

# **Adatbázis rendszerek II.**

## **Jegyzőkönyv**

**Készítette: Pázmán András**  
**Neptun: H2Z4X3**

**2023.04.07**

## A feladat leírása

A feladat témája egy kliens oldali nyilvántartó alkalmazás, amely egy cég belső képzéseit tartja nyilván az alkalmazottak számára.

Az adatbázis tartalmazza a kurzus nyilvántartásához elengedhetetlen adatok mint:

- az alkalmazott tulajdonságai
- az oktató
- kezdés részletei
- képesítés

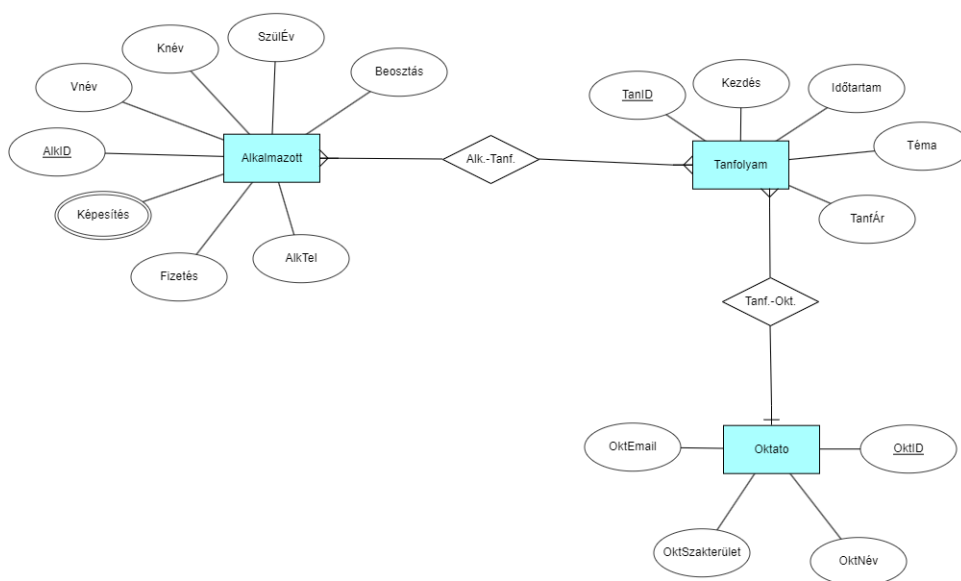
Ezek alapján a modellben megtalálhatóak az alábbi egyedek:

ALKALMAZOTT

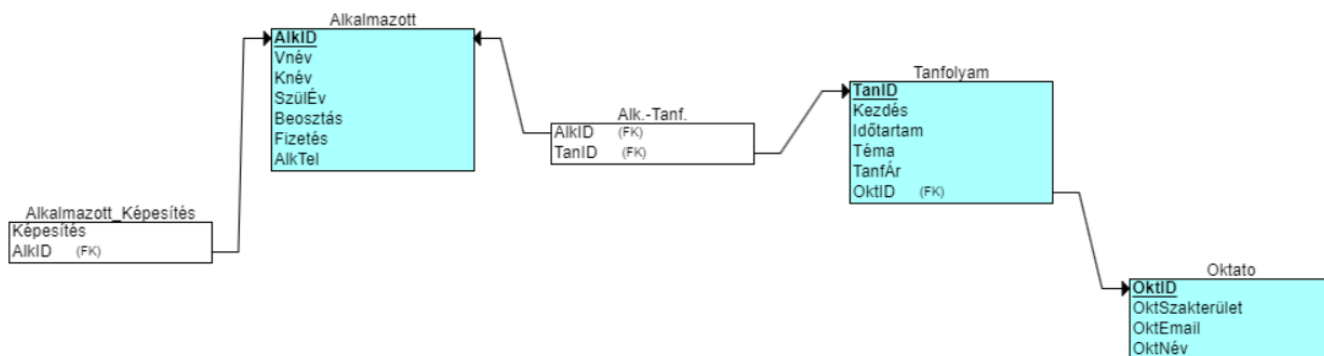
TANFOLYAM

OKTATO

### Az adatbázis ER modellje:



### Relációs diagram:



## A séma elemei

### ALKALMAZOTTAK:

- **AlkID:** Az alkalmazott egyedi azonosítója
- **Vnév:** Alkalmazott vezetékeve
- **Knév** Alkalmazott keresztnéve
- **SzülÉv:** Az alkalmazott születési éve (csak az évet tárolja)
- **Beosztás:** Az alkalmazott pozícióját jelöli ( pl.: könyvelő, műszakvezető stb.)
- **Fizetés:** Az alkalmazott fizetését tárolja
- **Képesítés:** összetett tulajdonság, azt jelzo hogy a dolgozó milyen plusz képesítést szerzett a céges tanfolyamokon
- **AlkTel:** Az alkalmazott telefonszámát tartalmazza, ( minden alkalmazottnak egy céges száma van)

### TANFOLYAM:

- **TanID:** A tanfolyam egyedi azonosítója
- **Kezdés:** A tanfolyam induló dátumát jelzi
- **Időtartam:** A tanfolyam időtartamát tárolja hetekben megadva ( pl.: 8 hét)
- **Téma:** A tanfolyamon szerzhető képesítés szakterülete
- **TanfÁr:** A tanfolyam árát tartalmazza ami 0 ha ingyenes  
más esetben az ár jelenik meg (pl.: 15 000ft )

### OKTATO:

- **OktID:** A tanfolyamon tanító oktató azonosítója
- **OktNév:** Az oktató vezetéke és keresztnévét tárolja
- **OktSzakterület:** Az oktató „képesítése” , az a terület amelyen tanít.
- **OktEmail:**

### ALKALMAZOTT KÉPESÍTÉS:

- **AlkID:** Az alkalmazott azonosítóját tartalmazza
- **Képesítés:** A megszerzett képesítést tartalmazza, fokozatot a „++” jelek jelzik

## **A táblákat feltöltő sql parancsok**

```
CREATE TABLE Alkalmazott (
```

```
    AlkID int PRIMARY KEY,
```

```
    Vnév varchar2(20) NOT NULL,
```

```
    Knév varchar2(20),
```

```
    SzülÉv int,
```

```
    Beosztas varchar2(20),
```

```
    Fizetes int,
```

```
    AlkTel VARCHAR2(15)
```

```
);
```

```
CREATE TABLE Oktato (
```

```
    OktID int PRIMARY KEY
```

```
    OktNev varchar2 (20),
```

```
    OktSzakterület varchar2 (20),
```

```
    OktEmail varchar2 (20)
```

```
);
```

```
CREATE TABLE Tanfolyam (
```

```
    TanID int PRIMARY KEY,
```

```
    Kezdes DATE NOT NULL,
```

```
    Idotartam int,
```

```
    Tema varchar2 (20),
```

```
    TanfAr int,
```

```
    FOREIGN KEY (OktID)
```

```
        REFERENCES Oktato(OktID)
```

```
);
```

```
CREATE TABLE AlkalmazottKepesites (
```

```
    AlkID int NOT NULL,
```

```
    Kepesites varchar(20),
```

```
    FOREIGN KEY (AlkID)
```

```
        REFERENCES Alkalmazott(AlkID)
```

```
);
```

```
CREATE TABLE AlkTanf (  
    AlkID int NOT NULL,  
    TanID int NOT NULL,  
    PRIMARY KEY (AlkID, TanID ,  
        FOREIGN KEY (AlkID) REFERENCES Alkalmazott(AlkID),  
        FOREIGN KEY (TanID) REFERENCES Tanfolyam  
);
```

```
INSERT INTO Alkalmazott VALUES(1, 'Kis', 'Janos', 1990, 'könyvelő', 210000, '+36 70 234 7766');
```

```
INSERT INTO Alkalmazott VALUES(2, 'Hor', 'Zsolt', 1980, 'számvitel', 220000, '+36 20 987 7766');
```

```
INSERT INTO Alkalmazott VALUES(3, 'Kukor', 'Ica', 1995, 'számvitel', 300000, '+36 30 123 7766');
```

```
INSERT INTO Alkalmazott VALUES(4, 'Külö', 'Nóra', 1988, 'pénzügy', 210000, '+36 10 567 7766');
```

```
INSERT INTO Alkalmazott VALUES(5, 'Véh', 'Cézár', 1978, 'hr', 310000, '+36 30 432 7766');
```

```
INSERT INTO Alkalmazott VALUES(6, 'Nagy', 'Elemér', 1982, 'hr', 320000, '+36 30 555 7766');
```

```
INSERT INTO Oktato VALUES(1, 'Orbán Sándor', 'finance', 'karolya@email.com');
```

```
INSERT INTO Oktato VALUES(2, 'Kocsis Emese', 'accountancy', 'karolya@email.com');
```

```
INSERT INTO Oktato VALUES(3, 'Takács György', 'hr', 'karolya@email.com');
```

```
INSERT INTO Oktato VALUES(4, 'Pataki Ernő', 'marketing', 'karolya@email.com');
```

```
INSERT INTO Tanfolyam VALUES(1, '2023-01-10', 8, 'konyveles++', 0, 2);
```

```
INSERT INTO Tanfolyam VALUES(2, '2023-02-14', 5, 'advanced finance', 20000, 1);
```

```
INSERT INTO Tanfolyam VALUES(3, '2023-03-01', 6, 'halado hr', 0, 3);
```

```
INSERT INTO Tanfolyam VALUES(4, '2023-04-01', 6, 'basic of marketing', 30000, 4);
```

```
INSERT INTO Tanfolyam VALUES(5, '2023-06-02', 8, 'konyveles++', 0, 2);
```

```
INSERT INTO AlkTanf VALUES(1,1);
```

```
INSERT INTO AlkTanf VALUES(2,2);
```

```
INSERT INTO AlkTanf VALUES(3,1);
```

```
INSERT INTO AlkTanf VALUES(4,4);
```

```
INSERT INTO AlkTanf VALUES(5,3);
```

```
INSERT INTO AlkalmazottKepesites VALUES(2, 'accounting++');
```

```
INSERT INTO AlkalmazottKepesites VALUES(3, 'finance++');
```

```
INSERT INTO AlkalmazottKepesites VALUES(3, 'marketing+');
```

```
INSERT INTO AlkalmazottKepesites VALUES(5, 'hr++');
```

## A program funkciói

Nulladik lépésként teszteltem hogy az adatbázis megfelelően működik-e. Ehhez néhány alap adattal töltöttem fel és consolos felületen és egy lekérdezéssel (ami több táblát kapcsol össze) végrehajtottam egy alap lekérdezést.

```
sqlite> .tables
AlkTanf      AlkalmazottKepesites  Tanfolyam
Alkalmazott  Oktato
sqlite> SELECT Alkalmazott.AlkID, Oktato.OktID, Oktato.OktNev FROM Alkalmazott JOIN AlkTanf ON Alkalmazott.AlkID = AlkTanf.TanfID JOIN Tanfolyam ON AlkTanf.TanfID = Tanfolyam.TanfID JOIN Oktato ON Tanfolyam.TanfID = Oktato.OktID;
1|1|Orbán Sándor
2|2|Kocsis Emese
3|3|Takács György
4|4|Pataki Ernő
sqlite>
```

### I. Bejelentkezési és menü modul

A felhasználó ahoz hogy dolgozni tudjon meg kell adni a felhasználónevét és jelszavát. Ezek egy másik adatbázisban tárolódnak függetlenül a felhasználói adatbázistól.

Jelen esetben: username: *user1*  
password: *password*

```
Username: user1
Password: password

Valasszon az alabbi menupontok kozul!:
=====
1. Rekodrok listazasa
2. Adatok felvitele
3. Adatok mentese
4. Szures
5. Egyeb muveletek
6. Meta
7. Exit
=====
|
```

Sikeres azonosítás után a felhasználó választhat a konzolon megjelenő menüpontok közül.

```
String url = "jdbc:sqlite:C:/appsqlite3/users.db";

String sql = "SELECT * FROM users WHERE username = ? AND password = ?";

try (Connection connect = DriverManager.getConnection(url);
    PreparedStatement pstmt = connect.prepareStatement(sql)) {

    pstmt.setString(1, username);
    pstmt.setString(2, password);

    ResultSet rs = pstmt.executeQuery();

    // igazgal tér vissza ha létezik
    return rs.next();
} catch (SQLException e) {
    System.out.println(e.getMessage());
}

return false;
```

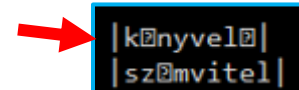
## 1. Rekordok listázása

A felhasználó az "1"-es menüpont kiválasztásával tud listázni, ehhez szükséges megadni a kiírandó tábla nevét:

```
=====
1
Tablak rekordjainak listazasa:
Adja meg a listazando tablat:
Alkalmazott
|1|      |Kis|      |Janos|    |1990|    |k nyvel |      |210000|    |+36 70 234 7766|
|2|      |Hor|      |Zsolt|    |1980|    |sz mvitel|     |220000|    |+36 20 987 7766|
|3|      |Kukor|    |Ica|      |1995|    |sz mvitel|     |300000|    |+36 30 123 7766|
```

Sajnos az eclips-ben hiába UTF-8 encoding van beállítva az ékezetes karakterekkel küszködik.

Többszörös próbálkozás után sem sikerült ennek a problémának a megoldása, ezért ahol lehet kerülöm az ékezetes szavak használatát.



k�nyvel�
sz�mvitel

## 2. Adatok felvitele:

Az adatok felvitelének felülete:

Jelenleg a programban az *Alkalmazott* és az *Oktato* nevű táblákra tudunk adatokat felvinni mivel ezek kerültek implementálásra. Az adatok felvitele paraméterezett kétfázisú parancsok használatával történik:

```
2
Az adatok felvitele
Adja meg melyik tablaba szeretne rekordot felvinni:
Alkalmazott
Vezeteknev: Kovacs
Keresztnev: Elemer
Szuetsi ev: 1985
Beosztas: marketing felelos
Fizetes: 320000
Telefon:: +36 70 223 6545
```

A listázás menüpontban meg is tekinthetjük hogy tényleg felvitte az adatoka:

```
|10|      |Vez|      |Elek|    |1994|    |alkalmazott|     |210000|    |+36 70 334 5567|
|11|      |Kovacs|    |Elemer|  |1985|    |marketing felelos| |320000|    |+36 70 223 6545|
```

Az ID-t nem a felhasználó adja meg, mivel ebből hibák eredhetnek ezért ez automatikusan kerül bevitelre . Ezt példaként kódreszlettel is szemléltetem:

```
//megekeress k az id -t az Id-t
String CommandText = "SELECT max(AlkID) FROM alkalmazott";
try {
    ResultSet rs = stmt.executeQuery(CommandText);
    if(rs.next()) {
        alkid = rs.getInt(1);
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

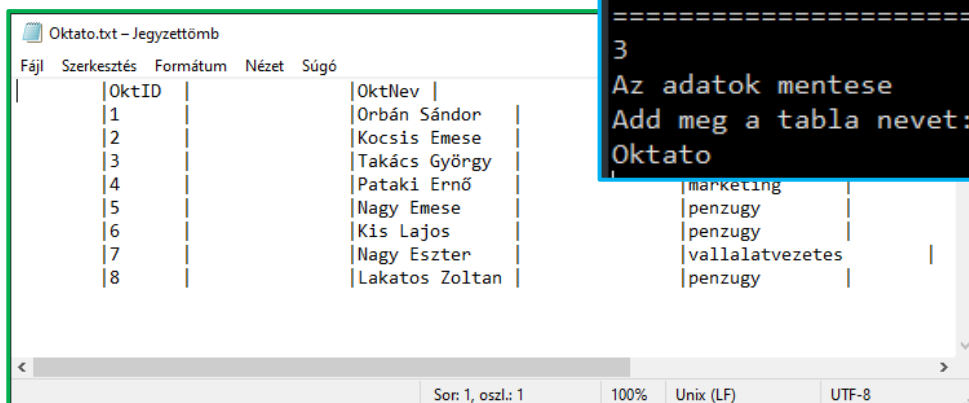


```
// Paraméterek beállítása
preparedStatement.setInt(1, alkid+1); //Automatikusan növeli az ID-t
preparedStatement.setString(2, vnev);
preparedStatement.setString(3, knev);
preparedStatement.setInt(4, Integer.parseInt(szulev));
preparedStatement.setString(5, beosztas);
preparedStatement.setInt(6, Integer.parseInt(fizetes));
preparedStatement.setString(7, alknel);

// Kétfázisú parancs végrehajtása
preparedStatement.executeUpdate();
```

### 3. Adatok mentése .txt állományba:

A felhasználónak meg kell adni a lementendő tábla nevét, és ha a mentés sikeres volt a program újra a menüválasztást dobja fel.



The screenshot shows a text editor window titled 'Oktato.txt - Jegyzettömb' with a table of teachers. The table has columns for 'OktID' and 'OktNev'. The data is as follows:

OktID	OktNev
1	Orbán Sándor
2	Kocsis Emese
3	Takács György
4	Pataki Ernő
5	Nagy Emese
6	Kis Lajos
7	Nagy Eszter
8	Lakatos Zoltan

Below the table, there is a terminal window showing a menu with 7 options. A red arrow points to option 3, 'Adatok mentése'. The terminal output shows the user has selected option 3 and is prompted to enter the table name, which is 'Oktato'.

```
1. Rekodrok listazasa
2. Adatok felvitele
3. Adatok mentese
4. Szures
5. Egyeb muveletek
6. Meta
7. Exit

=====
3
Az adatok mentese
Add meg a tabla nevét:
Oktato
```

Kódrészlet:

```
// Kiírjuk a tábla elemeit a fájlba
while (rs.next()) {
    for (int i = 1; i <= columnCount; i++) {
        String value = rs.getString(i);
        fwrite.write("\t" + "|" + value + "\t" + "|" + "\t");
    }
    fwrite.write("\n");
}
fwrite.close();

} catch (SQLException e) {
```

#### 4. Adatok szűrése

A program előre elkészített lehetőségeket ad, amelyet a felhasználó kiválaszthat amelyek egy al-menüben jelennek meg:

```
Valasszon az alábbi al-menupontok kozul!:
=====
1. Tanfolyam reszvetel
2. Legnagyobb fizetesek
3. Tanfolyamot oktato tanarok
4. Ingyenes tanfolyamkereso
5. Exit
=====
1
KisJanos, konyveles++
HorZsolt, advanced finance

2. Legnagyobb fizetesek
3. Tanfolyamot oktato tanarok
4. Ingyenes tanfolyamkereso
5. Exit
=====
2
Add meg a beosztas:
konyvelo
Kis Janos, 210000Ft

3. Tanfolyamot oktato tanarok
4. Ingyenes tanfolyamkereso
5. Exit
=====
3
Tema: konyveles++, Oktato nev: Orbán Sándor
Tema: advanced finance, Oktato nev: Kocsis Emese
Tema: halado hr, Oktato nev: Takács György
```

Kódrészlet:

```
try {
    Connection conn = connect();

    // Lekérdezés végrehajtása
    String query = "SELECT Alkalmazott.Vnev, Alkalmazott.Knev, Alkalmazott.Fizetes FROM Alkalmazott WHERE Alkalmazott.beosztas = '"+beosztas+"' ORDER BY fizetes DESC LIMIT 1;";
    Statement statement = conn.createStatement();
    ResultSet rs = statement.executeQuery(query);

    // Eredmény kiírása a konzolra
    while (rs.next()) {
        System.out.println(rs.getString("Vnev") + " " + rs.getString("Knev") + ", " + rs.getString("Fizetes") + "Ft");
    }

    // Kapcsolat lezárása
    rs.close();
    statement.close();
    conn.close();
    disconnect(conn);

} catch (Exception e) {
    System.err.println(e.getClass().getName() + ": " + e.getMessage());
    System.exit(0);
}
```

```
Connection conn = connect();
try {
    String query = "SELECT Tanfolyam.tema, Oktato.OktNev FROM Tanfolyam
INNER JOIN OKTATO ON Tanfolyam.TanID = Oktato.OktID ";
    Statement statement = conn.createStatement();
    ResultSet resultSet = statement.executeQuery(query);

    // Eredmény kiírása a konzolra kurzorral történő bejárás segítségével
    while (resultSet.next()) {
        String courseThema = resultSet.getString("tema");
        String teacherName = resultSet.getString("OktNev");
        System.out.println("Tema: " + courseThema + ", Oktato nev: " + teacherName );
    }
    // Kapcsolat Lezárása
    resultSet.close();
    statement.close();
    conn.close();

} catch (Exception e) {
    System.err.println(e.getClass().getName() + ": " + e.getMessage());
    System.exit(0);
}
disconnect(conn);
```

## 5. Egyéb műveletek

Itt a felhasználónak lehetősége van különböző műveletekre, mint a record törlés, update, tábla létrehozás ( egy előre megadott séma alapján)

```
Valasszon az alábbi al-menupontok kozul!:
```

```
=====
1. DELETE
2. UPDATE
3. CREATE TABLE
4. MOD
5. Exit
=====
```

```
=====
1
Record torlese
Adja meg a tablat melyben torolni szeretne:
Alkalmazott
Adja meg az alkalmazott ID melyet torni szeretne
1
```

```
try {
    conn.setAutoCommit(false);
} catch (SQLException e1) {

    e1.printStackTrace();
}

//delete ustasiats a tabla elemenek torlesere
try {
    deleteStmt = conn.prepareStatement("DELETE FROM " + tableName1 + " WHERE ALKID
= ?");
} catch (SQLException e1) {
    e1.printStackTrace();
}

try {

    //törles
    deleteStmt.setInt(1,alkid);

    //Tranzakció végleegsítése
    conn.commit();

    disconnect(conn);

    if (conn != null) {
```

## 6.Meta

Meta adatok megjelenítése:

```
=====
1. Rekodrok listazasa
2. Adatok felvitele
3. Adatok mentese
4. Szures
5. Egyeb muveletek
6. Meta
7. Exit
=====
6
TABLE NAME: Alkalmazott
Column name: AlkID, type: INT
Column name: Vnev, type: VARCHAR2(20)
Column name: Knev, type: VARCHAR2(20)
Column name: SzulEv, type: INT
Column name: Beosztas, type: VARCHAR2(20)
Column name: Fizetes, type: INT
Column name: AlkTel, type: VARCHAR2(15)
Columns:: 7
```

```
try {
    Connection conn = connect();
    // Metaadatok Lekérdezése
    DatabaseMetaData metaData = conn.getMetaData();
    // Táblák Lekérdezése
    ResultSet tables = metaData.getTables(null, null, "%", null);

    // Táblák adatainak kiírása a konzolra
    while (tables.next()) {
        String tableName = tables.getString("TABLE_NAME");
        System.out.println("TABLE NAME: " + tableName);

        ResultSet columns = metaData.getColumns(null, null, tableName, "%");
        int columnCount = 0;

        while (columns.next()) {
            String columnName = columns.getString("COLUMN_NAME");
            String columnType = columns.getString("TYPE_NAME");
            System.out.println("Column name: " + columnName + ", type: " + columnType);
            columnCount++;
        }

        System.out.println("Columns:: " + columnCount + "\n");
    }
    // Kapcsolat Lezárása
    conn.close();
    disconnect(conn);
} catch (SQLException e) {
```

## JDBC -ben megvalósítandó elemek:

Normál parancsok használata: - Tábla létrehozás előre beállított sablon szerint *.createTable()*

- Alkalmazott tanfolyam listázás *.selectEmployeeAndCourse()*

- Alkalmazott max fizetés *.selectMaxPayment(String beosztas)*

Paraméterezett kétfázisú parancsok használata:

- új rekord felvitelénél, PreparedStatement, *addRecord()*

- felhasználó azonosítása .. *authenticate(String username, String password)*

Kurzor kezelés:

- Adatok mentése *saveRecord()*

- Listázás *printTable()*

Tranzakció kezelés:

- tábla törlése *deleteTable()*

Metaadatok lekérdezése

- *getMeta()*