Escribe en este documento el código SQL necesario para realizar las siguientes acciones usando el SGBD MySQL sobre la BBDD empresa\_dep (disponible en Florida Oberta).

1. Dar de alta un nuevo departamento llamado SERVICIOS MEDICOS, con código 80, pendiente de ubicar. Posteriormente, dar de alta un empleado con código 7935, de apellido 'GIL' y de profesión 'PSICOLOGO', asignado al departamento de 'SERVICIOS MEDICOS' (obteniendo dinámicamente el código con una subconsulta, NO ES VÁLIDO ASIGNARLO DIRECTAMENTE) y dependiente directamente del 'PRESIDENTE' de la compañía, que sabemos que tiene código de empleado 7839. La fecha de alta debe ser la actual del sistema en el momento de la inserción y de momento no se le asigna ningún salario ni comisión. Finalmente, ubica el departamento ‘SERVICIOS MEDICOS’ en la ciudad de Valencia.

Se debe poder hacer sin poder averiguar previamente el código del departamento de 'SERVICIOS MÉDICOS'.

INSERT INTO dept (NUM\_DEP, DNOMBRE, LOC) VALUES ('80', 'SERVICIOS MEDICOS', 'Pendiente de ubicar');

SET AUTOCOMMIT=0;INSERT INTO emp (NUM\_EMP, APELLIDO, OFICIO, JEFE, FECHA\_ALTA, NUM\_DEP) VALUES ('7935', 'GIL', 'PSICOLOGO','7839',NOW(),(SELECT dept.NUM\_DEP FROM dept WHERE DNOMBRE="SERVICIOS MEDI"));

UPDATE dept SET LOC='VALENCIA' WHERE NUM\_DEP='80';

1. Dar de alta un pedido (con su detalle) con código 622, con fecha de hoy, sin tipo definido, para el cliente 100, para ser enviado dentro de quince días y con total no definido.

SET AUTOCOMMIT=0;

INSERT INTO pedido (NUM\_PEDIDO, FECHA\_PEDIDO, COD\_CLIENTE, FECHA\_ENVIO) VALUES ('622', NOW(),'100', DATE\_ADD(NOW(),INTERVAL 15 DAY));

Para el detalle, usar los datos del pedido 617, con un incremento del 10% en el precio de venta y en el importe (obteniéndolos dinámicamente con una subconsulta, NO ES VÁLIDO ASIGNARLOS DIRECTAMENTE). (ERROR)

set autocommit=0;

UPDATE empresa\_dep.detalle

SET

PRECIO\_VENTA = (PRECIO\_VENTA \* 0.10) + PRECIO\_VENTA

AND IMPORTE = (IMPORTE \* 0.10) + IMPORTE

WHERE

detalle.NUM\_PEDIDO = (SELECT

NUM\_PEDIDO

FROM

detalle

WHERE

detalle.NUM\_PEDIDO = '617');

Posteriormente, actualizar el total del pedido creado como la suma del importe de todos los detalles registrados en la BD para ese pedido (recuerda que el detalle corresponde a cada una de las líneas de pedido para un pedido concreto y el total del pedido debe almacenar el total del importe de todas las líneas de pedido).

SELECT SUM(IMPORTE) AS' TOTAL' FROM detalle WHERE NUM\_PEDIDO="617";

PISTA: busca la función adecuada en el manual de MySQL version 5.7 para añadir valores de tiempo (intervalos) a un valor de fecha.

1. Realizar las modificaciones pertinentes en la base de datos para que:
   1. Al eliminar un pedido, se eliminen todas las líneas de detalle del pedido.

ALTER TABLE empresa\_dep. detalle ADD FOREIGN KEY (NUM\_PEDIDO) REFERENCES pedido(NUM\_PEDIDO) ON CASCADE DELETE;

* 1. Al eliminar un cliente, se eliminen todos los pedidos del cliente.

ALTER TABLE empresa\_dep.pedido ADD FOREIGN KEY (NUM\_PEDIDO) REFERENCES client(COD\_CLIENTE) ON CASCADE DELETE;

1. Crear una vista que muestre el presupuesto global anual destinado a la nómina de cada departamento (asumimos que el presupuesto global corresponde al total del salario de todos sus empleados en para 14 meses (salario mensual más dos pagas extra)). La vista debe mostrar el número del departamento, su nombre y el presupuesto global para nóminas y debe ordenar el resultado de mayor a menor presupuesto.

Una vez creada, comprueba si se pueden realizar operaciones de inserción, modificación y/o borrado de datos en esta vista y explica porqué.

1. Crea una transacción para dar de alta de forma segura el pedido del ejercicio 2 en la base de datos, de forma que se asegure la coherencia de los datos si ocurre algún problema durante la ejecución de las consultas SQL necesarias para efectuar el alta.

En este caso, hay al menos una situación en la que hay que prever obligatoriamente una transacción: en el momento de dar de alta un nuevo pedido habrá que hacer de forma conjunta, como una única operación o unidad, la secuencia de operaciones siguientes:

* Consultar el código del cliente.
* Insertar el pedido.
* Consultar la tabla de productos para obtener el código de producto de cada producto que adquiera el cliente.
* Insertar el detalle de cada producto adquirido.
* Actualizar el total de la factura en la tabla de pedidos.

Teniendo esto en cuenta, crear las sentencias SQL para llevar a cabo las operaciones siguientes:

1. Desactivar la opción por defecto que permite que los resultados de cada instrucción SQL se hagan efectivos al momento.
2. Definir que las transacciones ocurran de forma aislada a otras transacciones de forma global.
3. Bloquear las tablas necesarias para lectura o para escritura.
4. Iniciar la transacción.
5. Crear las sentencias SQL para dar de alta el pedido del ejercicio 2.
6. Hacer que los cambios de la transacción se hagan efectivos.
7. Desbloquear las tablas.