

# 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 .....	5
4.1. RÚBRICA D'AVUACIÓ .....	<b>Error! No s'ha definit el marcador.</b>
5. CONCLUSIONS .....	6

### 2. INTRODUCCIÓ

Després d'haver dissenyat la vostra base de dades en l'Sprint 2, és moment de posar-la en pràctica. En aquest Sprint 3 haureu de crear l'estructura de la base de dades utilitzant SQL i un sistema gestor de bases de dades (SGBD) com ara SQLite, MySQL o MariaDB.

També inserireu dades de prova (coherents i reals) per tal de validar el funcionament del vostre sistema. Aquesta etapa fa de pont entre la teoria i la pràctica, i prepara la base per connectar-se amb la Raspberry Pi i el lector NFC.

Per tant, els objectius generals que heu de tindre en compte són:

- Crear la base de dades programada de zero, seguint el disseny de l'Sprint 2.
- Implementar-la en un sistema gestor real.
- Inserir dades fictícies de prova per verificar el seu funcionament.
- Generar el fitxer SQL i un informe tècnic breu.

### 3. INSTRUCCIONS

#### 3.1. SELECCIÓ DEL SISTEMA GESTOR (SGBD)

Abans de començar a escriure cap línia de codi SQL, primer heu de triar el sistema gestor de bases de dades que millor s'adapte al vostre entorn de treball i als vostres objectius.

Si voleu una solució lleugera i fàcil d'instal·lar directament sobre la Raspberry Pi o sobre el vostre ordinador sense moltes dependències, **SQLite** és la millor opció: no requereix instal·lació de servidors ni configuracions complexes, i el fitxer .db resultant és molt portàtil.

En canvi, si preferiu acostumar-vos a un entorn més “professional” i voleu practicar amb eines de servidor o interfícies gràfiques com XAMPP o DBeaver, podeu optar per **MySQL** o **MariaDB**. Aquests SGBD us permetran experimentar la creació d'usuaris, permisos i connexions remotes, a més de gestionar bases de dades més robustes.

Penseu en l'escala del projecte i en les vostres necessitats: si la vostra intenció és desplegar la solució en un entorn real de servidor, potser MySQL o MariaDB us resultaran més adequats; però si voleu centrar-vos ràpidament en la lògica i l'estructura SQL sense complicar-vos amb servidors, SQLite us estalviarà temps i feina.

#### 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 columnes, 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 o `AUTO_INCREMENT` en MySQL.
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)  
);
```

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

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 permeten:

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. Entorn i SGBD escollit: Explica breument per què has triat SQLite, MySQL o MariaDB, i on has executat el teu script (terminal, DB-Browser, DBeaver, etc.).
2. Estructura general del fitxer .sql: Resum dels fitxers i comandes principals (CREATE TABLE, INSERT INTO).
3. Exemple comentat: Inclou almenys una línia d'inserció de prova amb un comentari que expliqui el seu objectiu.
4. Captures de pantalla o evidències: Afegeix una imatge on es pugui veure el resultat d'una consulta SELECT satisfactòria.
5. Observacions i problemes trobats: Si heu hagut d'ajustar alguna instrucció o corregir errors de tipus de dades, descriu breument què ha passat i com ho heu solucionat.

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

#### 4. 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.

### 5. 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.
- Elecció d'un entorn tècnic adequat (SQLite, MySQL o MariaDB) segons les necessitats del projecte.
- 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.