**Solución punto 1**

select id\_fabricante, id\_producto, descripcion, precio, precio+(precio\*0.10) as precio\_con\_iva from productos;

para realizar la prueba se creo la tabla y se insertó los datos

CREATE TABLE dbo.productos (

id\_fabricante VARCHAR(50),

id\_producto VARCHAR(30) NOT NULL PRIMARY KEY,

descripcion VARCHAR (200) NOT NULL,

precio INT NOT NULL,

existencias INT NOT NULL,

);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Aci','41001','Arandela',58,277);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Aci','41002','Bisagra',80,150);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Aci','41003','Perno',112,80);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Aci','41004','Extractor',110,50);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Bic','41005','Perno',120,20);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Inc','41089','Clavos',500,30);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Qsa','XK47','Red',150,200);

insert into dbo.productos (id\_fabricante,id\_producto,descripcion,precio,existencias)

values('Bic','XK48','Red',200,200);

**Solución punto 2**

CREATE TABLE dbo.departamentos (

Codigo INT NOT NULL,

Nombre NVARCHAR(100) NOT NULL,

Presupuesto INT NOT NULL,

PRIMARY KEY(Codigo)

);

CREATE TABLE dbo.empleados (

DNI VARCHAR(8) NOT NULL,

Nombre NVARCHAR(100) NOT NULL,

Apellidos NVARCHAR(255) NOT NULL,

Departamento INT NOT NULL FOREIGN KEY REFERENCES departamentos(Codigo),

PRIMARY KEY (DNI)

);

select apellidos from empleados;

select distinct apellidos from empleados;

select \* from empleados where like ‘lopez’ or apellidos like ‘perez’;

select sum presupuesto from departamentos;

select \* (select caunt (1) where departamentos = codigo ) from departamentos;

select nombres, apellidos from empleados e inner join departamentos d on e.departamentos d.codigo where d.presupuesto >60000;

update from empleados set departamento=14 where departamento = 77 ;

update from empleados set departamento =null where departamento =14;

también puede asi :

delecte from empleados where departamentos = 14;

**Solución punto 3**

public int proximaFechaHabil(Calendar fechaInicial, int Numdias) {

//llenamos la lista de festivos que la traemos de la base de datos

//iniciamos la coneccion con sql server

Connection conexion;

String usuario = "sa";

String contraseña = "123";

String url = "jdbc:sqlserver://localhost:1433;databaseName=prueba";

try{

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

conexion = DriverManager.getConnection(url, usuario, contraseña);

System.out.println("Conectado.");

Statement stmt = conexion.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM diasFestivos");

//llenamos la lista de festivos que la traemos de la base de datos

List<Date> listaFechasNoLaborables = new ArrayList<>();

while (rs.next()) {

int dia = rs.getInt("DIA");

int mes = rs.getInt("MES");

int anio =2020;

//la fecha esta dada por anio mes dia

Date myDate = new Date(anio, mes, dia);

listaFechasNoLaborables.add(myDate);

}

int numerodias = 0;

Calendar fechaFinal = Calendar.getInstance();

fechaFinal.set(fechaInicial.YEAR,fechaInicial.MONTH,fechaInicial.DATE);

boolean diaHabil = false;

//mientras la fecha inicial sea menor o igual que la fecha final se cuentan los dias

while (fechaInicial.before(fechaFinal) || fechaInicial.equals(fechaFinal)) {

if (!listaFechasNoLaborables.isEmpty()) {

for (Date date : listaFechasNoLaborables) {

Date fechaNoLaborablecalendar = fechaInicial.getTime();

//si el dia de la semana de la fecha minima es diferente de sabado o domingo

if (fechaInicial.get(Calendar.DAY\_OF\_WEEK) != Calendar.SUNDAY && fechaInicial.get(Calendar.DAY\_OF\_WEEK) != Calendar.SATURDAY && !fechaNoLaborablecalendar.equals(date)) {

//se aumentan los dias de diferencia entre min y max

diaHabil = true;

} else {

diaHabil = false;

break;

}

}

} else {

if (fechaInicial.get(Calendar.DAY\_OF\_WEEK) != Calendar.SUNDAY && fechaInicial.get(Calendar.DAY\_OF\_WEEK) != Calendar.SATURDAY) {

//se aumentan los dias de diferencia entre min y max

numerodias++;

}

}

if (diaHabil == true) {

numerodias++;

}

//se suma 1 dia para hacer la validacion del siguiente dia.

fechaInicial.add(Calendar.DATE, 1);

}

return numerodias;

}catch(SQLException | ClassNotFoundException ex){

ex.printStackTrace();

System.out.println("Error.");

return Numdias;

}

}