

Dunaújvárosi Egyetem Bánki Donát Technikum

Projekt Feladat Dokumentáció

Projekt tervezője: Nagy Gergő

Projekt címe: Adatbázis kezelés

Osztály: 12.C

1. Bevezetés A relációs adatbázisok olyan rendszerek, amelyekben az adatokat táblákban tároljuk. Ezek az adatbázisok segítenek abban, hogy az információt rendszerezve, hatékonyan tároljuk és kezeljük. Ebben a dokumentumban egy egyszerű példán keresztül mutatjuk be az adatbázisok tervezését és használatát egy iskolai rendszerrel kapcsolatban.

2. Hogyan tervezünk meg egy adatbázist?

Az adatbázis elkészítése több lépésből áll:

2.1. Mi a cél? Először is meg kell határoznunk, hogy mire fogjuk használni az adatbázist. Példánkban egy iskola diákjait, tantárgyait és jegyeit szeretnénk nyilvántartani.

2.2. Az adatok felosztása (ER-diagram) A következő lépés az, hogy meghatározzuk, milyen adatokat kell tárolnunk, és hogyan kapcsolódnak egymáshoz. A mi példánkban három fő adatszoport van:

- **Diákok** (Student)
- **Tantárgyak** (Course)
- **Beiratkozások** (Enrollment) Ezek az adatok összekapcsolhatók, mert egy diák több tantárgyra is beiratkozhat, és egy tantárgyat több diák is felvehet.

2.3. Táblák kialakítása A diákokról, a tantárgyakról és a beiratkozásokról külön táblákat hozunk létre, hogy az információ rendszerezett legyen.

3. Alapvető adattípusok az adatbázisban

Minden adatbázisban különböző típusú adatokat tárolunk:

- **INTEGER:** Egész számok (pl. diákok azonosítója).
- **VARCHAR:** Szöveges adatok (pl. nevek, e-mail címek).
- **DATE:** Dátumok (pl. születési időpont).
- **BOOLEAN:** Igen/nem vagy igaz/hamis értékek.

Példa: Egy diák neve VARCHAR típusú lesz, az azonosítója pedig INTEGER.

4. Egy egyszerű iskolai adatbázis felépítése

4.1. Táblák struktúrája

- **Diákok tábla:** diákok adatai (ID, név, e-mail, születési dátum).
- **Tantárgyak tábla:** tantárgyak neve és kreditértéke.
- **Beiratkozások tábla:** egy diák melyik tantárgyra jelentkezett.

4.2. SQL parancsok

Táblák létrehozása:

```
CREATE TABLE Student (  
    student_id INT PRIMARY KEY,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    birth_date DATE,  
    email VARCHAR(100)  
);
```

Adatok hozzáadása:

```
INSERT INTO Student (student_id, first_name, last_name, birth_date, email)  
VALUES (1, 'Nagy', 'Ágnes', '2005-01-25', 'agnes.nagy@example.com');
```

Lekérdezés:

```
SELECT first_name, last_name FROM Student;
```

Módosítás:

```
UPDATE Student SET email = 'uj.email@example.com' WHERE student_id = 1;
```

Törlés:

```
DELETE FROM Student WHERE student_id = 1;
```

5. Adatbázis-kezelés és biztonság

Az adatokat nem csak tárolni kell, hanem biztonságosan is kezelni. Ennek egyik módszere a **tranzakciókezelés**, amely segít abban, hogy egy adatbázisműveletet vagy teljes egészében végrehajtunk, vagy visszavonjuk, ha hiba történik.

Példa egy tranzakcióra:

```
BEGIN TRANSACTION;  
INSERT INTO Student (student_id, first_name, last_name, birth_date, email)  
VALUES (2,'Német', 'Ákos', '200-02-13', 'akos.nemet@example.com');  
COMMIT;
```

Ha probléma történik, a tranzakciót vissza lehet vonni:

```
ROLLBACK;
```

magabiztosabbá váltam. Úgy érzem, hogy az itt megszerzett tudást a jövőben hatékonyan tudom majd alkalmazni különböző projektekben.

6. Mellékletek

6.1. További adatbázis-kezelési technikák

Az adatbázisok kezelése nemcsak a táblák létrehozásáról és az adatok tárolásáról szól. Vannak olyan plusz technikák is, amelyekkel az adatbázis gyorsabbá, biztonságosabbá és megbízhatóbbá tehető.

Néhány fontos téma, amit érdemes megismerni:

- **Indexelés és teljesítményoptimalizálás:**
Az indexek olyan speciális jelölők az adatbázisban, amelyek segítenek gyorsabban megtalálni az adatokat. Ez olyan, mintha egy könyvben a tartalomjegyzék segítene gyorsan megkeresni egy fejezetet.
- **Biztonsági mentések (backup) és visszaállítási (recovery) eljárások:**
Az adatvesztés elkerülése érdekében az adatbázisokról időnként másolatot (backupot) kell készíteni. Ha valami probléma történik (például véletlen törlés vagy rendszerhiba), a mentett adatokból vissza lehet állítani az adatbázist.
- **Jogosultságkezelés és felhasználói hozzáférések szabályozása:**
Az adatbázisokban nem mindenki férhet hozzá minden adathoz. Be lehet állítani, hogy ki mit láthat és mit módosíthat. Például egy diák csak a saját jegyeit nézheti meg, de egy tanár az összes diák eredményét láthatja.

Ezek az extra technikák segítenek abban, hogy az adatbázisok gyorsak, biztonságosak és jól működőek legyenek.

Önreflexió:

Ebben a dokumentumban megismerkedtem az adatbázisok alapjaival, a táblák létrehozásával és az alapvető SQL parancsokkal. Egy egyszerű iskolai rendszeren keresztül megtapasztaltam, hogyan lehet egy adatbázist megtervezni és kezelni. A munka során különösen érdekes volt látni, hogy az adatok rendszerezett tárolása mennyire megkönnyítheti az adminisztratív feladatokat. Bár kezdetben kihívást jelentett az SQL parancsok pontos szintaxisának elsajátítása, a gyakorlás során.