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

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

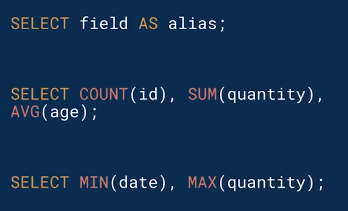
Selectr nombre de la columnas AS sirve para agregar un alias a la columna

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

SUM: suma los valores de la columna

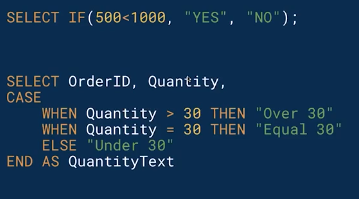
AVG : saca el valor promedio de toda la columna

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.

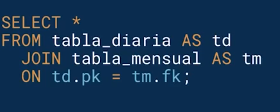
END AS qunatite text sirve para luego filtrar y/o ordenar

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.

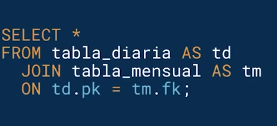
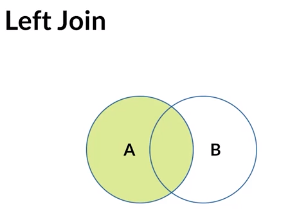
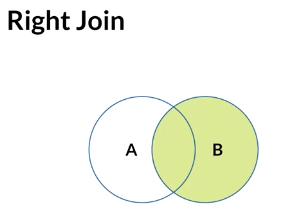


Diagrama de BEN



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

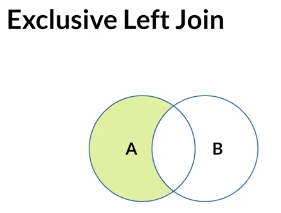


Traer los elementos de la tabal b sin importar si tiene equivalentes en la tabla a

EXCLUSIVE LEFT JOIN

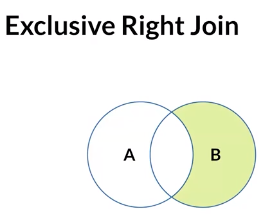
La clave es que estén seguro que no estén en la otra tabla

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



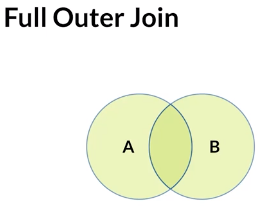
EXCLUSIVE RIGHT JOIN:

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



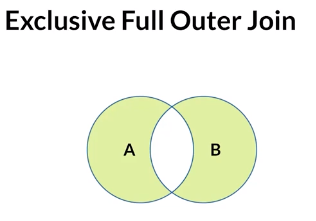
FULL OUTER JOIN

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

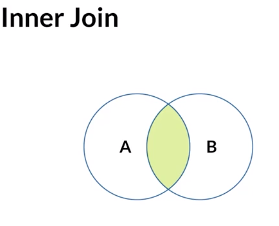


EXCLUSIVE FULL OUTER JOIN

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

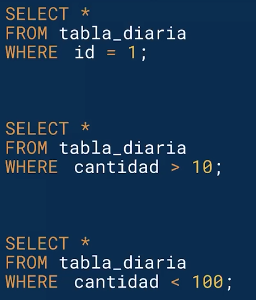
Where básico

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

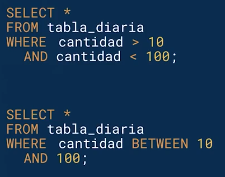


Where cantidad >10;

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.



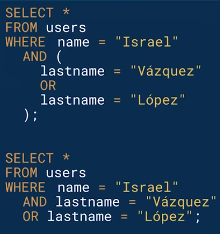
Where cantidad between 10 y 100;

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

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

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

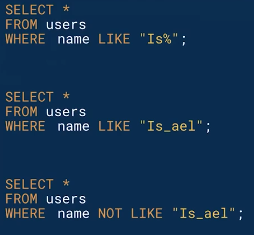
De la tabla usuarios traerme todos los llamados Israel de apellido Vásquez o López. Ojo con paréntesis.



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.



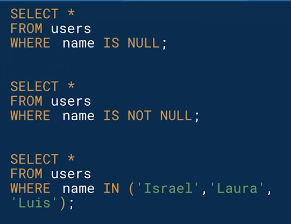
Is\_ ael: significa tráeme cual palabra que cumpla el inicio con IS y termine con AEL. Eso significa el guion bajo \_. Es el comodin.

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

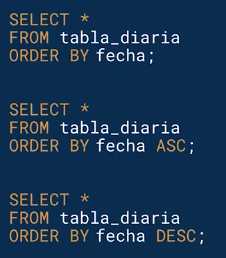
Proyección con selec

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



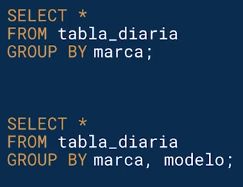
Con orden BY …..ASC: sirve para que los datos sean filtrados ascendentes

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

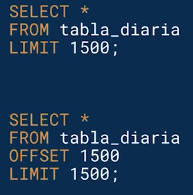
LOS INDICES: Sirve para búsquedas y ordenamientos sirve para hacer las extracciones muy rápidas

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.

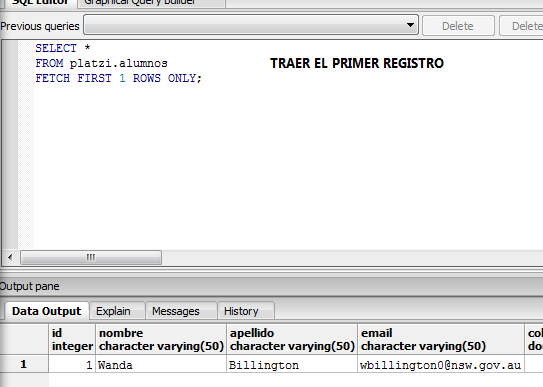


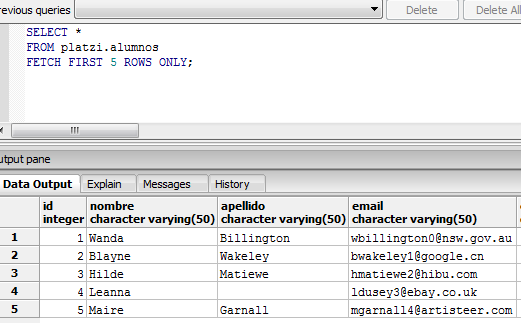
OFFSET quiere decir , de estos 1500, DESPUES tráigame los siguientes 1500.

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

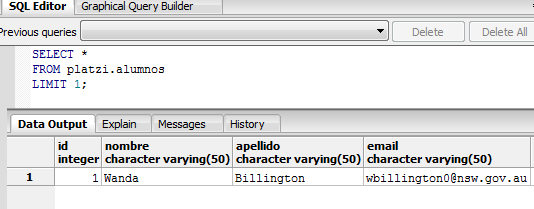
VIDEO 10:

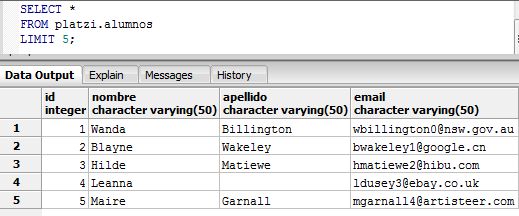
1. FETCH significa tráeme : EL PRIMER EJERCICIO Y EL PRIMER REGISTRO





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

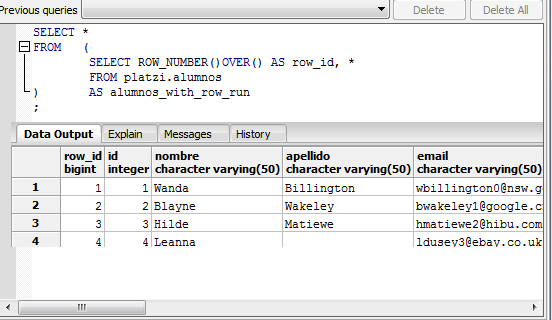
AS el alias se llamara row\_id (al principio) \* significa todo el resto de la tabla

FROM: de que tabla es que se saca la información

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

ROW significa registro

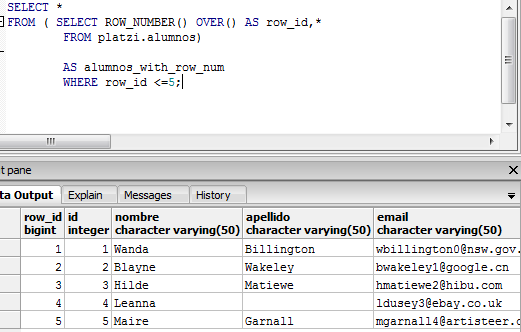
Si no se especifica nada trae toda la tabla

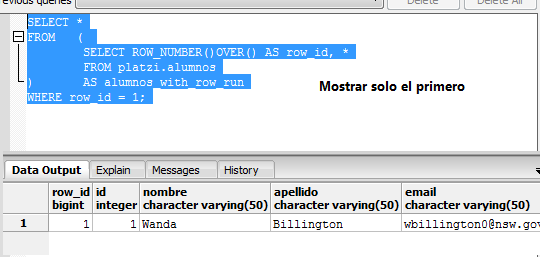


ROW\_ NUMBER es la función de ventana

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

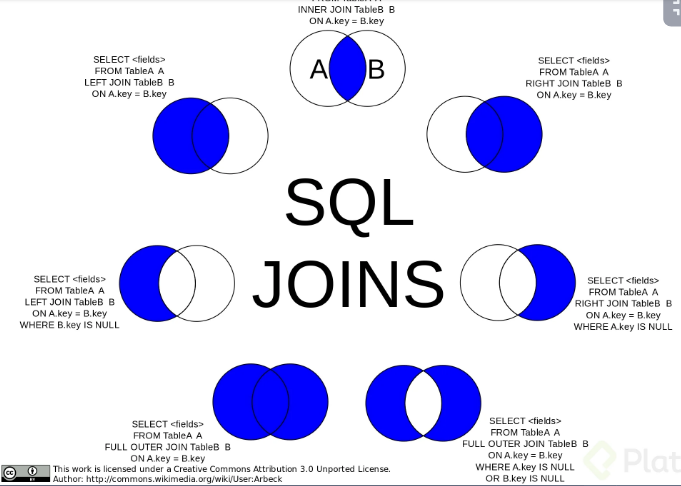
Como sacar los MINIMOS Y MAXIMOS

Video 18

SELFISH sirve para hacer un JOIN con la propia tabla

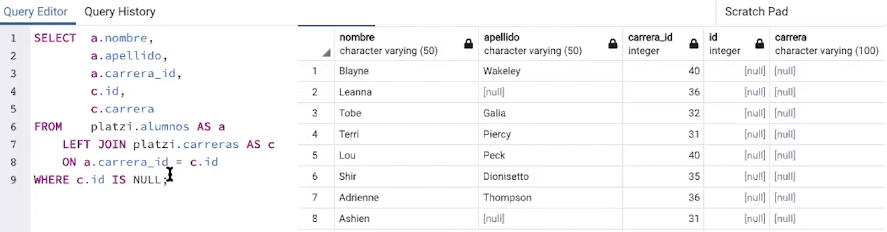
Video 19: LAS DIFERENCIAS: son necesarias para extraer cierto tipo de datos, Son los elementos que se encuentran en una tabla pero que no se encuentran en otra.

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

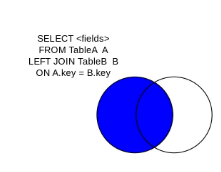
ON nos indica porque campo podemos unirlas las tablas, para este caso se realizara por carrera id tabla alumnos debe ser igual al campo id de la tabla carreras

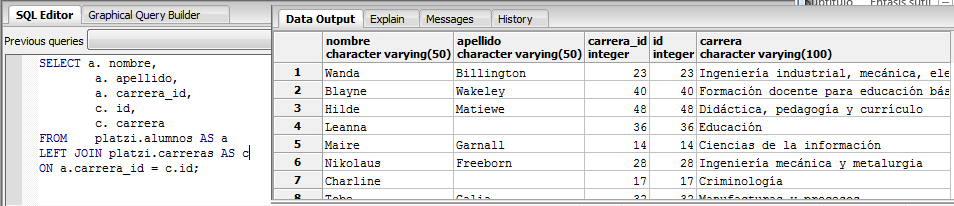


Nos muestra a los alumnos que cuentan con una carrera que ya no existe en la tabla carreras.

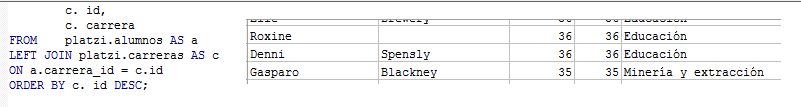
1. LEF JOIN INCLUSIVO o NORMAL:

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.

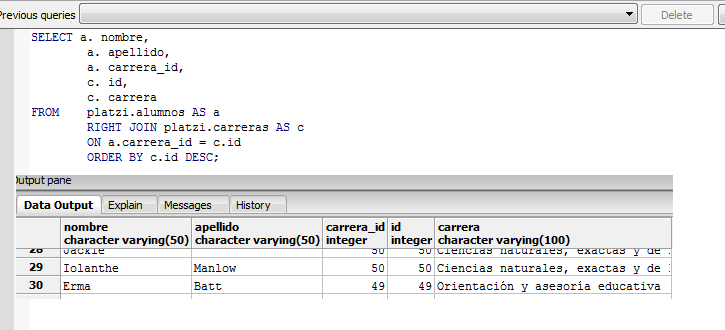




ORDENADO DECENDENTE

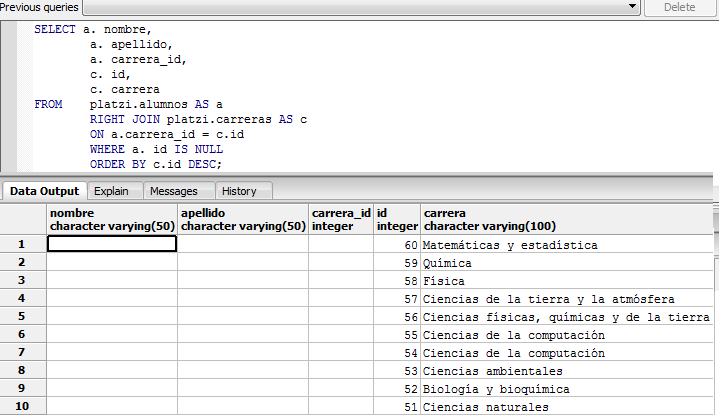


1. RIGHT JOIN: se cambia la palabra left por right y lo convierte, se traen las carreras sin importar un alumno si o no. Se traen todas las carreras con o sin alumno.

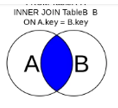
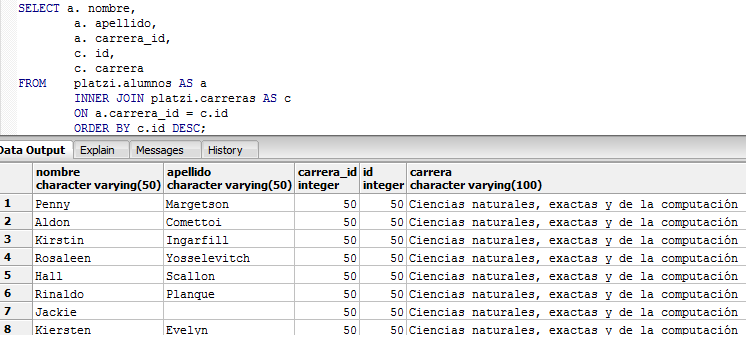


1. RIGHT JOIN EXCLUSIVO: se debe agregar la cláusula WHERE donde el id a sea null

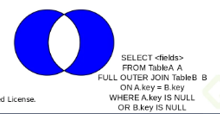
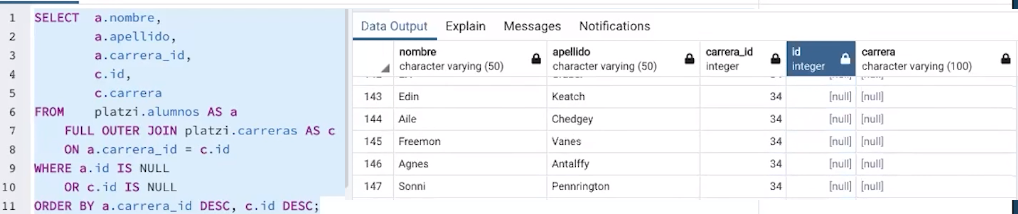
Es traer las carreras que no tiene alumnos, que existen a la derecha



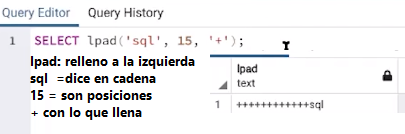
1. INNER JOIN: Es cuando no se especifica el JOIN, sirve para identificar la parte en común de ambas tablas. Pertenecen en las dos tablas. Es el mas común.



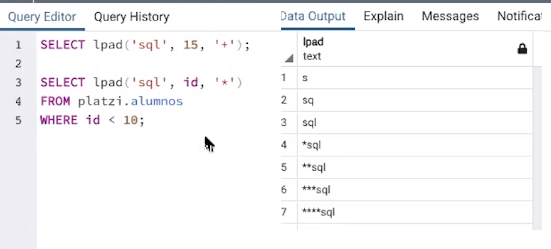
1. FULL OUTER JOIN: DIFERENCIA SIMETRICA: es la que sirve para identificar las que se encuentran A o se encuentra B pero no se encuentra en ambas.

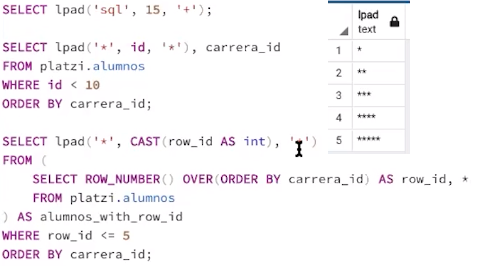
 

Video 21 TRIANGULANDO = METODO (lpad): sirve para hacer cadenas con longitudes que necesitemos muchas veces podemos llenarlas con espacios o guiones etc.

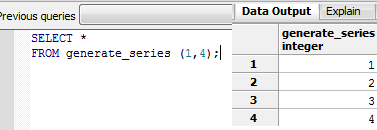


Triangulo creado con ID. Llega hasta 2.000 di

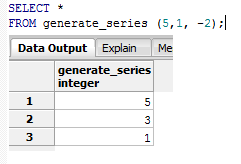




Video 23 GENERAR RANGOS: rangos básicos:

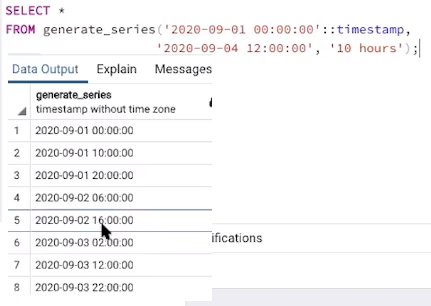
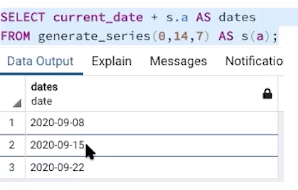


Rangos con delta o un paso (-2)

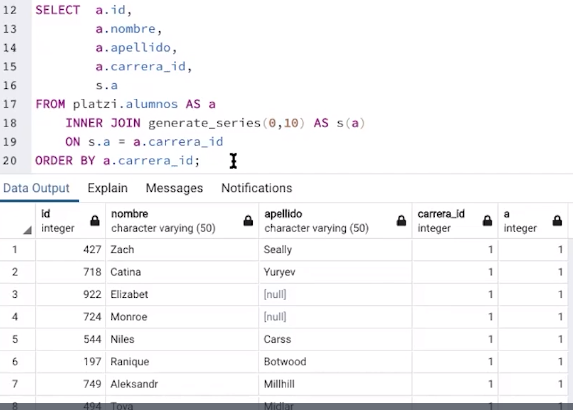


NOTA: cuando el primero es mayor que el segundo (5,1), genera un rango vacío, siempre se necesita un delta o un paso, que significa de cuanto en cuanto debe saltar.

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

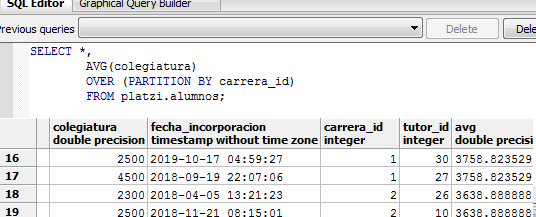
Video 25 QUERIES DISTRIBUIDOS

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

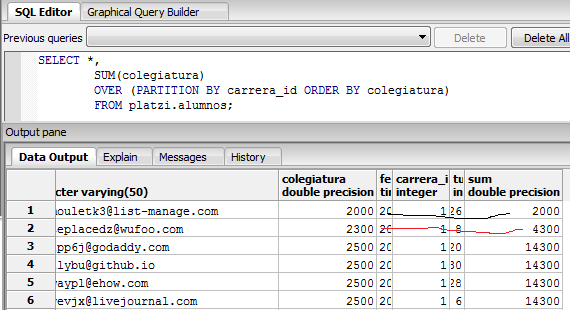
Videos26 WINDOS FUNTION: Es la relación entre una tupla o registro y el resto de registro en una sección de la tabla. Es la relación entre un registro y el resto de registro en una partición.

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.

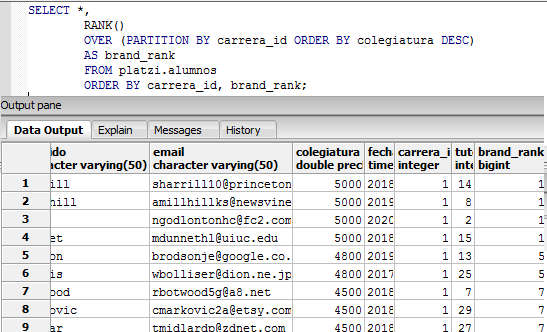


Ejemplo: sumar precio matricula, organizado por la columna carrera\_id y ordenado por precio matricula. Colegiatura es precio matricula.



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



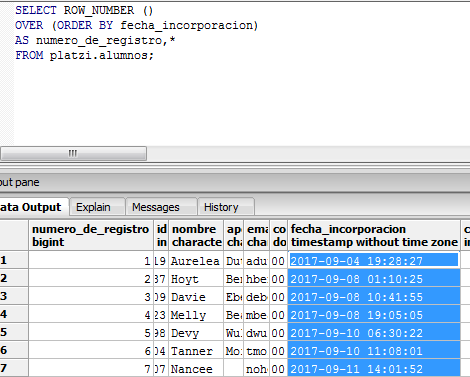
Video 27 PRINCIPALES WINDOW FUNCTIONS

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

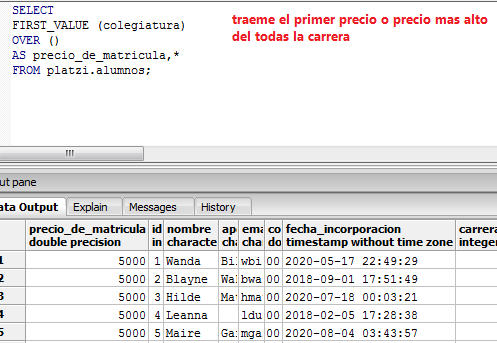
OVER () vació significa sin ninguna orden. (ORDER BY) significa ordenar por o filtrar por. (allí se le debe escribir por cual columna desea filtrar ejemplo por fecha de incorporación) (PARTITION BY significa agrupeme por )

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

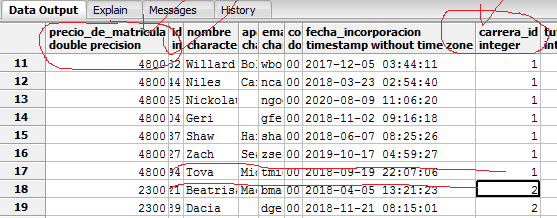
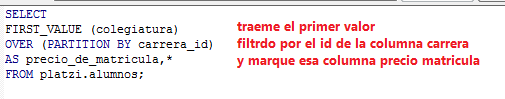
“\*” es \* al final significa añádale los datos del resto de la tabla.



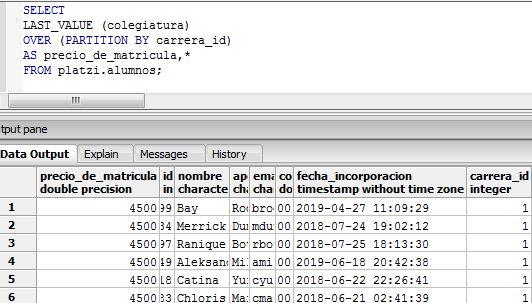
1. FIRST\_VALUE: Primer valor de una serie de datos



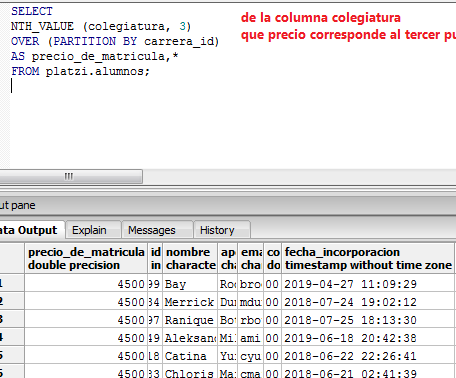
Ejemplo 2.



1. LAST\_VALUE Traer el ultimo valor

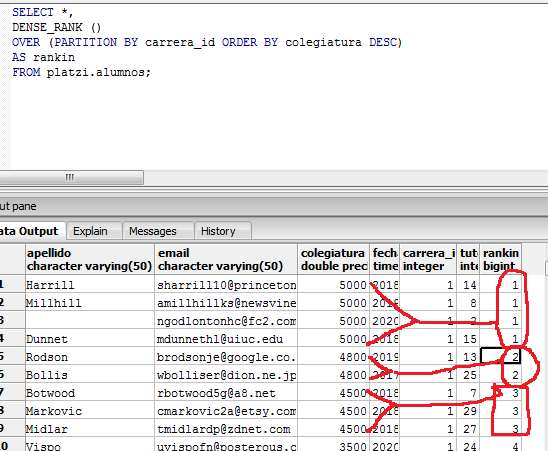


1. NTH\_VALUE: sirve para decirle que puesto debe traer, en este caso corresponde al tercer puesto.



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

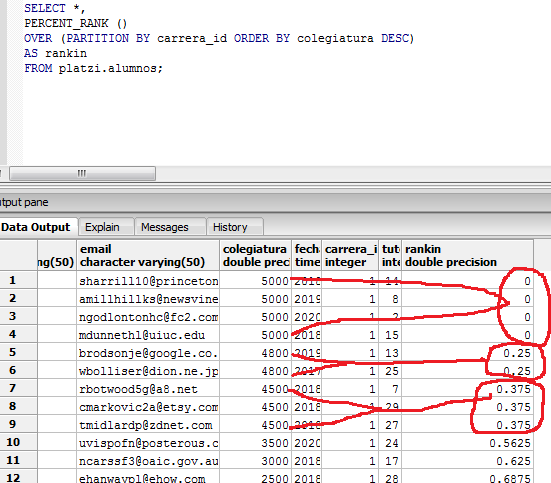
Con este código le decimos crea un columna organizada por la columna carrera y filtrada por el precio de matrícula de mayor a menor.



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 ascendente

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

ERROR

Comodin sustituir caracteres error \*

Tabal remota error fetchrermote

Limit 1 offset 105 error 105

Fetch firs 15 row limit 15 error falso

Rango dinamico error especificando en el select

Date parte error select y limit

Max y limit error max no se usa para un atabla completa

Convierte lef join where error full outer join

Valor inicial valor final error el valor despues del punto decimal

Query distribuido de red se considera despreciable error verdadero

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

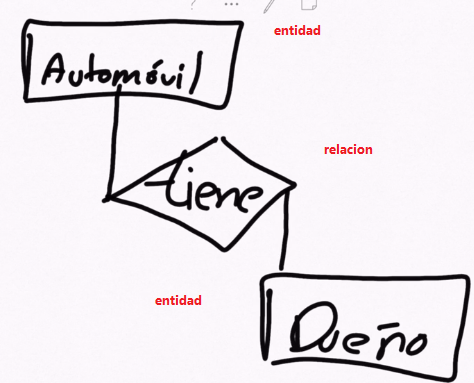
**ENTIDADES** son objetos del mundo real ejemplo identificados usuarios, comentarios, post, categorías.

**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)

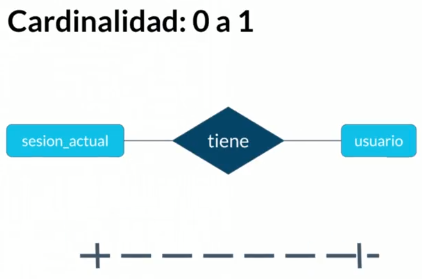
RELACIONES se representa con un rombo,



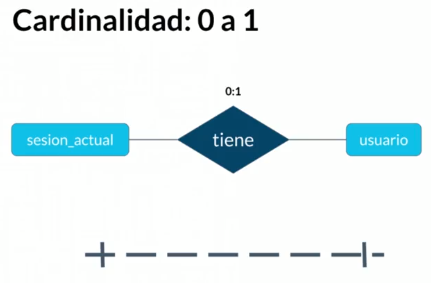
NOTA: cuando un atributo es multivaluado, eso quiere decir que tiene más de un atributo, y se convierte en una ENTIDAD SEPARADA.

CARDINALIAD: tiene que ver con números,

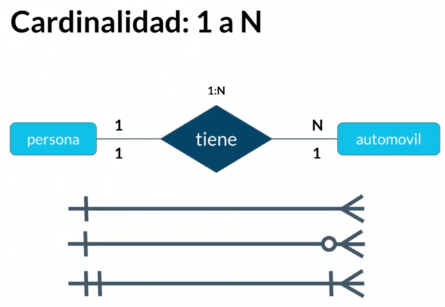
CARDINALIDAD 1 A 1: de un lado tengo un dato y del otro tengo solo uno también.



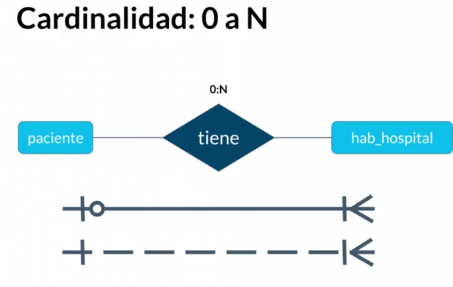
CARDINALIDAD 0 A 1 es llamada 1 a 1 opcional, puede existir que no haya nadie de un lado, tiene al menos uno.



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.

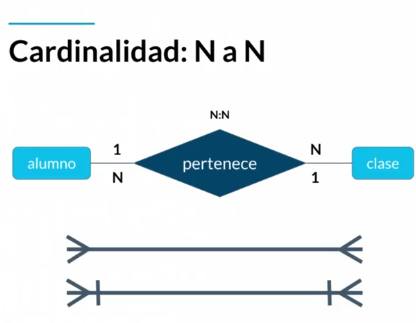
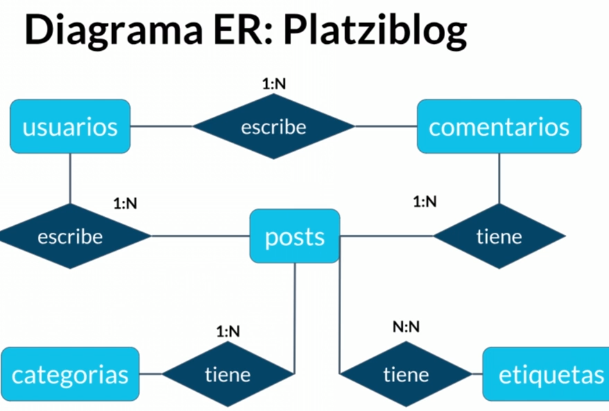


DIAGRAMA ENTIDAD RELACION es un mapa, debe ir antes de la creación de la base de datos.

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,

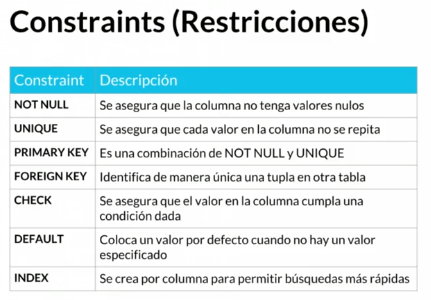
PRIMARY KEY: llave primaria, campo clave o campo llave, es not null y es unique, sirve para hacer la unión entre una tabla y se llama índice, sirve para hacer las relaciones.

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.

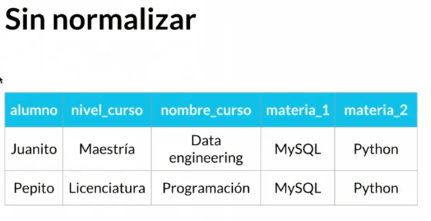
CHECK: permite definer las reglas que necesitamos ejemplo. solo deja entrar registro activos o inactivos.

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

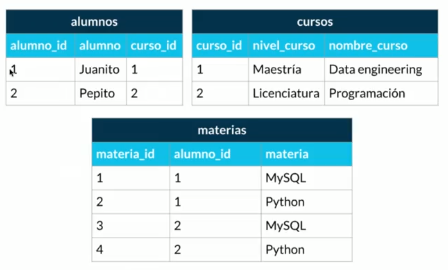
Vertical. No podemos tener campos repetidos.



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