UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Proyecto Final - Base de Datos para un restaurante

INTEGRANTES DEL EQUIPO: DATA MATRIX Studios

Alvarez Saldaña Ariana Lizeth
López González Erick
R Alan

PROFESOR

Arreola Fernando

ASIGNATURA

Bases de Datos

FECHA DE ENTREGA

26 de Noviembre de 2023

Índice

1. Introduccion

Este documento aborda la resolución de un problema relacionado con la creación de una base de datos para un restaurante que busca digitalizar sus operaciones. Para ello, se presenta un diseño, detallado más adelante en este documento, con el objetivo de cumplir con lo siguiente:

- Almacenar información sobre:
  - Empleados (Cocineros, Meseros y Administrativos).
  - Dependientes económicos de los empleados.
  - Platillos y bebidas.
  - La Categoría de los platillos y bebidas.
  - El contenido completo de una Orden
  - Información del Cliente para elaborar su Factura de consumo.

Además, se plantean los siguientes requerimientos específicos:

- Mostrar, a partir del número de un empleado, la cantidad de órdenes registradas por el restaurante en el día, así como el total pagado por dichas órdenes.
- Visualizar una vista que detalle la información del platillo más vendido.
- Permitir la obtención de los nombres de los productos que no se encuentran disponibles.
- Mostrar una vista que contenga la información necesaria para simular una factura de una orden.
- Al proporcionar una fecha o un rango de fechas, obtener el total del número de ventas y el monto total de ventas en ese período de tiempo.

Estos objetivos y requerimientos guiarán el diseño y desarrollo de la base de datos del restaurante.

2. Plan de trabajo

Para el proyecto, es necesario elaborar los siguientes productos:

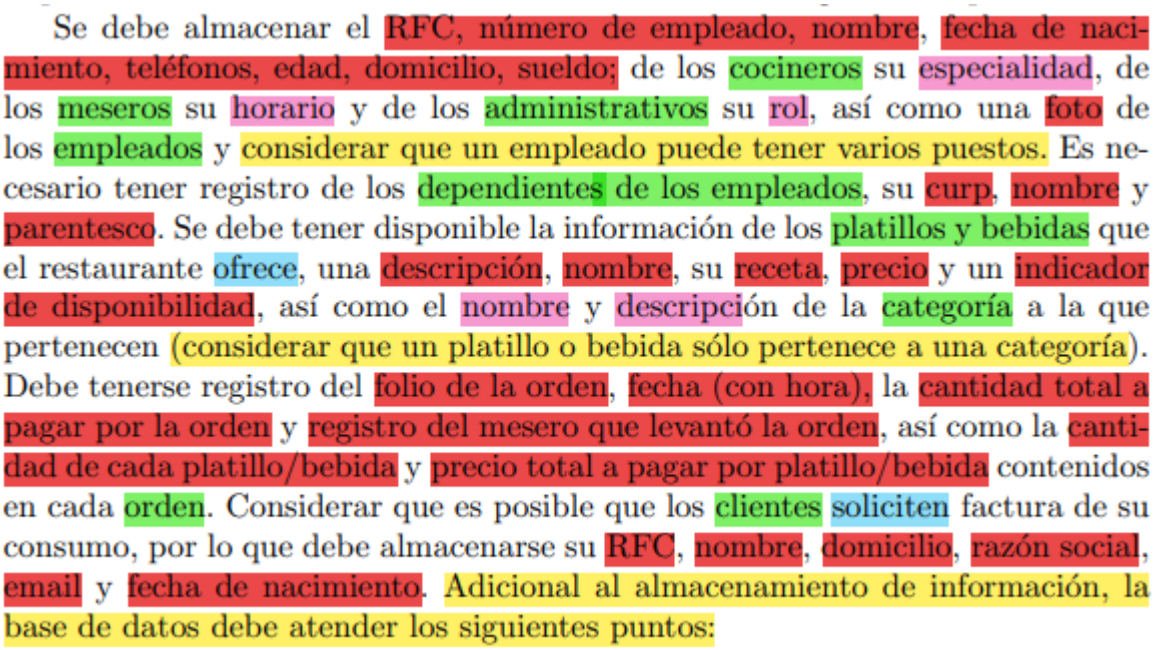
- La documentación pertiente.
- Los códigos fuente del Modelo Entidad Relación y Modelo Relacional.
- El script de creación de la base de datos y las tablas que almacenaran la información.
- El script para agregar información a las tablas.
- Los scripts de toda la programación a nivel de la base de datos.
- El código implementado para la etapa de presentación.
- La presentación para exponer el proyecto.

Dichas actividades se repartieron de la siguiente forma enter los integrantes:

- Alvarez Saldaña Ariana Lizeth
  - Elaboración del modelo relacional.
  - Creación de los insert para las tablas creadas.
- López González Erick
  - Creación del modelo entidad relación.
  - Creación del script de la base de datos y las tablas.
  - Creación del scripts de programación a nivel BD (Trigger e índice).
  - Creación del documento en Latex.
- R Alan
  - Creación del script que permita visualizar la inforación de un empleado por medio de una app.

3. Diseño

La primera parte de diseño consistió en el análisis del enunciado propuesto:



Se debe almacenar el RFC, número de empleado, nombre, fecha de nacimiento, teléfonos, edad, domicilio, sueldo; de los cocineros su especialidad, de los meseros su horario y de los administrativos su rol, así como una foto de los empleados y considerar que un empleado puede tener varios puestos. Es necesario tener registro de los dependientes de los empleados, su curp, nombre y parentesco. Se debe tener disponible la información de los platillos y bebidas que el restaurante ofrece, una descripción, nombre, su receta, precio y un indicador de disponibilidad, así como el nombre y descripción de la categoría a la que pertenecen (considerar que un platillo o bebida sólo pertenece a una categoría). Debe tenerse registro del folio de la orden, fecha (con hora), la cantidad total a pagar por la orden y registro del mesero que levantó la orden, así como la cantidad de cada platillo/bebida y precio total a pagar por platillo/bebida contenidos en cada orden. Considerar que es posible que los clientes soliciten factura de su consumo, por lo que debe almacenarse su RFC, nombre, domicilio, razón social, email y fecha de nacimiento. Adicional al almacenamiento de información, la base de datos debe atender los siguientes puntos:

Figura 1: Identificación inicial. Donde: Entidad - Verde, Atributo - Rojo, Atributo de entidad derivada - Rosa, Relaciones - Azul, Consideraciones - Amarillo.

De esta primera parte, se lograrón identificar a las siguientes entidades:

- Empleados (Cocineros, Meseros y Administrativos).
- Dependientes económicos de los empleados.
- Platillos y bebidas.
- La Categoría de los platillos y bebidas.
- La entidad Orden del cliente.
- Información del Cliente para elaborar su Factura de consumo.

Es entonces, y apoyados de las relaciones explícitas, relacionamos de la siguiente manera las entidades:

- Empleados tiene Dependientes (1:m)

  Donde: Un empleado tiene niguno o varios dependientes, pero un dependiente solo se relaciona con un empleado.

- Mesero sirve una Orden (1:m)

  Donde: Un mesero sirve una o muchas ordenes, pero una orden solo puede ser serivida por un mesero.

- Orden genera una Factura (1:1)

  Donde: Una orden genera una factura y una factura es generada por una orden.

- Una Factura es solicitada por un Cliente (m:1)

  Donde: Un cliente solicta una o varias facturas, pero una factura solo es solicitada por un cliente.

- Una Orden contiene Alimentos (m:m)

  Donde: Una orden contiene uno o muchos alimentos y uno o varios alimetos pueden ser contenidos en una orden.

- Un Alimento pertenece a una Categoria (m:1)

  Donde: Una categoría puede tener varias alimentos, pero un alimento puede tener opcionalmente una categoría.

Igualmente, se analizarón algunos detalles en los atributos y entidades propuestas:

- Entidad Dependiente:

  Para esta entidad se llegó a la conclusión de que es una entidad débil relacionada a la entidad empleado porque en la base de datos no nos interesa guardar dicha información a no ser que esté ligada a un empleado.

- Atributo teléfono de la entidad Empleado:

  Como el empleado puede guardar varios teléfonos, se tuvo que indicar que este atributo es multivaluado.

- Atributo total a pagar de la entidad Orden:

  Este atributo es un total a pagar que se calculará para mostrar en la factura, por lo que se indica que es del tipo derivado.

- Atributos ap Mat (apellido Materno) de todas la entidades que lo contienen:

  Finalmente, una persona puede o no tener un seguno apellido, por ello es necesario indicar que es opcional.

- Atributo cantidad platillo en la relación orden y alimento:

  Como nos interesa saber en cada orden cuantos platillo se pidieron, fue necesario integrar el atributo pertinente para llevar el registro y utilizarlo para el total a pagar.

Finalmnete, el resultado de este análisis es el siguiente modelo:

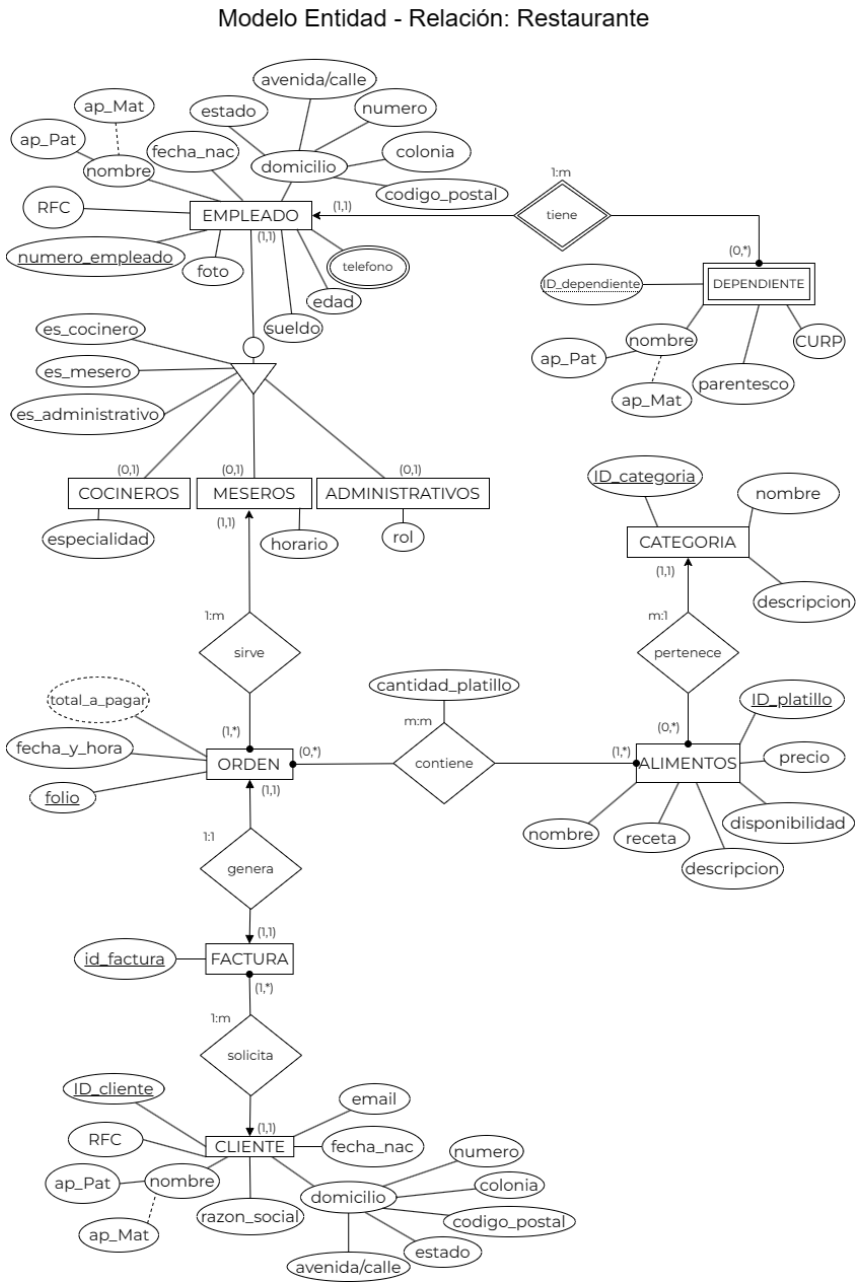


Figura 2: Versión final del modelo.

4. Implementación

Para la implementación de las tablas generadas en el modelo relacional, se codificó lo siguiente:

Listing 1: Creación de las tablas que almacenarán la información.

```
-- @Autor: DataMatrix Studios
-- @Fecha creacion: 2023
-- @Descripcion: Creacion de las tablas para la BD de un restaurante.

-- Creacion de la base de datos
-- create database Restaurate;
-- \c Restaurate

--
-- ENTIDAD EMPLEADO
--

create table EMPLEADO
(
        Numero_empleado numeric(4, 0) PRIMARY KEY,
        RFC varchar(13) not null,
        Nombre_empleado varchar(50) not null,
        Apellido_paterno varchar(50) not null,
        Apellido_materno varchar(50) null,
        Fecha_nacimiento Date not null,
        Edad numeric(3, 0) not null,
        Sueldo numeric(8, 2) not null,
        Foto bytea not null,
        -- Domicilio
        Estado varchar(50) not null,
        Avenida_calle varchar(50) not null,
        Numero_exterior_vivienda numeric(5, 0) null,
        Colonia varchar(50) not null,
        Codigo_postal numeric(6, 0) not null,
        -- Rol del empleado
        Es_cocinero bit null,
        Es_mesero bit null,
        Es_administrativo bit null
);

--
-- ATRIBUTO MULTIVALUADO DE EMPLEADO - TELEFONO
--

create table EMPLEADO_TELEFONO
```

```
(
        Numero_empleado numeric(4, 0),
        Telefono_Id numeric(10, 0),
        constraint EMPLEADO_TELEFONO_PK PRIMARY KEY(Numero_empleado,
    Telefono_Id)
);

--
-- ENTIDAD COCINERO - TIPO DE EMPLEADO
--

create table COCINERO
(
        Numero_empleado_cocinero numeric(4, 0) PRIMARY KEY,
        Especialdad varchar(50),
        constraint Numero_empleado_cocinero_Fk FOREIGN KEY
    (Numero_empleado_cocinero) references EMPLEADO(Numero_empleado)
);

--
-- ENTIDAD MESERO - TIPO DE EMPLEADO
--

create table MESERO
(
        Numero_empleado_mesero numeric(4, 0) PRIMARY KEY,
        Horario_descripcion varchar(500) not null,
        constraint Numero_empleado_mesero_Fk
    FOREIGN KEY (Numero_empleado_mesero)
    references EMPLEADO(Numero_empleado)
);

--
-- ENTIDAD ADMINISTRATIVO - TIPO DE EMPLEADO
--

create table ADMINISTRATIVO
(
        Numero_empleado_administrativo numeric(4, 0) PRIMARY KEY,
        Rol varchar(50) not null,
        constraint Numero_empleado_administrativo_Fk FOREIGN KEY
    (Numero_empleado_administrativo) references EMPLEADO(Numero_empleado)
);

--
-- ENTIDAD DEPENDIENTE - ENTIDAD DEBIL DE EMPLEADO
--
```

```
create table DEPENDIENTE
(
        ID_dependiente numeric(10, 0) not null,
        Numero_empleado numeric(4, 0) not null,
        Curp varchar(13) not null,
        Nombre_dependiente varchar(50) not null,
        Apellido_paterno varchar(50) not null,
        Apellido_materno varchar(50) null,
        Parentesco varchar(50) not null,
        constraint dependiente_pk PRIMARY KEY(ID_dependiente,
    Numero_empleado), constraint Numero_empleado_Fk
    FOREIGN KEY (Numero_empleado)
    references EMPLEADO(Numero_empleado)
);

--
-- ENTIDAD CLIENTE
--

create table CLIENTE
(
        ID_cliente numeric(10, 0) PRIMARY KEY,
        RFC_cliente varchar(13) not null,
        Nombre_cliente varchar(50) not null,
        Apellido_paterno varchar(50) not null,
        Apellido_materno varchar(50) null,
        Fecha_nacimiento Date not null,
        Razon_social varchar(500) not null,
        Email varchar(200) not null,
        -- Domicilio
        Estado varchar(50) not null,
        Avenida_calle varchar(50) not null,
        Numero_exterior_vivienda numeric(5, 0) not null,
        Colonia varchar(50) not null,
        Codigo_postal numeric(6, 0) not null
);

--
-- ENTIDAD FACTURA
--

create table FACTURA
(
        ID_factura numeric(10, 0) PRIMARY KEY,
        ID_cliente numeric(10, 0) not null,
        constraint ID_cliente_fk FOREIGN KEY (ID_cliente)
```

```
    references CLIENTE(ID_cliente)
);

--
-- ENITDAD ORDEN
--

create table ORDEN
(
        Folio_orden varchar(14) PRIMARY KEY,
        Fecha_orden date not null,
        Total_a_pagar numeric(12, 2) not null,
        Mesero_atendio numeric(4, 0) not null,
        ID_factura numeric(10, 0) not null,
        constraint Mesero_atendio_fk FOREIGN KEY
    (Mesero_atendio) references MESERO(Numero_empleado_mesero),
        constraint ID_factura_fk FOREIGN KEY (ID_factura)
    references FACTURA(ID_factura),
        constraint chk_folio_orden check (Folio_orden = 'ORD-%')
);

--
-- ENTIDAD CATEGORIA
--

create table CATEGORIA
(
        ID_Categoria numeric(3, 0) PRIMARY KEY,
        Nombre_categoria varchar(100) not null,
        Descripcion_categoria varchar(500) not null
);

--
-- ENTIDAD ALIMENTOS (PLATILLO Y BEBIDA)
--

create table ALIMENTOS
(
        ID_ALIMENTO numeric(6, 0) PRIMARY KEY,
        Nombre_platillo varchar(100) not null,
        Receta_platillo varchar(1000) not null,
        Descripcion_platillo varchar(500) not null,
        Disponibilidad_platillo varchar(10) not null,
        Precio_platillo numeric(10, 2) not null,
        ID_Categoria numeric(3, 0) not null,
        constraint ID_Categoria_Fk FOREIGN KEY (ID_Categoria)
 references CATEGORIA(ID_Categoria)
```

```
);

--
-- ENITDAD RELACION M:M - RELACION ENTRE ORDEN Y PLATILLO
--

create table ORDEN_CONTIENE_PLATILLO
(
        ID_ALIMENTO numeric(6, 0) not null,
        Folio_orden varchar(14) not null,
        Cantidad_platillo numeric(5, 0) not null,
        constraint ID_ALIMENTO_Folio_orden_pk PRIMARY KEY
    (ID_ALIMENTO, Folio_orden),
        constraint ID_ALIMENTO_Fk FOREIGN KEY (ID_ALIMENTO)
    references ALIMENTOS(ID_ALIMENTO),
        constraint Folio_orden_Fk FOREIGN KEY (Folio_orden)
    references ORDEN(Folio_orden)

);
```

Este script (s-01-creacion-tablas.sql) tiene que ser el primer ejecutado pues creará las tablas pertinente, donde igualmente, en caso de no tener la base creada aún, se pueden quitar los comentarios de las líneas abajo del comentario Çreación de la base de datosçrear la BD.

A continuación, se necesitan ejecutar los triggers e índices que coexistiran en la base de datos.

Para el índice, se propuso colocar uno en la tabla Empleado y otro en la tabla Cliente:

Listing 2: Creación de índece para empleado y cliente.

```
-- @Autor: DataMatrix Studios
-- @Fecha creacion:  2023
-- @Descripcion: Crear al menos, un indice, del tipo
--                que se prefiera y donde se prefiera.

--
-- Indice para Empleado
--

CREATE INDEX rfc_empleado_idx ON EMPLEADO(RFC);

--
-- Indice para Cliente
--

CREATE INDEX rfc_cliente_idx ON CLIENTE(RFC_cliente);
```

La elección de estos dos índices se debió a que podemos facilitar la búsqueda para los empleados y clientes a través de su RFC, un dato que se otorga comúnmente para generar la factura del consumo y que para los empleados se puede utilizar igualmente para facilitar su búsqueda.

Para el trigger se plantea lo siguiente:

Listing 3: Creación del trigger.

```
--@Autor: DataMatrix Studios
--@Fecha creacion: 2023
--@Descripcion: Se busca suplir lo siguiente: Cada que se agregue
--                un producto a la orden, debe actualizarse los
--                totales (por producto y venta), asi como validar
--                que el producto este disponible


CREATE OR REPLACE FUNCTION actualizar_totales_y_validar_producto()
RETURNS TRIGGER AS $$
DECLARE
    total_por_producto NUMERIC(12, 2);
    total_venta NUMERIC(12, 2);
BEGIN
    -- Validar si el producto esta disponible
    IF NOT EXISTS (
        SELECT 1
        FROM ALIMENTOS
        WHERE ID_ALIMENTO = NEW.ID_ALIMENTO AND Disponibilidad_platillo = '
    ) THEN
        RAISE EXCEPTION 'El producto no se encuentra disponible';
    END IF;

    -- Calcular el total por producto (cantidad * precio)
    total_por_producto := NEW.Cantidad_platillo * (
        SELECT Precio_platillo FROM ALIMENTOS WHERE ID_ALIMENTO = NEW.ID_ALI
    );

    -- Actualizar el total por producto en ORDEN_CONTIENE_PLATILLO
    UPDATE ORDEN_CONTIENE_PLATILLO
    SET total_por_producto = total_por_producto
    WHERE ID_ALIMENTO = NEW.ID_ALIMENTO AND Folio_orden = NEW.Folio_orden;

    -- Calcular el total de la venta (sumar los totales por producto para l
    total_venta := (
        SELECT SUM(total_por_producto) FROM ORDEN_CONTIENE_PLATILLO WHERE F
    );

    -- Actualizar el total de la venta en la tabla ORDEN
    UPDATE ORDEN SET Total_a_pagar = total_venta WHERE Folio_orden = NEW.Fo
```

```
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER actualizacion_totales_orden
AFTER INSERT ON ORDEN_CONTIENE_PLATILLO
FOR EACH ROW
EXECUTE FUNCTION actualizar_totales_y_validar_producto();
```

5. Presentación

Descripcin de lo que hace la modalidad seleccionada como forma de conexión hacia la base de datos.

6. Conclusiones

Alvarez Saldaña Ariana Lizeth: El proyecto tuvo sus complicaciones, sin embargo se logro obtener una comprension con respecto a la base de datos solicitado. Especificando algunas

López González Erick:

En la realización del proyecto, el problema más complicado fue la organización a nivel de equipo. Durante el desarrollo del proyecto, se inicio de forma correcta y se logró tener la propuesta del modelo entidad relación en la segunda semana del mes, pero es a partir de ello que la comunicación en el equipo se corto, esto aunado a que un miembro del equipo decidió retirarse, aumento la carga de trabajo para cada integrante y terminó por desembocar en la entrega incompleta del proyecto. Igualmente, a nivel personal se tuvieron complicaciones a la hora de asignarle tiempo al proyecto a causa de la carga de trabajo en cuanto a tareas, prácticas, proyectos y examenes que trae consigo el fin de semestre.

Hablando sobre el desarrollo del mismo, mi principal reto fue la solución a los problemas relacionados con la programación a nivel de Base de Datos, para intentar su solución se necesito la búsqueda de ejemplos de aplicación a través de la web, así como del apoyo del material brindado por el laboratorio de la asignatura. Si bien el material me ayudó mucho, fue complicado tener que estar manejando postgres a causa de que mi experiencia obtenida en el laboratorio se realizó en Oracle BD.

Finalmente, en cuanto aciertos, considero que el diseño inicial de la base de datos a través del modelo entidad relación, si bien no fue el más óptimo, sirvió como solución inicial para la parte de almacenar los datos propuestos.

R Alan :

Conclusión