



Participantes:

- Evangelista Cacha, Víctor Hugo
- Galindo Davila, Jared Julio (SubDelegado)
- Montalvo Alfajor, Jordy Joseph
- Patricio Arroyo, Ángel Hernán Alberto (Delegado)

Área: Programación para Desarrollo de
Software with Oracle

Bloque: 2202410-PIAD-316-TAL-NRC_27511

Semestre: III

Instructor: Wilman Silver Vásquez Gutiérrez

2024

Nombre de la Organización: Mi Pata Pet Hotel

Breve Reseña de la Organización: Somos profesionales con más 23 años de experiencia en administración de empresas y más de 8 años dedicados al rubro hotelero de canes. Nuestro equipo está conformado sobre todo por Pet Lovers, cuenta con experiencia en cuidado, sociabilización y distracción de canes.

Ubicación: Av. Los Álamos 183, Chilca 15870, Chilca, Perú

Misión: Proporcionar un entorno seguro, cómodo y enriquecedor para que los perros disfruten mientras sus dueños están lejos, ofreciendo servicios de cuidado de alta calidad que promuevan la salud, la felicidad y el bienestar de cada mascota.

Visión: Ser reconocidos como el principal destino de hospedaje para perros, ofreciendo instalaciones modernas, personal altamente calificado y un enfoque personalizado para satisfacer las necesidades individuales de cada mascota.

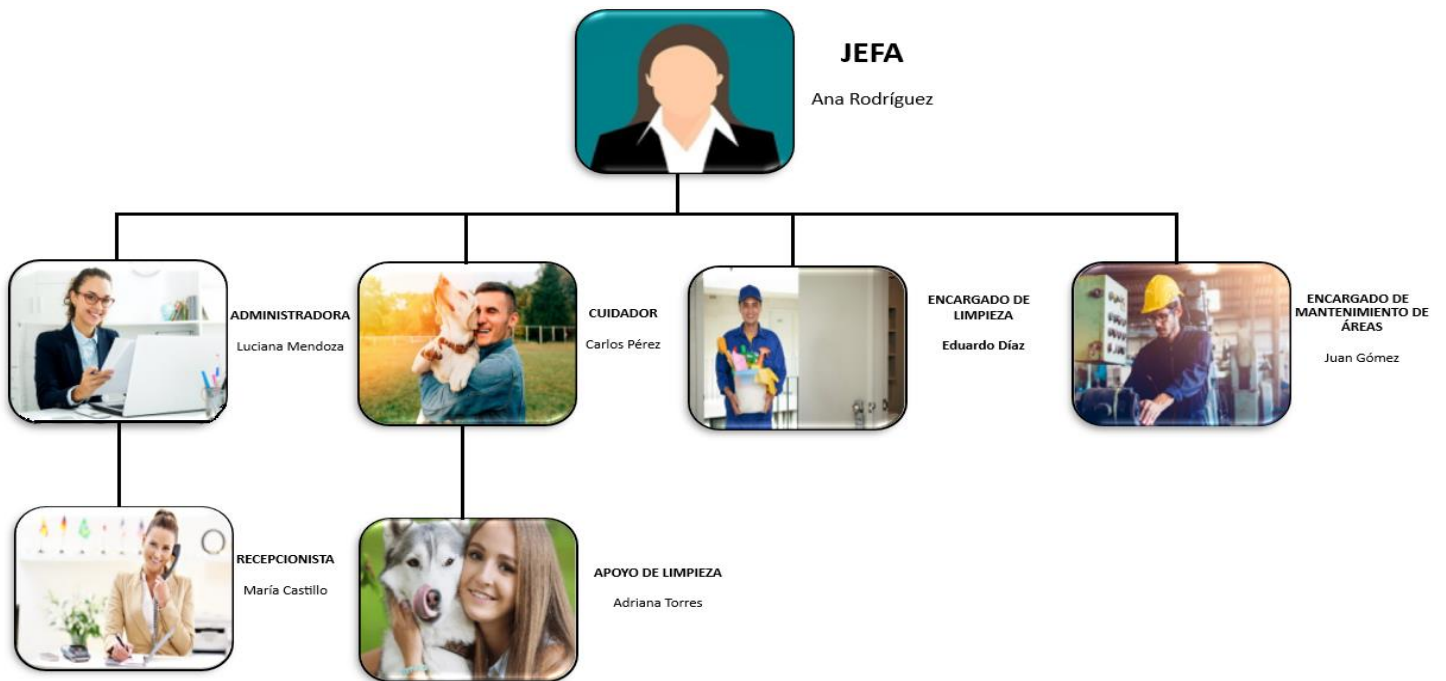
Objetivos:

- * Proporcionar alojamiento seguro y confortable para mascotas de todas las razas y tamaños.
- * Ofrecer servicios de cuidado personalizado, ejercicio adecuado, atención médica, baños, cortes de pelo y entre más.

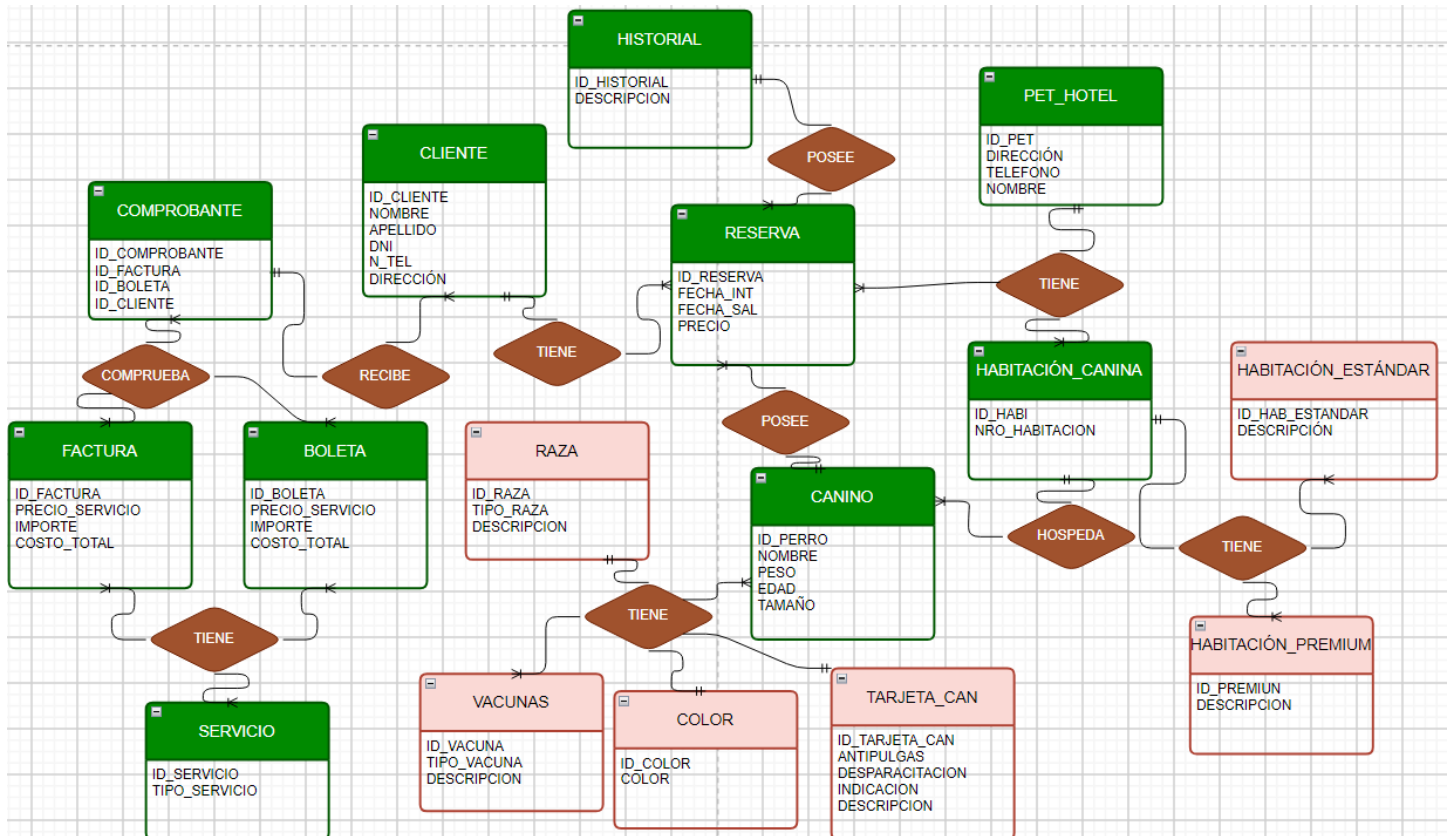
Sistema:

La tabla reserva actúa como Sistema de la Base de datos debido a que en esta se relacionan todas las demás tablas, garantizando la integridad de los datos y facilitando el seguimiento y análisis de las reservas, clientes y mascotas en el Pet Hotel.

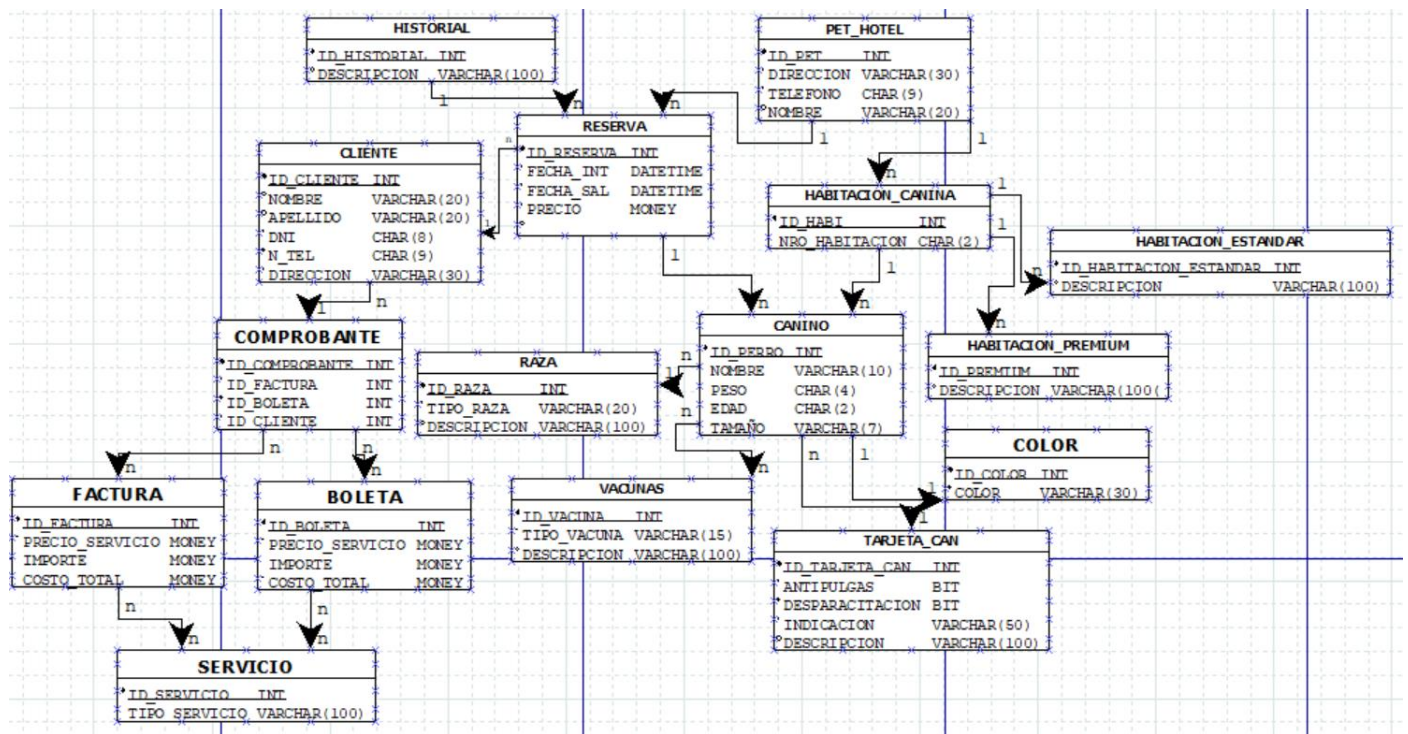
ORGANIGRAMA FUNCIONAL/ÁREAS:



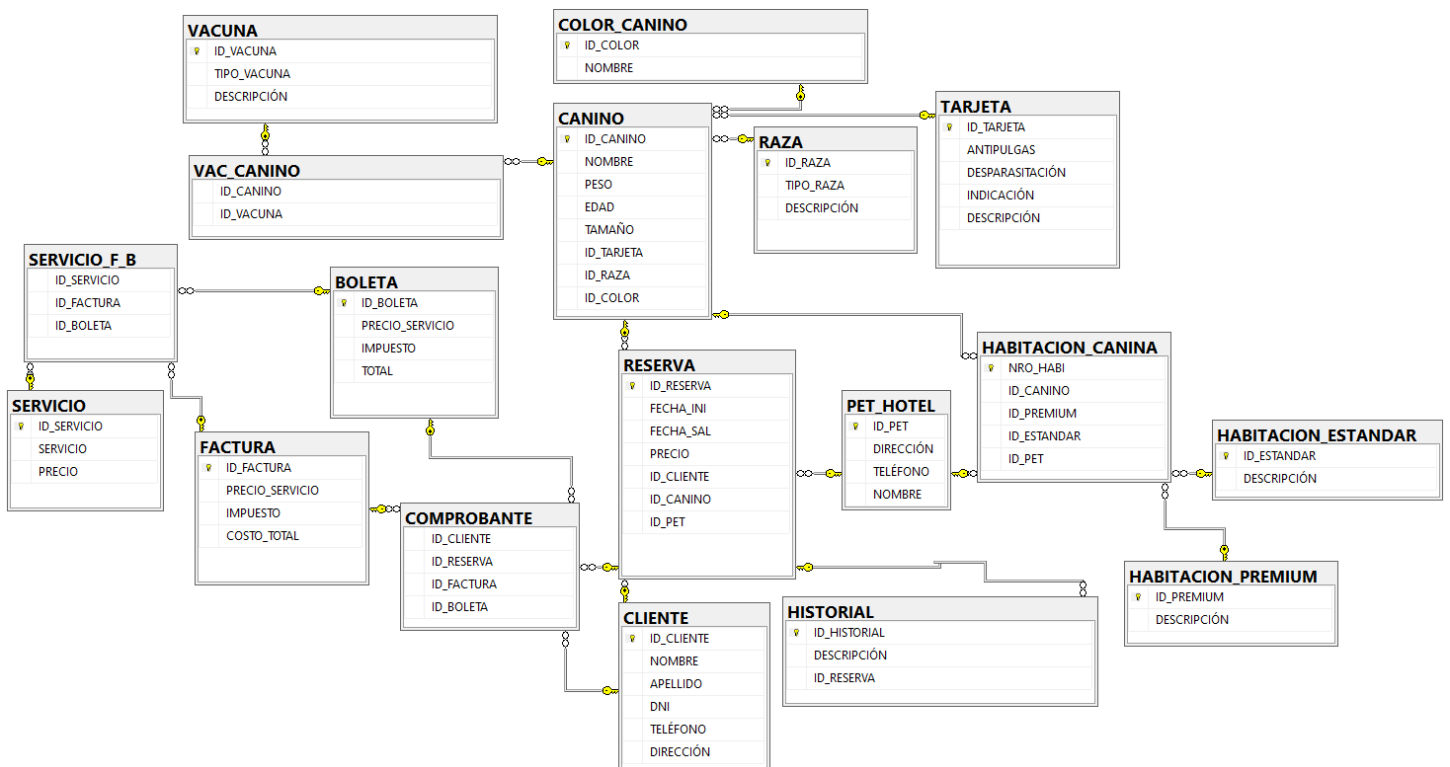
DRAW.IO:



DIA:



SQL SERVER:



CÓDIGO SQL:

--CREAMOS LA BASE DE DATOS:

CREATE DATABASE PET_HOTEL

GO

USE PET_HOTEL

GO

--CREAMOS LAS TABLAS:

CREATE TABLE RAZA(

ID_RAZA CHAR(4) PRIMARY KEY CHECK(ID_RAZA LIKE 'R[0-9][0-9][0-9]'),

TIPO_RAZA VARCHAR(20) NOT NULL UNIQUE,

DESCRIPCIÓN VARCHAR(MAX) NOT NULL

)

GO

CREATE TABLE TARJETA(

ID_TARJETA CHAR(4) PRIMARY KEY CHECK(ID_TARJETA LIKE 'T[0-9][0-9][0-9]'),

ANTIPULGAS BIT NOT NULL CHECK(ANTIPULGAS IN (0, 1)),

DESPARASITACIÓN BIT NOT NULL CHECK(DESPARASITACIÓN IN (0, 1)),

INDICACIÓN VARCHAR(50),

DESCRIPCIÓN VARCHAR(MAX)

)

GO

```
CREATE TABLE VACUNA(  
ID_VACUNA CHAR(4) PRIMARY KEY CHECK(ID_VACUNA LIKE 'V[0-9][0-9][0-9]'),  
TIPO_VACUNA VARCHAR(40) NOT NULL UNIQUE,  
DESCRIPCION VARCHAR(MAX)  
)  
GO
```

```
CREATE TABLE COLOR_CANINO(  
ID_COLOR CHAR(5) PRIMARY KEY CHECK(ID_COLOR LIKE 'CO[0-9][0-9][0-9]'),  
NOMBRE VARCHAR(50) NOT NULL UNIQUE  
)
```

```
CREATE TABLE CANINO(  
ID_CANINO CHAR(5) PRIMARY KEY CHECK(ID_CANINO LIKE 'CA[0-9][0-9][0-9]'),  
NOMBRE VARCHAR(20) NOT NULL,  
PESO INT NOT NULL,  
EDAD INT NOT NULL,  
TAMAÑO CHAR(10) NOT NULL,  
ID_TARJETA CHAR(4) NOT NULL FOREIGN KEY(ID_TARJETA) REFERENCES  
TARJETA(ID_TARJETA),  
ID_RAZA CHAR(4) NOT NULL FOREIGN KEY(ID_RAZA) REFERENCES RAZA(ID_RAZA),  
ID_COLOR CHAR(5) NOT NULL FOREIGN KEY(ID_COLOR) REFERENCES  
COLOR_CANINO(ID_COLOR),  
)  
GO
```

```
CREATE TABLE VAC_CANINO(  
ID_CANINO CHAR(5) NOT NULL FOREIGN KEY(ID_CANINO) REFERENCES  
CANINO(ID_CANINO),  
ID_VACUNA CHAR(4) NOT NULL FOREIGN KEY(ID_VACUNA) REFERENCES  
VACUNA(ID_VACUNA)  
)  
GO
```

```
CREATE TABLE HABITACION_ESTANDAR(  
ID_ESTANDAR CHAR(4) PRIMARY KEY CHECK(ID_ESTANDAR LIKE 'E[0-9][0-9][0-9]'),  
DESCRIPCIÓN VARCHAR(MAX) NULL  
)  
GO
```

```
CREATE TABLE HABITACION_PREMIUM(  
ID_PREMIUM CHAR(5) PRIMARY KEY CHECK(ID_PREMIUM LIKE 'PR[0-9][0-9][0-9]'),  
DESCRIPCIÓN VARCHAR(MAX) NULL  
)  
GO
```

```
CREATE TABLE PET_HOTEL(  
ID_PET CHAR(4) PRIMARY KEY CHECK(ID_PET LIKE 'P[0-9][0-9][0-9]'),  
DIRECCIÓN VARCHAR(40) NOT NULL,  
TELÉFONO CHAR(9) NOT NULL,  
NOMBRE VARCHAR(50) NOT NULL UNIQUE  
)  
GO
```

```
CREATE TABLE HABITACION_CANINA(  
    NRO_HABI CHAR(4) PRIMARY KEY CHECK(NRO_HABI LIKE 'H[0-9][0-9][0-9]'),  
    ID_CANINO CHAR(5) NOT NULL FOREIGN KEY(ID_CANINO) REFERENCES  
        CANINO(ID_CANINO),  
    ID_PREMIUM CHAR(5) NULL FOREIGN KEY(ID_PREMIUM) REFERENCES  
        HABITACION_PREMIUM(ID_PREMIUM),  
    ID_ESTANDAR CHAR(4) NULL FOREIGN KEY(ID_ESTANDAR) REFERENCES  
        HABITACION_ESTANDAR(ID_ESTANDAR),  
    ID_PET CHAR(4) NULL FOREIGN KEY(ID_PET) REFERENCES PET_HOTEL(ID_PET)  
)  
GO
```

```
CREATE TABLE CLIENTE(  
    ID_CLIENTE CHAR(4) PRIMARY KEY CHECK(ID_CLIENTE LIKE 'C[0-9][0-9][0-9]'),  
    NOMBRE VARCHAR(20) NOT NULL,  
    APELLIDO VARCHAR(20) NOT NULL,  
    DNI CHAR(8) NOT NULL UNIQUE,  
    TELÉFONO CHAR(9) NOT NULL UNIQUE,  
    DIRECCIÓN VARCHAR(40) NOT NULL UNIQUE  
)  
GO
```

```
CREATE TABLE SERVICIO(  
    ID_SERVICIO CHAR(4) PRIMARY KEY CHECK(ID_SERVICIO LIKE 'S[0-9][0-9][0-9]'),  
    SERVICIO VARCHAR(MAX) NOT NULL,  
    PRECIO MONEY NOT NULL  
)  
GO
```



```
CREATE TABLE FACTURA(  
ID_FACTURA CHAR(4) PRIMARY KEY CHECK(ID_FACTURA LIKE 'F[0-9][0-9][0-9]'),  
PRECIO_SERVICIO MONEY NOT NULL,  
IMPUESTO MONEY NOT NULL,  
COSTO_TOTAL MONEY NOT NULL  
)  
GO
```

```
CREATE TABLE BOLETA(  
ID_BOLETA CHAR(4) PRIMARY KEY CHECK(ID_BOLETA LIKE 'B[0-9][0-9][0-9]'),  
PRECIO_SERVICIO MONEY NOT NULL,  
IMPUESTO MONEY NOT NULL,  
TOTAL MONEY NOT NULL  
)  
GO
```

```
CREATE TABLE SERVICIO_F_B(  
ID_SERVICIO CHAR(4) NOT NULL FOREIGN KEY(ID_SERVICIO) REFERENCES  
SERVICIO(ID_SERVICIO),  
ID_FACTURA CHAR(4) NULL FOREIGN KEY(ID_FACTURA) REFERENCES  
FACTURA(ID_FACTURA),  
ID_BOLETA CHAR(4) NULL FOREIGN KEY(ID_BOLETA) REFERENCES BOLETA(ID_BOLETA)  
)
```

```
CREATE TABLE RESERVA(  
    ID_RESERVA CHAR(4) PRIMARY KEY CHECK(ID_RESERVA LIKE 'R[0-9][0-9][0-9]'),  
    FECHA_INI DATE NOT NULL DEFAULT(GETDATE()),  
    FECHA_SAL DATE NOT NULL,  
    PRECIO MONEY NOT NULL,  
    ID_CLIENTE CHAR(4) NOT NULL FOREIGN KEY(ID_CLIENTE) REFERENCES  
    CLIENTE(ID_CLIENTE),  
    ID_CANINO CHAR(5) NOT NULL FOREIGN KEY(ID_CANINO) REFERENCES  
    CANINO(ID_CANINO),  
    ID_PET CHAR(4) NOT NULL FOREIGN KEY(ID_PET) REFERENCES PET_HOTEL(ID_PET),  
)  
GO
```

```
CREATE TABLE COMPROBANTE(  
    ID_CLIENTE CHAR(4) NOT NULL FOREIGN KEY(ID_CLIENTE) REFERENCES  
    CLIENTE(ID_CLIENTE),  
    ID_RESERVA CHAR(4) NOT NULL FOREIGN KEY(ID_RESERVA) REFERENCES  
    RESERVA(ID_RESERVA),  
    ID_FACTURA CHAR(4) NULL FOREIGN KEY(ID_FACTURA) REFERENCES  
    FACTURA(ID_FACTURA),  
    ID_BOLETA CHAR(4) NULL FOREIGN KEY(ID_BOLETA) REFERENCES BOLETA(ID_BOLETA)  
)  
GO
```

```
CREATE TABLE HISTORIAL(  
    ID_HISTORIAL CHAR(5) PRIMARY KEY CHECK(ID_HISTORIAL LIKE 'HI[0-9][0-9][0-9]'),  
    DESCRIPCIÓN VARCHAR(MAX),  
    ID_RESERVA CHAR(4) NOT NULL FOREIGN KEY(ID_RESERVA) REFERENCES  
    RESERVA(ID_RESERVA)  
)
```

GO

--INSERTAMOS 5 REGISTROS EN CADA CAMPO DE CADA TABLA:

```
INSERT INTO RAZA VALUES ('R001', 'Labrador Retriever', 'Amigable y extrovertido'),
                           ('R002', 'Pastor Alemán', 'Inteligente y versátil'),
                           ('R003', 'Golden Retriever', 'Amigable y
tolerante'),
                           ('R004', 'Bulldog', 'Dócil y decidido'),
                           ('R005', 'POODLE', 'Inteligente y activo'),
                           ('R006', 'Beagle', 'Alegre y amigable'),
                           ('R007', 'Siberian Husky', 'Energético y sociable'),
                           ('R008', 'Dachshund', 'Valiente y curioso'),
                           ('R009', 'Boxer', 'Fuerte y leal'),
                           ('R010', 'Shih Tzu', 'Afable y juguetón'),
                           ('R011', 'Chihuahua', 'Pequeño pero valiente'),
                           ('R012', 'Doberman', 'Alerta y valiente'),
                           ('R013', 'Cocker Spaniel', 'Afectuoso y
juguetón'),
                           ('R014', 'Great Dane', 'Gentil y amistoso'),
                           ('R015', 'Pomeranian', 'Vivaz y extrovertido'),
                           ('R016', 'Shiba Inu', 'Leal y alerta'),
                           ('R017', 'Bichón Frisé', 'Alegre y juguetón'),
                           ('R018', 'Pug', 'Encantador y tranquilo'),
                           ('R019', 'Husky Siberiano', 'Energético y
amigable'),
                           ('R020', 'Maltese', 'Carismático y cariñoso')
```

GO

```

INSERT INTO TARJETA VALUES ('T001', 0, 1, 'Tratamiento antipulgas y desparasitación
mensual', 'Producto utilizado: XWY'),

('T002', 0, 1, 'Desparasitación mensual',
'Producto utilizado: HCC'),

('T003', 1, 0, 'Tratamiento antipulgas mensual',
'Producto utilizado: DPI'),

('T004', 0, 0, 'Sin tratamiento', 'No se aplicaron
tratamientos este mes'),

('T005', 1, 1, 'Tratamiento antipulgas y
desparasitación quincenal', 'Producto utilizado: LNA'),

('T006', 1, 0, 'Tratamiento antipulgas mensual',
'Producto utilizado: XWZ'),

('T007', 0, 1, 'Desparasitación mensual',
'Producto utilizado: AAE'),

('T008', 1, 1, 'Tratamiento antipulgas y
desparasitación mensual', 'Producto utilizado: DDP'),

('T009', 0, 0, 'Sin tratamiento', 'No se aplicaron
tratamientos este mes'),

('T010', 1, 1, 'Tratamiento antipulgas y
desparasitación quincenal', 'Producto utilizado: GPT'),

('T011', 0, 1, 'Desparasitación mensual',
'Producto utilizado: JPP'),

('T012', 1, 0, 'Tratamiento antipulgas mensual',
'Producto utilizado: MPO'),

('T013', 0, 0, 'Sin tratamiento', 'No se aplicaron
tratamientos este mes'),

('T014', 1, 1, 'Tratamiento antipulgas y
desparasitación quincenal', 'Producto utilizado: PQS'),

('T015', 0, 1, 'Desparasitación mensual',
'Producto utilizado: SYU'),

('T016', 0, 1, 'Desparasitación mensual',
'Producto utilizado: UJW'),

('T017', 1, 0, 'Tratamiento antipulgas mensual',
'Producto utilizado: XAZ'),

```

(T018', 0, 0, 'Sin tratamiento', 'No se aplicaron
tratamientos este mes'),

(T019', 1, 1, 'Tratamiento antipulgas y
desparasitación mensual', 'Producto utilizado: ACB'),

(T020', 0, 1, 'Desparasitación mensual',
'Producto utilizado: FED')

GO

INSERT INTO COLOR_CANINO VALUES('CO001', 'Negro'),

('CO002', 'Blanco'),

('CO003', 'Marrón'),

('CO004', 'Gris'),

('CO005', 'Rubio'),

('CO006', 'Crema'),

('CO007', 'Rojo'),

('CO008', 'Negro y crema'),

('CO009', 'Gris y crema'),

('CO010', 'Negro y fuego'),

('CO011', 'Negro y blanco'),

('CO012', 'Marrón y blanco'),

('CO013', 'Gris y blanco'),

('CO014', 'Negro y marrón'),

('CO015', 'Negro y gris'),

('CO016', 'Blanco y crema'),

('CO017', 'Marrón rojizo'),

('CO018', 'Negro, blanco y crema'),

('CO019', 'Marrón oscuro'),

('CO020', 'Blanco y negro')

GO

```
INSERT INTO CANINO VALUES ('CA001', 'Max', 25, 3, 'Mediano', 'T001', 'R001','CO006'),
                                ('CA002', 'Bella', 15, 2, 'Pequeño', 'T002',
'R001','CO003'),
                                ('CA003', 'Rocky', 30, 4, 'Grande', 'T003',
'R004','CO011'),
                                ('CA004', 'Luna', 18, 1, 'Pequeño', 'T012',
'R003','CO006'),
                                ('CA005', 'Coco', 22, 5, 'Mediano', 'T005',
'R004','CO016'),
                                ('CA006', 'Lucky', 20, 3, 'Mediano', 'T006',
'R006','CO018'),
                                ('CA007', 'Daisy', 12, 2, 'Pequeño', 'T007',
'R008', 'CO014'),
                                ('CA008', 'Charlie', 28, 4, 'Grande', 'T008',
'R002','CO010'),
                                ('CA009', 'Milo', 16, 1, 'Pequeño', 'T009',
'R014','CO011'),
                                ('CA010', 'Ruby', 24, 5, 'Mediano', 'T010',
'R013','CO017'),
                                ('CA011', 'Leo', 18, 3, 'Mediano', 'T011',
'R017','CO012'),
                                ('CA012', 'Sophie', 14, 2, 'Pequeño', 'T012',
'R001','CO001'),
                                ('CA013', 'Maximus', 32, 4, 'Grande', 'T013',
'R019','CO013'),
                                ('CA014', 'Bentley', 19, 1, 'Pequeño', 'T014',
'R014','CO011'),
                                ('CA015', 'Chloe', 26, 5, 'Mediano', 'T015',
'R018','CO008'),
                                ('CA016', 'Rosie', 17, 2, 'Pequeño', 'T012',
'R018','CO008'),
```



```
'R003','CO006'),
('CA017', 'Zeus', 30, 4, 'Grande', 'T017',
'R006','CO018'),
('CA018', 'Mia', 14, 1, 'Pequeño', 'T018',
'R008','CO014'),
('CA019', 'Rocky Jr.', 22, 3, 'Mediano', 'T019',
'R002','CO010')
GO

INSERT INTO VACUNA VALUES ('V001', 'Ehrlichiosis Canina', 'Vacuna contra la ehrlichiosis en
perros'),
('V002', 'Babesiosis Canina', 'Vacuna contra la
babesiosis en perros'),
('V003', 'Leishmaniasis Canina', 'Vacuna contra
la leishmaniasis en perros'),
('V004', 'Anaplasmosis Canina', 'Vacuna contra
la anaplasmosis en perros'),
('V005', 'Borreliosis Canina', 'Vacuna contra la
borreliosis en perros'),
('V006', 'Tos de las Perreras', 'Vacuna contra la
tos de las perreras en perros'),
('V007', 'Dermatitis Atópica', 'Vacuna contra la
dermatitis atópica en perros'),
('V008', 'Tétanos Canino', 'Vacuna contra el
tétanos en perros'),
('V009', 'Hepatitis Canina', 'Vacuna contra la
hepatitis canina'),
('V010', 'Leptospirosis Canina', 'Vacuna contra
la leptospirosis en perros'),
('V011', 'Rabia', 'Vacuna contra la rabia en
perros'),
('V012', 'Moquillo Canino', 'Vacuna contra el
moquillo en perros'),
```

parvovirus en perros'),
('V013', 'Parvovirus Canina', 'Vacuna contra la
influenza canina'),
('V014', 'Influenza Canina', 'Vacuna contra la
el coronavirus entérico en perros'),
('V015', 'Coronavirus Entérico', 'Vacuna contra
enfermedad de Lyme en perros'),
('V016', 'Lyme Canino', 'Vacuna contra la
giardiasis en perros'),
('V017', 'Giardiasis Canina', 'Vacuna contra la
la parainfluenza canina'),
('V018', 'Parainfluenza Canina', 'Vacuna contra
toxoplasmosis en perros'),
('V019', 'Toxoplasmosis', 'Vacuna contra la
contra la babesiosis en perros')
GO

INSERT INTO VAC_CANINO VALUES('CA003','V020'),

('CA004','V003'),

('CA002','V001'),

('CA015','V005'),

('CA020','V003'),

('CA001','V013'),

('CA004','V001'),

('CA003','V001'),

('CA005','V003'),

('CA017','V003'),

('CA012','V004'),

('CA011','V010'),

('CA010','V011'),
('CA002','V020'),
('CA002','V016'),
('CA003','V015'),
('CA004','V007'),
('CA006','V003'),
('CA015','V011'),
('CA020','V009')

GO

INSERT INTO HABITACION_ESTANDAR VALUES('E001', 'Habitación estándar con cama individual y baño compartido'),

('E002', 'Habitación estándar con cama doble y baño privado'),

('E003', 'Habitación estándar con dos camas individuales y vistas al
jardín'),

('E004', 'Habitación estándar con cama queen-size y balcón'),

('E005', 'Habitación estándar con dos camas dobles y desayuno
incluido'),

('E006', 'Habitación estándar con cama king-size y minibar'),

('E007', 'Habitación estándar con cama individual y escritorio'),

('E008', 'Habitación estándar con dos camas individuales y TV de
pantalla plana'),

('E009', 'Habitación estándar con cama doble y sofá-cama'),

('E010', 'Habitación estándar con cama queen-size y acceso a la
piscina'),

('E011', 'Habitación estándar con dos camas dobles y servicio de
habitaciones'),

('E012', 'Habitación estándar con cama king-size y área de estar'),

('E013', 'Habitación estándar con cama individual y vistas
panorámicas'),

('E014', 'Habitación estándar con cama doble y jacuzzi'),
('E015', 'Habitación estándar con dos camas individuales y bañera de hidromasaje'),
('E016', 'Habitación estándar con cama queen-size y gimnasio privado'),
('E017', 'Habitación estándar con dos camas dobles y acceso a la terraza'),
('E018', 'Habitación estándar con cama king-size y desayuno en la cama'),
('E019', 'Habitación estándar con cama individual y servicio de lavandería'),
('E020', 'Habitación estándar con dos camas individuales y cocina americana')
GO

INSERT INTO HABITACION_PREMIUM VALUES('PR001', 'Habitación premium con cama ortopédica y área de juegos para perros pequeños'),
('PR002', 'Habitación premium con piscina privada para perros medianos'),
('PR003', 'Habitación premium con jardín exclusivo para perros grandes'),
('PR004', 'Habitación premium con cama de lujo y ventana panorámica para perros pequeños'),
('PR005', 'Habitación premium con sala de estar y acceso directo al parque para perros medianos'),
('PR006', 'Habitación premium con cama ortopédica y área de juegos para perros grandes'),
('PR007', 'Habitación premium con piscina privada para perros pequeños'),
('PR008', 'Habitación premium con jardín exclusivo para perros medianos'),
('PR009', 'Habitación premium con cama de lujo y ventana panorámica para perros grandes'),
('PR010', 'Habitación

premium con sala de estar y acceso directo al parque para perros pequeños'),
('PR011', 'Habitación
premium con cama ortopédica y área de juegos para perros medianos'),
('PR012', 'Habitación
premium con piscina privada para perros grandes'),
('PR013', 'Habitación
premium con jardín exclusivo para perros pequeños'),
('PR014', 'Habitación
premium con cama de lujo y ventana panorámica para perros medianos'),
('PR015', 'Habitación
premium con sala de estar y acceso directo al parque para perros grandes'),
('PR016', 'Habitación
premium con cama ortopédica y área de juegos para perros pequeños'),
('PR017', 'Habitación
premium con piscina privada para perros medianos'),
('PR018', 'Habitación
premium con jardín exclusivo para perros grandes'),
('PR019', 'Habitación
premium con cama de lujo y ventana panorámica para perros pequeños'),
('PR020', 'Habitación
premium con sala de estar y acceso directo al parque para perros medianos')

GO

INSERT INTO PET_HOTEL VALUES('P001', 'Calle Paracas #23, Lima', '995241652', 'Hotel
Canino Feliz'),
('P002', 'Avenida Mayo #521, Arequipa',
'988231521', 'Pet Paradise'),
('P003', 'Calle Quilca #629, Trujillo',
'998521340', 'Doggy Haven'),
('P004', 'Calle Cusco #521, Chiclayo',
'994251342', 'Paws Palace'),
('P005', 'Calle Junín #983, Cusco',
'904325145', 'Barkington Hotel'),

'987654321', 'Happy Tails Retreat'), ('P006', 'Avenida Amazonas #234, Lima',

'976543210', 'Pampered Paws Resort'), ('P007', 'Calle Ucayali #789, Arequipa',

'955432109', 'Tail Wag Inn'), ('P008', 'Calle Tacna #345, Trujillo',

'966543210', 'Snuggle Pups Lodge'), ('P009', 'Avenida Piura #123, Chiclayo',

'933210987', 'Cozy Canine Cottage'), ('P010', 'Calle Puno #567, Cusco',

'922109876', 'Furry Friends Haven'), ('P011', 'Avenida Huancayo #876, Lima',

'977654321', 'Woofington Palace'), ('P012', 'Calle Ica #456, Arequipa',

Trujillo', '955678901', 'Purr-fect Paws Hotel'), ('P013', 'Calle Lambayeque #765,

Chiclayo', '944321098', 'Pawesome Retreat'), ('P014', 'Avenida Cajamarca #543,

'911234567', 'Fluffy Friends Lodge'), ('P015', 'Calle Huaraz #987, Cusco',

'922345678', 'Whiskers Wonderland'), ('P016', 'Avenida Chinchá #876, Lima',

Arequipa', '977890123', 'Tailored Tails Resort'), ('P017', 'Calle Moquegua #543,

'955678901', 'Cuddly Companions Inn'), ('P018', 'Calle Chimbote #765, Trujillo',

Chiclayo', '944321098', 'Paw-sitively Purrfect Lodge'), ('P019', 'Avenida Tumbes #543,

'911234567', 'Fuzzy Friends Retreat') ('P020', 'Calle Iquitos #987, Cusco',

GO


```
INSERT INTO HABITACION_CANINA VALUES('H001', 'CA001', 'PR001', NULL, 'P004'),
('H002', 'CA002', NULL,
'E002', 'P001'),
('H003', 'CA003', NULL,
'E003', 'P003'),
('H004', 'CA004', 'PR004',
NULL, 'P001'),
('H005', 'CA005', NULL,
'E005', 'P004'),
('H006', 'CA006', 'PR006',
NULL, 'P020'),
('H007', 'CA007', 'PR007',
NULL, 'P015'),
('H008', 'CA008', 'PR008',
NULL, 'P012'),
('H009', 'CA009', NULL,
'E009', 'P018'),
('H010', 'CA010', NULL,
'E010', 'P004'),
('H011', 'CA011', NULL,
'E011', 'P015'),
('H012', 'CA012', NULL,
'E012', 'P013'),
('H013', 'CA013', NULL,
'E013', 'P012'),
('H014', 'CA014', NULL,
'E014', 'P020'),
('H015', 'CA015', 'PR015',
NULL, 'P003'),
('H016', 'CA016', 'PR016',
NULL, 'P003'),
('H017', 'CA017', 'PR017',
NULL, 'P007'),
('H018', 'CA018', NULL,
```

'E018','P008'),

('H019', 'CA019', 'PR019',

NULL,'P019'),

('H020', 'CA020', NULL,

'E020','P016')

GO

INSERT INTO CLIENTE VALUES('C001', 'Juan', 'Pérez', '42565679', '987654321', 'Calle
Ayacucho #324, Lima'),

('C002', 'María', 'Gómez', '87654321',
'965786456', 'Avenida Cajamarca #942, Santa Rosa'),

('C003', 'Carlos', 'Rodríguez', '56789112',
'943567891', 'Calle San José #524, Callao'),

('C004', 'Laura', 'Hernández', '34569890',
'976345243', 'Calle Júpiter #155, Callao'),

('C005', 'Javier', 'López', '90123456',
'945809743', 'Avenida Quilca #421, Callao'),

('C006', 'Ana', 'Martínez', '65432109',
'964324578', 'Calle Huancavelica #210, Lima'),

('C007', 'Miguel', 'Sánchez', '89012395',
'976543211', 'Avenida La Marina #735, San Isidro'),

('C008', 'Elena', 'Ramírez', '43210987',
'968245789', 'Calle Tacna #621, Magdalena'),

('C009', 'Pedro', 'Fernández', '09876543',
'985125852', 'Avenida El Sol #124, Surco'),

('C010', 'Silvia', 'Díaz', '56789012', '901596357',
'Calle Las Flores #512, San Borja'),

('C011', 'Roberto', 'García', '12345678',
'901226848', 'Avenida Los Pinos #789, Miraflores'),

('C012', 'Carolina', 'Moreno', '78901234',
'901238526', 'Calle Los Olivos #345, San Miguel'),

('C013', 'Francisco', 'Peralta', '23456789',
'901234753', 'Avenida Los Robles #987, La Molina'),

('C014', 'Valeria', 'Cordova', '89012345',
'901234548', 'Calle Las Palmeras #654, San Isidro'),

('C015', 'Hugo', 'Navarro', '45678901',
'901234567', 'Avenida Los Laureles #231, Surquillo'),

('C016', 'Lorena', 'Vargas', '67890123',
'951156357', 'Calle Las Orquídeas #432, Barranco'),

('C017', 'Felipe', 'Guerrero', '34567890',
'901753159', 'Avenida Los Alamos #567, La Victoria'),

('C018', 'Daniela', 'Romero', '01234367',
'951369741', 'Calle Los Cerezos #876, Chorrillos'),

('C019', 'Ricardo', 'Salazar', '89014345',
'925486293', 'Avenida Los Pájaros #543, Jesús María'),

('C020', 'Camila', 'Chávez', '23956789',
'986243576', 'Calle Los Flamencos #789, Pueblo Libre')

GO

INSERT INTO SERVICIO VALUES ('S001', 'Paseo', 20.00),

30.00), ('S002', 'Habitaciones individuales',

15.00), ('S003', 'Alimentación personalizada',

('S004', 'Cuidado médico', 40.00),

('S005', 'Actividades recreativas', 25.00),

('S006', 'Piscina para perros', 35.00),

('S007', 'Entrenamiento básico', 30.00),

('S008', 'Salón de belleza', 25.00),

('S009', 'Masajes', 40.00),

20.00), ('S010', 'Áreas de juego al aire libre',

('S011', 'Servicio de transporte', 50.00),

('S012', 'Cámaras web', 20.00),

('S013', 'Entrega de informes diarios',

30.00),

('S014', 'Terapia de comportamiento',

45.00),

('S015', 'Fiestas de cumpleaños', 40.00),

('S016', 'Cuidado nocturno', 50.00),

('S017', 'Recreación acuática', 30.00),

('S018', 'Parque de juegos para perros',

35.00),

('S019', 'Clases de socialización', 25.00),

('S020', 'Servicios de fotografía', 45.00);

GO

INSERT INTO FACTURA VALUES('F001', 15.00, 16.20, 106.20),

('F002', 30.00, 23.40, 153.40),

('F003', 40.00, 21.24, 139.24),

('F004', 85.00, 33.30, 218.30),

('F005', 65.00, 30.60, 200.60),

('F006', 15.00, 24.30, 159.30),

('F007', 50.00, 27.00, 177.00),

('F008', 15.00, 24.30, 154.30),

('F009', 60.00, 21.60, 141.60),

('F010', 15.00, 12.60, 82.60),

('F011', 50.00, 18.00, 118.00),

('F012', 40.00, 28.80, 188.80),

('F013', 20.00, 20.70, 135.70),

('F014', 30.00, 27.00, 177.00),

('F015', 35.00, 17.10, 112.10),

('F016', 40.00, 18.00, 118.00),

('F017', 65.00, 29.34, 192.34),

('F018', 70.00, 34.20, 224.20),
('F019', 30.00, 27.00, 177.00),
('F020', 20.00, 17.64, 115.64)

GO

INSERT INTO BOLETA VALUES('B001', 45.00, 27.00, 177.00),

('B002', 90.00, 34.20, 224.20),
('B003', 85.00, 29.34, 192.34),
('B004', 70.00, 30.60, 200.60),
('B005', 35.00, 18.00, 118.00),
('B006', 35.00, 21.60, 141.60),
('B007', 40.00, 18.00, 118.00),
('B008', 70.00, 26.64, 174.64),
('B009', 150.00, 45.00, 295.00),
('B010', 50.00, 30.60, 200.60),
('B011', 50.00, 18.00, 118.00),
('B012', 40.00, 25.20, 165.20),
('B013', 25.00, 21.60, 141.60),
('B014', 15.00, 20.70, 135.70),
('B015', 35.00, 25.20, 165.20),
('B016', 65.00, 25.20, 165.20),
('B017', 200.00, 54.00, 354.00),
('B018', 85.00, 28.80, 188.80),
('B019', 100.00, 34.20, 224.20),
('B020', 20.00, 21.60, 141.60)

GO

```
INSERT INTO SERVICIO_F_B VALUES('S003', 'F004', NULL),  
  
                                ('S002', NULL, 'B002'),  
  
                                ('S005', NULL, 'B016'),  
  
                                ('S016', 'F004', NULL),  
  
                                ('S020', NULL, 'B001'),  
  
                                ('S003', 'F001', NULL),  
  
                                ('S001', NULL, 'B002'),  
  
                                ('S004', 'F003', NULL),  
  
                                ('S015', NULL, 'B002'),  
  
                                ('S002', NULL, 'B004'),  
  
                                ('S001', NULL, 'B020'),  
  
                                ('S016', 'F005', NULL),  
  
                                ('S001', 'F004', NULL),  
  
                                ('S004', NULL, 'B004'),  
  
                                ('S003', 'F005', NULL),  
  
                                ('S004', NULL, 'B016'),  
  
                                ('S011', 'F020', NULL),  
  
                                ('S020', NULL, 'B003'),  
  
                                ('S004', NULL, 'B003')
```

```
INSERT INTO RESERVA VALUES('R001', GETDATE(), GETDATE() + 7, 60.00, 'C001',  
                             'CA001', 'P007'),  
  
                                ('R002', GETDATE(), GETDATE() + 10, 80.00,  
                             'C002', 'CA002', 'P012'),  
  
                                ('R003', GETDATE(), GETDATE() + 5, 100.00,  
                             'C003', 'CA003', 'P020'),  
  
                                ('R004', GETDATE(), GETDATE() + 14, 50.00,  
                             'C004', 'CA004', 'P016'),
```



```
('R005', GETDATE(), GETDATE() + 8, 75.00,  
'C005', 'CA005', 'P007'),  
  
('R006', GETDATE(), GETDATE() + 12, 90.00,  
'C006', 'CA006', 'P020'),  
  
('R007', GETDATE(), GETDATE() + 6, 110.00,  
'C007', 'CA007', 'P007'),  
  
('R008', GETDATE(), GETDATE() + 9, 120.00,  
'C008', 'CA008', 'P012'),  
  
('R009', GETDATE(), GETDATE() + 15, 70.00,  
'C009', 'CA009', 'P018'),  
  
('R010', GETDATE(), GETDATE() + 11, 85.00,  
'C010', 'CA010', 'P016'),  
  
('R011', GETDATE(), GETDATE() + 7, 95.00,  
'C011', 'CA011', 'P006'),  
  
('R012', GETDATE(), GETDATE() + 14, 110.00, 'C012', 'CA012', 'P003'),  
  
('R013', GETDATE(), GETDATE() + 8, 78.00,  
'C013', 'CA013', 'P007'),  
  
('R014', GETDATE(), GETDATE() + 10, 100.00,  
'C014', 'CA014', 'P001'),  
  
('R015', GETDATE(), GETDATE() + 5, 60.00,  
'C015', 'CA015', 'P003'),  
  
('R016', GETDATE(), GETDATE() + 13, 80.00,  
'C016', 'CA016', 'P020'),  
  
('R017', GETDATE(), GETDATE() + 7, 120.00,  
'C017', 'CA017', 'P004'),  
  
('R018', GETDATE(), GETDATE() + 9, 55.00,  
'C018', 'CA018', 'P014'),  
  
('R019', GETDATE(), GETDATE() + 12, 70.00,  
'C019', 'CA019', 'P017'),  
  
('R020', GETDATE(), GETDATE() + 6, 105.00,  
'C020', 'CA020', 'P007')
```

GO

```
INSERT INTO COMPROBANTE VALUES('C001','R003', NULL, 'B020'),  
                                ('C002','R013', NULL, 'B003'),  
                                ('C003','R020', NULL, 'B001'),  
                                ('C004','R013','F020', NULL),  
                                ('C005','R008', 'F006', NULL),  
                                ('C006','R003','F004', NULL),  
                                ('C007','R005', NULL, 'B016'),  
                                ('C008','R005', NULL, 'B018'),  
                                ('C009','R013','F003', NULL),  
                                ('C010','R003', NULL, 'B002'),  
                                ('C011','R020','F005', NULL),  
                                ('C012','R014', NULL, 'B004'),  
                                ('C013','R003','F002', NULL),  
                                ('C014','R005','F001', NULL),  
                                ('C015','R013', NULL, 'B008'),  
                                ('C016','R006',NULL, 'B019'),  
                                ('C017','R003','F007', NULL),  
                                ('C018','R008','F019', NULL),  
                                ('C019','R020', NULL, 'B015'),  
                                ('C020','R003', NULL, 'B014')
```

GO

```
INSERT INTO HISTORIAL VALUES('HI001', 'Historial médico del perro durante la estancia en el  
hotel.','R003'),  
                                ('HI002', 'Observaciones y tratamientos  
realizados al perro.','R007'),  
                                ('HI003', 'Registro de comportamiento y  
actividades diarias del perro.','R012'),  
                                ('HI004', 'Notas sobre la alimentación y
```

preferencias del perro.', 'R020'),

bienestar del perro en el hotel.', 'R002'),

actividad física del perro.', 'R007'),

cuidados especiales.', 'R003'),

socialización con otros perros.', 'R012'),

descanso del perro.', 'R007'),

sensibilidades.', 'R012'),

juego al aire libre.', 'R002'),

corporal y signos vitales.', 'R003'),

interacción con el personal del hotel.', 'R005'),

o cambios en el entorno.', 'R018'),

comandos y entrenamiento.', 'R019'),

dieta y consumo de agua.', 'R020'),

cuidado personal del perro.', 'R003'),

comportamientos inusuales o preocupantes.', 'R005'),

adaptación del perro al entorno.', 'R006'),

médicos y consultas veterinarias.', 'R010')

('HI005', 'Seguimiento de la salud y

('HI006', 'Informe diario sobre la

('HI007', 'Seguimiento de la medicación y

('HI008', 'Registro de juegos y

('HI009', 'Observaciones sobre el sueño y

('HI010', 'Notas sobre posibles alergias o

('HI011', 'Registro de paseos y tiempo de

('HI012', 'Seguimiento de la temperatura

('HI013', 'Observaciones sobre la

('HI014', 'Registro de eventos especiales

('HI015', 'Notas sobre la respuesta a

('HI016', 'Informe detallado sobre la

('HI017', 'Seguimiento de la higiene y

('HI018', 'Registro de posibles

('HI019', 'Observaciones sobre la

('HI020', 'Seguimiento de eventos

GO

--EJECUTAMOS LAS TABLAS:

SELECT * FROM PET_HOTEL

SELECT * FROM RESERVA

SELECT * FROM HISTORIAL

SELECT * FROM CLIENTE

SELECT * FROM FACTURA

SELECT * FROM CANINO

SELECT * FROM HABITACION_CANINA

SELECT * FROM HABITACION_ESTANDAR

SELECT * FROM HABITACION_PREMIUM

SELECT * FROM TARJETA

SELECT * FROM VACUNA

SELECT * FROM RAZA

--AHORA HAREMOS USO DE LO QUE APRENDIMOS EN CLASE SOBRE CONSULTAS:

--CONSULTAS BÁSICAS:

----CANINO

---QUE MUESTRE LA EDAD DE LOS PERROS QUE TIENEN 3 DE EDAD

```
SELECT EDAD FROM CANINO
```

```
WHERE EDAD LIKE(3)
```

---QUE MUESTRE DEL PERRO LOS DE PESO MAS DE 15

```
SELECT PESO FROM CANINO
```

```
WHERE PESO < 15
```

---QUE MUESTRE QUE PERROS ESTÁN ENTRE LOS TAMAÑOS PEQUEÑO Y MEDIANO

```
SELECT ID_CANINO AS 'NÚMERO PERRO',
```

```
       NOMBRE AS 'NOMBRE PERRO',
```

```
       TAMAÑO AS 'TAMAÑO PERRO'
```

```
FROM CANINO
```

```
WHERE TAMAÑO IN ('Pequeño', 'Mediano')
```

----CLIENTE

--MUESTRA LOS APELLIDOS QUE TIENEN LA Z DE ULTIMO

```
SELECT * FROM CLIENTE
```

```
WHERE APELLIDO LIKE('%Z')
```

--MUESTRA LA DIRRECIION QUE SEAN AVENIDA

SELECT * FROM CLIENTE

WHERE DIRECCIÓN LIKE ('A%')

--MUESTRA LA DIRRECIION QUE SEAN CALLE

SELECT * FROM CLIENTE

WHERE DIRECCIÓN LIKE ('C%')

--MUESTRA LOS NOMBRES QUE TENGAN LA C DE INICIAL

SELECT * FROM CLIENTE

WHERE NOMBRE LIKE('C%')

---PET_HOTEL

--MUESTRA LOS NOMBRES QUE TENGAN DE INICIAL LA P

SELECT NOMBRE FROM PET_HOTEL

WHERE NOMBRE LIKE('P%')

--MUESTRA LOS NÚMEROS TELÉFONICOS QUE COMIENCEN POR 982

SELECT TELÉFONO FROM PET_HOTEL

WHERE TELÉFONO LIKE('982')

---MUESTRA LA DIRRECIION QUE SEAN CALLE

SELECT DIRECCIÓN FROM PET_HOTEL

WHERE DIRECCIÓN LIKE ('C%')

--MUESTRA LA DIRECCION QUE SEAN AVENIDA

SELECT DIRECCIÓN FROM PET_HOTEL

WHERE DIRECCIÓN LIKE ('A%')

--RESERVAS

--MUESTRA EL ID DE LAS RESERVAS CUYO PRECIO ESTÁ ENTRE 30 Y 100

SELECT ID_RESERVA FROM RESERVA

WHERE PRECIO BETWEEN 30 AND 100

--MUESTRA LA DIFERENCIA EN SEMANAS ENTRE LA FECHA DE ENTRADA Y DE SALIDA

SELECT ID_RESERVA AS 'NÚMERO RESERVA',

FECHA_INI AS 'FECHA INICIAL',

FECHA_SAL AS 'FECHA FINAL',

DATEDIFF(WEEK, FECHA_INI, FECHA_SAL) AS 'DIFERENCIA SEMANAS'

FROM RESERVA

--MUESTRA LA DIFERENCIA EN DÍAS ENTRE LA FECHA DE ENTRADA Y DE SALIDA

SELECT ID_RESERVA AS 'NÚMERO RESERVA',

FECHA_INI AS 'FECHA INICIAL',

FECHA_SAL AS 'FECHA FINAL',

DATEDIFF(DAY, FECHA_INI, FECHA_SAL) AS 'DIFERENCIA DÍAS'

FROM RESERVA

--MUESTRA LA DIFERENCIA EN HORAS ENTRE LA FECHA DE ENTRADA Y DE SALIDA

SELECT ID_RESERVA AS 'NÚMERO RESERVA',

FECHA_INI AS 'FECHA INICIAL',

FECHA_SAL AS 'FECHA FINAL',

DATEDIFF(HOUR, FECHA_INI, FECHA_SAL) AS 'DIFERENCIA HORAS'

FROM RESERVA

--CONSULTAS CON OPERACIONES DE UNIÓN:

--INNER JOIN:

/*

1. QUIERO VER QUE PERROS TIENEN HABITACIÓN ESTÁNDAR

Y SUS RESPECTIVOS DATOS DE SUS TARJETAS

*/

SELECT C.NOMBRE,

C.EDAD,

HC.NRO_HABI AS 'NÚMERO HABITACIÓN ESTÁNDAR',

T.ANTIPULGAS,

T.DESPARASITACIÓN,

T.INDICACIÓN

FROM CANINO C

INNER JOIN HABITACION_CANINA HC ON HC.ID_CANINO=C.ID_CANINO

INNER JOIN HABITACION_ESTANDAR HE ON HC.ID_ESTANDAR=HE.ID_ESTANDAR

INNER JOIN TARJETA T ON C.ID_TARJETA=T.ID_TARJETA

WHERE HC.ID_ESTANDAR IS NOT NULL

AND HC.ID_PREMIUM IS NULL

GO

--2. LO MISMO QUE EN EL CASO ANTERIOR, PERO CON LA HABITACIÓN PREMIUM

```
SELECT C.NOMBRE,  
       C.EDAD,  
       HC.NRO_HABI AS 'NÚMERO HABITACIÓN PREMIUM',  
       T.ANTIPULGAS,  
       T.DESPARASITACIÓN,  
       T.INDICACIÓN  
FROM CANINO C  
INNER JOIN HABITACION_CANINA HC ON HC.ID_CANINO=C.ID_CANINO  
INNER JOIN HABITACION_PREMIUM HP ON HC.ID_PREMIUM=HP.ID_PREMIUM  
INNER JOIN TARJETA T ON C.ID_TARJETA=T.ID_TARJETA  
WHERE HC.ID_ESTANDAR IS NULL  
AND HC.ID_PREMIUM IS NOT NULL  
GO
```

/*

3. AHORA REALIZAREMOS UNA CONSULTA QUE BUSQUE MOSTRAR EL NÚMERO DE CLIENTE,
SU NOMBRE, EL DÍA QUE INGRESÓ CON SU MASCOTA(NOMBRE); ADEMÁS DE ESO,
EL DÍA EN EL QUE SE RETIRARÁ

*/

```
SELECT C.ID_CLIENTE AS 'NÚMERO CLIENTE',  
       (C.NOMBRE+SPACE(1)+C.APELLIDO) AS 'NOMBRE Y APELLIDO CLIENTE',  
       CA.NOMBRE AS 'NOMBRE MASCOTA',  
       R.FECHA_INI AS 'FECHA INGRESO',  
       R.FECHA_SAL AS 'FECHA SALIDA'  
FROM CLIENTE C  
INNER JOIN RESERVA R ON R.ID_CLIENTE=C.ID_CLIENTE  
INNER JOIN CANINO CA ON R.ID_CANINO=CA.ID_CANINO
```

GO

/*

4. REALIZAMOS UNA CONSULTA QUE NOS DIGA EN QUE HOTEL
DE PERROS ESTÁ EL CLIENTE Y CUANTO ES EL COSTO QUE ESTE PAGARÁ

*/

```
SELECT P.NOMBRE AS 'NOMBRE DEL HOTEL DE PERROS',  
       P.DIRECCIÓN AS 'DIRECCIÓN DEL HOTEL DE PERROS',  
       (C.NOMBRE+SPACE(1)+C.APELLIDO) AS 'NOMBRE Y APELLIDO CLIENTE',  
       CONCAT('S/. ',R.PRECIO) AS 'PRECIO POR LA RESERVA DE MASCOTA'  
  
FROM PET_HOTEL P  
  
INNER JOIN RESERVA R ON R.ID_PET = P.ID_PET  
  
INNER JOIN CLIENTE C ON R.ID_CLIENTE = C.ID_CLIENTE  
  
ORDER BY R.PRECIO  
  
GO
```

/*

5. AHORA HABLAREMOS DE LOS PERROS VACUNADOS... ¿DE QUÉ TIPO DE RAZA ES?
¿QUÉ TIPOS DE VACUNAS POSEES? Y DEMÁS DATOS DE SU TARJETA

*/

```
SELECT C.NOMBRE AS 'NOMBRE DEL PERRO',  
       R.TIPO_RAZA AS 'RAZA DEL PERRO',  
       R.DESCRIPCIÓN AS 'DESCRIPCIÓN DE LA RAZA',  
       V.TIPO_VACUNA AS 'VACUNA QUE POSEE EL PERRO',  
       V.DESCRIPCIÓN AS 'DESCRIPCIÓN DE LA VACUNA',  
       T.ID_TARJETA AS 'NÚMERO DE TARJETA',  
       T.ANTIPULGAS,  
       T.DESPARASITACIÓN,
```

```
T.INDICACIÓN,  
T.DESCRIPCIÓN AS 'DESCRIPCIÓN DE LA TARJETA'  
  
FROM CANINO C  
  
INNER JOIN RAZA R ON C.ID_RAZA = R.ID_RAZA  
  
INNER JOIN TARJETA T ON T.ID_TARJETA = C.ID_TARJETA  
  
INNER JOIN VAC_CANINO PV ON C.ID_CANINO = PV.ID_CANINO  
  
INNER JOIN VACUNA V ON V.ID_VACUNA = PV.ID_VACUNA  
  
ORDER BY T.ID_TARJETA  
  
GO
```

--LEFT JOIN:

/*

1. MOSTRAREMOS A LOS CLIENTES CON NOMBRES, IDENTIFICADOR, DNI Y TELEFONO CON
SU IDENTIFICADOR

DE RESERVA Y EL PRECIO QUE PAGARÁN DE MENOR A MAYOR

*/

```
SELECT C.ID_CLIENTE,  
       C.NOMBRE AS 'NOMBRE_CLIENTE',  
       C.DNI,  
       C.TELÉFONO,  
       R.ID_RESERVA,  
       R.PRECIO  
  
FROM CLIENTE C  
  
LEFT JOIN RESERVA R ON C.ID_CLIENTE = R.ID_CLIENTE  
  
ORDER BY PRECIO ASC
```

/*

2. MOSTRAREMOS LAS HABITACIONES EL IDENTIFICADOR DEL CANINO Y

EL IDENTIFICADOR DEL PET HOTEL

*/

```
SELECT H.NRO_HABI AS 'HABITACIÓN',  
       H.ID_CANINO,  
       H.ID_PET  
FROM HABITACION_CANINA H  
LEFT JOIN HABITACION_ESTANDAR E ON H.NRO_HABI = E.ID_ESTANDAR  
ORDER BY ID_PET DESC
```

/*

3. MOSTRAREMOS EL NOMBRE DEL PERRO, SU PESO EN FORMA ASCENDENTE
Y TAMBIÉN EL TIPO DE RAZA

*/

```
SELECT C.NOMBRE,  
       C.PESO AS 'PESO KG',  
       R.TIPO_RAZA AS 'TIPO DE RAZA'  
FROM CANINO C  
LEFT JOIN RAZA R ON R.ID_RAZA = C.ID_RAZA  
ORDER BY PESO ASC
```

/*

4. MOSTRAREMOS EL CODIGO DE HABITACIÓN, EL IDENTIFICADOR DEL CANINO Y
DE SU TARJETA

*/

```
SELECT H.NRO_HABI AS 'CODIGO DE HABITACIÓN',  
       C.ID_CANINO,  
       C.ID_TARJETA  
FROM HABITACION_CANINA H
```

```
LEFT JOIN CANINO C ON C.ID_CANINO=H.ID_CANINO
```

```
ORDER BY ID_TARJETA ASC
```

```
/*
```

```
5. MOSTRAMOS TODOS LOS CANINOS CON SU IDENTIFICADOR, NOMBRE, TAMAÑO
```

```
Y SI CUENTA CON SUS ANTIPULGAS Y DESPARASITACIÓN EN SU TARJETA
```

```
*/
```

```
SELECT C.ID_CANINO,
```

```
        C.NOMBRE AS 'NOMBRE DE LA MASCOTA',
```

```
        C.TAMAÑO,
```

```
        T.ANTIPULGAS,
```

```
        T.DESPARASITACIÓN
```

```
FROM CANINO C
```

```
LEFT JOIN TARJETA T ON T.ID_TARJETA=C.ID_TARJETA
```

```
--RIGHT JOIN:
```

```
/*
```

```
1. MOSTRAREMOS QUE VACUNAS TIENE CADA PERRO, PERO TAMBIÉN
```

```
MOSTRAREMOS LAS DEMÁS VACUNAS QUE HAY, PERO NINGÚN PERRO POSEE
```

```
*/
```

```
SELECT C.NOMBRE AS 'NOMBRE PERRO',
```

```
        V.TIPO_VACUNA AS 'VACUNA',
```

```
        V.DESCRIPCIÓN AS 'DESCRIPCIÓN VACUNA'
```

```
FROM CANINO C
```

```
INNER JOIN VAC_CANINO VP ON VP.ID_CANINO = C.ID_CANINO
```

```
RIGHT JOIN VACUNA V ON V.ID_VACUNA = VP.ID_VACUNA
```

```
ORDER BY C.NOMBRE DESC
```

/*

2. MOSTRAREMOS TODAS LAS RAZAS JUNTO CON LA INFORMACIÓN DE LOS PERROS
QUE LAS TIENEN ASIGNADAS, INCLUSO SI NO HAY PERROS ASIGNADOS A ALGUNAS RAZAS

*/

```
SELECT C.NOMBRE AS 'NOMBRE PERRO',  
       R.TIPO_RAZA AS 'RAZA',  
       R.DESCRIPCIÓN AS 'DESCRIPCIÓN RAZA'  
  
FROM CANINO C  
  
RIGHT JOIN RAZA R ON R.ID_RAZA = C.ID_RAZA  
  
ORDER BY C.NOMBRE DESC
```

/*

3. MOSTRAREMOS CADA FACTURA, SERVICIOS, IMPUESTO Y COSTO TOTAL DE CADA
FACTURA QUE TENGA UN CLIENTE CON FACTURA Y LOS DEMÁS CLIENTES AUNQUE NO
TENGAN

FACTURA

*/

```
SELECT (C.NOMBRE+SPACE(1)+C.APELLIDO) AS 'NOMBRE Y APELLIDO',  
       S.SERVICIO,  
       CASE  
           WHEN COSTO_TOTAL IS NULL THEN NULL  
           ELSE CONCAT('S/. ',F.IMPUESTO)  
       END AS 'IGV',  
       CASE  
           WHEN COSTO_TOTAL IS NULL THEN NULL  
           ELSE CONCAT('S/. ',COSTO_TOTAL)  
       END AS 'COSTO TOTAL'
```



```
FROM COMPROBANTE CM
INNER JOIN FACTURA F ON CM.ID_FACTURA = F.ID_FACTURA
INNER JOIN SERVICIO_F_B FB ON FB.ID_FACTURA = F.ID_FACTURA
INNER JOIN SERVICIO S ON S.ID_SERVICIO = FB.ID_SERVICIO
RIGHT JOIN CLIENTE C ON C.ID_CLIENTE = CM.ID_CLIENTE

ORDER BY [COSTO TOTAL] DESC
```

/*

4. MOSTRAREMOS CUANTOS HISTORIALES TIENE CADA RESERVA Y COLOCAREMOS LAS RESERVAS

RESTANTES AUNQUE NO TENGAN HISTORIALES

*/

```
SELECT H.ID_HISTORIAL AS 'NÚMERO HISTORIAL',
       H.DESCRIPCIÓN AS 'DESCRIPCIÓN HISTORIAL',
       R.ID_RESERVA AS 'NÚMERO RESERVA'

FROM HISTORIAL H
RIGHT JOIN RESERVA R ON H.ID_RESERVA = R.ID_RESERVA

ORDER BY H.ID_RESERVA DESC
```

/*

5. MOSTRAREMOS TODOS LOS HOTELES DE PERROS Y SUS RESERVAS AUNQUE HAYAN HOTEL

DE PERROS SIN RESERVAS

*/

```
SELECT P.ID_PET AS 'NÚMERO HOTEL PERROS',
       P.NOMBRE AS 'NOMBRE HOTEL PERROS',
       R.ID_RESERVA AS 'NÚMERO RESERVA'

FROM RESERVA R
```

```
RIGHT JOIN PET_HOTEL P ON R.ID_PET = P.ID_PET
```

ORDER BY R.ID_RESERVA DESC

-- FULL JOIN:

/*

1. REALIZA UNA UNION ENTRE EL ID DE LAS TABLAS RESERVA, CLIENTE, FACTURA Y COMPROBANTE.

*/

SELECT R.ID_RESERVA, C.ID_CLIENTE, F.ID_FACTURA, B.ID_BOLETA

FROM RESERVA R

FULL JOIN COMPROBANTE C ON R.ID_RESERVA = C.ID_RESERVA

FULL JOIN FACTURA F ON R.ID_RESERVA = F.ID_FACTURA

FULL JOIN BOLETA B ON R.ID_RESERVA = B.ID_BOLETA

/*

2. REALIZA UNA UNIÓN DE LAS TABLAS RESERVA Y HISTORIAL

*/

SELECT R.ID_RESERVA, H.ID_HISTORIAL, R.FECHA_INI, R.FECHA_SAL, R.PRECIO

FROM RESERVA R

FULL JOIN HISTORIAL H ON R.ID_RESERVA = H.ID_RESERVA

/*

3. REALIZA UNA UNIÓN ENTRE LA HABITACIÓN CANINA Y SUS 2 TIPOS.

*/

SELECT H.NRO_HABI AS 'NÚMERO HABITACIÓN',

E.ID_ESTANDAR AS 'ESTÁNDAR',

P.ID_PREMIUM AS 'PREMIUM'

FROM HABITACION_CANINA H

```
FULL JOIN HABITACION_ESTANDAR E ON H.ID_ESTANDAR = E.ID_ESTANDAR
```

```
FULL JOIN HABITACION_PREMIUM P ON H.ID_PREMIUM = P.ID_PREMIUM
```

```
/*
```

```
4. REALIZA UNA UNIÓN ENTRE LA TABLA CANINO, RAZA Y COLOR.
```

```
*/
```

```
SELECT C.NOMBRE AS 'NOMBRE PERRO',
```

```
       R.TIPO_RAZA AS 'RAZA',
```

```
       CO.NOMBRE AS 'COLOR'
```

```
FROM CANINO C
```

```
FULL JOIN RAZA R ON R.ID_RAZA = C.ID_RAZA
```

```
FULL JOIN COLOR_CANINO CO ON CO.ID_COLOR = C.ID_COLOR
```

```
/*
```

```
5. VER TODOS LOS NOMBRES DE LOS CANINOS Y TODAS LAS VACUNAS REGISTRADAS.
```

```
*/
```

```
SELECT C.NOMBRE AS 'NOMBRE CANINO',
```

```
       V.TIPO_VACUNA AS 'VACUNA',
```

```
       V.DESCRIPCION AS 'DESCRIPCIÓN VACUNA'
```

```
FROM CANINO C
```

```
FULL JOIN VAC_CANINO PV ON PV.ID_CANINO = C.ID_CANINO
```

```
FULL JOIN VACUNA V ON PV.ID_VACUNA = V.ID_VACUNA
```

```
-----  
--SUBCONSULTAS
```

```
/*
```

```
1. SACAREMOS LOS PRECIOS MAYORES A LA MEDIA DE LA TABLA RESERVA
```

*/

```
SELECT ID_RESERVA AS 'NÚMERO RESESRVA',  
       CONCAT('S/.',STR(PRECIO)) AS 'PRECIO'  
FROM RESERVA WHERE PRECIO > (SELECT AVG(PRECIO) FROM RESERVA)
```

/*

2. MOSTRAREMOS TODOS LOS PERROS CUYO PESO ESTÁ ENTRE 15 Y 24 KILOS

*/

```
SELECT NOMBRE AS 'NOMBRE PERRO',  
       CONCAT(PESO,' KG') AS 'PESO'  
FROM CANINO WHERE PESO IN (SELECT PESO FROM CANINO WHERE PESO BETWEEN 15  
AND 24)
```

/*

3. MOSTRAREMOS TODOS LOS HOTELES DE PERRO CUYO NOMBRE TENGA 'TAIL'

*/

```
SELECT NOMBRE AS 'NOMBRE HOTEL'  
FROM PET_HOTEL WHERE NOMBRE IN (SELECT NOMBRE FROM PET_HOTEL WHERE  
NOMBRE LIKE '%TAIL%')
```

--VARIABLES LOCALES CON CONSULTAS

/*

1. SACARÉ LA CANTIDAD DE PERROS EN LA TABLA CANINO

*/

```
DECLARE @CANT_PERRO INT  
SELECT @CANT_PERRO=(SELECT COUNT(NOMBRE) FROM CANINO)  
PRINT CONCAT('LA CANTIDAD DE PERROS ES: ', @CANT_PERRO)
```

GO

/*

2. SACARÉ EL NOMBRE, APELLIDO, DNI, TELÉFONO Y DIRECCIÓN DEL CLIENTE C005

*/

DECLARE @NOMBRE_APE VARCHAR(40), @DNI CHAR(8), @TELÉFONO CHAR(9),
@DIRECCIÓN VARCHAR(50)

SELECT @NOMBRE_APE = (SELECT (NOMBRE + SPACE(1) + APELLIDO) FROM CLIENTE WHERE
ID_CLIENTE = 'C005')

SELECT @DNI = (SELECT DNI FROM CLIENTE WHERE ID_CLIENTE = 'C005')

SELECT @TELÉFONO = (SELECT TELÉFONO FROM CLIENTE WHERE ID_CLIENTE = 'C005')

SELECT @DIRECCIÓN = (SELECT DIRECCIÓN FROM CLIENTE WHERE ID_CLIENTE = 'C005')

PRINT 'NOMBRE Y APELLIDO: ' + @NOMBRE_APE

PRINT 'DNI: ' + @DNI

PRINT 'TELÉFONO: ' + @TELÉFONO

PRINT 'DIRECCIÓN: ' + @DIRECCIÓN

GO

/*

3. SACARÉ LA SUMA DE TODOS LOS COSTOS DE LA TABLA FACTURA

*/

DECLARE @SUMA_COSTO MONEY

SELECT @SUMA_COSTO = (SELECT SUM(COSTO_TOTAL) FROM FACTURA)

PRINT CONCAT('LA SUMA DE TODOS LOS COSTOS DE LA TABLA FACTURA ES: S/.
' ,@SUMA_COSTO)

GO

--FUNCIÓN ESCALAR

```
/*
```

```
1. EN ESTA FUNCIÓN ESCALAR MOSTRARÉ LA CANTIDAD DE HABITACIÓN PREMIUM HAY  
REGISTRADAS EN LA TABLA HABITACIÓN CANINA
```

```
*/
```

```
CREATE FUNCTION DBO.CONTADOR_PR() RETURNS INT
```

```
AS
```

```
    BEGIN
```

```
        DECLARE @CONTADOR_PR INT
```

```
        SELECT @CONTADOR_PR = COUNT(ID_PREMIUM) FROM  
HABITACION_CANINA WHERE ID_CANINO IS NOT NULL
```

```
        RETURN @CONTADOR_PR
```

```
    END
```

```
GO
```

```
PRINT CONCAT('LA CANTIDAD DE HABITACIÓN PREMIUM SON: ', DBO.CONTADOR_PR(), ' DE  
20')
```

```
GO
```

```
/*
```

```
2. EN ESTA FUNCIÓN MOSTRARÉ TODOS LOS DATOS DEL CANINO CUYO IDENTIFICADOR ES  
CA010
```

```
*/
```

```
CREATE FUNCTION DBO.DATOS_PERRO() RETURNS VARCHAR(MAX)
```

```
AS
```

```
    BEGIN
```

```
        DECLARE @DATOS VARCHAR(MAX), @EXTRA VARCHAR(30)
```

```
        SELECT @EXTRA = UPPER(NOMBRE) FROM CANINO WHERE ID_CANINO =  
'CA010'
```

```
        SET @DATOS = @EXTRA
```

```
        SELECT @EXTRA = STR(PESO) FROM CANINO WHERE ID_CANINO = 'CA010'
```

```

        SET @DATOS = CONCAT(@DATOS, ', UN PESO DE ', LTRIM(@EXTRA))

        SELECT @EXTRA = UPPER(TAMAÑO) FROM CANINO WHERE ID_CANINO =
'CA010'

        SET @DATOS = CONCAT(@DATOS, 'KG Y UN TAMAÑO ', @EXTRA, '.')

        RETURN @DATOS

    END

GO

PRINT 'EL PERRO CON IDENTIFICACIÓN CA010 TIENE DE NOMBRE ' + DBO.DATOS_PERRO()

GO

```

/*

3.-SACAMOS LA SUMA DE COSTO TOTAL DE LA TABLA FACTURA

*/

```
CREATE FUNCTION DBO.COSTO_SUM() RETURNS MONEY
```

```
AS
```

```
    BEGIN
```

```
        DECLARE @SUM MONEY
```

```
        SELECT @SUM = SUM(costo_total) FROM FACTURA
```

```
        RETURN LTRIM(@SUM)
```

```
    END
```

```
GO
```

```
PRINT CONCAT('Su costo total es: S/. ', DBO.COSTO_SUM())
```

```
GO
```

--FUNCIÓN DE TABLAS

/*

1. EN ESTA CONSULTA, UNÍ LA TABLA HISTORIAL CON LA TABLA RESERVA Y, DESPUÉS DE ESO, HICE QUE COMPARE

ENTRE LOS REGISTROS DE DESCRIPCIÓN LOS QUE SON 'SEGUIMIENTO DE LA MEDICACIÓN Y CUIDADOS ESPECIALES'

*/

CREATE FUNCTION T_HISTORIAL()

RETURNS TABLE

AS

RETURN (SELECT H.ID_HISTORIAL CODIGO,

H.DESCRIPCIÓN DESCRIPCION,

R.PRECIO,

R.FECHA_INI,

R.FECHA_SAL

FROM RESERVA R JOIN HISTORIAL H ON R.ID_RESERVA = H.ID_RESERVA)

GO

SELECT * FROM T_HISTORIAL() WHERE DESCRIPCION = 'Seguimiento de la medicación y cuidados especiales.'

GO

/*

2. HAREMOS UNA CONSULTA QUE ME DEVUELVA LOS IDENTIFICADORES DE LAS VACUNAS

JUNTO AL TIPO DE VACUNA, HACIENDO USO UNA FUNCIÓN DE TABLAS

*/

CREATE FUNCTION PV_VACUNA()

RETURNS TABLE

AS

RETURN (

SELECT V.ID_VACUNA AS CODIGO,

V.TIPO_VACUNA AS TIPO_DE_VACUNA

FROM VAC_CANINO P

INNER JOIN VACUNA V ON P.ID_VACUNA = V.ID_VACUNA


```
)
```

```
GO
```

```
SELECT * FROM PV_VACUNA() WHERE TIPO_DE_VACUNA='Leishmaniasis Canina'
```

```
GO
```

```
/*
```

3. MOSTRAREMOS UNA CONSULTA QUE ME DEVUELVA LOS SIGUIENTES CAMPOS:
IDENTIFICADORES DE LAS RAZAS, TIPO DE RAZA,

NOMBRE Y EL PESO, HACIENDO USO UNA FUNCIÓN DE TABLAS

```
*/
```

```
CREATE FUNCTION R_CANINO()
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN (
```

```
    SELECT R.TIPO_RAZA AS TIPO_DE_RAZA,
```

```
           C.NOMBRE,
```

```
           C.PESO AS PESO_KG
```

```
    FROM CANINO C
```

```
    INNER JOIN RAZA R ON C.ID_RAZA = R.ID_RAZA
```

```
)
```

```
GO
```

```
SELECT * FROM R_CANINO() WHERE TIPO_DE_RAZA='PASTOR ALEMÁN'
```

```
GO
```

```
/*
```

4. UNÍ LA TABLA CLIENTE CON LA TABLA FACTURA E HICE QUE BUSQUE EN EL CAMPO
SERVICIO O NOMBRE_SERVICIO SI TIENE 'Cuidado médico'

```
*/
```

```

CREATE FUNCTION T_CLIENTE()
RETURNS TABLE
AS
    RETURN (SELECT F.ID_FACTURA CÓDIGO,
                S.SERVICIO NOMBRE_SERVICIO,
                CONCAT(C.NOMBRE, SPACE(1), C.APELLIDO) AS 'NOMBRE
APELLIDO CLIENTE',
                C.DNI,
                C.DIRECCIÓN
            FROM CLIENTE C
            INNER JOIN COMPROBANTE CM ON CM.ID_CLIENTE = C.ID_CLIENTE
            INNER JOIN FACTURA F ON F.ID_FACTURA = CM.ID_FACTURA
            INNER JOIN SERVICIO_F_B FB ON FB.ID_FACTURA = F.ID_FACTURA
            INNER JOIN SERVICIO S ON S.ID_SERVICIO = FB.ID_SERVICIO)

GO

SELECT * FROM T_CLIENTE() WHERE NOMBRE_SERVICIO = 'Cuidado médico'

GO

```

/*

5. UNÍ LA TABLA CANINO CON LA TABLA RESERVA Y PUSE QUE ME BUSQUE EN LOS CAMPOS
PRECIO SI TIENE REGISTROS CON PRECIO S/. 110.00

*/

```

CREATE FUNCTION T_RESERVA()
RETURNS TABLE
AS
    RETURN (SELECT R.ID_RESERVA CODIGO,
                R.PRECIO PRECIO_SERVICIOS,

```

```
C.NOMBRE,
C.EDAD,
C.PESO,
C.TAMAÑO
FROM CANINO C JOIN RESERVA R ON C.ID_CANINO = R.ID_CANINO)
GO
SELECT * FROM T_RESERVA() WHERE PRECIO_SERVICIOS = 110.00
GO
```

--FUNCIÓN CON VALORES DE LÍNEA

/*

1.- RETORNAR LAS VACUNAS REGISTRADAS EN LA BBDD

*/

```
CREATE FUNCTION DBO.VACUNAS()
```

```
RETURNS @TABLA TABLE(
```

```
    ID_VACUNA VARCHAR(8),
```

```
    TIPO_VACUNA VARCHAR(40)
```

```
)
```

```
AS
```

```
    BEGIN
```

```
        INSERT INTO @TABLA SELECT ID_VACUNA,
```

```
        TIPO_VACUNA FROM VACUNA
```

```
        RETURN
```

```
    END
```

```
GO
```

```
SELECT * FROM DBO.VACUNAS()
```

```
GO
```

```
/*
```

```
2.-RETORNAR LOS CANINOS REGISTRADOS EN LA BBDD
```

```
*/
```

```
CREATE FUNCTION DBO.CANINOS()
```

```
RETURNS @TABLA TABLE(
```

```
    ID_CANINO VARCHAR(8),
```

```
    NOMBRE VARCHAR(20),
```

```
    PESO INT,
```

```
    EDAD INT
```

```
)
```

```
AS
```

```
    BEGIN
```

```
        INSERT INTO @TABLA SELECT ID_CANINO,
```

```
        NOMBRE, PESO, EDAD FROM CANINO
```

```
        RETURN
```

```
    END
```

```
GO
```

```
SELECT * FROM DBO.CANINOS()
```

```
GO
```

```
/*
```

```
3. RETORNAR LOS CLIENTES REGISTRADOS EN LA BBDD
```

```
*/
```

```
CREATE FUNCTION DBO.CLIENTES() RETURNS @TABLA TABLE(
```

```
    ID_CLIENTE VARCHAR(8),
```

```
    NOMBRE VARCHAR(20),
```

```

        APELLIDO VARCHAR(20),
        DNI CHAR(8),
        DIRECCIÓN VARCHAR(40)
    )
AS
    BEGIN
        INSERT INTO @TABLA SELECT ID_CLIENTE,
        NOMBRE, APELLIDO, DNI, DIRECCIÓN FROM CLIENTE
        RETURN
    END
GO
SELECT * FROM DBO.CLIENTES()

```

--VIEWS

--1. MOSTRAR QUE CANINO TIENE DE 4 A MÁS AÑOS

```

SELECT    ID_CANINO, NOMBRE, EDAD, PESO, TAMAÑO
FROM      dbo.CANINO WHERE EDAD >= 4

```

--2. MOSTRAR DE LA TABLA CLIENTE LOS NOMBRES Y APELLIDOS CUYO DNI COMIENZA CON 0

```

SELECT    ID_CLIENTE, NOMBRE, APELLIDO, DNI
FROM      dbo.CLIENTE WHERE DNI LIKE '0%'

```

--3. MOSTRAR DE LA TABLA PET HOTEL LOS NOMBRES DE QUE EMPIEZEN CON P

```

SELECT    ID_PET, DIRECCIÓN, NOMBRE, TELÉFONO
FROM      dbo.PET_HOTEL WHERE NOMBRE LIKE 'P%'

```

--4. UNÍ LA TABLA FACTURA Y CLIENTE; DESPUÉS DE ESO, FILTRÉ LAS TABLAS CON EL CAMPO COSTO TOTAL

-- PARA QUE MUESTRE LOS QUE SON MAYORES O IGUALES A S/. 100.00

```
SELECT TOP (100) PERCENT dbo.FACTURA.ID_FACTURA, dbo.SERVICIO.SERVICIO,  
dbo.FACTURA.COSTO_TOTAL  
  
FROM  dbo.CLIENTE INNER JOIN  
  
      dbo.COMPROBANTE ON dbo.CLIENTE.ID_CLIENTE = dbo.COMPROBANTE.ID_CLIENTE  
INNER JOIN  
  
      dbo.FACTURA ON dbo.COMPROBANTE.ID_FACTURA = dbo.FACTURA.ID_FACTURA  
INNER JOIN  
  
      dbo.SERVICIO_F_B ON dbo.FACTURA.ID_FACTURA = dbo.SERVICIO_F_B.ID_FACTURA  
INNER JOIN  
  
      dbo.SERVICIO ON dbo.SERVICIO_F_B.ID_SERVICIO = dbo.SERVICIO.ID_SERVICIO  
  
WHERE (dbo.FACTURA.COSTO_TOTAL >= 100)  
  
ORDER BY dbo.FACTURA.COSTO_TOTAL
```

--5. MOSTRAR EL CÓDIGO DE LAS RESERVAS CUYOS NOMBRES DE LOS CLIENTES COMIENCEN CON 'E'.

```
SELECT dbo.CLIENTE.ID_CLIENTE, dbo.CLIENTE.NOMBRE, dbo.CLIENTE.APELLIDO,  
dbo.RESERVA.ID_RESERVA  
  
FROM  dbo.CLIENTE INNER JOIN  
  
      dbo.RESERVA ON dbo.CLIENTE.ID_CLIENTE = dbo.RESERVA.ID_CLIENTE  
  
WHERE (dbo.CLIENTE.NOMBRE LIKE 'E%')
```

--6. MOSTRAR TODOS LOS PERROS CUYO COLOR SEA CREMA.

```
SELECT dbo.CANINO.NOMBRE AS [NOMBRE PERRO], dbo.COLOR_CANINO.NOMBRE AS  
COLOR  
  
FROM  dbo.CANINO INNER JOIN  
  
      dbo.COLOR_CANINO ON dbo.CANINO.ID_COLOR = dbo.COLOR_CANINO.ID_COLOR  
  
WHERE (dbo.COLOR_CANINO.NOMBRE = 'Crema')
```

--1. MOSTRAR QUE RESERVA SOBRE PASA LOS 100 SOLES

```
SELECT    ID_RESERVA, FECHA_INI, FECHA_SAL, PRECIO
FROM      dbo.RESERVA WHERE PRECIO >100
```

--2. MOSTRAR LOS RAZAS QUE COMIENCEN CON LA LETRA "B"

```
SELECT    ID_RAZA, TIPO_RAZA, DESCRIPCIÓN
FROM      dbo.RAZA WHERE TIPO_RAZA LIKE 'B%'
```

--UPDATE

-- 1.

```
UPDATE dbo.VACUNA
SET TIPO_VACUNA = 'distemper'
WHERE ID_VACUNA = 'V001'
```

-- 2.

```
UPDATE dbo.CLIENTE
SET NOMBRE = 'Fabiano', APELLIDO = 'Belladoni'
WHERE DNI = '42565679'
```

-- 3.

```
UPDATE RESERVA
SET FECHA_INI = '2022-03-01'
WHERE ID_RESERVA = 'R001'
```

GO

-- 4.

UPDATE CLIENTE

SET NOMBRE = 'Jordy'

WHERE ID_CLIENTE = 'C001'

-- 5.

UPDATE RAZA

SET TIPO_RAZA = 'Border Collie'

WHERE ID_RAZA = 'R017'

-- 6.

UPDATE CANINO

SET NOMBRE = 'Diego'

WHERE ID_CANINO = 'C007'

-- 7.

UPDATE PET_HOTEL

SET TELÉFONO = '990032324'

WHERE ID_PET = 'P009'

-- 8.

UPDATE CANINO

SET TAMAÑO = 'Pequeño'

WHERE ID_CANINO = 'C010'

--DELETE

-- 1.

DELETE FROM RESERVA

WHERE ID_RESERVA = 'R001';

-- 2.

DELETE FROM COLOR_CANINO

WHERE ID_COLOR = 'CO007';

-- 3.

DELETE FROM SERVICIO

WHERE SERVICIO = 'Cámaras web';

-- 4.

DELETE FROM HABITACION_CANINA

WHERE NRO_HABI = 'H001';

-- 5.

DELETE FROM RESERVA

WHERE ID_RESERVA = 'R001';

-- 6.

DELETE FROM CANINO

WHERE ID_CANINO = 'C007'

-- 7.

DELETE FROM HISTORIAL

WHERE ID_HISTORIAL = 'HI008'

-- 8.

DELETE FROM CANINO

WHERE ID_CANINO = 'C010'

--ACCEDEMOS A LOS DIAGRAMAS DE BASE DE DATOS CON EL SIGUIENTE CÓDIGO:

ALTER AUTHORIZATION ON DATABASE :: [PET_HOTEL] TO SA

CONEXION DE SQL SERVER CON JAVA NETBEANS

Este código en Java se centra en la conexión a una base de datos SQL Server y la ejecución de consultas mediante JDBC (Java Database Connectivity). Aquí hay una breve descripción de las clases y métodos principales:

Conexion (Clase): Esta clase se encarga de establecer la conexión con la base de datos SQL Server utilizando JDBC. Contiene métodos para obtener la conexión (`getConexion`) y ejecutar consultas (`ejecutaStatement`).

`getConexion()`: Este método utiliza la clase `DriverManager` para cargar el controlador JDBC y luego establecer la conexión utilizando la URL de conexión, el nombre de usuario y la contraseña proporcionados.

`ejecutaStatement(String query, Connection conexion, String... columnas)`: Este método ejecuta una consulta SQL especificada en el parámetro `query` utilizando la conexión proporcionada. También recibe un número variable de nombres de columnas (`columnas`) que se utilizarán para formatear y mostrar los resultados de la consulta.

Principal (Clase): Esta clase contiene el método `main`, que es el punto de entrada del programa. En este método, se crea una instancia de `Conexion`, se obtiene la conexión a la base de datos y se llama al método `ejecutaStatement` para ejecutar una consulta específica.

En resumen, este código muestra cómo establecer una conexión a una base de datos SQL Server desde Java utilizando JDBC y cómo ejecutar consultas SQL mediante una clase de conexión personalizada. También proporciona un ejemplo de cómo llamar al método `ejecutaStatement` para ejecutar una consulta y mostrar los resultados.

```
package Connection;

import java.sql.Statement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

/**
 *
 * @author Alberto Arroyo
 */
public class Conexion {

    private final String USUARIO = "pet_hotel";

    private final String CONTRASEÑA = "1234";

    private final String URLCONEXION = "jdbc:sqlserver://DESKTOP-STBFGEG\\MSSQL:1433;databaseName=PET_HOTEL;encrypt=true;trustServerCertificate=true";

    private Connection conexion;

    Statement stm;

    ResultSet rs;

    public Connection getConexion() throws SQLException{

        try {

            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

            conexion = DriverManager.getConnection(URLCONEXION, USUARIO, CONTRASEÑA);

        } catch (ClassNotFoundException c) {

            System.out.println("Error al conectarse a la base de datos " + c);

        }

    }

}
```

```

return conexion;

}

public void ejecutaStatement(String query, Connection conexion, String... columnas)
throws SQLException {

    try {

        stm = conexion.createStatement();

        rs = stm.executeQuery(query);

        while (rs.next()) {

            StringBuilder mensaje = new StringBuilder();

            for (int i = 0; i < columnas.length; i++) {

                mensaje.append(columnas[i]).append(": ").append(rs.getString(i + 1));

                if (i < columnas.length - 1) {

                    mensaje.append(", ");

                }

            }

            System.out.println(mensaje);

        }

    } catch (SQLException e) {

        System.out.println("Error: " + e);

    } finally {

        if (rs != null) {

            try {

                rs.close();

            } catch (SQLException e) {

                System.out.println("Error: " + e);

            }

        }

    }

}

```

```
        if (stm != null) {  
            try {  
                stm.close();  
            } catch (SQLException e) {  
                System.out.println("Error: " + e);  
            }  
        }  
        if (conexion != null) {  
            try {  
                conexion.close();  
            } catch (SQLException e) {  
                System.out.println("Error: " + e);  
            }  
        }  
    }  
}
```

```

package Connection;

import java.sql.Connection;
import java.sql.SQLException;

public class Principal {

    public static void main(String[] args) throws SQLException {

```

```

        Connection conn = null;

```

```

        Conexion conx = new Conexion();

```

```

        conn = conx.getConexion();

```

```

        /*

```

En el primer argumento, introduces la consulta sin ningún alias o alguna función que perjudique o modifique

cada columna; en el segundo argumento, lo dejas así. En el último argumento, colocas cada nombre de cada columna

que sacarás con tu consulta

Ejemplo:

```

        conx.ejecutaStatement("SELECT H.ID_HISTORIAL,\n" +
"        H.DESCRIPCIÓN,\n" +
"        R.ID_RESERVA\n" +
"FROM HISTORIAL H\n" +
"RIGHT JOIN RESERVA R ON H.ID_RESERVA = R.ID_RESERVA\n" +
"ORDER BY H.ID_RESERVA DESC", conn, "NÚMERO HISTORIAL", "DESCRIPCIÓN", "NÚMERO RESERVA");

```

```

        */

```

```

        conx.ejecutaStatement("Consulta", conn, "Nombres con una cantidad indefinida de argumentos introducidos según"

```

```

                + "tu consulta saque. Los separas con comas.");

```

```

    }

```

```

}

```