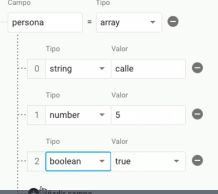
Number numéricos enteres flotantes punto decimal

Boolean verdadero o falso

Map mete un documento en otro documento

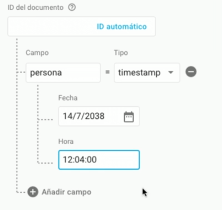


Array es un arreglo no importa el orden, para meter un grupo de datos, sin estructura ni identificador y si sirve para meter una serie de valores menos un Array.



Null sirve para crear el campo, en ese momento no tendrá valor,

Timestamp sirve para almacenar fecha día mes año y hora.



Geopoint se refiere a guardar localización geográfica. Del punto A y el punto B



Reference. Es para identificar que la colección está unido con otros documentos.

Video 45 sub-colecciones son colecciones que viven en el segundo nivel eso quiere decir que hace parte del documento anterior y no de la colección principal.

La regla a utilizar res : si es identidad separada y se deben realizar muchas consultas se debe es hacer un top level collection, si no hay que traer información fuera del documento, se debe realizar la sub-collection.

Video 46: importar datos a firebase

Las bases de datos no relacionales no están diseñadas para realizar querys o consultas

Video 47 practica para relacionar con una top level colección una post o noticia.

Copiar la referencia usuarios

En el posts añadir un campo y bautizar autor, con campo reference y pegar la referencia anterior.

Video 48

RECORDATORIO:

Tabla = colección

Tupla o renglón = documento

Colecciones se dividen en 2:

Top level collection

Subcollections (viven en documentos padre)

Video 49 BASES DE DATOS EN EL MUNDO REAL

Video 50 BIG DATA: grandes cantidades de datos, es guarda muchísimos millones de datos, en mili-segundos.

Cassandra nace de Facebook.

Video 51 DATA WAREHOUSE: grandes naves de capacidad de almacenaje, guardar grandes cantidades de datos. Sirve para guardar como archivo muerto y no se puede perder.

Los histórico de face o yotube se guardan en Data Warehouse. Guarda datos de manera eterna, y realizar búsqueda de esta información histórica. Ejemplo big queri.

BIGTABLE: es una solo tabla que guarda históricos de google, peo defici realizar consultas.

Video 52 DATA MINIG es una disciplina para minar datos, extraer datos de donde sea que estén, y de darles un sentido. Es poner orden en base de datos de archivos históricos.

Video 53 ETL: es herramienta que viene de DATA MINIG: extraer transformar y cargar datos, es tomar los datos de un lugar, luego inicia proceso de transformación y al final se debe volver a cargar.

Uso 1: sacar la información, limpiarla, y cargar en una nueva base de datos.

Uso 2: en tiempo real es tomar los datos vivos y sacar la información, limpiarla, y cargar en una nueva base de datos.

Y ETL tomas los datos vivos de la aplicación, los limpia, organiza y los envía a un DATA WAREHOUSE. ETL es una técnica que sirve para transforma los datos de una forma caotica a limpia.

Video 54 BUSINESS INTELLIGENCE: inteligencia para el negocio, tener la información oportuna clara y elementos para tomar decisiones adecuadas, BUSINESS INTELLIGENCE, buscar proveedor a todo el mundo con los datos y la información correcta para tomar buenas decisiones.

Es una disciplina de tomar los datos ya transformado, y darle sentido a toda la anterior información y aplicarla a algo del negocio.

Ejemplo, comportamiento de ventas, que audiencia tenemos edad, etc, temas que les gustan a nuestros clientes. Ayuda para predecir el después, entender tendencias.

La data viva no sirve para business inteligente.

Busca tendencia, se repetido, comportamiento, usuarios demográfica, busca patrones.

Buscar correlaciones, patrones.

Video 55 MACHINE LEARNING: son técnicas de inteligencia artificial identificación de patrones, es para tratamiento de datos para encontrar temas interesantes. Ayuda es hacer modelos, es encontrar patrones no obvios.

Caso de 1: Problema de clasificación: procesamiento de lenguaje natural, ingresar datos con lenguaje natural, ejemplo, se entrena con ejemplo temas de política, genera un patrón y sale a buscar en la base de datos los temas de política, gracias al modelo generado.

Caso 2. busque las ventas de rango del 90 al 95, genera un modelo y se le pasa las ventas nuevas del 95 al2020, y busca relaciones y presenta por gráficas. Sirve para predecir el futuro.

Se realiza un modelo, se entrena, se le entrega el modelo y el resulta es predicción.

Video 56 DATA SCIENCE: poca gente realiza esta actividad, es la persona que aplicar estadística, machine learnig, Busines intellligence, etl, etc, para tomar decisiones acertadas basadas en las ciencia de datos.

Video 57 sigue aprendiendo

Video 58 BASES DE DATOS RELACIONALES VS NO RELACIONALES

Bases de datos no relacionales, basados en documentos, sirve para mantener los datos vivos de una aplicación y las de grafos son útiles para encontrar relaciones.

Dependiendo del problema se puede utilizar las bases de datos no relaciones.

Machini lear encontrar patrones que el ser humano no encontraría.

El volumen de datos es grande se puede utilizar base de datos no relacionales

Video 59: ELEGIR UNA BASE DE DATOS:

EXAMEN

Importante registro información // permite conocimiento perpetue

12 reglas de codd // mandamientos

Entidad débil // identidad y existencia

Formas normales /// 4

Servicios administrados // admon una compañía dedicada

S= SQL // structured

Fuertemente lenguaje DDL /// inicio

Créate /// tabla

Alter table /// agregar una columna

DROP borrar base de datos /// cierto

Update crear nueva tupla /// falso

Sql corre con minimas modificacines /// cierto

Estructura básica query /// select from whereprinci

Top level collection /// falso

Dato firestore /// string

Tuplas rdbms firestore /// documentos

Base de datos basada en documentos /// estado actual de una aplicacion

Uso base de datos en grafos /// relaciones complejas y machine learning

Bussines intelligence /// ayudar a tomar mejores desiciones

Funciones básicas servicios machine /// predicción y clasificacion

Denotar relación entre 2 entidades // verbo

ERROR

Diferencia char y varchar // manera dinámica crece o encoje almacena cadenas optimiza el uso de memoria (error varchar puede contener caracteris especiales)

RDBMS open source /// postsql (error cassandra)

DLM /// trata el contenido d la base de datos meter actualizar borrar y etraer datos de las tablas. ( error al modificar la estructura de la bd)

Delet para red de protección /// where (error none)

Sentencia de consulta /// query (error créate)

Datos firestores /// base de datos aplicaciones we. Andoride, ios, muy visual, (error colección)

Sub colecciones cuando quremos indepentendiete /// falso (error cierto)

Entidades de RDBMS firestore /// (error objetos)