﻿#' \*\*\* SCRIPT PARA CREAR Y CARGAR LAS TABLAS DEL CURSO DE SQL

#' \*\*\* DEPARTAMENTOS, EMPLEADOS, CLIENTES, PRODUCTOS Y PEDIDOS

#' ######################## SELECCION DE LA BASE DE DATOS #######################

DROP DATABASE IF EXISTS CURSO;

CREATE DATABASE CURSO;

USE CURSO;

#' ######################## BORRADO DE TABLAS #######################

#' # POR SI YA EXISTEN Y SE QUIEREN VOLVER A CREAR ##################

#' # sI NO EXISTEN SE PRODUCIRAN UNOS ERRORES QUE NO DEBEN SER TENIDOS EN CUENTA ########

DROP TABLE IF EXISTS PEDIDOS;

DROP TABLE IF EXISTS PRODUCTOS;

DROP TABLE IF EXISTS CLIENTES;

DROP TABLE IF EXISTS EMPLEADOS;

DROP TABLE IF EXISTS DEPARTAMENTOS;

#' ######################## CREACION DE TABLAS #######################

#' ## TABLA DEPARTAMENTOS

CREATE TABLE DEPARTAMENTOS

( DEP\_NO INT(2),

DNOMBRE VARCHAR(14),

LOCALIDAD VARCHAR(10),

CONSTRAINT PK\_DEPARTAMENTOS\_DEP\_NO PRIMARY KEY (DEP\_NO))

ENGINE = INNODB;

#' ## TABLA EMPLEADOS

CREATE TABLE EMPLEADOS

( EMP\_NO INT(4),

APELLIDO VARCHAR(8),

OFICIO VARCHAR(10),

DIRECTOR INT(4),

FECHA\_ALTA DATE,

SALARIO FLOAT(6,2),

COMISION FLOAT(6,2),

DEP\_NO INT (2),

CONSTRAINT PK\_EMPLEADOS\_EMP\_NO PRIMARY KEY (EMP\_NO),

CONSTRAINT FK\_EMP\_DIRECTOR FOREIGN KEY (DIRECTOR)

REFERENCES EMPLEADOS(EMP\_NO),

CONSTRAINT FK\_EMP\_DEP\_NO FOREIGN KEY (DEP\_NO)

REFERENCES DEPARTAMENTOS(DEP\_NO)

)ENGINE = INNODB;

#' ## TABLA CLIENTES

CREATE TABLE CLIENTES

(CLIENTE\_NO INT(4),

NOMBRE VARCHAR(25),

LOCALIDAD VARCHAR(14),

VENDEDOR\_NO INT(4),

DEBE FLOAT(9,2),

HABER FLOAT(9,2),

LIMITE\_CREDITO FLOAT(9,2),

CONSTRAINT PK\_CLIENTE\_NO PRIMARY KEY(CLIENTE\_NO),

CONSTRAINT FK\_CLI\_EMP\_NO FOREIGN KEY (VENDEDOR\_NO)

REFERENCES EMPLEADOS(EMP\_NO) ON DELETE CASCADE

)ENGINE = INNODB;

#' ## TABLA PRODUCTOS

CREATE TABLE PRODUCTOS

(PRODUCTO\_NO INT(4),

DESCRIPCION VARCHAR(30),

PRECIO\_ACTUAL FLOAT(8,2),

STOCK\_DISPONIBLE INT(9),

CONSTRAINT PK\_PRODUCTO\_NO PRIMARY KEY (PRODUCTO\_NO)

)ENGINE = INNODB;

#' ## TABLA PEDIDOS

CREATE TABLE PEDIDOS

(PEDIDO\_NO INT(4),

PRODUCTO\_NO INT(4),

CLIENTE\_NO INT(4),

UNIDADES INT(4),

FECHA\_PEDIDO DATE,

CONSTRAINT PK\_PEDIDO\_NO PRIMARY KEY (PEDIDO\_NO),

CONSTRAINT FK\_PEDIDOS\_PRODUCTO\_NO FOREIGN KEY (PRODUCTO\_NO)

REFERENCES PRODUCTOS(PRODUCTO\_NO) ON DELETE CASCADE,

CONSTRAINT FK\_PEDIDOS\_CLIENTE\_NO FOREIGN KEY (CLIENTE\_NO)

REFERENCES CLIENTES(CLIENTE\_NO) ON DELETE CASCADE

)ENGINE = INNODB;

#' ##################### INTRODUCCIÓN DE DATOS ######################

#' ## DEPARTAMENTOS

INSERT INTO DEPARTAMENTOS VALUES(10, 'CONTABILIDAD', 'BARCELONA');

INSERT INTO DEPARTAMENTOS VALUES(20, 'INVESTIGACION', 'VALENCIA');

INSERT INTO DEPARTAMENTOS VALUES(30, 'VENTAS', 'MADRID');

INSERT INTO DEPARTAMENTOS VALUES(40, 'PRODUCCION', 'SEVILLA');

#' ## EMPLEADOS

INSERT INTO EMPLEADOS VALUES (7839,'REY', 'PRESIDENTE',NULL,'1981-11-17',6000, NULL, 10);

INSERT INTO EMPLEADOS VALUES (7698,'GARRIDO', 'DIRECTOR', 7839,'1981-05-01',3850.12,NULL, 30);

INSERT INTO EMPLEADOS VALUES (7782,'MARTINEZ','DIRECTOR', 7839,'1981-06-09',2450, NULL, 10);

INSERT INTO EMPLEADOS VALUES(7499,'ALONSO', 'VENDEDOR', 7698,'1981-02-23',1400, 400,30);

INSERT INTO EMPLEADOS VALUES (7521,'LOPEZ', 'EMPLEADO', 7782,'1981-05-08',1350.50,NULL,10);

INSERT INTO EMPLEADOS VALUES (7654,'MARTIN', 'VENDEDOR', 7698,'1981-09-28',1500, 1600, 30);

INSERT INTO EMPLEADOS VALUES (7844,'CALVO', 'VENDEDOR', 7698,'1981-09-08',1800, 0, 30);

INSERT INTO EMPLEADOS VALUES (7876,'GIL', 'ANALISTA', 7782,'1982-05-06',3350, NULL, 20);

INSERT INTO EMPLEADOS VALUES (7900,'JIMENEZ', 'EMPLEADO', 7782,'1983-03-24',1400, NULL, 20);

INSERT INTO EMPLEADOS VALUES (7901,'Cortez', 'VENDEDOR', 7782,'1983-03-24',1400, NULL, 20);

#' ## CLIENTES

INSERT INTO CLIENTES VALUES (101, 'DISTRIBUCIONES GOMEZ', 'MADRID', 7499, 0,0,5000);

INSERT INTO CLIENTES VALUES (102, 'LOGITRONICA S.L', 'BARCELONA', 7654,0,0,5000);

INSERT INTO CLIENTES VALUES (103, 'INDUSTRIAS LACTEAS S.A.', 'LAS ROZAS', 7844,0,0, 10000);

INSERT INTO CLIENTES VALUES (104, 'TALLERES ESTESO S.A.', 'SEVILLA', 7654, 0, 0, 5000);

INSERT INTO CLIENTES VALUES (105, 'EDICIONES SANZ', 'BARCELONA', 7499, 0,0,5000);

INSERT INTO CLIENTES VALUES (106, 'SIGNOLOGIC S.A.', 'MADRID', 7654,0,0,5000);

INSERT INTO CLIENTES VALUES (107, 'MARTIN Y ASOCIADOS S.L.', 'ARAVACA' , 7844,0,0, 10000);

INSERT INTO CLIENTES VALUES (108, 'MANUFACTURAS ALI S.A.', 'SEVILLA', 7654, 0, 0, 5000);

#' ## PRODUCTOS

INSERT INTO PRODUCTOS VALUES(10,'MESA DESPACHO MOD. GAVIOTA', 550, 50);

INSERT INTO PRODUCTOS VALUES (20, 'SILLA DIRECTOR MOD. BUFALO', 670, 25);

INSERT INTO PRODUCTOS VALUES (30, 'ARMARIO NOGAL DOS PUERTAS', 460, 20);

INSERT INTO PRODUCTOS VALUES (40, 'MESA MODELO UNIÓN',340, 15);

INSERT INTO PRODUCTOS VALUES (50, 'ARCHIVADOR CEREZO',1050, 20);

INSERT INTO PRODUCTOS VALUES (60, 'CAJA SEGURIDAD MOD B222', 280, 15);

INSERT INTO PRODUCTOS VALUES (70, 'DESTRUCTORA DE PAPEL A3',450, 25);

INSERT INTO PRODUCTOS VALUES (80, 'MODULO ORDENADOR MOD. ERGOS', 550, 25);

update productos set stock\_disponible = 45 where producto\_no = 10;

#' ## PEDIDOS

INSERT INTO PEDIDOS VALUES(1000, 20, 103, 3, '1999-10-06');

INSERT INTO PEDIDOS VALUES(1001, 50, 106, 2, '1999-10-06');

INSERT INTO PEDIDOS VALUES(1002, 10, 101, 4, '1999-10-07');

INSERT INTO PEDIDOS VALUES(1003, 20, 105, 4, '1999-10-16');

INSERT INTO PEDIDOS VALUES(1004, 40, 106, 8, '1999-10-20');

INSERT INTO PEDIDOS VALUES(1005, 30, 105, 2, '1999-10-20');

INSERT INTO PEDIDOS VALUES(1006, 70, 103, 3, '1999-11-03');

INSERT INTO PEDIDOS VALUES(1007, 50, 101, 2, '1999-11-06');

INSERT INTO PEDIDOS VALUES(1008, 10, 106, 6, '1999-11-16');

INSERT INTO PEDIDOS VALUES(1009, 20, 105, 2, '1999-11-26');

INSERT INTO PEDIDOS VALUES(1010, 40, 102, 3, '1999-12-08');

INSERT INTO PEDIDOS VALUES(1011, 30, 106, 2, '1999-12-15');

INSERT INTO PEDIDOS VALUES(1012, 10, 105, 3, '1999-12-06');

INSERT INTO PEDIDOS VALUES(1013, 30, 106, 2, '1999-12-06');

INSERT INTO PEDIDOS VALUES(1014, 20, 101, 4, '2000-01-07');

INSERT INTO PEDIDOS VALUES(1015, 70, 105, 4, '2000-01-16');

INSERT INTO PEDIDOS VALUES(1017, 20, 105, 6, '2000-01-20');

-- consultas multitabla

-- a) Producto caretsiano

/\* aplicando el producto cartesiano a dos tablas se obtiene una tabla con las columnas

de la primera tabla unidas a las columnas de la segunda tabla, y las filas de la tabla

resultante son todas las posibles concatenaciones de filas de la primera tabla con filas de la segunda tabla

sintaxis

Select \* from tabla1, tabla2;

ejemplo: \*/

select \* from departamentos,empleados;

-- Producto cartesiano con condición. Ejemplo seleccionar los empleados que estan en el departamento de contabilidad

select \* from departamentos,empleados where (Select dep\_no from departamentos where dnombre like "Contabilidad") = empleados.dep\_no; -- esto no es correct

select \* from departamentos,empleados where (Select dep\_no from departamentos where dnombre like "Contabilidad") = empleados.dep\_no and departamentos.dep\_no = empleados.dep\_no; -- esto es correcto

-- describe por que la segunda es la respuesta correcta....

/\* b join La instrucción SQL JOIN se utiliza para combinar dos o más tablas, tomando un campo común de las dos.

sintaxis:

select \* from tabla1 join tabla2 on tabla1.campo\_comun = tabla2.campo\_comun; en este caso el nombre de los campos en común no se llaman igual

select \* from tabla1 join tabla2 using(campo\_comun); en este caso se usa cuando el campo en comun se llama igual \*/

-- ejemplo la tabla departamentos y empleados tienen en común el campo dep\_no

select \* from departamentos join empleados;

-- o

select \* from departamentos join empleados using (dep\_no);

-- en la tabla clientes y la tabla empleados tiene en comun el campo vendedor\_no y emp\_no, donde vendedor\_no en la tabla clientes es

-- el número de empleado que atendio a cliente. En este caso los nombres de los campos en común tiene nombre diferente

select \* from clientes join empleados on clientes.vendedor\_no = empleados.emp\_no; -- solo es valida está sintaxis para este tipo de casos