

Universidad Nacional Mayor de San Marcos

Universidad del Perú, Decana de América

Facultad de Ingeniería de Sistemas e Informática

Centro de Responsabilidad Social y Extensión Universitaria



Módulo I

PROYECTO DE BASE DE DATOS-RESTAURANTE

Jhon Roly Ordoñez Leon

Programa de Especialización en SQL Server

2022

Índice general

1. Descripción del proyecto	1
2. Diagrama Entidad Relación (DER)	2
3. Código de la consulta SQL	3
4. Diagrama de la base de datos	19

Capítulo 1

Descripción del proyecto

Misión

Satisfacer las expectativas de nuestros clientes y lograr que tenga una experiencia memorable.

Visión

Ser destacado como el mejor restaurante en la localidad por nuestra gastronomía, atención y ambiente muy peculiar.

Valores

- Respeto
- Cortesía
- Honestidad
- Solidaridad

Descripción del restaurante

El restaurante es de alta cocina o también denominado *Gourmet*. Es decir, los alimentos son de gran calidad y se sirven a la mesa. El pedido es “a la carta” o se elige de un “menú”, por lo que los alimentos se cocinan al momento. El costo depende del servicio y de la calidad de los platos que se consumen. Existen meseros, dirigidos por un Maitre. El servicio, la decoración, la ambientación, la comida y las bebidas se escogen cuidadosamente.

Capítulo 2

Diagrama Entidad Relación (DER)

A continuación se muestra el Diagrama Entidad Relación (DER) del proceso que sigue el restaurante.

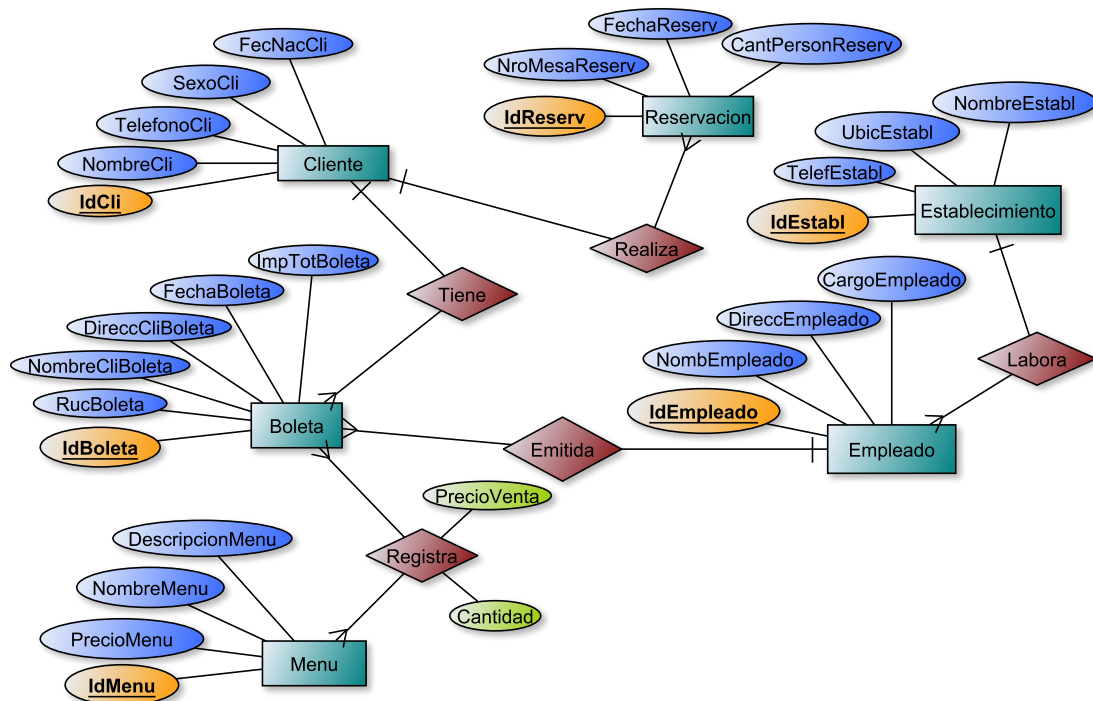


Figura 2.1: Diagrama Entidad Relación (DER) de un Restaurante

La lectura del diagrama es lo siguiente: un cliente puede realizar muchas reservaciones y una reservación puede ser realizado solamente por un cliente; un cliente puede tener muchas boletas y una boleta le pertenece solamente a un cliente; una boleta puede ser emitido por un empleado y un empleado puede emitir muchas boletas; un empleado labora solamente en un establecimiento y en un establecimiento laboran muchos empleados; una boleta puede registrar muchos menús y un menú se puede registrar en muchas boletas.

Capítulo 3

Código de la consulta SQL

```
1  =====
2  --                      Programa de Especializacion en SQL server - UNMSM
3  --                      MODULO I
4  -- Proyecto: Restaurante
5  -- Docente: Lic. Lidia Sanchez
6  -- Alumno: Jhon Roly Ordonez Leon
7  -- Fecha de entrega: 29 de diciembre de 2022
8  =====
9  =====
10 -- VALIDAR EXISTENCIA DE LA BD -----
11 =====
12 USE MASTER
13 GO
14 IF DB_ID('Restaurante') IS NOT NULL -- si es verdadero, la base de datos existe
15 BEGIN -- Inicio del bloque de codigo
16     USE MASTER -- Se activa el master
17     DROP DATABASE Restaurante -- Elimina la base de datos Restaurante
18 END -- Fin del bloque de codigo
19 GO
20 =====
21 -- CREAR BASE DE DATOS -----
22 =====
23 -- Creamos la base de datos
24 CREATE DATABASE Restaurante
25 ON PRIMARY -- filegroup (unidad logica de almacenamiento) por defecto
26 ( -- Archivo primario
27     NAME = 'Restaurante', -- Nombre logico
28     FILENAME = 'D:\BBDD\Restaurante.mdf', -- Nombre fisico
29     SIZE = 5mb, -- Tamano inicial
```

```

30     MAXSIZE = unlimited, -- Tamano maximo
31     FILEGROWTH = 3mb -- Crecimiento con tamano fijo o porcentaje
32 ),
33 (-- Archivo secundario
34     NAME = 'Restaurante_SEC', -- Nombre logico
35     FILENAME = 'D:\BBDD\Restaurante_SEC.ndf', -- Nombre fisico
36     SIZE = 10mb, -- Tamano inicial
37     MAXSIZE = 25mb, -- Tamano maximo
38     FILEGROWTH = 5% -- Crecimiento con tamano fijo o porcentaje
39 )
40 LOG ON
41 (-- Archivo de registro de transacciones
42     NAME = 'Restaurante_LOG', -- Nombre logico
43     FILENAME = 'D:\BBDD\Restaurante_LOG.ldf', -- Nombre fisico
44     SIZE = 8mb, -- Tamano inicial
45     MAXSIZE = 50mb, -- Tamano maximo
46     FILEGROWTH = 4mb -- Crecimiento con tamano fijo o porcentaje
47 )
48 GO
49 -- Veamos las propiedades de la base de datos
50 SP_HELPDB Restaurante -- (SP : Stored Procedure)
51 GO
52 -- Usamos la base de datos Restaurante
53 USE Restaurante
54 GO
55 =====
56 -- CREAR TABLAS -----
57 =====
58 -- Establecimiento
59 CREATE TABLE Establecimiento (
60     IdEstabl INT NOT NULL PRIMARY KEY, -- Llave primaria
61     TelefEstabl CHAR(9) NOT NULL CHECK (TelefEstabl LIKE '
62     9[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
63     UbicEstabl VARCHAR(100) NOT NULL,
64     NombreEstabl CHAR(6) NOT NULL CHECK(NombreEstabl like 'Tullpa'),
65 )
66 GO
67 -- Stored Procedure (SP) Help
68 SP_HELP Establecimiento
69 GO
70 -- Visualizamos los campos creados
71 SELECT * FROM Establecimiento

```

```

71      GO
72  -- Empleado
73  CREATE TABLE Empleado (
74      IdEmpleado INT NOT NULL PRIMARY KEY, -- Llave primaria
75      NombEmpleado VARCHAR(50) NOT NULL,
76      DireccEmpleado VARCHAR(100) NOT NULL,
77      CargoEmpleado VARCHAR(50) NOT NULL,
78      IdEstabl INT NOT NULL REFERENCES Establecimiento
79  )
80  GO
81  -- Stored Procedure (SP) Help
82      SP_HELP Empleado
83  GO
84  -- Visualizamos los campos creados
85      SELECT * FROM Empleado
86  GO
87  -- Cliente
88  CREATE TABLE Cliente (
89      IdCli INT NOT NULL PRIMARY KEY, -- Llave primara
90      NombreCli VARCHAR(50) NOT NULL,
91      TelefonoCli CHAR(9) NOT NULL CHECK (TelefonoCli LIKE '
92      9[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
93      SexoCli CHAR(5) NOT NULL CHECK (SexoCli IN ('Varon', 'Mujer')),
94      FecNacCli DATE NOT NULL
95  )
96  GO
97  -- Stored Procedure (SP) Help
98      SP_HELP Cliente
99  GO
100  -- Visualizamos los campos creados
101      SELECT * FROM Cliente
102  GO
103  -- Reservacion
104  CREATE TABLE Reservacion (
105      IdReserv INT NOT NULL PRIMARY KEY, -- Llave primaria
106      NroMesaReserv INT NOT NULL,
107      FechaReserv DATE NOT NULL DEFAULT GETDATE(),
108      CantPersonReserv INT NOT NULL,
109      IdCli INT NOT NULL REFERENCES Cliente
110  )
111  GO
112  -- Stored Procedure (SP) Help

```

```

112     SP_HELP Reservacion
113     GO
114     -- Visualizamos los campos creados
115     SELECT * FROM Reservacion
116     GO
117     -- Menu
118     CREATE TABLE Menu (
119         IdMenu INT NOT NULL PRIMARY KEY, -- Llave primaria
120         NombreMenu VARCHAR(50) NOT NULL,
121         DescripcionMenu VARCHAR(100) NULL
122     )
123     GO
124     -- Stored Procedure (SP) Help
125     SP_HELP Menu
126     GO
127     -- Visualizamos los campos creados
128     SELECT * FROM Menu
129     GO
130     -- Boleta
131     CREATE TABLE Boleta (
132         IdBoleta CHAR(7) NOT NULL PRIMARY KEY, -- Llave primaria
133         RucBoleta CHAR(11) NOT NULL CHECK(RucBoleta like '20123456784'),
134         NombreCliBoleta VARCHAR(50) NOT NULL,
135         DireccCliBoleta VARCHAR(100) NOT NULL,
136         FechaBoleta DATE NOT NULL,
137         ImpTotBoleta MONEY NOT NULL,
138         IdEmpleado INT NOT NULL REFERENCES Empleado,
139         IdMenu INT NOT NULL REFERENCES Menu,
140         IdCli INT NOT NULL REFERENCES Cliente,
141     )
142     GO
143     -- Stored Procedure (SP) Help
144     SP_HELP Boleta
145     GO
146     -- Visualizamos los campos creados
147     SELECT * FROM Boleta
148     GO
149     -- Detalle Menu - Boleta
150     CREATE TABLE DetalleMenuBoleta (
151         IdMenu INT NOT NULL REFERENCES Menu,
152         IdBoleta CHAR(7) NOT NULL REFERENCES Boleta,
153         PrecioVenta MONEY NOT NULL,

```



```

154     Cantidad INT NOT NULL,
155     PRIMARY KEY (IdMenu, IdBoleta) -- Llave primaria compuesta
156 )
157 GO
158 -- Stored Procedure (SP) Help
159 SP_HELP DetalleMenuBoleta
160 GO
161 -- Visualizamos los campos creados
162 SELECT * FROM DetalleMenuBoleta
163 GO
164 =====
165 -- INGRESAR REGISTROS -----
166 =====
167 -- Establecimiento
168 SELECT * FROM Establecimiento
169 GO
170 INSERT INTO Establecimiento (IdEstabl, TelefEstabl, UbicEstabl, NombreEstabl)
171 VALUES (1, '934128562', 'Jr Manco Capac, Huamanga, Ayacucho, Peru', 'Tullpa')
172 INSERT INTO Establecimiento (IdEstabl, TelefEstabl, UbicEstabl, NombreEstabl)
173 VALUES (2, '944128562', 'Jr Bolivar, Huamanga, Ayacucho, Peru', 'Tullpa')
174 INSERT INTO Establecimiento (IdEstabl, TelefEstabl, UbicEstabl, NombreEstabl)
175 VALUES (3, '954128562', 'Jr Los Ingenieros, Huamanga, Ayacucho, Peru', 'Tullpa
176 ')
177 INSERT INTO Establecimiento (IdEstabl, TelefEstabl, UbicEstabl, NombreEstabl)
178 VALUES (4, '964128562', 'Jr Amauta, Huamanga, Ayacucho, Peru', 'Tullpa')
179 INSERT INTO Establecimiento (IdEstabl, TelefEstabl, UbicEstabl, NombreEstabl)
180 VALUES (5, '974128562', 'Jr La Florida, Huamanga, Ayacucho, Peru', 'Tullpa')
181 GO
182 -- Visualizamos los registros
183 SELECT * FROM Establecimiento
184 GO
185 -- Eliminar el registro con el numero 934128562 de la tabla Establecimiento
186 SELECT ALL * FROM Establecimiento
187 GO
188 BEGIN
189     DELETE Establecimiento
190     WHERE TelefEstabl = '934128562'
191 END
192 GO
193 SELECT ALL * FROM Establecimiento
194 GO
195 -- Insertar nuevamente el registro con el numero 934128562 de la tabla

```

```

Establecimiento
195     SELECT ALL * FROM Establecimiento
196     GO
197     BEGIN
198         INSERT INTO Establecimiento (IdEstabl, TelefEstabl, UbicEstabl,
NombreEstabl)
199         VALUES (1, '934128562', 'Jr Manco Capac, Huamanga, Ayacucho, Peru', '
Tullpa')
200     END
201     GO
202     SELECT ALL * FROM Establecimiento
203     GO
204     -- Modificar del primer registro el campo de la ubiacion del Establecimiento
205     SELECT ALL * FROM Establecimiento
206     GO
207     BEGIN
208         UPDATE Establecimiento
209         SET UbicEstabl = 'Jr Los libertadores modificado'
210         WHERE IdEstabl = 1
211     END
212     SELECT ALL * FROM Establecimiento
213     GO
214     -- Empleado
215     SELECT ALL * FROM Empleado
216     GO
217     INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
IdEstabl)
218     VALUES (1, 'Lucia', 'Jr Hipolito Unanue', 'Chef', 5)
219     INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
IdEstabl)
220     VALUES (2, 'Pedro', 'Jr Amazonas, 234', 'Cocinero', 4)
221     INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
IdEstabl)
222     VALUES (3, 'Alfredo', 'Jr Maria Parado de Bellido', 'Cocinero', 3)
223     INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
IdEstabl)
224     VALUES (4, 'Bertha', 'Jr Los Libertadores', 'Cocinera', 5)
225     INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
IdEstabl)
226     VALUES (5, 'Carla', 'Jr Vista Alegre', 'Recepcionista', 1)
227     INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
IdEstabl)

```

```

228 VALUES (6, 'Adriana', 'Jr Los Libertadores 145', 'Mesera', 2)
229 INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
    IdEstabl)
230 VALUES (7, 'Carolina', 'Jr Manco Capac', 'Mesera', 3)
231 INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
    IdEstabl)
232 VALUES (8, 'Mariela', 'Jr Andres Avelino Caceres, 123', 'Mesera', 4)
233 INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
    IdEstabl)
234 VALUES (9, 'Benjamin', 'Jr Los Alamos', 'Mesero', 1)
235 INSERT INTO Empleado (IdEmpleado, NombEmpleado, DireccEmpleado, CargoEmpleado,
    IdEstabl)
236 VALUES (10, 'Rodrigo', 'Jr Tupac Amaru', 'Mesero', 4)
237 GO
238 -- Visualizamos los registros
239 SELECT ALL * FROM Empleado
240 GO
241 -- Cliente
242 SELECT ALL * FROM Cliente
243 GO
244 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
245 VALUES (1, 'Juan', '913129561', 'Varon', '01/04/1980')
246 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
247 VALUES (2, 'Maria', '933239462', 'Mujer', '04/09/1979')
248 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
249 VALUES (3, 'Pablo', '943349363', 'Varon', '02/07/1992')
250 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
251 VALUES (4, 'Juana', '953459264', 'Mujer', '08/03/2000')
252 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
253 VALUES (5, 'Martha', '963569165', 'Mujer', '11/06/1984')
254 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
255 VALUES (6, 'Julio', '973569161', 'Varon', '09/03/1985')
256 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
257 VALUES (7, 'Luis', '983569161', 'Varon', '07/09/1975')
258 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
259 VALUES (8, 'Carlos', '993569163', 'Varon', '01/05/1997')
260 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
261 VALUES (9, 'Tatiana', '923569164', 'Mujer', '07/07/2004')
262 INSERT INTO Cliente (IdCli, NombreCli, TelefonoCli, SexoCli, FecNacCli)
263 VALUES (10, 'Alejandra', '933569165', 'Mujer', '07/05/1989')
264 GO
265 -- Visualizamos los registros

```

```

266     SELECT ALL * FROM Cliente
267     GO
268 -- Reservacion
269     SELECT ALL * FROM Reservacion
270     GO
271     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
272     , IdCli)
273     VALUES (1, 4, '01/02/2021', 10, 1)
274     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
275     , IdCli)
276     VALUES (2, 6, '01/02/2022', 23, 2)
277     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
278     , IdCli)
279     VALUES (3, 2, '02/02/2022', 12, 3)
280     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
281     , IdCli)
282     VALUES (4, 4, '02/02/2022', 7, 4)
283     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
284     , IdCli)
285     VALUES (5, 3, '03/02/2022', 15, 5)
286     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
287     , IdCli)
288     VALUES (6, 5, '03/03/2022', 5, 6)
289     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
290     , IdCli)
291     VALUES (7, 4, '04/03/2022', 17, 7)
292     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
293     , IdCli)
294     VALUES (8, 2, '03/05/2022', 16, 8)
295     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
296     , IdCli)
297     VALUES (9, 1, '09/05/2022', 11, 9)
298     INSERT INTO Reservacion (IdReserv, NroMesaReserv, FechaReserv, CantPersonReserv
299     , IdCli)
300     VALUES (10,5, '07/06/2022', 6, 10)
301     GO
302 -- Visualizamos los registros
303     SELECT ALL * FROM Reservacion
304     GO
305 -- Menu
306     SELECT ALL * FROM Menu
307     GO

```

```

298 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
299 VALUES (1, 'Tamal', 'Tamal de arroz Chakupe molido hecho tamal, dorado con pato
    asado')
300 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
301 VALUES (2, 'Ceviche a la Brasa', 'Conocido en termino medio en brasas de carbon
    natural')
302 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
303 VALUES (3, 'Causa Nortena', NULL)
304 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
305 VALUES (4, 'Croquetas', 'Croquetas de cachetes y morras de Mero Murique' )
306 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
307 VALUES (5, 'Arroz con pato', 'Arroz con pato clasica del Fiesta')
308 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
309 VALUES (6, 'Pescado', NULL)
310 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
311 VALUES (7, 'Pollo', 'A la brasa con papas fritas mas ensaladas' )
312 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
313 VALUES (8, 'Cerdo', NULL)
314 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
315 VALUES (9, 'Trucha', 'Trucha rosada con papas fritas')
316 INSERT INTO Menu (IdMenu, NombreMenu, DescripcionMenu)
317 VALUES (10, 'Cuy Chactado', NULL)
318 GO
319 -- Visualizamos los registros
320 SELECT ALL * FROM Menu
321 GO
322 -- Boleta
323 SELECT ALL * FROM Boleta
324 GO
325 INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
    FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
326 VALUES ('0000101', '20123456784', 'Lucia', 'Jr Manco Capac', '03/04/2022', 60,
    6, 1,1)
327 INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
    FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
328 VALUES ('0000102', '20123456784', 'Carlos', 'Jr Cusco', '03/04/2022', 145, 7,
    2,2)
329 INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
    FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
330 VALUES ('0000103', '20123456784', 'Korina', 'Jr Los igenieros', '04/05/2022',
    35, 7, 3,3)
331 INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,

```

```

FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000104', '20123456784', 'David', 'Jr Los Alamos', '04/05/2022', 70, 7,
4, 4)
INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000105', '20123456784', 'Alfredo', 'Jr Junin', '05/05/2022', 90, 8,
5, 5)
INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000106', '20123456784', 'Aldair', 'Jr Lima', '05/05/2022', 65, 9,
6, 6)
INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000107', '20123456784', 'Fernando', 'Jr Los Libertadores', '
06/06/2022', 120, 10, 7, 7)
INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000108', '20123456784', 'Mario', 'Jr Atahualpa', '06/06/2022', 45, 6,
8, 8)
INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000109', '20123456784', 'Julia', 'Jr Ayacucho', '08/07/2022', 25, 7,
9, 9)
INSERT INTO Boleta (IdBoleta, RucBoleta, NombreCliBoleta, DireccCliBoleta,
FechaBoleta, ImpTotBoleta, IdEmpleado, IdMenu, IdCli)
VALUES ('0000110', '20123456784', 'Julian', 'Jr Junin', '08/07/2022', 50, 8,
10, 10)
GO
-- Visualizamos los registros
SELECT ALL * FROM Boleta
GO
-- Detalle Menu - Boleta
SELECT * FROM DetalleMenuBoleta
GO
INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
VALUES (1, '0000101', 25, 4)
INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
VALUES (2, '0000102', 99, 1)
INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
VALUES (3, '0000103', 90, 3)
INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
VALUES (4, '0000104', 45, 4)

```

```

360     INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
361     VALUES (5,'0000105', 90, 2)
362     INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
363     VALUES (6,'0000106', 25, 6)
364     INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
365     VALUES (7,'0000107', 25, 2)
366     INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
367     VALUES (8,'0000108', 45, 2)
368     INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
369     VALUES (9,'0000109', 30, 5)
370     INSERT INTO DetalleMenuBoleta (IdMenu, IdBoleta, PrecioVenta, Cantidad)
371     VALUES (10,'0000110', 50, 3)
372     GO
373     -- Visualizamos los registros
374     SELECT ALL * FROM DetalleMenuBoleta
375     GO
376     =====
377     -- STORED PROCEDURE (SP) -----
378     =====
379     -- Procedimiento almacenado que liste todos los campos de la tabla Cliente.
380     -- Listar unicamente los clientes del codigo sexo cliente enviado como parametro.
381     -- PASO 1: Listar todos los campos de la tabla Cliente
382     SELECT ALL * FROM Cliente
383     GO
384     -- PASO 2: Agregar la condicion
385     SELECT ALL * FROM Cliente AS Cli -- Donde "Cli" es el alias
386     WHERE Cli.SexoCli = 'Varon' -- Campo igual a dato
387     GO
388     -- PASO 3: Desarrollo del procedimiento almacenado (sp)
389     CREATE OR ALTER PROCEDURE USP_ListarClientexSexo -- Nombre del sp
390     @SexoCli CHAR(5) -- Parametro o variable
391     AS
392     BEGIN -- Inicio del bloque de codigo
393         SELECT ALL * FROM Cliente AS Cli
394         WHERE Cli.SexoCli = @SexoCli -- Campo igual a dato
395     END -- Fin del bloque de codigo
396     GO
397     -- PASO 4: Llamar al procedimiento almacenado
398     EXECUTE USP_ListarClientexSexo
399     @SexoCli = 'Varon';
400     GO
401     EXEC USP_ListarClientexSexo

```

```

402     @SexoCli = 'Mujer';
403     GO
404     USP_ListarClientexSexo
405     @SexoCli = 'Varon';
406     GO
407     -- Procedimiento almacenado que muestre nombre y descripcion de todos los menus
408     -- ordenado de manera ascendente por nombre del menu.
409     -- Mostrar unicamente los menus cuyo nombre inicie con la letra enviada como
410     -- parametro.
411     -- Asimismo, mostrar unicamente los menus cuya descripcion contenga la letra
412     -- enviada como parametro.
413     -- PASO 1: Listar los campos nombre y descripcion de la tabla Menu
414     SELECT NombreMenu, DescripcionMenu FROM Menu
415     GO
416     -- PASO 2: Agregar la condicion
417     SELECT M.NombreMenu, M.DescripcionMenu
418     FROM Menu AS M -- M es el alias de Menu
419     WHERE M.NombreMenu LIKE 'T%' AND
420     M.DescripcionMenu LIKE '%CON%'
421     ORDER BY M.NombreMenu ASC
422     GO
423     -- PASO 3: Desarrollo del procedimiento almacenado (sp)
424     CREATE OR ALTER PROCEDURE USP_ListarMenuxNombreDescripcion
425     @NombreMenu VARCHAR(50),
426     @DescripcionMenu VARCHAR(100) -- Parametros
427     AS
428     BEGIN
429         SELECT M.NombreMenu, M.DescripcionMenu FROM Menu AS M
430         WHERE M.NombreMenu LIKE @NombreMenu + '%' AND
431         M.DescripcionMenu LIKE '%' + @DescripcionMenu + '%'
432         ORDER BY M.NombreMenu ASC
433     END
434     GO
435     -- PASO 4: Llamar al procedimiento almacenado
436     EXECUTE USP_ListarMenuxNombreDescripcion
437     @NombreMenu = 'T', @DescripcionMenu = 'CON';
438     GO
439     EXEC USP_ListarMenuxNombreDescripcion
440     @NombreMenu = 'C', @DescripcionMenu = 'DE';
441     GO
442     USP_ListarMenuxNombreDescripcion
443     @NombreMenu = 'A', @DescripcionMenu = 'Arroz';

```



```

443      GO
444  =====
445  -- BD, Tablas y campos -----
446  =====
447  -- Visualizamos las bases de datos
448  BEGIN
449      --SELECT ALL * FROM SYS.DATABASES
450      SELECT NAME, DATABASE_ID, CREATE_DATE
451      FROM SYS.DATABASES
452      WHERE DATABASE_ID > 4
453  END
454  GO
455  -- Visualizamos las tablas
456  BEGIN
457      USE Restaurante
458      --SELECT ALL * FROM INFORMATION_SCHEMA.TABLES
459      SELECT TABLE_NAME, TABLE_TYPE
460      FROM INFORMATION_SCHEMA.TABLES
461      WHERE TABLE_TYPE = 'BASE TABLE'
462  END
463  GO
464  -- Visualizamos los campos de las tablas
465  BEGIN
466      USE Restaurante
467      -- Tabla Empleado
468      --SELECT ALL * FROM INFORMATION_SCHEMA.COLUMNS
469      SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
470      FROM INFORMATION_SCHEMA.COLUMNS
471      WHERE TABLE_NAME = 'Empleado'
472      -- Tabla Establecimiento
473      SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
474      FROM INFORMATION_SCHEMA.COLUMNS
475      WHERE TABLE_NAME = 'Establecimiento'
476      -- Tabla Cliente
477      SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
478      FROM INFORMATION_SCHEMA.COLUMNS
479      WHERE TABLE_NAME = 'Cliente'
480      -- Tabla Reservacion
481      SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
482      FROM INFORMATION_SCHEMA.COLUMNS
483      WHERE TABLE_NAME = 'Reservacion'
484      -- Tabla Menu

```

```

485     SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
486     FROM INFORMATION_SCHEMA.COLUMNS
487     WHERE TABLE_NAME = 'Menu'
488
489     -- Tabla Boleta
490     SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
491     FROM INFORMATION_SCHEMA.COLUMNS
492     WHERE TABLE_NAME = 'Boleta'
493
494     -- Tabla DetalleMenuBoleta
495     SELECT ALL TABLE_NAME, COLUMN_NAME, DATA_TYPE
496     FROM INFORMATION_SCHEMA.COLUMNS
497     WHERE TABLE_NAME = 'DetalleMenuBoleta'
498
499     END
500
501     GO
502
503     =====
504
505     -- CONSULTAS -----
506
507     =====
508
509     -- Crear una consulta que muestre todos los nombres
510     -- y cargos del empleado mas el importe total de la boleta
511     -- que se encuentre en un rango determinado.
512     -- Donde, el cargo del empleado sea enviado como parametro.
513
514     SELECT ALL * FROM Empleado
515     SELECT ALL * FROM Boleta
516     SELECT ALL * FROM DetalleMenuBoleta
517
518     GO
519
520     CREATE OR ALTER PROCEDURE USP_ListarEmpleadoBoletaxCargo
521     @CargoEmpleado VARCHAR(50)
522
523     AS
524
525     BEGIN
526
527         SELECT ALL E.NombEmpleado, E.CargoEmpleado, B.ImpTotBoleta
528         FROM Empleado AS E
529         INNER JOIN Boleta AS B
530         ON E.IdEmpleado = B.IdEmpleado
531         WHERE (E.CargoEmpleado = @CargoEmpleado) AND (B.ImpTotBoleta BETWEEN 70 AND
532 145)
533
534         ORDER BY B.ImpTotBoleta ASC
535
536     END
537
538     GO
539
540     EXECUTE USP_ListarEmpleadoBoletaxCargo @CargoEmpleado = 'Mesero';
541
542     GO
543
544     EXEC USP_ListarEmpleadoBoletaxCargo @CargoEmpleado = 'Mesera';
545
546     GO
547
548     USP_ListarEmpleadoBoletaxCargo @CargoEmpleado = 'Mesero';

```

```

526 GO
527 -- Crear una consulta que muestre todos nombres que inicien con el caracter
528 -- establecido y los cargos del empleado limitado hasta cierto caracter
529 -- Ordenado por el nombre de los empleados de manera ascendente.
530 SELECT ALL * FROM Empleado
531 SELECT ALL * FROM Boleta
532 GO
533 CREATE OR ALTER PROCEDURE USP_ListarEmpleadoBoletaxCargo
534 AS
535 BEGIN
536     SELECT ALL E.NombEmpleado, SUBSTRING(E.CargoEmpleado,1,6), B.ImpTotBoleta
537     FROM Empleado AS E
538     LEFT JOIN Boleta AS B
539     ON E.IdEmpleado = B.IdEmpleado
540     WHERE B.IdEmpleado IS NULL AND E.NombEmpleado LIKE 'A%'
541     ORDER BY E.NombEmpleado ASC
542 END
543 GO
544 EXECUTE USP_ListarEmpleadoBoletaxCargo;
545 GO
546 -- Creamos una consulta que muestre todos los campos de las tablas
547 -- establecimiento y empleado. Donde, el cargo del empleado sea enviado
548 -- como parametro, el nombre del establecimiento inicie con T
549 -- o L y ordenado de manera ascendente por cargo del empleado.
550 SELECT * FROM Establecimiento
551 SELECT * FROM Empleado
552 GO
553 CREATE OR ALTER PROCEDURE USP_ListarEmpleadoEstablecimientoxCargo
554 @CargoEmpleado VARCHAR(50)
555 AS
556 BEGIN
557     SELECT Establ.*, Empl.*
558     FROM Establecimiento AS Establ
559     INNER JOIN Empleado AS Empl
560     ON Establ.IdEstabl = Empl.IdEstabl
561     WHERE (Empl.CargoEmpleado = @CargoEmpleado) AND (Establ.NombreEstabl LIKE 'T%'
562     ,
563     OR Establ.NombreEstabl LIKE 'L%')
564     ORDER BY Empl.CargoEmpleado ASC
565 END
566 GO
567 EXECUTE USP_ListarEmpleadoEstablecimientoxCargo @CargoEmpleado = 'Chef';

```

```

567      GO
568      EXEC USP_ListarEmpleadoEstablecimientoxCargo @CargoEmpleado = 'Cocinero';
569      GO
570      USP_ListarEmpleadoEstablecimientoxCargo @CargoEmpleado = 'Mesera';
571      GO
572  =====
573  -- MANEJO DE FECHAS -----
574  =====
575      SELECT GETDATE() AS 'Dia'
576      SELECT DAY(GETDATE()) AS 'Dia'
577      SELECT MONTH(GETDATE()) AS 'Mes'
578      SELECT YEAR(GETDATE()) AS 'Anio'
579  go
580      SELECT DATEPART(YEAR, GETDATE()) AS 'Year';
581      SELECT DATEPART(QUARTER, GETDATE()) AS 'Quarter';
582      SELECT DATEPART(MONTH, GETDATE()) AS 'Month';
583      SELECT DATEPART(DAYOFYEAR, GETDATE()) AS 'DayOfYear';
584      SELECT DATEPART(DAY, GETDATE()) AS 'Day';
585      SELECT DATEPART(WEEK, GETDATE()) AS 'Week';
586      SELECT DATEPART(WEEKDAY, GETDATE()) AS 'WeekDay';
587      SELECT DATEPART(HOUR, GETDATE()) AS 'Hour';
588      SELECT DATEPART(MINUTE, GETDATE()) AS 'Minute';
589      SELECT DATEPART(SECOND, GETDATE()) AS 'Second';
590      SELECT DATEPART(MILLISECOND, GETDATE()) AS 'MilliSecond';
591      SELECT DATEPART(MICROSECOND, GETDATE()) AS 'MicroSecond';
592      SELECT DATEPART(NANOSECOND, GETDATE()) AS 'NanoSecond';
593  go
594      SELECT DATENAME(YEAR, GETDATE()) AS 'Year';
595      SELECT DATENAME(QUARTER, GETDATE()) AS 'Quarter';
596      SELECT DATENAME(MONTH, GETDATE()) AS 'Month';
597      SELECT DATENAME(DAYOFYEAR, GETDATE()) AS 'DayOfYear';
598      SELECT DATENAME(DAY, GETDATE()) AS 'Day';
599      SELECT DATENAME(WEEK, GETDATE()) AS 'Week';
600      SELECT DATENAME(WEEKDAY, GETDATE()) AS 'WeekDay';
601      SELECT DATENAME(HOUR, GETDATE()) AS 'Hour';
602      SELECT DATENAME(MINUTE, GETDATE()) AS 'Minute';
603      SELECT DATENAME(SECOND, GETDATE()) AS 'Second';
604      SELECT DATENAME(MILLISECOND, GETDATE()) AS 'MilliSecond';
605      SELECT DATENAME(MICROSECOND, GETDATE()) AS 'MicroSecond';
606      SELECT DATENAME(NANOSECOND, GETDATE()) AS 'NanoSecond';
607      SELECT DATENAME(ISO_WEEK, GETDATE()) AS 'Week';
608  go

```

Capítulo 4

Diagrama de la base de datos

A continuación se presenta el diagrama de la base de datos generado en SQL Server Management Studio.

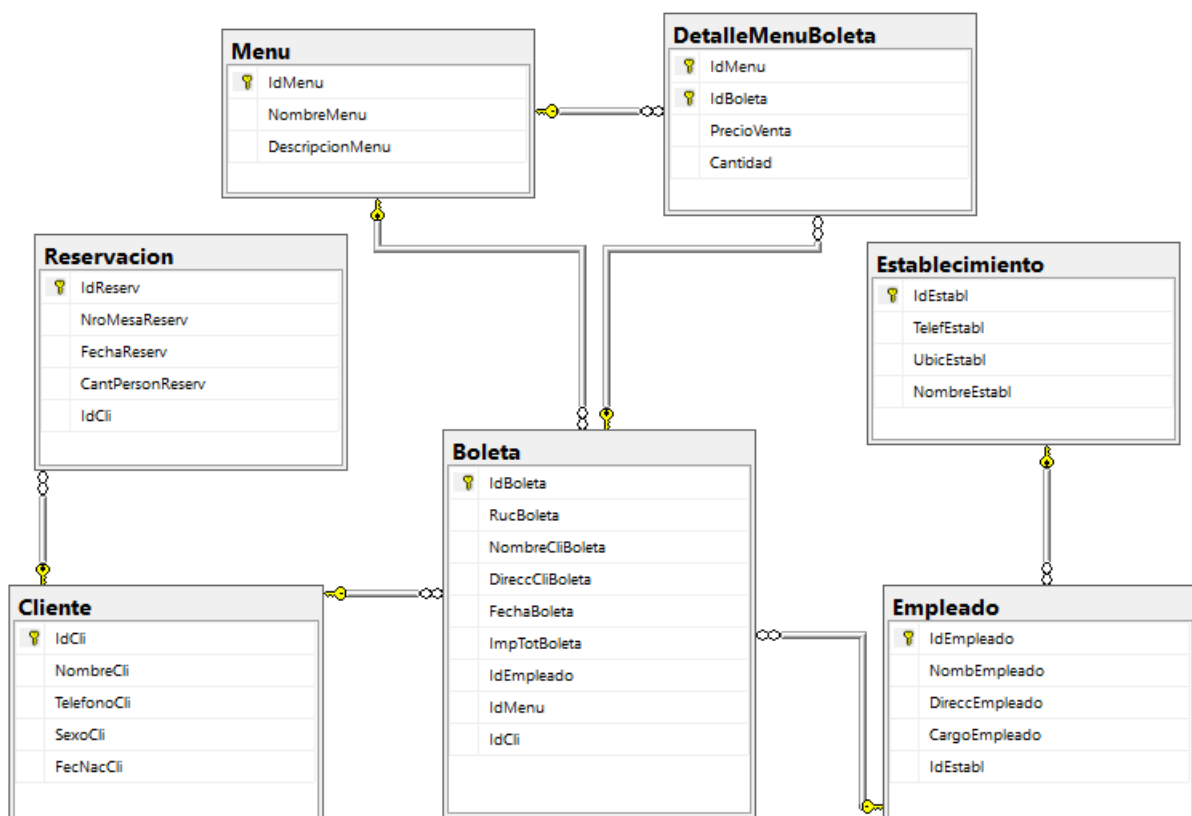


Figura 4.1: Diagrama de la base de datos

La lectura del diagrama de la base de datos es lo siguiente: un cliente puede realizar muchas reservaciones y una reservación puede ser realizado solamente por un cliente; un cliente puede tener muchas boletas y una boleta le pertenece solamente a un cliente; una boleta puede ser emitido por un empleado y un empleado puede emitir muchas boletas; un empleado labora solamente en un establecimiento y en

un establecimiento laboran muchos empleados; una boleta puede registrar muchos menús y un menú se puede registrar en muchas boletas.