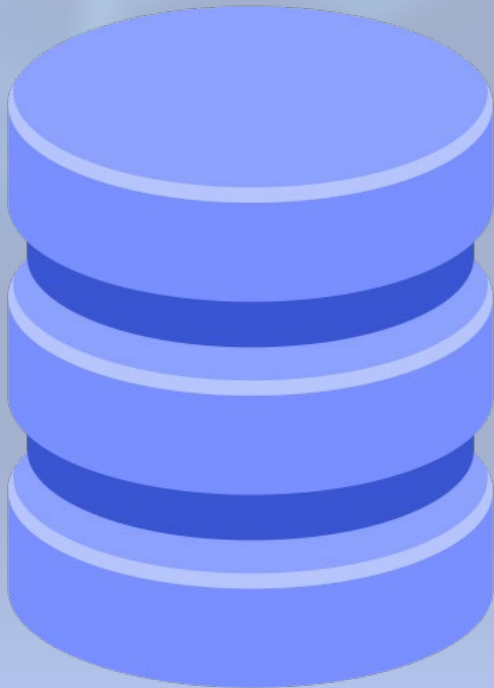


MYSQL: EVENTOS III



GESTIÓN DE BASES DE DATOS
JUAN CARLOS NAVIDAD GARCÍA

1. El próximo domingo, se ponen de oferta las habitaciones individuales, bajando su precio un 50%. A partir del lunes, vuelven a tener su precio original.

```
CREATE EVENT IF NOT EXISTS oferta
ON SCHEDULE EVERY 1 DAY
STARTS '2023-04-02 00:00:00'
ENDS '2023-04-03 00:00:00'
DO
CALL oferta();
```

```
CREATE EVENT IF NOT EXISTS no_oferta
ON SCHEDULE EVERY 1 DAY
STARTS '2023-04-03 00:00:00'
ENDS '2023-04-03 00:01:00'
DO
CALL no_oferta();
```

```
CREATE PROCEDURE oferta ()
BEGIN
DROP TABLE IF EXISTS habitaciones_bk;
CREATE TABLE habitaciones_bk LIKE habitaciones;
INSERT INTO habitaciones_bk SELECT * FROM habitaciones;
UPDATE habitaciones SET preciodia = preciodia * 0.5 WHERE categoría="Individual";
END//
```

```
CREATE PROCEDURE no_oferta ()
BEGIN
UPDATE habitaciones SET preciodia = (SELECT habitaciones_bk.preciodia FROM
habitaciones_bk,habitaciones WHERE habitaciones_bk.numhab=habitaciones.numhab);
END //
```

2. Crear un evento que añada hoy una consumición cada 15 minutos desde las 12 de la mañana a las 8 de la tarde del cliente que queráis (en realidad se añade a una ocupación).

```
CREATE EVENT IF NOT EXISTS consumicion
ON SCHEDULE EVERY 15 MINUTE
STARTS '2023-03-27 12:00:00'
ENDS '2023-03-27 20:00:00'
DO
INSERT INTO consumiciones (codocupacion,codprod,fecha,unidades)
VALUES(17,020,CURDATE(),1);
```

3. Crear una tabla con los campos "Fecha"(PK) y "Total", la llamáis CajaConsumiciones. Una vez hecho, crear un evento, para que al final del día,

se rellene dicha tabla con el total que se ha consumido cada día. Se pueden utilizar funciones y/o procedimientos como apoyo.

```
CREATE TABLE CajaConsumiciones AS
(SELECT fecha,sum(unidades*precio) AS Total
FROM consumiciones,productos
WHERE consumiciones.codprod=productos.codpro
GROUP BY fecha);
```

```
CREATE PROCEDURE cajadia ()
BEGIN
DECLARE caja DECIMAL(10,2);
SELECT sum(unidades*precio) AS Total INTO caja
FROM consumiciones,productos
WHERE consumiciones.codprod=productos.codpro AND fecha=CURDATE();
INSERT INTO cajaconsumiciones VALUES(curdate(),caja);
END //
```

```
CREATE EVENT IF NOT EXISTS consumicion
ON SCHEDULE EVERY 24 HOUR
STARTS '2023-03-27 23:59:00'
DO
CALL cajadia();
```

4. Crear un evento que recorra la tabla de ocupaciones, y si la fecha de hoy se encuentra en el intervalo entre fecha entrada y fecha salida, actualizamos a "ocupada" la habitación. El evento se ejecutará 24 veces al día. Se pueden utilizar funciones y/o procedimientos como apoyo.

```
CREATE FUNCTION libre_ocupa () RETURNS VARCHAR(35) DETERMINISTIC
BEGIN
DECLARE resul VARCHAR(35);
UPDATE habitaciones SET estado="Ocupada" WHERE numhab IN
(SELECT numhab FROM ocupaciones WHERE CURDATE() BETWEEN fentrada AND fsalida);
SET resul=(SELECT "Habitaciones actualizadas");
RETURN resul;
END //
```

```
CREATE EVENT IF NOT EXISTS consumicion
ON SCHEDULE EVERY 1 HOUR
DO
SELECT libre_ocupa() FROM ocupaciones WHERE CURDATE() BETWEEN fentrada AND fsalida;
```

5. Crear un evento que se ejecute cada hora y media para hacer copia de seguridad de las tablas consumiciones y ocupaciones. El evento debe comenzar su ejecución ahora mismo.

```
CREATE EVENT IF NOT EXISTS consumicion
ON SCHEDULE EVERY 1.5 HOUR
STARTS NOW()
DO
DROP TABLE IF EXISTS ocupaciones_bk,consumiciones_bk;
CREATE TABLE ocupaciones_bk LIKE ocupaciones;
INSERT INTO ocupaciones_bk SELECT * FROM ocupaciones;
CREATE TABLE consumiciones_bk LIKE consumiciones;
INSERT INTO consumiciones_bk SELECT * FROM consumiciones;
```

```
> [] consumiciones_bk 19
> [] habitaciones 10
> [] ocupaciones 18
> [] ocupaciones_bk 18
```