

Proyecto Final: Hoteles Luna

ChiquesSQL

Reporte Ejecutivo

FUNDAMENTOS DE BASES DE DATOS

2
0
2
4

IVANA IX CHEL BONILLA NEGRETE

DYLAN ENRIQUE JUÁREZ MARTÍNEZ

DANIEL ROJO MATA

Índice general

1.	Resumen	3
2.	Estructura del proyecto	4
3.	Modelo Entidad Relación	4
3.1.	Entidades y atributos	4
3.2.	Relaciones	6
4.	Modelo Relacional	8
4.1.	Entidades Fuertes	8
4.2.	Relaciones que generan tabla; Muchos a muchos (N:M)	12
4.3.	Relaciones que modifican; Uno a Muchos (1:N)	12
4.4.	Relaciones Uno a Uno (1:1)	13
4.5.	Dominio de atributos y restricciones	13
4.6.	Restricciones	15
4.7.	Dependencias funcionales:	15
5.	Restricciones de Dominio	15
6.	Decisiones de diseño para el DDL	22
7.	Poblamiento de la Base de Datos	23
7.1.	Tabla hotel	23
7.2.	Tabla telefonoHotel	25
8.	Consultas	26
9.	Conclusiones	36

Imágenes utilizadas

1.	Hoteles que son petFriendly.	26
2.	Habitaciones a cargo (idHabitacion) por miembro del personal de apoyo. La consulta está ordenada por apellido paterno.	26
3.	Huéspedes que han reservado un evento más de una vez, ya que podemos interpretar de ello que son clientes a quienes les agradó el servicio y probablemente vuelvan a hacerlo.	27
4.	Empleados que cumplen años el mes próximo por orden de día (incluye el hotel en que trabajan).	28
5.	Muestra las nacionalidades de los huéspedes más recurrentes que visitan la cadena.	29
6.	Huéspedes con membresía que han rentado una habitación Penthouse y tienen su membresía activa.	29
7.	Los 20 empleados con mayor antigüedad y a qué hotel pertenecen (para darles un agradecimiento – por su prolongada confianza).	30
8.	Habitaciones con costo mayor a 4500 por tipo de habitación.	30
9.	Salones con capacidad mayor a 100 que han tenido eventos este año.	31
10.	Huéspedes con alta probabilidad de volver a hospedarse que tengan mascotas.	31
11.	La información de hoteles cuyo precio de habitación individual sea mayor a 250. Deberán ordenar la información a partir del precio.	32
12.	Huéspedes con membresía que hayan organizado un evento entre las fechas 01 de junio del 2023 al 31 de marzo del 2024 (el último año).	33
13.	Los recepcionistas y personal de comida que vivan en el mismo estado y que no se encuentren trabajando en el mismo hotel.	34
14.	Huéspedes con mascotas y que se hayan alojado en una habitación Penthouse.	34
15.	El personal de limpieza y el personal de apoyo, que compartan estado y que se encuentren trabajando en el mismo hotel. La información está ordenada a partir del estado.	35
16.	Personal de la cadena ordenado por hotel.	35

1. Resumen

El proyecto consiste en diseñar y desarrollar un sistema de gestión de información para la cadena de hoteles "Hoteles Luna". La cadena busca modernizar su forma de administrar la información de sus hoteles, que actualmente se realiza mediante registros físicos en archivos, lo cual no es óptimo en la era digitalizada actual.

El objetivo es proporcionar una solución que permita administrar los datos de manera sencilla, centralizada y consistente. Para lograrlo, es necesario tener en cuenta varios aspectos:

- **Información de los Hoteles:** Nombre, dirección, teléfono(s), número de habitaciones, número de habitaciones disponibles, número de habitaciones ocupadas, hora de check-in y hora de check-out.
- **Servicios:** Se debe tener un registro de los servicios que ofrece cada hotel, como piscina, estacionamiento, restaurante, bar, etc.
- **Tipos de Habitaciones:** Se deben considerar cuatro tipos de habitaciones: individuales, dobles, cuádruples y penthouse, con información sobre número de habitación, número de camas, disponibilidad y costo por noche.
- **Información de Huéspedes:** Datos básicos de los huéspedes, incluyendo nombre completo, nacionalidad, teléfono(s), fecha de nacimiento, correo electrónico(s) y género.
- **Membresías:** Los huéspedes pueden obtener membresías si se hospedan al menos 12 veces al año, con la posibilidad de que queden inactivas si no se cumple este requisito.
- **Mascotas:** Algunos hoteles son pet-friendly, permitiendo a los huéspedes hospedarse con sus mascotas por un costo extra.
- **Personal del Hotel:** Se debe registrar el personal del hotel, incluyendo recepcionistas, personal de apoyo, limpieza interna y servicio de comida, con información como RFC, nombre completo, dirección, teléfono(s), fecha de inicio de contratación, fecha de nacimiento, correo electrónico(s), género y horario.
- **Salones para Eventos:** Los hoteles pueden ofrecer salones para eventos especiales, con información sobre el evento, número de asistentes y validación de disponibilidad.
- **Forma de Pago:** Se debe registrar la forma de pago de los huéspedes, ya sea en efectivo o con tarjeta, incluyendo el número de tarjeta y la fecha de vencimiento en caso de ser tarjeta.
- **Cobro a Huéspedes:** Al momento de cobrar a un huésped, se debe tener en cuenta el número de noches que estará ocupando, el número de huéspedes, la presencia de mascotas, y los servicios adicionales solicitados.
- **Encuesta de Salida:** Se proporcionará una encuesta de salida para que los huéspedes califiquen su experiencia en el hotel.

En esencia, el proyecto implica el diseño y desarrollo de una base de datos funcional que cumpla con todos estos requisitos para que "Hoteles Luna" pueda operar de manera eficiente. Es crucial cumplir con las reglas de negocio establecidas para garantizar el éxito del proyecto y potenciales oportunidades futuras con el cliente.

2. Estructura del proyecto

La estructura del proyecto consta de tres directorios principales: **Diagramas**, **Docs** y **SQL**, junto con un archivo **README**.

En el directorio **Diagramas** se encuentran archivos relacionados con los diagramas del proyecto. El directorio **Docs** contiene documentos, como un diccionario y un reporte ejecutivo.

El directorio **SQL** alberga archivos relacionados con consultas, definición y manipulación de datos.

Por último, el archivo **README** proporciona el nombre de los integrantes del equipo.

```
.
|-- Diagramas
|   |-- ERChiquesSQL.drawio
|   |-- ERChiquesSQL.png
|   |-- RelacionalChiquesSQL.drawio
|   |-- RelacionalChiquesSQL.png
|-- Docs
|   |-- DiccionarioChiquesSQL.pdf
|   |-- ReporteEjecutivoChiquesSQL.pdf
|-- SQL
|   |-- Consultas.sql
|   |-- DDL.sql
|   |-- DML.sql
|-- README_ChiquesSQL.pdf
```

Usted está leyendo el **ReporteEjecutivo**.

3. Modelo Entidad Relación

La construcción del diagrama se realizó por pasos: primero se identificaron las posibles entidades con sus atributos, luego se añadieron las relaciones, después se modificaron entidades para añadir superentidades, se establecieron cardinalidades y participaciones y finalmente se agregaron detalles relacionados a las relaciones y sus atributos.

3.1. Entidades y atributos

- Hotel: ID hotel, número de habitaciones ocupadas, nombre del establecimiento, teléfono, número total de habitaciones, número de habitaciones disponibles (calculado a partir de

las habitaciones totales y ocupadas), dirección (compuesto por: número exterior, número interior, colonia, calle, estado), servicio (piscina, estacionamiento, restaurante, bar, etc.), petfriendly, hora check-out y hora check-in. Identificador: ID hotel. Se eligió una llave sustituta dado que no se encontró una llave natural para la entidad.

- Persona: ID de persona, nombre completo (compuesto por nombre, apellido paterno y apellido materno), teléfono, fecha de nacimiento, correo electrónico y género. Identificador: ID de persona. Nuevamente se eligió una llave sustituta dado que facilita las búsquedas.
- Habitación: ID de Habitación, número de camas, número de habitación, disponibilidad, costo por noche. Identificador: ID de Habitación. Inicialmente se consideró utilizar el número de habitación como identificador, sin embargo, se determinó que este no sería único puesto que se trata de una cadena de hoteles. Por lo que se optó por uno sustituto.
- Pago: ID de pago, número de huéspedes (Adultos y niños), número de mascotas, número de noches y cobro total (atributo calculado a partir de los rubros anteriores). Identificador: ID de pago. Se optó por ese identificador dado que no se tiene una combinación de atributos que nos proporcione una llave única.
- Forma de pago: ID de forma de pago es su único atributo e identificador. La finalidad de dicha entidad es proporcionar una conexión con las entidades Tarjeta y Efectivo.
- Tarjeta (hereda de la superentidad *Forma de pago*): Fecha de vencimiento y número de tarjeta. Identificador heredado: ID forma.
- Huésped: Hereda el atributo ID de persona, y adicionalmente cuenta con atributo propio nacionalidad. Identificador: ID de persona.
- Evento: Número de asistentes, número de membresía, ID de evento, nombre del huésped y nombre del salón. Identificador: ID de evento. Se eligió un identificador sustituto dado que ninguna combinación entre los atributos existentes es necesariamente única.
- Empleado: idPersona, RFC, horario (compuesto por la hora de entrada y salida), fecha de contratación, dirección (compuesto por número interior, número exterior, colonia, calle y estado). Identificador: idPersona. Su identificador es heredado de la superentidad Persona.
- Salón: Nombre del salón, ID de salón, capacidad, horario de atención (compuesto por la hora de inicio y fin), dirección (compuesto por número exterior, número interior, colonia, calle y estado). Identificador: ID de salón. Se eligió una llave sustituta por simplicidad y dado que una llave compuesta podía resultar no única bajo circunstancias especiales.
- Encuesta: ID de encuesta, posibilidad de recomendación, limpieza de las instalaciones, servicio recibido, posibilidad de volverse a hospedar y comentario. Identificador: ID de encuesta. Se eligió un identificador sustituto dado que no se encontró uno que cumpliera con ser natural.
- Efectivo (hereda de la superentidad *Forma de pago*).
- Individual (hereda de la superentidad *Habitación*).

- Doble (hereda de la superentidad *Habitación*).
- Cuadruple (hereda de la superentidad *Habitación*).
- Penthouse (hereda de la superentidad *Habitación*).
- Membresía: Activa y número de membresía. Identificador: Número de membresía. Se eligió como identificador natural al número de membresía, esto puesto que cumple con ser único, tiene significado en nuestro caso de uso y no cambia.
- Servicio extra: ID de servicio, precio y número de platos. Identificador: ID de servicio.
- Servicio de comida (hereda de la superentidad *Empleado*).
- Recepcionista (hereda de la superentidad *Empleado*).
- Personal de apoyo (hereda de la superentidad *Empleado*): Número de habitaciones a cargo (atributo calculado).
- Limpieza interna (hereda de la superentidad *Empleado*).

3.2. Relaciones

- Hotel DISPONER Habitación
Cardinalidad: N:M muchos a muchos porque cada hotel dispone de todas las habitaciones y todas las habitaciones disponen de un hotel.
Participación:
Parcial de lado del Hotel porque puede no disponer de todas las habitaciones (temas de reparación o reformas).
Total del lado de habitaciones porque si hay una habitación, ésta forzosamente dispone a un hotel (para nuestro caso).
- Hotel TENER Persona
Cardinalidad: 1:N uno a muchos porque cada hotel puede tener muchas personas y una persona solo puede tener un hotel, esto considerando que cada persona, empleado o huésped no va a pertenecer a más de un hotel simultáneamente porque no lo necesita.
Participación: Total de ambos lados, porque el hotel tendrá a todos los registrados y todos las personas de nuestro interés para este caso pertenecen a un hotel (de otra forma no estarían registrados).
- Huésped RENTAR Habitación
Cardinalidad: N:M uno a muchos porque un huésped puede rentar más de una habitación, por ejemplo, si lleva niños y queda como titular de la habitación de los menores. Del otro lado, una habitación puede rentarse por más de un huésped.
Participación: Total de lado del huésped porque todos necesariamente tienen habitación (o no serían huéspedes). Parcial del lado de habitación porque no todas las habitaciones necesariamente están ocupadas.

- Huésped ELEGIR Forma de pago
Cardinalidad: 1:1 Uno a uno porque se especifica que un cliente puede elegir efectivo o tarjeta. Mientras una forma de pago puede ser elegida por muchos clientes, pero no con los mismos datos.
Participación: Total para los huéspedes porque todos eligen una forma de pago al registrarse y parcial para las formas de pago porque puede darse el caso de que no todas son elegidas por un huésped.
- Huésped RESERVAR Evento
Cardinalidad: 1:N un huésped puede reservar más de un evento, por ejemplo, si desea reservar uno para navidad y otro para año nuevo, mientras un evento no puede ser reservado por más de un cliente ya que se traslaparían y habría quejas al reservar a dos personas distintas.
Participación: Parcial de lado de huéspedes porque no todos los deben reservar eventos, es un servicio extra. Por otro lado, es total de lado de eventos porque siempre que se lleve a cabo uno, este solo podrá ser reservado por un huésped.
- Huésped OBTENER Membresía
Cardinalidad: 1:1 un huésped solo puede obtener una membresía (las membresías no son acumulables) y una membresía solo puede ser obtenida por un huésped (no son transferibles ni asignadas por grupo).
Participación: Parcial de lado de huéspedes porque no todos los huéspedes son acreedores a membresías (solo si cumplen con el número de visitas anual) y total de lado de membresía porque estas son un beneficio exclusivo de los huéspedes, en otras palabras, siempre que se asigne una será a un huésped.
- Evento VALIDAR Salón
Cardinalidad: 1:1 un evento ocupa un salón, y un salón es ocupado por un solo evento a la vez (no hay eventos simultáneos en el mismo espacio).
Participación: Total de lado de evento porque todos deben contar con un salón para efectuarse. Parcial de lado de salón porque puede haber Salones sin ocuparse.
- Huésped RESPONDER Encuesta
Cardinalidad: 1:N un huésped puede responder más de una encuesta si visita el hotel varias veces y una encuesta tendrá la opinión solo del cliente en cuestión.
Participación: Total de ambos lados porque todos a todos los huéspedes se les realizará la encuesta de salida y todas las encuestas registradas son respondidas por huéspedes.
- huésped REALIZAR Pago
Cardinalidad: 1:1 un huésped realizará el pago correspondiente al final de su estadía y un pago será realizado por un cliente solo cliente.
Participación: Total de ambos lado porque todos los huéspedes realizarán pagos. Mientras cada pago necesariamente será realizado por un huésped (cliente).

- Pago AGREGAR Servicio Extra

Cardinalidad: 1:N un pago puede incluir muchos servicios, mientras un servicio no puede ser agregado a más de un pago.

Participación: Parcial de lado de pago porque no todos los pagos agregan servicios extra (el cliente puede no solicitar alguno). Total de lado de servicio extra porque siempre que se preste el servicio este se agregará el monto final a pagar.

- Personal de apoyo ATENDER Evento

Cardinalidad: N:M muchos a muchos. Un empleado puede atender muchos eventos si no se traslapan y un evento puede ser atendido por muchos integrantes del personal de apoyo.

Participación: Parcial de lado de Empleado porque no todo el personal necesariamente atenderá cada evento. Total de lado de evento porque siempre que se efectúe uno será atendido por el personal del hotel.

- Personal de apoyo SER RESPONSABLE Habitación

Cardinalidad 1:N uno a muchos. Un empleado puede atender muchas habitaciones y una habitación es atendida por solo un empleado. Participación: Total de ambos lados porque todas las habitaciones son atendidas por un miembro del personal de apoyo y cada miembro del personal de apoyo será responsable de atender una o más habitaciones.

Relaciones con atributos:

- RESERVAR: tiene los atributos fechaEvento, horaInicio y horaFin.

4. Modelo Relacional

A continuación se describe la traducción del modelo Entidad Relación al Modelo Relacional.

4.1. Entidades Fuertes

Un tipo de entidad fuerte se convierte en una relación con los mismos atributos que la describen. La llave primaria de la relación es el identificador del tipo de entidad.

Los atributos de color rojo y azul fueron añadidos con base a algún tipo de relación, para saber la relación involucrada, véase la sección 4.3.

1. hotel(idHotel, nombreEstablecimiento, horarioInicio, horarioFin, petFriendly, servicio, numeroInterior, numeroExterior, colonia, calle, estado)

- Los atributos numeroHabitacionesOcupadas, numeroHabitacionesDisponibles y TotalHabitaciones no se añaden puesto que son atributos calculados.
- Al ser telefono un atributo multivaluado, se crea una tabla con una columna que corresponda a la llave primaria del tipo de entidad del que éste es atributo y otra para el atributo.

2. telefonoHotel(idHotel, telefonoHotel)

- De la entidad **Habitacion**, se genera una especialización total con disyunción. Por lo que se crean las siguientes tablas: INDIVIDUAL, DOBLE, CUADRUPLE, PENTHOUSE. Éstas heredan las características de la entidad padre, pero la entidad padre no aparece.

3. individual(idHabitacion, numeroDeHabitacion, numeroDeCamas, disponibilidad, costoPorNoche, **idPersona**)

4. doble(idHabitacion, numeroDeHabitacion, numeroDeCamas, disponibilidad, costoPorNoche, **idPersona**)

5. cuadruple(idHabitacion, numeroDeHabitacion, numeroDeCamas, disponibilidad, costoPorNoche, **idPersona**)

6. penthouse(idHabitacion, numeroDeHabitacion, numeroDeCamas, disponibilidad, costoPorNoche, **idPersona**)

- De la entidad **Persona**, se genera una especialización total con disyunción. Por lo que se crean las siguientes tablas: EMPLEADO y HUESPED. Éstas deben heredar las características de la entidad padre, pero la entidad padre no aparece. Sin embargo, nótese que a su vez, la entidad **Empleado** genera una especialización total con disyunción. De igual manera, se crean las siguientes tablas: SERVICIODECOMIDA, RECEPCIONISTA, PERSONALDEAPOYO, LIMPIEZAINTERNA, pero ahora, éstas últimas deben heredar las propiedades de la entidad **Persona** y las de la entidad *Empleado*.
Lo relacionado a HUESPED solo hereda lo de **Persona**.

7. huesped(idPersona, nombre, paterno, materno, fechaNacimiento, genero, nacionalidad, **idHotel**, **idFormaEfectivo**, **idFormaTarjeta**)

- Al ser **telefono** y **correoElectronico** atributos multivaluados, se crea una tabla con una columna que corresponda a la llave primaria del tipo de entidad del que éste es atributo y otra para el atributo. Se cambia el nombre a *telefonoHuesped* y *correoHuesped* para evitar confusiones.

8. telefonoHuesped(idPersona, telefonoHuesped)

9. correoHuesped(idPersona, correoHuesped)

10. servicioComida(idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, fechaContratación, horarioInicio, HorarioFin, numeroInterior, numeroExterior, colonia, calle, estado, idHotel)
 - Al ser **telefono** y **correoElectronico** atributos multivaluados y puesto que tenemos una especialización total con disyunción se crea una tabla con una columna que corresponda a la llave primaria del tipo de entidad del que éste es atributo y otra para el atributo. Ambas tablas se nombran: *telefonoServicioComida* y *correoServicioComida*.
11. telefonoServicioComida(idPersona, telefonoServicioComida)
12. correoServicioComida(idPersona, correoServicioComida)
13. recepcionista(idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, fechaContratación, horarioInicio, HorarioFin, numeroInterior, numeroExterior, colonia, calle, estado, idHotel)
 - En personalDeApoyo no se agrega el atributo **numeroDeHabitacionesACargo**, puesto que es un atributo calculado.
 - Análogamente a lo realizado con los multivaluados en el servicio de comida. Se crean dos tablas: *telefonoRecepcionista* y *correoRecepcionista*.
14. telefonoRecepcionista(idPersona, telefonoRecepcionista)
15. correoRecepcionista(idPersona, correoRecepcionista)
16. personalDeApoyo(idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, fechaContratación, horarioInicio, HorarioFin, numeroInterior, numeroExterior, colonia, calle, estado, idHotel)
 - Nuevamente crean dos tablas: *telefonoPersonalApoyo* y *correoPersonalApoyo*.
17. telefonoPersonalApoyo(idPersona, telefonoPersonalApoyo)
18. correoPersonalApoyo(idPersona, correoPersonalApoyo)
19. limpiezaInterna(idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, fechaContratación, horarioInicio, HorarioFin, numeroInterior, numeroExterior, colonia, calle, estado, idHotel)

- Nuevamente crean dos tablas: telefonoLimpiezaInterna y correoLimpiezaInterna.

20. telefonoLimpiezaInterna(idPersona, telefonoLimpiezaInterna)

21. correoLimpiezaInterna(idPersona, correoLimpiezaInterna)

22. evento(idEvento, numMembresia, nombreSalon, numAsistentes, fechaEvento, horaInicio, horaFin, idPersona, idSalon)

- En evento no se agrega el atributo NombreHuesped, puesto que es un atributo calculado.

23. salon(idSalon, horarioInicio, HorarioFin, nombreSalon, capacidad, numeroInterior, numeroExterior, colonia, calle, estado)

24. encuesta(idEncuesta, servicioRecibido, limpieza, recomendación, comentario, volverHospedar, idPersona)

- Para el caso de encuesta se dejan los nombres de los atributos cortos, esto para mayor facilidad. Además observamos que recibe los atributos de la relación al ir de uno a muchos.

25. membresia(numeroMembresia, activa, idPersona)

- De la entidad FormaDePago, se genera una especialización total con disyunción. Por lo que se crean las siguientes tablas: EFECTIVO, TARJETA. Éstas heredan las características de la entidad padre, pero la entidad padre no aparece.

26. efectivo(idFormaEfectivo)

27. tarjeta(idFormaTarjeta, numeroTarjeta, vencimiento)

28. servicioExtra(idServicio, numeroPlatos, idPago)

- No añadimos el atributo Precio, puesto que se trata de un atributo calculado.

29. pago(idPago, numeroNinos, numeroAdultos, numeroMascotas, numeroNoches)

4.2. Relaciones que generan tabla; Muchos a muchos (N:M)

Las relaciones (N:M) se convierten en una tabla, los atributos que la conforman se forman con el identificador de cada una de las entidades que relaciona junto con los atributos de la relación (si existen).

1. disponerIndividual(idHabitacion, idHotel)
2. disponerDoble(idHabitacion, idHotel)
3. disponerCuadruple(idHabitacion, idHotel)
4. disponerPenthouse(idHabitacion, idHotel)
 - Las tablas de DISPONER resultan de la especialización con disyunción. Ambos atributos son llaves foráneas provenientes del hotel y la habitación.
5. atender(idEvento, idPersona)
 - Ambos atributos son llaves foráneas provenientes del evento y el personal de apoyo responsable de atenderlo.
6. rentarDoble(idPersona, idHabitacion)
7. rentarIndividual(idPersona, idHabitacion)
8. rentarCuadruple(idPersona, idHabitacion)
9. rentarPenthouse(idPersona, idHabitacion)
 - Las tablas de RENTAR resultan de la especialización con disyunción. Ambos atributos son llaves foráneas provenientes del huésped y la habitación.

4.3. Relaciones que modifican; Uno a Muchos (1:N)

Los atributos que se añadan se remarcan con color rojo en la sección 4.1.

1. TENER: En este caso, en la tabla PERSONA se debe incluir la llave de la tabla HOTEL más los atributos de la relación. Sin embargo, esta tabla no existe, pues ésta generó una especialización total con disyunción. Entonces, la llave de la tabla hotel debe ser añadida a: huesped, servicioComida, recepcionista, personalDeApoyo, y limpiezaInterna.

2. RESERVAR: En la tabla EVENTO se añade la llave de la tabla huesped, más los atributos de la relación, a saber; fechaEvento, horaInicio, horaFin.
3. AGREGAR: En la tabla servicioExtra, se añade la llave de la tabla pago, más los atributos de la relación AGREGAR, sin embargo, ésta no cuenta con alguno.
4. RESPONDER: La llave de huesped idPersona se agrega a la tabla encuesta, más los atributos de la relación AGREGAR, sin embargo, ésta no cuenta con alguno.
5. SERRESPONSABLE: En este caso, en la tablas generadas por la especialización de habitacion (individual, doble, cuádruple Y penthouse) agregamos la llave de personalDeApoyo herencia de persona. Considerando que no hay atributos es la relación, eso es todo.

4.4. Relaciones Uno a Uno (1:1)

Los atributos que se añadan se remarcen con color azul en la sección 4.1.

1. ELEGIR: Es una relación total de un lado por lo que en la tabla huesped se añade únicamente la llave de efectivo y tarjeta (idFormaEfectivo e idFormaTarjeta), sin más por el momento ya que la relación no tiene atributos.
2. OBTENER: Nuevamente es una relación total de un lado, de manera que a la tabla membresia se añade la llave de huesped idPersona, atributo llave heredado de persona. Adicionalmente, el atributo de la relación no se agrega puesto que se trata de un atributo calculado.
3. realizarPago(idPersona, nombre, paterno, materno, fechaNacimiento, genero, nacionalidad, numeroNinos, numeroAdultos, numeroMascotas, numeroNoches, idPago)
 - Es una relación total por ambos lados, así que se incluyen los atributos de huesped, REALIZAR Y pago con sus respectivas llaves foráneas.
4. VALIDAR: Es una relación total de un lado y puesto que la relación no tiene atributos únicamente se añadirá la llave de salon idSalon a evento.

4.5. Dominio de atributos y restricciones

- | | |
|-------------------------------|--|
| ■ numeroInterior: VARCHAR(10) | ■ numeroHabitacionesOcupadas: INTEGER |
| ■ numeroExterior: INTEGER | |
| ■ colonia: VARCHAR(60) | ■ nombreEstablecimiento: VARCHAR(50) |
| ■ calle: VARCHAR(70) | ■ telefono: INTEGER |
| ■ estado: VARCHAR(50) | ■ numeroHabitacionesDisponibles: INTEGER |

GER

- numerototaldehabitaciones: INTEGER
- idHotel: VARCHAR(20)
- servicio: VARCHAR(100)
- petfriendly: VARCHAR(2)
- horarioInicio: TIME
- horarioFin: TIME
- idEmpleado: VARCHAR(20)
- fechaContratacion: DATE
- RFC: VARCHAR(13)
- horaInicio: TIME
- horaFin: TIME
- numeroHabitacionesaCargo: INTEGER
- idHabitacion: VARCHAR(10)
- disponibilidad: VARCHAR(13)
- costoPorNoche: FLOAT
- numeroDeCamas: INTEGER
- numeroHabitacion: INTEGER
- fechaNacimiento: DATE
- correo (cualquiera de los correos): VARCHAR(50)
- genero: VARCHAR(1)
- idPersona: VARCHAR(20)
- apellidoPaterno: VARCHAR(50)
- apellidoMaterno: VARCHAR(50)
- nombre: VARCHAR(50)
- nacionalidad: VARCHAR(50)
- fechaEvento: DATE
- horaInicio: TIME
- horaFin: TIME
- nombreSalon: VARCHAR(50)
- numAsistentes: INTEGER
- nombreHuesped: VARCHAR(50)
- numMembresía: INTEGER
- idEvento: VARCHAR(15)
- capacidad: INTEGER
- idSalon: VARCHAR(10)
- nombreSalon: VARCHAR(50)
- recomendacion: INTEGER
- comentario: VARCHAR(150)
- volverHospedar: INTEGER
- idEncuesta: VARCHAR(10)
- servicioRecibido: INTEGER
- limpieza: INTEGER
- idServicio: VARCHAR(10)
- numeroPlatos: INTEGER
- numeroNinos: INTEGER
- numeroAdultos: INTEGER
- numeroMascotas: INTEGER
- numeroNoches: INTEGER
- idPago: VARCHAR(15)
- idForma: VARCHAR(8)
- numeroTarjeta: INTEGER
- fechaVencimiento: DATE
- activa: VARCHAR(2)

4.6. Restricciones

- No habrá valores de tipo NULL.
- La evaluación en las columnas Recomendacion, servicioRecibido, Limpieza y VolverHospedar será de tipo entero entre 1 a 5.
- Para el atributo numeroInterior se permitirán valores enteros o 'SN' para el caso donde no se tenga el dato.
- El atributo Genero solo permitirá los valores 'M' (mujer) o 'H' (hombre).
- petfriendly y activa solo permitirá los valores 'SI' Y 'NO'.
- En variables de tipo entero o flotante solo se recibirán valores mayores o iguales a cero.

4.7. Dependencias funcionales:

- RFC \rightarrow nombre, fecha de nacimiento
- Número de visitas al año \rightarrow Activa
- Capacidad del salón \rightarrow Número de asistentes en el evento

5. Restricciones de Dominio

De manera general, se asegura que los atributos no sean vacíos o de tipo null.

Además, para los teléfonos, se garantiza que se reciban 10 dígitos entre 0 y 9.

Otras consideraciones incluyen: la verificación de formatos específicos como el RFC que debe seguir la estructura usual de 13 dígitos alfanuméricos; la validación de campos booleanos que deben ser estrictamente TRUE o FALSE; la restricción de valores permitidos en ciertos campos como el género, que solo puede ser 'M' o 'H', y el campo petFriendly, que solo puede aceptar 'SI' o 'NO'.

Además, se aplican reglas de positividad a campos numéricos como el número exterior de las direcciones y el costo por noche de las habitaciones.

También se establecen valores por defecto para ciertos campos, como 'SN' para el número interior. Estas reglas aseguran la integridad, consistencia y precisión de los datos ingresados en el sistema.

■ Restricciones de dominio para la Tabla Hotel

1. **hotel_nombre_no_vacio** : Se asegura que el nombre del establecimiento no sea la cadena vacía `CHECK(nombreEstablecimiento <> '')`;
2. **hotel_horaInicio_no_null** : Se asegura que la hora de check-in no sea NULL `CHECK(horaInicio IS NOT NULL)`;
3. **hotel_horaFin_no_null** : Se asegura que la hora de check-out no sea NULL `CHECK(horaFin IS NOT NULL)`;

4. **hotel_petfriendly_no_vacio** : Se asegura que petFriendly no sea la cadena vacía
CHECK(petFriendly <>"");
 5. **petfriendly_valores_permitidos** : Se asegura que la columna petFriendly solo acepte los valores SI y NO CHECK (petFriendly IN ('SI', 'NO'));
 6. **hotel_servicio_no_vacio** : Se asegura que servicio no sea la cadena vacía
CHECK(servicio <>"");
 7. **hotel_numeroInterior_no_vacio** : Se asegura que numeroInterior no sea la cadena vacía CHECK(numeroInterior <>"");
 8. **hotel_numeroExterior_no_null** : Se asegura que numeroExterior no sea NULL
CHECK(numeroExterior IS NOT NULL);
 9. **hotel_colonia_no_vacia** : Se asegura que la colonia no sea la cadena vacía CHECK(colonia <>"");
 10. **hotel_calle_no_vacia** : Se asegura que la calle no sea la cadena vacía CHECK(calle <>"");
 11. **hotel_estado_no_vacio** : Se asegura que el estado no sea la cadena vacía CHECK(estado <>"");
 12. **numeroInterior** – Se escribe 'SN' para el caso donde no se tenga el dato
SET DEFAULT('SN');
ALTER TABLE hotel ADD CONSTRAINT numeroExterior_check
CHECK(numeroExterior > 0);– Se asegura de que el número exterior solo reciba valores positivos
- **Restricciones de dominio para la Tabla telefonoHotel**
 1. **telefonoHotel_no_vacio** : para garantizar que el campo telefonoHotel no esté vacío
CHECK (telefonoHotel <>"");
 2. **telefono_Hotel_valido**: Se asegura de recibir 10 dígitos entre 0 y 9 cada uno.
 - **Restricciones de dominio para las Tabla telefono con los tipos de personal**
 1. **telefonoTipodePersonal_no_vacio** : para garantizar que el campo telefonoPersona no esté vacío. CHECK (telefonoPersona <>"");
 2. **telefonoPersona_valido**: Se asegura de recibir 10 dígitos entre 0 y 9 cada uno.
 - **Restricciones de dominio para la Tabla Correo Persona**
 1. **correoPersona_no_vacio** : Restricción CHECK para garantizar que el campo correoElectronico no esté vacío. CHECK (correoElectronico <>"");
 - **Restricciones de dominio para la Tabla Empleado**
 1. Estas columnas **RFC ,nombre, paterno, materno, fechaNacimiento, genero, horario, numeroInterior, numeroExterior, colonia, calle, estado, fechaContratacion** no pueden tener valor NULL por lo que agregaremos la instruccion Set NOT NULL.

2. **genero_valido** : para asegurar que el género sea 'M' o 'H'
3. **numeroInterior** : Se escribe 'SN' para el caso donde no se tenga el dato
SET DEFAULT('SN');
4. **RFC_check**:Se asegura de que los 13 dígitos del RFC coincidan con la estructura usual. CHECK(RFC '[A-Z]4[0-9]6[A-Z]2[0-9]1');
5. **genero_valido**: para asegurar que el género sea 'M' o 'H'. CHECK (genero IN ('M', 'H'));
6. **numeroExterior_check**:Se asegura de que el número exterior solo reciba valores positivos.CHECK(numeroExterior >0);

■ **Restricciones de dominio para la Tabla Efectivo**

1. **idFormaEfectivo** : La columna idForma no puede ser null por lo que agregamos la instruccion
SET NOT NULL

■ **Restricciones de dominio para la Tabla Huesped**

1. Estas columnas **nombre, paterno, materno, fechaNacimiento, genero, idForma** no pueden tener valor null, por lo que agregamos la instruccion SET NOT NULL.
2. **genero_valido**: Restricción CHECK para asegurar que el género sea 'M' o 'H'.
CHECK (genero IN ('M', 'H'));

■ **Restricciones de dominio para la Tabla TelefonoHuesped**

1. **telefonoPersona_no_vacio** : para garantizar que el campo telefonoPersona no esté vacío.CHECK (telefonoPersona <>'');;
2. **telefonoHuesped_valido**: Se asegura de recibir 10 dígitos entre 0 y 9 cada uno.

■ **Restricciones de dominio para la Tabla CorreoHuesped**

1. **CorreoPersona_no_vacio** : para garantizar que el campo correoElectronico no esté vacío. CHECK (correoElectronico <>'');

■ **Restricciones de dominio de la Tabla Habitacion**

1. Estas columnas **idPersona, RFC, disponibilidad, costoPorNoche, numeroDeHabitacion, numeroDeCamas** no pueden tener valor nulo, por lo que agregamos la instruccion SET NOT NULL.
2. **disponibilidad_check** : Restricción para asegurar que disponibilidad sea TRUE o FALSE. CHECK (disponibilidad IN (TRUE, FALSE));;
3. **costoPorNoche_positivo_check**: Restricción para asegurar que costoPorNoche sea positivo. CHECK (costoPorNoche >= 0);
4. **numeroDeHabitacion_positivo_check**: Restricción para asegurar que numeroDeHabitacion sea positivo.CHECK (numeroDeHabitacion >0);

■ Restricciones de dominio de la Tabla Doble

1. Estas columnas **idHabitacion**, **idHuesped**, **RFC**, **disponibilidad**, **costoPorNoche**, **numeroDeHabitacion**, **numeroDeCamas** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
2. **disponibilidad_check_doble** : Restricción para asegurar que disponibilidad sea TRUE o FALSE. CHECK (disponibilidad IN (TRUE, FALSE));
3. **costoPorNoche_positivo_check_doble** : Restricción para asegurar que costoPorNoche sea positivo. CHECK (costoPorNoche \geq 0);
4. **numeroDeHabitacion_positivo_check_doble** : Restricción para asegurar que numeroDeHabitacion sea positivo. CHECK (numeroDeHabitacion $>$ 0);
5. **numeroDeCamas_doble** : Restricción para asegurar que numeroDeCamas sea mayor o igual a 1. CHECK (numeroDeCamas \geq 1);

■ Restricciones de dominio de la Tabla Cuadruple

1. Estas columnas **idPersona**, **RFC**, **disponibilidad**, **costoPorNoche**, **numeroDeHabitacion**, **numeroDeCamas** no pueden tener valor nulo, por lo que agregamos la instruccion SET NOT NULL.
2. **disponibilidad_check_doble** : Restricción para asegurar que disponibilidad sea TRUE o FALSE. CHECK (disponibilidad IN (TRUE, FALSE));
3. **costoPorNoche_positivo_check_doble** : Restricción para asegurar que costoPorNoche sea positivo. CHECK (costoPorNoche \geq 0);
4. **numeroDeHabitacion_positivo_check_doble** : Restricción para asegurar que numeroDeHabitacion sea positivo. CHECK (numeroDeHabitacion $>$ 0);
5. **numeroDeCamas_doble** : Restricción para asegurar que numeroDeCamas sea mayor o igual a 2. CHECK (numeroDeCamas \geq 2);

■ Restricciones de dominio de la Tabla PentHouse

1. Estas columnas **idPersona**, **RFC**, **disponibilidad**, **costoPorNoche**, **numeroDeHabitacion**, **numeroDeCamas** no pueden tener valor nulo, por lo que agregamos la instruccion SET NOT NULL.
2. **disponibilidad_check_doble** : Restricción para asegurar que disponibilidad sea TRUE o FALSE. CHECK (disponibilidad IN (TRUE, FALSE));
3. **costoPorNoche_positivo_check_doble** : Restricción para asegurar que costoPorNoche sea positivo. CHECK (costoPorNoche \geq 0);
4. **numeroDeHabitacion_positivo_check_doble** : Restricción para asegurar que numeroDeHabitacion sea positivo. CHECK (numeroDeHabitacion $>$ 0);
5. **numeroDeCamas_doble** : Restricción para asegurar que numeroDeCamas sea mayor o igual a 2. CHECK (numeroDeCamas \geq 2);

■ Restricciones de dominio de la Tabla ServicioComida

1. Estas columnas **nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, numeroExterior, colonia, calle, estado, idHotel** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
2. **numeroInterior**: se escribe "SN" donde no se tenga el dato. SET DEFAULT('SN');
3. **RFC_check**: Se asegura de que los 13 dígitos del RFC coincidan con la estructura usual. CHECK(RFC '[A-Z]4[0-9]6[A-Z]2[0-9]1');
4. **genero_valido**: para asegurar que el género sea 'M' o 'H'. CHECK (genero IN ('M', 'H'));
5. **numeroExterior_check**: Se asegura de que el número exterior solo reciba valores positivos. CHECK(numeroExterior >0);

■ Restricciones de dominio de la Tabla Recepcionista

1. Estas columnas **nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, numeroExterior, colonia, calle, estado, idHotel** no pueden tener valor nulo, por lo que agregamos la instruccion SET NOT NULL.
2. **numeroInterior**: se escribe "SN". en caso de que no se tenga el dato. SET DEFAULT('SN')
3. **RFC_check**: Se asegura de que los 13 dígitos del RFC coincidan con la estructura usual. CHECK(RFC '[A-Z]4[0-9]6[A-Z]2[0-9]1');
4. **genero_valido**: para asegurar que el género sea 'M' o 'H'. CHECK (genero IN ('M', 'H'));
5. **numeroExterior_check**: Se asegura de que el número exterior solo reciba valores positivos. CHECK(numeroExterior >0);

■ Restricciones de dominio de la Tabla PersonalDeApoyo

1. Estas columnas **nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, numeroExterior, colonia, calle, estado, idHotel** no pueden tener valor nulo, por lo que agregamos la instruccion SET NOT NULL.
2. **numeroInterior**: se escribe "SN". en caso de que no se tenga el dato. SET DEFAULT('SN')
3. **RFC_check**: Se asegura de que los 13 dígitos del RFC coincidan con la estructura usual. CHECK(RFC '[A-Z]4[0-9]6[A-Z]2[0-9]1');
4. **genero_valido**: para asegurar que el género sea 'M' o 'H'. CHECK (genero IN ('M', 'H'));
5. **numeroExterior_check**: Se asegura de que el número exterior solo reciba valores positivos. CHECK(numeroExterior >0);

■ Restricciones de dominio de la Tabla LimpiezaInterna

1. Estas columnas **nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, genero, RFC, numeroExterior, colonia, calle, estado, idHotel** no pueden tener valor nulo, por lo que agregamos la instruccion SET NOT NULL.

2. **numeroInterior:** se escribe "SN". en caso de que no se tenga el dato. SET DEFAULT('SN')
3. **RFC_check:**Se asegura de que los 13 dígitos del RFC coincidan con la estructura usual. CHECK(RFC '[A-Z]4[0-9]6[A-Z]2[0-9]1');
4. **genero_valido:** para asegurar que el género sea 'M' o 'H'. CHECK (genero IN ('M', 'H'));
5. **numeroExterior_check:**Se asegura de que el número exterior solo reciba valores positivos. CHECK(numeroExterior >0);

■ Restricciones de dominio de la Tabla Salon

1. Estas columnas **idSalon, horario, nombreSalon, capacidad, numeroExterior, colonia, calle, estado** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
2. **capacidad_check:**Se asegura de que la capacidad solo reciba valores positivos. CHECK(capacidad >0);
3. **numeroExterior_check:**Se asegura de que el número exterior solo reciba valores positivos. CHECK(numeroExterior >0);

■ Restricciones de dominio de la Tabla Evento

1. Estas columnas **idEvento, nombreSalon, numAsistentes, fechaEvento, horaInicio, horaFin, idSalon** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
2. **numAsistentes_check:** Se asegura de que el número de asistentes solo reciba valores positivos. CHECK(numAsistentes >0);

■ Restricciones de dominio de la Tabla Encuesta

1. Estas columnas **idEncuesta, servicioRecibido, limpieza, recomendacion, volverHospedar, idPersona** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
2. **check_servicioRecibido:** Restricción de evaluación del servicio recibido. CHECK (servicioRecibido BETWEEN 1 AND 5);
3. **check_limpieza:** Restricción de evaluación de la limpieza. CHECK (limpieza BETWEEN 1 AND 5);
4. **check_recomendacion:** Restricción de evaluación de la recomendación. CHECK (recomendacion BETWEEN 1 AND 5);
5. **check_volverHospedar:** Restricción de evaluación de volver a hospedarse. CHECK (volverHospedar BETWEEN 1 AND 5);
6. **comentario:** Valor predeterminado para comentario. SET DEFAULT '';

■ Restricciones de dominio de la Tabla membresia

1. Estas columnas **numeroMembresia, activa, idPersona** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
 2. **activa_check** : restriccion para activa (Solo 'SI' o 'NO')
CHECK (activa IN ('SI', 'NO'));
- **Restricciones de dominio de la Tabla Encuesta**
 1. Estas columnas **idForma, numeroTarjeta, vencimiento** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
 2. **numerotarjeta_valido**: Se asegura de recibir 16 dígitos entre 0 y 9 cada uno.
 - **Restricciones de dominio de la tabla Pago**
 1. Estas columnas **idPago, NumeroNinos, NumeroAdultos, NumeroMascotas, NumeroNoches** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
 2. **NumeroNinos_check**: Se asegura de que el número de niños solo reciba valores positivos. CHECK(NumeroNinos >0);
 3. **NumeroAdultos_check**: Se asegura de que el número de adultos solo reciba valores positivos. CHECK(NumeroAdultos >0);
 4. **NumeroMascotas_check**: Se asegura de que el número de mascotas solo reciba valores positivos. CHECK(NumeroMascotas >0);
 5. **NumeroNoches_check**: Se asegura de que el número de noches solo reciba valores positivos. CHECK(NumeroNoches >0);
 - **Restricciones de dominio de la Tabla ServicioExtra**
 1. Estas columnas **idServicio, numeroPlatos, idPago** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
 2. **numeroPlatos_check**: Se asegura de que el número de platos solo reciba valores mayores o iguales a cero. CHECK(numeroPlatos >= 0);
 - **Restricciones de dominio de la Tabla Disponer**
 1. Estas columnas **IdHabitacion, IdHotel** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
 - **Restricciones de dominio de las Tabla Atender para las distintas habitaciones**
 1. Estas columnas **IdEvento, IdPersona** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.
 - **Restricciones de dominio de las Tabla Rentar para las distintas habitaciones**
 1. Estas columnas **IdPersona** no pueden tener valor nulo, por lo que, agregamos la instruccion SET NOT NULL.

6. Decisiones de diseño para el DDL

- Se decidió utilizar **CASCADE** como política de mantenimiento de llaves foráneas, puesto que garantiza que las relaciones entre tablas se mantengan consistentes automáticamente, minimizando errores humanos que podrían surgir al tratar de mantener manualmente estas relaciones.

Para el caso de uso correspondiente, en el archivo DDL, se utilizó: **ON DELETE CASCADE** y **ON UPDATE CASCADE** en todas las llaves foráneas de la base de datos, con esto se asegura que cualquier eliminación o actualización en una tabla principal se propagará automáticamente a las tablas relacionadas.

- Se reorganizó el orden de las tablas en el documento DDL para asegurar la creación correcta de todas ellas y mantener sus integridades. Además, se añadieron algunas correcciones al final del documento DDL para garantizar la ejecución adecuada del script. El orden es el siguiente:

- | | |
|-----------------------------|------------------------|
| 1. hotel | 21. doble |
| 2. telefonoHotel | 22. cuadruple |
| 3. efectivo | 23. penthouse |
| 4. tarjeta | 24. salon |
| 5. huesped | 25. evento |
| 6. telefonoHuesped | 26. encuesta |
| 7. correoHuesped | 27. membresia |
| 8. personaldDeApoyo | 28. pago |
| 9. servicioComida | 29. servicioExtra |
| 10. Recepcionista | 30. disponerIndividual |
| 11. limpiezaInterna | 31. disponerDoble |
| 12. telefonoServicioComida | 32. disponerCuadruple |
| 13. telefonoRecepcionista | 33. disponerPenthouse |
| 14. telefonoPersonalApoyo | 34. atender |
| 15. telefonoLimpiezaInterna | 35. rentarIndividual |
| 16. correoPersonalApoyo | 36. rentarDoble |
| 17. correoServicioComida | 37. rentarCuadruple |
| 18. correoRecepcionista | 38. rentarPenthouse |
| 19. correoLimpiezaInterna | 39. realizarPago |
| 20. individual | |

7. Poblamiento de la Base de Datos

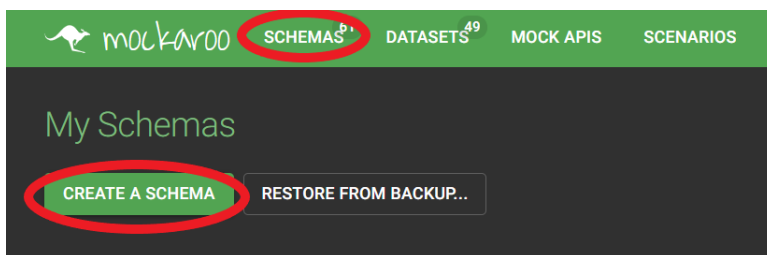
Durante la población de la base de datos se utilizó la página: <https://www.mockaroo.com/>. A continuación se explicarán los puntos más importantes del proceso. Ejemplificaremos dos casos: el primero es el poblamiento de la tabla **hotel**, la cual no cuenta con llaves foráneas, y el segundo caso, la tabla **telefonoHotel** la cual sí tiene.

7.1. Tabla hotel

Primero evaluamos los campos a considerar, el tipo de datos y sus respectivos CONSTRAINT.

```
-- Tabla hotel
CREATE TABLE hotel (
  idHotel VARCHAR(20),
  nombreEstablecimiento VARCHAR(50),
  horaCheckIn TIME,
  horaCheckOut TIME,
  petFriendly CHAR(2),
  servicio VARCHAR(100),
  numeroInterior VARCHAR(10),
  numeroExterior INT,
  colonia VARCHAR(60),
  calle VARCHAR(70),
  estado VARCHAR(50)
);
```

Luego, volvimos a la página de Mockaroo y seleccionamos en la parte superior izquierda **SCHEMAS** seguida de **CREATE A SCHEMA**.



Esto nos llevó a la siguiente página, donde creamos los campos necesarios para cada elemento de la tabla. Observemos que se tomó el nombre idéntico de cada columna, además de que el tipo de dato se eligió considerando su compatibilidad con el dominio del atributo en la tabla.

En algunos casos, fue necesario señalar más especificaciones, tal es el caso del **numeroInterior** que no tiene una restricción de nulidad o el tipo *Regular Expression* de **nombreEstablecimiento** y **petFriendly**, en este último señalamos las distintas respuesta posibles en el apartado *Options*.

Field Name	Type	Options
idHotel	Row Number	blank: 0 % Σ X
nombreEstablecimien	Regular Expression	(Hotel Luna) blank: 0 % Σ X
horaCheckIn	Time	from: 3:00 PM to: 3:00 PM format: 24 Hour blank: 0 % Σ X
horaCheckOut	Time	from: 1:00 PM to: 1:00 PM format: 24 Hour blank: 0 % Σ X
petFriendly	Regular Expression	(SI NO) blank: 0 % Σ X
servicio	Regular Expression	(Lavanderia, Bar, Salones de eventos, Gimnasio Bar, Salones de eventos, Tienda de regalos, G blank: 0 % Σ X
numeroInterior	Street Number	blank: 50 % Σ X
numeroExterior	Street Number	blank: 0 % Σ X
colonia	Plant Common Name	blank: 0 % Σ X
calle	Street Name	blank: 0 % Σ X
estado	State	Coahuila De Zaragoza Campeche Veracruz Llave Chiapas Puebla Chihuahua Aragon Baja California Guanajuato Nuevo Leon Morelos Tabasco Oaxaca Tamaulipas All Countries blank: 0 % Σ X

Una vez que rellenamos todos los campos procedimos a elegir el número de filas y generamos el archivo en formato CSV.

+ ADD ANOTHER FIELD
GENERATE FIELDS USING AI...

Rows: 200
Format: CSV
Line Ending: Windows (CRLF)
Include: ☒ header ☐ BOM

Si bien pudo generarse el archivo SQL directamente se eligió de esta manera para guardar todos los datos en formato CSV (ya que por las especificaciones dadas mockaroo genera datos nuevos cada vez).

Finalmente, se cargó el formato CSV creado en el dataset de Mockaroo y se utilizó en un nuevo schema:

Field Name	Type	Options					
idHotel	Dataset Column	hotel	idHotel	sequential	blank: 0 %	Σ	×
nombreEstablecimiento	Dataset Column	hotel	nombreEstablecimiento	sequential	blank: 0 %	Σ	×
horaCheckIn	Dataset Column	hotel	horaCheckIn	sequential	blank: 0 %	Σ	×
horaCheckOut	Dataset Column	hotel	horaCheckOut	sequential	blank: 0 %	Σ	×
petFriendly	Dataset Column	hotel	petFriendly	sequential	blank: 0 %	Σ	×
servicio	Dataset Column	hotel	servicio	sequential	blank: 0 %	Σ	×
numeroInterior	Dataset Column	hotel	numeroInterior	sequential	blank: 50 %	Σ	×
numeroExterior	Dataset Column	hotel	numeroExterior	sequential	blank: 0 %	Σ	×
colonia	Dataset Column	hotel	colonia	sequential	blank: 0 %	Σ	×
calle	Dataset Column	hotel	calle	sequential	blank: 0 %	Σ	×
estado	Dataset Column	hotel	estado	sequential	blank: 0 %	Σ	×

Notemos que se debe crear el archivo utilizando el nombre de la tabla utilizado en el DDL para evitar problemas en la ejecución del código al cargarlo en nuestra base de datos.

Rows: 200 Format: SQL Table Name: hotel ☐ include CREATE TABLE

7.2. Tabla telefonoHotel

El procedimiento realizado fue análogo al anterior con la única excepción de que en la creación de los datos para el archivo csv se utilizó el dataset de la tabla hotel para reutilizar lo idHotel generados que telefonoHotel utiliza como llave foránea.

Field Name	Type	Options					
idHotel	Dataset Column	hotel	idHotel	random	blank: 0 %	Σ	×
telefonoHotel	Digit Sequence	55#####			blank: 0 %	Σ	×

8. Consultas

```

1  -- consulta 1: Hoteles que sean petFriendly.
2  SELECT * FROM hotel
3  WHERE petFriendly='SI';

```

	idhotel [PK] character varying (20)	nombrestablecimiento character varying (50)	horacheckin time without time zone	horacheckout time without time zone	petfriendly character	servicio character varying (100)
1	HL2	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
2	HL3	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...
3	HL4	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
4	HL5	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
5	HL6	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
6	HL7	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...
7	HL8	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
8	HL11	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
9	HL13	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...
10	HL15	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
11	HL16	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
12	HL17	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
13	HL18	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
14	HL20	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
15	HL23	Hotel Luna	15:00:00	13:00:00	SI	Lavandería, Bar, Salones de eventos, Gimnasio
16	HL30	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...

Figura 1: Hoteles que son petFriendly.

```

5  -- Consulta 2: Habitaciones a cargo (idHabitacion) por miembro del personal de apoyo. La consulta está ordenada por apellido paterno.
6  SELECT p.idPersona, p.nombre, p.apellidoPaterno, p.apellidoMaterno, i.idHabitacion AS habitaciones_individuales_a_cargo,
7         d.idHabitacion AS habitaciones_dobles_a_cargo, cu.idHabitacion AS habitaciones_cuadriples_a_cargo,
8         pt.idHabitacion AS habitaciones_penthouse_a_cargo
9  FROM personalDeApoyo p
10 JOIN individual i ON p.idPersona = i.idPersona
11 JOIN doble d ON p.idPersona = d.idPersona
12 JOIN cuadruple cu ON p.idPersona = cu.idPersona
13 JOIN penthouse pt ON p.idPersona = pt.idPersona
14 GROUP BY p.idPersona, i.idHabitacion, d.idHabitacion, cu.idHabitacion, pt.idHabitacion
15 ORDER BY p.apellidoPaterno;

```

	idpersona character varying (20)	nombre character varying (100)	apellidoPaterno character varying (100)	apellidoMaterno character varying (100)	habitaciones_individuales_a_cargo character varying (10)	habitaciones_dobles_a_cargo character varying (10)	habitaciones_cuad character varying (10)
15	AP117	Aida	Arnaudon	Beatey	IND28	DBL163	CUA62
16	AP117	Aida	Arnaudon	Beatey	IND28	DBL163	CUA143
17	AP117	Aida	Arnaudon	Beatey	IND28	DBL122	CUA143
18	AP117	Aida	Arnaudon	Beatey	IND28	DBL163	CUA62
19	AP117	Aida	Arnaudon	Beatey	IND28	DBL69	CUA62
20	AP117	Aida	Arnaudon	Beatey	IND28	DBL122	CUA143
21	AP56	Rosalie	Bankhurst	Broszkiewicz	IND14	DBL211	CUA211
22	AP56	Rosalie	Bankhurst	Broszkiewicz	IND14	DBL211	CUA65
23	AP56	Rosalie	Bankhurst	Broszkiewicz	IND133	DBL211	CUA65
24	AP56	Rosalie	Bankhurst	Broszkiewicz	IND133	DBL211	CUA211

Figura 2: Habitaciones a cargo (idHabitacion) por miembro del personal de apoyo. La consulta está ordenada por apellido paterno.

```
17 -- Consulta 3: Huéspedes que han reservado un evento más de una vez, ya que podemos interpretar de ello que son clientes a
18 -- quienes les agradó el servicio y probablemente vuelvan a hacerlo.
19 SELECT h.*
20 FROM huesped h
21 JOIN evento e ON h.idPersona = e.idPersona
22 GROUP BY h.idPersona
23 HAVING COUNT(*) > 1;
```

	idpersona [PK] character varying (20)	idhotel character varying (20)	nombre character varying (100)	paterno character varying (100)	materno character varying (100)	fechanacimiento date	genero character	nacionalidad character varying (100)
1	H286	HL112	Brig	Fetherston	Seneschal	1965-07-28	M	Tlingit-Haida
2	H53	HL155	Allan	Paulsson	Bianco	1958-12-11	M	Malaysian
3	H56	HL37	Ronda	Henlon	Martinovsky	1991-09-04	H	Colombian
4	H6	HL111	Halsey	Clancey	Waby	1996-03-12	H	Asian Indian
5	H197	HL166	Jock	Kerslake	McGivena	1980-09-13	H	Micronesian
6	H105	HL172	Ynes	Espinheira	Jull	1955-01-27	H	Polynesian
7	H91	HL168	Annabella	Bannester	Larmor	2002-04-03	H	Japanese
8	H288	HL165	Malia	Willgoss	Willavoys	1968-11-26	H	Potawatomi
9	H174	HL93	Lynnet	Piddlehinton	Ruter	1977-03-24	H	Seminole
10	H72	HL171	Gawain	Letessier	Spickett	1951-11-11	M	Salvadoran
11	H199	HL20	Steve	Fidian	Breadmore	1990-09-06	M	Japanese
12	H260	HL122	Steffie	De la Zenne	Proffitt	1980-06-16	H	Ecuadorian
13	H32	HL98	Xerxes	Matisoff	McPharlain	1987-11-23	H	Chickasaw
14	H238	HL92	Monte	Weich	Grimwade	1951-12-09	M	Pima

Figura 3: Huéspedes que han reservado un evento más de una vez, ya que podemos interpretar de ello que son clientes a quienes les agradó el servicio y probablemente vuelvan a hacerlo.

```

25 -- Consulta 4: Empleados que cumplen años el mes próximo por orden de día (incluye el hotel en que trabajan).
26 SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel
27 FROM (
28     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel
29     FROM recepcionista
30     UNION ALL
31     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel
32     FROM personalDeApoyo
33     UNION ALL
34     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel
35     FROM servicioComida
36     UNION ALL
37     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel
38     FROM limpiezaInterna
39 ) AS empleados
40 WHERE EXTRACT(MONTH FROM fechaNacimiento) = 7
41 ORDER BY EXTRACT(DAY FROM fechaNacimiento);

```

Data Output Messages Graph Visualiser X Notifications

	idpersona character varying (20)	nombre character varying (100)	apellidoPaterno character varying (100)	apellidoMaterno character varying (100)	fechanacimiento date	idhotel character varying (20)
9	SC97	Marvin	Baggally	Kernock	1960-07-06	HL153
10	AP93	Marylin	Cobb	Wilshaw	1979-07-07	HL147
11	SC164	Vin	Dobrowolski	Hammerton	2001-07-08	HL166
12	R184	Octavius	Gascoyen	Magenny	1998-07-08	HL19
13	R199	Myranda	Jiroutek	Myford	1953-07-08	HL26
14	AP44	Bill	Ollier	Skip	1965-07-08	HL100
15	AP85	Hildagard	Stambridge	Brabban	1972-07-08	HL187

Total rows: 58 of 58 Query complete 00:00:00.271

Figura 4: Empleados que cumplen años el mes próximo por orden de día (incluye el hotel en que trabajan).

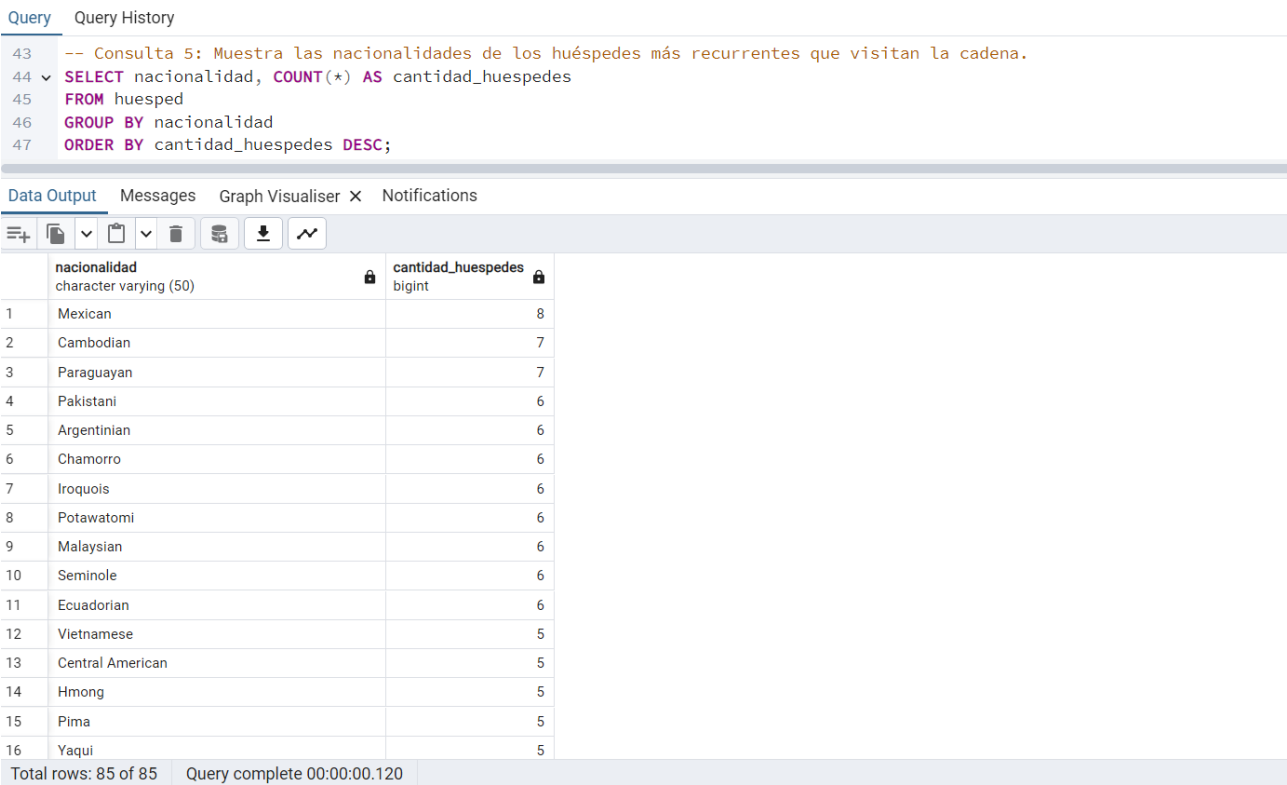


Figura 5: Muestra las nacionalidades de los huéspedes más recurrentes que visitan la cadena.

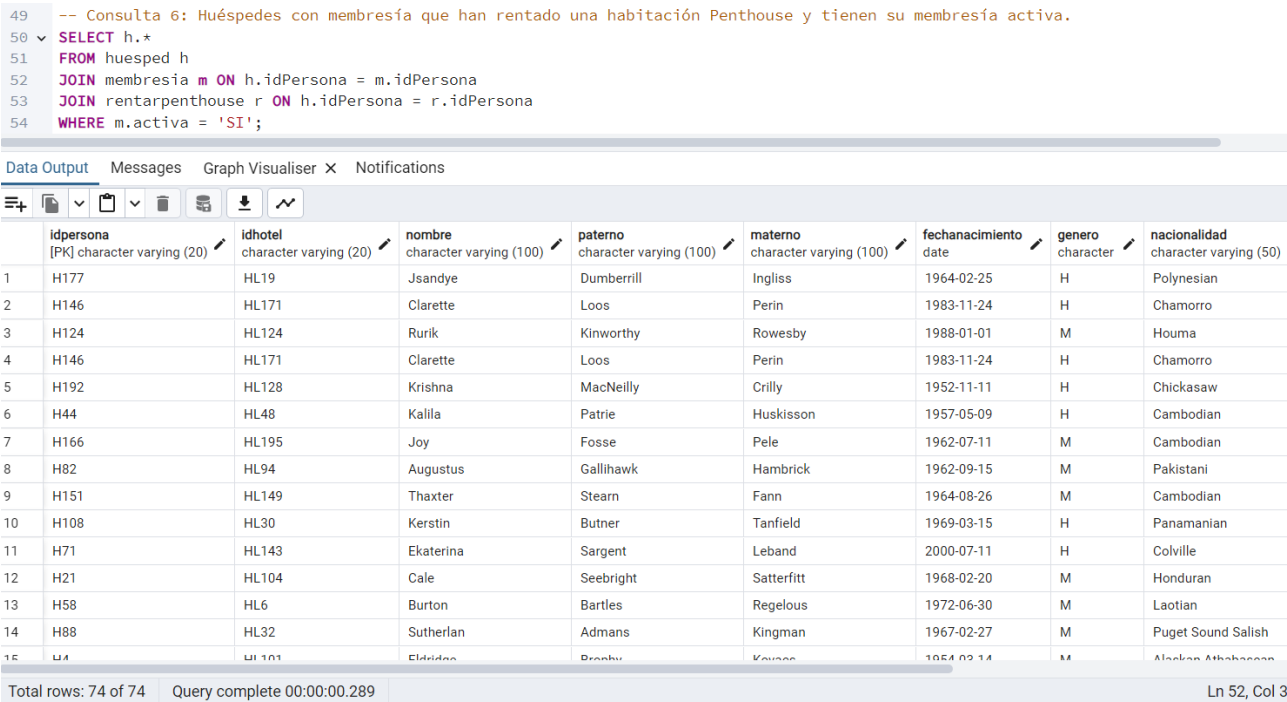


Figura 6: Huéspedes con membresía que han rentado una habitación Penthouse y tienen su membresía activa.

```

56 -- Consulta 7: Los 20 empleados con mayor antigüedad y a qué hotel pertenecen (para darles un agradecimiento
57 -- por su prolongada confianza).
58 v SELECT e.*, h.nombreEstablecimiento
59 FROM (
60     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel, fechaContratacion
61     FROM recepcionista
62     UNION ALL
63     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel, fechaContratacion
64     FROM personalDeApoyo
65     UNION ALL
66     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel, fechaContratacion
67     FROM servicioComida
68     UNION ALL
69     SELECT idPersona, nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento, idHotel, fechaContratacion
70     FROM limpiezaInterna
71 ) AS e
72 JOIN hotel h ON e.idHotel = h.idHotel
73 ORDER BY e.fechaContratacion ASC
74 LIMIT 20;

```

	idpersona character varying (20)	nombre character varying (100)	apellidoPaterno character varying (100)	apellidoMaterno character varying (100)	fechanacimiento date	idhotel character varying (20)	fechacontratacion date	nombrestableci character varying
17	AP102	Doe	Hendron	Torra	1992-06-11	HL43	1990-06-14	Hotel Luna
18	R77	Bronnie	Jacke	Giberd	1973-03-12	HL150	1990-06-18	Hotel Luna
19	R161	Kaitlin	Alty	Clewley	1963-11-30	HL163	1990-06-23	Hotel Luna
20	R25	Tildy	Gammett	McKearnen	1995-08-08	HL110	1990-06-29	Hotel Luna

Figura 7: Los 20 empleados con mayor antigüedad y a qué hotel pertenecen (para darles un agradecimiento – por su prolongada confianza).

```

76 -- Consulta 8: Habitaciones con costo mayor a 4500 por tipo de habitacion.
77 v SELECT idHabitacion, costopornoche
78 FROM (
79     SELECT idHabitacion, costopornoche FROM doble
80     UNION ALL
81     SELECT idHabitacion, costopornoche FROM cuadruple
82     UNION ALL
83     SELECT idHabitacion, costopornoche FROM individual
84     UNION ALL
85     SELECT idHabitacion, costopornoche FROM penthouse
86 ) AS habitaciones_costosas
87 WHERE costopornoche > 4500;

```

	idhabitacion character varying (10)	costopornoche double precision
1	IND252	5199
2	IND253	4999
3	PTH1	4955
4	PTH5	4547
5	PTH7	4730
6	PTH12	4816
7	PTH14	4668
8	PTH15	4781
9	PTH17	4647
10	PTH22	4780
11	PTH24	4844

Total rows: 71 of 71 Query complete 00:00:00.110

Figura 8: Habitaciones con costo mayor a 4500 por tipo de habitacion.

```

89 -- Consulta 9: Salones con capacidad mayor a 100 que han tenido eventos este año
90 SELECT s.*
91 FROM salon s
92 JOIN evento e ON s.idSalon = e.idSalon
93 WHERE s.capacidad > 100 AND EXTRACT(YEAR FROM e.fechaEvento) = EXTRACT(YEAR FROM CURRENT_DATE);

```

	idsalon [PK] character varying (20)	horarioinicio time without time zone	horariofin time without time zone	nombresalon character varying (100)	capacidad integer	numerointerior character varying (10)	numeroexterior integer	colonia character varying (60)
1	SALON19	09:21:01	12:00:00	Crimson	258	[null]	322	Red Rosemallow
2	SALON25	08:40:28	12:00:00	Pink	137	[null]	83	Parry's Thistle
3	SALON27	08:25:36	12:00:00	Teal	239	[null]	75	Mt. Hamilton Tickseed
4	SALON60	08:30:18	12:00:00	Blue	135	[null]	37	Shasta Snakeroot
5	SALON76	09:24:59	12:00:00	Indigo	280	16	5027	Alaskan Glacier Butter
6	SALON113	08:08:26	12:00:00	Red	231	[null]	6112	Woodland Mirrorplant
7	SALON123	08:23:06	12:00:00	Goldenrod	413	81	256	Slender Grama
8	SALON135	08:31:36	12:00:00	Puce	198	[null]	8	Bog Willowherb
9	SALON148	09:09:58	12:00:00	Khaki	383	41	3639	Jepson's Blue Wildrye
10	SALON180	09:42:58	12:00:00	Goldenrod	282	50	49819	Coastal Goosefoot
11	SALON184	08:14:23	12:00:00	Pink	434	25	85	Santa Rita Mountain Ti
12	SALON185	09:05:53	12:00:00	Green	302	48	1	Pearly Everlasting
13	SALON187	09:22:56	12:00:00	Green	106	[null]	1	Flavoparmelia Lichen
14	SALON195	08:45:31	12:00:00	Puce	217	[null]	7	Mourning Milkvetch
15	SALON196	09:16:41	12:00:00	Puce	141	[null]	34	Wedgeleaf Draba

Total rows: 20 of 20 Query complete 00:00:00.247 Ln 93, Col 9

Figura 9: Salones con capacidad mayor a 100 que han tenido eventos este año.

```

95 -- Consulta 10: Huespedes con alta probabilidad de volver a hospedarse que tengan mascotas.
96 SELECT h.idPersona, h.nombre, h.paterno, h.materno, r.numeroMascotas
97 FROM huesped h
98 JOIN realizarPago r ON h.idPersona = r.idPersona
99 JOIN encuesta en ON h.idPersona = en.idPersona
100 WHERE en.volverHospedar = 5 AND r.numeroMascotas > 0;

```

	idpersona character varying (20)	nombre character varying (100)	paterno character varying (100)	materno character varying (100)	numeromascotas integer
1	H1	Culver	Winscomb	Dorning	2
2	H2	Cad	MacGaffey	Giovannardi	2
3	H10	Tarah	Relph	Auletta	2
4	H22	Sandy	Dinwoodie	Pottage	1
5	H38	Jonis	Hearst	Large	1
6	H39	Ronnica	Edeson	Hannaby	1
7	H41	Filippa	Hurring	Flatt	2
8	H52	Cesya	Whitewood	Reisen	2
9	H54	Corabella	Plowell	Scales	2
10	H58	Burton	Bartles	Regelous	1
11	H84	Melisa	Stonman	Vodden	1
12	H99	Francesca	Zamora	Dunderdale	2
13	H101	Jo-ann	Lammerich	Hurry	2
14	H110	Candace	Jacks	Kingham	1
15	H111	Kassey	Knowlman	Shales	1

Total rows: 33 of 33 Query complete 00:00:00.111

Figura 10: Huéspedes con alta probabilidad de volver a hospedarse que tengan mascotas.


```

105 -- Consulta 11: La información de hoteles cuyo precio de habitación individual sea mayor a 250. Deberan ordenar la
106 -- información a partir del precio.
107
108 v SELECT h.*, i.costoPorNoche
109 FROM hotel h
110 JOIN disponerIndividual di ON h.idHotel = di.idHotel
111 JOIN individual i ON di.idHabitacion = i.idHabitacion
112 WHERE i.costoPorNoche > 250
113 ORDER BY i.costoPorNoche;

```

Data Output Messages Graph Visualiser x Notifications						
	idhotel character varying (20)	nombreestablecimiento character varying (50)	horacheckin time without time zone	horacheckout time without time zone	petfriendly character	servicio character varying (100)
1	HL46	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
2	HL3	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...
3	HL110	Hotel Luna	15:00:00	13:00:00	NO	Lavandería, Bar, Salones de eventos, Gimnasio
4	HL21	Hotel Luna	15:00:00	13:00:00	NO	Lavandería, Bar, Salones de eventos, Gimnasio
5	HL89	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
6	HL176	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...
7	HL175	Hotel Luna	15:00:00	13:00:00	NO	Bar, Salones de eventos, Tienda de regalos, Gimnasio
8	HL103	Hotel Luna	15:00:00	13:00:00	SI	Bar, Salones de eventos, Tienda de regalos, Gimnasio
9	HL184	Hotel Luna	15:00:00	13:00:00	NO	Bar, Salones de eventos, Tienda de regalos, Gimnasio
10	HL178	Hotel Luna	15:00:00	13:00:00	NO	Salones de eventos, Gimnasio, salón de belleza, pisci...
11	HL115	Hotel Luna	15:00:00	13:00:00	SI	Salones de eventos, Gimnasio, salón de belleza, pisci...
12	HL144	Hotel Luna	15:00:00	13:00:00	NO	Salones de eventos, Gimnasio, salón de belleza, pisci...

Total rows: 206 of 206 Query complete 00:00:00.387

Figura 11: La información de hoteles cuyo precio de habitación individual sea mayor a 250. Deberán ordenar la información a partir del precio.

```

115 -- Consulta 12: Huespedes con membresia que hayan organizado un evento entre las fechas 01 de junio del 2023 al 31 de
116 -- marzo del 2024 (el último año).
117 |
118 v SELECT e.idPersona, h.nombre, h.paterno, h. materno FROM evento e
119 JOIN huesped h ON e.idPersona=h.idPersona
120 JOIN membresia mem ON h.idPersona=mem.idPersona
121 WHERE fechaEvento BETWEEN '2023-06-01' AND '2024-05-31' AND activa LIKE 'SI';

```

	idpersona character varying (50)	nombre character varying (100)	paterno character varying (100)	materno character varying (100)
1	H2	Cad	MacGaffey	Giovannardi
2	H4	Eldridge	Brophy	Kovacs
3	H6	Halsey	Clancey	Waby
4	H8	Matilde	Pigram	Morch
5	H16	Leandra	Pass	Raffles
6	H20	Toni	Drejer	Niven
7	H20	Toni	Drejer	Niven
8	H21	Cale	Seebright	Satterfitt
9	H28	Brigham	Gregore	Stranio
10	H39	Ronnica	Edeson	Hannaby
11	H79	Hali	Werrilow	Vaugham
12	H88	Sutherland	Admans	Kingman
13	H91	Annabella	Bannester	Larmor
14	H92	Weylin	Celes	Gummory

Total rows: 21 of 21 Query complete 00:00:00.111

Figura 12: Huéspedes con membresía que hayan organizado un evento entre las fechas 01 de junio del 2023 al 31 de marzo del 2024 (el último año).

```

123 -- Consulta 13: Los recepcionistas y personal de comida que vivan en el mismo estado y que no se encuentren trabajando
124 -- en el mismo hotel.
125
126 ✓ SELECT DISTINCT r.idPersona, r.nombre, r.apellidoMaterno, r.apellidoPaterno, r.estado, r.idHotel,
127      sc.idPersona, sc.nombre, sc.apellidoMaterno, sc.apellidoPaterno, sc.estado, sc.idHotel
128 FROM recepcionista r
129 JOIN servicioComida sc ON r.estado = sc.estado
130 WHERE r.idHotel != sc.idHotel;

```

	idpersona character varying (20)	nombre character varying (100)	apellidomaterno character varying (100)	apellidopaterno character varying (100)	estado character varying (50)	idhotel character varying (20)	idpersona character varying (20)
1	R86	Elizabeth	Ghiardielli	Dahlback	Veracruz Llave	HL16	SC47
2	R71	Jaime	Gaitung	Stroulger	Tamaulipas	HL148	SC95
3	R159	Dedie	Domel	Sambells	Veracruz Llave	HL197	SC176
4	R50	Gar	Lait	Switzer	Chiapas	HL26	SC106
5	R124	Terza	Semrad	Pidgeley	Veracruz Llave	HL1	SC108
6	R141	Corey	Wohlers	Maypole	Oaxaca	HL194	SC94
7	R169	Rice	Lob	Hall	Veracruz Llave	HL84	SC178
8	R194	Gaile	Frankling	Moulster	Veracruz Llave	HL121	SC159
9	R20	Florenza	Zarfati	McCreath	Veracruz Llave	HL45	SC116
10	R155	Olivie	Shailer	Sotworth	Tabasco	HL168	SC53
11	R161	Kaitlin	Clewley	Alty	Veracruz Llave	HL163	SC143
12	R139	Demetri	Trimmill	Ivanchenkov	Chiapas	HL90	SC197
13	R75	Lorie	Meakin	Tuxill	Chiapas	HL18	SC186

Total rows: 1000 of 5244 Query complete 00:00:00.315

Figura 13: Los recepcionistas y personal de comida que vivan en el mismo estado y que no se encuentren trabajando en el mismo hotel.

```

132 -- Consulta 14: Huespedes con mascotas y que se hayan alojado en una habitación Penthouse.
133 ✓ SELECT DISTINCT h.idPersona, rph.idHabitacion, h.nombre, h.paterno, h.materno, rp.numeroMascotas
134 FROM huesped h
135 JOIN realizarPago rp ON h.idPersona = rp.idPersona
136 JOIN rentarPenthouse rph ON h.idPersona = rph.idPersona
137 WHERE rp.numeroMascotas > 0;

```

	idpersona character varying (20)	idhabitacion character varying (20)	nombre character varying (100)	paterno character varying (100)	materno character varying (100)	numeromascotas integer
1	H124	PTH167	Rurik	Kinworthy	Rowesby	1
2	H4	PTH236	Eldridge	Brophy	Kovacs	1
3	H132	PTH81	Karole	Lochet	Graveson	1
4	H15	PTH117	Enid	Lorain	Merriton	2
5	H55	PTH81	Beverlie	Muddiman	Ruckhard	2
6	H204	PTH106	Rahel	Slott	Shorto	2
7	H33	PTH97	Case	Frany	Daily	2
8	H112	PTH76	Burk	Hissie	Jennery	2
9	H160	PTH114	Beale	Fyldes	MacCostigan	1
10	H246	PTH113	Kali	Belsey	Garratt	1
11	H156	PTH166	Alfie	Giacobillo	Baggett	2
12	H177	PTH67	Jsandye	Dumberrill	Ingliss	1
13	H163	PTH33	Trent	Legon	Desforges	1
14	H15	PTH157	Enid	Lorain	Merriton	2
15	H136	PTH243	Shaw	Bartolacci	Custy	1

Total rows: 119 of 119 Query complete 00:00:00.163

Figura 14: Huéspedes con mascotas y que se hayan alojado en una habitación Penthouse.

```

140 -- Consulta 15: El personal de limpieza y el personal de apoyo, que compartan estado y que se encuentren
141 -- trabajando en el mismo hotel. La información está ordenada a partir del estado.
142
143 SELECT DISTINCT l.idPersona AS ID_Limpieza, l.nombre AS Nombre_Limpieza, l.apellidoPaterno AS Apellido_Paterno_Limpieza, l.idHotel AS ID_Hotel,
144 a.idPersona AS ID_PersonalDeApoyo, a.nombre AS Nombre_PersonalDeApoyo, a.apellidoPaterno AS Apellido_Paterno_PersonalDeApoyo, l.estado
145 FROM limpiezaInterna l
146 JOIN personalDeApoyo a ON l.idHotel = a.idHotel AND l.estado = a.estado
147 ORDER BY l.apellidoPaterno;

```

	id_limpieza character varying (20)	nombre_limpieza character varying (100)	apellido_paterno_limpieza character varying (100)	hotel character varying (20)	id_personaldeapoyo character varying (20)	nombre_personaldeapoyo character varying (100)	apellido_paterno_personaldeapoyo character varying (100)
1	LI43	Giordano	Aleksandrkin	HL38	AP138	Brandi	Rollason
2	LI147	Felecia	Ambroix	HL186	AP51	Chrystel	Wass
3	LI120	Michaeline	Andretti	HL163	AP142	Brandais	Ryland
4	LI5	Kaleb	Bozworth	HL7	AP155	Karl	Bozworth
5	LI163	Nadya	Braim	HL167	AP54	Marty	Buckles
6	LI66	Rutledge	Bruno	HL86	AP153	Azul	Lowless
7	LI36	Romain	Burman	HL191	AP11	Brett	Hess
8	LI51	Phyllys	Comelli	HL155	AP24	Herschel	Sokill
9	LI49	Bette	Coulling	HL131	AP117	Aida	Arnaudon
10	LI12	Lennard	De Angelis	HL7	AP155	Karl	Bozworth
11	LI161	Baird	Dossett	HL48	AP109	Felipe	Mulliner
12	LI198	Johnath	Dovydenas	HL161	AP55	Stevana	Onthank
13	LI122	Aurie	Exeter	HL86	AP153	Azul	Lowless

Total rows: 32 of 32 Query complete 00:00:00.101 Ln 148, Col

Figura 15: El personal de limpieza y el personal de apoyo, que compartan estado y que se encuentren trabajando en el mismo hotel. La información está ordenada a partir del estado.

Query Query History

```

149 -- Consulta extra: Personal de la cadena ordenado por hotel.
150 SELECT e.idHotel, e.nombre, e.apellidoPaterno, e.apellidoMaterno, e.Puesto
151 FROM (
152     SELECT h.nombreEstablecimiento, p.nombre, p.apellidoPaterno, p.apellidoMaterno, p.idHotel, 'repcionista' AS Puesto
153     FROM hotel h
154     JOIN recepcionista p ON h.idHotel = p.idHotel
155     UNION ALL
156     SELECT h.nombreEstablecimiento, p.nombre, p.apellidoPaterno, p.apellidoMaterno, p.idHotel, 'personalDeApoyo'
157     FROM hotel h
158     JOIN personalDeApoyo p ON h.idHotel = p.idHotel
159     UNION ALL
160     SELECT h.nombreEstablecimiento, p.nombre, p.apellidoPaterno, p.apellidoMaterno, p.idHotel, 'servicioComida'
161     FROM hotel h
162     JOIN servicioComida p ON h.idHotel = p.idHotel
163     UNION ALL
164     SELECT h.nombreEstablecimiento, p.nombre, p.apellidoPaterno, p.apellidoMaterno, p.idHotel, 'limpiezaInterna'
165     FROM hotel h
166     JOIN limpiezaInterna p ON h.idHotel = p.idHotel
167 ) AS e
168 ORDER BY e.idHotel, e.apellidoPaterno, e.apellidoMaterno, e.nombre;

```

	idhotel character varying (20)	nombre character varying (100)	apellidoPaterno character varying (100)	apellidoMaterno character varying (100)	puesto text
753	HL99	Rivi	Bordiss	Tichelaar	repcionista
754	HL99	Madlin	Hegerty	Rickeard	limpiezaInterna
755	HL99	Katharyn	Segeswoeth	Pedro	repcionista
756	HL99	Layne	Verne	Arnoud	servicioComida

Total rows: 756 of 756 Query complete 00:00:00.348

Figura 16: Personal de la cadena ordenado por hotel.

9. Conclusiones

Al analizar las consultas realizadas en la base de datos de la cadena hotelera, se pueden extraer varias conclusiones importantes.

Se observa que algunos hoteles de la cadena son **pet-friendly**, lo que indica una política de inclusión de mascotas que podría atraer a un segmento específico de clientes. Además, el hecho de asignar habitaciones específicas a diferentes miembros del personal de apoyo sugiere una estructura organizativa bien definida y una distribución equitativa de responsabilidades.

También se identifica una tendencia entre los huéspedes que reservan eventos en más de una ocasión, lo que sugiere un alto nivel de satisfacción con los servicios de eventos ofrecidos por la cadena hotelera. Esto puede interpretarse como una señal de 'lealtad' o confianza por parte de estos clientes, lo que los convierte en un segmento valioso para el negocio.

Asimismo, el reconocimiento de empleados que cumplen años el próximo mes puede fortalecer la moral y el sentido de pertenencia dentro del equipo de trabajo. Esto podría contribuir a mejorar el ambiente laboral y fomentar una cultura de aprecio mutuo en la empresa.

Otro aspecto relevante es la diversidad de nacionalidades entre los huéspedes más recurrentes, lo que sugiere un atractivo internacional de la cadena hotelera. Identificar estas nacionalidades principales podría ayudar a adaptar los servicios y las estrategias de marketing para satisfacer las necesidades específicas de estos grupos de clientes.

En conjunto, las consultas proporcionan una visión amplia y detallada de diversos aspectos operativos y de mercado de la cadena hotelera, lo que puede servir de base para la toma de decisiones estratégicas y operativas, si es que así se requiera en un lapso futuro.