

Az alábbi feladatban egy részadatbázisát látja. Az adatmodell táblái a következők:

SZAMLAK		
P	* ID	NUMBER (*,0)
	* NEV	VARCHAR2 (250 BYTE)
	* BANKKARTYASZAM	CHAR (16 BYTE)
	* EGYENLEG	NUMBER (*,0)
PK_SZAMLAK (ID)		
PK_SZAMLAK (ID)		

szamlak tábla: A banknál nyilvántartott számlákat tartalmazza. Minden rekord egy elsődleges kulcs értékkel (*id* mező), egy tulajdonosnévvel (*nev* mező), egy a számlához kapcsolódó bankkártya számával (*bankkartyaszam* mező), és egy a bankszámla egyenlegét nyilvántartó (*egyenleg* mező) mezővel írható le.

Oldja meg az alábbi feladatokat:

1. Az adatbázis létrehozásához szükséges SQL forrást a C# projekten belül az *SQL\create.sql* fájlban találja. Egészítse ki a szkriptet az alábbiak szerint, majd annak futtatásával hozza létre az adattáblákat.
 - A *szamlak* tábla *bankkartyaszam* mezője csak egyedi értékeket vehet fel.
 - A *szamlak* tábla *egyenleg* mezőjébe csak pozitív értékek kerülhetnek.

A kiegészített szkriptet a futtatás után mentse a C# projekten belül az *SQL\create.sql* fájlba (írja felül annak tartalmát)!

2. Készítsen tárolt függvényt *get_next* néven. A függvény generálja a *szamlak* tábla következő lehetséges elsődleges kulcs értékét.
 - A legelső használható azonosító érték a 10000.
 - Nem üres tábla esetén, a táblában előforduló legkisebb *id* értéket csökkentjük 3-mal a következő *id* előállításához.
 - A függvény helyes működéséhez a hívó adatbázis felhasználónak ne kelljen jogosultsággal rendelkeznie az *szamlak* táblán.

A szkriptet mentse a C# projekten belül az *SQL\get_next.sql* fájlba!

3. Készítsen tárolt függvényt *check_data* néven. A függvény feladata a bankkártya számának ellenőrzése.
 - A kártyaszám mindig 16 karakter, amely mindegyike számjegy.
 - Az ellenőrzés során balról jobbra haladva a páratlan pozícióban álló számjegyeket 2-vel megszorozzuk. Ha a szorzat értéke 9-nél nagyobb, a szorzatból kivonunk 9-et. Az így kapott számok összegéhez hozzáadjuk a páros sorszámú számokat. Ha ez az összeg 0-ra végződik, az azonosító helyes.
 - A függvény nem csak PL/SQL-ből, hanem egyszerű SQL utasításból is hívható.

A szkriptet mentse a C# projekten belül az *SQL\check_data.sql* fájlba!

4. Készítsen tárolt eljárást *insert_data* néven. Az eljárás feladata az új rekord beszúrása a *szamlak* táblába.
 - Az eljárás paraméterben kapja a szükséges adatokat.
 - Kimenő paramétert biztosít a hozzáadott sorok számának visszaadására.
 - Az eljárás ellenőrzi a bemenő adatokat és ha mindent rendben talál, akkor beszúrja a rekordot.
 - Implicit kurzorattribútum segítségével ellenőrzi, hogy sikerült-e a beszúrás.

A szkriptet mentse a C# projekten belül az *SQL\insert_data.sql* fájlba!

5. Egészítse ki a letöltött C# projektet az alábbiak szerint.
 - Models\Records\Record.cs: Készítse el a *szamlak* tábla rekordjának reprezentáns osztályát.
 - Models\Managers\OracleDatabase.cs:
 - getConnection(): visszaad egy paraméterezett Oracle adatbáziskapcsolatot a hívás helyére.
 - Select(): lekérdezi a *szamlak* tábla rekordjait.
 - Insert(Record record): beszúrja a rekordot a tárolt eljárás hívásán keresztül
 - Delete(Record record): törli a paraméterben kapott rekordot az adatbázisból.
 - CheckData(string data): meghívja az ellenőrző függvényt és kezeli annak visszatérési értékét.
 - Views\WinForms\ZHForm.cs
 - Valósítsa meg a DataGridView kezelését (inicializálás + adatokkal való feltöltés). A táblázatban a rekord azonosítójának és kártyaszámának kell megjelenennie.
 - Valósítsa meg a Delete gomb Click eseményét.
 - Valósítsa meg az Insert gomb Click eseményét.
 - Amennyiben a textBox_kartyaszam mezőjébe elkezdjük begépelni a számlaszámot, úgy azt minden karakter leütés esetén ellenőrizzük a szerveren implementált tárolt függvénnyel.