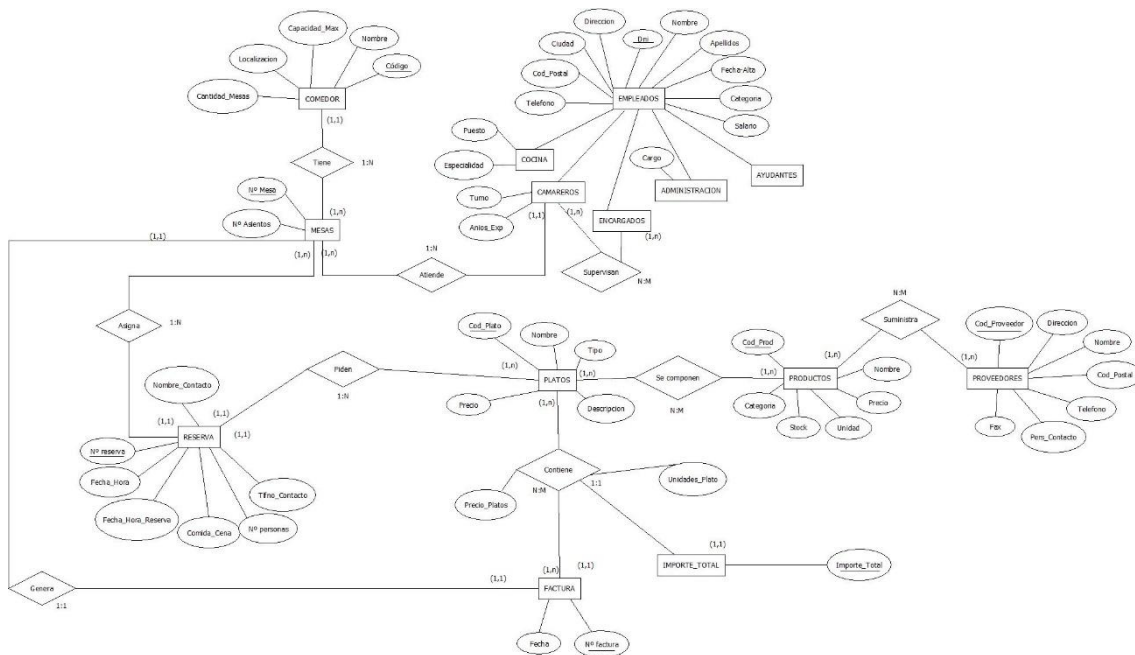


DISEÑO CONCEPTUAL MODELO ENTIDAD/RELACIÓN



DISEÑO LÓGICO DE TABLAS

TABLA EMPLEADO: Recoge la información principal sobre los empleados, recoge los atributos que son común a todos estos, independientemente del cargo que ocupen dentro de la empresa. Se establece como clave principal el DNI de cada empleado, siendo a su vez clave foránea para cada tipo de empleado.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Dni	Dni del empleado	Alfanumérico.	Clave principal. No nulo
Nombre	Nombre del empleado	Alfanumérico.	No nulo.
Apellidos	Apellidos del empleado	Alfanumérico.	No nulo.
Direccion	Dirección del empleado.	Alfanumérico.	No nulo.
Ciudad	Ciudad del empleado.	Alfanumérico.	No nulo.
Cod_Postal	Código postal de residencia del empleado.	Numérico.	No nulo.
Fecha_Alta	Fecha de alta en la empresa del empleado.	Tipo fecha.	No nulo.
Categoría	Categoría profesional del empleado.	Tipo alfanumérico.	No nulo.
Salario	Salario del empleado.	Tipo numérico decimal.	No nulo.

TABLA CAMAREROS: Recoge como clave principal el DNI de empleado, que será su identificador. Además de los atributos de Turno en el que trabaja y los años de experiencia.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Id_Camarero	DNI del camarero.	Alfanumérico.	Clave primaria. Clave foránea: EMPLEADO → Dni
Turno	Turno en el que trabaja el camarero (Comida/Cena)	Alfanumérico.	No nulo.
Años_Exp	Años de experiencia.	Numérico.	No nulo.

TABLA COCINEROS: Recoge como clave principal el DNI de empleado, que será su identificador. Además de los atributos de Puesto en la cocina en el que trabaja y la especialidad.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Id_Cocinero	DNI del cocinero.	Alfanumérico.	Clave primaria. Clave foránea: EMPLEADO → Dni
Puesto	Puesto que ocupa en la cocina.	Alfanumérico.	No nulo.
Especialidad	Cuál es su especialidad (carnes, pescado, postres, etc.)	Alfanumérico.	No nulo.

TABLA ADMINISTRACION: Recoge como clave principal el DNI de empleado, que será su identificador. Además del cargo de administración que ocupa.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Id_Admin	DNI del empleado de administración.	Alfanumérico.	Clave primaria. Clave foránea: EMPLEADO → Dni
Cargo	Cargo que ocupa en la administración del restaurante.	Alfanumérico.	No nulo.

TABLA ENCARGADO: Recoge como clave principal el DNI de empleado, que será su identificador. Además de los atributos de Turno en el que trabaja y los años de experiencia.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Id_Encargado	DNI del encargado de cocinero.	Alfanumérico.	Clave primaria. Clave foránea: EMPLEADO → Dni
Turno	Turno en el que trabaja el camarero (Comida/Cena)	Alfanumérico.	No nulo.
Años_Exp	Años de experiencia.	Numérico.	No nulo.

TABLA ENCARGADO_CAMARERO: Tercera tabla que se genera como consecuencia de la relación entre encargados y camareros, debido a que tiene una relación con cardinalidad N:M (muchos a muchos), ya que un camarero puede ser supervisado por varios encargados y a su vez un encargado supervisa a varios camareros. Se establecen como claves foráneas los identificadores de camarero y encargado.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Id_Encargado	DNI del encargado de camarero.	Alfanumérico.	Clave primaria. Clave foránea: ENCARGADO → Id_Encargado
Id_Camarero	DNI del camarero	Alfanumérico	Clave primaria. Clave foránea: EMPLEADO → Id_Camarero

TABLA AYUDANTES: Recoge como clave principal el DNI de empleado, que será su identificador.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Dni	DNI del empleado.	Alfanumérico.	Clave primaria. Clave foránea: EMPLEADO → Dni

TABLA COMEDOR: Recoge como clave principal el código del comedor, que será su identificador. Además de los atributos del Nombre del comedor, la capacidad máxima de comensales, el número de mesas que contiene y su localización. Sería la entidad padre de la entidad mesa con la que existe una relación.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Codigo_Comedor	Código identificador único del comedor.	Numérico.	Clave primaria.
Nombre	Nombre del comedor.	Alfanumérico.	No nulo.
Capac_Max	Capacidad máxima del comedor (nº de comensales)	Numérico.	No nulo.
Número_mesas	Número de mesas que hay en el comedor.	Numérico.	No nulo.
Localizacion	Localizacion del comedor.	Alfanumérico.	No nulo.

TABLA RESERVA: Recoge como clave principal el número de reserva, que será su identificador. Además de los atributos de la fecha y hora tanto del momento en el que se hace la reserva, como a la hora a que está destinada la reserva. También recoge el atributo tipo, que indica si es comida o cena; el número de personas para las que se reserva y un teléfono de contacto.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Num_Reserva	Número de reserva	Numérico.	Clave primaria.
Fecha_Hora	Fecha y hora en la que se ha hecho la reserva.	Fecha.	No nulo.
Fecha_Hora_Comida	Fecha y hora acordada para comer/cenar.	Fecha.	No nulo.
Tipo	Si es comida o cena.	Alfanumérico.	No nulo.
Num_Personas	Número de personas que van a acudir.	Numérico.	No nulo.
Tlfno_Contacto	Teléfono de contacto de la persona que ha realizado la reserva.	Numérico.	No nulo

TABLA MESAS: Recoge como clave principal el número de mesa, que será su identificador. Además de los atributos de número de asientos que tiene la mesa. Hereda de las entidades COMEDOR y CAMAREROS las claves foráneas de Dni del camarero, código del comedor y número de reserva. Ya que, la entidad MESAS mantiene una relación de 1:N con estas entidades. Debido a que, por una parte, 1 mesa es atendida por un camarero, mientras que un camarero puede atender a varias mesas. Por otra parte, un comedor tiene varias mesas, pero 1 mesa solo puede estar en un comedor. Por otro lado, se asignarían varias mesas a 1 reserva, mientras que 1 mesa solo podría estar asignada a 1 reserva. Por lo que, debido a esta cardinalidad, hereda las claves primarias de ambas entidades.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Numero_Mesa	Número identificador único de mesa.	Numérico.	Clave principal. No nulo.
Numero_Asientos	Número de asientos en cada mesa.	Numérico.	No nulo.
Comedor	Código identificador único del comedor.	Alfanumérico.	Clave foránea: COMEDOR → Codigo_Comedor
Camarero	Dni del camarero que atiende la mesa.	Alfanumérico.	Clave foránea: CAMARERO → Id_Camarero
Num_Reserva	Número de reserva al que se le asigna esta mesa.	Numérico.	Clave foránea: RESERVA→Num_Reserva

TABLA PLATOS: Recoge como clave principal el código del plato, que será su identificador. Además de los atributos del nombre del plato, el tipo de plato, su precio y una breve descripción.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Cod_Plato	Código identificador de plato.	Numérico	Clave principal.
Nombre	Nombre del plato.	Alfanumérico.	No nulo.
Tipo	Tipo de plato (carne, pescado, postre, etc.)	Alfanumérico.	No nulo.
Precio	Precio del plato.	Numérico decimal.	No nulo.
Descripcion	Descripción del plato.	Alfanumérico.	No nulo.

TABLA PRODUCTOS: Recoge como clave principal el código del producto, que será su identificador. Además de los atributos del nombre del producto, la categoría a la que pertenece (carne, pescado, verdura, etc.), el precio del producto, la unidad de medida de este y el stock existente para cada producto.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Cod_Producto	Código identificativo único del producto	Numérico.	Clave primaria.
Nombre	Nombre del producto.	Alfanumérico.	No nulo.
Categoría	Categoría del producto.	Alfanumérico.	No nulo.
Precio	Precio del producto.	Numérico.	No nulo.
Unidad	Unidad de medida del producto (kg, l, docenas, etc.).	Alfanumérico.	No nulo.
Stock	Stock del producto.	Numérico.	No nulo.

TABLA PLATO_PRODUCTOS: Esta tabla surge a partir de la relación SE_COMPONE entre las entidades PLATOS y PRODUCTOS, ya que esta posee una cardinalidad N:M, debido a que un plato está compuesto de varios productos, a la vez que un producto puede estar incluido en varios platos. Es por esto que esta tabla quedaría conformada por las claves foráneas que son las claves primarias de ambas entidades (Código del plato y Código del producto).

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Plato	Código identificativo único del plato	Numérico.	Clave primaria. Clave foránea PLATO → Cod_Plato
Producto	Código identificativo único del producto.	Numérico.	Clave primaria. Clave foránea PRODUCTO → Cod_Producto

TABLA PROVEEDORES: Recoge como clave principal el código del proveedor, que será su identificador. Además de los atributos del nombre de la empresa proveedora, su dirección, Código Postal, Teléfono, Fax y Nombre de la persona de contacto con ese proveedor.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Cod_Proveedor	Código identificativo único del proveedor.	Numérico.	Clave primaria.
Nombre	Nombre del proveedor.	Alfanumérico.	No nulo.
Direccion	Dirección del proveedor.	Alfanumérico.	No nulo.

Cod_Postal	Código postal del proveedor.	Numérico.	No nulo.
Telefono	Teléfono de contacto del proveedor.	Numérico.	No nulo.
Fax	Fax del proveedor.	Numérico.	No nulo.
Persona_Contacto	Nombre de la persona de contacto.	Alfanumérico.	No nulo.

TABLA PROVEEDORES_PRODUCTO: Esta tabla surge a partir de la relación SUMINISTRA entre las entidades PRODUCTOS Y PROVEEDORES, ya que esta posee una cardinalidad N:M, debido a que un proveedor suministraría varios productos, a la vez que un producto puede ser suministrado por varios proveedores. Es por esto que esta tabla quedaría conformada por las claves foráneas que son las claves primarias de ambas entidades (Código identificativo del proveedor y Código del producto).

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Proveedor	Código identificativo único del proveedor.	Numérico.	Clave primaria. Clave foránea PROVEEDORES → Cod_Proveedor.
Producto	Código identificativo único del producto.	Numérico.	Clave primario. Clave foránea PRODUCTO → Cod_Producto.

TABLA FACTURA: Recoge como clave principal el número de factura, que será su identificador. Además de los atributos de la fecha de emisión de la factura y hereda como claves foráneas las claves principales de las entidades MESAS, ya que mantiene una relación de cardinalidad 1:N, debido a que una factura pertenece a una mesa y 1 a mesa solo puede estar asignada una factura. Por lo que la clave principal de cualquiera de las entidades se puede propagar a cualquiera de ellas.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Num_Factura	Número identificativo único de factura.	Numérico.	Clave primaria.
Fecha	Fecha de la factura.	Fecha.	No nulo.
Numero_Mesa	Número identificativo único de la mesa.	Numérico.	Clave foránea. MESAS → Numero_Mesa.
Cod_Plato	Número identificativo único del plato.	Numérico.	Clave primaria. Calve foránea PLATO → Num_Plato.

TABLA FACTURA_CONTIENE_PLATOS: Esta tabla surge por la relación que existe entre las entidades FACTURA y PLATOS, debido a que esta tiene una cardinalidad N:M debido a que una factura puede contener varios platos, a la vez que el mismo plato puede estar en varias facturas una o más veces. Por ello contiene las claves primarias de ambas entidades (Num_Factura y Cod_Plato) además de los atributos que intervienen en esta relación, como son Unidades_Plato y el importe, que es el resultado del producto entre el precio del plato y las unidades del mismo.

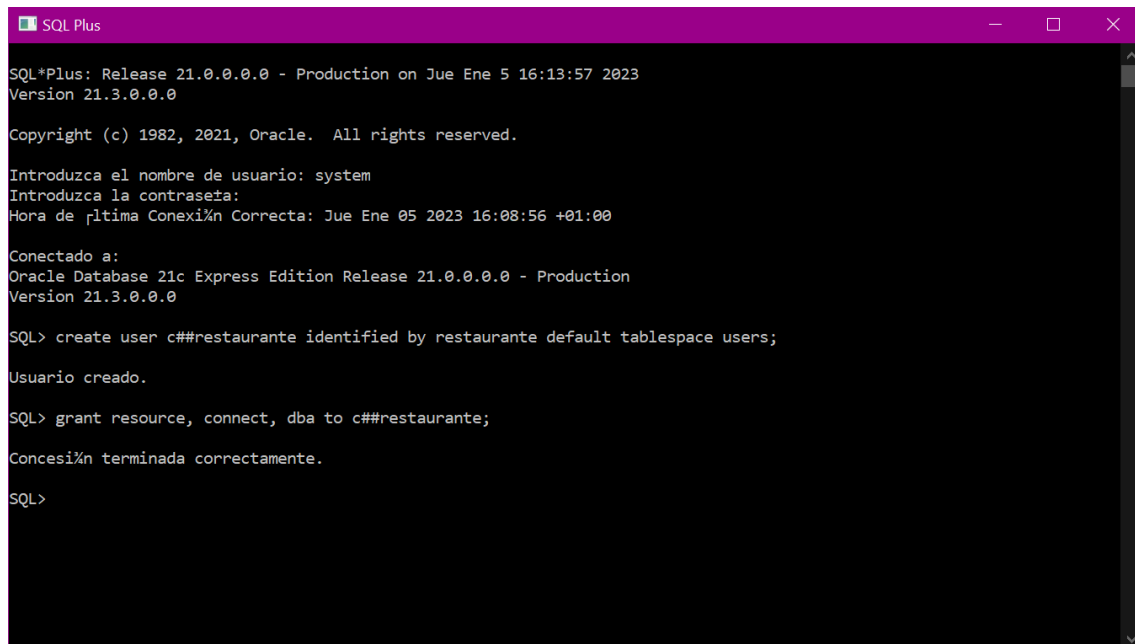
Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Num_Factura	Número identificativo único de factura.	Numérico	Clave primaria. Clave foránea: FACTURA → Num_Factura
Cod_Plato	Código identificativo único de cada plato.	Numérico	Clave primaria. Clave foránea: PLATOS → Cod_Plato
Unidades_Plato	Unidades de plato que contiene la factura.	Numérico	No nulo.
Importe	Unidades del plato por su precio.	Numérico decimal.	No nulo. PLATO.precio * Unidades_Plato

TABLA IMPORTE_TOTAL: Por último, he visto aconsejable incluir una tercera tabla de importe total, que recoja para cada factura la suma de los importes de todos los platos teniendo como clave foránea el número de factura.

Nombre Atributo	Descripción	Tipo de dato	Restricciones (PK, FK, Nulos, Único, etc..)
Num_Factura	Número identificativo único de factura.	Numérico.	Clave primaria. Clave foránea FACTURA → Num_Factura.
Importe_Total	Importe total de todos los platos resultado de la suma de todos los importes.	Numérico decimal.	No nulo.

CODIFICACIÓN DE TABLAS EN SQL DEVELOPER

En primer lugar, creamos el usuario de la base de datos, para ello, en SQL Plus introducimos las siguientes sentencias:



```
SQL*Plus: Release 21.0.0.0.0 - Production on Jue Ene 5 16:13:57 2023
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Introduzca el nombre de usuario: system
Introduzca la contraseña:
Hora de última Conexión Correcta: Jue Ene 05 2023 16:08:56 +01:00

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> create user c##restaurante identified by restaurante default tablespace users;

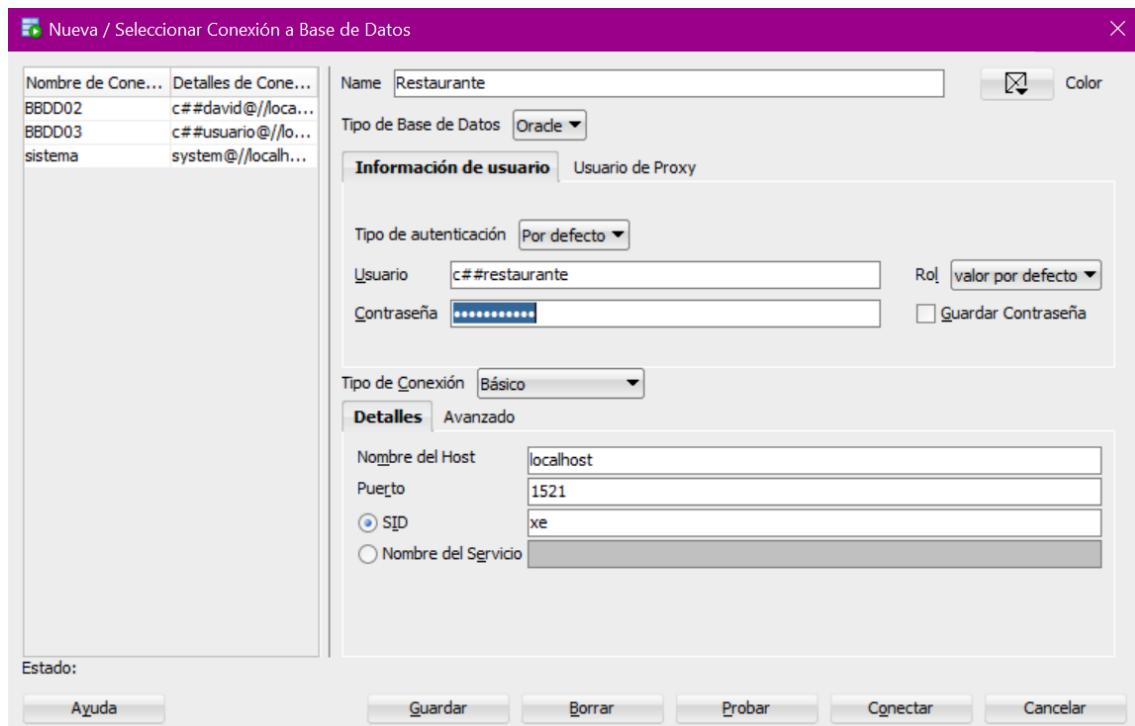
Usuario creado.

SQL> grant resource, connect, dba to c##restaurante;

Concesión terminada correctamente.

SQL>
```

Una vez creado el usuario, conectamos con este en SQL Oracle Developer para crear una Base de datos con nombre Restaurante y comenzamos a crear las tablas.



The screenshot shows the 'Nueva / Seleccionar Conexión a Base de Datos' (New / Select Database Connection) dialog box. The 'Name' field is set to 'Restaurante'. The 'Tipo de Base de Datos' (Database Type) is set to 'Oracle'. Under the 'Información de usuario' (User Information) tab, the 'Tipo de autenticación' (Authentication Type) is 'Por defecto' (Default). The 'Usuario' (User) field is 'c##restaurante' and the 'Rol' (Role) is 'valor por defecto' (default value). The 'Contraseña' (Password) field is masked with dots. There is a checkbox for 'Guardar Contraseña' (Save Password). Under the 'Tipo de Conexión' (Connection Type) dropdown, 'Básico' (Basic) is selected. The 'Detalles' (Details) tab is active, showing 'Nombre del Host' (Host Name) as 'localhost', 'Puerto' (Port) as '1521', and 'SID' as 'xe'. There are buttons for 'Ayuda' (Help), 'Guardar' (Save), 'Borrar' (Delete), 'Probar' (Test), 'Conectar' (Connect), and 'Cancelar' (Cancel).

A continuación se muestran las sentencias introducidas para crear la base de datos:

```
CREATE TABLE EMPLEADO(  
Dni VARCHAR2(9) PRIMARY KEY,
```

Nombre VARCHAR2(20) NOT NULL,
Apellidos VARCHAR2(50) NOT NULL,
Dirección VARCHAR2(100) NOT NULL,
Ciudad VARCHAR2(30) NOT NULL,
Cod_Postal NUMBER(5) NOT NULL,
Fecha_Alta DATE NOT NULL,
Categoria VARCHAR2(15) NOT NULL,
Salario NUMBER (5,2) NOT NULL);

CREATE TABLE **CAMAREROS**(
Id_Camarero VARCHAR2(9) NOT NULL PRIMARY KEY,
Turno VARCHAR2(10) NOT NULL,
Años_Exp NUMBER(2) NOT NULL,
CONSTRAINT FK_ID_CAMARERO FOREIGN KEY (Id_Camarero) REFERENCES EMPLEADO(Dni));

CREATE TABLE **COCINEROS**(
Id_Cocinero VARCHAR2(9) NOT NULL PRIMARY KEY,
Puesto VARCHAR2(15) NOT NULL,
Especialidad VARCHAR2(25) NOT NULL,
CONSTRAINT FK_ID_COCINERO FOREIGN KEY (Id_Cocinero) REFERENCES EMPLEADO(Dni));

CREATE TABLE **ADMINISTRACION**(
Id_Administracion VARCHAR2(9) NOT NULL PRIMARY KEY,
Cargo VARCHAR2(20) NOT NULL,
CONSTRAINT FK_ID_ADMINISTRACION FOREIGN KEY (Id_Administracion) REFERENCES EMPLEADO(Dni));

CREATE TABLE **ENCARGADO**(
Id_Encargado VARCHAR2(9) NOT NULL PRIMARY KEY,
Turno VARCHAR2(10) NOT NULL,
Años_Exp NUMBER(2) NOT NULL,

CONSTRAINT FK_ID_ENCARGADO FOREIGN KEY (Id_Encargado) REFERENCES EMPLEADO(Dni));

CREATE TABLE **AYUDANTES**(

Id_Ayudante VARCHAR2(9) NOT NULL PRIMARY KEY,

CONSTRAINT FK_ID_AYUDANTE FOREIGN KEY (Id_Ayudante) REFERENCES EMPLEADO(Dni));

CREATE TABLE **ENCARGADO_CAMARERO**(

Id_Camarero VARCHAR2(9) NOT NULL,

Id_Encargado VARCHAR2(9) NOT NULL,

CONSTRAINT FK_CAMARERO FOREIGN KEY (Id_Camarero) REFERENCES
CAMAREROS(Id_Camarero),

CONSTRAINT FK_ENCARGADO FOREIGN KEY (Id_Encargado) REFERENCES
ENCARGADO(Id_Encargado));

CREATE TABLE **COMEDOR**(

Cod_Comedor VARCHAR2(3) PRIMARY KEY,

Nombre VARCHAR(15) NOT NULL,

Capac_Max NUMBER(3) NOT NULL,

Numero_Mesas NUMBER(2) NOT NULL,

Localizacion VARCHAR2(10) NOT NULL);

CREATE TABLE **RESERVA**(

Num_Reserva NUMBER(10) PRIMARY KEY,

Fecha_Hora DATE NOT NULL,

Fecha_Hora_Comida DATE NOT NULL,

Tipo VARCHAR2(6) NOT NULL,

Num_Personas NUMBER(3) NOT NULL,

Tlfnno_Contacto NUMBER(9) NOT NULL);

CREATE TABLE **MESAS**(

Id_Mesa NUMBER(3) PRIMARY KEY,

Num_Asientos NUMBER(3) NOT NULL,

Cod_Comedor VARCHAR2(3) NOT NULL,

Id_Camarero VARCHAR2(9) NOT NULL,
Num_Reserva NUMBER(10) NOT NULL,
CONSTRAINT FK_CAMARERO_MESA FOREIGN KEY (Id_Camarero) REFERENCES
CAMAREROS(Id_Camarero),
CONSTRAINT FK_COD_COMEDOR FOREIGN KEY (Cod_Comedor) REFERENCES
COMEDOR(Cod_Comedor),
CONSTRAINT FK_NUM_RESERVA FOREIGN KEY (Num_Reserva) REFERENCES
RESERVA(Num_Reserva));

CREATE TABLE **PLATOS**(
Cod_Plato NUMBER(10) PRIMARY KEY,
Nombre VARCHAR2(30) NOT NULL,
Tipo VARCHAR2(15) NOT NULL,
Precio NUMBER (3,2) NOT NULL,
Descripcion VARCHAR2(100) NOT NULL);

CREATE TABLE **PRODUCTOS**(
Cod_Producto NUMBER(10) PRIMARY KEY,
Nombre VARCHAR2(30) NOT NULL,
Categoria VARCHAR2(15) NOT NULL,
Precio NUMBER(3,2) NOT NULL,
Unidad VARCHAR2(15) NOT NULL,
Stock NUMBER(4) NOT NULL);

CREATE TABLE PLATO_**PRODUCTOS**(
Cod_Plato NUMBER(10) NOT NULL,
Cod_Producto NUMBER(10) NOT NULL,
CONSTRAINT FK_COD_PLATO FOREIGN KEY (Cod_Plato) REFERENCES PLATOS(Cod_Plato),
CONSTRAINT FK_COD_PRODUCTO FOREIGN KEY (Cod_Producto) REFERENCES
PRODUCTOS(Cod_Producto));

CREATE TABLE **PROVEEDORES**(
Cod_Proveedor NUMBER(10) PRIMARY KEY,
Nombre VARCHAR2(20) NOT NULL,
Direccion VARCHAR2(50) NOT NULL,

Cod_Postal NUMBER(5) NOT NULL,
Telefono NUMBER(9) NOT NULL,
Fax NUMBER(9) NOT NULL,
Persona_Contacto VARCHAR2(20) NOT NULL);

CREATE TABLE **PROVEEDORES_PRODUCTO**(
Cod_Proveedor NUMBER(10) NOT NULL,
Cod_Producto NUMBER(10) NOT NULL,
CONSTRAINT FK_COD_PROVEEDOR FOREIGN KEY (Cod_Proveedor) REFERENCES
PROVEEDORES(Cod_Proveedor),
CONSTRAINT FK_COD_PRODUCTO_PROVEEDOR FOREIGN KEY (Cod_Producto) REFERENCES
PRODUCTOS(Cod_Producto));

CREATE TABLE **FACTURA**(
Num_Factura NUMBER(10) PRIMARY KEY,
Fecha_Factura DATE NOT NULL,
Id_Mesa NUMBER(10) NOT NULL,
Cod_Plato NUMBER(10) NOT NULL,
CONSTRAINT FK_NUMERO_MESA FOREIGN KEY (Id_Mesa) REFERENCES MESAS(Id_Mesa),
CONSTRAINT FK_COD_PLATO_FACT FOREIGN KEY (Cod_Plato) REFERENCES
PLATOS(Cod_Plato));

CREATE TABLE **FACTURA_CONTIENE_PLATOS**(
Num_Factura NUMBER (10) NOT NULL,
Cod_Plato NUMBER (10) NOT NULL,
Unidades NUMBER (3) NOT NULL,
Importe NUMBER (3,2) NOT NULL,
CONSTRAINT FK_NUM_FACTURA FOREIGN KEY (Num_Factura) REFERENCES
FACTURA(Num_Factura),
CONSTRAINT FK_PLATO FOREIGN KEY (Cod_Plato) REFERENCES PLATOS(Cod_Plato));

CREATE TABLE **IMPORTE_TOTAL**(
Num_Factura NUMBER(10) NOT NULL,
Importe_Total NUMBER(4,2) NOT NULL,

CONSTRAINT FK_NUM_FACTURA_IMPORTE FOREIGN KEY (Num_Factura) REFERENCES
FACTURA(Num_Factura)
);

EJERCICIO 2. MODIFICACION RESTAURANTE.

Tabla EMPLEADOS

2.1. Añade un índice que facilite búsquedas frecuentes por Apellidos y Nombre sin duplicados.

La sentencia en SQL developer sería:

```
CREATE UNIQUE INDEX IDX_Empleados_Nombre_Apellidos  
ON EMPLEADO (Nombre, Apellidos);
```

The screenshot shows the SQL Developer interface with the 'EMPLEADO' table selected. The 'Índices' (Indexes) tab is active, displaying a list of indexes. A new index, 'C##RESTAURANTE IDX_EMPLEADOS_NOMBRE_APELLIDOS', has been created, highlighted in yellow. It is a unique index on the columns 'NOMBRE' and 'APELLIDOS'.

INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
C##RESTAURANTE	SYS_C008626	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	DNI
C##RESTAURANTE	IDX_EMPLEADOS_NOMBRE_APELLIDOS	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	NOMBRE, APELLIDOS

2.2. El Restaurante se ha inaugurado el día 1 de Junio de 2022. Comprueba que la fecha de alta de los empleados no sea anterior a esa fecha.

Para comprobar que la fecha introducida sea anterior a la fecha de inauguración se crearía una restricción de tipo CHECK a la tabla Empleado para que compruebe que el valor de la columna Fecha_Alta con la siguiente sentencia en SQL:

```
ALTER TABLE EMPLEADO
```

```
ADD CONSTRAINT CK_Fecha_Alta CHECK (Fecha_Alta >= TO_DATE ('01-06-2022', 'DD-MM-YYYY'));
```

The screenshot shows the SQL Developer interface with the 'EMPLEADO' table selected. The 'Restricciones' (Constraints) tab is active, displaying a list of constraints. A new constraint, 'CK_Fecha_Alta', has been created, highlighted in yellow. It is a CHECK constraint on the 'Fecha_Alta' column, ensuring it is greater than or equal to the date '01-06-2022'.

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
CK_Fecha_Alta	Check	Fecha_Alta >= TO_DATE ('01-06-2022', 'DD-MM-YYYY')	(null)	(null)	(null)	(null)	ENABLE
SYS_C008618	Check	"NOMBRE" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLE
SYS_C008619	Check	"APELLIDOS" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLE

Tabla CAMAREROS

2.3. El turno de trabajo sólo puede tomar 3 valores: mañana, tarde y noche. Añade esa restricción teniendo en cuenta que un camarero puede tener más de un turno (no utilices CHECK).

Al tratar de insertar la siguiente sentencia para asignar a la columna turno de la tabla CAMAREROS los valores de mañana, tarde y noche sin utilizar CHECK me da error, ya que no puedo insertar varios valores a una misma columna con el método INSERT INTO TABLA(Columna) VALUES:

```
INSERT INTO CAMAREROS(Turno) VALUES
```

```
('mañana'),
```

```
('tarde'),
```

```
('noche');
```

La única forma de la que he sido capaz de incluir la restricción es utilizando CHECK:

```
ALTER TABLE CAMAREROS
```

```
ADD CONSTRAINT CK_Turno CHECK (Turno IN ('mañana', 'tarde', 'noche'));
```

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL									
Acciones...									
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	CK_TURNO	Check	Turno IN ('mañana', 'tarde', 'noche')	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
2	FK_ID_CAMARERO	Foreign_Key	(null)	C##RESTAURANTE	EMPLEADO	SYS_C008626	NO ACTION	ENABLED	NOT DEFERRABLE
3	SYS_C008627	Check	"ID_CAMARERO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
4	SYS_C008628	Check	"TURNO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
5	SYS_C008629	Check	"AÑOS_EXP" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
6	SYS_C008630	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

Tabla PROVEEDORES

2.4. Añade las columnas Apellidos y Nombre entre el código y la dirección.

SQL no permite una vez creada la tabla añadir en determinadas posiciones otras columnas. Para agregar las columnas Apellidos y Nombre entre ambas columnas intenté la siguiente sentencia, pero me dio error:

```
ALTER TABLE PROVEEDORES
```

```
ADD COLUMN Nombre VARCHAR(20) AFTER Cod_Proveedor,
```

```
ADD COLUMN Apellidos VARCHAR(50) AFTER Nombre;
```

Asique la solución pasaría por borrar la tabla PROVEEDORES y crearla de nuevo, esta vez con estas dos columnas en las posiciones indicadas. A través de la siguientes sentencia:

```
DROP TABLE PROVEEDORES CASCADE CONSTRAINTS PURGE;
```

```
CREATE TABLE PROVEEDORES(
```

```
Cod_Proveedor NUMBER(10) PRIMARY KEY,
```

```
Nombre VARCHAR2(20) NOT NULL,
```

```
Apellidos VARCHAR2(50) NOT NULL,
```

```
Direccion VARCHAR2(50) NOT NULL,
```

```
Cod_Postal NUMBER(5) NOT NULL,
```

```
Telefono NUMBER(9) NOT NULL,
```

Fax NUMBER(9) NOT NULL,

Persona_Contacto VARCHAR2(20) NOT NULL);

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones						
Acciones...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_PROVEEDOR	NUMBER(10,0)	No	(null)	1 (null)	
2	NOMBRE	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3	APELLIDOS	VARCHAR2(50 BYTE)	No	(null)	3 (null)	
4	DIRECCION	VARCHAR2(50 BYTE)	No	(null)	4 (null)	
5	COD_POSTAL	NUMBER(5,0)	No	(null)	5 (null)	
6	TELEFONO	NUMBER(9,0)	No	(null)	6 (null)	
7	FAX	NUMBER(9,0)	No	(null)	7 (null)	
8	PERSONA_CONTACTO	VARCHAR2(20 BYTE)	No	(null)	8 (null)	

Tabla MESAS

2.5. El valor por defecto del número de asientos en las mesas es 4.

Para establecer el número 4 como el valor por defecto de la columna Num_Asientos se utiliza la palabra clave DEFAULT:

```
ALTER TABLE MESAS
```

```
MODIFY Num_Asientos INT DEFAULT 4;
```

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles						
Acciones...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MESA	NUMBER(3,0)	No	(null)	1 (null)	
2	NUM_ASIENTOS	NUMBER(38,0)	No	4	2 (null)	
3	COD_COMEDOR	VARCHAR2(3 BYTE)	No	(null)	3 (null)	
4	ID_CAMARERO	VARCHAR2(9 BYTE)	No	(null)	4 (null)	
5	NUM_RESERVA	NUMBER(10,0)	No	(null)	5 (null)	

Tabla PRODUCTOS


2.6. Añade un índice por Categoría. Muestra todos los índices de la tabla.


La sentencia para crear el índice sería la siguiente:





```
CREATE INDEX IDX_CATEGORIA ON PRODUCTOS (Categoría);
```

La sentencia para mostrar todos los índices de la tabla sería:

```
SELECT INDEX_NAME FROM ALL_INDEXES WHERE TABLE_NAME = 'PRODUCTOS';
```



Salida de Script x


Resultado de la Consulta x





SQL | Todas las Filas Recuperadas:

	INDEX_NAME
1	SYS_C008673
2	IDX_CATEGORIA

2.7. Añade una restricción en la tabla, de forma que el Stock sea entero de 4 cifras, sin signo y que no admita nulos.

En primer lugar, se utiliza esta sentencia para cambiar a tipo de dato de la columna Stock, anteriormente de tipo NUMBER a tipo entero (INT):

```
ALTER TABLE PRODUCTOS
```

```
MODIFY Stock INT;
```

Como no podemos modificar directamente desde la sentencia anterior el número de cifras con INT(4), añadimos una restricción de CHECK para que Stock no admita nulos y tenga como máximo un valor de 5 cifras, es decir que sea mayor o igual que 0 y menor que 10000 (valor de 0 a 9999).

```
ALTER TABLE PRODUCTOS
```

```
ADD CONSTRAINT CHK_STOCK CHECK (Stock IS NOT NULL AND Stock >= 0 AND Stock < 10000);
```

2.8. Borra el índice que acabas de crear.

Para borrar el índice anterior de Categoría utilizamos la siguiente sentencia:

```
DROP INDEX IDX_CATEGORIA;
```

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL										
Acciones...										
	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	C##RESTAURANTE	SYS_C008673	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	COD_PRODUCTO

Tabla PLATOS

2.9. Borra la tabla PROVEEDORES. ¿Qué ocurre? Borra previamente las claves ajenas.

En primer lugar se ha procedido a borrar la clave foránea FK_COD_PROVEEDOR (Clave primaria de la tabla PROVEEDOR) de la tabla PROVEEDORES_PRODUCTO con la sentencia:

```
ALTER TABLE PROVEEDORES_PRODUCTO
```

```
DROP CONSTRAINT FK_COD_PROVEEDOR;
```

Seguidamente se procede a borrar la tabla PROVEEDOR con la sentencia:

```
DROP TABLE PROVEEDORES;
```

BASE DE DATOS Restaurante Villagarcía de Arriba

2.10. Borra la base de datos

Para borrar la base de datos he utilizado la siguiente sentencia:

```
DROP USER c##restaurante CASCADE;
```

Esto eliminará el esquema "restaurante" y todos los objetos que contiene, como tablas, vistas, procedimientos almacenados, etc. La opción "CASCADE" indica que se deben eliminar todas las dependencias del esquema, como claves ajenas que se refieren a tablas del esquema.