

Pràctica 3

CREACIÓ I INSERCIÓ
DE DADES A LA BASE
DE DADES

1. ÍNDEX

1. ÍNDEX.....	1
2. INTRODUCCIÓ	2
3. INSTRUCCIONS	3
3.1. SELECCIÓ DEL SISTEMA GESTOR (SGBD)	3
3.2. CREACIÓ DE L'ESTRUCTURA AMB SQL.....	3
3.3. INSERCIÓ DE DADES DE PROVA	4
3.4. VALIDACIÓ DEL FUNCIONAMENT AMB SELECT	4
4. RESULTAT ESPERAT	7
4.1. RÚBRICA D'AVALUACIÓ	Error! No s'ha definit el marcador.
5. CONCLUSIONS	8

2. INTRODUCCIÓ

Després d'haver dissenyat l'estructura de la base de dades durant l'Sprint 2, és el moment de posar-la en pràctica mitjançant la seva creació real. En aquest Sprint 3, treballareu amb **SQLite**, un sistema gestor de bases de dades lleuger, senzill i altament compatible amb la Raspberry Pi i amb scripts escrits en Python.

SQLite ha estat escollit perquè:

- No requereix instal·lació d'un servidor extern.
- Emmagatzema les dades en un únic fitxer `.db`, la qual cosa facilita la seva integració i transport.
- És ideal per a sistemes incrustats com la Raspberry Pi, ja que ofereix un rendiment eficient i una configuració mínima.
- Funciona perfectament amb Python a través de la biblioteca estàndard `sqlite3`.

Els objectius generals d'aquest Sprint són:

- **Crear la base de dades des de zero** mitjançant instruccions SQL, seguint l'esquema que vau definir en l'Sprint 2.
- **Implementar la base de dades en SQLite**, adaptant-la al sistema real de registre que després connectareu amb la Raspberry Pi i el lector NFC.
- **Afegir dades fictícies i coherents** per fer proves de funcionament i validar les relacions entre taules.
- **Exportar el fitxer `.sql`** amb la creació i inserció de les dades.
- **Redactar un informe tècnic breu** que expliqui l'estructura de la base de dades, les decisions preses i els possibles usos pràctics.

Aquest pas és fonamental per consolidar la connexió entre el disseny conceptual i la implementació real d'una base de dades funcional, preparant el sistema per al seu ús pràctic en el dispositiu Raspberry Pi.

3. INSTRUCCIONS

3.1. APROXIMACIÓ A SQLite

En aquest projecte treballarem exclusivament amb **SQLite** com a sistema gestor de bases de dades (SGBD), per les seves característiques lleugeres, la seva facilitat d'ús i la seva compatibilitat total amb la Raspberry Pi i amb llenguatges com Python.

SQLite és un SGBD **basat en fitxers** que no requereix cap servidor ni configuració complexa. Tota la base de dades s'emmagatzema en un únic fitxer `.db`, que es pot moure, copiar o integrar fàcilment dins d'un projecte. Aquesta portabilitat i simplicitat el converteixen en una eina ideal per a entorns educatius i per a dispositius embarcats com la Raspberry Pi.

Tot i ser lleuger, SQLite admet estructures complexes, clau primària i forana, tipus de dades, consultes SQL avançades i integració directa amb aplicacions mitjançant biblioteques oficials. A més, forma part de la biblioteca estàndard de Python, cosa que facilita la seva incorporació sense instal·lar mòduls addicionals.

Aquesta etapa del projecte us permetrà:

- Treballar amb **codi SQL real**, totalment compatible amb altres SGBD més grans com MySQL.
- Crear i gestionar una base de dades funcional sense dependre de cap entorn gràfic ni servidor.
- Centrar-vos en la **lògica del model de dades** i en el **funcionament pràctic** de les consultes i les relacions.

SQLite serà, doncs, l'eina ideal per connectar el vostre disseny amb el sistema final que s'executarà sobre la Raspberry Pi, permetent proves àgils i una integració directa amb els scripts de lectura de targetes NFC.

3.2. CREACIÓ DE L'ESTRUCTURA AMB SQL

Una vegada hàgeu triat el vostre SGBD, el següent pas és transformar el model conceptual definit en l'Sprint 2 en una estructura física mitjançant comandes SQL. Obriu el vostre editor o client SQL i comenceu definint, si cal, la base de dades amb una instrucció `CREATE DATABASE nom_bbdd;`

A continuació, per a cada entitat del vostre projecte (per exemple, usuaris, targetes, accessos, dispositius), escriviu una sentència `CREATE TABLE` que incloga:

1. Nom de la taula, seguit de l'obertura de parèntesis.
2. Definició de col·umnes, indicant el nom del camp, el tipus de dada (INTEGER, TEXT, DATETIME, BOOLEAN, etc.), i si pot ser NOT NULL o admet DEFAULT.
3. Clau primària (PRIMARY KEY), perquè cada registre tingui un identificador únic, sovint un camp id amb AUTOINCREMENT en SQLite
4. Claus foranes (FOREIGN KEY), que enllacen una taula amb una altra; cal especificar la referència (REFERENCES taula(camp)) i opcionalment comportaments com ON DELETE CASCADE.

Com a exemple, vos adjuntem un exemple de taula on es poden veure els elements anteriors:

```
CREATE TABLE accessos (  
    id_acces INTEGER PRIMARY KEY AUTOINCREMENT,  
    id_usuari INTEGER NOT NULL,  
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (id_usuari) REFERENCES usuaris(id_usuari)  
);
```

Assegureu-vos de respectar l'ordre lògic (primer les taules sense dependències, després les que en tenen) i de desar el vostre script en un fitxer .sql. D'aquesta forma, podreu executar-lo de cop i recrear tota la base de dades en qualsevol moment, cosa que facilita proves, còpies de seguretat i correccions.

3.3. INSERCIÓ DE DADES DE PROVA

Un cop hegeu creat la vostra base de dades i hàgiu inserit els registres de prova, és imprescindible comprovar que tot funciona tal com esperàveu. Per això, heu d'executar diverses instruccions SELECT que us permetin:

1. Verificar la coherència dels enllaços: Per exemple, feu una consulta que unisca la taula d'accessos amb la d'usuaris per comprovar que cada registre d'accés correspon a un usuari vàlid.
2. Validar la integritat de les dades: Comproveu que no hi haja accessos sense usuari (ni usuaris sense identificador, si el vostre model ho impedeix).
3. Simular consultes reals: Quin usuari ha fet més entrades en un dia concret? Quants accessos s'han registrat per espai o dispositiu?
4. Detectar possibles errors: Si una taula retorna menys files de les esperades, potser alguna clau forana no està ben configurada o, reviseu els tipus de dades: si una data apareix nul·la o incorrecta, potser el camp no és NOT NULL o no té un DEFAULT adequat.

Graveu-vos les vostres consultes i els seus resultats, ja que formaran part de l'evidència del vostre procés de validació.

3.4. VALIDACIÓ DEL FUNCIONAMENT AMB SELECT

Perquè qualsevol company o futur desenvolupador pugui entendre com heu treballat, cal lliurar un informe tècnic breu (màxim 3 pàgines) que inclogui:

1. Estructura general del fitxer .sql: Resum dels fitxers i comandes principals (CREATE TABLE, INSERT INTO).
2. Exemple comentat: Inclou almenys una línia d'inserció de prova amb un comentari que expliqui el seu objectiu.
3. Captures de pantalla o evidències: Afegeix una imatge on es pugui veure el resultat d'una consulta SELECT satisfactòria.
4. Observacions i problemes trobats: Si heu hagut d'ajustar alguna instrucció o corregir errors de tipus de dades, descriviu breument què ha passat i com ho heu solucionat.

PRÀCTICA 3- CREACIÓ I INSERCIÓ DE DADES A LA BASE DE DADES

Aquest document ha de ser clar i concís, i complementa el vostre fitxer .sql perquè el projecte siga fàcil de mantenir i d'integrar en la Raspberry Pi.

4. PREPARACIÓ DE LA API

Per tal de poder insertar les dades a la base de dades SQLite des de la Raspberry Pi, primer heu de crear una API que farà de pont entre la lectura de targetes NFC i la base de dades. Aquesta API serà un programa escrit en Python que utilitzarà el microframework Flask per gestionar les peticions HTTP i comunicar-se amb la base de dades.

A continuació us expliquem els passos que heu de seguir per preparar i posar en marxa aquesta API al vostre ordinador utilitzant Visual Studio Code (VS Code):

4.1. INSTAL·LACIÓ DELS REQUISITS

Assegureu-vos que teniu instal·lat python 3. Podeu descarregar-lo a [python.Org](https://python.org). Instal·leu Visual Studio Code des de code.visualstudio.com. Des de la terminal integrada de VS Code, instal·leu el paquet Flask amb la comanda:

```
bash Copiar Editar pip install flask
```

4.2. CREACIÓ DEL PROJECTE I FITXER DE L'API

Creeu una carpeta per al projecte, per exemple `api_visites`. Dins aquesta carpeta, creeu un fitxer anomenat `api.py`. Copieu-hi el codi base de la API, que connecta amb la base de dades SQLite i permet afegir registres amb una petició POST.

4.4. EXECUTAR L'API

Obriu el projecte a VS Code i Obriu el fitxer `api.py`. Aneu a la terminal integrada i executeu:

```
python api.py
```

L'API quedarà escoltant a la IP local, al port 5000. Ara, la Raspberry Pi podrà enviar dades a l'API i aquesta s'encarregarà d'insertar-les a la base de dades SQLite.

4.5. RESUM

Amb aquesta API podreu:

- Rebre dades de la Raspberry Pi mitjançant peticions HTTP POST.
- Afegir registres a la base de dades SQLite de manera segura.
- Preparar el vostre sistema per a la integració amb el lector NFC i la gestió de visites.

5. RESULTAT ESPERAT

En finalitzar l'Sprint 3, cada grup haurà d'entregar:

- Fitxer SQL: Un sol arxiu .sql que incloga totes les comandes CREATE DATABASE (si s'escau), CREATE TABLE, definició de claus, i les instruccions INSERT INTO amb les dades de prova.
- Informe tècnic breu: Document (PDF, Word o Markdown) de màxim 3 pàgines amb la documentació descrita a la secció 3.4.
- Evidències de validació: Captures de pantalla o exportacions de les consultes SELECT que demostrin que la base de dades funciona correctament i que les relacions s'han creat de forma adequada.

D'aquesta manera, haureu demostrat que sou capaços de portar el disseny conceptual del vostre projecte fins a una implementació real, amb totes les dades necessàries per al següent pas: la connexió amb la Raspberry Pi i el lector NFC.

6. CONCLUSIONS

Aquest tercer sprint ha marcat una fita clau en el desenvolupament del vostre projecte: passar del disseny teòric a la implementació pràctica de la base de dades. Fins ara, havíeu pensat i planificat quines entitats necessitava el vostre sistema, com s'estructurarien les dades i quines relacions hi hauria entre elles. Ara, heu après a fer-ho realitat mitjançant comandes SQL dins d'un sistema gestor de bases de dades real.

Durant aquest procés, heu treballat habilitats fonamentals per a qualsevol projecte tecnològic:

- Traducció del disseny conceptual a codi executable.
- Ús correcte del llenguatge SQL, tant per a crear estructures com per a inserir dades.
- Validació i comprovació del funcionament mitjançant consultes.
- Documentació tècnica clara i funcional, que facilita la comunicació dins del grup i la futura integració amb la Raspberry Pi.