Universidad Autónoma de Madrid

Departamento de Ingeniería Informática

Modelado, Almacenamiento y Gestión de Datos

Práctica 2

Creación de la Base de Datos

Álvaro del Val

Índice

1. Objetivos

El objetivo de esta práctica es crear una base de datos relacional en PostgreSQL para el modelo relacional realizado en la práctica 1. Como objetivos específicos: crear las tablas y cargar los datos en una base de datos relacional.

No existe un único diseño posible para un mismo conjunto de datos. Para permitir una corrección uniforme de la práctica, os proporcionamos un modelo relacional específico que debéis implementar; en la memoria, deberéis identificar las semejanzas y diferencias con el que entregasteis en la primera práctica.

Para esta segunda práctica, se requiere:

- Creación de la base de datos
- Preprocesamiento del conjunto de datos para crear un conjunto de ficheros intermedios.
- Carga de los datos en la base de datos
- Memoria en pdf

2. Creación de la base de datos

El modelo relacional lo tendréis disponible en Moodle a partir del viernes 4 de octubre.

En esta parte de la práctica, debéis:

- Familiarizaros con el uso de servidor y clientes de consola (psql) y gráficos (pgAdmin 4 o dBeaver) para PostgreSQL. Ver el documento de ayuda sobre PostgreSQL que tenéis en Moodle. En particular, debéis conocer y saber usar los comandos de terminal *dropdb*, *createdb* y *psql*, y ejecutar ficheros de código SQL desde la terminal.
- Crear la base de datos **airbnb** con propietario *alumnodb* y clave *alumnodb*. Este usuario ya existe en los laboratorios.

- Crear un fichero **airbnb-schema.sql** con el código SQL necesario para crear todas las tablas (con sus restricciones asociadas) necesarias para implementar el modelo relacional proporcionado

Para crear este código, podéis usar el método que prefiráis, se puede por ejemplo exportar el modelo relacional de *dbdiagram.io* para generar una primera versión del código, hacer modificaciones desde un cliente GUI como dBeaver, y finalmente generar el código final usando *pgdump* (que es el comando para hacer volcados de las bases de datos de PostgreSQL). También se puede usar el comando *csvsql* de *csvtoolkit*.

No cambies los nombres de las tablas que se proporcionan en el modelo relacional. Es necesario mantener estos nombres para permitir la corrección tanto de esta práctica como de las siguientes.

3. Preprocesamiento de datos

Como ya habéis observado en la primera práctica, para crear vuestro diseño relacional, es necesario un preprocesamiento de los ficheros csv originales para obtener los contenidos de las tablas que finalmente componen la base de datos. Por ejemplo, se necesita una tabla de las "amenities" permitidas, y otra tabla que capture la relación $n - n$ entre listings y amenities (o más bien, entre id's de listings e id's de amenities). Hay también otras tablas del modelo relacional que deben generarse a partir de los datos.

El objetivo de este apartado de la práctica es preprocesar los datos originales para poder luego rellenar las tablas creadas en el paso anterior. Para ello, tenéis que hacer un procesamiento similar de los ficheros csv originales,por ejemplo procesar las columnas json, convertir los precios en numéricos, gestionar problemas con valores inexistentes.... Es posible además que identifiquéis errores en los datos, etc.

Debéis proporcionar todo el código Python necesario para realizar este preprocesamiento. Podéis bien crear un notebook (*airbnb.ipynb*)que produzca el output deseado tras procesar todas sus celdas, o un único programa python *airbnb.py* que al ejecutarlo produzca de una sola vez todo el output necesario. En ambos casos, debe

ser posible realizar todo el procesamiento simplemente indicando (bien con una variable en la primera celda del notebook, bien como argumento del script de python) el directorio donde se encuentran los ficheros .csv originales.

Aunque existen diferentes maneras de realizar este paso (por ejemplo, el programa podría crear todo un conjunto de comandos INSERT), sugerimos que la mejor manera es crear un nuevo fichero csv para cada una de las tablas a rellenar, que luego pueda procesarse con *copy* (ver apartado siguiente). Esto permite mantener separados los datos de los programas usados para manipularlo.

Para facilitar la corrección, podéis nombrar cada uno de los ficheros generados como *proc_nombre_tabla.csv*, por ejemplo, *proc_cities.csv* contendría los nombres de ciudades admisibles, esto es, las filas de la tabla *cities*, y *proc_amenities.csv* el csv para rellenar la tabla *amenities*. (El prefijo *proc_* indica que el fichero es un fichero "procesado.ª partir de los originales, y nos permite distinguir fácilmente ambos tipos de ficheros).

El código debe en ambos casos estar adecuadamente estructurado, definiendo funciones para todas las tareas principales para así evitar la repetición de código.

4. Rellenar las tablas

Cada una de las tablas puede ahora rellenarse usando un comando de la shell *psql* como por ejemplo

```
\copy listings from 'proc_listings.csv' csv header delimiter ',' quote '"';
```

Todos estos comandos a su vez pueden recopilarse en un único script *createdb_script.sql* que puede ejecutarse con:

```
psql -U alumnodb airbnb < createdb_script.sh
```

Se proporciona un Makefile que puede crear y rellenar la base de datos si se usan los nombres de ficheros sugeridos. Se pueden modificar estos nombres como se desee, siempre que se modifique el Makefile para que al escribir "make.en la terminal se (re)cree la base de datos completa.

Una vez creada la base de datos y rellenadas todas las tablas, podéis generar un volcado completo (copia de seguridad) de la BD usando

```
pg_dump -U alumnodb airbnb > airbnb_dump.sql
```

No debéis entregar esa copia, pero os puede ser útil para trabajar.

5. Memoria

La memoria es un documento en formato pdf que debe incluir, de forma concisa pero completa, al menos los siguientes apartados:

- Semejanzas y diferencias detalladas entre el diseño relacional de vuestra primera práctica y el que os proporcionamos nosotros. ¿Están justificadas estas diferencias? ¿Pueden considerarse errores (vuestros o nuestros) o simplemente diferentes opciones o alternativas de diseño?
- Detalles sobre el preprocesamiento de los datos en python. ¿Cómo habéis planteado el problema y qué enfoque habéis usado? ¿Cómo habéis estructurado el código? ¿Qué problemas habéis encontrado? ¿Ha habido que modificar o eliminar datos?
- Cualquier otro aspecto que consideréis relevante

6. Entregable

Debéis entregar un único fichero zip llamado *magd_XXXX_pYY.zip*, donde XXXX es vuestro grupo de prácticas (7261 grupo del martes, 7262 grupo del viernes), e YY es vuestro número de pareja, por ejemplo 03 si sois la pareja 3 (los números de parejas están publicados en la pèstaña de presentación de las prácticas).

El contenido de este fichero debe ser un **único directorio** de nombre *magd_XXXX_pYY* (es decir, igual que el zip pero sin la extensión)

Tiene que estar todo contenido dentro de un directorio porque, si no, al descomprimir todas las entregas se sobreescribirían los ficheros de unas parejas con los de otras.

Este directorio debe contener los siguientes ficheros:

- Fichero *airbnb_schema.sql* que contiene los comandos necesarios de SQL para crear las tablas de la base de datos **airbnb**
- Fichero python *airbnb.ipynb* (si usáis un notebook) o *airbnb.py* (si usáis un único script) que al ejecutarse indicando (en el notebook o por linea de comandos) el directorio con los ficheros csv originales, genere todos los ficheros csv
- Fichero *createdb_script.sql* con todos los comandos de copia de las ficheros csv procesados a las tablas de la base de datos **airbnb**
- Fichero Makefile para recrear la base de datos
- Memoria en formato pdf

No incluyáis ninguno de los ficheros de datos, ni el volcado final de la base de datos (aunque podríamos pediros este último si tenemos problemas a la hora de ejecutar vuestro código).

7. Criterios de corrección

Para **aprobar** es necesario:

- Que el Makefile funcione sin errores.
- Entregar fichero sql para la creación de la base de datos y que este incluya todas las tablas y atributos presentes en el diagrama relacional (no más de 3 errores).
- Entregar fichero *airbnb.py* (o *airbnb.ipynb*) en el que al menos un 50 % del dataset sea preprocesados y cargado a la base de datos.
- Entregar una memoria respondiendo a las preguntas que se os formulan.

Para obtener una nota en el **rango 5-6.9** es necesario:

- Satisfacer todos los requerimientos del apartado anterior.
- Que el 70 % del dataset sea correctamente preprocesado y cargado a la base de datos.
- Que las respuestas en la memoria no tengan más de 3 errores u omisiones, y que estén correctamente redactadas.

Para obtener una nota en el **rango 7-8.9** es necesario:

- Satisfacer todos los requerimientos del apartado anterior.
- Que el dataset completo sea correctamente preprocesado y cargado a la base de datos (se permite filtrado de datos erróneos).
- Que la descripción de las diferencias del diseño suministrado en la P2 con respecto al diseño entregado por la pareja en la P1 sea completo (sin errores/omisiones).

Para obtener una nota en el **rango 9-10** es necesario:

- Satisfacer todos los requerimientos del apartado anterior.
- Que se hayan tomado decisiones correctas en el preprocesado de datos para solucionar problemas presentes en los datos y evitar descartar datos complicados.
- Que, en la memoria, la comparación de diseño con respecto al entregado en la P1 demuestre una comprensión profunda del diseño de bases de datos

También se valorarán la presentación y la redacción de la memoria, y que se comenten otros aspectos relevantes del contenido de la práctica.