

Proyecto creación bases de datos para Tadeo Tours.

Alaix Andrés
Correa Adriana

Universidad de Bogotá Jorge Tadeo Lozano.

Bases de Datos

Jun Gabriel Villamil

abril 2024

NORMALIZACION

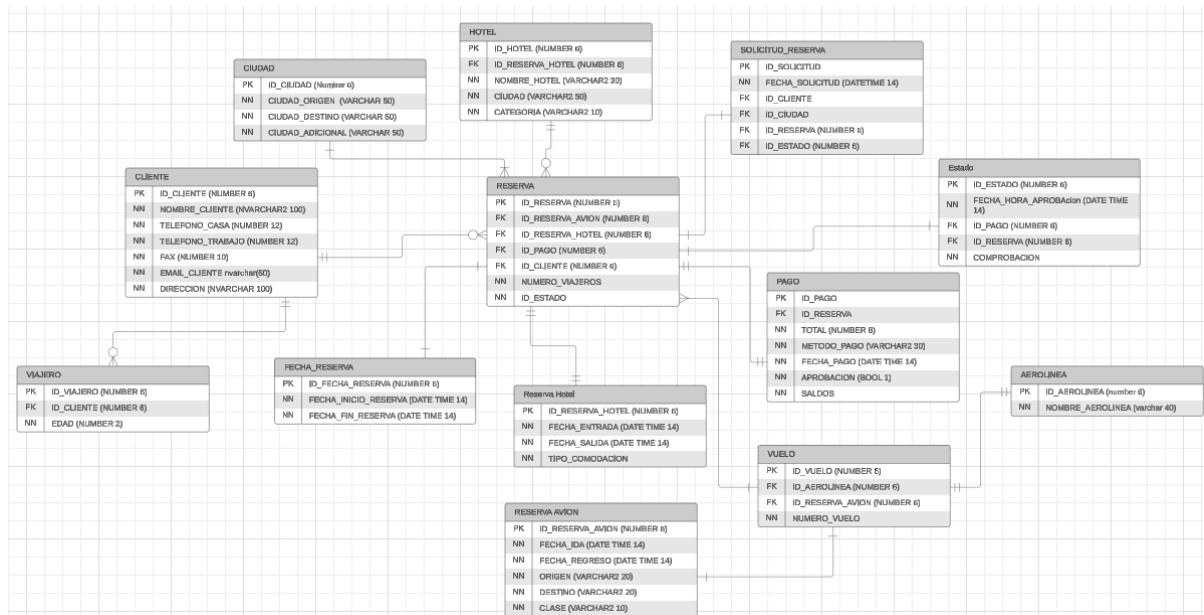
1. Se identifico las entidades y atributos principales de la problemática planeada, como lo son ciudad, hotel, solicitud reserva, pago, vuelo y viajero.

Nos aseguramos de que no existiera redundancias o repeticiones de datos en esas tablas que creamos

2. Se crearon dependencias en las tablas que se requerían y el modelo quedo con las siguientes tablas ciudad, hotel, solicitud reserva, estado, aerolínea, pago, vuelo, reserva avión, reserva, cliente, viajero y fecha reserva. Esto para separar información y estructurarla de una mejor manera. En vez de incluir los detalles de la reserva en una tabla, se crearon tablas separadas para reserva avión y reserva hotel.
3. Se eliminaron las dependencias transitivas asegurando que cada atributo no clave dependa únicamente de la clave primaria.

Después se establecieron las relaciones entre las tablas mediante el uso de llaves foráneas y foráneas para garantizar la integridad de las tablas.

Modelo relacional



Enlace de acceso: https://lucid.app/lucidchart/e72e7232-f72a-4f98-ad9d-22d832741846/edit?viewport_loc=-2085%2C-1153%2C4136%2C1572%2C0_0&invitationId=inv_351128f5-9177-4e0c-9a98-01120f757bc4

Proceso de creación:

Tomaremos la tabla de cliente como ejemplo para explicar su proceso de creación. Este enfoque lógico es aplicable a todas las demás tablas. Inicialmente creamos dos tablas: una llamada "clientes" y otra llamada "información cliente".

En la tabla de "clientes", teníamos un campo de identificación de cliente (ID cliente) y una relación con la clave foránea de reserva.

Por otro lado, en la tabla "información cliente", se almacenaba la información de contacto y dirección del cliente. En esta tabla, se establecía nuevamente una relación con el campo de ID cliente.

Bajo esta perspectiva, la información dentro de las tablas estaría redundante. Siguiendo el principio de normalización de bases de datos, se debe evitar la redundancia de información. Además, mantener dos tablas que no solo contenían datos duplicados, sino que también podían ser consolidadas en una sola, resultaba poco eficiente. El funcionamiento de la base de datos seguiría siendo el mismo tras esta consolidación.

En general, para todo el modelo, seguimos este principio: los datos de una tabla no deben repetirse en otra tabla. Además, si es necesario, creamos tablas adicionales para almacenar información relacionada con una tabla principal. Un ejemplo de esto es la tabla de reservas, donde se relaciona la información de reserva de hotel y reserva de avión para evitar la duplicación de datos.

Categorización

Tablas DDL

```
CREATE TABLE ciudad(  
    id_ciudad number,  
    ciudad_origen nvarchar2(100) not null,  
    ciudad_destino nvarchar2(100) not null,  
    ciudad_adicional nvarchar(50),  
    constraint pk_id_ciudad primary key (id_ciudad)  
);
```

```
--Secuencias de ciudad  
CREATE SEQUENCE seq_ciudad  
START WITH 1  
INCREMENT BY 1;
```

Tablas DML

```
-- INSERT INTO ciudad
INSERT INTO ciudad (id_ciudad, ciudad_origen, ciudad_destino, ciudad_adicional) VALUES
(1, 'Bogotá', 'Cali', 'Cartagena'),
(2, 'Barranquilla', 'Santa Marta', NULL),
(3, 'Bogotá', 'Pereira', NULL),
(4, 'Medellín', 'Bucaramanga', NULL),
(5, 'Pereira', 'Armenia', NULL);
(6, 'Medellín', 'Cali', NULL),
(7, 'Cartagena', 'Santa Marta', NULL),
(8, 'Cali', 'Bogotá', NULL),
(9, 'Cali', 'Bucaramanga', NULL),
(10, 'Pereira', 'Bogotá', 'Armenia');
(11, 'Bogotá', 'Cartagena', 'Pereira'),
(12, 'Cartagena', 'Barranquilla', 'Santa Marta'),
(13, 'Cali', 'pereira', NULL),
(14, 'Cali', 'Medellin', NULL),
(15, 'Pereira', 'Armenia', NULL);
```