**UNIVERSIDAD DE CALDAS**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA EN SISTEMAS Y COMPUTACIÓN**

**PROYECTO BASES DE DATOS I**

**Integrantes:**

Oivar Micolta Cuero

Diego Alejandro Tobón Henao

1. **Universo del discurso**

Se desea desarrollar una base de datos para el hotel “Hotel El Paraíso” que permita gestionar la información de las siguientes entidades: habitaciones, reservas de habitaciones, empleados, clientes que llegan hospedarse en el hotel, servicios que ofrece el hotel, productos y proveedores. También se gestionar la información de los pagos que realiza el cliente por el alojamiento y los servicios que consume.

En el hotel trabajan muchos empleados y se desea conocer información de cada uno de ellos, tales como: id (número de documento), nombre, apellido, teléfono, correo, salario. Un empleado realiza un servicio a un cliente específico.

También se debe administrar la información de los clientes que se alojan en el hotel. De cada cliente se debe conocer su id, nombre, apellido y teléfono.

Las habitaciones deben tener un id, numero de habitación, estado (“Disponible” u “Ocupada”) y tarifa. Se debe permitir hacer reservas de las habitaciones disponibles, por lo cual se desea conocer la información de la reserva. De cada reserva se desea conocer su id, estado (“Pendiente” o “Confirmada”) id del cliente, id del empleado, fecha de entrada y fecha de salida, habitación y tarifa. Cada habitación pertenece a una categoría que está conformada por id y nombre.

Los clientes pueden solicitar un servicio a sus respectivas habitaciones, cada servicio debe tener asignado un id, nombre, numero de la habitación, id del cliente, productos, cantidad del producto y costo del servicio. Un cliente puede solicitar muchos servicios, pero un servicio es asignado a un solo cliente.

Entre los servicios que ofrece el hotel, también se ofrecen muchos productos para los clientes, de los cuales se desea conocer su id, nombre y precio.

1. **Análisis de requerimientos:**

Para la base de datos se necesita lo siguiente:

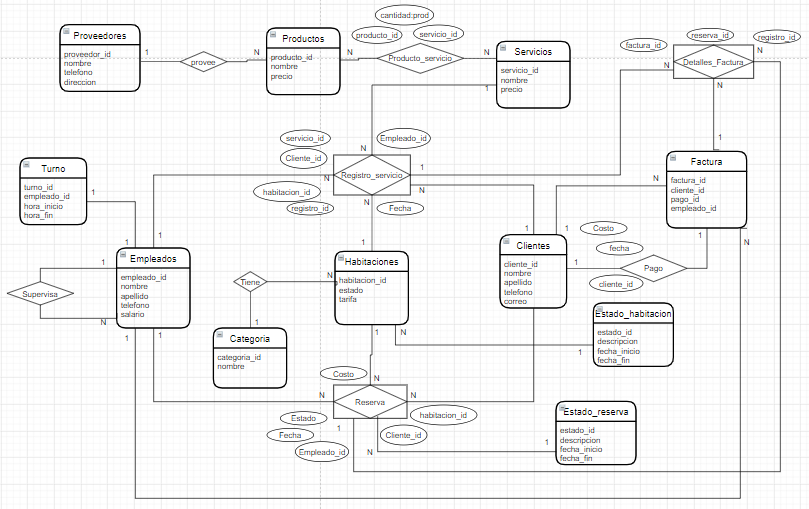
Para la reserva de habitaciones se necesitan que la tabla cliente se relacione con las tablas habitación y empleado y habrá una agregación entre ellas que se llamará reserva que luego en el modelo relacional se convertirá en una tabla. Para conocer la disponibilidad de una habitación en una fecha determinada se crea una tabla estado\_habitación que permite hacer esta verificación. Además, habitación estará relacionada con la tabla categoría para conocer a que categoría pertenece dicha habitación.

Un empleado realiza un servicio a un cliente determinado, para ello se necesitan las tablas empleado, cliente y servicio que estarán relacionadas y por ser una relación ternaria se formará una agregación llamada registro\_servicio que luego se convertirá en una tabla, además esta nueva tabla estará relacionada con habitación para saber a dónde se debe dirigir el servicio. Estos servicios pueden o no incluir uno o varios productos, por tanto, la tabla servicios debe relacionarse con productos, pero entre estas dos tablas genera una relación de n a n por lo se hace necesario que esta relación se convierta en una nueva tabla que contendrá los id de cada una. También, la tabla productos estará relacionada con proveedores para conocer quién provee los productos.

La tabla empleados estará relacionada con la tabla turnos, ya que cada empleado tendrá asignado un turno de trabajo. Además, un empleado puede ser supervisor de otro u otros empleados, por lo cual habrá una relación reflexiva hacia la misma.

Finalmente, se necesitará saber qué debe pagar el cliente en total, tanto del alquiler de la habitación y de un o varios servicios registrados. Para ello, el cliente debe tener asociado un pago que a su vez se relaciona con una factura. Así, en la factura estarán registrados los datos del pago, del cliente y del empleado que lo atendió. La factura debe tener unos detalles más especificados, por lo cual habrá otra tabla llamada detalles\_factura, que tendrá id de la factura y que además estará relacionada con los registros de los servicios. También tendrá una relación con la reserva. De esta manera se podrá calcular el valor total que debe pagar el cliente por su estadía y por los servicios solicitados.

1. **Diagrama Entidad Relación:**



1. **Modelo relacional:**

**Factura**(pk: factura\_id, fk: cliente\_id, fk: pago\_id, fk: empleado\_id)

**Detalles\_factura**(pk: factura\_id, fk: reserva\_id, fk: registro\_id)

**Pago**(pk: pago\_id, fk: cliente\_id, fk: registro\_id, fecha, costo)

**Proveedores**(pk: proveedor\_id, nombre, teléfono, dirección)

**Productos**(pk: producto\_id, nombre, precio, fk: proveedor\_id)

**Servicios**(pk: servicio\_id, nombre, precio)

**Producto\_servicio**(producto\_id, fk: servicio\_id, cantidad\_producto)

**Registro\_servicio**(pk: registro\_id, fk: empleado\_id, fk: cliente\_id, fk: habitación\_id, servicio\_id, fecha)

**Empleado**(pk: empleado\_id nombre, apellido, teléfono, salario, fk: supervisor\_id)

**Turnos**(pk: turno\_id, hora\_inicio, hora\_fin)

Turno\_empleado()

**Clientes**(pk: cliente\_id, nombre, apellido, teléfono)

**Categoria**(pk: categoría\_id, nombre)

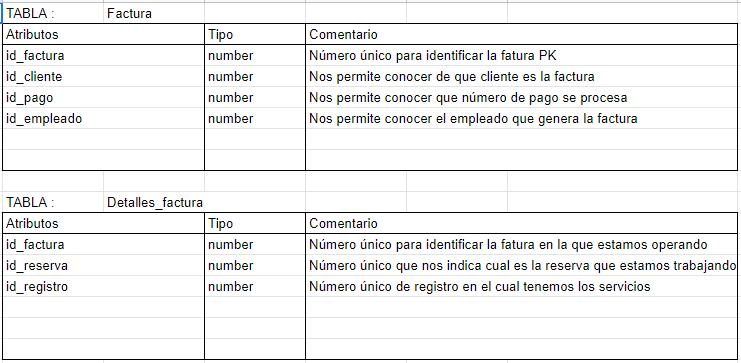
**Habitacion**(pk: habitación\_id, fk: estado\_id, fk: categoría\_id, tarifa)

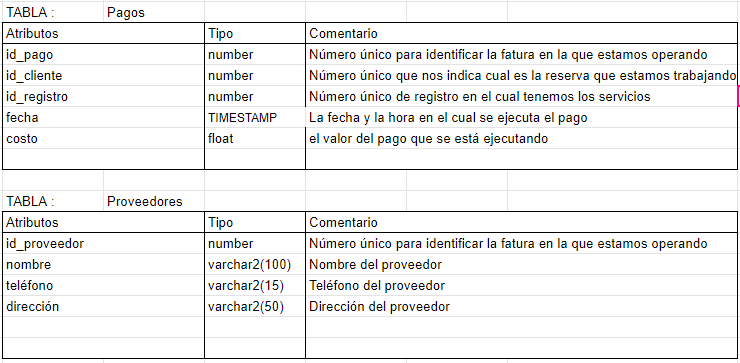
**Estado\_habitacion**(pk: estado\_id, descripción, fecha\_inicio, fecha\_fin)

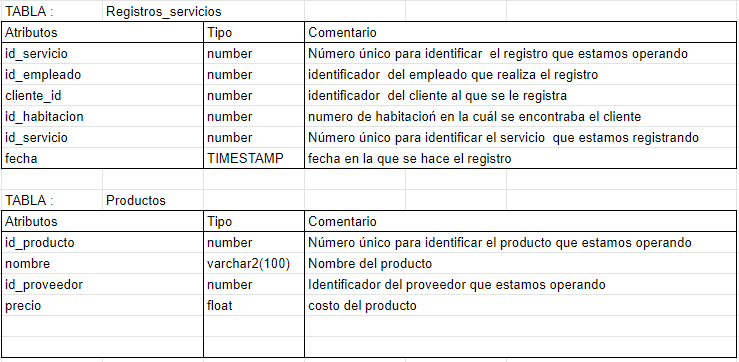
**Reserva**(PK: reserva\_id, fk: estado\_id, fk: habitación\_id, fk: empleado\_id, fk: cliente\_id; fk: habitación\_id, fecha, costo)

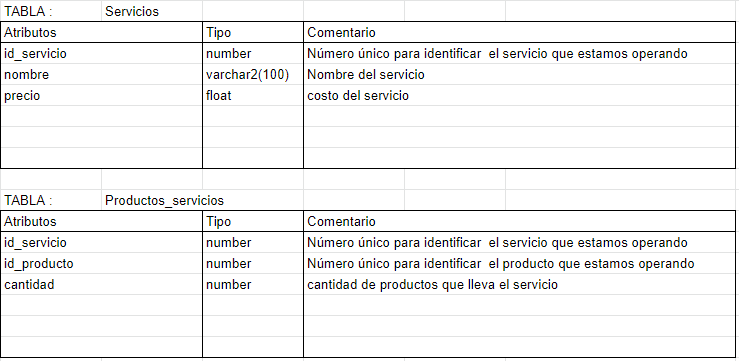
**Estado\_reserva**(pk: fk: estado\_id, descripción, fecha\_inicio, fecha\_fin)

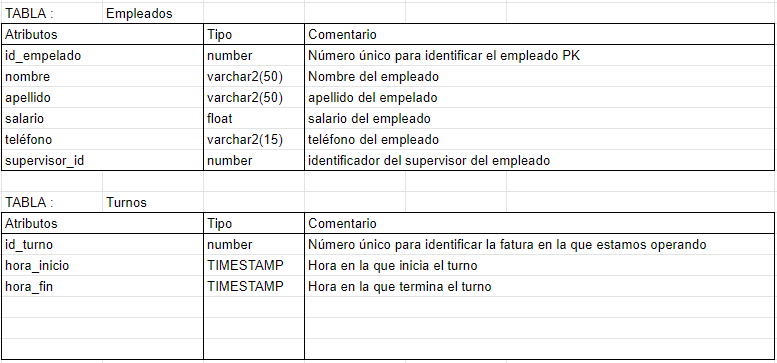
1. **Diccionario de Datos**

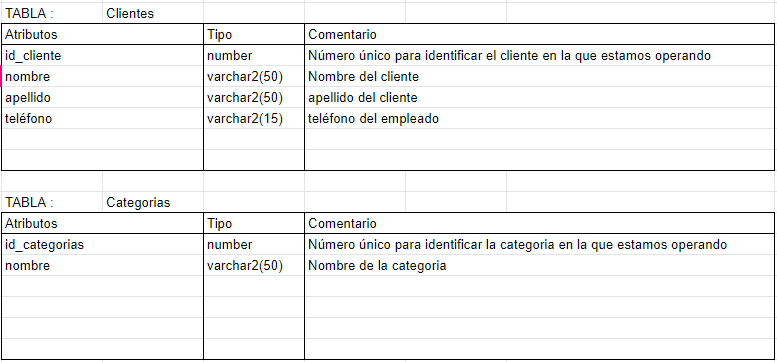


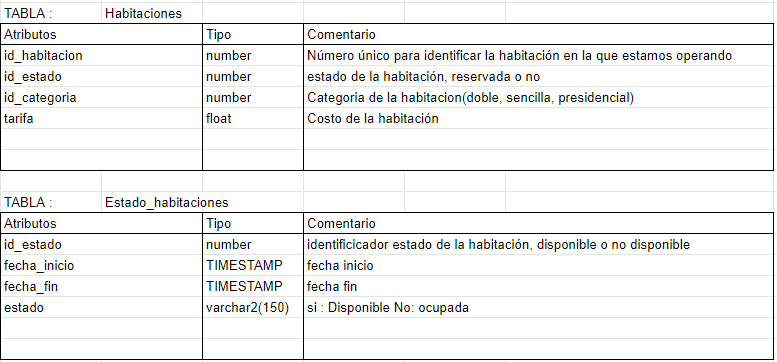


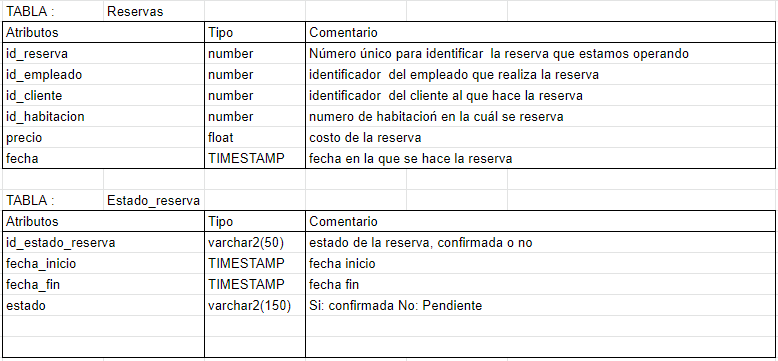












**Consultas SQL:**

**--Servicio mas solicitado en la habitacion mas reservada**

SELECT DISTINCT s.descripcion AS servicio, h.habitacion\_id

FROM servicios s JOIN registro\_servicio rs ON s.servicio\_id = rs.servicio\_id

JOIN habitacion h ON rs.habitacion\_id = h.habitacion\_id

WHERE h.habitacion\_id = (SELECT habitacion\_id

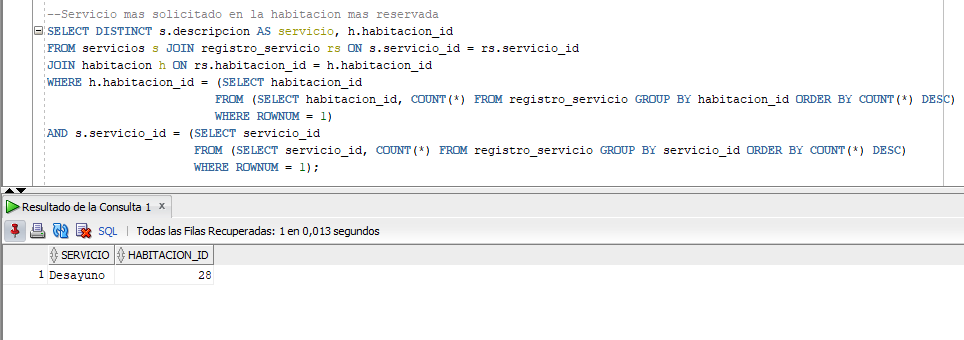
FROM (SELECT habitacion\_id, COUNT(\*) FROM registro\_servicio GROUP BY habitacion\_id ORDER BY COUNT(\*) DESC)

WHERE ROWNUM = 1)

AND s.servicio\_id = (SELECT servicio\_id

FROM (SELECT servicio\_id, COUNT(\*) FROM registro\_servicio GROUP BY servicio\_id ORDER BY COUNT(\*) DESC)

WHERE ROWNUM = 1)



**--Nombre y salario del empleado que prestó el servicio más solicitado**

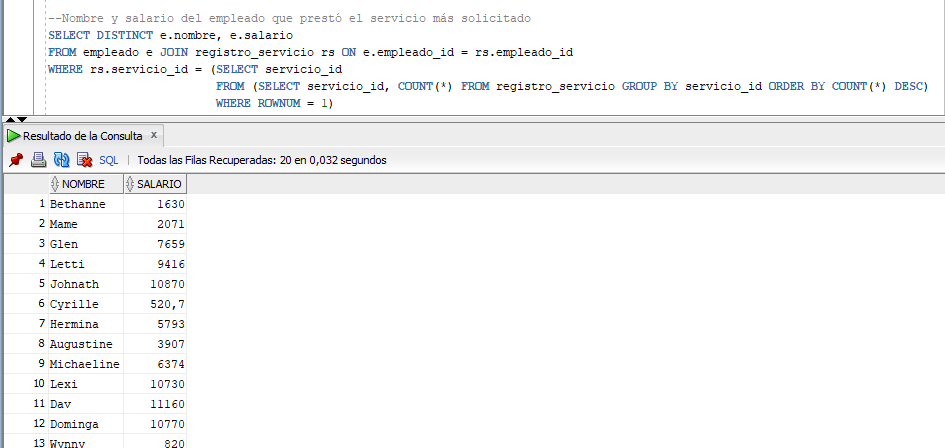
SELECT DISTINCT e.nombre, e.salario

FROM empleado e JOIN registro\_servicio rs ON e.empleado\_id = rs.empleado\_id

WHERE rs.servicio\_id = (SELECT servicio\_id

FROM (SELECT servicio\_id, COUNT(\*) FROM registro\_servicio GROUP BY servicio\_id ORDER BY COUNT(\*) DESC)

WHERE ROWNUM = 1)



**--Todas las reservas que se convirtieron en alquiler**

SELECT \*

FROM reserva r JOIN estado\_reserva er ON r.estado\_id = er.estado\_reserva\_id

WHERE er.estado = 'Confirmada';

**--Clientes con mas de 4 facturas**

SELECT cliente\_id, COUNT(\*) cant

FROM factura GROUP BY cliente\_id

HAVING COUNT(\*) > 4;

**--Valor total que pagó el cliente con mas facturas**

SELECT SUM(p.precio)

FROM pagos p JOIN factura f ON p.pago\_id = f.pago\_id

JOIN clientes c ON f.cliente\_id = c.cliente\_id

WHERE c.cliente\_id IN (SELECT cliente\_id FROM factura GROUP BY cliente\_id

HAVING COUNT(\*) IN (SELECT MAX(factura\_id) FROM factura GROUP BY cliente\_id))

**Consultas NoSQL:**

**--Servicio mas solicitado en la habitación mas reservada**

db.reserva.aggregate([

{

$unwind:'$habitaciones'

},

{

$group:{\_id:'$habitaciones.id\_habitacion', cant:{$sum:1}}

},

{

$sort : {suma:-1}

},

{ $limit : 1

}

])

db.factura.aggregate([

{

$lookup:

{

from: "productosXservicios",

localField: "servicio\_id",

foreignField: "registroXservicios",

as: "servicio"

}},

{

$group: {\_id: "$servicio.servicio\_id", veces: {$sum: 1}},

},{

$sort : {suma:-1}

},

{ $limit : 1 },

])

**--Nombre y salario del empleado que prestó el servicio más solicitado**

db.factura.aggregate([

{

$lookup:

{

from: "productosXservicios",

localField: "registroXservicios",

foreignField: "productosXservicios",

as: "prodtXServic"

}},

{

$lookup:

{

from: "empleado",

localField: "empleado\_id",

foreignField: "id\_empleado",

as: "empl"

}},

{ $unwind : "$prodtXServic" },

{ $unwind : "$empl" },

{

$group: {\_id: "$prodtXServic.servicio\_id", suma: {$sum: 1}},

},{

$sort : {suma:-1}

},

{ $limit : 1 },

])

**--empleado que mas ha facturado**

db.factura.aggregate([ { $lookup: { from: "empleado", localField: "empleado\_id", foreignField: "id\_empleado", as: "empl" } }, { $group: {\_id: "$empleado\_id", suma: {$sum: 1}} }, { $sort : {suma:-1} }, ])