

Proyecto Final

Segunda entrega

1. Vistas

Vista 1: tb_agrupodo_por_mes_gastos_clientes

Descripción: es una vista que permite visualizar, el total de gastos realizado de cada cliente por mes, mostrando, id_cliente, nombre, apellido.

Objetivo: Visualizar el total de gastos mensuales por cliente.

Tablas/Datos: Las Tablas utilizadas son tb_clientes, tb_citas, tb_citas_serv y tb_servicios.

Las cuales son unidas a través de Join, haciendo las siguientes combinaciones:

- Tb_clientes unida con Tb_citas a través del campo "id_clientes".
- Tb_citas unida a Tb_citas_serv a través del campo "id_citas".
- Tb_citas_serv unida a Servicios a través del campo "id_servicios"

Luego usamos un Group by por el campo "Id_clientes" de la Tb_clientes y por mes usando la función "MONTHNAME" en el campo "Fecha" de la Tb_citas

Los datos de cada tabla que se utilizan con sus respectivas funciones para esta vista son:

- `peluqueria`.`clientes`.`id_Clientes` AS `id_Clientes`,
- `peluqueria`.`clientes`.`Nombre` AS `Nombre`,
- `peluqueria`.`clientes`.`Apellido` AS `Apellido`,
- MONTHNAME(`peluqueria`.`citas`.`Fecha`) AS `MES`,
- SUM(`peluqueria`.`servicios`.`Precio`) AS `Total_Gastado`

Vista 2: tb_cantidad_citas_clientes

Descripción: es una vista creada para poder visualizar la cantidad de citas que tienen cada cliente

Objetivo: poder visualizar la cantidad de citas que tienen cada cliente

Tablas/Datos: Las tablas utilizadas son:

- Clientes
- Citas

Realizamos un Join con la tabla "Citas", para poder contar las citas de cada cliente.

Luego usamos un Group by donde agrupamos por los siguientes campos y sus tablas:

- `clientes`.`id_Clientes` ,

- `clientes`.`Nombre` ,
- `clientes`.`Apellido`

Los datos utilizados con sus respectivas funciones son los siguientes:

- `peluqueria`.`clientes`.`id_Clientes` AS `id_Clientes`,
- `peluqueria`.`clientes`.`Nombre` AS `Nombre`,
- `peluqueria`.`clientes`.`Apellido` AS `Apellido`,
- COUNT(`peluqueria`.`citas`.`id_Citas`) AS `count(citas.id_Citas)`

Vista3: tb_clientes_edad

Descripción: Es una lista que permite visualizar la edad de cada uno de los clientes.

Objetivo: Generar una vista que rápidamente permita ver los clientes más longevos.

Tablas/Datos: La tabla que se utiliza para generar la vista es Clientes, utilizando las siguientes columnas:

- `peluqueria`.`clientes`.`id_Clientes` AS `id_clientes`,
- `peluqueria`.`clientes`.`Nombre` AS `Nombre`,
- `peluqueria`.`clientes`.`Apellido` AS `Apellido`,
- (YEAR(NOW()) - YEAR(`peluqueria`.`clientes`.`Fecha_Nacimiento`)) AS `Edad`

Generamos un campo llamado “Edad” realizando una operación que consiste en restar el año actual (obtenido usando la función Now() combinado con Year() para obtener solo el año) menos el año recibido en el campo “Fecha_Nacimiento” donde usamos la función Year() para obtener solo el año.

Vista 4: tb_detalle_gastos_clientes

Descripción: Es una vista que permite visualizar un detalle de gastos en los servicios por cliente, donde al estar ordenado de mayor a menor (Desc) por el campo “precio” de la tabla de “servicios”, podemos ver los cliente que gastaron más en servicios..

Objetivo: tener un detalle de los gastos de cada cliente en los servicios de la peluquería

Tablas/Datos: Las Tablas utilizadas son Clientes, Citas, Citas_serv y Servicios.

Las cuales son unidas a través de Join, haciendo las siguientes combinaciones:

- Tb_clientes unida con Tb_citas a través del campo “id_clientes”.
- Tb_citas unida a Tb_citas_serv a través del campo “id_citas”.
- Tb_citas_serv unida a Servicios a través del campo “id_servicios”

Los datos de cada tabla que se utilizan con sus respectivas funciones para esta vista son:

- `clientes`.`id_Clientes` AS `id_Clientes`,
- `clientes`.`Nombre` AS `nombre`,
- `clientes`.`Apellido` AS `Apellido`,
- `servicios`.`Nombre` AS `Servicio`,
- `citas`.`Fecha` AS `Fecha`,
- `servicios`.`Precio` AS `Precio`

Vista 5: tb_total_compras_x_proveedor

Descripción: Es una vista que nos permite ver el total de compras realizado por mes a cada proveedor.

Objetivo: Nos permite ver el total de compras realizado por proveedor, para identificar fácilmente el proveedor con el que más estamos invirtiendo.

Tablas/Datos: Las Tablas utilizadas son Proveedores y Detalles_Compras. Las cuales son unidas a través de Join a través del campo "id_proveedores".

Los datos de cada tabla que se utilizan con sus respectivas funciones para esta vista son:

- `proveedores`.`Nombre` AS `Nombre`,
- MONTHNAME(`detalles_compras`.`Fecha_Compra`) AS `MES`,
- SUM(`detalles_compras`.`Subtotal`) AS `Total_Compras`

Genero un campo "Mes" para obtener el nombre del mes, aplicando la función "MONTHNAME" en el campo Fecha_Compras de la tabla "Detalles_Compras" y también, generamos otro campo llamado "Total_Compras", en el cual aplicamos la función SUM() al campo Subtotal de la tabla de "detalles_compras", de este modo obtenemos el total de compras en dinero.

Además, realizo un Group by por los siguientes campos:

- `Proveedores`.`Nombre`
- MONTHNAME(`detalles_compras`.`Fecha_Compra`)

2. Funciones

Función 1: Calcular_edad

Descripción: En este caso realicé una función que me permite calcular la edad enviando la fecha de nacimiento, lo cual devuelve un valor entero indicando la edad o mejor dicho la diferencia entre la fecha enviada y la fecha actual.

Objetivo: Permitirme obtener la edad de los clientes o empleados sabiendo la fecha de nacimiento o incluso, poder usarla para medir la antigüedad de un cliente o empleado también.

Tablas/Datos: realice el siguiente calculo en la función resultado = (YEAR(NOW()) - YEAR(fecha_recibida));

Función 2: Calcular_Descuento

Descripción: En este caso es una función que recibe 2 valores float, donde uno de los valores será el precio y el otro el porcentaje de descuento.

Objetivo: obtener fácilmente el precio de un producto aplicando el descuento que necesitemos a un producto y donde además ya tenga incluido el IVA también (por defecto la función tiene un 0.1 de IVA).

Tablas/Datos: realizamos el siguiente calculo con las variables float que recibimos en la función.

*Precio_con_Descuento= valor_producto + iva - (valor_producto * porcentaje_descuento);*

Return Precio_con_Descuento;

3. Stored Procedures

Procedure1: Ingresar_New_Clientes:

Descripción: permite poder realizar un Insert a la tabla Clientes, donde exige para su carga que los campos indicados sean distintos de vacíos, de poder ejecutarse con éxito la carga, devolverá una variable que tendrá un mensaje indicando que la carga fue exitosa y además se ejecuta una consulta a la tabla "clientes" ordenada por id_cliente de manera Desc con lo cual podremos ver el registro cargado. Ahora bien si algunos de los campos ingresados está vacío, obtendremos en la variable de salida un mensaje indicando que la carga no fue exitosa y que tienen que completar todos los campos.

Objetivo: Poder realizar insert en la tabla cliente de manera mas simple.

Tablas/Datos:

Utilizamos la tabla clientes y solicitamos los siguientes campos en el stored procedure para le **Insert** de nuevos clientes:

- **IN** id_dni varchar(20),
- **IN** nomb varchar(50),

- **IN** apell varchar(50),
- **IN** correo varchar(100),
- **IN** f_ingreso date,
- **IN** f_nacimiento date,
- **OUT** Respuesta varchar(100)

Para poder asegurarnos que las fechas ingresadas por el usuario en los campos “f_nacimiento” y “f_ingreso” tenga el formato de mysql, utilizamos la función **str_to_date()** para convertir la misma en el formato correcto.

Procedure2: ORDENAR_PRODUCTOS

Descripción: Permite ordenar los productos según el campo que le pasemos y lo va hacer de manera ascendente o Descendente, dependiendo del segundo valor que pasemos siendo “A” para Ascendente y “D” para Descendente, de compartir cualquier otro valor tomara e valor por defecto de la consulta. Resuelto el stored procedures mostrara la tabla productos según los valores que hayamos compartido.

Objetivo: poder obtener de manera rápida los datos de la tabla producto ordenados según la necesidad que tengamos.

Tablas/Datos: Utilizamos la tabla Productos y solicitamos los siguientes campos en el stored procedures para poder hacer el ordenamiento:

- Campo_tabla (compo de la tabla de productos por el cual queremos ordenar)
- tipo (tipo de ordenamiento A o D)

4. Archivos SQL

Script creación de objetos y Script inserción de datos:

Link de Gut Hub, donde podrán encontrar todos los Script:

https://github.com/gabrielcan/ProyectoFinal_Curso_SQL_CoderHouse.git