**Kérdésem:**

SQL-ben le tudom úgy menteni a felhasználók jelszavát, hogy

1. 'Személy' táblában: 'felh\_jelszo' [védett tábla legyen]
2. 'Felhasznalok' **táblában** a "jelszo\_masodik\_utolso\_szamjegye" **oszlopot**

Azaz nem egy helyen tárolva a bizalmas adatokat.

Ugyanakkor egy operátor láthassa a 2. és utolsó számjegyet, de a teljes jelszavát az ügyfélnek NE!

**Válasz:**

Igen, SQL-ben lehetséges megvalósítani:

1. **Használj két táblát**: Egyik tábla a felhasználók jelszavait tárolja biztonságosan, a másik pedig a jelszó második és utolsó számjegyét tartalmazza.
2. **Tárold a jelszavakat biztonságosan**: Használj erős jelszó hash-elést (pl. bcrypt) a jelszavak tárolásához a biztonsági tábla oszlopában.

**Az SQL Kód:**

CREATE TABLE Felhasznalok (

    felh\_id INT AUTO\_INCREMENT PRIMARY KEY,

    felh\_nev VARCHAR(20) UNIQUE NOT NULL, -- Egy f.nevből CSAK egy lehet az oszlopban!

    elofiz\_kat ENUM('0', '1', '2', '3'),

    jogositvany\_szama\_FK INT NOT NULL,

    jelszo\_masodik\_utolso\_szamjegye VARCHAR(2),

    FOREIGN KEY (jogositvany\_szama\_FK)

    REFERENCES Szemely(jogositvany\_szama)

);

CREATE TABLE Szemely (

    jogositvany\_szama VARCHAR(15) PRIMARY KEY,

    jogositvany\_ervenyesseg DATE NOT NULL,

    jogositvany\_lejarata DATE NOT NULL,

    V\_nev VARCHAR(35) NOT NULL,

    K\_nev VARCHAR(35) NOT NULL,

    Szig\_szam VARCHAR(15) NOT NULL,

    felh\_jelszo VARCHAR(10) NOT NULL, -- Hash-elt jelszó (python bcrypt-tel kéne)

    szul\_datum DATE NOT NULL,

    telefon VARCHAR(19) NOT NULL,

    email VARCHAR(50) NOT NULL,

    kor AS (YEAR(CURDATE()) - YEAR(szul\_datum)) PERSISTENT,

    jogosítvány\_ideje AS (YEAR(CURDATE()) - YEAR(jogositvany\_szama)) PERSISTENT

);

**Adatok Hash-elése és Mentése:**

A jelszavak hash-eléséhez például a **bcrypt** **használható**. Itt egy egyszerű **Python** példa a **hash**-elésre és az **adatbázisba** történő **mentésre**:

import bcrypt

import mysql.connector

# Kapcsolódás az adatbázishoz

conn = mysql.connector.connect(

host="localhost",

user="root",

password="password",

database="PowerGoDB"

)

cursor = conn.cursor()

# Felhasználói jelszó

password = "my\_secure\_password"

# Jelszó hash-elése

hashed = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())

# Második és utolsó számjegy kivonása

second\_digit = password[1]

last\_digit = password[-1]

second\_and\_last = second\_digit + last\_digit

# Adatok beszúrása a Szemely és Felhasznalok táblákba

cursor.execute("INSERT INTO Szemely

(V\_Nev, K\_Nev, szul\_datum, telefon, email, felh\_jelszo)

VALUES (%s, %s, %s, %s, %s, %s)",("Kiss", "János", "1980-01-01", "0036705005050", "janos.kiss@example.com", hashed))

szemely\_id = cursor.lastrowid

cursor.execute("INSERT INTO Felhasznalok (szemely\_id, felh\_nev, jelszo\_masodik\_utolso\_szamjegye, elofiz\_kat)

VALUES (%s, %s, %s, %s)", (szemely\_id, "janos.kiss", second\_and\_last, '1'))

conn.commit()

cursor.close()

conn.close()

**Összegzés**

* A jelszavak biztonságos **tárolása hash**-**eléssel** történik a **Szemely** **táblában**.
* A **Felhasznalok** **tábla** csak a **jelszó második és utolsó számjegyét tárolja**, hogy bizonyos műveletekhez elegendő információ álljon rendelkezésre.
* Az **operátor** így **megtekintheti** a **jelszó** **második** és **utolsó** **számjegyét** **anélkül**, hogy a teljes jelszót látná, biztosítva a felhasználói adatok bizalmasságát.