Dunaújvárosi Egyetem Bánki Donát Technikum

Projekt feladat dokumentáció

Tartalom

Tantárgy neve: Mikrokontrollerek

Projekt tervezői: Fogas Bence

Projekt címe: Videótéka

Osztály: 12.B

Dátum: 2025.05.23.

Az ötlet rövid leírása:

Ez a projekt egy videotéka működését támogató relációs adatbázisrendszer megtervezését és létrehozását valósítja meg MySQL környezetben. Az adatbázis célja, hogy nyilvántartsa a kölcsönözhető filmeket, a kölcsönzéseket, valamint a ügyfelek **és** dolgozók adatait.

Működési elv

Az adatbázis célja, hogy pontosan nyomon kövesse a kölcsönzéseket. Automatizálással és lekérdezésekkel segít elkerülni az emberi hibákat, például:

- Dupla kölcsönzést
- Elveszett filmek nyomon nem követését
- Ügyfelek tartozásának figyelmen kívül hagyását

Film adatainak rögzítése

- A dolgozó felviszi a rendszerbe az új filmeket a filmek táblába.
- Megadja a film címét, műfaját (kategoriak), formátumát (formatumok) és megjelenési évét.

Ezzel elérhető, hogy minden film visszakereshető legyen.

Ügyfél regisztráció

- A kölcsönző személy adatait (név, cím, email) rögzítik az ugyfelek táblába.

Minden kölcsönzés egy konkrét ügyfélhez kötődik.

Kölcsönzés rögzítése

Amikor egy ügyfél kikölcsönöz egy filmet:

- Új sor kerül a kolcsonzesek táblába.
- Rögzítésre kerül:
    - **film ID** (melyik filmet vitték el)
    - **ügyfél ID** (ki vitte el)
    - **dolgozó ID** (ki adta ki)

- **kölcsönzés dátuma**
- **visszahozás határideje** (pl. +3 nap)

A film státusza kolcsonozheto = FALSE-ra változtatható, ha nem kölcsönözhető másnak.

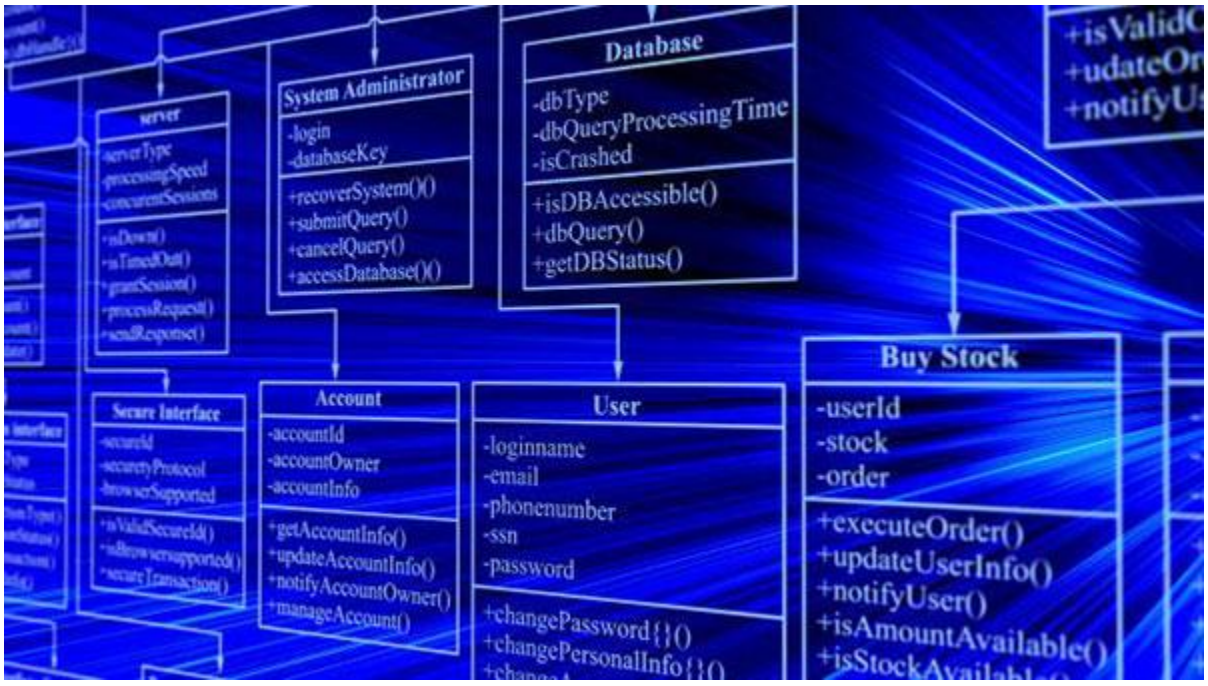
Film visszahozása

- Amikor az ügyfél visszahozza a filmet, a kolcsonzesek táblában a **visszahozva** mezőbe beíródik az aktuális dátum.
- A film ismét kölcsönözhető lesz (kolcsonozheto = TRUE).

Lekérdezések, riportok

Az adatbázisból könnyedén lekérdezhető például:

- Mely filmek vannak éppen kikölcsönözve?
- Mely ügyfelek nem hozták vissza időben a filmeket?
- Hány kölcsönzést bonyolított egy dolgozó egy hónapban?



MySQL adatbázis

```sql
-- Adatbázis létrehozása
CREATE DATABASE videoteka;
USE videoteka;

-- Kategóriák (pl. akció, vígjáték)
CREATE TABLE kategoriak (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nev VARCHAR(50) NOT NULL
);

-- Formátumok (DVD, Blu-ray, stb.)
CREATE TABLE formatumok (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tipus VARCHAR(30) NOT NULL
);

-- Filmek
CREATE TABLE filmek (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cim VARCHAR(100) NOT NULL,
    kategoria_id INT,
    formatum_id INT,
    megjelenes_eve YEAR,
    kolcsonozheto BOOLEAN DEFAULT TRUE,
    FOREIGN KEY (kategoria_id) REFERENCES kategoriak(id),
    FOREIGN KEY (formatum_id) REFERENCES formatumok(id)
);

-- Ügyfelek
CREATE TABLE ugyfelek (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nev VARCHAR(100) NOT NULL,
    email VARCHAR(100),
    telefonszam VARCHAR(20),
    cim TEXT
);

-- Dolgozók
CREATE TABLE dolgozok (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nev VARCHAR(100) NOT NULL,
    beosztas VARCHAR(50)
);

-- Kölcsönzések
CREATE TABLE kolcsonzesek (
    id INT AUTO_INCREMENT PRIMARY KEY,
    film_id INT NOT NULL,
    ugyfel_id INT NOT NULL,
    dolgozo_id INT,
    kolcsonzes_datuma DATE NOT NULL,
    visszahozas_hatarideje DATE NOT NULL,
    visszahozva DATE,
    FOREIGN KEY (film_id) REFERENCES filmek(id),
    FOREIGN KEY (ugyfel_id) REFERENCES ugyfelek(id),
    FOREIGN KEY (dolgozo_id) REFERENCES dolgozok(id)
);
```

Fejlesztési lehetőségek

1. Webes felület készítése

Készíthetsz egy felhasználóbarát webalkalmazást, ahol:

- ügyfelek listázhatók, szerkeszthetők,
- filmek böngészhetők és kereshetők,
- kölcsönzések rögzíthetők és visszahozhatók.

Technológiák:
Frontend – HTML/CSS, JavaScript (pl. React vagy Vue)
Backend – PHP / Node.js / Python (Flask, Django)
Adatbázis – MySQL (már adott)

---

2. Asztali alkalmazás fejlesztése

Készíthetsz Windows vagy Linux környezetre futó asztali programot:

- például C#, Java, Python (Tkinter, PyQt),
- egyszerű kezelőfelülettel rendelkező kölcsönző program.

---

3. Mobilalkalmazás

Mobilos kliens készítése, ahol ügyfelek:

- böngészhetik a filmeket,
- látják saját kölcsönzéseiket,
- akár előre foglalhatnak egy filmet.

Technológiák: Flutter, React Native, Kotlin, Swift.

---

4. Riasztás és figyelmeztetés rendszerek

Késleltetett visszahozások figyelése:

- email vagy SMS emlékeztető küldése,
- napi riport dolgozóknak a lejárt kölcsönzésekről.

---

5. Riportok és statisztikák

- Legtöbbet kölcsönzött filmek listája
- Legaktívabb ügyfelek
- Kölcsönzési trendek grafikonon

Készíthetők exportálható PDF vagy Excel formátumú riportok.

---

6. Jogosultságkezelés

- Külön szerepkörök (admin, dolgozó, vendég)
- Minden felhasználó csak a saját engedélyeihez fér hozzá

---

7. Online elérés / Cloud rendszer

- Távoli elérés böngészőből vagy mobilról
- Szerverre telepített adatbázis + API-k

---

8. Film állomány kezelése

- Képek és előzetes videók feltöltése
- Filmek fizikai példányszámainak nyilvántartása

---

9. Tesztadatok generálása

- Automatizált tesztadat-generálás a rendszer próbálásához

Tapasztalatok

Az első kihívást az jelentette, hogy hogyan bontsuk le a videotéka működését különálló adatbázis-táblákra. Kezdetben nehéz volt eldönteni, milyen kapcsolatokat és kulcsokat használjunk a filmek, ügyfelek és kölcsönzések között. A táblák közötti kapcsolatok (például film_id, ugyfel_id) kezelése során figyelni kellett a hivatkozási integritásra. A FOREIGN KEY kapcsolatok hibát adhattak, ha rossz sorrendben hoztuk létre a táblákat vagy nem létező rekordra hivatkoztunk. A kölcsönzési és visszahozási dátumok kezelése során felmerültek problémák például a NULL értékekkel (amikor egy film még nincs visszahozva), és a lekérdezések megfogalmazása is figyelmet igényelt. Például: mi történik, ha egy filmet egyszerre többen szeretnének kikölcsönözni? Kezdetben nem volt elégséges a kolcsonozheto mező, így felmerült az igény a példányszámok kezelésére is.

Az adatbázis logikus és alapos tervezése elengedhetetlen, még mielőtt egyetlen SQL-parancsot írunk. Fontos a valóság pontos leképezése: ha kihagyunk egy valós szituációt (pl. késés, hibás adatok), az utólag nehézkes lehet. A FOREIGN KEY kapcsolatok használata segít megőrizni az adatok konzisztenciáját, de csak akkor működnek jól, ha helyesen tervezzük meg őket.Érdemes kis tesztadatokkal próbálkozni az elején, hogy hamar észrevegyük a logikai hibákat.A projekt során jelentősen fejlődtek az SQL-ismereteink, különösen a JOIN-ok és a lekérdezések optimalizálása terén.

Önreflexió

A projekt elkészítése során megtapasztalhattam egy összetettebb adatbázis elkészítésének fortélyait, a munka során tapasztalatot szereztem a különböző táblák és a hozzájuk tartozó adatok létrehozásáról és kezeléséről, amely rendkívül hasznos.