

Az SQL nyelv tanítása

Czenky Márta

marta.czenky@t-online.hu
SZIE GÉK

Absztrakt Az SQL nyelv tapasztalataim szerint jobban tanítható, mint a hagyományos procedurális programnyelvek. A tanítás során ennek ellenére számos olyan kérdéssel találkozunk, melyek a hallgatók számára problémát jelentenek. Ezeket a problémákat, az általam nyújtott és javasolt megoldási módokat, a tanításban alkalmazott feladatokat, a számonkérés és hallgatói felmérés eredményét kívánom ismertetni.

1. Az SQL nyelvről

Az SQL szabványos interaktív és programozási nyelv, mely lehetővé teszi relációs adatbázisokban lekérdezések, adatkezelő és adatfelügyeleti műveletek végrehajtását. Az ANSI és az ISO is szabványosította, mely szabványt az adatbázis-kezelő rendszerek gyártói elfogadják és alkalmazzák.

Az SQL nyelv magasan strukturált, egyszerű szintaktikájú, korlátozott számú utasítással rendelkezik, mégis lehetővé teszi összetett lekérdezések, adatbázis műveletek végrehajtását.

Egy SQL utasítás jóval rövidebb, mint a procedurális programnyelveken írt programok. Ez egyúttal azt is jelenti, hogy az utasítás áttekinthetőbb, és könnyebben elkészíthető, mint a hagyományos programok.

A nyelv deklaratív, ami előnyt és hátrányt is jelent. Hátrány lehet, hogy a hallgatók számára újszerű és szokatlan, mert általában megelőző tanulmányaik során valamilyen procedurális programnyelvet tanulnak. Előny, hogy nem kell algoritmust írni, csak az eredményt kell megfogalmazni a nyelv segítségével, melynek során nem lépésekben, hanem halmazokban kell gondolkodni.

2. Oktatási célok és körülmények

Az SQL nyelv tanulása során kitűzött célok, a hallgatókkal szembeni elvárások a következők:

- Megértsék az SQL nyelv szerkezetét, ne csak a nyelv szintaktikáját, de szemantikáját is tanulják meg.
- Interaktív módon használják az SQL nyelvet (SQL-92). Képesek legyenek egy sémával adott adatbázist kezelni.

- Képesek legyenek tárolt eljárások, triggerek írására, a kurzorral történő feldolgozásra (SQL-99).
- Megismerkedjenek a különböző adatbázis-kezelő szoftverek SQL nyelvjárásaival

A nyelv tanulására egy egy féléves, heti 2 órás kurzus fele áll rendelkezésre. Ez feszített tanítási tempót követel meg, az órai gyakorlásra nem nagyon van lehetőség.

A tanítás az MS SQL Server Express segítségével történik. Javasolom a hallgatóknak, hogy ennek az adatbázis-kezelő rendszernek és másoknak is (MySQL, Access, Oracle) egyénileg tanulmányozzák a szabványtól eltérő megoldásait.

3. Sorrend az oktatásban

Az SQL nyelv tanításakor két lehetséges út közül választhatunk. Az egyik, hogy a logikai adatbázisterv ismeretében a DDL utasításaival létrehozzuk a fizikai adatbázist, az adatbázist kezelő felhasználókat, illetve felhasználói csoportokat, beállítjuk az adatbázis objektumaira vonatkozó jogosultságokat, feltöltjük a megfelelő DML utasításokkal az adatbázist, és csak ezt követően foglalkozunk a lekérdezésekkel.

A másik út, hogy először a SELECT utasítást tanítjuk meg a hallgatóknak, és csak ezt követően foglalkozunk a DDL és DML utasításaival. Az első az adatbázisok építésének szokásos menete, ez lenne a logikus sorrend. Tanári tapasztalataim szerint viszont a második célszerűbb a következő okok miatt:

- A lekérdezések során megszokják a hallgatók az SQL deklaratív voltát.
- Ellentétben a DDL utasításaival a SELECT eredményt visszaadó utasítás, mely segíthet az utasítás helyességének ellenőrzésében.
- A SELECT révén megtanulják a záradékokból felépülő utasítások használatának módját.
- A DDL utasítások némelyike nagyon összetett, például a CREATE TABLE, mely különböző CONSTRAINT záradékaival, Null érték, default érték és hivatkozási integritás kezelésével, az alkalmazható adattípusok megismerésével kezdetben inkább elrettentik a hallgatót.

4. Algoritmusok

A SELECT utasítás megtanulása kezdetben nagyon egyszerű, a csak a SELECT, az egytáblás FROM és a WHERE záradékokat tartalmazó utasítások nagyon hasonlóságot mutatnak a beszélt nyelven megfogalmazott kérdésekhez. Bonyolultabb utasítások megszerkesztése viszont már problémát jelenthet a hallgatók számára. Ezért az utasítások írásakor a következő algoritmust alkalmazását javaslom:

1. Számba kell venni, hogy a megjelenítendő adatokat milyen táblázatok tartalmazzák.
2. Meg kell határozni, hogy hogyan kell ezeket a táblázatokat összekapcsolni.
3. Milyen feltételnek eleget tevő sorokból kell az adatokat megjeleníteni.
4. Szükséges-e az adatok csoportosítása vagy sem.
5. Milyen feltételnek eleget tevő csoportokat kell kiválasztani.
6. Milyen adatokat kell megjeleníteni.
7. Kell-e az eredményt rendezni.

A fenti sorrend nagyban hasonlít a SELECT utasítás kiértékelési sorrendjéhez. Ami ebből az algoritmusból kimaradt, az az alkérdések használata szükségességének vizsgálata. Általában a WHERE és HAVING záradékokban használjuk őket, tehát a feltételek írásakor merül fel használatuknak igénye. Amíg a hallgatók nem kellően gyakorlottak célszerű először az alkérdést megszerkeszteni, helyességét ellenőrizni, és csak ezt követően az őt tartalmazó utasítást megírni.

A nyelv egyéb utasításai nem ennyire összetettek, ezért a hallgatók könnyebben megértik és megtanulják azokat. Természetesen itt is javasolom a záradékok sorrendje szerinti fokozatos építkezést. Kivételt képez talán a CREATE TABLE utasítás, melyben a CONSTRAINT záradékok elkészítése problémát jelenthet. Ez esetben célszerű először az oszlopdefiníciókat megírni, majd az egyoszlopos, végül a többszlopos CONSTRAINT záradékot.

5. Problémák és a kiküszöbölésükre javasolt módszerek

Az SQL nyelv tanulása során a hallgatók számára problémát jelentenek:

- A nyelv deklaratív volta. Mint jeleztük, programozást általában már tanulmányaik elején tanulnak, de ennek során a procedurális programozási technikát ismerik meg. Szóltunk róla, hogy a deklarativitás előnyt is jelent, saját tanári tapasztalatom szerint gyorsan megszokják, és az SQL általában jobban tanítható, mint a procedurális programnyelvek.
- Az utasítások egy részének kiértékelésekor nincs közvetlen visszajelzés (DDL, DML egy része, DCL). A hallgatóknak meg kell szokni, hogy ha nem kapnak az utasítás végrehajtása során hibajelzést, akkor az utasítás szintaktikailag helyes, de hogy a kívánt műveletet eredményezi-e, azt valamilyen módon ellenőrizni kell.
- Az eredményt visszaadó utasítások esetén az eredmény helyességének végiggondolása. Példaként említjük a táblázatok összekapcsolását, amikor egy JOIN művelet hiánya a ténylegesnél nagyobb számú, és nem valós kapcsolatokat tükröző sorokat eredményez.
- Az adatbázis-kezelő rendszer hibajelzéseinek megértése. Ez a probléma elsősorban a nem magyarított programok esetén jelentkezik. A hibajelzések-

kel még az is gond, hogy többnyire a szintaktikai hibákat jelzik, a szemantikai hibákat nem.

[3]-ban felsorolt problémák: az adatbázis sémára nem jól emlékeznek a hallgatók, nem értik a relációs modellt és az SQL alapelemeit, az SQL nyelv deklaratív volta.

A megszerkesztett SQL utasítások helyességének ellenőrzésére több módszer is lehetséges. Megnyitni a táblázatokat és szemmel ellenőrizni az eredményt. Természetesen ez csak kezdetben járható út, amikor néhány táblás, táblánként kevés sort tartalmazó adatbázissal dolgozunk. Az eredményben megjeleníteni azokat az oszlopokat is, amelyekre feltételt írunk elő. Más SQL lekérdezéssel próbáljuk meg ellenőrizni az eredményt. Például adatott feltételnek eleget tevő sorok megjelenítése mellett, a feltételnek eleget tevő sorok számát is lekérdezzük. Csoportosításnál ellenőrzés lehet az oszlop lekérdezése DISTINCT kulcsszóval. Logikailag végiggondolni, hogy az eredmény helyes-e. Például a JOIN műveletnél eldönteni, hogy mely táblázat sorainak száma domináns a művelet eredménye sorainak számánál.

Az SQL nyelv konkrét utasításainak tanulása során felmerülő problémák:

- Kiválasztási feltételek írása. Nem teljes relációk, a logikai műveletek és a predikátumok helytelen alkalmazása.
- Több táblázatból történő lekérdezés esetén a JOIN művelet teljes vagy részleges elhagyása.
- Csoportosítás szükségességének felismerése.
- A csoportjellemzők mellett más oszlopok adatai lekérdezésének megkísérlése.
- Aggregát függvények táblákon vagy csoportokon történő működése közti különbségtétel.
- Alkérdeések többszintű egymásba ágyazása.
- Alkérdeéseknél, ha az alkérdés egynél több értéket eredményez az ALL, ANY predikátumok elhagyása.
- Korrelált alkérdeések írása.
- Új adatsorok bevitelkor az oszloplista elemeinek megfelelő típusú értékek megadása az értéklistán.
- UPDATE és DELETE utasításoknál a WHERE feltétel véletlen elhagyása, ami nem kívánt hatást eredményez.
- Táblázatok létrehozásakor a CONSTRAINT záradékok használata.

A kiválasztási feltételeknél elsősorban a nem teljes relációk írása jelent problémát, például $ár >= 400$ and $≤ 500$. Annak ellenére elkövetik a hallgatók ezt a hibát, hogy megelőző programozási tanulmányaik során relációk, összetett logikai feltételek írásával már találkoztak. A hibára való hangsúlyos rámutatás, vagy a többoperandusú logikai műveletek helyett a predikátumok használata kiküszöböli a hibát.

A JOIN művelettel nem az SQL tanulásánál találkozunk először. A relációs műveletek ismertetésénél konkrét példát mutatunk a keresztszoratra és a természetes összekapcsolásra, amit akkor általában meg is értenek. Az SQL-nél hangsúlyozni kell, hogy ha egynél több táblázat tartalmazza a szükséges adatokat, a táblázatokat össze kell kapcsolni. Példán bemutatok két tábla keresztszorzatát, és jelzem, hogy ebből az elsődleges és idegen kulcs egyezősége szerint kell a sorokat kiválasztani. Még problémásabb, ha két táblázat közvetlenül nem, csak egy harmadik táblán keresztül kapcsolható össze. Segíthet a probléma megoldásánál az is, hogy példán, táblánál újra bemutatjuk a műveleteket.

A relációs műveletek tanításakor a csoportosítást egy konkrét néhány soros táblázaton mutatom be, tapasztalataim szerint a megértéssel nincs gond. A probléma annak felismerése a kérdés ismeretében, hogy a táblázatot csoportosítani kell-e. Az ilyen típusú kérdések: adja meg típusonként a kazetták számát, határozza meg hallgatóként a jegyek átlagát, stb. Arra kell rávilágítani, hogy egy oszlop különböző értékei szerint, az adott értéket tartalmazó sorokkal kell műveletet végezni- ez pedig csoportosítást igényel.

A csoportosítással a másik gond, hogy nemcsak a csoportjellemzőket, hanem más oszlopok tartalmát is le akarják kérdezni. Például milyen érdemjegyei vannak a jeles tanulmányi átlagú hallgatóknak. Példán kell bemutatni, hogy ez a kérdés csak egy másik utasítással válaszolható meg.

Az aggregát függvények táblákon és csoportokon dolgozhatnak, az első esetben a SELECT kifejezéslistája csak aggregát függvényt tartalmazhat, a második esetben olyan oszlopot is, melyre csoportosítottunk. A legfőbb hiba, hogy a SELECT a második formájú, de a csoportosítás elmarad. Erre az adatbázis-kezelő rendszerek hibajelzéssel figyelmeztetnek is. Megoldás a szabály megtanulása.

Az alkérdések megtanulása több problémát is rejt. Először is az alkérdés eredményét nem látják, tehát biztosnak kell lenni abban, hogy az tartamilag helyes. Problémát jelent az alkérdések többszintű egymásba ágyazása. Következésképpen végig kell gondolni, hogy milyen táblából milyen adathalmazt kell előállítani és milyen sorrendben. A korrelált alkérdéseknek már a megértése is nehéz, konkrét példán keresztül táblánál kell bemutatni, hogy történik a végrehajtása. Az ilyen alkérdés írásakor azt kell végiggondolni, hogy a külső kérdés táblázatának mely oszlopára kell feltételt előírni a belső kérdésben.

Az INSERT utasításoknál az oszloplista és értéklista elemszám egyezőségének, illetve az elemek adattípus szerinti megfelelésének megsértése az előforduló hiba. A séma megjelenítése segíthet a probléma kiküszöbölésében, illetve annak pontos ismerete, hogy melyik konstans milyen típusú.

Az UPDATE és DELETE utasításokban a WHERE záradék nem kötelező, ilyenkor a kívánt művelet a teljes táblán végrehajtódik. Az utasítások végrehajtása előtt célszerű felszólítani a hallgatókat, hogy ellenőrizzék a WHERE záradék meglétét. Másik lehetséges megoldás, hogy az utasításokat tranzakcióban használjuk, ahol egy teljes táblára vonatkozó műveletsor érvényteleníthető.

Táblázatok létrehozásakor nem az oszlopdefiníciók, hanem az elsődleges és idegen kulcs kijelölése, a hivatkozási épség fenntartási módjának megadása jelent problémát. A hallgatóknak rögzíteni kell, hogy az egy oszlopra vonatkozó előírás közvet-

lenül az oszlop definíciót követi, a több oszlopra vonatkozó pedig az összes oszlop definiálása után helyezendő el. A záradék helyes megadásához célszerű javasolni a sűgő használatát.

Fenti problémák megoldásában a tanár többféle támogatást adhat.

- Először is már tanítás közben felhívja a figyelmet a fenti problémákra.
- Példát mutat a helytelen és helyes megoldásokra.
- Többször megismétli azokat a szabályokat, melyekkel ezek elkerülhetők.
- Hosszabb, összetettebb záradékok írásánál javasolja a sűgő használatát.
- A problémás utasításokat gyakoroltatja, megmutatva a jó megoldást, hogy a hallgatók saját megoldásaikat ellenőrizhessék.
- Tesztfeladatokkal segíti az ismeretek rögzítését.
- Önálló, otthoni gyakorlásra ösztönöz, nyilvánossá téve a gyakorló kérdéssorok megoldásait ellenőrzés céljából.

A hallgatók által elkövetett általános hibák:

6. Gépelési hibák, kulcsszavak elírása.
7. Hibás oszlop vagy táblázatnevek. Ennek oka lehet, hogy az adatbázis sémára nem jól emlékeznek.
8. Az utasítások szintaktikai szabályait és működésmódját vagy nem értik, vagy nem tanulják meg, ezért alkalmazni sem tudják.
9. Magát a relációs modellt, a relációs műveleteket nem értik vagy nem tanulják meg.

A gépelési hibák nem a gyakorlatlanságból adódnak, mivel a mai hallgatók döntő többsége rendelkezik otthon is számítógéppel, és gyakorlott számítógép felhasználónak számít. A kulcsszavak elírása elsősorban azoknál a hallgatóknál fordul elő, akik az angolt nem ismerik. Ezt a hibát kiküszöbölni a kulcsszavak megtanulásával lehet.

Az oszlop és táblázatnevek elírásának megelőzésére célszerű a táblázatok felépítését és kapcsolatait megjelenítő grafikus diagram használatára ösztönözni a hallgatókat.

A szintaktikai szabályok és működésmód rögzítésének leghatékonyabb módja a számítógépes teszt, melynek során a hallgató ellenőrizheti válaszainak helyességét.

A relációs modell tanítása természetesen megelőzi az SQL nyelv tanítását, melynek során szintén tesztekkel, valamint számonkéréssel lehet serkenteni a hallgatókat az ismeretek megtanulására. A megértési problémák pedig a tanítás során folyamatos ellenőrző kérdésekkel előzhető meg.

6. Programozás SQL nyelven

Az SQL nyelv nemcsak interaktívan végrehajtható utasításkészlettel rendelkezik, hanem hosszú ideje lehetőség van a nyelv segítségével eljárások írására is. Az ide vonatkozó utasításokat szabványosították is.

Az SQL programozás a következő területeket foglalja magában:

- A nyelvben rendelkezésre álló deklarációs, elágazó, ciklus és egyéb utasítások ismertetése.
- A rekordhalmaz és kurzor fogalmának, a rekordhalmaz feldolgozási menetének bemutatása.
- A tárolt eljárások és függvények írása szabályainak megbeszélése.
- A triggerek funkciójának és felépítésének megismerése.

E témakör tanításánál probléma:

- A hallgatók programozási ismeretei szegényesek, több éve tanultak valamilyen procedurális programozási nyelvet.
- Ugyancsak nem megfelelő az algoritmizálási készségük.
- Eljárások és függvények írásának és hívásának szabályai, ha tanulták is valaha, teljesen feledésbe ment.
- A kurzor fogalma, a kurzoros feldolgozás menete, a kurzorok egymásba ágyazása.
- A rekordhalmaz fogalma, mely a memóriában áll rendelkezésre, abban a mozgást a feldolgozás során nekik kell megvalósítani.
- A triggerek működése, mely nem kézzelfogható, végrehajtásukhoz nem futtatni kell őket.

A tanítás során külön algoritmizálási gyakorlatra nincs idő. Az oktatás során többnyire rövid eljárásokat írunk, és reméljük, hogy a feledésbe került algoritmizálási ismeretek előjönnek. Ugyancsak ez mondható el az utasításokra, nem írunk külön példát az elágazás vagy ciklus utasítás működésének bemutatására. Az eljárások és függvények írásának és hívásának szabályait viszont részletesen ismertetjük, és természetesen a gyakorlatban is kipróbáljuk.

A kurzoros feldolgozás pár lépését nem nehéz megérteni, és rögzíteni kell, hogy használni tudják. Nem szokatlan a programozásban az, hogy a program írója fejben tartja a soron következő lépéseket. Ezért úgy gondoljuk, hogy az alap feldolgozási algoritmus bemutatása után a rekordhalmaz feldolgozása, a benne való mozgás némi gyakorlással elsajátítható. A triggerek működését egy olyan példán mutatjuk be, ahol szemmel látszik az eredmény, egy táblázatban végrehajtandó felújítási műveleteket egy másik táblázatban naplózzuk.

7. Számítógép bevonása az oktatásba

Tanári tapasztalatom szerint a számítógép bevonása a tanulásba nagyon eredményes. Ez több szinten valósul meg. Egyrészt az órákon a tananyagot tartalmazó bemutató kivetítésre kerül. Másodsor a tanulásához, gyakorláshoz az oktatási anyagok elhelyezésre kerülnek egy olyan szerverre, melyet a hallgatók otthonról is elérhetnek. Harmadszor valamilyen oktatási keretrendszert használunk a tanításban, számonkérésben. [5] szerint számítógép igénybevételel automatizálhatók az oktatásban a leckék, a sugórendszer, házi feladat és vizsga, valamint az adatbázisok elérése.

Az otthon megoldandó írott kérdések és tesztsorok ellenőrzése órán történik meg, amikor már a hallgató elfelejtette, mi okozott számára problémát. A számítógépes teszteknel beállítható, hogy közvetlen a válasz után, vagy a teszt kitöltésének végén a hallgató a helyes eredményt is megtekinthesse.

Saját oktatói gyakorlatomban a Moodle rendszert használom. A tesztkérdések többfélék lehetnek: felelt válaszolósak – egy vagy több jó válasz, párosító, rövid válaszos, esszé, stb. A tesztekben többnyire feleletválaszolós és párosító kérdéseket használunk, ahol előre adott válaszokból kell választani, vagy párosítani. Ezekkel a kérdésekkel az alábbi fajta ismeretek ellenőrizhetők:

- Szintaktikai és szemantikai szabályok.
- Záradékfunkciók ismerete.
- Szintaktikailag, tartalmilag jó és hibás utasítások közül a helyes kiválasztása.

A leghatékonyabb tanulás, ha a hallgató maga szerkeszti meg egy adott kérdéshez az SQL utasítást. Erre a rövid válaszos és az esszékérdések alkalmasak. Probléma van az ellenőrzéssel. Az esszékérdés tanári értékelést igényel, mely időben hosszabb ideig tart. A rövid válaszos kérdéseknél a helyes megoldást megadjuk a gépnek, de ha a hallgató más betűtípussal, több szóközzel, vagy zárójellel adja meg válaszát, már nem fog a rendszer azonosságot találni. Nem kap választ a hallgató arra sem, hogy ha hibás a megoldása miért az.

Ilyen problémák kiküszöbölésére több felsőoktatási intézményben saját SQL oktató rendszert fejlesztettek ki. Ilyen rendszerek például: eSQL, SQL-Tutor, AsseSQL, SQLater, SQLify [1], [3], [4]. Ezen rendszerek főbb jellemzői:

- Adatbázis sémák megjelenítése.
- Utasítások lépésenkénti végrehajtása és eredmény megjelenítés.
- Angolul megfogalmazott kérdésekre SQL utasítás generálása.
- Utasítások hibáinak kijelzése. Többosztű hibarendszer, nemcsak a szintaktikai, hanem a tartalmi hibákat is jelzi
- SQL utasítások helyességének automatikus értékelése.

8. Néhány gyakorlati tesztkérdés

Saját gyakorlatomban az SQL nyelv tanítását közel 120 tesztkérdés segíti. Néhány példa ezek közül:

1. Mi a funkciója a SELECT utasítás egyes záradékainak? Rendelje össze az oszlopokat!

SELECT	rendezés
FROM	csoportosítás
WHERE	választás sorokból
GROUP BY	választás csoportokból
HAVING	táblázatok megadása
ORDER BY	oszlopok, kifejezések megadása

2. Milyen műveletet hajt végre az adatbázis-kezelő rendszer a táblázatokkal, ha csak táblázatnevek szerepelnek a FROM záradékban?
 - Természetes összekapcsolás.
 - Egyesítés.
 - Keresztszorzat.
3. Melyik a helyes kifejezés, ha azt akarjuk megvizsgálni, hogy az Ár oszlop értéke 100-500 közé esik?
 - $\text{Ár} \geq 100 \text{ AND } \leq 500$
 - $\text{Ár} >= 100 \text{ AND } \text{Ár} <= 500$
 - $\text{Ár BETWEEN } 100 \text{ AND } 500$
4. A LIKE predikátum mintájában
 - a % karakter tetszőleges számú, a _ egyetlen karaktert helyettesít.
 - a _ karakter tetszőleges számú, a % egyetlen karaktert helyettesít.
 - nem használható helyettesítő karakter.
5. Ha a SELECT záradék kifejezés listája csak aggregát függvényt tartalmaz, a SELECT utasításban
 - szerepelnie kell GROUP BY záradéknak.
 - nem szerepelhet GROUP BY záradék.
 - az utasítás GROUP BY záradékkal vagy a nélkül is helyes lehet.
6. Az adatbázis a következő táblázatot tartalmazza:

Kölcsönzők(kölcsönzőkód, kölcsönzőnév, cím)

Kölcsönzés(kölcsönzőkód, kazettakód, elvitel, visszavitel)

Melyik utasítás helyes, ha a kölcsönző neve mellett az általa kölcsönzött kazetta kódját és az elvitel dátumát akarjuk megjeleníteni?

- SELECT kölcsönzőnév, kazettakód, elvitel
FROM kölcsönzők k INNER JOIN kölcsönzés z
- SELECT kölcsönzőnév, kazettakód, elvitel
FROM kölcsönzők k INNER JOIN kölcsönzés z
ON k.kölcsönzőkód=z.kölcsönzőkód
- SELECT kölcsönzőnév, kazettakód, elvitel
FROM kölcsönzők k, kölcsönzés z
WHERE k.kölcsönzőkód=z.kölcsönzőkód

9. Hallgatói felmérés értékelése

Az Adatbázis-kezelés tárgyat tanuló hallgatók körében kérdőíves felmérést készítettem, ahol a kérdések egy része konkrétan az SQL nyelvre vonatkozott. Ebben a fejezetben röviden ismertetem a felmérés eredményét. A kérdőívet 46 hallgató töltötte ki.

Informatikai tárgy tanításakor fontos, hogy mennyire motiváltak a hallgatók, mennyire érdekli őket az informatika.

Kérdés	Lehetséges válaszok	Hallgatói válaszok
Mennyire érdekli az informatika?	nem érdekel	0%
	nem érdekel, de ismeretét szükségesnek tartom	0%
	érdekel, de nem akarok profi informatikus lenni	82%
	nagyon érdekel, profi informatikus akarok lenni	18%
Jelölje meg, hogy melyik állítással ért egyet!	jó érzékem van az informatikához	36%
	az informatika kedvenc tárgyaim közé tartozik	21%
	az informatika gyorsan fejlődik, ezért önállóan is képzem magam	68%
	az informatika szeretete motivál a tanulásban	29%
Mennyire használ számítógépet és mire?	nem használok számítógépet	0%
	alkalomszerűen használok, de főleg ját-	0%

Kérdés	Lehetséges válaszok	Hallgatói válaszok
	szom, internetezek, levelezek	
	rendszeresen használom, de főleg játszom, internetezek, levelezek	14%
	rendszeresen használom feladataim megoldására	86%

1. Táblázat: Hallgatói motiváltság

A válaszokból kitűnik, hogy minden hallgatót érdekel az informatika, nagyrészüket, mivel ez a tudományterület gyorsan fejlődik, önállóan is képi magát, és valamennyi hallgató rendszeresen használ számítógépet. A hallgatók tehát kellően motiváltak a tanulásban, és ezt tanári tapasztalatom is megerősíti.

A tanítás szempontjából érdekes az is, hogy mennyien rendelkeznek már némi SQL ismerettel. A tárgy felvétele előtt a hallgatók 29%-a foglalkozott már adatbázis-kezeléssel, konkrétan SQL-el 18%, MySQL-el 18%, MySQL adatbázisok kezelésével PHP nyelvből 14%. Akik foglalkoztak már SQL-el általában ismereteiket kívánják rendszerezni és elmélyíteni. Számukra a nyelv tanítása során haladó megoldásokat is mutatni kell.

Megkértem a hallgatókat, hogy jelöljék meg az alábbi állítások közül melyekkel értenek egyet.

Állítás	Egyetért
A tárgyat könnyűnek tartom.	7%
Az SQL nyelvet deklaratív volta miatt könnyű megtanulni, és használni.	25%
Egy bonyolult SQL lekérdezési feladat megrettent.	61%
A programozást szeretem, ezért az adatbázis programozást könnyűnek tartom.	7%
Az önállóan megoldandó tesztek nehéznek tartom.	14%
Élveztem a számítógépes tesztekkel, így könnyebb volt a tanulás.	43%

2. Táblázat: A tárgyat és az SQL-t minősítő állítások

A válaszokból kitűnik, hogy a hallgatók a tárgyat és az SQL-t is nehéznek tartják, és a többség szerint megtanulni sem könnyű. Ennek a megállapításnak két dolog is ellentmond. Tanári tapasztalatom szerint az SQL sokkal jobban, kisebb erőfeszítéssel tanulható, mint a hagyományos procedurális nyelvek. Az algoritmus megtalálása sokkal nehezebb, mint az SQL utasítás megszerkesztése. Másrészt a hallgatói számonkérés sem azt a véleményt támasztja alá. Az SQL zárthelyi dolgozat 7 tesztkérdésből és 10 darab lekérdezési feladatból állt, a maximális pontszám 40 pont. A hallgatói eredmények átlaga 30,4 pont, ami 76%-os teljesítmény, mely a mi osztályozási

gyakorlatunk szerint négyes érdemjegynek felel meg. A teszteknek az oktatásban való alkalmazása a hallgatók szerint is jó megoldás.

Megkérdeztem azt is, hogy az Adatbázis-kezelés tárgy témái közül, mit törölnének a hallgatók a tematikából. Az adatmodellezést senki, az SQL nyelv tanítását 7%, eljárások írását SQL-ben 43%.

Az órai munka során felmerülő problémák közül négy vonatkozott az SQL nyelv-re.

Probléma	Hallgatók aránya
nyelvi szabályok megértése (SQL)	36%
nyelvi szabályok betartása, alkalmazása (SQL)	39%
lekérdezések kódolása (SQL utasítások megszerkesztése)	29%
eljárások kódolása (SQL nyelven)	39%

3. Táblázat: Órai munkában megjelenő problémák

A százalékos arányok elég magasak. A nyelvi szabályok megértése terén javíthat, ha minden szabályra azonnal példát is mutatunk. A szabályok betartásának és alkalmazásának előfeltétele, hogy a hallgató megtanulja azokat, tehát a tanárnak erre kell ösztönözni. Ezt követően a gyakorlás hozhat eredményt. A lekérdezések és eljárások kódolása annál sikeresebb, minél többet gyakorolják. A szűk órai keretek miatt ezt a hallgatóknak önállóan kell megtenni, a tanárnak pedig biztosítani kell a megoldásokat, hogy a hallgatók ellenőrizhessék magukat.

Megkérdeztem azt is, hogy az SQL nyelv tanulása során mi jelentettek problémát a hallgatók számára.

Probléma	Hallgatói arány
nyelvi szabályok megtanulása, alkalmazása	18%
kiválasztási feltételek megírása	4%
predikátumok használata	21%
aggregát függvények használata	21%
csoportosítás	14%
összekapcsolási (join) műveletek	29%
alkérdéses lekérdezések	39%
tábla és egyéb objektumok létrehozása, szerkezet módosítása	14%
táblázatok adatainak felújítása	7%
adatfelügyeleti utasítások	18%

4. Táblázat: Az SQL nyelv tanuláskor megjelenő problémák

A válaszok tükrözik, hogy az általam tárgyalt problémákat többnyire ők is problémának tartják. Kiugró a JOIN művelet és az alkérdés alkalmazása magas százaléka. Az ezeket tartalmazó utasítások megszerkesztése valóban nem automatikus, végig kell gondolni, hogy mely táblázatokat milyen feltételekkel kell összekapcsolni, illetve, hogy mit kérdezzünk le alkérdéssel, és annak az eredményét melyik záradékban hogyan kívánjuk felhasználni. Megoldás lehet az utasítások lépésenkénti szerkesztése, illetve a megfelelő mennyiségű gyakorlás.

A hallgatóknak egy általuk választott feladathoz kell házi feladatként egy Access alkalmazást elkészíteni, melynek tartalmaznia kell minimum tíz, SQL nyelven megírt lekérdezést. A hallgatók 32%-a számára ez gondot jelentett. Ennek okának a nem elegendő gyakorlást tartom.

Végül megkérdeztem azt is, hogy a tanultak alapján milyen adatbázis tervezési, kezelési feladatra tartják magukat alkalmasnak. 29% azt válaszolta, hogy nem sajátította el olyan mélyen az anyagot, hogy azt alkalmazni tudja. Mindössze 14% válaszolta azt, hogy alkalmasnak tartja magát relációs adatbázisok kezelésére SQL nyelven.

Irodalom

1. Julia Coleman Prior: *Online assessment of SQL query formulation skills*, Proceedings of the fifth Australasian conference on Computing education - Volume 20, Adelaide, Australia. Pages: 247 – 256, 2003
2. Czenky Márta: *Adatmodellezés, SQL és Access alkalmazás, SQL Server és ADO*, ComputerBooks Kiadó, Budapest, 2005
3. S. Dekeyser – M. de Raadt - Tien Yu Lee: *Computer assisted assessment of SQL query skills*, ACM International Conference Proceeding Series; Vol. 242, Proceedings of the eighteenth conference on Australasian database - Volume 63, Ballarat, Victoria, Australia, Pages: 53 – 62, 2007
4. Antonija Mitrovic: *Learning SQL with a computerized tutor*, Technical Symposium on Computer Science Education, Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education, Atlanta, Georgia, United States, Pages: 307 – 311, 1998
5. Jeffrey D. Ullman: *Improving the Efficiency of Database-System Teaching*, ACM SIGMOD International Conference on Management of Data, pp 1-3 ,2003