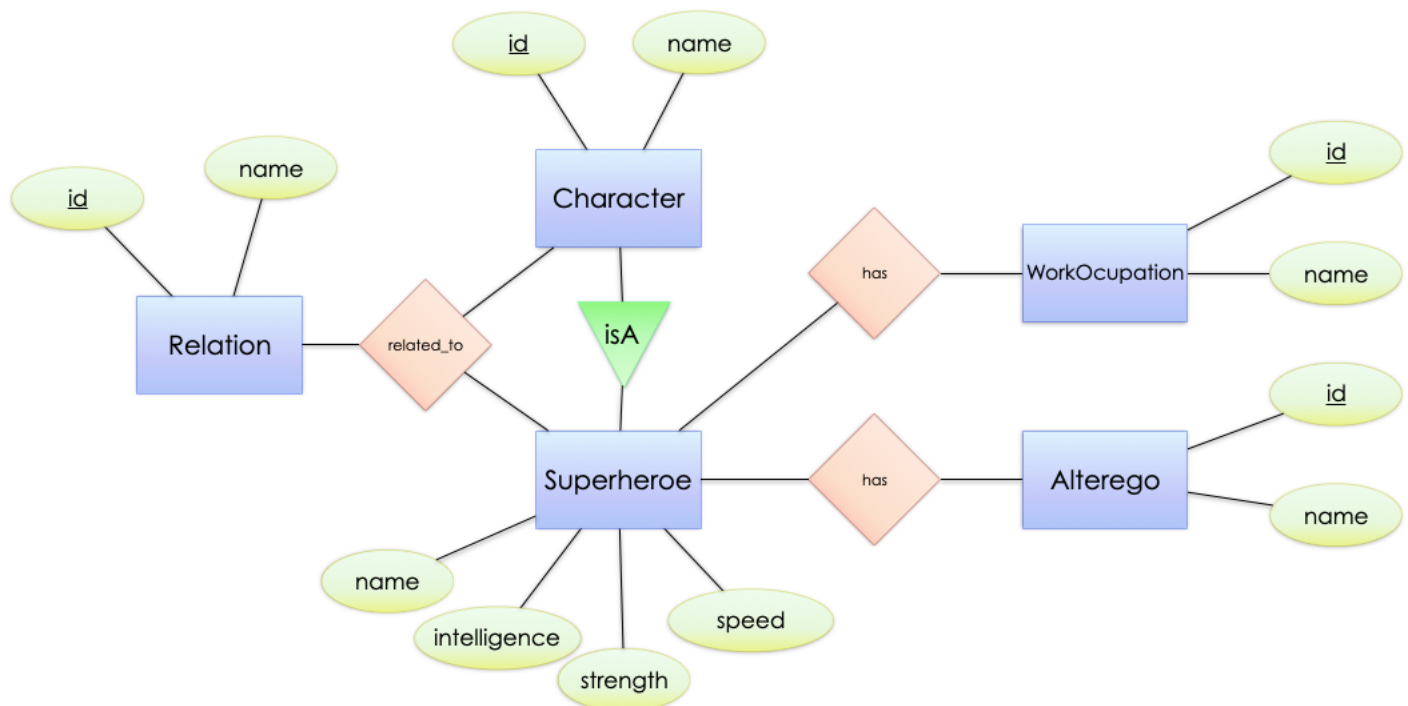


Laboratorio 5 - LDD y Acceso Programático

Profesores: Claudio Gutiérrez
Matías Toro
Auxiliares: Scarlett Plaza
Daniel Radrigán
Cristian Salazar
Fran Zautzik

Considere el siguiente modelo entidad relacion que corresponde a una base de datos de super-heroes:



En este laboratorio usted deberá crear las tablas necesarias correspondientes al modelo entidad relación y luego importar los datos a partir de un archivo csv.

La creación de tablas e importación deberá ser hecha en el servidor del curso en el esquema **superheroes**.

- Para conectarse al servidor directamente, usted puede usar una terminal en Linux/Mac/Windows o Putty en Windows. El servidor está hosteado en `cc3201.dcc.uchile.cl` y escucha en el puerto 240. Su usuario es `cc3201` y la contraseña se publicó en u-cursos en Material Docente. El comando que deberían ejecutar en la terminal para conectarse al servidor desde Linux/Mac es:

```
ssh -p 240 cc3201@cc3201.dcc.uchile.cl
```

Para conectarse a la base de datos basta con ejecutar el comando `psql cc3201`. Para cerrar la sesión utilice el comando `\q`.

- Para conectarse al servidor programáticamente, puede crear una conexión en python como se describe a continuación:

```
conn = psycopg2.connect(host="cc3201.dcc.uchile.cl",  
    database="cc3201",  
    user="cc3201",  
    password="j'<3_cc3201", port="5440")
```

Usted deberá entregar un archivo de texto (`.txt`) y un archivo python (`.py`) con el desarrollo de los ejercicios que siguen.

- P1.** 30 PUNTOS Cree las tablas necesarias de modo que satisfagan el modelo entidad relación en el schema **superheroes**, agregando las restricciones de llave primaria y foránea según corresponda. Tenga en cuenta que el nombre de cada tabla debe tener un prefijo único para cada grupo, como por ejemplo **equiporocket_tablita**.

Hints:

- El campo id lo puede crear de tipo **serial**.
- Sugerimos traducir la herencia (isA) usando dos tablas (para **character** y **superhero**).

- P2.** 60 PUNTOS Crea y corra un archivo python para poblar la base de datos creada en el punto anterior a partir del archivo de datos csv provisto en ucursos. Este archivo tiene varios errores e inconsistencias, por lo que le sugerimos leer las siguientes consideraciones:

- Le recomendamos hacer la tarea en python. Para instalar la librería necesaria para conectarse a la base de datos utilice `pip3` ejecutando `pip3 install psycopg2-binary`
- A veces el nombre del pariente es el nombre de superhéroe y otras veces el nombre real, por lo que se sugiere procesar el archivo csv dos veces: una para ingresar los superhéroes y otra para procesar las relaciones (y crear personajes que no sean superhéroes). Esto significa que al buscar parientes debe buscar en ambas tablas (puede escoger el primer valor encontrado).
- Algunos strings pueden venir con comillas dobles extras (e.g. **adventurer**, **"agent"**, **mechanic**)

Por lo tanto le sugerimos seguir los siguientes pasos.

- (a) **Creación del superhéroe.** Lea el archivo. Por cada personaje (fila):

- i. Seleccione el id del personaje por nombre en caso de ya exista. De lo contrario insertelo y recupere su id. Puede asumir por simplicidad que no hay dos personajes (characters) que se llamen igual. Si hay personajes sin nombre, por simplicidad puede usar el de superhéroe.
- ii. Seleccione el id del superhéroe dado el id del personaje del punto anterior. En caso de no existir, crealo.
- iii. Para cada alter ego (asuma que están separados por “,” o “;”)
 - A. Elimine espacios blancos al comienzo y al final, y comillas dobles.
 - B. Busque si ya existe el alter ego para ese superhéroe. Si no existe, insertelo.
- iv. Para cada ocupación/oficio (asuma que estan separadas por “,” o “;”)
 - A. Elimine espacios blancos al comienzo y al final, y comillas dobles.
 - B. Seleccione el id de la ocupación dado el nombre de esta. Si no existe, créala. Note que un mismo oficio puede venir en distintas combinaciones de mayúsculas y/o minúsculas en distintas filas.
 - C. Busque si ya existe un elemento en la tabla intermedia entre superhéroe y ocupación. Si no existe insertela.

(b) **Creación de los parientes.** Lea de nuevo el archivo. Por cada personaje:

- i. Seleccione el id del personaje (ya debería existir según lo de arriba)
- ii. Ignore filas donde la relación es el string “-”.
- iii. Para cada relación (asuma que están separadas por “,” o “;”)
 - A. Elimine espacios blancos al comienzo y al final, y comillas dobles.
 - B. Separe el pariente de su parentesco. Ignore el estado del pariente (deceased). Para validar un pariente y separar el nombre de su parentesco (e.g. en `Ashley Zolomon (wife, deceased)`), use el comando `m = re.search("(^[^+)] []*\((([^)]+)+\)\)", texto)`. Luego, si `m` está definido, `m.group(1)` apunta al nombre (e.g. `Ashley Zolomon`), y `m.group(2)` al tipo de relación y estado (e.g. “wife, deceased”).
 - C. Obtenga el id del personaje correspondiente al pariente, ya sea por el nombre de superheroe o de personaje. Si no encuentra al personaje, cree uno nuevo.
 - D. Busque si existe la relación de parentesco. Si no existe, cree una nueva.

(c) Haga “commit” de sus cambios.

P3. 10 PUNTOS Entregue las consultas sql que responden las siguientes preguntas

- (a) **Pregunta:** Calcule los nombres de los 3 superheroes con más parientes (note que hay nombres de superheroes que se repiten, e.g. Batman (Bruce Wayne), Batman (Terry McGinnis) y Batman (Dick Grayson)).

- (b) **Pregunta:** Calcule los nombres de los 3 personajes no superheroes con más parientes.
- (c) **Pregunta:** Calcule los nombres de los 5 superheroes con más parientes superheroes.
- (d) **Pregunta:** Calcule el nombre de relación más común
- (e) **Pregunta:** Calcule los 3 trabajos más populares