**JEGYZŐKÖNYV**

Adatbázis rendszerek II.

Féléves feladat

Készítette: **Mihály Gergő**

Neptunkód: **OY6ICJ**

Dátum: **2025. 05. 12.**

**A feladat leírása**

Egy Java nyelven írt alkalmazás, és egy PL/SQL nyelven írt program elkészítése egy relációs adatbázishoz. A relációs adatbázis egy egyetemi nyilvántartó rendszert modellez. Három darab táblából áll, az elsőben (Student tábla) a hallgatók vannak, neptun kóddal, névvel, a már teljesített kredit számukkal, a születési idejükkel és a email címükkel, a másodikban (Course tábla) a felvehető tantárgyak vannak névvel, a kredit számukkal, a vizsga időpontjával, és azzal, hogy kötelező tárgy-e. A harmadik tábla (Has tábla) egy kapcsoló tábla, azt írja le, hogy melyik hallgató melyik tárgyakat vette fel a félévre.

**A táblákat létrehozó SQL parancsok**

**Course tábla:**

CREATE TABLE Course(

id INTEGER(2) PRIMARY KEY,

name CHAR(30) NOT NULL,

credit INTEGER(2) NOT NULL,

exam\_date DATE NOT NULL,

required INTEGER(1) NOT NULL,

CHECK(required = 1 OR required = 0)

);

**Student tábla:**

CREATE TABLE Student(

neptun CHAR(6) PRIMARY KEY,

name CHAR(30) NOT NULL,

credit\_sum INTEGER(2) NOT NULL,

date\_of\_birth DATE NOT NULL,

email CHAR(30) NOT NULL

);

**Has tábla:**

CREATE TABLE Has(

neptun CHAR(6),

id INTEGER(2),

FOREIGN KEY(neptun) REFERENCES Student(neptun), FOREIGN KEY(id) REFERENCES Course(id)

);

**A táblákat feltöltő SQL parancsok**

**Course tábla:**

INSERT INTO Course VALUES(1, 'Analízis 1.', 5, '2025-06-08', 1);

INSERT INTO Course VALUES(2, 'Lineáris Algebra', 6, '2025-06-18', 1);

INSERT INTO Course VALUES(3, 'Villámvédelem', 2, '2025-06-20', 0);

INSERT INTO Course VALUES(4, 'Programozás', 7, '2025-06-24', 1);

INSERT INTO Course VALUES(5, 'Fizika', 3, '2025-06-27', 1);

**Student tábla:**

INSERT INTO Student VALUES('m2fqc5', 'Kis Pista', 30, '1985-01-20', 'kispista@gmail.com');

INSERT INTO Student VALUES('8jq3qn', 'Nagy Lajos', 83, '1985-04-25', 'nagylajos@gmail.com');

INSERT INTO Student VALUES('nhsbeg', 'Kovacs Imre', 15, '1989-11-03', 'kovacsimre@gmail.com');

INSERT INTO Student VALUES('xs0tfa', 'Orosz Vince', 38, '1991-01-21', 'oroszvince@gmail.com');

INSERT INTO Student VALUES('2bk1k6', 'Gáspár Ödön', 54, '1998-09-23', 'gasparodon@gmail.com');

**Has tábla:**

INSERT INTO Has VALUES('m2fqc5', 4);

INSERT INTO Has VALUES('m2fqc5', 5);

INSERT INTO Has VALUES('8jq3qn', 5);

INSERT INTO Has VALUES('8jq3qn', 4);

INSERT INTO Has VALUES('nhsbeg', 2);

INSERT INTO Has VALUES('nhsbeg', 1);

INSERT INTO Has VALUES('nhsbeg', 5);

INSERT INTO Has VALUES('xs0tfa', 1);

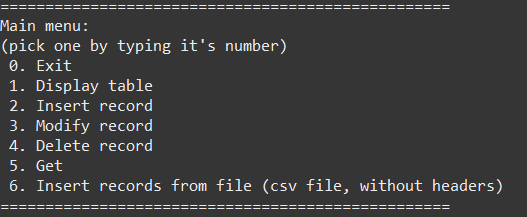
INSERT INTO Has VALUES('2bk1k6', 3);

INSERT INTO Has VALUES('2bk1k6', 2);

**A JDBC program**

A program egy konzolos applikáció, egy menürendszerben lehet navigálni a funkciók között.

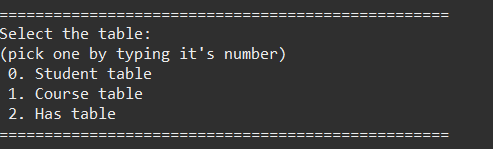
**A program főmenüje:**



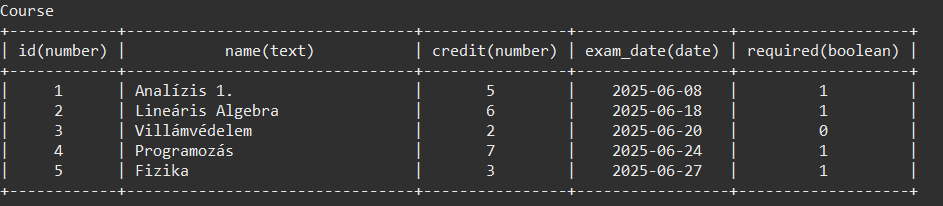
**“1. Display table“:**

A táblák teljes tartalmát lehet lekérdezni vele.

**A menüje:**



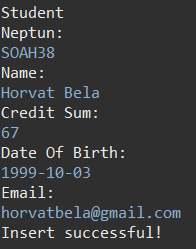
**A Course tábla lekérdezése:**

****

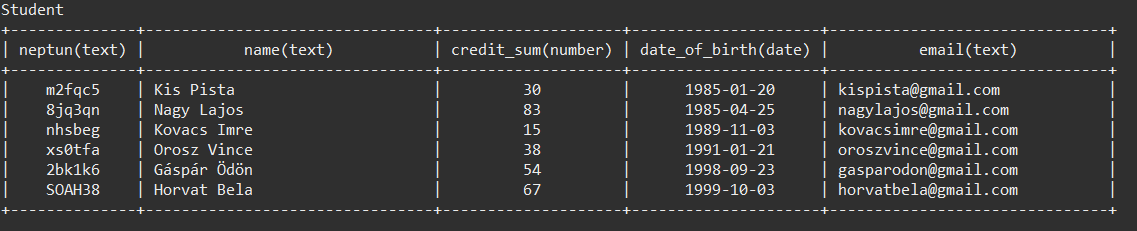
**“2. Insert record”:**

Egy rekordot lehet felvinni vele a táblákba.

**A Student táblára az adatok megadása:**

****

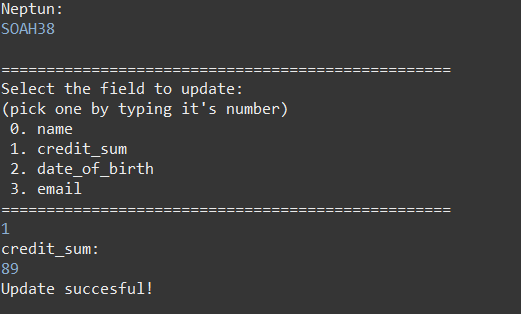
**A felvitel után a Student tábla:**

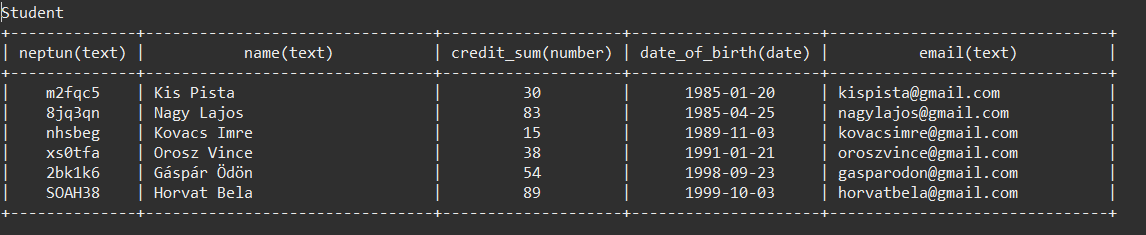
****

**“3. Modify record”:**

A táblák rekordjait lehet vele módosítani.

**A Student táblára:**

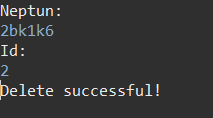
****

**A Student tábla a módosítás után:** 

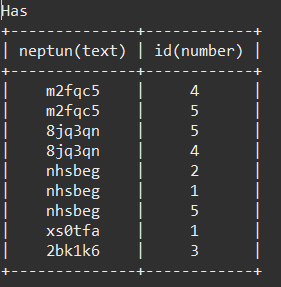
**“4. Delete record”:**

A táblák rekordjait lehet törölni vele.

**A Has táblára:**

****

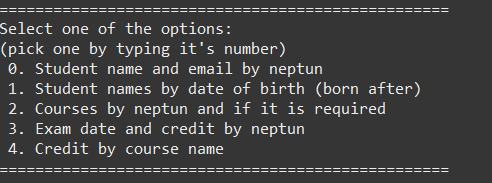
**A Has tábla a törlés után:**

****

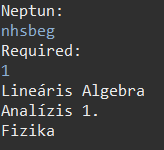
**“5. Get”:**

5 darab előre definiált lekérdezést lehet vele megvalósítani.

**A lekérdezések:**

****

**“2. Courses by neptun and if it is required”:**

****

**A JDBC program extra funkciói**

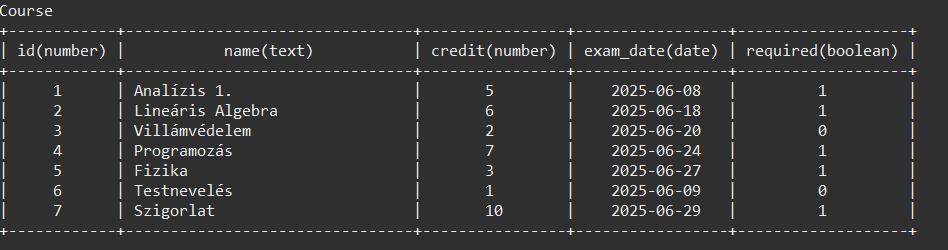
**“6. Insert records from file”:**

Több rekord felvitelét valósítja meg. A bemeneti fájl egy fejléc nélküli csv fájl, ennek az útvonalát meg kell adni a programnak. A program mellett található egy testInputs nevű mappa, amiben mindhárom táblára találhatóak bemeneti fájlok.

**A course.csv fájl tartalma:**

****

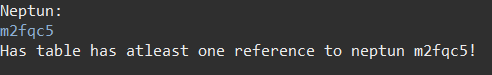
**A Course tábla a felvitel után:**

****

**Anomália kezelés:**

A törlési műveletnél csak akkor enged a program törölni egy rekordot, ha nincs rá hivatkozás a Has kapcsolótáblában.

**Programüzenet, a nem megengedett törlési műveletnél:**

****

**Logolás:**

A programnak indításakor meg lehet adni, hogy debug, vagy felhasználói módban fusson (ha paraméterként meg van adva, hogy “debug”, akkor debug módban indul, egyébként felhasználói módban).

**Felhasználói mód:**

Ebben a módban csak a hibákat logolja a program a képernyőre, pl.: ha a program nem találja meg a JDBC drivert, vagy egy SQL parancs nem tudott lefutni.

**Log, ha nem találja a program a JDBC drivert:**

****

**Debug mód:**

Ebben a módban minden SQL műveletről logot készít a program, akár sikeres, akár nem.

**Log, egy sikeres lekérdezésről:**

****

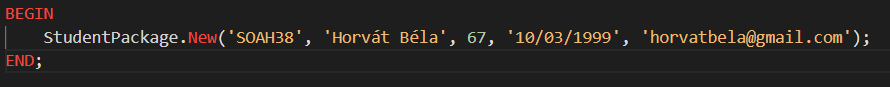
**A PL/SQL program**

A program 3 csomagból épül fel, minden táblához tartozik egy, ebben vannak megvalósítva a funkciók.

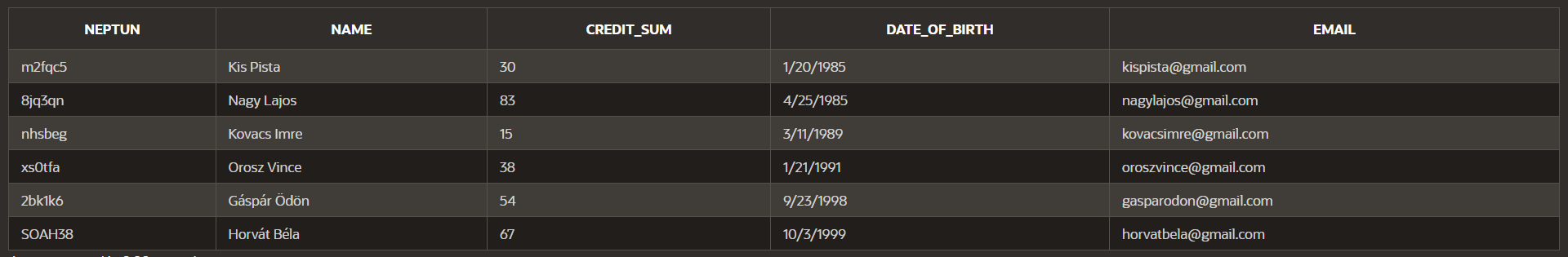
**Adatok felvitele**

Mindhárom csomagban implementált, a New nevű függvénnyel érhető el.

**Adatfelvitel hívása a Student táblára:**

****

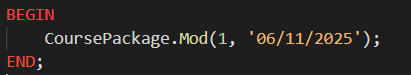
**A Student tábla adat felvitel után:**

****

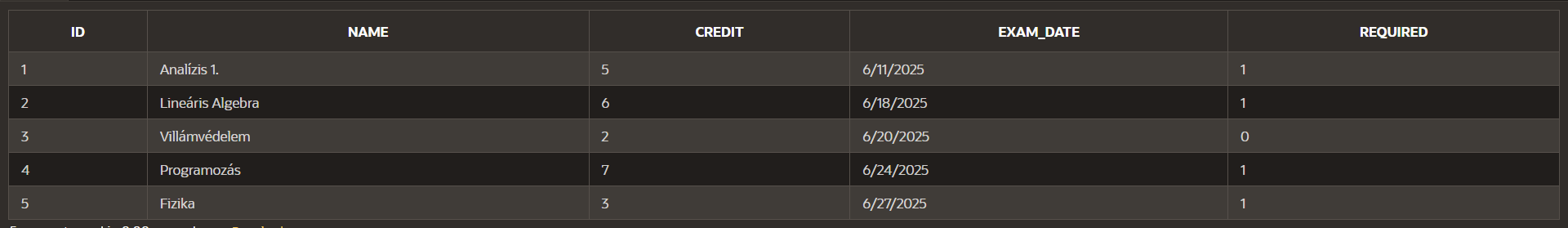
**Adatok módosítása**

Csak a fő táblákra implementált, a Mod függvénnyel érhető el.

**Adatmódosítás hívása a Course táblára:**

****

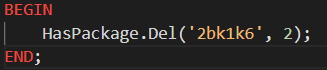
**A Course tábla a módosítás után:**

****

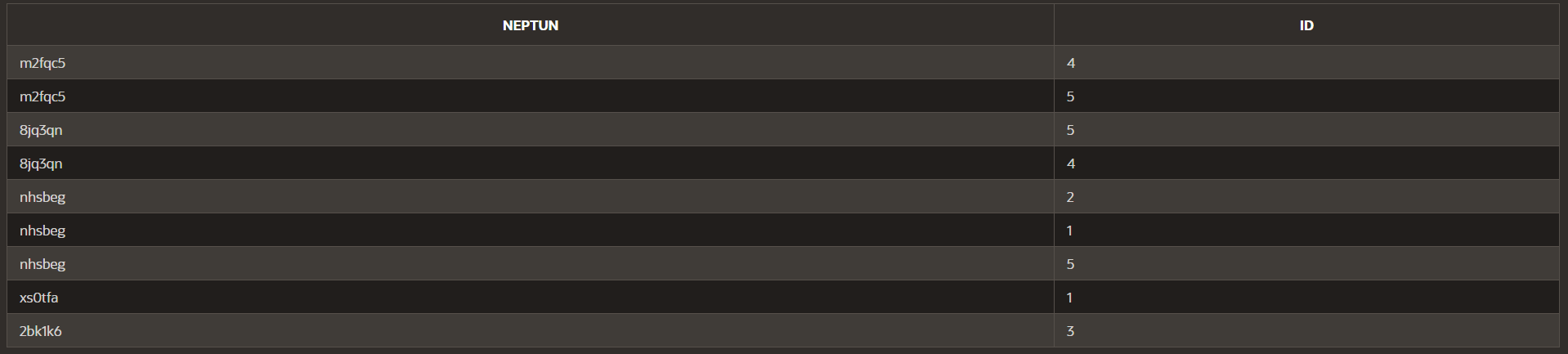
**Adatok törlése**

Mind három táblára implementálva, a Del függvénnyel érhető el.

**Adattörlés hívása a Has táblára:**

****

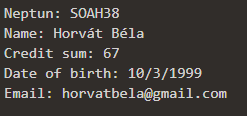
**A Has tábla a törlés után:**

****

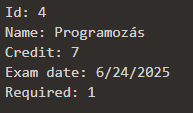
**Rekord lekérdezése**

Csak a fő táblákra implementált, a GetStudent és a GetCourse függvényekkel érhető el.

**Rekord lekérdezés eredménye a Student táblára:**

****

**Rekord lekérdezés eredménye a Course táblára:**

****

**Aggregált lekérdezés**

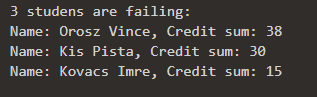
Csak a fő táblákra implementált

**Student tábla**

A GetFailingStudents függvénnyel érhető el, kiírja mennyi

hallgató áll bukásra (credit\_sum < 50) és kik azok a hallgatók.

**A lekérdezés eredménye:**

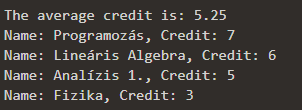


**Course tábla:**

A GetAvgCreditOfRequiredCourses függvénnyel érhető el,

kiírja a kötelező tárgyak kredit átlagát, és a kötelező tárgyakat.

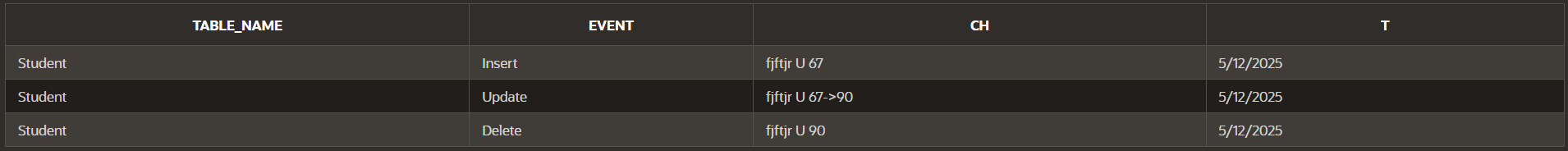
**A lekérdezés eredménye:**

****

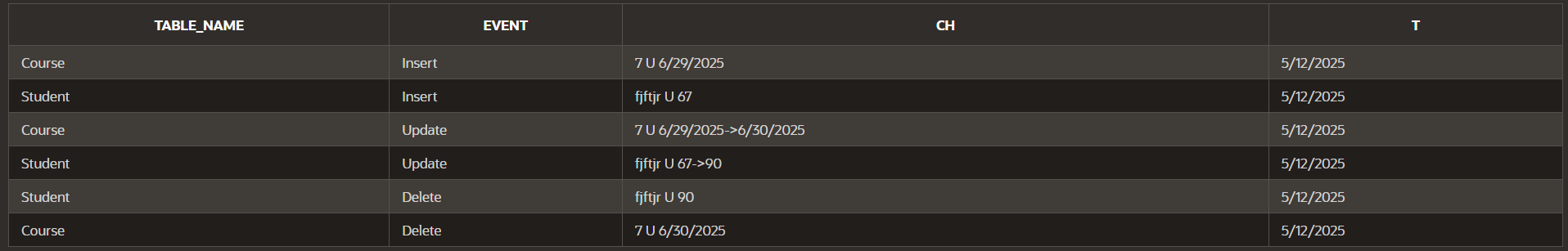
**Trigger módosítás naplózására:**

A két fő táblára van implementálva, INSERT, UPDATE és DELETE műveleteket naplóznak egy Logs nevű táblába. A Student táblában a credit\_sum mező van naplózva, a Course táblában pedig az exam\_date.

**A Student tábla 3 módosítása után a Logs tábla:**

****

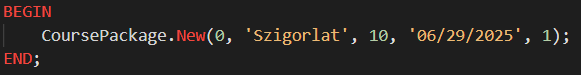
**A Course tábla 3 módosítása után a Logs tábla:**

****

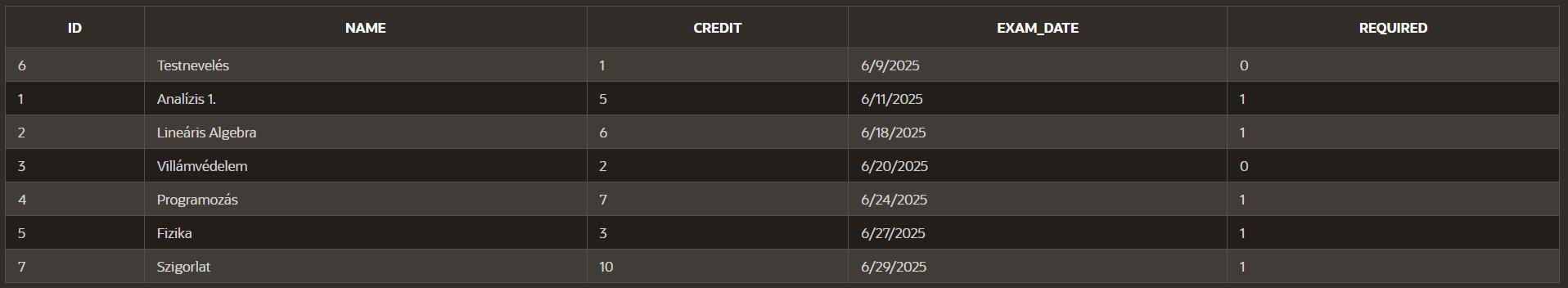
**Trigger automatikus kulcsérték megadásra**

A Course táblára van implementálva.

**Adatfelvitel hívása a Course táblára:**

****

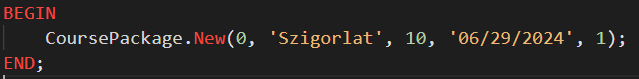
**A Course tábla az adatfelvitel után:**

****

**Trigger a módosítás kontrollálására**

A Course táblában van implementálva az exam\_date mezőre, nem engedi az aznapitól korábbi időpont felvitelét.

**Adatfelvitel hívása a Course tábára:**

****

**A hibaüzenet:**

****