



**Universidad
Politécnica
de Chiapas**

Bases de datos

Práctica 2_1

Integrantes

- Luis Fernando Hernández Morales
 - David Pérez Sánchez
- Carlos André Sánchez Malpica
- Jesús Israel Santiago Gutiérrez
- Víctor Fernando Gil Calderón

Tabla de contenido

I.	MySQL.....	2
II.	POSTGRESQL.....	4
III.	SQL Server	6

Creación de tablas

```
CREATE SCHEMA `jardineria`;  
  
CREATE TABLE `Productos` (  
  
  `idProducto` INT NOT NULL AUTO_INCREMENT,  
  
  `tipo` VARCHAR(45) NOT NULL,  
  
  `nombre` VARCHAR(45) NOT NULL,  
  
  `fechaIngreso` VARCHAR(45) NOT NULL,  
  
  `condicionActual` VARCHAR(45) NOT NULL,  
  
  PRIMARY KEY (`idProducto`))  
  
ENGINE = InnoDB;
```

```
CREATE TABLE `Calendario` (  
  
  `idCalendario` INT NOT NULL AUTO_INCREMENT,  
  
  `idProducto` INT NOT NULL,  
  
  `diaRiego` VARCHAR(45) NOT NULL,  
  
  PRIMARY KEY (`idCalendario`),  
  
  INDEX `fk_cal_producto_idx` (`idProducto` ASC),  
  
  CONSTRAINT `fk_cal_producto`  
  
    FOREIGN KEY (`idProducto`)  
  
    REFERENCES `Productos` (`idProducto`)  
  
    ON DELETE NO ACTION  
  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
CREATE TABLE `Historial` (  
  
  `id` INT NOT NULL AUTO_INCREMENT,  
  
  `foto` VARCHAR(45) NOT NULL,  
  
  `idProducto` INT NOT NULL,  
  
  `fecha` VARCHAR(45) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  INDEX `fk_histo_producto_idx` (`idProducto` ASC),  
  
  CONSTRAINT `fk_histo_producto`  
  
    FOREIGN KEY (`idProducto`)  
  
    REFERENCES `Productos` (`idProducto`)  
  
    ON DELETE NO ACTION  
  
    ON UPDATE NO ACTION)  
  
ENGINE = InnoDB;
```

Características de MySQL

En MySQL puedes especificar qué motor de bases de datos usar, en MySQL Server esto no es posible. Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice. Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma.

POSTGRESQL

```
CREATE DATABASE jardineria
```

```
WITH OWNER = postgres
```

```
ENCODING = 'UTF8'
```

```
TABLESPACE = pg_default
```

```
LC_COLLATE = 'es_MX.UTF-8'
```

```
LC_CTYPE = 'es_MX.UTF-8'
```

```
CONNECTION LIMIT = -1;
```

```
CREATE TABLE "public"."producto"(  
    "idproducto" integer NOT NULL,  
    "tipo" character varying(45) NOT NULL,  
    "nombre" character varying(45) NOT NULL,  
    "fechaingreso" character varying(45) NOT NULL,  
    CONSTRAINT producto_pkey PRIMARY KEY ("idproducto")  
);
```

```
CREATE TABLE "public"."historial"(  
    "id" integer NOT NULL,  
    "foto" character varying(45) NOT NULL,  
    "idproducto" integer NOT NULL,  
    "fecha" character varying(45) NOT NULL,  
    CONSTRAINT historial_pkey PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "public"."calendario"(

    "idcalendario" integer NOT NULL,

    "idproducto" integer NOT NULL,

    "diariego" character varying(45) NOT NULL,

    CONSTRAINT calendario_pkey PRIMARY KEY ("idcalendario")

);
```

Características

Creación de la base de datos

Tenemos que definir con `WITH OWNER` qué usuario de postgres va tener el acceso a la base de datos, la codificación que es estándar `ENCODING` igual que con `MYSQL`, `TABLESPACE` es en donde se almacenarán nuestros archivos (puede ser referenciado por su nombre a la hora de crear objetos de base de datos, la hora con la que estará configurada la zona horaria `LC_COLLATE`, `LC_CTYPE` nos permite al momento de que nosotros hacemos una consulta, el orden en que este nos regresa va ser de acuerdo a lo que es en México) y por último `CONNECTION LIMIT` son las conexiones que permitiremos a la base de datos se representa con un `-1` cuando sea ilimitado la cantidad de conexiones (podemos usar `MAX_CONNECTIONS` como un delimitador de conexiones).

Creación de la tabla

Tenemos que especificar como pública dentro del esquema de la base de datos “jardineria” (en nuestro caso) y con comillas dobles para que el esquema pueda reconocerlo `CREATE TABLE "public"."<nombre de la tabla>"` cada atributo que contenga nuestra tabla debe estar entre comillas dobles posteriormente el tipo cabe destacar que el nombre con que se pone el tipo cambia bastante a comparación de `MySQL` en `MySQL` ponemos `int(2)` en cambio en `Postgresql` ponemos `integer(2)`, así como un `character varying(45)` de postgres en `MySQL` solo hay que referenciar con un `varchar(45)`. Una parte un poco más compleja sería poner llaves primarias dado que hay que especificar `CONSTRAINT <nombre de la tabla>_pkey PRIMARY KEY` (“<el atributo que sea de la base de datos>”) que en nuestro caso sería la sentencia siguiente de la tabla calendario:

```
CONSTRAINT calendario_pkey PRIMARY KEY ("idcalendario")
```

Y sea nombre de la base de datos o atributos de esta siempre se colocara con minúsculas.

