



**Universidad Autónoma de Nuevo León**  
**Facultad de Ciencias Fisico Matematicas**



**PIA Base de datos**

**Laboratorio de Base de Datos**

**Idalia Alejandra Delgado Moreno**  
**2132987**

**San Nicolás de los Garza, Nuevo León, al 21 de mayo del 2024**

## 1. Diseño de una base de datos de acuerdo al proceso descrito (30%)

a. Crear el diagrama entidad - relación que describa el proceso de la compañía.

[https://lucid.app/lucidchart/ffeb545d-291b-441d-9022-99c3ad38d06c/edit?viewport\\_loc=1003%2C-1331%2C2673%2C1163%2C0\\_0&invitationId=inv\\_8929c1e4-8b5c-4c5d-bce0-5f9bbea151ba](https://lucid.app/lucidchart/ffeb545d-291b-441d-9022-99c3ad38d06c/edit?viewport_loc=1003%2C-1331%2C2673%2C1163%2C0_0&invitationId=inv_8929c1e4-8b5c-4c5d-bce0-5f9bbea151ba)

### Relaciones

- Un hotel puede tener muchos empleados(1:N)
- Un hotel puede tener muchos restaurantes (1:N)
- Un hotel puede tener muchas habitaciones (1:N)
- Los Empleados solo pueden tener un puesto (1:1)
- Las habitaciones pueden tener múltiples reservaciones siempre y cuando no estén en el mismo periodo de tiempo (1:N)
- Varias reservaciones le pueden pertenecer al mismo cliente (N:1)
- Los clientes pueden tener de 0 a 1 número opcional de tarjeta (1:0)
- Los Restaurantes pueden contener múltiples platillos (1:N)

b. Transformar el diagrama ER a un diagrama relacional, aplicando las 3 reglas de integridad o el proceso de normalización.

[https://lucid.app/lucidchart/ffeb545d-291b-441d-9022-99c3ad38d06c/edit?viewport\\_loc=1003%2C-1331%2C2673%2C1163%2C0\\_0&invitationId=inv\\_8929c1e4-8b5c-4c5d-bce0-5f9bbea151ba](https://lucid.app/lucidchart/ffeb545d-291b-441d-9022-99c3ad38d06c/edit?viewport_loc=1003%2C-1331%2C2673%2C1163%2C0_0&invitationId=inv_8929c1e4-8b5c-4c5d-bce0-5f9bbea151ba)

Mismo Link que el anterior, está al lado

Normalización, la mayoría de las tablas cumplían con las tres reglas de la normalización, que son las siguientes:

#### Primera Forma Normal (1FN)

Asegurarse de que los datos en cada columna son atómicos (es decir, no divisibles).

#### Segunda Forma Normal (2FN)

Asegurarse de que cada columna no clave depende de la clave primaria completa.

#### Tercera Forma Normal (3FN)

Eliminar las dependencias transitivas (es decir, los atributos que dependen de otros atributos que no son clave primaria).

Las únicas donde hice cambios fue en la de Empleados, Reservaciones y clientes y fueron modificadas.

## 2. Script SQL para la creación de la base de datos y cada una de sus tablas, debe incluir la definición de relaciones (llaves primarias y foráneas) y constraints (10%)

Se adjuntará en el ZIP de entrega (el archivo .sql)

Observación: en la tabla de reservaciones, puse el valor por DEFAULT de totalPagar como 0.00 porque este se calcula con la función del inciso g, sumando otrosCargos + comida + noches x precioporNoches.

## 3. La compañía solicita los siguientes stored procedures (SP) y funciones (FN) (20%):

a. SP para el registro de un huésped.

Código del SP:

CREATE PROCEDURE RegistrarHuesped (

```

@nombre VARCHAR(100),
@apellidoPaterno VARCHAR(100),
@apellidoMaterno VARCHAR(100),
@telefono VARCHAR(15),
@correo VARCHAR(100),
@RFC VARCHAR(13),
@cuentaPrincipal VARCHAR(20),
@numCuentaOpcional VARCHAR(20) = NULL
)
AS
BEGIN
    DECLARE @clienteID INT;
    INSERT INTO Clientes (nombre, apellidoPaterno, apellidoMaterno, telefono, correo,
RFC, cuentaPrincipal)
    VALUES (@nombre, @apellidoPaterno, @apellidoMaterno, @telefono, @correo,
@RFC, @cuentaPrincipal);

    SET @clienteID = SCOPE_IDENTITY();

    IF @numCuentaOpcional IS NOT NULL
    BEGIN
        INSERT INTO otraCuenta (clienteID, numCuentaOpcional)
        VALUES (@clienteID, @numCuentaOpcional);
    END
END;

```

#### Cómo usarlo:

EXEC RegistrarHuesped 'Idalia', 'Delgado', 'Moreno', '8132936780', 'idalia@gmail.com', 'IDADM45415231', '45325701151189851';

SELECT \* FROM Clientes;

100 %

Results Messages

	clienteID	nombre	apellidoPaterno	apellidoMaterno	telefono	correo	RFC	cuentaPrincipal
1	1	Idalia	Delgado	Moreno	8132936780	idalia@gmail.com	IDADM45415231	45325701151189851

#### Si usamos la tabla de la cuenta opcional:

EXEC RegistrarHuesped 'Alejandra', 'Moreno', 'Delgado', '8132936790', 'ale3706022@gmail.com', 'ALEMD45415231', '45325701151189852', '45325701151189858';

#### Para ver la tabla con la info completa, creamos un nuevo SP:

```

CREATE PROCEDURE VerClientesYCuentasOpcionales
AS
BEGIN
    SELECT c.clienteID, c.nombre, c.apellidoPaterno, c.apellidoMaterno, c.telefono, c.correo,
c.RFC, c.cuentaPrincipal, oc.numCuentaOpcional
    FROM Clientes c
    LEFT JOIN otraCuenta oc ON c.clienteID = oc.clienteID
END
EXEC VerClientesYCuentasOpcionales

```

	clienteID	nombre	apellidoPaterno	apellidoMaterno	telefono	correo	RFC	cuentaPrincipal	numCuentaOpcional
1	1	Idalia	Delgado	Moreno	8132936780	idalia@gmail.com	IDADM45415231	45325701151189851	NULL
2	2	Alejandra	Moreno	Delgado	8132936790	ale3706022@gmail.com	ALEMD45415231	45325701151189852	45325701151189858

## b. SP para el registro de una reservación.

CREATE PROCEDURE RegistrarReservaciones2

@clienteID INT,  
 @habitacionID INT,  
 @fechaInicio DATE,  
 @fechaFin DATE,  
 @totalPagar DECIMAL(10, 2) = 0.00,  
 @otrosCargos DECIMAL(10, 2),  
 @fechaCheckIn DATE,  
 @horaCheckIn TIME,  
 @fechaCheckOut DATE,  
 @horaCheckOut TIME

AS

BEGIN

DECLARE @reservacionID INT  
 SET @totalPagar = 0.00

INSERT INTO Reservaciones (clienteID, habitacionID, fechaInicio, fechaFin, totalPagar, otrosCargos)

VALUES (@clienteID, @habitacionID, @fechaInicio, @fechaFin, @totalPagar, @otrosCargos)

SET @reservacionID = SCOPE\_IDENTITY()

INSERT INTO CheckInCheckOut (reservaciónID, fechaCheckIn, horaCheckIn, fechaCheckOut, horaCheckOut)

VALUES (@reservacionID, @fechaCheckIn, @horaCheckIn, @fechaCheckOut, @horaCheckOut)

UPDATE habitaciones

SET ocupada = 1

WHERE habitacionID = @habitacionID

END

### Cómo usarlo:

EXEC RegistrarReservaciones2

@clienteID = 4,  
 @habitacionID = 26,  
 @fechaInicio = '2024-06-08',  
 @fechaFin = '2024-06-15',  
 @otrosCargos = 50.00,  
 @fechaCheckIn = '2024-06-08',  
 @horaCheckIn = '12:00:00',  
 @fechaCheckOut = '2024-06-15',  
 @horaCheckOut = '11:00:00';

Y si usamos EXEC VerTodasLasReservaciones; este SP que creé, se puede observar, que el registro que agregó con éxito:

	reservaciónID	nombreCliente	apellidoPaternoCliente	apellidoMaternoCliente	telefonoCliente	correoCliente	RFCCliente	cuentaPrincipalCliente	cuentaOpcionalCliente	habitacionID	nombreHabitacion	precioPorNoche	maxi
1	1	Idalia	Delgado	Moreno	8132936780	idalia@gmail.com	IDADM45415231	45325701151189851	NULL	23	802	450.00	3
2	1	Idalia	Delgado	Moreno	8132936780	idalia@gmail.com	IDADM45415231	45325701151189851	NULL	23	802	450.00	3
3	2	Alejandra	Moreno	Delgado	8132936790	ale3706022@gmail.com	ALEMD45415231	45325701151189852	45325701151189858	26	204	150.00	3
4	3	Mauricio	Ortiz	Torres	81329367880	mautorres@gmail.com	MAUOT45415231	45325701151156851	NULL	2	102	100.00	2
5	4	Mauricio	Ortiz	Torres	81329367880	mautorres@gmail.com	MAUOT45415231	45325701151156851	NULL	2	102	100.00	2
6	6	Blanca	Moreno	González	8132936780	Blanca@gmail.com	BLAMG45415231	45325801151189851	NULL	26	204	150.00	3

2024-06-01 2024-06-05

2024-06-01 2024-06-07

2024-06-01 2024-06-07

2024-06-01 2024-06-07

2024-06-08 2024-06-15

Los dos que están en la habitación 26 están registrados en fechas diferentes.

**c. SP para la consulta de una reservación, para la búsqueda se utiliza el email o RFC, y la fecha de inicio**

CREATE PROCEDURE ConsultarReservacion

@correo VARCHAR(100),

@RFC VARCHAR(13),

@fechaInicio DATE

AS

BEGIN

DECLARE @clienteID INT

SELECT @clienteID = clienteID

FROM Clientes

WHERE correo = @correo OR RFC = @RFC

IF @clienteID IS NOT NULL

BEGIN

SELECT R.reservaciónID, R.clienteID, R.habitacionID, R.fechaInicio, R.fechaFin,  
R.totalPagar, R.otrosCargos,

C.fechaCheckIn, C.horaCheckIn, C.fechaCheckOut, C.horaCheckOut

FROM Reservaciones R

INNER JOIN CheckInCheckOut C ON R.reservaciónID = C.reservaciónID

WHERE R.clienteID = @clienteID AND R.fechaInicio = @fechaInicio

END

ELSE

BEGIN

PRINT 'Cliente no encontrado'

END

END

EXEC ConsultarReservacion @correo = NULL, @RFC = 'ALEMD45415231', @fechaInicio = '2024-06-01';												
100 %												
Results Messages												
	reservaciónID	clienteID	habitacionID	fechaInicio	fechaFin	totalPagar	otrosCargos	fechaCheckIn	horaCheckIn	fechaCheckOut	horaCheckOut	
1	2	2	26	2024-06-01	2024-06-07	800.00	50.00	2024-06-01	12:00:00.0000000	2024-06-07	11:00:00.0000000	

```
EXEC ConsultarReservacion @correo = NULL, @RFC = 'ALEMD45415232', @fechaInicio = '2024-06-01';
```

100 %

Messages

Cliente no encontrado

Completion time: 2024-05-21T17:47:17.0065537-06:00

**d. SP para el registro de un huésped (mismo inciso que el a).**

**e. SP para el registro del consumo de un platillo en restaurant.**

CREATE PROCEDURE RegistrarConsumoRestaurante

@reservacionID INT,

@platilloID INT,

@fechaConsumo DATE,

@costo DECIMAL(10, 2)

AS

BEGIN

INSERT INTO ConsumoRestaurante (reservacionID, platilloID, fechaConsumo, costo)

VALUES (@reservacionID, @platilloID, @fechaConsumo, @costo)

END

EXEC RegistrarConsumoRestaurante

```
SELECT * FROM ConsumoRestaurante;
```

100 %

Invalid object name 'Consumo'

Results Messages

	consumoID	reservaciónID	platilloID	fechaConsumo	costo
1	1	2	3	2024-05-23	80.00

**f. SP para la consulta de la cuenta de un huésped en un restaurante, recibe el id de la habitación, genera el siguiente reporte Nombre de platillo | Precio | Fecha**

CREATE PROCEDURE ConsultarCuentaRestaurante

@habitacionID INT

AS

BEGIN

SELECT M.platillo AS 'Nombre de platillo',

M.precio AS 'Precio',

CR.fechaConsumo AS 'Fecha'

FROM ConsumoRestaurante CR

INNER JOIN Menus M ON CR.platilloID = M.platilloID

INNER JOIN Reservaciones R ON CR.reservaciónID = R.reservaciónID

WHERE R.habitacionID = @habitacionID;

END

EXEC ConsultarCuentaRestaurante @habitacionID = 2;

EXEC ConsultarCuentaRestaurante @habitacionID = 2;			
100 %			
Results Messages			
	Nombre de platillo	Precio	Fecha
1	Pasta	80.00	2024-05-23
2	Pasta	80.00	2024-05-23
3	Pasta	80.00	2024-05-23
4	Pasta	80.00	2024-05-23

**g. FN que calcule el total de la cuenta de una reservación (habitación y consumo de restaurante en caso de existir), recibe el id de la reservación**

```

CREATE FUNCTION CalcularTotalCuentaReservacion
(
    @reservacionID INT
)
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @total DECIMAL(10, 2)

    DECLARE @precioPorNoche DECIMAL(10, 2)
    SELECT @precioPorNoche = precioPorNoche
    FROM Habitaciones
    WHERE habitacionID = (SELECT habitacionID FROM Reservaciones WHERE
reservaciónID = @reservacionID)

    DECLARE @dias INT
    SELECT @dias = DATEDIFF(day, fechaInicio, fechaFin) + 1
    FROM Reservaciones
    WHERE reservaciónID = @reservacionID

    DECLARE @totalHabitacion DECIMAL(10, 2)
    SET @totalHabitacion = @precioPorNoche * @dias

    DECLARE @otrosCargos DECIMAL(10, 2)
    SELECT @otrosCargos = otrosCargos
    FROM Reservaciones
    WHERE reservaciónID = @reservacionID

    DECLARE @totalConsumoRestaurante DECIMAL(10, 2)
    SELECT @totalConsumoRestaurante = SUM(costo)
    FROM ConsumoRestaurante
    WHERE reservaciónID = @reservacionID

    SET @total = @totalHabitacion + @otrosCargos
    IF @totalConsumoRestaurante IS NOT NULL

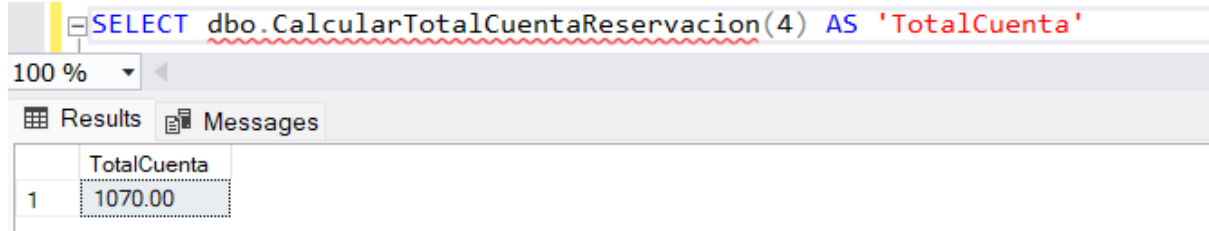
```

```
SET @total = @total + @totalConsumoRestaurante
```

```
RETURN @total
```

```
END
```

```
SELECT dbo.CalcularTotalCuentaReservacion(4) AS 'TotalCuenta'
```



The screenshot shows a SQL query window with the following text: `SELECT dbo.CalcularTotalCuentaReservacion(4) AS 'TotalCuenta'`. Below the query window, the 'Results' tab is active, displaying a single row of data. The column is named 'TotalCuenta' and the value is '1070.00'.

	TotalCuenta
1	1070.00

Después lo actualizamos porque en una función no se puede usar el UPDATE (por lo menos no me dejó):

```
UPDATE Reservasiones
```

```
SET totalPagar = dbo.CalcularTotalCuentaReservacion(4)
```

```
WHERE reservaciónID = 4
```

**h. SP que obtenga el listado las reservaciones entre 2 fechas, recibe una fecha inicial y una final, el filtrado se hace con la fecha de check out de la reservación, debe generar un reporte con el siguiente formato: Nombre de huésped | Num de hab | Check in | Check out | Total de la cuenta**

```
CREATE PROCEDURE ListarReservacionesEntreFechas
```

```
@fechaInicial DATE,
```

```
@fechaFinal DATE
```

```
AS
```

```
BEGIN
```

```
SELECT
```

```
c.nombre + ' ' + c.apellidoPaterno + ' ' + c.apellidoMaterno AS 'Nombre de huésped',
```

```
h.nombre AS 'Num de hab',
```

```
cico.fechaCheckIn AS 'Check in',
```

```
cico.fechaCheckOut AS 'Check out',
```

```
r.totalPagar AS 'Total de la cuenta'
```

```
FROM Reservasiones r
```

```
INNER JOIN Clientes c ON r.clienteID = c.clienteID
```

```
INNER JOIN Habitaciones h ON r.habitacionID = h.habitacionID
```

```
INNER JOIN CheckInCheckOut cico ON r.reservaciónID = cico.reservaciónID
```

```
WHERE cico.fechaCheckOut BETWEEN @fechaInicial AND @fechaFinal
```

```
END
```

```
EXEC ListarReservacionesEntreFechas '2024-01-01', '2024-12-31'
```



EXEC ListarReservacionesEntreFechas '2024-01-01', '2024-12-31'

100 %

Results Messages

	Nombre de huésped	Num de hab	Check in	Check out	Total de la cuenta
1	Idalia Delgado Moreno	802	NULL	2024-06-05	500.00
2	Alejandra Moreno Delgado	204	2024-06-01	2024-06-07	800.00
3	Mauricio Ortiz Torres	102	2024-06-01	2024-06-07	750.00
4	Mauricio Ortiz Torres	102	2024-06-01	2024-06-07	1070.00

\*\*Aparece NULL en el primero porque me equivoqué al hacer el SP de registrar reservaciones, por eso se llama RegistrarReservaciones2 el que puse más arriba\*\*

**i. SP que obtenga el listado de habitaciones disponibles al momento.**

CREATE PROCEDURE ListarHabitacionesDisponibles

AS

BEGIN

SELECT

h.habitacionID AS 'ID de Habitación',

h.nombre AS 'Nombre de Habitación',

h.precioPorNoche AS 'Precio por Noche'

FROM Habitaciones h

WHERE h.ocupada = 0

END

EXEC ListarHabitacionesDisponibles

EXEC ListarHabitacionesDisponibles

100 %

Results Messages

	ID de Habitación	Nombre de Habitación	Precio por Noche
1	1	101	100.00
2	3	103	100.00
3	4	201	150.00
4	5	202	150.00
5	6	203	150.00
6	7	301	200.00
7	8	302	200.00
8	9	303	200.00
9	10	401	250.00
10	11	402	250.00
11	12	403	250.00
12	13	501	300.00
13	14	502	300.00
14	15	503	300.00
15	16	601	350.00
16	17	602	350.00
17	18	603	350.00
18	19	701	400.00
19	20	702	400.00
20	21	703	400.00
21	22	801	450.00
22	24	803	450.00
23	25	104	100.00
24	27	304	200.00
25	28	404	250.00
26	29	504	300.00
27	30	604	350.00

#### 4. Inserts para (5%):

##### a. La creación de los 8 hoteles

INSERT INTO hoteles (nombre, dirección, telRecepcion) VALUES

('Hotel Colina Sur', 'Calle Cuauhtémoc #123, Colonia Juárez, Ciudad de México, CDMX, México', '8123456789'),

('Hotel Valle', 'Avenida Benito Juárez #456, Colonia Centro, Guadalajara, Jalisco, México', '8155555555'),

('Hotel Costa', 'Calle Hidalgo #789, Fraccionamiento Revolución, Monterrey, Nuevo León, México', '8132109876'),

('Hotel Buenas Tardes', 'Boulevard Miguel Hidalgo #321, Colonia Reforma, Puebla, Puebla, México', '5544443333'),

('Hotel Flor de Loto', 'Avenida Juárez #654, Fraccionamiento Las Palmas, León, Guanajuato, México', '5598765432'),

('Hotel Sato', 'Calle Morelos #987, Colonia Del Valle, Tijuana, Baja California, México', '5566667777'),

('Hotel Ba Sing Se', 'Avenida Insurgentes #234, Colonia Centro Histórico, Querétaro, Querétaro, México', '8132109876'),

('Hotel Zaofu', 'Boulevard Adolfo López Mateos #567, Colonia Las Águilas, Morelia, Michoacán, México', '8166667777');

SELECT \* FROM hoteles

100 %

Results Messages

	hotelID	nombre	dirección	telRecepcion
1	1	Hotel Colina Sur	Calle Cuauhtémoc #123, Colonia Juárez, Ciudad de ...	8123456789
2	2	Hotel Valle	Avenida Benito Juárez #456, Colonia Centro, Guadal...	8155555555
3	3	Hotel Costa	Calle Hidalgo #789, Fraccionamiento Revolución, M...	8132109876
4	4	Hotel Buenas Tardes	Boulevard Miguel Hidalgo #321, Colonia Reforma, P...	5544443333
5	5	Hotel Flor de Loto	Avenida Juárez #654, Fraccionamiento Las Palmas, ...	5598765432
6	6	Hotel Sato	Calle Morelos #987, Colonia Del Valle, Tijuana, Baja ...	5566667777
7	7	Hotel Ba Sing Se	Avenida Insurgentes #234, Colonia Centro Histórico, ...	8132109876
8	8	Hotel Zaofu	Boulevard Adolfo López Mateos #567, Colonia Las Á...	8166667777

##### b. La creación de 30 habitaciones

SELECT * FROM habitaciones						
100 %						
Results Messages						
	habitacionID	hotelID	nombre	precioPorNoche	maximoHuespedes	ocupada
1	1	1	101	100.00	2	0
2	2	1	102	100.00	2	1
3	3	1	103	100.00	2	0
4	4	2	201	150.00	3	0
5	5	2	202	150.00	3	0
6	6	2	203	150.00	3	0
7	7	3	301	200.00	4	0
8	8	3	302	200.00	4	0
9	9	3	303	200.00	4	0
10	10	4	401	250.00	2	0
11	11	4	402	250.00	2	0
12	12	4	403	250.00	2	0
13	13	5	501	300.00	3	0
14	14	5	502	300.00	3	0
15	15	5	503	300.00	3	0
16	16	6	601	350.00	4	0
17	17	6	602	350.00	4	0
18	18	6	603	350.00	4	0
19	19	7	701	400.00	2	0
20	20	7	702	400.00	2	0
21	21	7	703	400.00	2	0
22	22	8	801	450.00	3	0
23	23	8	802	450.00	3	1
24	24	8	803	450.00	3	0
25	25	1	104	100.00	2	0
26	26	2	204	150.00	3	1
27	27	3	304	200.00	4	0
28	28	4	404	250.00	2	0
29	29	5	504	300.00	3	0
30	30	6	604	350.00	4	0

### c. La creación de 10 platillos

Como tiene una FK la tabla de Menus, primero agregué la de restaurantes:

INSERT INTO restaurantes (hotelID, nombre, categoria) VALUES

- (1, 'Restaurante Abuelita', 'Mexicana'),
- (2, 'Restaurante Napoli', 'Italiana'),
- (3, 'Restaurante Sabor Oriental', 'Japonesa'),
- (4, 'Restaurante La Cabaña del Chef', 'Mexicana'),
- (5, 'Restaurante La Trattoria Bella Italia ', 'Italiana'),
- (6, 'Restaurante Izakaya Sakura', 'Japonesa'),
- (7, 'Restaurante La Puesta del Gallo', 'Mexicana'),

(8, 'Restaurante Ristorante Al Dente ', 'Italiana');

SELECT \* FROM restaurantes

100 %

Results Messages

	restaurantID	hotelID	nombre	categoria
1	1	1	Restaurante Abuelita	Mexicana
2	2	2	Restaurante Napoli	Italiana
3	3	3	Restaurante Sabor Oriental	Japonesa
4	4	4	Restaurante La Cabaña del Chef	Mexicana
5	5	5	Restaurante La Trattoria Bella Italia	Italiana
6	6	6	Restaurante Izakaya Sakura	Japonesa
7	7	7	Restaurante La Puesta del Gallo	Mexicana
8	8	8	Restaurante Ristorante Al Dente	Italiana

INSERT INTO Menus (restaurantID, platillo, precio) VALUES

(1, 'Tacos', 120.00),

(1, 'Enchiladas', 100.00),

(2, 'Pasta', 80.00),

(2, 'Pizza', 120.50),

(3, 'Sushi', 100.00),

(3, 'Ramen', 70.00),

(4, 'Quesadillas', 40.00),

(4, 'Chilaquiles', 75.00),

(5, 'Lasagna', 85.00),

(5, 'Risotto', 95.00);

SELECT \* FROM Menus

100 %

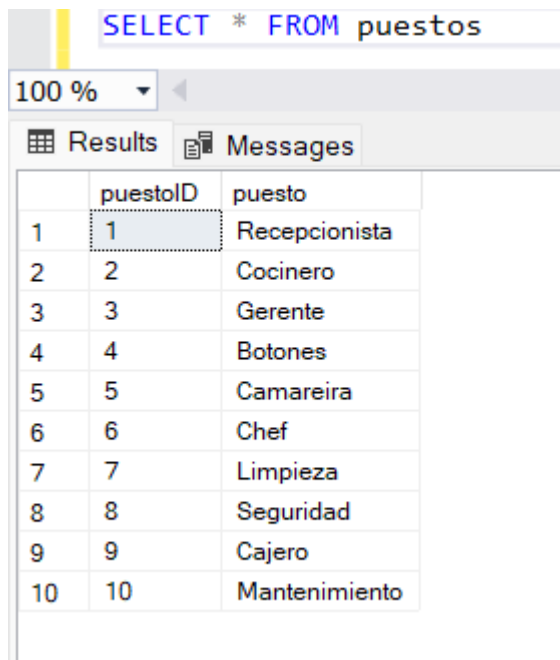
Results Messages

	platilloID	restaurantID	platillo	precio
1	1	1	Tacos	120.00
2	2	1	Enchiladas	100.00
3	3	2	Pasta	80.00
4	4	2	Pizza	120.50
5	5	3	Sushi	100.00
6	6	3	Ramen	70.00
7	7	4	Quesadillas	40.00
8	8	4	Chilaquiles	75.00
9	9	5	Lasagna	85.00
10	10	5	Risotto	95.00

**d. La creación de 20 empleados.**

**Igual que el inciso anterior, en este caso yo hice una tabla que tenía todos los puestos disponibles que hay en el hotel.**

```
INSERT INTO Puestos (puesto)
VALUES
('Recepcionista'),
('Cocinero'),
('Gerente'),
('Botones'),
('Camareira'),
('Chef'),
('Limpieza'),
('Seguridad'),
('Cajero'),
('Mantenimiento');
```



	puestoID	puesto
1	1	Recepcionista
2	2	Cocinero
3	3	Gerente
4	4	Botones
5	5	Camareira
6	6	Chef
7	7	Limpieza
8	8	Seguridad
9	9	Cajero
10	10	Mantenimiento

```
INSERT INTO Empleados (empleadID, hotelID, nombre, apellidoPaterno, apellidoMaterno,
fechaNacimiento, RFC, puestoID, jefeID, fechaIngreso, sueldoMensual, activo, fechaBaja)
VALUES
(1, 1, 'Juan', 'López', 'García', '1990-05-15', 'LOGJ900515123', 1, NULL, '2020-02-10',
1500.00, 1, NULL),
(2, 1, 'María', 'Martínez', 'Hernández', '1985-10-22', 'MAHM851022456', 3, NULL,
'2018-06-20', 3000.00, 1, NULL),
(3, 1, 'Carlos', 'González', 'Rodríguez', '1992-03-08', 'GORC920308789', 2, NULL,
'2021-01-05', 1800.00, 1, NULL),
(4, 2, 'Laura', 'Pérez', 'Sánchez', '1988-07-12', 'PESL880712234', 5, NULL, '2019-11-15',
1400.00, 1, NULL),
(5, 2, 'Pedro', 'Díaz', 'López', '1983-12-30', 'DILP831230567', 4, NULL, '2020-08-28',
1200.00, 1, NULL),
```

(6, 2, 'Ana', 'Ramírez', 'Martínez', '1995-09-18', 'RAMA950918890', 7, NULL, '2022-03-12', 1300.00, 1, NULL),  
 (7, 3, 'Luis', 'Hernández', 'Gómez', '1987-04-05', 'HEGL870405123', 9, NULL, '2017-07-03', 2000.00, 1, NULL),  
 (8, 3, 'Sofía', 'Gutiérrez', 'Fernández', '1993-11-20', 'GUFS931120678', 8, NULL, '2020-06-18', 1900.00, 1, NULL),  
 (9, 3, 'Javier', 'Martínez', 'Jiménez', '1980-02-25', 'MAJJ800225345', 10, NULL, '2019-04-22', 1700.00, 1, NULL),  
 (10, 4, 'Elena', 'Sánchez', 'Pérez', '1991-08-08', 'SAPR910808901', 6, NULL, '2021-09-10', 2500.00, 1, NULL),  
 (11, 4, 'Alejandra', 'Gómez', 'Martínez', '1990-01-03', 'GOMA900103567', 1, NULL, '2020-11-28', 1500.00, 1, NULL),  
 (12, 5, 'Diego', 'Fernández', 'Ramírez', '1984-09-29', 'FERD840929123', 3, NULL, '2017-10-15', 2800.00, 1, NULL),  
 (13, 6, 'Carmen', 'Gómez', 'Hernández', '1996-03-25', 'GOHC960325901', 2, NULL, '2020-02-17', 1800.00, 1, NULL),  
 (14, 6, 'Lucía', 'García', 'Martínez', '1993-10-11', 'GALG931011234', 1, NULL, '2021-11-12', 1600.00, 1, NULL),  
 (15, 7, 'Manuel', 'Martínez', 'González', '1987-04-16', 'MAMG870416567', 8, NULL, '2019-06-05', 2000.00, 1, NULL),  
 (16, 1, 'Juan', 'Pérez', 'López', '1986-07-23', 'PELJ860723123', 1, 1, '2022-01-15', 1500.00, 1, NULL),  
 (17, 2, 'María', 'González', 'Rodríguez', '1990-12-05', 'GONM901205456', 3, 2, '2023-03-20', 3000.00, 1, NULL),  
 (18, 3, 'Carlos', 'Fernández', 'Hernández', '1995-04-18', 'FEHC950418789', 2, 3, '2023-09-05', 1800.00, 1, NULL),  
 (19, 4, 'Laura', 'Hernández', 'Sánchez', '1988-01-30', 'HERS880130234', 6, 4, '2022-08-12', 2500.00, 1, NULL),  
 (20, 5, 'Pedro', 'Martínez', 'Gómez', '1993-06-14', 'MARG930614567', 4, 5, '2023-11-18', 1200.00, 1, NULL);

	empleadoID	hotelID	nombre	apellidoPaterno	apellidoMaterno	fechaNacimiento	RFC	puestoID	jefeID	fechaIngreso	sueldoMensual	activo	fechaBaja
1	1	1	Juan	López	García	1990-05-15	LOGJ900515123	1	NULL	2020-02-10	1500.00	1	NULL
2	2	1	María	Martínez	Hernández	1985-10-22	MAHM851022456	3	NULL	2018-06-20	3000.00	1	NULL
3	3	1	Carlos	González	Rodríguez	1992-03-08	GORC920308789	2	NULL	2021-01-05	1800.00	1	NULL
4	4	2	Laura	Pérez	Sánchez	1988-07-12	PESL880712234	5	NULL	2019-11-15	1400.00	1	NULL
5	5	2	Pedro	Díaz	López	1983-12-30	DILP831230567	4	NULL	2020-08-28	1200.00	1	NULL
6	6	2	Ana	Ramírez	Martínez	1995-09-18	RAMA950918890	7	NULL	2022-03-12	1300.00	1	NULL
7	7	3	Luis	Hernández	Gómez	1987-04-05	HEGL870405123	9	NULL	2017-07-03	2000.00	1	NULL
8	8	3	Sofía	Gutiérrez	Fernández	1993-11-20	GUFS931120678	8	NULL	2020-06-18	1900.00	1	NULL
9	9	3	Javier	Martínez	Jiménez	1980-02-25	MAJJ800225345	10	NULL	2019-04-22	1700.00	1	NULL
10	10	4	Elena	Sánchez	Pérez	1991-08-08	SAPR910808901	6	NULL	2021-09-10	2500.00	1	NULL
11	11	4	Alejandra	Gómez	Martínez	1990-01-03	GOMA900103567	1	NULL	2020-11-28	1500.00	1	NULL
12	12	5	Diego	Fernández	Ramírez	1984-09-29	FERD840929123	3	NULL	2017-10-15	2800.00	1	NULL
13	13	6	Carmen	Gómez	Hernández	1996-03-25	GOHC960325901	2	NULL	2020-02-17	1800.00	1	NULL
14	14	6	Lucía	García	Martínez	1993-10-11	GALG931011234	1	NULL	2021-11-12	1600.00	1	NULL
15	15	7	Manuel	Martínez	González	1987-04-16	MAMG870416567	8	NULL	2019-06-05	2000.00	1	NULL
16	16	1	Juan	Pérez	López	1986-07-23	PELJ860723123	1	1	2022-01-15	1500.00	1	NULL
17	17	2	María	González	Rodríguez	1990-12-05	GONM901205456	3	2	2023-03-20	3000.00	1	NULL
18	18	3	Carlos	Fernández	Hernández	1995-04-18	FEHC950418789	2	3	2023-09-05	1800.00	1	NULL
19	19	4	Laura	Hernández	Sánchez	1988-01-30	HERS880130234	6	4	2022-08-12	2500.00	1	NULL
20	20	5	Pedro	Martínez	Gómez	1993-06-14	MARG930614567	4	5	2023-11-18	1200.00	1	NULL

## 5. Diccionario de datos (5%).

### 1. Tabla: hoteles

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
hotelID	INT		PRIMARY KEY	Identificador único del hotel
nombre	VARCHAR	100	NOT NULL	Nombre del hotel
dirección	VARCHAR	255	NOT NULL	Dirección del hotel
telRecepcion	VARCHAR	15	NOT NULL	Teléfono de recepción

### 2. Tabla: habitaciones

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
habitacionID	INT		PRIMARY KEY	Identificador único de la habitación
hotelID	INT		FOREIGN KEY (hotelID)	Identificador del hotel
nombre	VARCHAR	100	NOT NULL	Nombre de la habitación
precioPorNoche	DECIMAL	10, 2	NOT NULL	Precio por noche de la habitación

maximoHuespedes	INT		NOT NULL	Máximo número de huéspedes
ocupada	BIT		NOT NULL	Indicador de si la habitación está ocupada

### 3. Tabla: restaurantes

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
restaurantID	INT		PRIMARY KEY	Identificador único del restaurante
hotelID	INT		FOREIGN KEY (hotelID)	Identificador del hotel
nombre	VARCHAR	100	NOT NULL	Nombre del restaurante
categoria	VARCHAR	50	NOT NULL	Categoría del restaurante

### 4. Tabla: Menus

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
platilloID	INT		PRIMARY KEY	Identificador único del platillo



restaurantID	INT		FOREIGN KEY (restaurantID)	Identificador del restaurante
platillo	VARCHAR	100	NOT NULL	Nombre del platillo
precio	DECIMAL	10, 2	NOT NULL	Precio del platillo

#### 5. Tabla: Clientes

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
clienteID	INT		PRIMARY KEY	Identificador único del cliente
nombre	VARCHAR	100	NOT NULL	Nombre del cliente
apellidoPaterno	VARCHAR	100	NOT NULL	Apellido paterno del cliente
apellidoMaterno	VARCHAR	100	NOT NULL	Apellido materno del cliente
telefono	VARCHAR	15	NOT NULL	Número de teléfono del cliente
correo	VARCHAR	100	NOT NULL	Correo electrónico del cliente
RFC	VARCHAR	13	NOT NULL	RFC del cliente
cuentaPrincipal	VARCHAR	20	NOT NULL	Número de cuenta principal del cliente

#### 6. Tabla otroCuenta

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
clienteID	INT		FOREIGN KEY (clienteID)	Identificador del cliente
numCuentaOpcional	VARCHAR	20	NOT NULL	Número de cuenta opcional del cliente

#### 7. Tabla Reservasiones

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
reservaciónID	INT		PRIMARY KEY	Identificador único de la reservación
clienteID	INT		FOREIGN KEY (clienteID)	Identificador del cliente
habitacionID	INT		FOREIGN KEY (habitacionID)	Identificador de la habitación
fechaInicio	DATE		NOT NULL	Fecha de inicio de la reservación
fechaFin	DATE		NOT NULL	Fecha de fin de la reservación
totalPagar	DECIMAL	10, 2	NOT NULL DEFAULT 0.00	Total a pagar por la reservación
otrosCargos	DECIMAL	10, 2	NOT NULL	Otros cargos adicionales

#### 8. Tabla CheckInCheckOut

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
id	INT		PRIMARY KEY	Identificador único del registro
reservaciónID	INT		FOREIGN KEY (reservaciónID)	Identificador de la reservación
fechaCheckIn	DATE			Fecha de check-in
horaCheckIn	TIME			Hora de check-in
fechaCheckOut	DATE			Fecha de check-out
horaCheckOut	TIME			Hora de check-out

## 9. Tabla Empleados

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
empleadoID	INT		PRIMARY KEY	Identificador único del empleado
hotelID	INT		FOREIGN KEY (hotelID)	Identificador del hotel
nombre	VARCHAR	100	NOT NULL	Nombre del empleado
apellidoPaterno	VARCHAR	100	NOT NULL	Apellido paterno del empleado
apellidoMaterno	VARCHAR	100	NOT NULL	Apellido materno del empleado

fechaNacimiento	DATE		NOT NULL	Fecha de nacimiento del empleado
RFC	VARCHAR	13	NOT NULL	RFC del empleado
puestoID	INT		FOREIGN KEY (puestoID)	Identificador del puesto
jefeID	INT		FOREIGN KEY (jefeID)	Identificador del jefe (empleado)
fechaIngreso	DATE		NOT NULL	Fecha de ingreso
sueldoMensual	DECIMAL	10, 2	NOT NULL	Sueldo mensual
activo	BIT		NOT NULL	Indicador de si el empleado está activo
fechaBaja	DATE			Fecha de baja (si corresponde)

#### 10. Tabla Puestos

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
puestoID	INT		PRIMARY KEY	Identificador único del puesto
puesto	VARCHAR	50	NOT NULL	Nombre del puesto

#### 11. Tabla ConsumoRestaurante

Nombre de Columna	Tipo de Dato	Tamaño	Restricciones	Descripción
-------------------	--------------	--------	---------------	-------------

consumoID	INT		PRIMARY KEY	Identificador único del consumo
reservacionID	INT		FOREIGN KEY (reservacionID)	Identificador de la reservación
platilloID	INT		FOREIGN KEY (platilloID)	Identificador del platillo
fechaConsumo	DATE		NOT NULL	Fecha del consumo
costo	DECIMAL	10, 2	NOT NULL	Costo del platillo consumido