/\* \*/

create database proyectoTecknicStore ;

use proyectoTecknicStore ;

create table TipoUsuario (

idTipoUsuario int not null auto\_increment,

TipoUsuario VARCHAR(45) not null ,

primary key(idTipoUsuario))

;

create table TipodeDocumento (

idTipodeDocumento int not null auto\_increment,

TipoDocumento VARCHAR(45) not null ,

primary key(idTipodeDocumento))

;

create table Persona (

idPersona int not null auto\_increment,

pNombre VARCHAR(45) not null ,

sNombre VARCHAR(45) NULL,

pApellido VARCHAR(45) not null ,

sApellido VARCHAR(45) NULL,

Telefono VARCHAR(45) not null ,

Documento VARCHAR(45) not null ,

FechaNacimiento DATE NULL,

TelefonoAux VARCHAR(45) NULL,

TipoDcoumento INT not null ,

primary key(idPersona),

foreign key (TipoDcoumento)

references TipodeDocumento (idTipodeDocumento)

)

;

create table Uusario (

idUsuario INT not null ,

Correo VARCHAR(45) not null ,

Clave varbinary (255) not null ,

UsarName VARCHAR(45) not null ,

TipoUsuario INT not null ,

primary key(idUsuario),

foreign key (TipoUsuario)

references TipoUsuario (idTipoUsuario),

constraint fkUusarioPersona1

foreign key (idUsuario)

references Persona (idPersona)

)

;

CREATE TABLE Administrador (

Persona\_idPersona INT NOT NULL,

Uusario\_idUsuario INT NOT NULL,

PRIMARY KEY (Persona\_idPersona),

FOREIGN KEY (Persona\_idPersona) REFERENCES Persona (idPersona),

FOREIGN KEY (Uusario\_idUsuario) REFERENCES Uusario (idUsuario));

create table TipoCambio (

idTipoCambio int not null auto\_increment,

TipoCambio VARCHAR(45) not null ,

primary key(idTipoCambio))

;

create table RegistroCambios (

idRegistroCambios int not null auto\_increment,

Fecha DATE not null ,

TipoCambio INT not null ,

IdAdministrador INT not null ,

primary key(idRegistroCambios),

constraint fkRegistroCambiosTipoCambio2

foreign key (TipoCambio)

references TipoCambio (idTipoCambio),

constraint fkRegistroCambiosAdministrador2

foreign key (IdAdministrador)

references Administrador (Persona\_idPersona));

create table CargosTecnico (

idCargosTecnico int not null auto\_increment,

CargosTecnico VARCHAR(45) not null ,

primary key(idCargosTecnico));

create table Tecnico (

IdTecnico INT not null ,

CargosTecnicoidCargosTecnico INT not null ,

idUsuarioTecnico INT not null ,

primary key(IdTecnico),

constraint fkTecnicoCargosTecnico2

foreign key (CargosTecnicoidCargosTecnico)

references CargosTecnico (idCargosTecnico),

constraint fkTecnicoPersona1

foreign key (IdTecnico)

references Persona (idPersona),

constraint fkTecnicoUusario1

foreign key (idUsuarioTecnico)

references Uusario (idUsuario)

);

create table Marcas (

idMarcas int not null auto\_increment,

Marcas VARCHAR(45) not null ,

primary key(idMarcas));

create table TipoEquipo (

idTipoEquipo int not null auto\_increment,

TipoEquipo VARCHAR(45) not null ,

primary key(idTipoEquipo));

create table Equipo (

idEquipo int not null auto\_increment,

descInconveniente VARCHAR(600) not null ,

idMarca INT not null ,

idTipoEquipo INT not null ,

idUusario INT not null ,

primary key(idEquipo),

constraint fkEquipoMarcas2

foreign key (idMarca)

references Marcas (idMarcas),

constraint fkEquipoTipoEquipo1

foreign key (idTipoEquipo)

references TipoEquipo (idTipoEquipo));

create table TipoServicio (

idTipoServicio int not null auto\_increment,

TipoServicio VARCHAR(45) not null ,

primary key(idTipoServicio));

create table Servicio (

idServicio int not null auto\_increment,

descDaño VARCHAR(45) not null ,

ServicioTerminado VARCHAR(45) not null ,

idEquipoaReparar INT not null ,

IdTecnicoEncargado INT not null ,

idDueñoServicio INT not null ,

TipoServicioidTipoServicio INT not null ,

primary key(idServicio),

constraint fkServicioEquipo2

foreign key (idEquipoaReparar)

references Equipo (idEquipo),

constraint fkServicioTecnico2

foreign key (IdTecnicoEncargado)

references Tecnico (IdTecnico),

constraint fkServicioTipoServicio1

foreign key (TipoServicioidTipoServicio)

references TipoServicio (idTipoServicio)

);

create table Factura (

idFactura int not null auto\_increment,

idServicio INT not null ,

idDueñoFactura INT not null ,

FechaFactura DATE not null ,

CostoTotal VARCHAR(45) not null ,

TiempoGarantia VARCHAR(45) not null ,

primary key(idFactura),

constraint fkFacturaServicio2

foreign key (idServicio)

references Servicio (idServicio)

);

create table DetalleFactura (

idDetalleFactura int not null auto\_increment,

idServicioSolicitado INT not null ,

CostoUnitario VARCHAR(45) not null ,

idFactura INT not null ,

primary key(idDetalleFactura),

constraint fkDetalleFacturaTipoServicio1

foreign key (idServicioSolicitado)

references TipoServicio (idTipoServicio),

constraint fkDetallefacturaFactura2

foreign key (idFactura)

references factura (idFactura)

);

USE proyectoTecknicStore ;

/\*\*----------------------------------------------------------------------------------------

/\*\*-----------------------------------------------------------------------------INSERTS--------------------------------------------\*\*/

/\*-----------------------------------------------------------------------------------------\*/

-- Tabla tipousuario

INSERT INTO tipousuario (TipoUsuario)

VALUES

('Tecnico'),

('Administrador'),

('Cliente');

-- Tabla tipodedocumento

INSERT INTO tipodedocumento (idTipodeDocumento, TipoDocumento)

VALUES

(1, 'CC'),

(2, 'TI'),

(3, 'CE'),

(4, 'Pasaporte');

-- Tabla Persona

INSERT INTO Persona (pNombre, sNombre, pApellido, sApellido, Telefono, Documento, FechaNacimiento, TelefonoAux, TipoDcoumento)

VALUES

('Juan', 'Carlos', 'Gomez', 'Lopez', '123-456-7890', '123456789', '1990-05-15', '987-654-3210', 1),

('Maria', 'Isabel', 'Rodriguez', 'Perez', '987-654-3210', '987654321', '1985-12-10', '555-555-5555', 2),

('Pedro', NULL, 'Fernandez', NULL, '555-123-4567', '555123456', '1995-08-20', NULL, 1),

('Ana', 'Maria', 'Lopez', 'Gutierrez', '777-888-9999', '777888999', '1980-03-25', '333-333-3333', 3),

('Luis', NULL, 'Martinez', NULL, '444-555-6666', '444555666', '1998-11-05', NULL, 2);

-- Tabla uusario

SET @clave = 'sadeffa';

INSERT INTO uusario (idUsuario, Correo, Clave, UsarName, TipoUsuario)

VALUES

(1, 'usuario1@ejemplo.com', AES\_ENCRYPT('123456', @clave), 'usuario1', 1),

(2, 'usuario2@ejemplo.com', AES\_ENCRYPT('67890', @clave), 'usuario2', 2),

(3, 'usuario3@ejemplo.com', AES\_ENCRYPT('54321', @clave), 'usuario3', 1),

(4, 'usuario4@ejemplo.com', AES\_ENCRYPT('98765', @clave), 'usuario4', 2),

(5, 'usuario5@ejemplo.com', AES\_ENCRYPT('24680', @clave), 'usuario5', 3);

SELECT Correo, cast(AES\_DECRYPT(Clave, @clave)as char) AS Clave\_desencriptado

FROM uusario;

-- Tabla tipocambio

INSERT INTO tipocambio (idTipoCambio, TipoCambio)

VALUES

(1, 'deshabilitar usuario'),

(2, 'cambiar estado del'),

(3, 'cambiar descripcion de un servicio'),

(4, 'eliminar un usuario'),

(5, 'eliminar un producto o servicio');

-- Tabla marcas

INSERT INTO marcas (idMarcas, Marcas)

VALUES

(1, 'Android'),

(2, 'PlayStation'),

(3, 'Nokia'),

(4, 'Samsung'),

(5, 'Huawei');

-- Tabla tipoequipo

INSERT INTO tipoequipo (idTipoEquipo, TipoEquipo)

VALUES

(1, 'Pc'),

(2, 'Consola'),

(3, 'Pc-Gamer'),

(4,'Celular'),

(5, 'Consola de audio');

-- Tabla equipo

INSERT INTO equipo (idEquipo, descInconveniente, idMarca, idTipoEquipo, idUusario)

VALUES

(1, 'tiene un daño', 1, 5, 1),

(2, 'tiene un daño', 2, 4, 2),

(3, 'tiene un daño', 3, 3, 3),

(4, 'tiene un daño', 4, 2, 4),

(5, 'tiene un daño', 5, 1, 5);

-- Tabla tiposervicio

INSERT INTO tiposervicio (idTipoServicio, TipoServicio)

VALUES

(1,'Reparación de hardware'),

(2,'Reparación de software'),

(3,'Actualización de software'),

(4,'Mantenimiento preventivo'),

(5,'Servicio de consultoría');

-- Tabla cargostecnico

INSERT INTO cargostecnico (idCargosTecnico, CargosTecnico)

VALUES

(1,'Técnico de hardware'),

(2,'Técnico de software'),

(3,'Especialista en redes'),

(4,'Ingeniero de sistemas'),

(5,'Técnico de reparación de dispositivos móviles');

-- Tabla Tecnico

INSERT INTO Tecnico (IdTecnico, CargosTecnicoidCargosTecnico, idUsuarioTecnico)

VALUES

(1, 1, 1),

(2, 2, 2),

(3, 1, 3),

(4, 3, 4),

(5, 4, 5);

-- Tabla Servicio

INSERT INTO Servicio (descDaño, ServicioTerminado, idEquipoaReparar, IdTecnicoEncargado, idDueñoServicio, TipoServicioidTipoServicio)

VALUES

('Problema de pantalla, Reparado', '1', 1, 2, 1, 2),

('Problema de batería, En espera', '2', 2, 3, 2, 3),

('No enciende, Reparado', '3', 1, 4, 1, 1),

('Problema de sonido, En proceso', '4', 3, 5, 3, 4),

('Pérdida de datos, Reparado', '5', 2, 1, 4, 5);

INSERT INTO administrador (Persona\_idPersona, Uusario\_idUsuario) VALUES (1, 5),

(2, 4),

(3, 3),

(4, 2),

(5, 1);

-- Tabla RegistroCambios

INSERT INTO RegistroCambios (Fecha, TipoCambio, IdAdministrador)

VALUES

('2023-09-15', 1, 1),

('2023-09-14', 2, 2),

('2023-09-13', 1, 3),

('2023-09-12', 3, 4),

('2023-09-11', 2, 5);

-- Tabla Factura

INSERT INTO Factura (idServicio, idDueñoFactura, FechaFactura, CostoTotal, TiempoGarantia)

VALUES

(1, 1, '2023-09-15', '100.00', '30 días'),

(2, 2, '2023-09-14', '75.50', '60 días'),

(3, 3, '2023-09-13', '200.00', '90 días'),

(4, 4, '2023-09-12', '50.25', '45 días');

-- Tabla DetalleFactura

INSERT INTO DetalleFactura (idServicioSolicitado, CostoUnitario, idFactura)

VALUES

(1, 50.000, 1),

(2, 25.500, 2),

(3, 75.000, 3),

(4, 20.250, 4);

/\*--------------------------------------------------------------------------------

/\*\*------------------------------------------------------------------CONSULTAS----------------------------------------------------------------\*\*/

/\*--------------------------------------------------------------------------------\*/

use proyectotecknicstore;

/\*\*obtener los cambios realizados por un administrador especifico con su nombre, tipo de cambio, fecha y numero de doc\*\*/

select rc.idAdministrador as ID\_del\_admin, tc.TipoCambio as Cambio\_Realizado, concat(p.pNombre, " ", p.pApellido) as Nombre\_admin, p.Documento as Documento\_del\_admin, rc.Fecha as Fecha\_del\_cambio

from RegistroCambios rc

join Administrador a on rc.IdAdministrador = a.Persona\_idPersona

join Persona p on a.Persona\_idPersona = p.idPersona

join TipoCambio tc on rc.TipoCambio = tc.idTipoCambio;

/\* ejemplo con usuario especifico

where a.Uusario\_idUsuario = 1;

\*/

/\*\*Listar los Servicios con sus detalles\*\*/

select S.idServicio, S.descDaño, S.ServicioTerminado, TS.TipoServicio

from Servicio S

join TipoServicio TS on S.TipoServicioidTipoServicio = TS.idTipoServicio

left join DetalleFactura DF on S.idServicio = DF.idServicioSolicitado;

/\*\*Listar los Tecnicos con su cargo\*\*/

select concat(p.pNombre, " ", p.pApellido) as Nombre\_tecnico, p.Documento as Documento\_del\_tecnico, ct.CargosTecnico

from persona p

join Tecnico t on p.idPersona = t.idTecnico

join CargosTecnico ct on t.CargosTecnicoidCargosTecnico = ct.idCargosTecnico;

/\*\*listar las facturas entre una fecha especifica a otra\*\*/

select idFactura, FechaFactura, CostoTotal, TiempoGarantia

from Factura

where FechaFactura between '2023-09-01' and '2023-09-30';

/\*\*el tecnico con mayor numero de servicios y la cantidad\*\*/

select concat(p.pNombre, " ", p.pApellido) as Nombre\_Tecnico, count(s.idServicio) as Cantidad\_Servicios\_Completados

from Técnico t

join Persona p on t.IdTecnico = p.idPersona

join Servicio s on t.IdTecnico = s.IdTecnicoEncargado

where t.IdTecnico = ( select IdTecnicoEncargado

from Servicio

group by IdTecnicoEncargado

order by COUNT(\*) desc

limit 1

)

group by t.IdTecnico;

/\*-----------------------------------------CONSULTAS LOREN ----------------------------------------------------------------\*\*/

/\*--------------------------------------------------------------------------------\*/

/\*Consulta para encontrar el número de servicios realizados por cada tipo de servicio::p\*/

select TipoServicio.TipoServicio, COUNT(\*) as NumServicios

from Servicio

join TipoServicio ON Servicio.TipoServicioidTipoServicio = TipoServicio.idTipoServicio

group by TipoServicio.TipoServicio;

/\*Consulta para tener los servicios reparados con sus facturas correspondientes\*/

select Servicio.descDaño, Factura.CostoTotal

from Servicio

join Factura on Servicio.idServicio = Factura.idServicio

where Servicio.ServicioTerminado = 1;

/\*Consulta para encontrar todos los servicios pendientes que estan en reparación y sus descripciones de inconvenientes:\*/

select Servicio.descDaño, Servicio.ServicioTerminado

from Servicio

where Servicio.ServicioTerminado = 2;

/\*encontrar los clientes que han realizado más de 3 servicios\*/

select Persona.pNombre, Persona.sNombre, Persona.pApellido, Persona.sApellido, count(Servicio.idServicio) as NumServicios

from Persona

join Servicio on Persona.idPersona = Servicio.idDueñoServicio

where Persona.TipoDcoumento = 1

group by Persona.idPersona

having count(Servicio.idServicio) < 3;

/\*encontrar los servicios con la fecha de inicio y finalización\*/

select Servicio.idServicio, Servicio.descDaño, Servicio.ServicioTerminado, RegistroCambios.Fecha as FechaInicio

from Servicio

join RegistroCambios on Servicio.idServicio = RegistroCambios.IdAdministrador

where RegistroCambios.TipoCambio = 3;

/\*Obtener los tipos de servicio más visitados (los que se han solicitado más veces)\*/

select TipoServicio.TipoServicio, count(Servicio.idServicio) as VecesSolicitado

from TipoServicio

join Servicio on TipoServicio.idTipoServicio = Servicio.TipoServicioidTipoServicio

group by TipoServicio.TipoServicio

order by VecesSolicitado DESC

LIMIT 5;

/\*--------------------------------------------------------------------------------

/\*\*-------------------------CONSULTAS ALEX----------------------------------\*\*/

/\*Consultas\*/

/\* 1) Se requiere listar los servicios prestados hasta el momento junto con sus detalles.

En este caso el tipo de servicio y cuál fue el costo total de este y diferenciandose que es un precio en pesos "COLOMBIANOS (COP)". Para analizar cuáles son los servicios más prestados y el precio promedio. \*/

select Servi.idServicio, Servi.descDaño, Servi.ServicioTerminado, TipoServi.TipoServicio, concat('$', format(ifnull(sum(DetaFac.CostoUnitario),0),3),' COP') as "Costo total"

from Servicio Servi

join TipoServicio TipoServi on Servi.TipoServicioidTipoServicio = TipoServi.idTipoServicio

left join DetalleFactura DetaFac on Servi.idServicio = DetaFac.idServicioSolicitado

group by Servi.idServicio, Servi.descDaño, Servi.ServicioTerminado, TipoServi.TipoServicio;

/\* 2) Encontrar los servicios cuya descripción de daño contenga

“Reparado” y mostrar la descripción de daño y la fecha de inicio del cambio, con el fin de analizar cuántos equipos se han reparado y promediar por cuantos dias se repara un equipo\*/

select Servicio.descDaño, RegistroCambios.Fecha as "Fecha cambio"

from Servicio

join RegistroCambios on Servicio.idServicio = RegistroCambios.IdAdministrador

where Servicio.descDaño like "%Reparado%" ;

/\*3) Obtener informacion detallada sobre los servicios,incluyendo el nombre del cliente,

el tipo de servicio, la marca del equipo en reparacion la descripcion del daño etc.

\*/

select concat(person.pNombre," ", person.sNombre," ", person.pApellido," ", person.sApellido) as "nombre completo del cliente", servi.idServicio, marca.Marcas as "marca del equipo",servi.descDaño as "descripcion daño del equipo", servi.ServicioTerminado as "estado servicio" ,tiposervi.TipoServicio as "tipo de servicio"

from Servicio servi

join TipoServicio tiposervi on servi.TipoServicioidTipoServicio = tiposervi.idTipoServicio

join Equipo equip on servi.idEquipoaReparar = equip.idEquipo

join Marcas marca on equip.idMarca = marca.idMarcas

join Tecnico tecnic on servi.IdTecnicoEncargado = tecnic.IdTecnico

join Persona person on servi.idDueñoServicio = person.idPersona;

/\*4) se necesita los nombres de los tecnicos que han realizado servicios

de reparacion de hardware y han cobrado mas de $49.999

\*/

select perso.pNombre, perso.sNombre, perso.pApellido, perso.sApellido, tiposers.TipoServicio, concat('$', format(cast(detafac.CostoUnitario as decimal(10, 3)), 3), ' COP') as CostoUnitario

from Persona perso

inner join Tecnico tecnic on perso.idPersona = tecnic.IdTecnico

inner join Servicio servi on tecnic.IdTecnico = servi.IdTecnicoEncargado

inner join TipoServicio tiposers on servi.TipoServicioidTipoServicio = tiposers.idTipoServicio

inner join Factura fac on servi.idServicio = fac.idServicio

inner join DetalleFactura detafac on fac.idFactura = detafac.idFactura

where tiposers.TipoServicio = 'Reparación de hardware'

and cast(detafac.CostoUnitario as decimal(10, 3)) >= 50.000;

/\*5)

/\*--------------------------------------------------------------------------------

/\*\*------------------------------------------------------------------CONSULTAS LUIS ----------------------------------------------------------------\*\*/

/\*--------------------------------------------------------------------------------\*/

-- Ejercicio 1 --

-- Encontrar el nombre de las personas que ha gastado más dinero en servicios, junto con el total gastado--

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

-- EXPLICACION CODIGO

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

-- (CONCAT) concatena el nombre y el apellido de la persona en una única cadena, y se asigna el alias.

-- (SUM )calcula la suma total de los costos totales de las facturas relacionadas con los servicios que ha utilizado cada persona.

-- (FROM) se establece la tabla Persona como la tabla principal para la consulta y se le asigna el alias p para abreviar su uso en la consulta.

-- (INNER JOIN) --Esto establece una unión interna con la tabla "uusario , Sevicio y factura"

-- (GROUP BY ) -- agrupa los resultados por la columna idPersona de la tabla Persona. Esto significa que obtendrás una fila de resultados para cada persona.

-- (ORDER BY) -- Ordena los resultados en orden descendente según la columna TotalGastado

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

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreCliente , SUM(f.CostoTotal) AS TotalGastado

FROM Persona p

INNER JOIN uusario u ON p.idPersona = u.idUsuario

INNER JOIN Servicio s ON p.idPersona = s.idDueñoServicio

INNER JOIN Factura f ON s.idServicio = f.idServicio

GROUP BY p.idPersona

ORDER BY TotalGastado DESC

;

-- Ejercicio 2--

-- Buscar el tipo de servicio más común entre los servicios que aún no se han terminado --

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

-- EXPLICACION CODIGO

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

-- (SELECT) Esta parte del código selecciona dos columnas

-- (COUNT(\*)) contamos cuántas filas cumplen con las condiciones especificadas en la consulta y asigna el alias "

-- (FROM) Se establece la tabla Servicio como la tabla principal para la consulta y se le asigna el alias s para abreviar su uso en la consulta.

-- (INNER JOIN) Aquí se realiza una unión interna con la tabla tiposervicio.

-- (WHERE ) dice que muestre las filas en las que el valor de la columna "ServicioTerminado" de la tabla "Servicio" sea igual a 2.

-- (GROUP BY) agrupa los resultados por la columna "TipoServicio" de la tabla "tiposervicio".

-- (ORDER BY)Ordena los resultados en orden descendente según la columna "Cantidad"

-- (LIMIT 1) Limita el resultado a solo una fila, lo que significa que obtendrás el tipo de servicio con la mayor cantidad de registros.

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

SELECT ts.TipoServicio, COUNT(\*) AS Cantidad

FROM Servicio s

INNER JOIN tiposervicio ts ON s.TipoServicioidTipoServicio = ts.idTipoServicio

WHERE s.ServicioTerminado = 2

GROUP BY ts.TipoServicio

ORDER BY Cantidad DESC

LIMIT 1;

-- Ejercicio 3

-- técnicos que han trabajado en al menos tres servicios y mostrar la cantidad de servicios en los que han trabajado:

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

-- EXPLICACION CODIGO

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

-- (SELECT) Selecciona los capos que deseamos mostrar .

-- (CONCAT) concatena el nombre y el apellido de la persona .

-- (COUNT) Cuenta la cantidad de registros en la tabla Servicio .

-- (FROM) Se establece la tabla "Tecnico" como la tabla principal para la consulta y se le asigna el alias "t" para abreviar su uso en la consulta.

-- (INNER JOIN) Se realiza una unión interna con la tabla Persona Y Usuario

-- (LEFT JOIN) Se realiza una unión izquierda con la tabla "Servicio"

-- (WHERE) Esta cláusula WHERE filtra las filas en las que el valor de la columna "TipoUsuario" por ejemplo tipo usuario= 3

-- (GROUP BY) agrupa los resultados por la columna "IdTecnico" de la tabla "Tecnico".

-- (HAVING ) se utiliza después del GROUP BY para filtrar los resultados del grupo.

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

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreTecnico, COUNT(s.IdTecnicoEncargado) AS CantidadServicios

FROM Tecnico t

INNER JOIN Persona p ON t.idUsuarioTecnico = p.idPersona

INNER JOIN uusario u ON p.idPersona = u.idUsuario

LEFT JOIN Servicio s ON t.IdTecnico = s.IdTecnicoEncargado

WHERE u.TipoUsuario = 3

GROUP BY t.IdTecnico

HAVING CantidadServicios <3 ;

-- ejercicio 4

-- Listar el nombre de las personas que han realizado servicios de reparación de hardware

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

-- EXPLICACION CODIGO

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

-- (SELECT) selecciona los campos para imprimir

-- (CONCAT) concatena el nombre y el apellido de la persona .

-- (FROM) Se establece la tabla Persona como la tabla principal.

-- (WHERE) establece una condición que debe cumplirse para que una fila de la tabla Persona se incluya en el resultado de la consulta.

-- La condición se basa en la comparación del valor de la columna "idPersona" en la tabla "Persona" con un conjunto de valores devueltos por una subconsulta.

-- (Subconsulta) La condición en esta subconsulta requiere que el valor de "TipoServicioidTipoServicio" en la tabla "Servicio" sea igual a 1

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

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreCliente

FROM Persona p

WHERE p.idPersona IN (

SELECT s.idDueñoServicio

FROM Servicio s

WHERE s.TipoServicioidTipoServicio = 1

);

-- Ejercicio 5

-- Obtener el nombre de los técnicos que tienen al menos un servicio en espera

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

-- EXPLICACION CODIGO

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

-- (SELECT) selecciona los campos para imprimir

-- (CONCAT) concatena el nombre y el apellido de la persona .

-- (FROM) Se establece la tabla Persona como la tabla principal.

-- (INNER JOIN)Se realiza una unión interna (INNER JOIN) entre la tabla "Persona" y el resultado de una subconsulta llamada "ServiciosEspera".

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

-- SUB CONSULTA

-- Selecciona el identificador del técnico encargado de un servicio, representado por "IdTecnicoEncargado".

-- COUNT(\*) La usamos contar el número de servicios en los que "ServicioTerminado" tiene un valor igual a 2 .

-- Le da un alias "ServiciosEnEspera" a la columna resultante que contiene el recuento de servicios en espera.

-- Establece que esta subconsulta se realiza en la tabla "Servicio".

-- Luego, se realiza una unión entre la tabla Persona p y el resultado de la subconsulta ServiciosEspera utilizando la igualdad entre "p.idPersona" y "ServiciosEspera.IdTecnicoEncargado

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

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

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreTecnico, ServiciosEnEspera

FROM Persona p

INNER JOIN (

SELECT IdTecnicoEncargado, COUNT(\*) AS ServiciosEnEspera

FROM Servicio

WHERE ServicioTerminado = 2

GROUP BY IdTecnicoEncargado

) ServiciosEspera ON p.idPersona = ServiciosEspera.IdTecnicoEncargado;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Consultas de Esteban Cardenas\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*contar la cantidad de servicios completados por cada cliente y mostrar sus nombres junto con la cantidad de servicios en una lista.\*\*/

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreCliente, COUNT(s.idServicio) AS CantidadServiciosCompletados

FROM Persona p

INNER JOIN Servicio s ON p.idPersona = s.idDueñoServicio

WHERE s.ServicioTerminado = 1

GROUP BY NombreCliente;

/\*\* identificar cuáles son los tipo de servicio que han sido completados o finalizados\*\*/

SELECT ts.TipoServicio, COUNT(\*) AS Cantidad

FROM Servicio s, TipoServicio ts

WHERE s.TipoServicioidTipoServicio = ts.idTipoServicio AND s.ServicioTerminado = 2

GROUP BY ts.TipoServicio

ORDER BY Cantidad ASC

LIMIT 1;

/\* Obtener informacion sobre los servicios,incluyendo el nombre del cliente,

el tipo de servicio, la marca del equipo en reparacion la descripcion del daño etc.

\*/

select concat(person.pNombre," ", person.sNombre," ", person.pApellido," ", person.sApellido) as "nombre completo del cliente", servi.idServicio, marca.Marcas as "marca del equipo",servi.descDaño as "descripcion daño del equipo", servi.ServicioTerminado as "estado servicio" ,tiposervi.TipoServicio as "tipo de servicio"

from Servicio servi

join TipoServicio tiposervi on servi.TipoServicioidTipoServicio = tiposervi.idTipoServicio

join Equipo equip on servi.idEquipoaReparar = equip.idEquipo

join Marcas marca on equip.idMarca = marca.idMarcas

join Tecnico tecnic on servi.IdTecnicoEncargado = tecnic.IdTecnico

join Persona person on servi.idDueñoServicio = person.idPersona;

/\*\*Mostrar las facturas junto con los nombres de los dueños de las facturas\*\*/

SELECT f.idFactura, CONCAT(p.pNombre, ' ', p.pApellido) AS NombreCliente

FROM Factura f

JOIN Persona p ON f.idDueñoFactura = p.idPersona;

/\*\*Mostrar los nombres de los técnicos y sus números de teléfono auxiliar (si tienen) en una sola columna\*\*/

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreTecnico,

IFNULL(p.TelefonoAux, 'No disponible') AS TelefonoAuxiliar

FROM Tecnico t

JOIN Persona p ON t.IdTecnico = p.idPersona;

/\*\*Mostrar los nombres completos de los técnicos junto con sus cargos\*\*/

SELECT CONCAT(p.pNombre, ' ', p.pApellido) AS NombreTecnico, ct.CargosTecnico AS Cargo

FROM Tecnico t

JOIN Persona p ON t.IdTecnico = p.idPersona

JOIN CargosTecnico ct ON t.CargosTecnicoidCargosTecnico = ct.idCargosTecnico;

/\*\*Listar todos los tipos de documentos junto con la cantidad de personas que tienen cada tipo de documento\*\*/

SELECT TipoDocumento, COUNT(\*) AS CantidadPersonas

FROM TipodeDocumento

JOIN Persona ON TipodeDocumento.idTipodeDocumento = Persona.TipoDcoumento

GROUP BY TipoDocumento;