

Problemática:

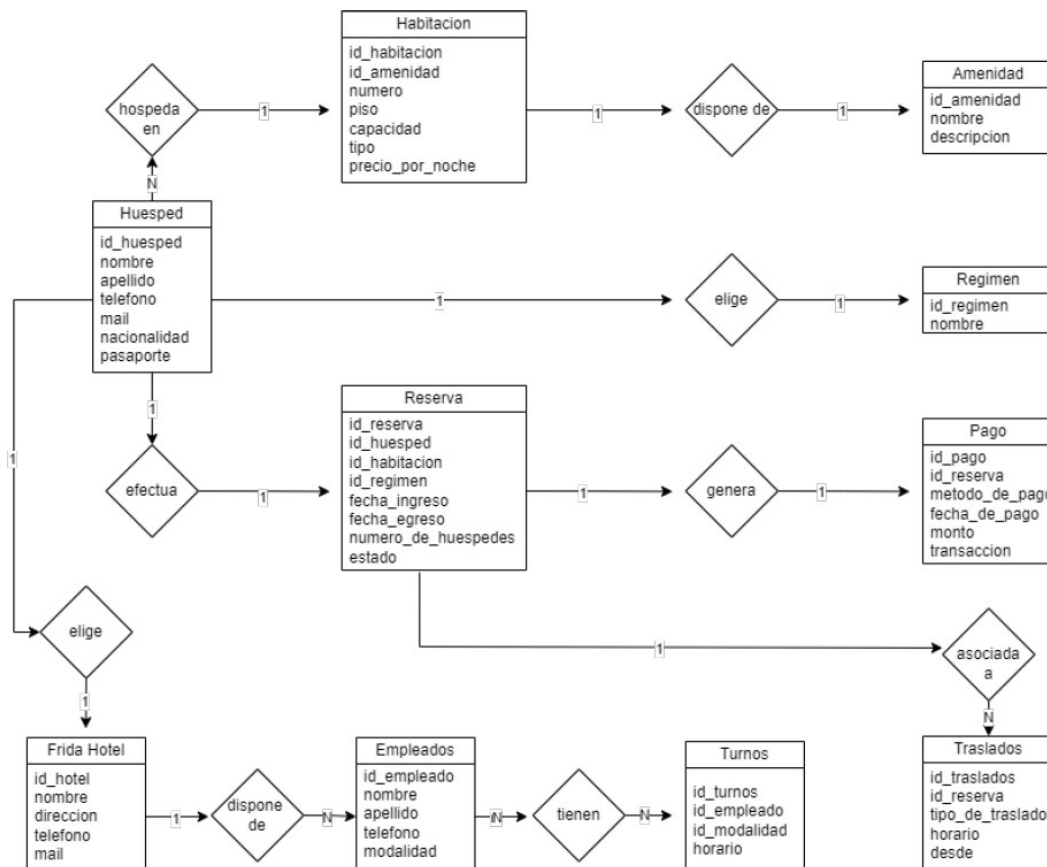
Fernández & Co. es una empresa familiar que ha decidido incursionar en la rama hotelera. Ya poseen la infraestructura y el personal contratado, pero necesitan de un sistema para organizar las reservas y centralizar la información de manera que:

- Se obtenga un registro de huéspedes y sus reservas
- Evitar problemas de overbooking
- Asociar los planes de comida a los huéspedes
- Registrar los pagos en tiempo y forma y asociarlos a su reserva correspondiente
- Evitar todo tipo de inconsistencias y mantener la excelencia

Para el caso de una reserva puntual en un momento dado:

1. Uno o más huéspedes (si son una familia o amigos) puede ocupar una habitación durante esa reserva (N a 1).
2. Una habitación puede ser ocupada por uno o varios huéspedes a la vez ya que algunas tienen capacidad hasta 5 personas (1 a N).
3. Una habitación puede tener muchas amenidades (1 a N).
4. Una amenidad puede estar presente en varias habitaciones (1 a N).
5. Un huésped solo puede elegir un solo régimen de comidas (1 a 1).
6. Un régimen de comidas puede ser elegido por varios huéspedes (1 a N).
7. Un huésped efectúa una sola reserva (1 a 1).
8. Una reserva solo puede estar asociada a un huésped, más allá de que se hospede con su familia (1 a 1).
9. Una reserva genera un único pago (1 a 1).
10. Un pago corresponde a una única reserva (1 a 1).
11. Una reserva puede estar asociada a varios traslados debido a que el hotel cuenta con traslados de ida y vuelta y traslados hacia la isla (1 a N).
12. Un empleado puede tener varios turnos y en esos turnos puede haber varios empleados (N a N).
13. Como estamos hablando de un hotel puntual, decimos que el huésped que realiza la reserva solo puede elegir el Hotel Frida (1 a 1).
14. El hotel dispone de varios empleados (1 a N).

Diagrama Entidad Relación:



Listado de Tablas:

A continuación, se exponen las tablas con sus respectivos detalles:

Tabla Huésped

Campo	Tipo de Dato	PK	FK	NULL
<i>id_huesped</i>	INT	✓		
<i>nombre</i>	VARCHAR(20)			
<i>apellido</i>	VARCHAR(20)			
<i>teléfono</i>	VARCHAR(50)			✓
<i>email</i>	VARCHAR(50)			
<i>nacionalidad</i>	VARCHAR(20)			✓
<i>pasaporte</i>	VARCHAR(20)			

Tabla Amenidad

Campo	Tipo de Dato	PK	FK	NULL
<i>id_amenidad</i>	INT	✓		
<i>nombre</i>	VARCHAR(20)			
<i>descripcion</i>	VARCHAR(50)			

Tabla Habitación

Campo	Tipo de Dato	PK	FK	NULL
<i>id_habitacion</i>	INT	✓		
<i>id_amenidad</i>	INT		✓	
<i>numero</i>	INT			
<i>piso</i>	INT			
<i>capacidad</i>	INT			
<i>tipo</i>	VARCHAR(20)			
<i>precio_por_noche</i>	FLOAT			

Tabla Régimen

Campo	Tipo de Dato	PK	FK	NULL
<i>id_regimen</i>	INT	✓		
<i>nombre</i>	VARCHAR(20)			

Tabla Reserva

Campo	Tipo de Dato	PK	FK	NULL
<i>id_reserva</i>	INT	✓		
<i>id_huesped</i>	INT		✓	
<i>id_habitacion</i>	INT		✓	
<i>id_regimen</i>	INT		✓	
<i>fecha_ingreso</i>	DATE			
<i>fecha_egreso</i>	DATE			
<i>numero_de_huespedes</i>	INT			
<i>estado</i>	VARCHAR(20)			

Tabla Pago

Campo	Tipo de Dato	PK	FK	NULL
<i>id_pago</i>	INT	✓		
<i>id_reserva</i>	INT		✓	
<i>metodo_de_pago</i>	VARCHAR(20)			
<i>fecha_de_pago</i>	DATE			
<i>monto</i>	FLOAT			
<i>transaccion</i>	INT			

Tabla Hotel

Campo	Tipo de Dato	PK	FK	NULL
<i>id_hotel</i>	INT	✓		
<i>nombre</i>	VARCHAR(20)			
<i>direccion</i>	VARCHAR(20)			
<i>telefono</i>	VARCHAR(20)			✓
<i>mail</i>	VARCHAR(50)			

Tabla Empleados

Campo	Tipo de Dato	PK	FK	NULL
<i>id_empleado</i>	INT	✓		
<i>nombre</i>	VARCHAR(20)			
<i>apellido</i>	VARCHAR(20)			
<i>telefono</i>	VARCHAR(20)			✓
<i>modalidad</i>	VARCHAR(20)			

Tabla Turno

Campo	Tipo de Dato	PK	FK	NULL
<i>id_turno</i>	INT	✓		
<i>id_empleado</i>	INT		✓	
<i>horario</i>	VARCHAR(20)			

Tabla Traslado

Campo	Tipo de Dato	PK	FK	NULL
<i>id_traslado</i>	INT	✓		
<i>id_reserva</i>	INT		✓	
<i>tipo_de_traslado</i>	VARCHAR(20)			
<i>horario</i>	TIME			
<i>desde</i>	VARCHAR(20)			

Vistas:

1. vista_huespedes_reservas

Esta vista muestra la información de cada huesped junto con los datos de su reserva, habitación y estado actual, permitiendo así saber que huéspedes tienen reservas confirmadas, pendientes o finalizadas.

Incluye las tablas: huesped, reserva y habitación.

```
CREATE VIEW vista_huespedes_reservas AS
SELECT
    h.id_huesped,
    CONCAT(h.nombre, ' ', h.apellido) AS huesped,
    r.id_reserva,
    hab.numero AS nro_habitacion,
    hab.tipo AS tipo_habitacion,
    r.fecha_ingreso,
    r.fecha_egreso,
    r.estado
FROM reserva r
JOIN huesped h ON r.id_huesped = h.id_huesped
JOIN habitacion hab ON r.id_habitacion = hab.id_habitacion;
```

La consulta podría ser: `SELECT * FROM vista_huespedes_reservas WHERE estado = 'confirmada';`

2. vista_pagos_detalle

Esta vista muestra la información de los pagos realizados por cada huésped, permitiendo saber el método de pago, monto y fecha, pudiendo tener así un control de los ingresos por reservas.

Incluye las tablas: pago, reserva y huesped.

```
CREATE VIEW vista_pagos_detalle AS
SELECT
    p.id_pago,
    CONCAT(h.nombre, ' ', h.apellido) AS huesped,
    p.metodo_de_pago,
    p.fecha_de_pago,
    p.monto,
    p.transaccion,
    r.id_reserva,
    r.estado
FROM pago p
JOIN reserva r ON p.id_reserva = r.id_reserva
JOIN huesped h ON r.id_huesped = h.id_huesped;
```

La consulta podría ser: `SELECT * FROM vista_pagos_detalle WHERE metodo_de_pago = 'efectivo';`

3. vista_empleados_turno

Esta vista muestra el nombre de los empleados junto a su modalidad de trabajo y el turno asignado, permitiendo así tener la información organizada del personal y sus horarios.

Incluye las tablas: empleado y turno.

```
CREATE VIEW vista_empleados_turnos AS
SELECT
    e.id_empleado,
    CONCAT(e.nombre, ' ', e.apellido) AS empleado,
    e.telefono,
    e.modalidad,
    t.horario AS turno
FROM empleado e
JOIN turno t ON e.id_empleado = t.id_empleado;
```

La consulta podría ser: `SELECT * FROM vista_empleados_turnos WHERE turno = 'mañana';`

Funciones:

1. nombre_completo_huesped

Esta función devuelve el nombre completo de un huesped uniendo nombre y apellido evitando usar CONCAT. Usa información de la tabla huesped, campos nombre y apellido.

DELIMITER //

```
CREATE FUNCTION nombre_completo_huesped(  
    p_nombre VARCHAR(50),  
    p_apellido VARCHAR(50)  
)  
RETURNS VARCHAR(100)  
DETERMINISTIC  
BEGIN  
    RETURN CONCAT(p_nombre, ' ', p_apellido);  
END //
```

DELIMITER;

La consulta podría ser: SELECT nombre_completo_huesped(nombre, apellido) AS nombre_completo
FROM huesped;

2. tota_pago_reserva

Esta función indica el total pagado por una reserva sumando todos los pagos asociados, nos permite conocer el monto total abonado por una reserva sin hacer SUM manual en cada consulta.

DELIMITER //

```
CREATE FUNCTION total_pago_reserva(p_id_reserva INT)  
RETURNS DECIMAL(10,2)  
DETERMINISTIC  
BEGIN  
    DECLARE total DECIMAL(10,2);  
    SELECT IFNULL(SUM(monto),0) INTO total  
    FROM pago  
    WHERE id_reserva = p_id_reserva;  
    RETURN total;  
END //
```

DELIMITER;

La consulta podría ser: SELECT total_pago_reserva(5) AS total_pagado;

Procedimientos:

1. reservas_por_estado

Este procedimiento muestra todas las reservas permitiéndonos filtrar por estado que se encuentran las mismas (confirmada, pendiente, check in)

DELIMITER //

```
CREATE PROCEDURE reservas_por_estado(IN p_estado VARCHAR(50))
BEGIN
    SELECT r.id_reserva,
           h.nombre AS huesped,
           r.fecha_ingreso,
           r.fecha_egreso,
           r.estado
    FROM reserva r
    JOIN huesped h ON r.id_huesped = h.id_huesped
    WHERE r.estado = p_estado;
END//
```

DELIMITER;

La consulta podría ser: CALL reservas_por_estado('confirmada');

2. total_por_metodo

Este procedimiento muestra todas las reservas permitiéndonos filtrar por estado que se encuentran las mismas (confirmada, pendiente, check in)

DELIMITER //

```
CREATE PROCEDURE total_por_metodo(IN p_metodo VARCHAR(50))
BEGIN
    SELECT p_metodo AS metodo,
           SUM(monto) AS total_facturado
    FROM pago
    WHERE metodo_de_pago = p_metodo
    GROUP BY metodo_de_pago;
END//
```

DELIMITER;

La consulta podría ser: CALL total_por_metodo('efectivo');

Triggers:

1. actualizar_estado_pago

De esta manera se puede actualizar el estado automáticamente cuando se realiza un pago. Cuando se inserta un pago para una reserva pendiente, la pasa a confirmada.

DELIMITER //

```
CREATE TRIGGER actualizar_estado_pago
AFTER INSERT ON pago
FOR EACH ROW
BEGIN
    UPDATE reserva
    SET estado = 'confirmada'
    WHERE id_reserva = NEW.id_reserva
    AND estado = 'pendiente';
END//
```

2. validar_monto_pago

De esta manera se evitan montos negativos en los pagos.

DELIMITER //

```
CREATE TRIGGER validar_monto_pago
BEFORE INSERT ON pago
FOR EACH ROW
BEGIN
    IF NEW.monto <= 0 THEN
        SET NEW.monto = 0;
    END IF;
END//
```

Script:

CREATE SCHEMA Hotel;
USE Hotel;

```
CREATE TABLE huesped (  
  id_huesped INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  apellido VARCHAR(50) NOT NULL,  
  telefono VARCHAR(50),  
  email VARCHAR(50) NOT NULL,  
  nacionalidad VARCHAR(50),  
  pasaporte VARCHAR(50) NOT NULL  
);
```

```
INSERT INTO huesped (nombre, apellido, telefono, email, nacionalidad, pasaporte) VALUES  
( 'Kippy', 'Ellar', '362-612-3257', 'kellar0@lulu.com', 'Czech Republic', 'CCRTIT2TBTL'),  
( 'Cherilyn', 'Carson', '412-704-6345', 'ccarson1@mit.edu', 'China', 'GENODEF1GLU'),  
( 'Tristan', 'Rowly', '548-767-7655', 'trowly2@chronoengine.com', 'Bulgaria', 'GENODEF1M07'),  
( 'Winnny', 'Ivain', '155-980-9506', 'wivain3@so-net.ne.jp', 'Vietnam', 'MELIDEH1XXX'),  
( 'Teddie', 'Lochet', '311-179-1079', 'tlochett4@cbc.ca', 'Mayotte', 'RZTIAT22268'),  
( 'Corine', 'Cran', '222-218-4005', 'ccran5@techcrunch.com', 'China', 'SVRNUS33'),  
( 'Dennis', 'Flecknell', '470-693-9673', 'dflecknell6@chron.com', 'Uruguay', 'ICRAITRRFZ0'),  
( 'Aundrea', 'Goscar', '581-166-8665', 'agoscar7@nyu.edu', 'Thailand', 'GENODEF1LRU'),  
( 'Yolande', 'Spadari', '783-789-6217', 'yspadari8@ft.com', 'China', 'SPWTAT21XXX'),  
( 'Malachi', 'Bousler', '710-658-5331', 'mbousler9@msn.com', 'Sweden', 'CEPANL2AXXX'),  
( 'Arthur', 'Sweett', '564-682-4512', 'asweetta@rediff.com', 'Venezuela', 'BAPPIT21193'),  
( 'Russell', 'Beneteau', '827-790-1037', 'rbeneteaub@wsj.com', 'Denmark', 'CITIUS33'),  
( 'Juan', 'Perez', '899-112-9843', 'juanperez@gmail.com', 'Denmark', 'FGTIKS04'),  
( 'Gina', 'Hames', '116-925-8620', 'ghames0@accuweather.com', 'Ghana', 'BAPPIT21243'),  
( 'Muire', 'Hempshall', '769-930-1479', 'mhempshall2@amazon.com', 'Argentina', 'ICRAITRRKT0'),  
( 'Darrelle', 'Gaythor', '263-697-5834', 'dgaythor3@senate.gov', 'Argentina', 'SBININBB684'),  
( 'Denna', 'Weeden', '373-160-8160', 'dweeden4@youku.com', 'Switzerland', 'PARXEE22XXX'),  
( 'Anallise', 'Sreenan', '672-859-9759', 'asreenan1@ocn.ne.jp', 'Colombia', 'FTSBDEFAXXX');
```

```
CREATE TABLE amenidad(  
  id_amenidad INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  descripcion VARCHAR(50) NOT NULL  
);
```

```
INSERT INTO amenidad (nombre, descripcion) VALUES  
( 'Simple', 'sin_amenidad'),  
( 'Spa', 'jacuzzi_en_habitacion'),  
( 'Piscina', 'piscina_privada'),  
( 'Gym', 'acceso_gimnasio_vip');
```

```
CREATE TABLE habitacion(  
  id_habitacion INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  id_amenidad INT NOT NULL,  
  numero INT NOT NULL,  
  piso INT NOT NULL,  
  capacidad INT NOT NULL,  
  tipo VARCHAR(50) NOT NULL,  
  precio_por_noche DECIMAL(10,2) NOT NULL,  
  FOREIGN KEY (id_amenidad) REFERENCES amenidad(id_amenidad)  
);
```

```
INSERT INTO habitacion (id_amenidad, numero, piso, capacidad, tipo, precio_por_noche) VALUES  
(1, 101, 1, 2, 'Doble_simple', 100000.00),  
(1, 102, 1, 1, 'Individual_simple', 70000.00),
```

```
(1, 103, 1, 3, 'Triple_simple', 150000.00),
(1, 104, 1, 4, 'Cuadruple_simple', 200000.00),
(1, 105, 1, 5, 'Quintuple_simple', 300000.00),
(2, 201, 2, 2, 'Doble_premium', 200000.00),
(2, 202, 2, 1, 'Individual_premium', 140000.00),
(2, 203, 2, 3, 'Triple_premium', 300000.00),
(2, 204, 2, 4, 'Cuadruple_premium', 400000.00),
(2, 205, 2, 5, 'Quintuple_premium', 600000.00),
(3, 301, 3, 2, 'Doble_suite', 250000.00),
(3, 302, 3, 1, 'Individual_suite', 190000.00),
(3, 303, 3, 3, 'Triple_suite', 350000.00),
(3, 304, 3, 4, 'Cuadruple_suite', 450000.00),
(3, 305, 3, 5, 'Quintuple_suite', 650000.00),
(4, 401, 4, 2, 'Doble_vip', 300000.00),
(4, 402, 4, 1, 'Individual_vip', 240000.00),
(4, 403, 4, 3, 'Triple_vip', 400000.00),
(4, 404, 4, 4, 'Cuadruple_vip', 500000.00),
(4, 405, 4, 5, 'Quintuple_vip', 700000.00);
```

```
CREATE TABLE regimen(
  id_regimen INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL
);
```

```
INSERT INTO regimen (nombre) VALUES
('all_inclusive'),
('media_pension'),
('desayuno'),
('solo_hospedaje');
```

```
CREATE TABLE reserva(
  id_reserva INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  id_huesped INT NOT NULL,
  id_habitacion INT NOT NULL,
  id_regimen INT NOT NULL,
  fecha_ingreso DATE NOT NULL,
  fecha_egreso DATE NOT NULL,
  numero_de_huespedes INT NOT NULL,
  estado VARCHAR(50) NOT NULL,
  FOREIGN KEY (id_huesped) REFERENCES huesped(id_huesped),
  FOREIGN KEY (id_habitacion) REFERENCES habitacion(id_habitacion),
  FOREIGN KEY (id_regimen) REFERENCES regimen(id_regimen)
);
```

```
INSERT INTO reserva (id_huesped, id_habitacion, id_regimen, fecha_ingreso, fecha_egreso,
numero_de_huespedes, estado) VALUES
(11, 4, 1, '2025-09-30', '2025-10-13', 4, 'check_in'),
(2, 20, 4, '2025-10-31', '2025-11-11', 5, 'confirmada'),
(10, 1, 4, '2025-11-30', '2025-12-10', 2, 'confirmada'),
(18, 5, 3, '2026-01-31', '2026-02-20', 5, 'pendiente'),
(5, 18, 1, '2026-01-05', '2026-01-25', 3, 'pendiente'),
(15, 7, 2, '2025-09-05', '2025-09-25', 1, 'finalizado'),
(1, 2, 2, '2025-12-05', '2025-12-14', 1, 'confirmada'),
(8, 15, 3, '2025-12-17', '2025-12-27', 5, 'confirmada'),
(3, 8, 3, '2026-05-05', '2026-05-25', 3, 'pendiente'),
(6, 3, 4, '2025-12-01', '2025-12-11', 3, 'pendiente'),
(4, 11, 2, '2025-10-20', '2025-11-05', 2, 'confirmada'),
(9, 6, 1, '2025-09-30', '2025-10-10', 2, 'check_in'),
(12, 9, 4, '2026-01-23', '2026-02-13', 4, 'confirmada'),
(13, 10, 3, '2025-12-24', '2025-12-30', 5, 'confirmada'),
(7, 12, 3, '2025-12-05', '2025-12-24', 1, 'pendiente'),
```

```
(14, 17, 2, '2026-02-02', '2026-02-10', 1, 'confirmada'),  
(16, 14, 1, '2025-11-05', '2025-11-14', 4, 'confirmada'),  
(17, 13, 4, '2026-01-02', '2026-01-20', 3, 'pendiente');
```

```
CREATE TABLE pago(  
    id_pago INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    id_reserva INT NOT NULL,  
    metodo_de_pago VARCHAR(50) NOT NULL,  
    fecha_de_pago DATE NOT NULL,  
    monto DECIMAL (10,2) NOT NULL,  
    transaccion INT NOT NULL,  
    FOREIGN KEY (id_reserva) REFERENCES reserva(id_reserva)  
);
```

```
INSERT INTO pago (id_reserva, metodo_de_pago, fecha_de_pago, monto, transaccion) VALUES  
(10, 'efectivo', '2025-09-30', 2800000.00, 1000023),  
(7, 'tarjeta_de_credito', '2025-09-21', 8400000.00, 1000037),  
(5, 'tarjeta_de_debito', '2025-09-20', 1100000.00, 1000029),  
(4, 'tarjeta_de_credito', '2025-09-30', 6300000.00, 1000032),  
(16, 'efectivo', '2026-01-05', 8400000.00, 1000030),  
(12, 'tarjeta_de_debito', '2025-09-30', 2940000.00, 1000044),  
(2, 'efectivo', '2025-12-05', 700000.00, 1000035),  
(8, 'tarjeta_de_credito', '2025-09-01', 7150000.00, 1000043),  
(15, 'tarjeta_de_credito', '2025-09-21', 3300000.00, 1000040),  
(17, 'tarjeta_de_debito', '2025-10-02', 1650000.00, 1000042),  
(3, 'tarjeta_de_credito', '2025-10-01', 3400000.00, 1000028),  
(13, 'tarjeta_de_credito', '2025-09-05', 2200000.00, 1000035),  
(14, 'efectivo', '2026-01-23', 8800000.00, 1000047),  
(18, 'tarjeta_de_credito', '2025-10-05', 4200000.00, 1000050),  
(11, 'tarjeta_de_credito', '2025-10-06', 3040000.00, 1000051),  
(6, 'tarjeta_de_credito', '2025-09-30', 2160000.00, 1000041),  
(1, 'tarjeta_de_debito', '2025-09-03', 4500000.00, 1000049),  
(9, 'tarjeta_de_credito', '2025-09-11', 6650000.00, 1000048);
```

```
CREATE TABLE hotel(  
    id_hotel INT NOT NULL PRIMARY KEY,  
    nombre VARCHAR(20) NOT NULL,  
    direccion VARCHAR(20) NOT NULL,  
    telefono VARCHAR(20) NOT NULL,  
    mail VARCHAR (50) NOT NULL  
);
```

```
INSERT INTO hotel (id_hotel, nombre, direccion, telefono, mail) VALUES  
(1, 'frida hotel', 'corrientes_1111', '111-9363-0327', 'fridahotel@hotel.com');
```

```
CREATE TABLE empleado(  
    id_empleado INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(20) NOT NULL,  
    apellido VARCHAR(20) NOT NULL,  
    telefono VARCHAR(20) NOT NULL,  
    modalidad VARCHAR (20) NOT NULL  
);
```

```
INSERT INTO empleado (nombre, apellido, telefono, modalidad) VALUES  
( 'Gayelord', 'Bullman', '266-309-5857', 'cama_adentro'),  
( 'Curcio', 'Bussey', '616-955-0855', 'cama_adentro'),  
( 'Stephana', 'Harmston', '356-555-0694', 'franquero'),  
( 'Bob', 'Climo', '660-304-0087', 'cama_adentro'),  
( 'Elonore', 'Jury', '884-920-8871', 'cama_adentro'),  
( 'Alexa', 'Mildmott', '348-865-2361', 'cama_adentro'),  
( 'Fonsie', 'Bebbell', '804-581-5289', 'franquero'),
```

```
('Leesa', 'Tolotti', '337-210-4604', 'cama_adentro'),  
('Una', 'Jenman', '898-653-3044', 'cama_adentro'),  
('Lucille', 'Lambourne', '231-438-6714', 'cama_adentro');
```

```
CREATE TABLE turno(  
    id_turno INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    id_empleado INT NOT NULL,  
    horario VARCHAR(20) NOT NULL,  
    FOREIGN KEY (id_empleado) REFERENCES empleado(id_empleado)  
);
```

```
INSERT INTO turno (id_empleado, horario) VALUES  
(8, 'mañana'),  
(9, 'tarde'),  
(1, 'noche'),  
(10, 'mañana'),  
(5, 'tarde'),  
(6, 'noche'),  
(2, 'mañana'),  
(3, 'tarde'),  
(4, 'mañana'),  
(7, 'noche');
```

```
CREATE TABLE traslado(  
    id_traslado INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    id_reserva INT NOT NULL,  
    tipo_de_traslado VARCHAR(20) NOT NULL,  
    horario TIME NOT NULL,  
    desde VARCHAR(20) NOT NULL,  
    FOREIGN KEY (id_reserva) REFERENCES reserva(id_reserva)  
);
```

```
INSERT INTO traslado (id_reserva, tipo_de_traslado, horario, desde) VALUES  
(18, 'limusina', '13:00', 'aeroparque'),  
(3, 'van', '14:35', 'ezeiza'),  
(7, 'minibus', '18:05', 'ezeiza'),  
(11, 'minibus', '19:06', 'ezeiza'),  
(14, 'van', '20:00', 'ezeiza'),  
(4, 'minibus', '23:00', 'ezeiza'),  
(8, 'van', '00:30', 'ezeiza'),  
(12, 'limusina', '14:20', 'ezeiza'),  
(15, 'auto_privado', '17:15', 'aeroparque'),  
(1, 'auto_privado', '19:00', 'ezeiza'),  
(5, 'van', '20:10', 'ezeiza'),  
(9, 'auto_privado', '19:15', 'aeroparque'),  
(13, 'van', '23:00', 'ezeiza'),  
(16, 'limusina', '08:00', 'ezeiza'),  
(2, 'van', '07:30', 'ezeiza'),  
(6, 'van', '05:00', 'ezeiza'),  
(10, 'minibus', '03:30', 'aeroparque'),  
(17, 'van', '01:00', 'ezeiza');
```