

EVALUACION HITO 4



BASE DE DATOS II

ALUMNO: Adán Abinadi Tudela Cabero
Evaluacion - Ing. De Sistemas

DOCENTE: Lic. William R. Barra Paredes.

Cochabamba, 9 de diciembre de 2019

1. Resolver los siguientes ejercicios.

- Generar la serie fibonacci.
 - El objetivo es sumar todos los números de la serie fibonacci desde una cadena.
 - Es decir usted tendrá solo la cadena generado con los primeros N números de la serie fibonacci y a partir de ellos deberá sumar los números de esa serie.
 - Puede utilizar el siguiente código para generar la serie fibonacci.

```
• CREATE FUNCTION sumaFibonacci(limite INTEGER) RETURNS TEXT
BEGIN
    DECLARE respuesta TEXT DEFAULT '0';
    DECLARE x INTEGER DEFAULT 0;
    DECLARE y INTEGER DEFAULT 1;
    DECLARE z INTEGER DEFAULT 0;
    DECLARE i INTEGER DEFAULT 0;
    DECLARE sum BIGINT DEFAULT 0;
    REPEAT
        SET z = x + y;
        SET x = y;
        SET y = z;
        SET sum = sum + x;
        SET respuesta = CONCAT(respuesta, ', ', x);
        SET i = i + 1;
    UNTIL i > (limite-2) END REPEAT;
    RETURN CONCAT(respuesta, ' - SUMATOTAL: ', sum);
END;

SELECT sumaFibonacci(10);
```

```
`sumaFibonacci(10)`
```

```
1 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 - SUMATOTAL: 88
```

- Crear una Vista de nombre **detalle_proyecto_persona** de las personas que no hayan nacido en la ciudad de **COCHABAMBA**.
 - La Vista debe de tener los siguientes campos (**NOMBRES, APELLIDOS, PROYECTO, TIPO_PROYECTO, NIVEL_PROYECTO, SEXO y DEPARTAMENTO**).
 - Consideraciones de la columna **NIVEL_PROYECTO**
 - Si la edad es mayor que 0 y menor igual a 20 mostrar 'MI PRIMER PROYECTO'
 - Si la edad es mayor que 20 y menor igual a 25 mostrar 'MI SEGUNDO PROYECTO'
 - Si la edad es mayor que 25 y menor igual a 30 mostrar 'MI TERCER PROYECTO'
 - Considere la imagen siguiente para tener el contexto de qué datos debe de mostrar cada columna.

```
• CREATE VIEW detalle_proyecto_persona AS
SELECT P.nombre NOMBRES,
       P.apellidos APELLIDOS,
       PRY.nombreProy PROYECTO,
       PRY.tipoProy TIPO_PROYECTO,
       case
```

```

when P.edad <=20 then 'MI PRIMER PROYECTO'
when P.edad <=25 then 'MI SEGUNDO PROYECTO'
when P.edad <=30 then 'MI TERCER PROYECTO'
ELSE 'CONTENIDO NO ENCONTRADO'
end AS NIVEL_PROYECTO,
P.sexo SEXO,D.nombre DEPARTAMENTO
FROM PERSONA AS P
INNER JOIN DETALLE_PROYECTO DP on P.id_per = DP.id_per
INNER JOIN DEPARTAMENTO D on P.id_dep = D.id_dep
INNER JOIN PROYECTO PRY on DP.id_proy = PRY.id_proy
WHERE D.nombre !='Cochabamba' ;
DROP VIEW detalle_proyecto_persona;

```

	NOMBRES	APELLIDOS	PROYECTO	TIPO_PROYECTO	NIVEL_PROYECTO	SEXO	DEPARTAMENTO
1	nombre4	apellidos4	Alfabetizacion	EDUCACION	MI SEGUNDO PROYECTO	m	La Paz
2	nombre4	apellidos4	Creacion de Escuelas	EDUCACION	MI SEGUNDO PROYECTO	m	La Paz
3	nombre5	apellidos5	Apoyo al dibujo	ARQUITECTURA	MI TERCER PROYECTO	m	Santa Cruz
4	nombre6	apellidos6	Apoyo al dibujo	ARQUITECTURA	MI PRIMER PROYECTO	f	Santa Cruz

- Crear TRIGGERS Before o After para **INSERT, UPDATE y DELETE** aplicado a la tabla PROYECTO.
- Crear una tabla de auditoría de acuerdo al siguiente:

```

CREATE TABLE auditoria_proyecto
(
    operation    CHAR(1) NOT
NULL,
    stamp        TIMESTAMP NOT
NULL,
    userid       TEXT    NOT NULL,
    hostname     TEXT    NOT NULL,
    idProy       INTEGER NOT
NULL,
    descProyBefore TEXT    NOT
NULL,
    descProyAfter TEXT    NOT
NULL
);

```

Consideraciones:

Los campos:

descProyBefore : Son los valores antes de la acción (insert - update - delete).

descProyAfter: Son los valores después de la acción (insert - update - delete).

Nota. Debe crear triggers.

- Crear un procedimiento almacenado SP para insertar datos a la tabla de auditoría.
- Considere la siguiente tabla de auditoría para ver el comportamiento esperado.

	stamp	userid	hostname	idProy	descProyBefore	descProyAfter
1	I	2019-12-02 05:20:23	root@localhost	DESKTOP-0I87J52	6 Empty value - INSERT ACTION	Apoyo al dibujo ARQUITECTURA
2	U	2019-12-02 05:23:11	root@localhost	DESKTOP-0I87J52	6 Apoyo al dibujo ARQUITECTURA	Apoyo al dibujo II ARQUITECTURA
3	D	2019-12-02 05:25:11	root@localhost	DESKTOP-0I87J52	6 Apoyo al dibujo II ARQUITECTURA II	Empty value - Delete action

- Crear un TRIGGER BEFORE INSERT para la tabla PERSONA.

<pre>CREATE TABLE auditoria_persona (operation CHAR(1) NOT NULL, stamp TIMESTAMP NOT NULL, userid TEXT NOT NULL, hostname TEXT NOT NULL, fullname TEXT NOT NULL, edad INTEGER NOT NULL, sexo TEXT NOT NULL);</pre>	<p>Si es día lunes(2) y además la persona es de la ciudad de La Paz no se debe insertar el registro y se debe generar un error indicando lo siguiente.</p> <p>MESSAGE_TEXT = 'No se admite inserciones en días lunes usuarios de la ciudad de LP';</p> <p>Si es otro día insertar los registros en la tabla de auditoria (auditoria_persona).</p> <p>Para saber en qué día se encuentra debe de utilizar : select DAYOFWEEK(now())</p>
---	---

- Debe de crear una función para verificar que es día lunes y que el usuario es de la ciudad de La Paz. la función recibe dos parámetros.
 - Día de la semana(INTEGER).
 - El ID de la ciudad (INTEGER)

GER	4	4	nombre4	apellidos4	1996-10-30	24	nombre4@gmail.com	m	2
AN D	5	5	nombre5	apellidos5	1992-10-30	28	nombre5@gmail.com	m	3
	6	6	nombre6	apellidos6	1999-10-30	19	nombre6@gmail.com	f	3
) IN	7	<generated>	nombre4	apellidos4	1996-10-30	24	nombre4@gmail.com	m	2
Mon									

[45000][1644] No se admite inserciones en días lunes usuarios de la ciudad de LP

7	8	nombre4	apellidos4	1996-10-30	24	nombre4@gmail.com	m	5	4
---	---	---------	------------	------------	----	-------------------	---	---	---

ong.auditoria_proyecto [DBAII] × ong.auditoria_persona [DBAII] ×

1 row

operation stamp userid hostname fullname edad sexo

1 I 2019-12-02 06:17:49 root@localhost DESKTOP-OI87J52 nombre4 apellidos4 24 m

```
SELECT DAYNAME(CURRENT_DATE());
SELECT dayofweek(now());
CREATE TABLE auditoria_persona
(
  operation CHAR(1) NOT NULL,
  stamp    TIMESTAMP NOT NULL,
  userid   TEXT    NOT NULL,
  hostname TEXT    NOT NULL,
  fullname TEXT    NOT NULL,
  edad     INTEGER NOT NULL,
  sexo     TEXT    NOT NULL
);
create function verificaDiaCiudad(diaciudad text)returns boolean
```

```

begin
    declare dia integer default 0;
    declare ciuda integer default 0;
    declare res boolean default false;
    set dia = locate('Wednesday',diaciudad);
    SET ciuda = locate('2',diaciudad);#LaPaz es numero 2 en id
    if dia > 0 and ciuda >0
        then
            set res = true;
        end if;
    return res;
    #return if(diaciudad = 'Wednesday2',true,false);
end;
drop function verificaDiaCiudad;
CREATE TRIGGER triggerPersonaver
    BEFORE INSERT ON PERSONA
    FOR EACH ROW
    BEGIN
        DECLARE DIA boolean DEFAULT
        verificaDiaCiudad(concat(DAYNAME(CURRENT_DATE()),NEW.id_dep));
        if DIA = true
            then
                signal sqlstate '45000' set message_text = 'NO SE ADMITE INSERCIONES
LOS MIERCOLES PARA LA PAZ';
            ELSE
                INSERT INTO auditoria_persona
                    (operation, stamp, userid, hostname, fullname, edad, sexo) SELECT
                        'I', now(), user(),@@hostname,concat(new.nombre,'
',new.apellidos),NEW.edad,NEW.sexo;
            end if ;
    END;
insert into PERSONA(id_per, nombre, apellidos, fecha_nac, edad, email, id_dep,
id_prov, sexo) values
(123,'nombre6', 'apellidos6', '1999-10-30', 19, 'nombre6@gmail.com', 2, 5,'f');

```

[45000][1644] NO SE ADMITE INSERCIONES FINES DE SEMANA

- **Opcional**, resolver el ejercicio anterior haciendo que la función solo reciba un parámetro.

```
create function verificaDiaCiudad(diaciudad text)returns boolean
begin
    declare dia integer default 0;
    declare ciuda integer default 0;
    declare res boolean default false;
    set dia = locate('Wednesday',diaciudad);
    SET ciuda = locate('2',diaciudad);#LaPaz es numero 2 en id
    if dia > 0 and ciuda >0
        then
            set res = true;
        end if;
    return res;
    #return if(diaciudad = 'Wednesday2',true,false);
end;
drop function verificaDiaCiudad;
```