

Tarea 2

Bases de Datos para Ingeniería – Laboratorio 2023
13 octubre de 2023

Objetivos:

- Realizar consultas en Álgebra Relacional
- Realizar consultas en SQL, interactuando con una base de datos alojada en un servidor de base de datos PostgreSQL

Material de referencia:

- Material del curso
- Capítulos 6 y 7, Fundamentals of Database Systems, Elmasri & Navathe, 7th Edition, Pearson, 2016
- [Tutorial PostgreSQL](#)

Formato de entrega:

- Las consultas en Álgebra Relacional deben entregarse en un archivo PDF. **IMPORTANTE:** En este PDF, indicar el número de grupo y los integrantes
- Cada consulta en SQL debe almacenarse en un archivo con extensión .sql (ej: consulta1.sql). Cada uno de estos archivos sólo debe contener una consulta. Todos estos archivos deben ser entregados en un archivo .tar.gz llamado consultas.tar.gz.

Entrega: Cada grupo tendrá tiempo para entregar la tarea hasta el 12/11/2023 a las 23:59 hs.

Introducción

Se cuenta con la base de datos (esquema y datos) de un sistema de reservas y estadías que se utilizó hasta hace un tiempo. Sin embargo, esta base de datos representa a una realidad que ya no se adecúa a la nueva realidad, descrita en la tarea 1, en lo que respecta a reservas y estadías. Se lo llamará “sistema anterior” A su vez, al nuevo sistema correspondiente a la realidad de la tarea 1 se lo llamará “sistema nuevo”

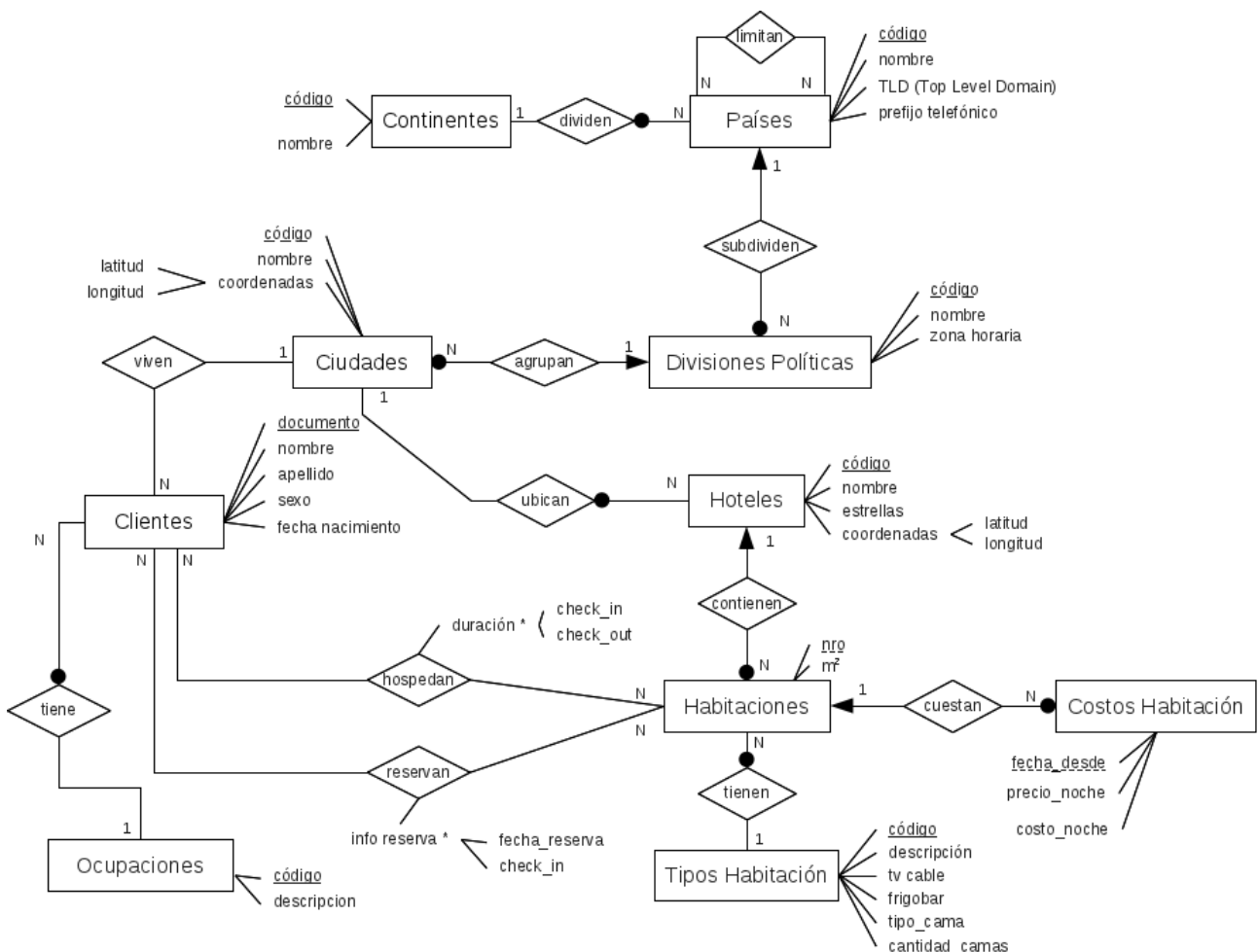
Como paso intermedio a la construcción del sistema nuevo se decidió realizar un prototipo aprovechando el esquema y los datos del sistema anterior.

En la sección 1 se presenta el MER y las tablas que se definieron para la base de datos relacional del sistema anterior. En la sección 2 se presentan las tablas que se usarán en el prototipo. La tarea 2 corresponde a una parte del armado del prototipo.

Por otro lado, sobre el sistema anterior se tenían definidas un conjunto de consultas en SQL. Lamentablemente estas consultas se perdieron y solo se cuenta con la documentación de lo que se quería. En la sección 3 se describen dichas consultas a incluir en el prototipo.

1. Sistema anterior

A continuación se presenta un MER que representa la realidad modelada por el sistema anterior.



RNEs:

- La relación limitan es irreflexiva.
 $\forall p \in \text{PAISES}, (p,p) \notin \text{LIMITAN}$
- La relación limitan es simétrica.
 $\forall a, b \in \text{PAISES} ((a, b) \in \text{LIMITAN} \Rightarrow (b, a) \in \text{LIMITAN})$
- La fecha de check-in de un hospedaje es menor que la fecha de check-out.
 $\forall e \in \text{ESTADIAS}, e.\text{check-in} < e.\text{check-out}$
- El costo de una habitación debe ser menor o igual que su precio.
 $\forall c \in \text{COSTOS_HABITACION}, c.\text{precio_noche} \leq c.\text{precio_noche}$

El MER descrito se representó mediante una base de datos relacional con el siguiente conjunto de tablas:

continentes (continente_codigo, nombre)

Representa los continentes. Cada continente se identifica por su código y posee un nombre.

países (pais_codigo, nombre, tld prefijo_telefonico, continente_codigo)

Representa los países. Cada país se identifica con un código y se almacena su nombre, su TLD (Top Level Domain), el prefijo para llamadas de larga distancia y el continente al que pertenece.

divisiones politicas (pais_codigo, division_politica_codigo, nombre, zona_horaria)

Representa las diferentes divisiones políticas de cada país. De cada división política se conoce un código, que es único para cada país, su nombre y su zona horaria.

ciudades (pais_codigo, division_politica_codigo, ciudad_codigo, nombre, latitud, longitud)

Representa las ciudades. De cada ciudad se conoce un código, que es único dentro de la división política donde está ubicada, su nombre y sus coordenadas geográficas (latitud y longitud).

limitan (pais1_codigo, pais2_codigo)

Representa los límites entre países.

hoteles (hotel_codigo, nombre, estrellas, latitud, longitud, pais_codigo, division_politica_codigo, ciudad_codigo)

Para cada hotel se conoce un código, su nombre, la cantidad de estrellas del hotel (en el rango de 1 a 5), sus coordenadas geográficas (latitud, longitud) y el país, división política y ciudad en la que encuentra ubicado.

tipos habitacion (tipo_habitacion_codigo, descripción, tv_cable, frigobar, tipo_cama, cant_cammas)

Representa los tipos de habitación que proveen los hoteles. De cada tipo de habitación se conoce un código, una descripción, si dispone de TV cable, si dispone de frigobar, el tipo de cama (Simple, Matrimonial, Matrimonial King Size) y la cantidad de camas que contiene.

habitaciones (hotel_codigo, nro_habitacion, tipo_habitacion_codigo, m2)

Representa las habitaciones de los hoteles. De cada habitación se conoce un nro_habitación, el hotel al cual pertenece, un tipo de habitación y los metros cuadrados que ocupa.

clientes(cliente_documento, nombre, apellido, sexo, fecha_nacimiento, ocupacion_codigo, pais_codigo, division_politica_codigo, ciudad_codigo)

Representa las personas del sistema. De cada cliente se conoce su documento, su nombre, apellido, sexo, fecha de nacimiento, ocupación y la ciudad de residencia.

ocupaciones (ocupacion_codigo, descripcion)

Representa las ocupaciones de los clientes. De cada ocupación se conoce su código y descripción.

reservas anteriores (hotel_codigo, nro_habitacion, cliente_documento, fecha_reserva, check_in)

Representa **las reservas en el sistema anterior** efectuadas por los clientes. De cada reserva se conoce el hotel, el número de habitación reservada, el documento del cliente que hizo la reserva, la fecha en que efectuó la reserva y la fecha en la que el cliente realizará el checkin.

estadias anteriores (hotel_codigo, nro_habitacion, cliente_documento, check_in, check_out)

Representa **las estadias en el sistema anterior** efectuadas por los clientes. De cada estadía se conoce el hotel, el número de habitación ocupada, el documento del cliente, la fecha de check_in y la fecha de check_out.

costos habitacion (hotel_codigo, nro_habitacion, fecha_desde, costo_noche, precio_noche)

Representa el histórico de costos y precios de las habitaciones. Se conoce el hotel, el número de habitación, la fecha de comienzo de vigencia de los montos. El valor de costo_noche representa el costo de mantenimiento de la habitación para el hotel, mientras que precio_noche es el monto que se le cobra al cliente.

Sobre este conjunto de tablas se cumplen las siguientes dependencias de inclusión:

$\Pi_{\text{continente_codigo}} \text{ paises} \subseteq \Pi_{\text{continente_codigo}} \text{ continentes}$

$\Pi_{\text{pais_codigo}} \text{ divisiones_politicas} \subseteq \Pi_{\text{pais_codigo}} \text{ paises}$

$\Pi_{\text{pais_codigo}, \text{division_politica_codigo}} \text{ ciudades} \subseteq \Pi_{\text{pais_codigo}, \text{division_politica_codigo}} \text{ divisiones_politicas}$

$\Pi_{\text{pais1_codigo}} \text{ limitan} \subseteq \Pi_{\text{pais_codigo}} \text{ paises}$

$\Pi_{\text{pais2_codigo}} \text{ limitan} \subseteq \Pi_{\text{pais_codigo}} \text{ paises}$

$\Pi_{\text{tipo_habitacion_codigo}} \text{ habitaciones} \subseteq \Pi_{\text{tipo_habitacion_codigo}} \text{ tipos_habitacion}$

$\Pi_{\text{hotel_codigo}} \text{ habitaciones} \subseteq \Pi_{\text{hotel_codigo}} \text{ hoteles}$

$\Pi_{\text{hotel_codigo}, \text{nro_habitacion}} \text{ reservas_anteriores} \subseteq \Pi_{\text{hotel_codigo}, \text{nro_habitacion}} \text{ habitaciones}$

$\Pi_{\text{cliente_documento}} \text{ reservas_anteriores} \subseteq \Pi_{\text{cliente_documento}} \text{ clientes}$

$\Pi_{\text{hotel_codigo}, \text{nro_habitacion}} \text{ estadias_anteriores} \subseteq \Pi_{\text{hotel_codigo}, \text{nro_habitacion}} \text{ habitaciones}$

$\Pi_{\text{cliente_documento}} \text{ estadias_anteriores} \subseteq \Pi_{\text{cliente_documento}} \text{ clientes}$

$\Pi_{\text{hotel_codigo}, \text{nro_habitacion}} \text{ costos_habitacion} \subseteq \Pi_{\text{hotel_codigo}, \text{nro_habitacion}} \text{ habitaciones}$

$\Pi_{\text{pais_codigo}, \text{division_politica_codigo}, \text{ciudad_codigo}} \text{ clientes} \subseteq \Pi_{\text{pais_codigo}, \text{division_politica_codigo}, \text{ciudad_codigo}} \text{ ciudades}$

$\Pi_{\text{ocupacion_codigo}} \text{ clientes} \subseteq \Pi_{\text{ocupacion_codigo}} \text{ ocupaciones}$

$\Pi_{\text{pais_codigo}, \text{division_politica_codigo}, \text{ciudad_codigo}} \text{ hoteles} \subseteq \Pi_{\text{pais_codigo}, \text{division_politica_codigo}, \text{ciudad_codigo}} \text{ ciudades}$

2. Sistema nuevo

Las tablas de la base de datos relacional que se usan en el prototipo del sistema nuevo son las mismas que las del sistema anterior con el objetivo de contar con datos para revisar las consultas SQL que se van definiendo.

El esquema relacional del sistema anterior presenta diferencias de representación sólomente en lo que respecta a las reservas y las estadías de acuerdo a la tarea 1. A saber,

- una habitación para una fecha de llegada (check_in) puede ser reservada en distintas fechas de reserva.
- una misma estadía puede tener varias reservas
- una estadía en una habitación puede realizarse sin reserva

Se deberán tomar decisiones para representar las reservas y las estadías en el sistema nuevo debido a la diferencias mencionadas. Estas decisiones se irán guiando en la sección 3 donde se describe lo que se pide en esta tarea 2.

3. Consultas

Se pide resolver las consultas descriptas abajo, escribiendo las expresiones en los lenguajes que se indican en cada una. Justificar su respuesta en caso que no sea posible escribir la expresión en alguno de ellos.

En el caso de SQL, ejecutar la expresión en la base de datos del prototipo del sistema nuevo cuyo esquema y datos para importar¹ se encuentran en "2023_Tarea2_EsquemaDatos.zip" en el eva. Tener en cuenta las restricciones generales que se indican al final de esta sección.

Se comienza con la consulta que no involucra los conceptos de reservas y estadías.

Consulta 1
Esquema del resultado: hotel_codigo, hotel_nombre
Devolver el código y nombre de los hoteles del continente europeo ('EU') con más de 3 estrellas y cuyo nombre es igual al nombre de la ciudad donde se ubican.
Resolverla en: <ul style="list-style-type: none">• álgebra relacional, y• SQL

Con las consultas siguientes se irá elaborando y tomando decisiones para representar las estadías y las reservas del sistema nuevo a partir de las tablas del sistema anterior (reservas_anteriores y estadías_anteriores)

Consulta 2 – Estadías en el sistema nuevo
Esquema del resultado: hotel_codigo, nro_habitacion, cliente_documento, check_in
Las estadías que nos interesan en el sistema nuevo son las estadías que tienen reservas. Escribir una consulta SQL que devuelva las estadías del sistema anterior que tienen reservas. Usando esta consulta, <u>definir una vista</u> ² con el nombre estadías y los atributos que se detallan en el esquema resultado. Esta vista representa a las estadías en el sistema nuevo.

1 Para importar, use el comando psql <https://www.postgresql.org/docs/15/app-psql.html>

2 <https://www.postgresql.org/docs/15/sql-createview.html>

Se decide que las reservas del sistema anterior que interesan en el sistema nuevo serán las reservas con estadía en el sistema anterior.

Consulta 3 – Reservas con estadía en el sistema anterior sin el precio

Esquema del resultado: hotel_codigo, nro_habitacion, cliente_documento, fecha_reserva, check_in, check_out

En el sistema anterior, las reservas de una estadía, son aquellas que coinciden con la habitación y la fecha de llegada.

Devolver las reservas del sistema anterior (sin el precio) y que tienen estadía. En caso que una estadía tenga más de 1 reserva, devolver la reserva con la última fecha de reserva (fecha de reserva mayor) y el cliente correspondiente a ella.

Además si existen dos estadías asociadas a la reserva con última fecha de reserva, considerar la que tiene fecha de check_out menor.

Resolverla en:

- SQL

Usando la consulta SQL, definir una vista con el nombre **reservas_anteriores_sinPrecio** y los atributos que se detallan en el esquema resultado.

Consulta 4 – Reservas en el sistema anterior con estadía y con el precio

Esquema del resultado: hotel_codigo, nro_habitacion, cliente_documento, fecha_reserva, check_in, check_out, precio_noche

Aprovechando la vista definida en la consulta 3 (**reservas_anteriores_sinPrecio**), escribir una consulta SQL que agregue el precio por noche a esas reservas.

El precio por noche de la reserva es el último precio de la habitación con respecto a la fecha de check_in. Por ejemplo, si la fecha de check_in de la reserva es 1/5/23 y los precios de la habitación son (31/8/22, 100), (15/1/23, 150) y (15/8/23, 200), el precio de la reserva es 150.

Nota: Una vez definida, la vista en SQL se trabaja en las consultas como una tabla más.

Usando esta consulta, definir una vista con el nombre **reservas** y los atributos que se detallan en el esquema resultado.

Habiendo definido las reservas y estadías del sistema nuevo, a continuación se siguen con las consultas que necesitan de ellas. Por lo tanto, para las consultas 5 en adelante que se piden a continuación tener en cuenta que:

- NO se debe utilizar ni la tabla **reservas_anteriores** ni la tabla **estadias_anteriores**, y
- las referencias a “reservas” y a “estadías” hacen mención a reservas y estadías en el sistema nuevo. Es decir, a las vistas **estadias** y **reservas** definidas anteriormente.

Consulta 5
Esquema del resultado: hotel_codigo, nombre, estrellas, ciudad_nombre
Devolver el identificador, nombre, cantidad de estrellas y nombre de la ciudad de ubicación de los hoteles para los cuales se tienen reservas pero ninguna estadía.
Resolverla en: <ul style="list-style-type: none">• álgebra relacional, y• SQL

Consulta 6
Esquema del resultado: ocupacion_codigo, descripcion, estadia_promedio
Interesa conocer las duraciones promedio de las estadías de las personas según su ocupación. Devolver para cada ocupación el identificador, descripción y el promedio de duración de los hospedajes. El hospedaje promedio debe calcularse con dos dígitos decimales (utilizar la función ROUND ³). No se deben realizar redondeos intermedios.

Restricciones generales a los efectos del objetivo de la tarea

- Respetar los nombres de atributos indicados en la sección “**Esquema del resultado**” de cada consulta. **Las soluciones que no las respeten serán consideradas incorrectas.**
- No utilizar vistas, a menos que se indique en la letra.
- No utilizar (SELECT ... FROM) en la lista de atributos de un SELECT.
- No usar subconsultas en el FROM en ningún nivel de SELECT.
- Las únicas funciones de agregación que se pueden usar son MIN, MAX, COUNT y AVG
- Para cualquier otra cláusula que pretenda utilizar que no sea de la forma básica (SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY ...), consultar si está permitido su uso.

3 <https://www.postgresql.org/docs/15/functions-math.html>