

**UNIVERSIDAD PRIVADA
“FRANZ TAMAYO”**



FACULTAD: INGENIERIA

CARRERA: INGENIERIA DE SISTEMAS

ASIGNATURA: BASE DE DATOS II

ESTUDIANTE:

➤ Ronald Uriel Choque Paco – SIS6972733

DOCENTE: Ing. William Roddy Barra Paredes

GESTION: 2022

EL ALTO – BOLIVIA

INTRODUCCIÓN AL MANEJO DE BASE DE DATOS COMANDOS BÁSICOS I CON DATAGRIP

1. Revisando Consola de DataGrip
 - a) Insertar 2 registros y mostrar esos valores ingresados.
 - b) Puede crear cualquier tabla.
 - c) Todo esto debe ser trabajador desde Datagrip.

```
CREATE DATABASE ejemplo;
USE ejemplo;

CREATE TABLE table1
(
    id_table INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
    nombre_table VARCHAR(100) NOT NULL
);
```

2. Eliminar la base de datos universidad y sus tablas.
3. Creación de una base de datos. (Crear la base de datos universidad)
4. Crear la tabla estudiante, con las siguientes características.
5. Agregar nuevos registros a la tabla creada estudiantes en la base de datos universidad.
6. Seleccionar todos los registros de la tabla estudiantes.
7. Mostrar el ID del último valor agregado.
8. Agregar un nuevo campo a la tabla estudiantes.
9. Agregar 2 nuevos campos con una sola instrucción.
10. Eliminar el campo DIRECCION de la tabla estudiantes.

```
#respuesta 3
CREATE DATABASE universidad;
USE universidad;
#respuesta 4
CREATE TABLE estudiantes
(
    id_est      INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
    nombres     VARCHAR(100),
    apellidos   VARCHAR(100),
    edad        INTEGER,
    fono        INTEGER,
    email       VARCHAR(50)
);
#respuesta 5
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email) VALUES
('nombre1', 'apellidos1', 12, 123456, 'nombre1@gmail.com'),
('nombre2', 'apellidos2', 15, 123456, 'nombre2@gmail.com'),
('nombre3', 'apellidos3', 19, 123456, 'nombre3@gmail.com');
#respuesta 6
SELECT est.*
FROM estudiantes AS est;
#respuesta 7 para seleccionar el ultimo registro
```

```

select last_insert_id();
#respuesta 8
ALTER TABLE estudiantes ADD COLUMN direccion VARCHAR(50);
#respuesta 9 para que al agregar un campo y que no salga nulo
ALTER TABLE estudiantes ADD COLUMN ciudad VARCHAR(50) DEFAULT 'El Alto';
#respuesta 10
ALTER TABLE estudiantes DROP COLUMN direccion;

```

EJERCICIOS CONSULTAS I.

11. Mostrar todos los registros de la tabla estudiantes.
12. Mostrar los registros de aquellos estudiantes que su nombre sea igual a 'nombre3'.
13. Mostrar los registros de los estudiantes donde la edad sea mayor a 18 años.
14. Mostrar los registros donde cuyo ID sea PAR. (o IMPAR).

```

#respuesta 11
SELECT est.*
FROM estudiantes AS est;
#respuesta 12
SELECT est.*
FROM estudiantes AS est
WHERE nombres = 'nombre3';
#respuesta 13
SELECT est.*
FROM estudiantes AS est
WHERE edad >= 19;
#respuesta 14
SELECT est.*,
IF(id_est % 2 = 1, 'impar', 'par') AS ID_PAR_O_IMPAR
FROM estudiantes AS est;

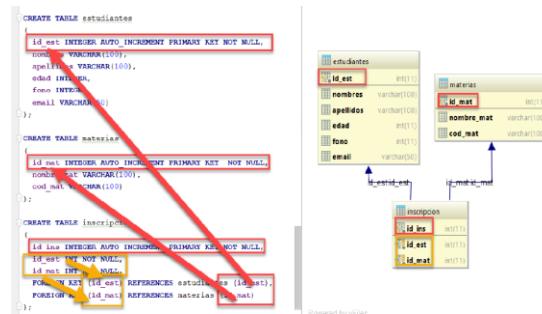
```

- Captura de la solución de la pregunta 14:

	id_est	nombres	apellidos	edad	fono	email	ciudad	ID_PAR_O_IMPAR
1	1	nombre1	apellidos1	12	123456	nombre1@gmail.com	El Alto	impar
2	2	nombre2	apellidos2	15	123456	nombre2@gmail.com	El Alto	par
3	3	nombre3	apellidos3	19	123456	nombre3@gmail.com	El Alto	impar

MANEJO DE PRIMARY KEY Y FOREIGN KEY.

15. Generar las tablas de acuerdo a la siguiente imagen.



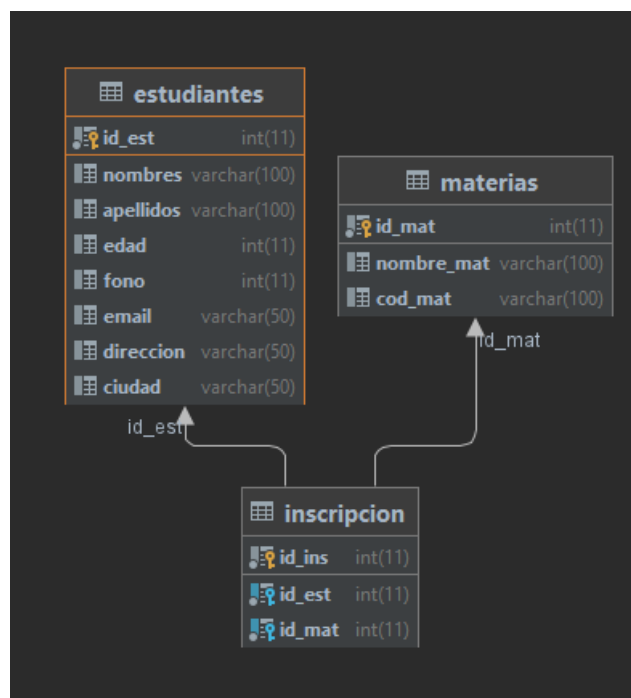
#respuesta 15

CREATE TABLE materias

```
(
  id_mat INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
  nombre_mat VARCHAR(100),
  cod_mat VARCHAR(100)
);
```

CREATE TABLE inscripcion

```
(
  id_ins INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
  id_est INTEGER NOT NULL,
  id_mat INTEGER NOT NULL,
  FOREIGN KEY (id_est) REFERENCES estudiantes (id_est),
  FOREIGN KEY (id_mat) REFERENCES materias (id_mat)
);
```



16. Generar las tablas de acuerdo a la siguiente imagen.

a) Crear una base de datos denominada Librería.

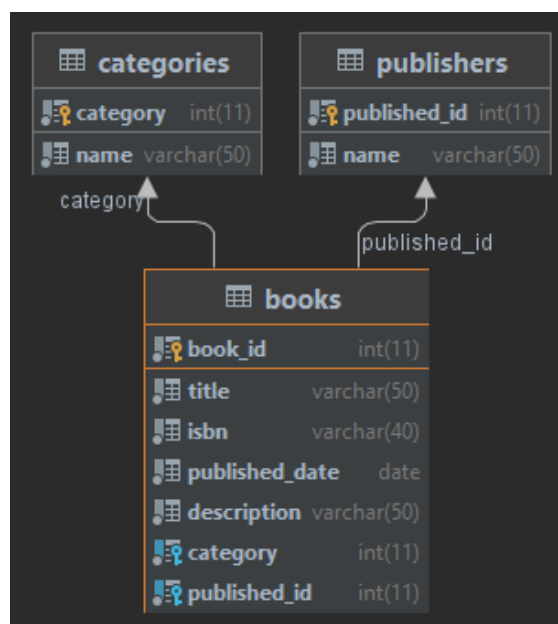
b) Hacer uso de esa base de datos.

```
#respuesta 16
CREATE DATABASE libreria;
USE libreria;

CREATE TABLE categories
(
    category INTEGER PRIMARY KEY NOT NULL,
    name VARCHAR(50) NOT NULL
);

CREATE TABLE publishers
(
    published_id INTEGER PRIMARY KEY NOT NULL,
    name VARCHAR(50) NOT NULL
);

CREATE TABLE books
(
    book_id INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
    title VARCHAR(50) NOT NULL,
    isbn VARCHAR(40) NOT NULL,
    published_date DATE NOT NULL,
    description VARCHAR(50) NOT NULL,
    category INTEGER NOT NULL,
    published_id INTEGER NOT NULL,
    FOREIGN KEY (category) REFERENCES categories(category),
    FOREIGN KEY (published_id) REFERENCES publishers(published_id)
);
```



PREGUNTA DE DISEÑO DE BASE DATOS.

Dado un escenario de empleados que trabajan en departamentos dentro de una empresa, como debería ser su

sistema de base de datos relacional.

- El objetivo es saber en qué empresa y en qué área trabaja una persona.
- Debe de crear una base de datos relacional denominada EMPRESA.
- Debería de crear 3 tablas mínimamente. (Empleado - Empresa - Area).

```
CREATE DATABASE empresa;
USE empresa;

CREATE TABLE compania
(
    id_co INTEGER PRIMARY KEY NOT NULL,
    nombre_co VARCHAR(50) NOT NULL
);
INSERT INTO compania (id_co, nombre_co) VALUES
(111, 'coca cola'),
(222, 'pepsi'),
(333, 'paceña');

CREATE TABLE areas
(
    id_area INTEGER PRIMARY KEY NOT NULL,
    nombre_area VARCHAR(50) NOT NULL
);
INSERT INTO areas (id_area, nombre_area) VALUES
(212, 'gerencia'),
(313, 'planta'),
(414, 'publicidad'),
(515, 'recursos y distribucion');

CREATE TABLE empleados
(
    id_empleado INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    telefono INTEGER NOT NULL,
    id_co INTEGER NOT NULL,
    id_area INTEGER NOT NULL,
    FOREIGN KEY (id_co) REFERENCES compania(id_co),
    FOREIGN KEY (id_area) REFERENCES areas(id_area)
);
INSERT INTO empleados (nombre, apellido, telefono, id_co, id_area) VALUES
('nombre1', 'apellido1', 123456, 111, 313),
('nombre2', 'apellido2', 123457, 222, 414),
('nombre3', 'apellido3', 123458, 111, 313),
('nombre4', 'apellido4', 123459, 333, 212),
('nombre5', 'apellido5', 123450, 222, 313),
('nombre6', 'apellido6', 123451, 333, 515),
```

```
('nombre7', 'apellido7', 123452, 111, 212);
```

```
SELECT emp.nombre, emp.apellido, c.nombre_co, a.nombre_area  
FROM empleados AS emp  
INNER JOIN compania c on emp.id_co = c.id_co  
INNER JOIN areas a on emp.id_area = a.id_area;
```

Visualizacion del resultado:

	nombre	apellido	nombre_co	nombre_area
3	nombre3	apellido3	coca cola	planta
4	nombre4	apellido4	paceña	gerencia
5	nombre5	apellido5	pepsi	planta
6	nombre6	apellido6	paceña	recursos y distribucion
7	nombre7	apellido7	coca cola	gerencia