Almacenes de Datos

Actividad 1

**Descripción del problema.**

Realice a partir de la siguiente problemática, diseñe el diagrama Entidad- Relación y las siguientes consultas en SQL.

El Restaurante “Cornelio Reyna” actualmente con sede en Guadalajara, se ha caracterizado por el alto flujo de **Clientes** que allí llegan con el fin de conseguir todos aquellos productos, tales como: Platos típicos de la región, Platos internacionales, etc. El dueño del restaurante debe realizar la reserva de **Mesas** (Solicitar número de personas, **Área** de Fumador o no, Día y Hora) y la Gestión de **Pedidos** (Productos de limpieza, de Alimentos, Control de Gastos y Alta y Baja de Meseros y Chef).

La **Reserva** de mesas puede llevarse a cabo por teléfono o directamente en el restaurant (Acción hecha por el capitán de meseros). El cliente escoge el plato de su preferencia y establece su preparación (es decir, si lleva ciertos condimentos o no), por lo cual se prosigue a informarle el precio, las mesas están separadas para el Fumador y de No Fumador y llevan un indicador del número de comensales en la mesa.

El dueño busca automatizar este negocio, de tal forma que un mesero tenga asignado cierto número de mesas y si un comensal solicita un platillo a otro mesero, pueda el otro mesero ingresar dicho elemento. Las **Formas de Pago** para el cliente son tarjeta de crédito y débito, el establecimiento no acepta pago en efectivo. La contratación de un mesero, está a cargo del capitán de meseros, el cual deberá de ingresar los datos personales (nombre completo, fecha de nacimiento, domicilio, teléfono, celular, 2 referencias personales y disponibilidad de horario y asignar el número de mesas). También el capitán de meseros, podrá dar de baja a meseros o administrar las mesas (las mesas se identifican un número consecutivo) para cada mesero.

El dueño requiere que la **Orden** de los comensales, pase inmediatamente con el Chef en jefe, y no usar el típico papel de anotaciones. El Chef en jefe, observa en un monitor la llegada de las órdenes (número de mesa y platillo), en 3 columnas, la 1era columna representa a los clientes que han pedido por Primera vez, la 2ª columna, clientes que desean cambiar o agregar platillos a cuenta y la última columna, Clientes VIP, cuya atención debe ser inmediata.

El dueño le interesa además tener un listado de los ingredientes utilizados en la elaboración de cada platillo, de tal forma que cuando un comensal pregunte sobres su elaboración, el mesero tenga la información disponible o si el comensal pregunta si está elaborado con “X” producto que no es de su agrado, el mesero podrá responderle.

Al final del día, el dueño requiere saber el número total de mesas atendidas, la ganancia de cada mesa, la ganancia total y el listado de los platillos más vendidos, así como el listado de insumos que deberá comprar y estar listo para el día siguiente.

1. Listar el nombre del mesero y el número total de mesas que atendió por día.

SELECT employees.nombre, fecha, tables.numero FROM reservations

INNER JOIN tables ON tables.id = reservations.table\_id

INNER JOIN employees ON employees.id = tables.employee\_id WHERE fecha = "";

1. Obtener el número total de reservaciones realizadas por teléfono y personalmente.

SELECT count(\*), tipo\_reservacion FROM reservations GROUP BY tipo\_reservacion;

1. Mostrar todos los ingredientes para un platillo determinado.

SELECT ingredientes FROM platillos WHERE nombre = "nombre" ;

1. Lista el listado de artículos de limpieza o insumos que están por terminar.

SELECT nombre, existencia FROM products WHERE existencia < 6;

1. Crear un disparador que informe cuando un insumo ha llegado a su límite para quitarlo del menú, no es un borrado, solo indicador de que se ha terminado.

DELIMITER $$

CREATE TRIGGER mensaje AFTER UPDATE ON products

FOR EACH ROW

BEGIN

IF (new.existencia < 6)

THEN

SELECT CONCAT("Stock bajo en el producto ", NEW.nombre, " quedan ", new.existencia , " piezas") INTO OUTFILE '/tmp/respuesta.txt';

End IF;

END $$

END;

1. Mostrar los platillos más consumidos separados por nacional o internacional.

SELECT COUNT(platillo\_id), platillos.tipo\_platillo FROM orders

INNER JOIN platillos ON platillos.id = orders.platillo\_id GROUP BY platillo\_id ORDER BY platillos.tipo\_platillo;

1. Listar el listado de platillos consumidos por mesa ordenados por mesa.

SELECT tables.numero, platillos.nombre, reservations.id FROM orders

INNER JOIN reservations ON reservations.id = orders.reservation\_id

INNER JOIN tables ON tables.id = reservations.table\_id

INNER JOIN platillos ON platillos.id = orders.platillo\_id ORDER BY id ;

1. Crear un procedimiento almacenado en el cual cuando se compre más insumos (que estaban en el mínimo), aparte de incrementar su stock, deberá habilitar en el menú de dicho platillo.

DELIMITER $$

CREATE PROCEDURE actualizar\_almacen (IN cantidad INT, IN product\_id INT) BEGIN

DECLARE old int;

SELECT existencia INTO old FROM products WHERE id = product\_id;

UPDATE products SET existencia = old + cantidad WHERE id = product\_id;

END $$

DELIMITER ;

1. Mostrar un menú de platillos (separados/agrupados por Nacional/Internacional)

SELECT tipo\_platillo FROM platillos where tipo\_platillo = “Internacional/Nacional” ;

1. Contar el número de platillos atendidos por columna al día.

Columna 1 🡪Primera orden

Columna 2 🡪Más de una orden

Columna 3 🡪 VIP

SELECT COUNT(\*)FROM reservations INNER JOIN orders ON orders.reservation\_id = reservations.id WHERE fecha = “fecha” AND orders.estado = “estado a buscar”;

1. Agrupar por ingredientes los platillos. El ingrediente o ingredientes es dado por el usuario.

SELECT nombre, ingredientes, descripcion FROM platillos WHERE ingredientes LIKE “%dato%%dato%”;

1. Mostrar las ganancias totales (Ventas por día) agrupadas por columnas.

SELECT SUM(venta), fecha FROM reservations GROUP BY fecha;

1. Sentencias en SQL para:

* Alta |
* Eliminar |
* Modificar | platillos, insumos, meseros, chef y jefe de meseros
* Buscar |

El proceso de eliminación es lógico no físico.

1. Crear un disparador cuando se adicione un nuevo platillo para un cliente VIP y se envíe una alerta al jefe de meseros para indicarle que mesa y el nombre del mesero que lo está atendiendo.

BEGIN declare a varchar(50);

declare b varchar(50);

declare c varchar(50);

SELECT customers.tipo\_cliente into a FROM reservations

INNER JOIN customers ON reservations.customer\_id = customers.id where reservations.id = new.reservation\_id;

IF (a = "VIP")

THEN

SELECT tables.numero INTO b FROM reservations INNER JOIN tables ON reservations.table\_id = tables.id WHERE reservations.id = new.reservation\_id;

SELECT employees.nombre INTO c FROM orders INNER JOIN employees ON employees.id = orders.employee\_id WHERE orders.id = new.id;

SELECT CONCAT("Cliente VIP en la mesa ", b, " atendida por ", c) INTO outfile "/tmp/mesero.txt";

End IF;

END