**Práctica Nro. 5**

**Asignatura: Tecnologías de Información**

**Ejercicio:**

1. Considere el siguiente esquema relacional:

Empleado(idEmpleado,nombreEmpleado,fechaIngreso)

Servicio (idServicio, nombreServicio,valorServicio)

vehiculo(idVehiculo, marcaVehiculo, modeloVehiculo)

prestación\_servicio(id\_ps, idServicio,idEmpleado,idVehiculo,fecha)

FK: idServicio->servicio, idEmpleado->Empleado, idVehiculo->vehiculo

1. Usando SQL crear la tabla para el sistema de Prestación de Servicios del taller automotriz
   1. CREATE TABLE Servicio (
   2. idServicio int(4) UNSIGNED ZEROFILL NOT NULL,
   3. nombreServicio varchar(50),
   4. valorServicio int,
   5. PRIMARY KEY (idServicio)
   6. );

CREATE TABLE Empleado (

idEmpleado int(3) UNSIGNED ZEROFILL NOT NULL ,

nombreEmpleado varchar(50),

fechaIngreso date,

PRIMARY KEY (idEmpleado)

);

CREATE TABLE Vehiculo (

idVehiculo char(6) NOT NULL,

marcaVehiculo varchar(20),

modeloVehiculo varchar(20),

PRIMARY KEY (idVehiculo)

);

* 1. CREATE TABLE prestacionServicio (
     + 1. idPs int(8) AUTOINCREMENT,
       2. idServicio int(4) UNSIGNED ZEROFILL NOT NULL,

idEmpleado int(3) UNSIGNED ZEROFILL NOT NULL ,

* + - 1. idVehiculo char(6) NOT NULL,
      2. fecha date,
      3. PRIMARY KEY (idPs),
      4. FOREIGN KEY (idServicio) REFERENCES Servicio,
      5. FOREIGN KEY (idEmpleado) REFERENCES Empleado,
      6. FOREIGN KEY (idVehiculo) REFERENCES Vehiculo );

1. Inserte los datos del empleado Mario Molina que ingresó a la organización el 10/02/2010
   * 1. INSERT INTO Empleado( idEmpleado, nombreEmpleado, fechaIngreso)
     2. VALUES (1, “Mario Molina”, 10/02/2010);
     3. o bien,
     4. INSERT INTO Empleado
     5. VALUES (1, “Mario Molina”, 10/02/2010);
2. Mostrar la cantidad de prestaciones de servicio ejecutadas entre el 01 de octubre del 2018 y el 23 de noviembre del 2018.

SELECT COUNT(\*)

FROM prestacionServicio

WHERE fecha BETWEEN 01/10/2018 AND 23/11/2018;

1. Mostrar la cantidad total de prestaciones realizadas agrupadas por idVehiculo

SELECT idVehiculo, COUNT(idServicio)

FROM prestacionServicio

GROUP BY idVehiculo;

1. Mostrar los vehículos con la menor cantidad de prestaciones de servicio realizadas.

SELECT idVehiculo, COUNT(idServicio)

FROM prestacionServicio

GROUP BY idVehiculo

ORDER BY COUNT(idServicio) ASC;

1. Mostrar el nombre del empleado, nombre del Servicio y modelo del vehículo en la prestación de servicio N°7 (idPs)

SELECT Empleado.nombreEmpleado, Servicio.nombreServicio, Vehiculo.modeloVehiculo

FROM Empleado, Servicio, Vehiculo, prestacionServicio

WHERE Empleado.idEmpleado = prestacionServicio.idEmpleado

AND Servicio.idServicio = prestacionServicio.idServicio

AND Vehiculo.idVehiculo = prestacionServicio.idVehiculo;

AND prestacionServicio. idPs =7;

1. Como quedaría el SQL de la tabla prestacionServicio, si es que: la restricción de las claves foráneas no se realiza con el método RESTRICT(recuerde que por defecto se realiza este tipo de restricción) , sino que cuando se haga un update se realizan con método CASCADE y cuando se haga un DELETE se dejan en NULL, salvo para el idServicio, que se quiere que en caso de que se borre un servicio, las referencias a este queden en 9999.
   1. CREATE TABLE prestacionServicio (

id\_ps int,

idServicio int DEFAULT 9999,

idEmpleado int,

idVehiculo int,

fecha date,

* + - 1. PRIMARY KEY (id\_ps)
      2. FOREIGN KEY (idServicio) REFERENCES Servicio

ON UPDATE CASCADE

ON DELETE SET DEFAULT

* + - 1. FOREIGN KEY (idEmpleado) REFERENCES Empleado

ON UPDATE CASCADE

ON DELETE SET NULL

* + - 1. FOREIGN KEY (idVehiculo) REFERENCES vehiculo)

ON UPDATE CASCADE

ON DELETE SET NULL

1. Como ejercicio de estudio ingrese datos a todas las tablas, para que pueda comprobar empíricamente todas las consultas y restricciones puestas a las claves foráneas.