Disseny i Administració de Bases de Dades

Sessions 1, 2, 3 i 4: SQL amb SQLite, MySQL i PostgreSQL. Permisos. Normalització. Connexió des de programes i SQL injection.

José L. Balcázar, Jordi Esteve

EPSEVG – CS – UPC – Hivern 2024

Sessió 1

- Introducció a SQLite
- Consultes SQL en una B.D. no normalitzada

Què és en realitat SQLite?

En el fons no és un SGBD

SQLite

és, en esència, un format obert amb el que guardar tota la informació d'una determinada base de dades en un únic fitxer.

Què és en realitat SQLite?

En el fons no és un SGBD

SQLite

és, en esència, un format obert amb el que guardar tota la informació d'una determinada base de dades en un únic fitxer.

- Addicionalment, disposem de programes capaços d'operar amb aquest fitxer de diverses maneres:
  - Llegir-lo, actualitzar-lo,
  - modificar-lo a través d'un intèrpret d'instruccions SQL;
  - o bé oferir APIs per gestionar la base de dades des d'altres programes.

Què és en realitat SQLite?

En el fons no és un SGBD

SQLite

és, en esència, un format obert amb el que guardar tota la informació d'una determinada base de dades en un únic fitxer.

- Addicionalment, disposem de programes capaços d'operar amb aquest fitxer de diverses maneres:
  - Llegir-lo, actualitzar-lo,
  - modificar-lo a través d'un intèrpret d'instruccions SQL;
  - o bé oferir APIs per gestionar la base de dades des d'altres programes.
- Avantatge: Compatibilitat!
  - depenent només d'un únic fitxer, pots usar la mateixa base de dades des de diferents programes, escrits en diferents llenguatges, i fins i tot
  - pots portar la base de dades en una memòria USB o enviar-la amb facilitat com un fitxer adjunt.

Instal·lació

No està clar que significa "instal·lar" SQLite

Per usar SQLite

només cal tenir disponible algun programa capaç de llegir i escriure fitxers que entenguin aquest format.

Instal·lació

No està clar que significa "instal·lar" SQLite

Per usar SQLite

només cal tenir disponible algun programa capaç de llegir i escriure fitxers que entenguin aquest format.

- Des de línia de comandes (trobareu CLI per diferents plataformes a sqlite.org),
- Administradors basats en GUI:

  sqliteexpert, sqlitebrowser, Sqliteman, SQLite Manager, SQLiteSpy, SQLite Administrator, SQLite Maestro...
- APIs per llenguatges de programació: C, Java...

  Veurem exemples des de `Python`.

Instal·lació

No està clar que significa "instal·lar" SQLite

Per usar SQLite

només cal tenir disponible algun programa capaç de llegir i escriure fitxers que entenguin aquest format.

- Des de línia de comandes (trobareu CLI per diferents plataformes a sqlite.org),
- Administradors basats en GUI:

  sqliteexpert, sqlitebrowser, Sqliteman, SQLite Manager, SQLiteSpy, SQLite Administrator, SQLite Maestro...
- APIs per llenguatges de programació: C, Java...

  Veurem exemples des de `Python`.

Cal tenir en compte la coherència de versions.

- La versió 3 de SQLite no és compatible cap enrera.
- És molt aconsellable usar només SQLite3.

El recurs d'emergència

és l'intèrpret de SQL en línia de comandes (CLI).
Convé agafar una mica d'experiència perquè és el que gairebé sempre pots comptar amb ell en casos extrems.

- Es pot baixar i executar sense instal·lació.
- És possible que el puguis instal·lar també mitjançant els mecanismes habituals (apt-get. . . , si ets root).
- Pots baixar-lo precompilat i acostuma a funcionar.
- En cas de que no funcionin els sistemes anteriors, pots baixar el codi font i compilar-lo.

Ús de l'intèrpret

Admet SQL i instruccions addicionals

Les instruccions SQL sempre acaben amb ";" com requereix SQL.

Les instruccions "no SQL" van precedides d'un punt (i no acaben amb punt i coma!):

- .exit .quit
- .databases .tables .schema .fullschema
- .mode .output .separator
- .import .read
- .timer
- ...

Experiments amb SQLite, I

Aprenentatge de SQL: Repetició dels SQLs de la classe de teoria d'àlgebra relacional

Documentació SQLite: `https://sqlite.org`
Sintaxis SQL: `https://sqlite.org/lang.html`

- Executa el programa: `sqlite3`
- Surt d'ell (`.exit` o `.quit`).
- Executa'l un altre cop amb un nom de fitxer nou afegint l'extensió típica db: `sqlite3 fitxer.db`
- Crea algunes taules (CREATE TABLE) i insereix algunes tuples (INSERT INTO).
- Surt del programa i comprova l'existència i mida del `fitxer.db`.
- Repeteix els exemples de la classe de teoria d'àlgebra relacional: Obre la base de dades `set_theory.db` i prova les comandes del fitxer `set_theory.txt`, mirant d'endevinar els resultats abans d'executar-les.

Experiments amb SQLite, II

Importació de dades des de fitxers CSV o SQL

A la carpeta `/home/public/dabd/02accounts` del servidor `ubiwan.epsevg.upc.edu` trobaràs el material necessari.

- Crea una nova base de dades:

  `sqlite3 accounts1.db`
- Crea una taula `accounts` amb aquests camps:

  `(acc_id int, type char(1), balance real, owner text, owner_id int, phone int, address text)`
- Mira el contingut del fitxer `accounts.txt` que conté les dades a inserir i esbrina com usar-lo per afegir les dades a la taula anterior.
- Crea una nova base de dades:

  `sqlite3 accounts2.db`
- Mira el contingut del fitxer `accounts.sql` que conté instruccions SQL amb la taula a crear i les dades a inserir i esbrina com usar-lo. Comprova que el resultat és el mateix que important el fitxer .txt.

Comptes bancaris

Dels personatges del Quixot

Considerem un context d'activitat bancària representat mitjançant una única relació, amb els següents atributs:

- Número d'un compte bancari.
- Tipus del compte (corrent, estalvi, llibreta...).
- Saldo.
- Nom del titular del compte (un mateix compte pot tenir diferents titulars).
- Direcció, telèfon i NIF del titular.

Alguns titulars apareixen amb noms diferents; sabem que es tracta del mateix personatge quan el seu NIF coincideix.

El telèfon és fixe, no mòbil. Per tant, persones que viuen en la mateixa direcció (com Sancho Panza i la seva muller Teresa) tenen també el mateix telèfon.

Consultes exemple

Sobre la taula de comptes

Construeix consultes que corresponen a:

- DNI de tots els titulars, sense repeticions.
- DNI de Gaspar Gregorio.
- Comptes amb saldo superior a 1000.
- Comptes que no són de tipus 'L'.
- Saldo disponible complert per cadascun dels titulars (sumant tots els seus comptes), sense repeticions.
- Nom i telèfon sense repeticions a on el DNI sigui el d'Alonso Quijano (complicadeta, caldrà fer alguna subquery. 2 tuples).
- Parells de noms de la mateixa persona, identificada per DNI, sense repeticions (complicadeta, caldrà fer joins. 5 tuples).
- Compte amb el saldo major (és el 119774916201 amb 9818.59 eur).
- Compte amb el saldo menor (és el 171174310952 amb 28.89 eur).
- Noms de titulars sense repeticions que comencen amb "Caballero de" (4 tuples).

Consultes exemple 2

Sobre la taula de comptes

Construeix consultes que corresponen a:

- Titular que té el major saldo sumant tots els seus comptes (és el 6435323 amb 17351.02 eur).
- Titular que té el menor saldo sumant tots els seus comptes (és el 6152436 amb 687.78 eur).
- Saldo total i saldo mig de tots els comptes del banc arrodonit a 2 decimals (106118.38 eur i 2210.80 eur respectivament).
- DNI de tots els titulars, sense repeticions, amb el número de comptes que té cada titular.
- Llistat amb el número de titulars amb un compte, número de titulars amb dos comptes, ... (Ha de donar 1—8, 2—4, 3—10, 4—1, 6—1, 7—1)
- Comptes, sense repeticions, amb el número de titulars que té cada compte.
- Llistat amb el número de comptes amb un titular, número de comptes amb dos titulars, ... (Ha de donar 1—33, 2—15)

Altres operacions

Sobre la taula de comptes

Construeix instruccions SQL que corresponen a:

- Incementa amb 100 eur d'interessos el saldo dels comptes de tipus "L".
- Intercanvia els tipus "L" i "C" de tots els comptes.
- Al titular que tingui el major saldo sumant tots els seus comptes del banc li regalem un nou compte de tipus 'C' amb un saldo de 300 eur. El número del compte nou el calcularem incrementant el número del compte més gran de tot el banc.
- Crea una vista que mostri els comptes amb el seu saldo sense duplicitats.
- Altres que tu mateix t'inventis!

Sessió 2

- Introducció a MySQL
- Importació i exportació a/des de MySQL
- Introducció a PostgreSQL
- Importació i exportació a/des de PostgreSQL

MySQL

Repetim amb un gestor SQL diferent

- Connecta't a `ubiwan.epsevg.upc.edu`:
  `ssh username@ubiwan.epsevg.upc.edu` però amb el teu usuari de Linux.
- Connecta't a MySQL amb el client CLI oficial:
  `mysql -u username -p`
  però amb el teu usuari `est_xxxxxxxx` i amb la contrasenya `dB.xxxxxxxx` quan te la demani (a on `xxxxxxxx` són les 8 xifres del teu dni canviant la primera xifra per una lletra)
- `\h` per l'ajuda
- `\q` per sortir
- `show databases;` per llistar les bases de dades
  només pots accedir a la que té el mateix nom que el teu usuari `est_xxxxxxxx`
- `\u databasename` per canviar de base de dades
- `show tables;` per llistar taules i vistes
- `desc table/view;` per veure l'esquema d'una taula o vista

MySQL II

Provem SQL sobre MySQL

- Ja pots usar SQL al teu gust: Fes:

  un `CREATE TABLE pets...` que permeti guardar el nom, data de naixement, tipus (gat, gos, conill, ...) i pes de mascotes, després mirat el seu esquema i sobre d'aquesta taula creada fes:

  uns quants `INSERT` per inserir varis gats i gossos

  uns `SELECT`. Per ex. per comptar animals de cada tipus

  un `UPDATE` per actualitzar el pes

  un `DELETE` per eliminar un que ha passat a millor vida

  un `CREATE VIEW...` amb el pes mitjà segons el tipus de mascota: Gat 5.45, Gos 40.5

  uns `SELECT` sobre la vista

  un `DROP VIEW`.

  un `CREATE VIEW...` amb el nom i edat (anys) de cada mascota: Gina 6.0493, Momo 7.7096, Mica 1.6274

MySQL III

Importació i exportació de SQL

- Pots importar SQL amb:

  `mysql -u username -p databasename < data.sql`

  1) Importa el fitxer `accounts.sql` però alguna cosa no sortirà bé, molta precaució...

  2) Destrueix la taula mal creada, arregla el problema i torna a importar el fitxer SQL fins que funcioni bé.

- Pots exportar SQL amb (el nom de la taula és opcional):

  `mysqldump -u username -p databasename [tablename] > data.sql`

  1) Exporta la taula de mascotes al fitxer `pets.sql`. Si vols poder importar-la en altres SGBD pot ser convenient usar l'opció `--compatible=ansi`.

MySQL IV

Connexions remotes

- Instal·la't al portàtil el client CLI mysql i connecta't remotament a la teva base de dades del servidor ubiwan:

  `mysql -h ubiwan.epsevg.upc.edu -P 3306 -u username -p`

  El servidor MySQL escolta per defecte pel port 3306 a no ser que es configuri diferent.
- Instal·la't al portàtil un client gràfic (GUI) per MySQL, per exemple el client oficial MySQL Workbench i connecta't remotament a la teva base de dades del servidor ubiwan i prova les opcions que facilita.

PostgreSQL

Repetim amb un tercer gestor SQL diferent

- Connecta't a `ubiwan.epsevg.upc.edu`:
  `ssh username@ubiwan.epsevg.upc.edu` però amb el teu usuari de Linux.
- Connecta't a PostgreSQL amb el client CLI oficial:
  `psql -U username`
  però amb el teu usuari `est_xxxxxxxx` i amb la contrasenya `dB.xxxxxxxx` quan te la demani (a on `xxxxxxxx` són les 8 xifres del teu dni canviant la primera xifra per una lletra)
- `\h` per l'ajuda sintaxis SQL, `\?` per l'ajuda comandes CLI
- `\q` per sortir
- `\l` per llistar les bases de dades
  només pots accedir a la que té el mateix nom que el teu usuari `est_xxxxxxxx`
- `\c databasename` per canviar de base de dades
- `\d` per llistar taules, vistes, sequències
- `\d table/view` per veure l'esquema d'una taula o vista

PostgreSQL II

Provem SQL sobre PostgreSQL

- Ja pots usar SQL al teu gust: Fes:

  un `CREATE TABLE movies...` que permeti guardar el nom, any, director i puntuació de pel·lícules, després mirat el seu esquema i sobre d'aquesta taula creada fes:

  uns quants `INSERT` per inserir vàries pel·lícules de Stanley Kubrick i de Quentin Tarantino

  uns `SELECT`. Per ex. per obtenir totes les pel·lícules ordenades per director ascendent i les del mateix director per puntuació descendent

  un `CREATE VIEW...` amb la mitjana de la puntuació i l'any de la primera pel·lícula de cada director

  uns `SELECT` sobre la vista

  un `UPDATE` per canviar alguna puntuació

PostgreSQL III

Importació i exportació de SQL

- Pots importar SQL amb:

  `psql -U username databasename < data.sql`

  1) Importa el fitxer `accounts.sql` però alguna cosa no sortirà bé, molta precaució...

  2) Destrueix la taula mal creada, arregla el problema i torna a importar el fitxer SQL fins que funcioni bé.

  3) Importa el fitxer `pets.sql` exportat des de MySQL, però no sortirà bé. Caldrà canviar les cometes 'per cometes" i eliminar algunes coses específiques de MySQL.

- Pots exportar SQL amb (el nom de la taula és opcional):

  `pg_dump -U username databasename [-t tablename] > data.sql`

  Format més compatible afegint opcions: `--no-tablespaces --no-owner --no-acl --column-inserts`

  1) Exporta la taula de pel·lícules al fitxer `movies.sql` i després la importes a MySQL.

PostgreSQL IV

Connexions remotes

- Instal·la't al portàtil el client CLI psql i connecta't remotament a la teva base de dades del servidor ubiwan:

  `psql -h ubiwan.epsevg.upc.edu -p 5432 -U username`

  El servidor Postgres escolta per defecte pel port 5432 a no ser que es configuri diferent.
- Instal·la't al portàtil un client gràfic (GUI) per PostgreSQL com per exemple PgAdmin 4 i connecta't remotament a la teva base de dades del servidor ubiwan i prova les opcions que facilita.

Sessió 3

- Information schema del SGBD
- Permisos del SGBD
- Normalització: Creació de taules normalitzades
- Consultes en una B.D. normalitzada

information_schema del SGBD

El propi SGBD guarda la informació interna dins de taules/vistes

- A MySQL podem connectar a la bd `information_schema`:

  SHOW databases;

  \u information_schema

  SHOW tables;

  SELECT table_name, table_schema FROM tables;

  SELECT table_schema, column_name, column_type FROM columns WHERE table_name='accounts';

- A PostgreSQL ho trobarem dins de l'esquema (és similar a un espai de noms) anomenat `information_schema`:

  \dnS

  SELECT table_name, table_schema FROM information_schema.tables;

  SELECT udt_catalog, column_name, udt_name FROM information_schema.columns WHERE table_name='accounts';

information_schema del SGBD

Exercicis

Construïu a MySQL i a PostgreSQL queries que obtingin:

- les taules que tenen més de 50 registres (a PostgreSQL no és possible),
- quantes taules visibles s'anomenen 'accounts',
- quantes columnes són de qualsevol varietat de tipus 'int',
- quantes columnes són de qualsevol varietat de tipus 'char',
- quantes vistes hi ha

Permisos

GRANT i REVOKE

- Podem donar permisos amb la comanda GRANT:
  GRANT permís ON database.tablename TO username;
- Podem treure permisos amb la comanda REVOKE:
  REVOKE permís ON database.tablename FROM username;

On el permís pot ser, entre d'altres, SELECT, INSERT, UPDATE, DELETE, ALL... Podem indicar que el permís només afecti a algunes columnes, per exemple GRANT permís(columnes) ON ... Canvia permisos de la taula de pel·lícules a PostgreSQL (no teniu permissos per gestionar els permissos al MySQL d'ubiwan):

- Treu el permís d'eliminar i prova de fer un DELETE.
- Treu el permís d'actualitzar i prova de fer un UPDATE.
- Treu el permís d'inserir i prova de fer un INSERT.
- Treu el permís de consultar i prova de fer un SELECT.
- Torna a donar tots els permisos a la taula i prova si ja pots fer SELECT, INSERT, UPDATE i DELETE.
- Treu el permís de consultar el camp nom de les pel·lícules. Cal fer un REVOKE SELECT i un GRANT SELECT (columnes).

Normalització

Esquemes relacionals amb claus

Treballarem el problema 3 dels comptes bancaris (sense variants).

- Llista les dependències funcionals i normalitza a 3FN.
- Defineix els esquemes relacionals amb les claus primàries, alternatives i foranes.
- Crea les taules corresponents a MySQL/PostgreSQL.
- Traspassa tota la informació de la taula `accounts` sense normalitzar a aquestes noves taules normalitzades.
- Torna a fer les consultes SQL de la 1a sessió sobre aquestes noves taules.

Sessió 4

- Funcionament de les claus foranes
- Connexió d'un programa Python a un SGBD
- SQL injection

Claus foranes

Comprovem el funcionament de les claus foranes sobre SQLite

- Crea les taules `tec`, `maq`, `evsup` i `supervisio` (pots usar els CREATE TABLE del fitxer `maqfact_foreign_keys.sql`).
- Fixat que hi ha claus foranes, i que les clàusules de propagació de modificacions són CASCADE pels UPDATEs i NO ACTION pels DELETEs.
- Comprova si es poden afegir dades a `supervisio` sense haver afegit res a `tec`, `maq` o `evsup`.
- Recorda que la comprovació de claus foranes està desactivada a SQLite fins que no facis `PRAGMA foreign_keys = ON;`
- Fes pas a pas les queries del fitxer `maqfact_foreign_keys.sql`, pensant quin resultat obtindràs abans d'executar cada query.

Connectar un programa a una base de dades

Accedir a SQLite des d'un programa Python usant la llibreria sqlite3

- Mira el codi del programa `users_sqlite_inj.py`. La forma de construir les queries és molt inadecuada, permet fer SQL injection.
- Executa'l vàries vegades per crear una taula, afegir varis usuaris, llistar-los i comprovar si un usuari-contrasenya és correcte (l'opció s permet veure les queries que es construeixen):

```
./users_sqlite_inj.py -h
./users_sqlite_inj.py -is
./users_sqlite_inj.py -as
./users_sqlite_inj.py -ls
./users_sqlite_inj.py -s
```

SQL injection

Com prevenir un atac

- Afegeix un nou usuari (`./users_sqlite_inj.py -as`) amb la següent contrasenya:

  `');drop table users;--`

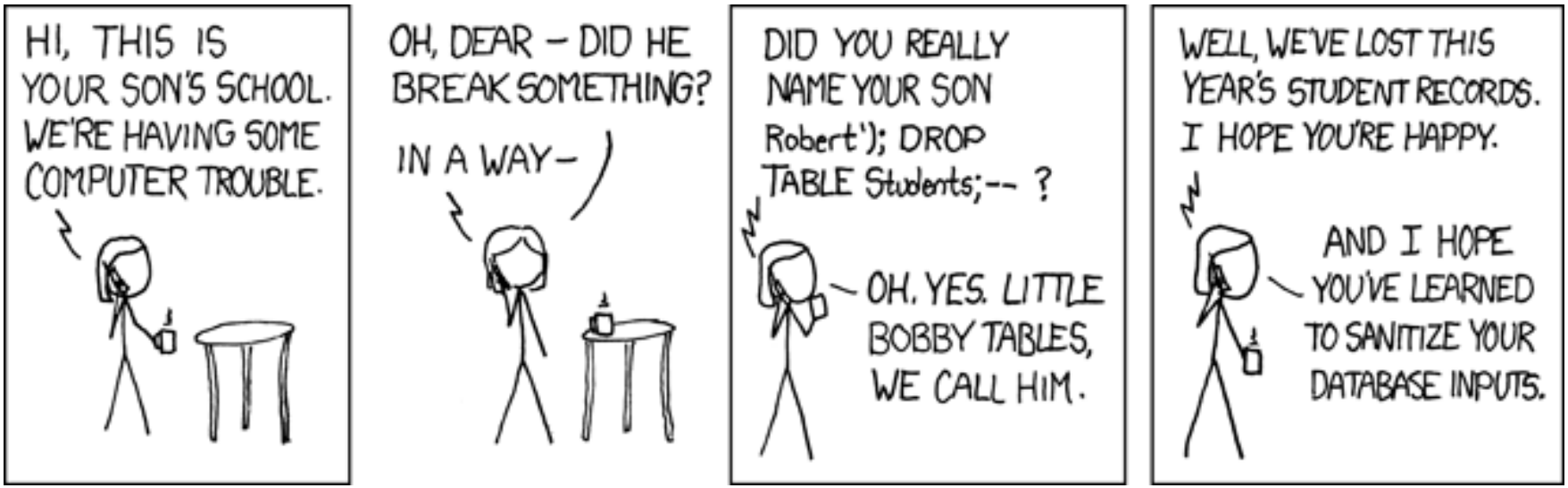
  i llista posteriorment els usuaris: CATÀSTROFE!
- Torna a crear la taula i afegir usuaris. Després elimina l'usuari (`./users_sqlite_inj.py -ds`) que té per nom:

  `' OR username LIKE '%`

  i llista posteriorment els usuaris: CATÀSTROFE!
- Mira el codi del programa `users_sqlite_no_inj.py` per veure com es resol el problema (usant el placeholder ?), executa'l varis cops i intenta fer SQL injection.

SQL injection II

Sanitize your database inputs



Font: https://xkcd.com/327/

Connectar un programa a una base de dades II

Provant PostgreSQL i MySQL

- Adapta el programa Python per accedir a PostgreSQL usant la llibreria psycopg.
- Et caldrà fer molts pocs canvis, ja que ambdós llibreries implementen el protocol DB API 2.0 de Python.
- Configura la connexió per accedir a la teva B.D. postgres d'ubiwan, indicant host, user, password i database.
- Executa'l vàries vegades per crear una taula, afegir varis usuaris, llistar-los i comprovar si un usuari-contrasenya és correcte.
- Prova de fer SQL injection, no hauria de ser possible si has usat els placeholders de psycopg.
- Si vols pots fer el mateix amb MySQL.