BASE DE DATOS

MELANIE INGRID VILLCA

LABORATORIO 3 H2

31 de agosto de 2022

```
CREATE DATABASE function_aggregation;

USE function_aggregation;

CREATE TABLE estudiantes
(
id_est INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,
nombres VARCHAR(50),
apellidos VARCHAR(50),
edad INTEGER,
gestion INTEGER,
fono INTEGER,
email VARCHAR(100),
direccion VARCHAR(100),
genero VARCHAR(10)
);

SELECT est.*
FROM estudiantes AS est;

INSERT INTO estudiantes (nombres, apellidos, edad, fono, email,
direccion, genero) VALUES
      ('Miguel' ,'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com',
'Av. 6 de Agosto', 'masculino'),
      ('Sandra' ,'Mavir Uria', 25, 2832116, 'sandra@gmail.com', 'Av.
6 de Agosto', 'femenino'),
      ('Joel' ,'Adubiri Mondar', 30, 2832117, 'joel@gmail.com', 'Av.
6 de Agosto', 'masculino'),
```

```
      ('Andrea' ,'Arias Ballesteros', 21, 2832118,
'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino'),
      ('Santos' ,'Montes Valenzuela', 24, 2832119,
'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');

#Crear la consulta SQL
#que modifique el campo gestion
# y que asigne a todos el valor 2022

UPDATE estudiantes
SET gestion = '2022'
WHERE id_est > 0;

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 INT NOT NULL,
id_mat INT NOT NULL,
semestre VARCHAR(20),
gestion INTEGER,
FOREIGN KEY (id_est) REFERENCES estudiantes (id_est),
FOREIGN KEY (id_mat) REFERENCES materias (id_mat)
);

INSERT INTO materias (nombre_mat, cod_mat) VALUES
   ('Introduccion a la Arquitectura','ARQ-101'),
   ('Urbanismo y Diseno','ARQ-102'),
   ('Dibujo y Pintura Arquitectonico','ARQ-103'),
   ('Matematica discreta','ARQ-104'),
   ('Fisica Basica','ARQ-105');
```

```
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES
    (1, 1, '1er Semestre', 2015),
    (1, 2, '2do Semestre', 2015),
    (2, 4, '1er Semestre', 2016),
    (2, 3, '2do Semestre', 2016);
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion) VALUES
    (3, 3, '2do Semestre', 2017),
    (3, 1, '3er Semestre', 2017),
    (4, 4, '4to Semestre', 2017),
    (5, 5, '5to Semestre', 2017);

SELECT est.*
FROM materias AS est;

SELECT est.*
FROM inscripcion AS est;

# Mostrar los nombres y apellidos de los estudiantes inscritos en la
materia
# ARQ-104, adicionalmente mostrar el nombre de la materia.

SELECT est.nombres, est.apellidos, mat.nombre_mat
FROM inscripcion AS ins
    INNER JOIN estudiantes AS est ON ins.id_est=est.id_est
    INNER JOIN materias AS mat ON ins.id_mat=mat.id_mat
WHERE mat.cod_mat = 'ARQ-104';

#Contar cuantos registros tiene la tabla estudiantes
SELECT COUNT(est.id_est) AS 'Cantidad de estudiantes'
FROM estudiantes AS est;

#Mostrar el promedio de edad en la tabla estudiantes.
SELECT AVG(est.edad)
FROM estudiantes AS est;

#Mostrar la máxima edad que se tiene en la tabla estudiantes.
SELECT MAX(est.edad)
FROM estudiantes AS est;
```

```
#Mostrar la mínima edad que se tiene en la estudiantes.
SELECT MIN(est.edad)
FROM estudiantes AS est;

#Determinar la maxima edad  de los estudiantes
#cuyo genero sea 'masculino'/'femenino'
#y ademas la edad sea mayor de 20

#Manejo de funciones
#Crear una función que devuelve el máximo valor del campo edad en la
tabla estudiantes.

CREATE FUNCTION get_max_edad()
   RETURNS INTEGER
   BEGIN
       RETURN 10;
   END;

SELECT get_max_edad();

CREATE OR REPLACE FUNCTION get_max_edad()
   RETURNS INTEGER
   BEGIN
       RETURN (
           SELECT MAX(est.edad)
           FROM estudiantes AS est
    );
   END;

SELECT get_max_edad() AS MaxEdad;

#Crear una función que obtenga la menor edad de los estudiantes.
CREATE FUNCTION min_edad()
   RETURNS INTEGER
   BEGIN
       RETURN (
           SELECT MIN(est.edad)
```

```
            FROM estudiantes AS est
    );
  END;
SELECT min_edad() AS MinEdad;

#Crear una función que obtenga el promedio de las edades.
CREATE FUNCTION prom_edad()
  RETURNS INTEGER
  BEGIN
      RETURN (
            SELECT AVG(est.edad)
            FROM estudiantes AS est
    );
  END;
SELECT prom_edad() AS AvgEdad;

#Crear una función que obtenga la mayor edad de los estudiantes (cuyo
sexo seas masculino o femenino).

CREATE FUNCTION get_max_edad_varones()
  RETURNS INTEGER
  BEGIN
      RETURN (
            SELECT MAX(est.edad)
            FROM estudiantes AS est
            WHERE est.genero = 'masculino'
    );
  END;

#Mostrar el registro de la tabla estudiantes (nombre y apellidos)
donde cuyo id_est sea el máximo.

CREATE FUNCTION max_id_tabla_est() RETURNS integer
BEGIN
  RETURN(
      SELECT MAX(est.id_est)
      FROM estudiantes AS est
      );
```

```
end;

SELECT max_id_tabla_est() as ID_MAX;

SELECT est.nombres, est.apellidos
FROM estudiantes AS est
WHERE est.id_est = max_id_tabla_est();

#Mostrar todos los registros de la tabla estudiantes (nombres y
apellidos)
#si la suma de las edades de los estudiantes masculino/femenino sea
par.
#Crear una función que obtenga la suma de las edades de los
estudiantes masculino/femenino

CREATE FUNCTION suma_edades() RETURNS INTEGER
BEGIN
   RETURN (
        SELECT SUM(est.edad)
        FROM estudiantes AS est
        WHERE est.genero = 'masculino'
        );
end;

SELECT suma_edades();

SELECT est.nombres, est.apellidos
FROM estudiantes AS est
WHERE suma_edades() % 2 = 0;

#Mostrar todos los registros de la tabla estudiantes (nombres y
apellidos)
# si la suma de las edades de las estudiantes femeninos sea par.
#Crear una función que obtenga la suma de las edades de las
estudiantes femeninos.

CREATE FUNCTION suma_edades_femenino() RETURNS INTEGER
BEGIN
```

```
    RETURN (
        SELECT SUM(est.edad)
        FROM estudiantes AS est
        WHERE est.genero = 'femenino'
        );
end;

SELECT suma_edades_femenino();

SELECT est.nombres, est.apellidos
FROM estudiantes AS est
WHERE suma_edades_femenino() % 2 = 0;

#Volver parametrizable la anterior función.

CREATE FUNCTION suma_edades_v1(genero_valor VARCHAR(10)) RETURNS
INTEGER
    BEGIN
        RETURN (
            SELECT SUM(est.edad)
        FROM estudiantes AS est
        WHERE est.genero = genero_valor
            );
    END;

SELECT suma_edades_v1('masculino');

SELECT est.nombres, est.apellidos
FROM estudiantes AS est
WHERE suma_edades_v1('masculino') % 2 = 0;

#Volver parametrizable la anterior función. V2

CREATE FUNCTION suma_edades_v2(genero VARCHAR(50)) RETURNS INTEGER
BEGIN
#Esto sirve para declarar una variable en una funcion
    DECLARE sumaEdad INTEGER DEFAULT 0;
```

```
  SELECT SUM(est.edad) INTO sumaEdad
  FROM estudiantes AS est
  WHERE est.genero = genero;

  RETURN sumaEdad;
END;

SELECT est.nombres, est.apellidos
FROM estudiantes AS est
WHERE suma_edades_v2('masculino') %2 = 0;


CREATE OR REPLACE FUNCTION get_promedio_v2(genero VARCHAR(10))
RETURNS INTEGER
   BEGIN
       DECLARE promedio REAL DEFAULT 0;

       SELECT AVG(est.edad) INTO promedio
       FROM estudiantes AS est
       WHERE est.genero = genero;

       RETURN promedio;
   END;
SELECT get_promedio_v2('femenino');

SELECT est.nombres, est.apellidos
FROM estudiantes AS est
WHERE get_promedio_v2('masculino') % 2 = 0;


#Crear una función que permita concatenar dos columnas.
#Concatena el nombre y apellidos de la tabla estudiante.
#El nombre de la función deberá ser getNombreCompleto
#La función deberá recibir 2 parámetros (nombre y apellidos)

create function getNombreCompleto(par1 varchar(25),par2 varchar(25))
returns varchar(50)
begin
```

```
  declare concatenado varchar(50) DEFAULT '';
  set concatenado = CONCAT(par1, ' - ', par2);
  return concatenado;
end;

select getNombreCompleto('Pepito', 'Pep');

SELECT getNombreCompleto(est.nombres, est.apellidos) AS Fullname
FROM estudiantes AS est;

# Generar el siguiente formato de salida
# Concatenar nombres y apellidos de la siguiente forma: Nombres:
William, Apellidos: Barra
# Concatenar gestion y edad de la siguiente forma: Gestion : 2022 -
Edad(10)


create function getNombreCompleto1(par1 varchar(25),par2 varchar(25))
returns varchar(50)
begin
  declare concatenado varchar(50) DEFAULT '';
  set concatenado = CONCAT('Nombres: ',par1, ' , ', 'Apellidos:
',par2);
  return concatenado;
end;

create function getNombreCompleto2(par3 varchar(25),par4 varchar(25))
returns varchar(50)
begin
  declare concatenado varchar(50) DEFAULT '';
  set concatenado = CONCAT('Gestion: ',par3, ' -
','Edad','(',par4,')');
  return concatenado;
end;
```

```
SELECT getNombreCompleto1(est.nombres, est.apellidos) AS
NOMBRE_COMPLETO, getNombreCompleto2(est.gestion, est.edad) AS
GESTION_EDAD
FROM estudiantes AS est;

#

create function get1(par1 varchar(25),par2 varchar(25),par3
varchar(25),par4 varchar(25))
returns varchar(100)
begin
  declare concatenado varchar(100) DEFAULT '';
  set concatenado = CONCAT('Nombres: ',par1, ' , ', 'Apellidos:
',par2,' ','Gestion: ',par3, ' - ','Edad','(',par4,')');
  return concatenado;
end;


SELECT get1(est.nombres, est.apellidos,est.gestion, est.edad) AS
NOMBRECOMPLETO_GESTION_EDAD
FROM estudiantes AS est;

#SELECT CONCAT_WS

# Mostrar el nombre, apellidos y el semestre de todos los estudiantes
que estén inscritos.
# Inscritos en la gestión 2015.

SELECT est.nombres, est.apellidos, ins.semestre
FROM estudiantes AS est
INNER JOIN inscripcion AS ins ON ins.id_est = est.id_est
WHERE ins.gestion = 2015;

# Mostrar el nombre, apellidos y el semestre de todos los estudiantes
que estén inscritos.
# Inscritos en la gestión 2015.
# En la materia ARQ-101.
```

```
SELECT est.nombres, est.apellidos, ins.semestre
FROM estudiantes as est
    INNER JOIN inscripcion as ins ON ins.id_est = est.id_est
    INNER JOIN materias as mat ON ins.id_mat = mat.id_mat
WHERE ins.gestion = 2015 AND mat.cod_mat = 'ARQ-101';

# Mostrar el nombre, apellidos y el semestre de todos los estudiantes
que estén inscritos.
# Inscritos en la gestión 2015.
# En la materia ARQ-101.
# Inscritos en el 5to semestre.
SELECT est.nombres, est.apellidos, ins.semestre
FROM estudiantes as est
    INNER JOIN inscripcion as ins ON ins.id_est = est.id_est
    INNER JOIN materias as mat ON ins.id_mat = mat.id_mat
WHERE ins.gestion = 2015 AND mat.cod_mat = 'ARQ-101' AND ins.semestre
= '5to Semestre';
```