Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 1

#Sesión 1 Fundamentos de SQL

 # RETO 1

Usando la base de datos tienda, muestra la descripción de las tablas articulo, puesto y venta. Por cada tipo de dato que encuentras llena la siguiente tabla, a mano. Usa la Documentación de MySQL como referencia.

```
SHOW TABLES

DESCRIBE articulo;

DESCRIBE puesto;

DESCRIBE venta;
```

 # RETO 2

Usando la base de datos tienda, escribe consultas que permitan responder las siguientes preguntas.

```
#¿Cuál es el nombre de los empleados con el puesto 4?

SELECT nombre

FROM empleado

WHERE id_puesto = 4;

#¿Qué puestos tienen un salario mayor a $10,000?

SELCT *

FROM puesto

WHERE salario > 10000

#¿Qué articulos tienen un precio mayor a $1,000 y un

iva mayor a 100?

SELECT *

FROM articulo

WHERE precio > 1000

  AND iva > 100;
```

```
#¿Qué ventas incluyen los artículo 135 o 963 y fueron
 hechas por los empleados 835 o 369?
SELECT *
FROM venta
WHERE id_articulo IN (135,963)
  AND id_empleado IN (835,369);
```

#RETO3

```
# Usando la base de datos tienda, escribe una
consulta que permita obtener el top 5 de puestos
por salarios.
SELECT *
FROM tienda
ORDER BY salario DESC
LIMIT 5;
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 2

#SESION 2

#RETO 1

#Usando la base de datos tienda, escribe consultas que permitan responder las siguientes preguntas.

#¿Qué artículos incluyen la palabra Pasta en su

nombre?

```
SELECT *
FROM articulo
WHERE nombre LIKE '%PASTA%';
```

#¿Qué artículos incluyen la palabra Cannelloni en

su nombre?

```
SELECT *
FROM articulo
WHERE nombre LIKE '%Cannelloni%';
```

#¿Qué nombres están separados por un guión (-) por ejemplo Puree - Kiwi?

```
SELECT *
FROM articulo
WHERE nombre LIKE '% - %';
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#RETO 2

#Usando la base de datos tienda, escribe consultas que permitan responder las siguientes preguntas.

#¿Cuál es el promedio de salario de los puestos?

```
SELECT avg(salario)
FROM puesto;
```

#¿Cuántos artículos incluyen la palabra Pasta en su nombre?

```
SELECT count(*)
FROM articulo
WHERE nombre LIKE '%pasta%';
```

#¿Cuál es el salario mínimo y máximo?

```
SELECT min(salario), max(salario)
FROM puesto;
```

#¿Cuál es la suma del salario de los últimos cinco puestos agregados?

```
SELECT max(id_puesto) - 5
FROM puesto;
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#RETO 3

#Usando la base de datos tienda, escribe consultas que permitan responder las siguientes preguntas.

#¿Cuántos registros hay por cada uno de los puestos?

```
SELECT nombre, count(*)
FROM puesto
GROUP BY nombre;
```

#¿Cuánto dinero se paga en total por puesto?

```
SELECT nombre, sum(salario)
FROM puesto
GROUP BY nombre;
```

#¿Cuál es el número total de ventas por vendedor?

```
SELECT id_empleado, count(clave) AS ventas
FROM venta
GROUP BY id_empleado;
```

#¿Cuál es el número total de ventas por artículo?

```
SELECT id_articulo, count(*)
FROM venta
GROUP BY id_articulo;
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#RETO 4

#Usando la base de datos tienda, escribe consultas que permitan responder las siguientes preguntas.

#¿Cuál es el nombre de los empleados cuyo sueldo es menor a $10,000?

```
SELECT nombre, apellido_paterno
FROM empleado
WHERE id_puesto IN
 (SELECT id_puesto
    FROM puesto
    WHERE salario > 10000);
```

#¿Cuál es la cantidad mínima y máxima de ventas de

cada empleado?

```
SELECT id_empleado, min(total_ventas), max(total_ventas)
FROM
 (SELECT clave, id_empleado, count(*) total_ventas
    FROM venta
    GROUP BY clave, id_empleado) AS sq
GROUP BY id_empleado;
```

#¿Cuál es el nombre del puesto de cada empleado?

```
SELECT nombre, apellido_paterno,
(SELECT nombre FROM puesto WHERE id_puesto = e.id_puesto)
FROM empleado AS e;
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 3

#SESION 3

#RETO 1

#Usando la base de datos tienda, escribe consultas que permitan responder las siguientes preguntas.

#¿Cuál es el nombre de los empleados que realizaron cada venta?

```
SELECT clave, nombre, apellido_paterno
FROM venta AS v
JOIN empleado AS e
  ON v.id_empleado = e.id_empleado
ORDER BY clave;
```

¿Cuál es el nombre de los artículos que se han vendido?

```
SELECT clave, nombre
FROM venta AS v
JOIN articulo AS a
  ON v.id_articulo = a.id_articulo
ORDER BY clave;
```

#¿Cuál es el total de cada venta?

```
SELECT clave, round(sum(precio),2) AS total
FROM venta AS v
JOIN articulo AS a
  ON v.id_articulo = a.id_articulo
GROUP BY clave
ORDER BY clave;
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#RETO 2

#Usando la base de datos tienda, define las siguientes vistas que permitan obtener la siguiente información.

#Obtener el puesto de un empleado.

```
CREATE VIEW puestos AS
SELECT concat(e.nombre, ' ', e.apellido_paterno), p.nombre
FROM empleado e
JOIN puesto p
  ON e.id_puesto = p.id_puesto;

SELECT *
FROM puestos;
```

#Saber qué artículos ha vendido cada empleado.

```
CREATE VIEW empleado_articulo AS
SELECT v.clave, concat(e.nombre, ' ', e.apellido_paterno) nombre, a.nombre articulo
FROM venta v
JOIN empleado e
  ON v.id_empleado = e.id_empleado
JOIN articulo a
  ON v.id_articulo = a.id_articulo
ORDER BY v.clave;

SELECT *
FROM  empleado_articulo;
```

```
#Saber qué puesto ha tenido más ventas.
CREATE VIEW puesto_ventas AS
SELECT p.nombre, count(v.clave) total
FROM venta v
JOIN empleado e
  ON v.id_empleado = e.id_empleado
JOIN puesto p
  ON e.id_puesto = p.id_puesto
GROUP BY p.nombre;

SELECT *
FROM puesto_ventas
ORDER BY total DESC
LIMIT 1;
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 4

#SESION 4

#RETO 1

#Usando la base de datos sample_mflix, proyecta los datos que se solicitan.

#Fecha, nombre y texto de cada comentario.

{date:1, name:1, text:1}

#Título, elenco y año de cada película.

{title:1, cast:1, year:1}

#Nombre y contraseña de cada usuario.

{name:1, password:1}

#RETO 2

#Usando la base de datos sample_mflix, agrega proyeccciones, filtros, ordenamientos y límites que permitan contestar las siguientes preguntas.

#¿Qué comentarios ha hecho Greg Powell?

{name: "Greg Powell"}

#¿Qué comentarios han hecho Greg Powell o Mercedes Tyler?

{$or: [{name: "Greg Powell"}, {name: "Mercedes Tyler"}]}

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#¿Cuál es el máximo número de comentarios en una película?

#Para responder esta pregunta, necesitamos tres cosas.

#Proyectar el número de comentarios

{num_mflix_comments: 1}

#Ordenar el número de comentarios de forma descendente.

{num_mflix_comments:-1}

#Limitar los resultados a 1.

#¿Cuál es título de las cinco películas más comentadas?

Para responder esta pregunta, necesitamos tres cosas.

#Proyectar el título de las películas.

{title: 1}

#Ordenar el número de comentarios de forma descendente.

{num_mflix_comments: -1}

#Limitar los resultados a 5.

Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 5

#SESION 5

#RETO 1

#Usando la base de datos sample_airbnblistingsAndReviews, realiza los siguientes filtros:

#Propiedades que no permitan fiestas.

{house_rules: /No Parties/i}

#Propiedades que admitan mascotas.

{house_rules: /Pets Allowed/i}

#Propiedades que no permitan fumadores.

{house_rules: /No Smoking/i}

#Propiedades que no permitan fiestas ni fumadores.

{house_rules: /No Smoking|No Parties/i}

#RETO 2

#Usando la colección sample_airbnb.listingsAndReviews, agrega un filtro que permita obtener todas las publicaciones que tengan 50 o más comentarios, que la valoración sea mayor o igual a 80, que cuenten con conexión a Internet vía cable y estén ubicada en Brazil.

{number_of_reviews: {$gte: 50},

"review_scores.review_scores_rating":

{$gte: 80}, amenities: {$in: [/Ethernet/]},

"address.country_code": "BR" }

Luis David Espinosa Caldera #data-analysis-gdl-20-07

 # RETO 3

#Usando la colección sample_airbnb.listingsAndReviews, mediante el uso de agregaciones, encontrar el número de publicaciones que tienen conexión a Internet, sea desde Wifi o desde cable (Ethernet).

#Primero filtramos los documentos con Internet desde Wifi o desde cable. Para ello usamos $match que permite realizar filtros dentro de agregaciones.

```
{
  amenities: {$in: ["Wifi", "Ethernet"]}
}
```

#Ahora contamos el número de registros resultantes con $group. Los agrupamientos al igual que en SQL necesitan un campo por el cual agrupar y una función de agrupamiento.

Dado que contaremos los registros no necesitamos campo, así que ponemos _id: null.

Para agrupar usaremos la función $sum con el parámetro 1. Es decir, por cada documento sumará

un 1, trayendo al final el total de documentos.

```
{
  _id: null,
  total: {
    $sum: 1
  }
}
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 6

#SESION 6

#RETO 1

#Con base en el ejemplo 1, modifica el agrupamiento para que muestre el costo promedio por habitación por país de las propiedades de tipo casa.

#Filtramos las propiedades con $match

```
{
  property_type: 'House',
  bedrooms: {$gte: 1}
}
```

#Agregamos el costo por recámara con $addFields

```
{
  costo_recamara: {$divide: ["$price", "$bedrooms"]}
}
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#Agrupamos la suma de recámaras y del total agrupando en este caso por país. Para ello usamos $group.

```
{
  _id: "$address.country",
  recamaras: {
    $sum: 1
  },
  total: {
    $sum: "$costo_recamara"
  }
}
```

#Agregamos el campo costo promedio para cada país con $addFields, creamos un alias al _id para hacer más claro el valor que guarda.

```
{
  pais: "$_id",
  costo_promedio: {$divide: ["$total", "$recamaras"]}
}
```

#Agregamos una proyección para quitar campos

irrelevantes con project.

```
{
  _id:0,
  pais:1,
  costo_promedio:1
}
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

#RETO 2

#Usando las colecciones comments y users, se requiere conocer el correo y contraseña de cada persona que realizó un comentario. Construye un pipeline que genere como resultado estos datos.

#Primero, obtenemos la relación con $lookup.

```
{
  from: 'users',
  localField: 'name',
  foreignField: 'name',
  as: 'usuario'
}
```

#Posteriormente, obtenemos el objeto del arreglo, su campo password y finalmente proyectamos los datos necesarios.

```
$addFields
{
  usuario_objeto: {$arrayElemAt: ["$usuario", 0]}
}

$addFields
{
  usuario_password: "$usuario_objeto.password"
}

$project
{
  _id:0,
  name:1,
  email:1,
  usuario_password:1
}
```

Luis David Espinosa Caldera #data-analysis-gdl-20-07

Trabajo en clase – Sesión 7

#SESION 7

#RETO 1

#Definir los campos y tipos de datos para la tabla
movies haciendo uso de los archivos movies.dat y
README.

Primero se revisan los registros abriendo el archivo.

Y luego se revisa la documentación (archivo README)

MOVIES FILE DESCRIPTION

Movie information is in the file "movies.dat" and is in the following
format:

MovieID::Title::Genres

- Titles are identical to titles provided by the IMDB (including
year of release)

- Genres are pipe-separated and are selected from the following genres:

    * Action
    * Adventure
    * Animation
    * Children's
    * Comedy
    * Crime
    * Documentary
    * Drama
    * Fantasy
    * Film-Noir
    * Horror

Así que se definen los siguientes campos y tipo para crear la tabla movies en SQL:

id INT PRIMARY KEY

title VARCHAR(80)

generos VARCHAR(80)

#Crear la tabla movies (recuerda usar el mismo nombre

del archivo sin la extensión para vincular nombres

de tablas con archivos).

```
CREATE TABLE IF NOT EXISTS movies (
  id INT PRIMARY KEY,
  title VARCHAR(80),
  generos VARCHAR(80)
);
```

#Definir los campos y tipos de datos para la tabla

ratings haciendo uso de los archivos ratings.dat y

README.

Ahora se revisan la estructura registros abriendo el archivo restante.

Y luego se revisa la documentación (archivo README)

RATINGS FILE DESCRIPTION

All ratings are contained in the file "ratings.dat" and are in the

following format:

UserID::MovieID::Rating::Timestamp

- UserIDs range between 1 and 6040
- MovieIDs range between 1 and 3952
- Ratings are made on a 5-star scale (whole-star ratings only)
- Timestamp is represented in seconds since the epoch as returned by time(2)
- Each user has at least 20 ratings

USERS FILE DESCRIPTION

User information is in the file "users.dat" and is in the following

format:

UserID::Gender::Age::Occupation::Zip-code

All demographic information is provided voluntarily by the users and is

...

Así que se definen los siguientes campos y tipo para crear la tabla ratings en SQL:

userid INT

movieid INT

rating INT

time_stamp BIGINT

#Crear la tabla ratings

```
CREATE TABLE IF NOT EXISTS ratings (
  userid INT,
  movieid INT,
  rating INT,
  time_stamp BIGINT
);
```