

Adatbázis rendszerek 2

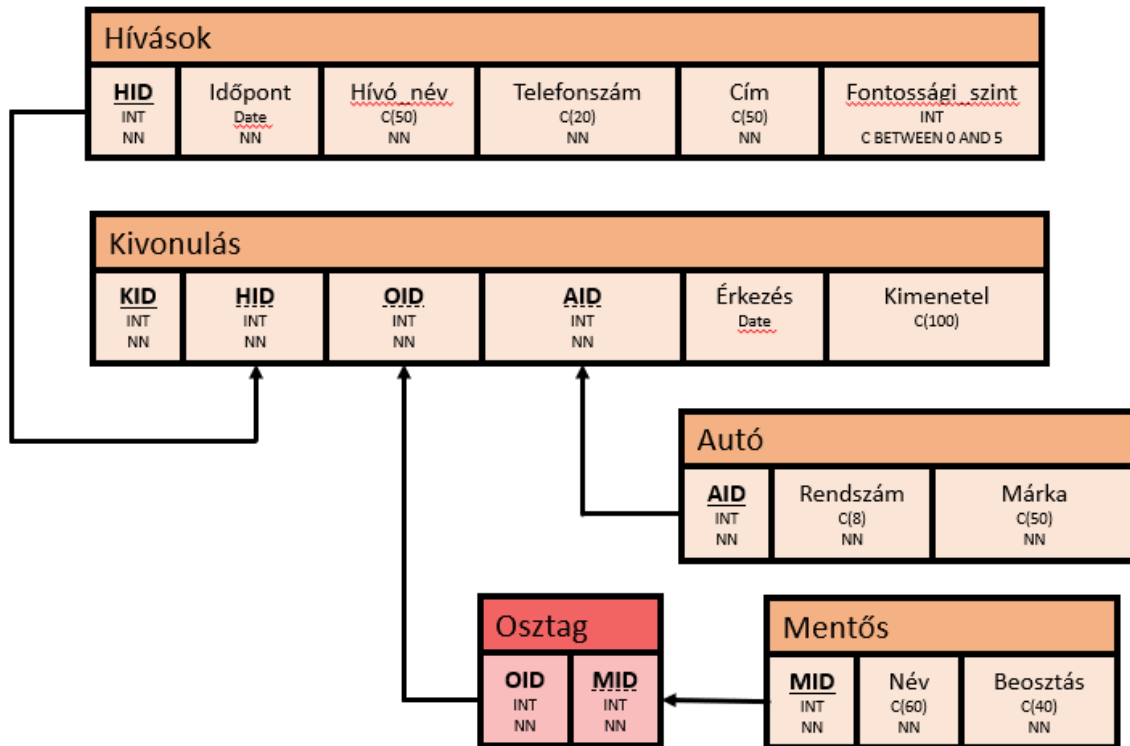
Második féléves egyéni feladat megoldása

Sikora Dávid Ádám
IRE699

Gyakorlatvezető: Dr. Kovács László
Gyakorlat időpontja: Hétfő 10:00
Képzés: BSc nappali

A feladat leírása:

A feladat egy mentőszolgálat adatbázisának serveroldali programjának megtervezése. A feladathoz használt adatbázis sémája a következő:



Az adatbázis 3 fő táblát tartalmaz és 2 kapcsolótáblát. Az első a **Hívások** tábla, melyben a beérkező hívások nyilvántartását tartalmazza. Itt a legutolsó adattag a fontossági szint, melyet az adatbázis ellenőriz, hogy mindenképp 0 és 5 közötti szám legyen, itt 0 a hamis riasztás, 5 az életveszélyes szituáció. A következő két tábla az **Autó** és **Mentős** tábla, melyek a mentőautók és a mentős dolgozók nyilvántartásáért felelnek. Az **Osztag** tábla összeköti az egyes mentős dolgozókat és egy ID alá rendeli őket. Végül pedig a **Kivonulás tábla** köti össze a hívást a kivonuló mentőautóval és az osztaggal, valamint tartalmazza továbbá a kiérkezés pontos időpontját és a végkifejletet.

Az adatbázis sémájának létrehozásához szükséges SQL kódok:

```
CREATE TABLE HIVAS (
```

```
    HID INT PRIMARY KEY,  
    Idopont TIMESTAMP NOT NULL,  
    Hivo_nev VARCHAR2(50) NOT NULL,  
    Telefonszam VARCHAR2(20) NOT NULL,  
    Cim VARCHAR2(50) NOT NULL,  
    Fontossagi_szint INT NOT NULL CHECK(Fontossagi_szint BETWEEN 0 AND 5) )
```

```
CREATE TABLE AUTO(
```

```
    AID INT PRIMARY KEY,  
    Rendszam VARCHAR2(8) NOT NULL,  
    Marka VARCHAR2(50) NOT NULL)
```

```
CREATE TABLE MENTOS(
```

```
    MID INT PRIMARY KEY,  
    Nev VARCHAR2(60) NOT NULL,  
    Beosztas VARCHAR2(40) NOT NULL)
```

```
CREATE TABLE OSZTAG(
```

```
    OID INT PRIMARY KEY,  
    MID INT REFERENCES MENTOS(MID))
```

```
CREATE TABLE KIVONULAS (  
    KID INT PRIMARY KEY,  
    HID REFERENCES HIVAS(HID),  
    OID REFERENCES OSZTAG(OID),  
    AID REFERENCES AUTO(AID),  
    Erkezes TIMESTAMP NOT NULL,  
    Kimenetel VARCHAR2(50) NOT NULL)
```

PL/SQL csomag létrehozása:

```
CREATE OR REPLACE PACKAGE MENTOSZOLGALAT AS  
    PROCEDURE uj_hivas (nev VARCHAR2, telefonszam VARCHAR2, cim VARCHAR2, fontossagi_szint INT );  
    PROCEDURE uj_auto (rendszam VARCHAR2, marka VARCHAR2);  
    PROCEDURE uj_mentos (nev VARCHAR2, beosztas VARCHAR2);  
    PROCEDURE uj_osztag (KID INT, MID INT );  
    PROCEDURE KIVONULAS_HOZZARENDELES(hid INT, aid INT, PID OUT INT);  
    PROCEDURE KIVONULAS_KIMENETEL(id INT, Perkezes TIMESTAMP, Pkimenetel VARCHAR2);  
    FUNCTION NAPI_HIVASOK_SZAMA(NAP DATE) RETURN INTEGER;  
    PROCEDURE HIVASOK_SZURES_NEV(PNEV VARCHAR2, KURZOR OUT SYS_REFCURSOR);  
    PROCEDURE HIVASOK_SZURES_IDOSZAK(TOL TIMESTAMP, IG TIMESTAMP, KURZOR OUT SYS_REFCURSOR);  
    PROCEDURE TORLES_MIND;  
    PROCEDURE RAND_FELTOLT_AUTO;  
    PROCEDURE FILE_FELTOLT_HIVAS;  
END;
```

A csomag generáló kódja:

```
CREATE OR REPLACE PROCEDURE uj_hivas (nev VARCHAR2, telefonszam VARCHAR2, cim VARCHAR2,  
fontossagi_szint INT ) AS  
    PID INT := 0;  
  
BEGIN  
    SELECT COUNT(*) INTO PID FROM HIVAS;  
  
    IF PID > 0 THEN  
        SELECT MAX(hid)+1 INTO PID FROM hivas;  
    ELSE  
        PID:=1;  
    END IF;  
    INSERT INTO HIVAS VALUES(PID,SYSTIMESTAMP,nev,telefonszam,cim,fontossagi_szint);  
    COMMIT;  
EXCEPTION
```

```
    WHEN OTHERS THEN ROLLBACK;  
END;
```

```
CREATE OR REPLACE PROCEDURE uj_auto (rendszam VARCHAR2, marka VARCHAR2) AS  
    PID INT := 0;
```

```
BEGIN  
    SELECT COUNT(*) INTO PID FROM auto;  
  
    IF PID > 0 THEN  
        SELECT MAX(aid)+1 INTO PID FROM auto;  
    ELSE  
        PID:=1;  
    END IF;  
    INSERT INTO auto VALUES(PID,rendszam,marka);  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN ROLLBACK;  
END;
```

```
CREATE OR REPLACE PROCEDURE uj_mentos (nev VARCHAR2, beosztas VARCHAR2) AS  
    PID INT := 0;
```

```
BEGIN  
    SELECT COUNT(*) INTO PID FROM mentos;  
  
    IF PID > 0 THEN  
        SELECT MAX(mid)+1 INTO PID FROM mentos;  
    ELSE  
        PID:=1;  
    END IF;  
    INSERT INTO mentos VALUES(PID,nev,beosztas);  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN ROLLBACK;  
END;
```

```
CREATE OR REPLACE PROCEDURE uj_osztag (KID INT, MID INT ) AS
```

```
BEGIN  
    INSERT INTO osztag VALUES(KID, MID);  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN ROLLBACK;  
END;
```

```
CREATE OR REPLACE PROCEDURE KIVONULAS_HOZZARENDELES(hid INT, aid INT, PID OUT INT) AS
```

```
BEGIN  
    PID :=0;  
    SELECT COUNT(*) INTO PID FROM kivonulas;
```

```
IF PID > 0 THEN
    SELECT MAX(kid)+1 INTO PID FROM kivonulas;
ELSE
    PID:=1;
END IF;
INSERT INTO kivonulas VALUES(PID,hid,aid,null,null);
COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        begin
            ROLLBACK;
            PID := -1;
        end;
END;
```

```
CREATE OR REPLACE PROCEDURE KIVONULAS_KIMENETEL(id INT, Perkezes TIMESTAMP, Pkimenetel VARCHAR2) AS
BEGIN
    UPDATE KIVONULAS SET erkezes = Perkezes, kimenetel = Pkimenetel WHERE kid = id;
    COMMIT;
END;
```

```
CREATE OR REPLACE FUNCTION NAPI_HIVASOK_SZAMA(NAP DATE) RETURN INTEGER AS
RET INTEGER;
BEGIN
    SELECT COUNT(*) INTO RET FROM HIVAS WHERE idopont<NAP+INTERVAL '1' DAY AND idopont>NAP-
INTERVAL'1'DAY;
    RETURN RET;
END;
```

```
CREATE OR REPLACE PROCEDURE HIVASOK_SZURES_NEV(PNEV VARCHAR2, KURZOR OUT SYS_REFCURSOR) AS
BEGIN
    OPEN KURZOR FOR SELECT * FROM HIVAS WHERE hivo_nev = PNEV;
END;
```

```
CREATE OR REPLACE PROCEDURE HIVASOK_SZURES_IDOSZAK(TOL TIMESTAMP, IG TIMESTAMP, KURZOR OUT
SYS_REFCURSOR) AS
BEGIN
    OPEN KURZOR FOR SELECT * FROM HIVAS WHERE idopont >= TOL AND idopont <= IG;
END;
```

```
CREATE OR REPLACE PROCEDURE TORLES_MIND AS
BEGIN
    DELETE FROM OSZTAG;
    DELETE FROM KIVONULAS;
    DELETE FROM MENTOS;
    DELETE FROM AUTO;
    DELETE FROM HIVAS;
    COMMIT;
```

```
END;
```

```
CREATE OR REPLACE PROCEDURE RAND_FELTOLT_AUTO IS
```

```
  ID INT := 1;
  TYPE ST IS VARRAY(5) OF VARCHAR2(15);
  RENDSZ ST;
  MARKA ST;
  RAND INT;
  BEGIN
    DELETE FROM AUTO;
    MARKA := ST('FORD','VW','KIA','MERCEDES','HONDA');
    RENDSZ := ST('AAA-123','AAAA-123','ABCD-345','XYZ-987','ASDX-999' );

    FOR VA IN 1..5 LOOP
      SELECT MOD(ABS(DBMS_RANDOM.RANDOM()),5) +1 INTO RAND FROM DUAL;
      INSERT INTO AUTO VALUES(VA, RENDSZ(VA), MARKA(RAND));
    END LOOP;
  END;
```

```
CREATE OR REPLACE PROCEDURE FILE_FELTOLT_HIVAS AS
```

```
  FA UTL_FILE.FILE_TYPE;
  SOR VARCHAR(200);
  IND INT :=0;
  HIV HIVAS%ROWTYPE;
  BEGIN
    DELETE FROM HIVAS;
    FA := UTL_FILE.FOPEN('DIR','HIVASOK_SD.TXT', 'R');
    LOOP
      IND :=1;
      UTL_FILE.GET_LINE(FA, SOR);

      BEGIN
        FOR I IN (SELECT REGEXP_SUBSTR (SOR, '[^;]+' , 1, level) AS ADAT FROM DUAL CONNECT BY
REGEXP_SUBSTR(SOR, '[^;]+' , 1, level) IS NOT NULL) LOOP
          IF IND = 0 THEN
            HIV.hivo_nev := I.ADAT;
          ELSIF IND = 1 THEN
            HIV.telefonszam := I.ADAT;
          ELSIF IND = 2 THEN
            HIV.cim := I.ADAT;
          ELSIF IND = 3 THEN
            HIV.fontossagi_szint := TO_NUMBER(I.ADAT);
          ELSE
            RAISE_APPLICATION_ERROR(-20000, 'NEM MEGFELELŐ ADATFILE!');
          END IF;
          IND := IND+1;

          uj_hivas(HIV.hivo_nev, HIV.telefonszam, HIV.cim, HIV.fontossagi_szint);
        END LOOP;
      END;
```

```
END LOOP;

EXCEPTION
  WHEN OTHERS THEN ROLLBACK;
END;
END LOOP;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    UTL_FILE.FCLOSE(FA);
END;
```

JDBC program:

Az egyes funkciók leteszteléséhez elkészült terminál megjelenésű java programban csatlakozás és bejelentkezés után lehetőség nyílik minden funkció leteszteléséhez. Az adott funkciókat a főmenü pontjaival és az egyes pontokban szereplő almenükkel választhatjuk ki.

A JDBC alkalmazás forrásfájljai a következő github inken találhatóak:

[Redd-15/AdatB2: Adatbázis Rendszerek 2 tantárgy beadandó félévközi feladatai \(github.com\)](https://github.com/Redd-15/AdatB2)