

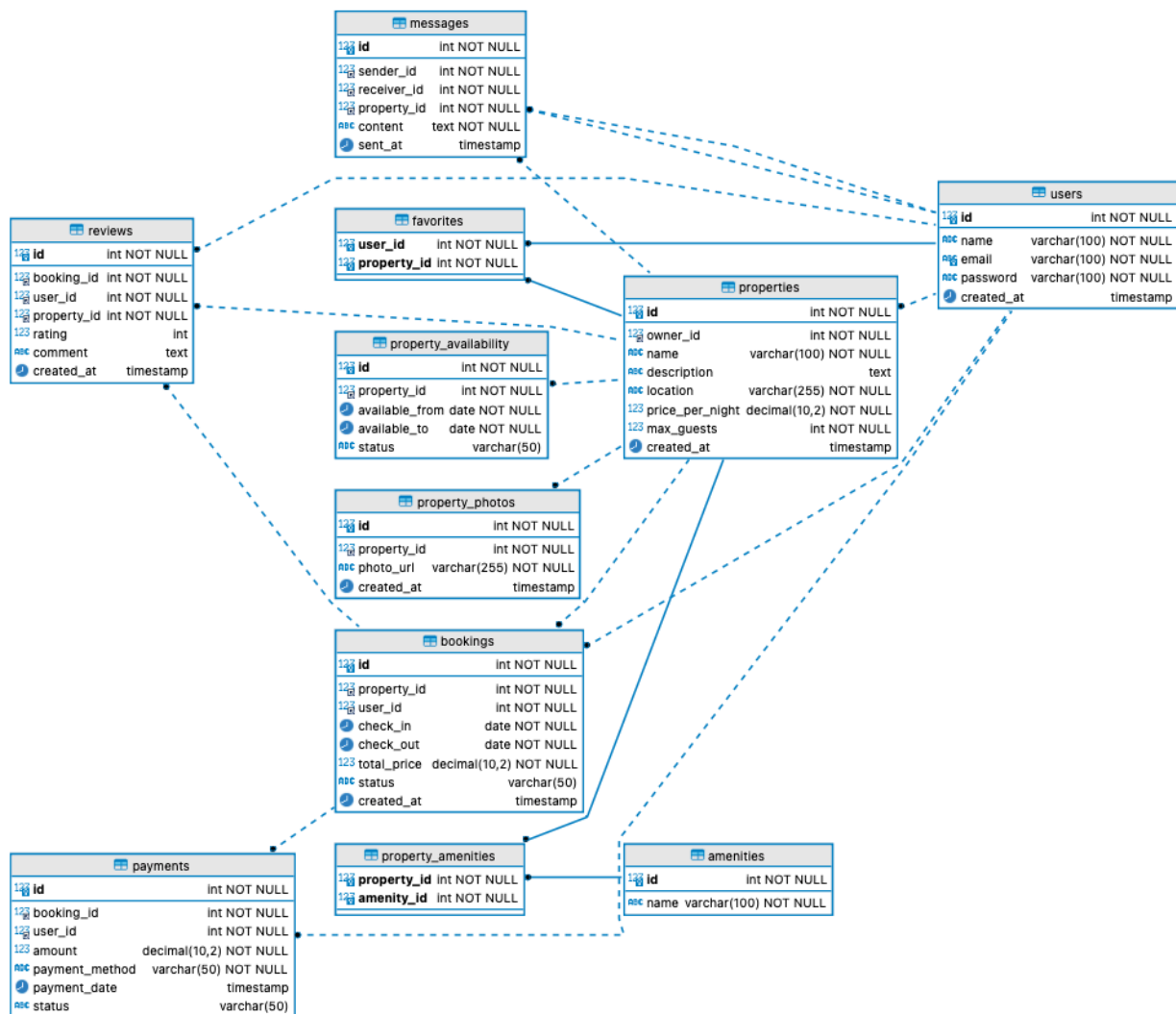
# Bases de Datos 2024

## Parcial I: SQL

Juan Cabral, Sergio Canchi, Tadeo Cocucci,  
Ramiro Demasi, Guido Ivetta

### Contexto

Se dispone de la base de datos `airbnb\_like` que se detalla en el siguiente diagrama:



# Consignas

1. Listar las 7 propiedades con la mayor cantidad de reviews en el año 2024.
2. Obtener los ingresos por reservas de cada propiedad.  
Esta consulta debe calcular los ingresos totales generados por cada propiedad. Ayuda: hay un campo **price\_per\_night** en la tabla de **properties** donde los ingresos totales se computan sumando la cantidad de noches reservadas para cada reserva multiplicado por el precio por noche.
3. Listar los principales usuarios según los pagos totales.  
Esta consulta calcula los pagos totales realizados por cada usuario y enumera los principales 10 usuarios según la suma de sus pagos.
4. Crear un trigger **notify\_host\_after\_booking** que notifica al anfitrión sobre una nueva reserva. Es decir, cuando se realiza una reserva, notifique al anfitrión de la propiedad mediante un mensaje.
5. Crear un procedimiento **add\_new\_booking** para agregar una nueva reserva.  
Este procedimiento agrega una nueva reserva para un usuario, según el ID de la propiedad, el ID del usuario y las fechas de entrada y salida. Verifica si la propiedad está disponible durante las fechas especificadas antes de insertar la reserva.
6. Crear el rol **admin** y asignarle permisos de creación sobre la tabla **properties** y permiso de actualización sobre la columna **status** de la tabla **property\_availability**.
7. Si ejecuto esta consulta:

```
START TRANSACTION;  
  UPDATE reviews  
  SET comment = 'bad'  
  WHERE rating = 1;  
COMMIT;
```

Se va a asignar como comentario **"bad"** a todos los reviews con **rating=1**.

*¿Por qué esto no contradice la propiedad de Durabilidad de ACID en las transacciones, ya que modifico los datos contenidos en la tabla **reviews**, cuando ya fueron cargados al comienzo del parcial?*

Responder como un comentario al final del archivo **soluciones.sql** que entregan en el parcial.

## Puntos a tener en cuenta

- Algunos nombres de columnas o tablas pueden ser palabras reservadas de SQL. Para evitar problemas, utilizar *`backticks`*:
  - E.g. **SELECT** **u.username** **FROM** **`user`** **u**;
- Mostrar únicamente los campos pedidos en la consigna y en el orden en el que se los pide (tanto a nivel fila como a nivel columna).
- Buscar hacer la consulta de la forma más sencilla posible.
- Se evaluará el correcto formato de las soluciones:
  - El código entregado debe ser legible.
  - Utilizar indentación de 4 espacios.
  - No escribir toda la consulta en una sola línea. Usen buen criterio para separar partes de la consulta.
  - Utilizar mayúsculas para denotar palabras clave de SQL (e.g. **`SELECT`**, **`INSERT`**, **`FROM`**, etc.).
  - Utilizar CTEs (i.e. **`WITH`**).

### Detalle Importante

El alumno puede usar sus notas personales e internet, pero queda prohibido utilizar modelos generativos (Chat-GPT, LLAMA, etc) y comunicarse con otras personas oral o digitalmente.

Romper alguna de estas dos restricciones invalida el examen

## Entrega

- Se entregará un archivo comprimido **`soluciones.sql.gz`** o **`soluciones.zip`** (con **`soluciones.sql`** adentro) con las soluciones de los 9 ejercicios. Separar las soluciones mediante comentarios de SQL.
- La entrega se hará mediante el Aula Virtual en el [correspondiente apartado](#).
  - Tendrán hasta las 17:00 para que se considere una entrega completa. La recomendación es empezar a subir el archivo a las 16:55 para evitar cualquier eventualidad.