

DEBRECENI EGYETEM
Informatikai Kar

TANULMÁNYI NYILVÁNTARTÁS
RENDSZERTERVE

Témavezető:
Dr. Kovács Béláné
tudományos munkatárs

Készítette:
Beke Béla
IV. EPM

Debrecen
2009

Tartalomjegyzék

1. Bevezetés.....	4
2.A Microsoft Accessről röviden.....	6
3.A jelenlegi rendszer bemutatása.....	7
4. Elvárások az új rendszerrel szemben.....	8
5. Adatfolyam diagram.....	10
5.1.A tanulmányi nyilvántartás adatfolyam diagramjai:.....	10
5.1.1.-0.- Főfolyamatok (0. szint):.....	10
5.1.2.-1.- Iskolai adatok nyilvántartása (1. szint).....	11
5.1.3.-2.- Iskola tevékenységek (1.szint).....	11
5.1.4.-3-. Félévi / év végi statisztikák (1.Szint).....	12
5.1.5. -1.6-Osztályok és diákok karbantartása (2.Szint).....	12
5.1.6.-2.1-Év eleji tevékenységek (2.Szint).....	13
5.1.7.-2.2- Év végi / Évközi tevékenységek (2.Szint).....	13
5.2.Folyamatok	14
5.2.1.Fő folyamat.....	14
5.2.2.Folyamat 'Iskolai adatok nyilvántartása'.....	14
5.2.3.Folyamat 'Osztályok és diákok karbantartása'.....	17
5.2.4.Folyamat 'Iskolai tevékenységek'.....	18
5.2.5.Folyamat 'Év eleji tevékenységek'.....	19
5.2.6.Folyamat 'Év végi / Évközi tevékenységek'.....	21
5.3.Logikai adatmodellezés	23
5.4.Egyed-Kapcsolat diagram:.....	24
5.5.Entitás lista.....	25
5.6.Entitások.....	25
6.Felhasználói dokumentáció.....	30
6.1.Bejelentkezés az alkalmazásba.....	30
6.2.Munkakörnyezet beállítása.....	30
6.3.Az iskolatitkár funkciói a programba.....	31
6.4.Alapadatok karbantartása.....	31
6.4.1.Iskola adatai:.....	31
6.4.2.Tanévek	32
6.4.3.Tanárok, Tantárgyak, Feljegyzéstípusok.....	32
6.4.4.Osztályok.....	33
6.4.5.Diákok.....	34
6.5.Tanév eleji teendők	34
6.5.1.Diákok osztályba rendezése.....	34
6.5.2.Osztályfőnökök karbantartása.....	35
6.5.3.Osztályok tantárgyai	35
6.5.4.Tantárgyak diákjai.....	36
6.5.5.Diákok osztályváltása.....	37
6.6.Tanév / Félév végi teendők.....	37
6.6.1.Félévi /Évvégi jegyek rögzítése.....	37
6.6.2.Diákok hiányzásai	38
6.6.3.Diákokkal kapcsolatos feljegyzések.....	39
6.6.4.Pótvizsga eredmények rögzítése.....	39
6.6.5.Diákok átadása a következő tanévbe.....	40

6.6.6.Végzett és felsőbb éves tanulók státuszának beállítása.....	40
6.7.Kimutatások, Statisztikák.....	41
6.7.1.Iskolastatisztika.....	41
6.8.Tantárgyi Bukások.....	41
7.Összefoglalás.....	42
8.Irodalomjegyzék.....	43
9.Melléletek.....	44
9.1.1. számú melléklet – Az adatbázis szerkezetét kialakító kód	44
9.2.Iskolastatisztika és Bukási statisztika jelentések	60

1. Bevezetés

A tanulói adatok és változásainak nyilvántartása, a diákok tanulmányi előmenetelének követése minden középszintű oktatási intézménynek fontos feladata. A tanulók általános iskolai jelentkezésétől, a középiskolai eredményeiken át, az érettségi utáni szakképesítés és vagy felsőfokú tanulmányok követéséig.

A középiskolák vezetőinek feladatai között szerepel az iskola jelenlegi tanulmányi helyzetének naprakész ismerete, eredményeinek elemzése, ezek összevetése a korábbi évek eredményeivel, valamint a tapasztalt tendenciák vizsgálata. Az eredmények alapján az iskola további munkájának megtervezése. A szülők, a fenntartó, az oktatási minisztérium folyamatos, időszakos és eseti tájékoztatása az egyes tanulók és az iskola tanulmányi munkájáról, eredményeiről és terveiről. Ezen feladatok ellátásához szükséges adatok összegyűjtése a papír alapú és részben elektronikusan, általában táblázatkezelőben kezelt kapcsolódó táblákban, tárolt adatok segítségével nehézkes és időigényes. A fentiek alapján elkészített kimutatások, jelentések, statisztikák aktualizálása az adatok változása esetén, csak erőforrás pazarló módon valósítható csak meg.

Ezen feladatok, minél tökéletesebb, gyorsabb megvalósításának egyik legfontosabb eszköze egy iskolai szinten is egységes tanulmányi nyilvántartó rendszer és egy ehhez kapcsolódó vezetői információs rendszer, amely az iskolák működéséhez szükséges és a működés során keletkező adatokat konzisztens módon kezeli, lehetővé teszi a jelenlegi és a jövőben felmerülő információ igény kielégítését. Lecsökkentve a redundanciát, az adatbiztonsági levárásoknak megfelelő szintűre, és az ebből fakadó nyilvántartási és jelentési hibákat, segítve az iskola vezetőségének, tanárainak adminisztrációs munkáját

Szakdolgozatomban ezen igen bonyolult rendszer egyik részének megtervezését tűztem ki célul. Ezen részrendszer egyik fontos feladata, hogy az Önkormányzat, mint fenntartó, részére tanév közben illetve félévek végén esedékes rendszeres adatszolgáltatást (például: a félévenkénti iskolastatisztika, osztálystatisztikák, tantárgyi statisztikák... stb.) biztosítsa, valamint a tanulóknak, mint az iskola legfontosabb szereplőinek, a tanulmányi és egyéb előmenetelét, adatváltozásainak követését biztosítsa. Figyelembe véve a jelenlegi, a debreceni Gábor Dénes EMK a debreceni Gábor Dénes EMSZKI-ban, és a korábbi, munkahelyemen a Brassai Sámuel Gimnázium és MSZKI-ban szerzett tapasztalataimat és vezetőség illetve a munkaközösség vezetők részéről felmerült igényeket. A tervezés során nagy hangsúlyt fektettem arra, hogy a általam elkészített rendszerterv később bővíthető legyen az aktuálisan felmerülő igényeknek megfelelően. A rendszerterv elkészítése során az SSADM rendszertan szemléletmódját eszközeit követtem

A szakdolgozat részét képezi a fenti rendszerterv egy részletének megvalósítása is. A megvalósításhoz a minden közoktatási intézmény számára jogtisztán elérhető, és az iskola

informatikai végzettségű dolgozói által ismert és a tanórákon is tanított Microsoft Access adatbázis kezelő rendszert választottam. Választásomkor szemelőt tartottam, a középfokú intézmény(ek) anyagi és személyi feltételeit.

2. A Microsoft Accessről röviden

A Microsoft Access egy Windows környezetbe illesztett relációs adatbázis kezelő rendszer melynek segítségével az információ könnyen tárolható, kereshető, rendezhető és megjeleníthető. A rendszer teljes mértékben kihasználja a Windows grafikus lehetőségeit, objektumorientált megközelítésnek köszönhetően az adatbázis elemek és a kezelési módszerek egységesebbek és összehangoltak.

A szoftver számos eszközt biztosít a felhasználók és az alkalmazás fejlesztők számára egyaránt. Néhányat kiemelve a sok közül:

- lekérdezés, űrlap és jelentés varázslók,
- vizuális tervezők,
- szabványos SQL felület,
- makró programozási lehetőség,
- VBA (Visual Basic for Application),
- adatbázis karbantartó eszközök,
- felhasználókezelés.

A nagyobb adatbázisok, összetettebb funkcióinak megvalósításai, túl mutatnak a beépített varázslók által kínált lehetőségeken, ilyen alkalmazások fejlesztéséhez a Microsoft Access beépített VBA programozási lehetőséget biztosít a fejlesztők számára. A VBA egy rendkívül hatékony és gyors fejlesztést biztosító programozási nyelv. A Microsoft Access, adatok kezelésére alapvetően két adat-hozzáférési modellt támogat ezek közül az egyik a DAO (Data Access Object), amely a Microsoft által fejlesztett JET Adatbázist motort használja. A DAO-t a Microsoft Access által is alapértelmezésként használt MDB adatbázis formátumhoz optimalizálták. A DAO és egyben az Access egyik erőssége abban rejlik, hogy támogatja az ODBC adatforrások elérésén túl a Paradox, dBase, FoxPro és Excel formátumokat is. Támogatja több adatbázis egyidejű elérését, felhasználók és csoportok kialakítását, a felhasználók jogosultságalapú hozzáféréseit az adatbázis objektumaihoz, rekordzárolási zárolási lehetőségek beállítását az objektumok által kezelt rekordokra. A másik, és egyben a Microsoft által is ma már elsődlegesen támogatott, adat-hozzáférési modell az ADO (ActiveX Data Objects), ami az OLE DB adat-hozzáférési megoldásait felhasználva, biztosít elérést relációs és nem relációs adatbázisok, levelezőrendszerek, fájlrendszerek eléréséhez.

Szakdolgozatom megírása során DAO adat-hozzáférési modellt használtam.

3. A jelenlegi rendszer bemutatása

Jelenleg a tanulmányi adatok nyilvántartása az iskolában papír alapon történik. Az iskolai nyilvántartás legfontosabb dokumentumai:

- A *beírási napló* az iskolatitkár által kezelt dokumentum, amely az iskolába beiratkozott diákok személyes adatait illetve a beiratkozás és a távozás időpontját tartalmazza.
- A *törzskönyv* az osztály diákjainak, az adott képzés folyamán elért eredményeinek, tanulmányi előmenetelének legfontosabb dokumentuma, amely tartalmazza a tanulók törzslapszámát, anyakönyvi és lakcím adatait, általános iskolai és középiskolai bizonyítványának számát, valamint évfolyamonként a tanuló osztálynaplóbeli sorszámát, tantárgyait, a tantárgyak éves óraszámát és év végi érdemjegyeit, valamint tanév és évfolyam során a tanulmányi munkához kapcsolódó záradékokat.
- Az *osztálynapló*, egy osztály éves munkájának legfontosabb dokumentuma, amely sorszámozott lapjai tartalmazzák a tanulók törzslapszámát, anyakönyvi és lakcím adatait, tantárgyait, a tantárgyakból év közben kapott jegyeket havi bontásba és félévi és az év végi érdemjegyeit, valamint tanév és évfolyam során kapott a tanulmányi munkához kapcsolódó jutalmakat büntetéseket, a tanuló hiányzásait havi bontásban és félévenként összesítve. Tartalmazza még az órarendet, a tanított tantárgyak neveit és tanárait, az osztály ülésrendjét, és különböző dátumokhoz köthető létszámadatokat, az év végi tanulmányi eredmények összesítését.
- A *bizonyítvány*, amely a törzslap alapján kerül kitöltésre és az abban található adatokat mindenben megegyezik. Ez a dokumentum a diák tulajdonát képviseli, ezzel igazolja hitelt érdemlően eddigi tanulmányi előmenetelét.
- Az *ellenőrzőkönyv* az osztálynapló adatait tartalmazza „optimális” esetben, a szülők tájékoztatásra valamint a szülő és az iskola illetve az osztályfőnök közötti információáramlás egyik fontos dokumentuma.

A tanulók személyes adatainak és tanulmányi eredményeinek nyilvántartása, jelenleg több országos kötelező használatú rendszerben(KIR, Taninform) is rögzítésre kerül, azonban ezek a rendszerek jelenleg még az iskola vezetőinek számára nem biztosítanak könnyen hozzáférhető és rugalmasan módosítható jelentéskészítő eszközöket, így az iskola a fenntartó felé küldött jelentéseit munkafüzetekben tárolt táblázatokból nyert adatok alapján készíti el, szövegszerkesztő program segítségével.

4. Elvárások az új rendszerrel szemben

Az új rendszerrel kapcsolatban az iskolák vezetése, munkaközösségei az alábbi elvárásokat fogalmazták meg:

- A tanulók adatainak egységes központi tárolása, módosítása, a papír alapú tárolással összhangban.
- Több felhasználós működés, jogosultságalapú hozzáférés
- A tanulói adatok gyors elérése, módosítása.:
- Tantárgyankénti csoportbontások helyes kezelése osztályon belül. Pl.: Az idegen nyelvet, vagy a szakmai tárgyakat csoportbontásban tanuló diákok esetén meg lehessen mondani, hogy ki tanította és osztályozta az adott diákot az adott tanévben
- Összevont tantárgyak kezelése (pl.: szakmacsoportos alapozó oktatás, ezen tárgyak több részből állnak, melyek mindegyikéből kap jegyet a diák, és minden részből külön ,külön megbukhat és pótvizsgázhat, de a bizonyítványába csak az összevont tantárgyi jegy kerül be.)
- Tanulói előmenetel nyomon követése, erre vonatkozó adatok megjeleníthetősége
- Szülők adatainak tárolása, kapcsolattartás céljából. pl.: telefonszám
- Az év közbeni osztályfőnök váltások kezelése
- Félévenként statisztikák gyors automatikus elkészítése

Iskolastatisztika:

- Létszámadatok alakulása (osztályonként, évfolyamonként, osztálytípusonként, iskolai szinten, nemek, létszámcsoporthoz (debreceni, bejáró, albérlő, kollégista),osztályozhatóság szerint)
- Átlagok alakulása (osztályonként, évfolyamonként és osztálytípusonként iskolai szinten, nemek, létszámcsoporthoz (debreceni, bejáró, albérlő, kollégista) rangsorba állítva)
- Átlagok megoszlása minősített bontásban(kitűnő, jeles, jó, ..., stb), %-os megoszlása megadásával (osztályonként, évfolyamonként és osztálytípusonként iskolai szinten)
- Tantárgyak sorrendje iskolai szinten, a szerzet jegyek eloszlásának és százalékos eloszlásának feltüntetésével
- Kitűnő, és jeles tanulók névsora osztály és átlag feltüntetésével.

- Bukások száma egy, kettő, három vagy több tantárgyból és összesen, százalékos arány feltüntetésével (osztályonként, évfolyamonként és osztálytípusonként iskolai szinten)
- Bukásmentes osztályok listája
- Hiányzási osztály rangsor átlag alapján, elméleti, gyakorlati órák külön összesítve és átlagolva
- nem mulasztott tanulók névsora

Osztálystatisztika

- Létszámadatok alakulása (nemek, létszámcsoporthoz (debreceni, bejáró, albérlő, kollégista), osztályozhatóság szerint), összlétszám
- Átlagok alakulása (diákonként rangsorolva, nemek, létszámcsoporthoz (debreceni, bejáró, albérlő, kollégista) rangsorba állítva)
- Átlagok megoszlása minősített bontásban(kitűnő, jeles, jó, ..., stb), %-os megoszlása megadásával
- Tantárgyak sorrendje osztály szinten, a szerzett jegyek eloszlásának és százalékos eloszlásának feltüntetésével
- Kitűnő, és jeles tanulók névsora osztály és átlag feltüntetésével.
- Bukások száma egy, kettő, három vagy több tantárgyból diáknév, tantárgy, tanár név feltüntetésével
- Hiányzási adatok, elméleti, gyakorlati órák külön összesítve és átlagolva
- nem mulasztott tanulók névsora
- nem osztályozható tanulók névsora

Tantárgyi statisztikák

- A tantárgyból szerzett jegyek megoszlása és átlaga (osztályonként, évfolyamonként, osztálytípusonként és iskolai szinten)

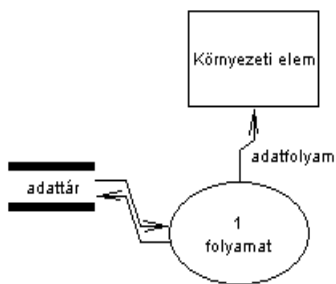
Bukási statisztika iskolai szinten

- tantárgynév, és évfolyam szerint csoportosítva a tanuló nevének osztályának és tanárának feltüntetésével

5. Adatfolyam diagram

Az adatfolyam diagram a rendszer és környezete, valamint a rendszer folyamatainak ,és a közöttük áramló adatok, megjelenítésére szolgál. Az adatfolyam diagramok négy legfontosabb eleme: a folyamat , környezeti elem, adatfolyam, adattár. Az általam használ CASE eszköz az SSADM-ben szokásostól kicsit eltérő jelöléseket és elnevezéseket használ.

- A környezeti elemnek megfelelő véglezárót (terminátort) egy az elem nevét tartalmazó téglalap jelöli

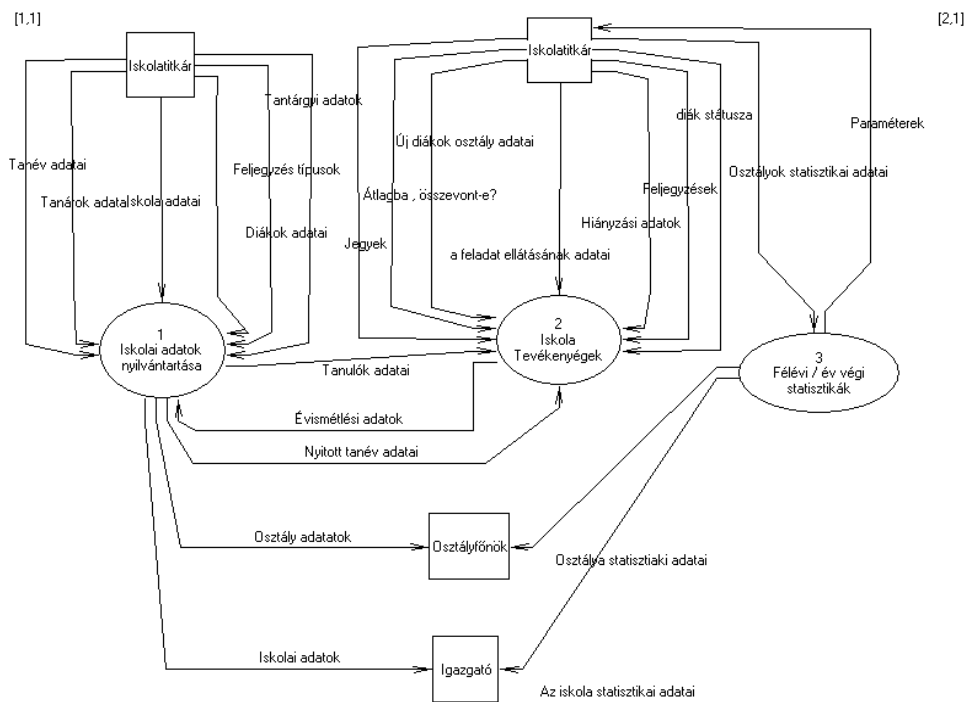


jelölik

- A folyamatokat, a folyamat nevét és számát tartalmazó ellipszis jelöli.
- Az adatfolyamokat folytonos nyilak
- az adattárakat pedig alul és felül vastagon szegélyezett téglalapok

5.1. A tanulmányi nyilvántartás adatfolyam diagramjai:

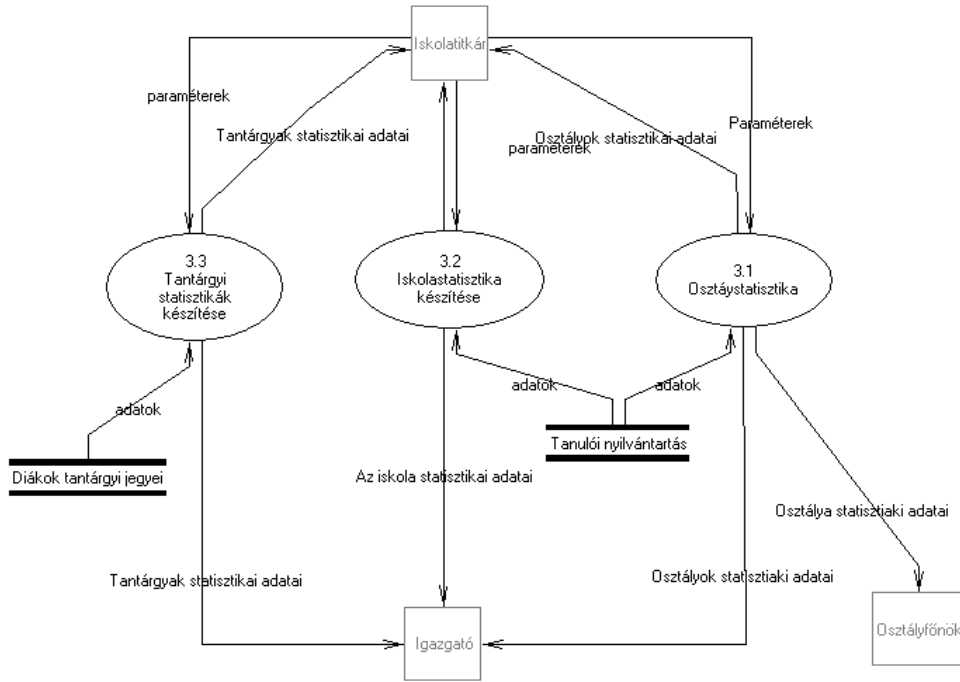
5.1.1. -0.- Főfolyamatok (0. szint):



5.1.4. -3-. Félévi / év végi statisztikák (1.Szint)

[1.1]

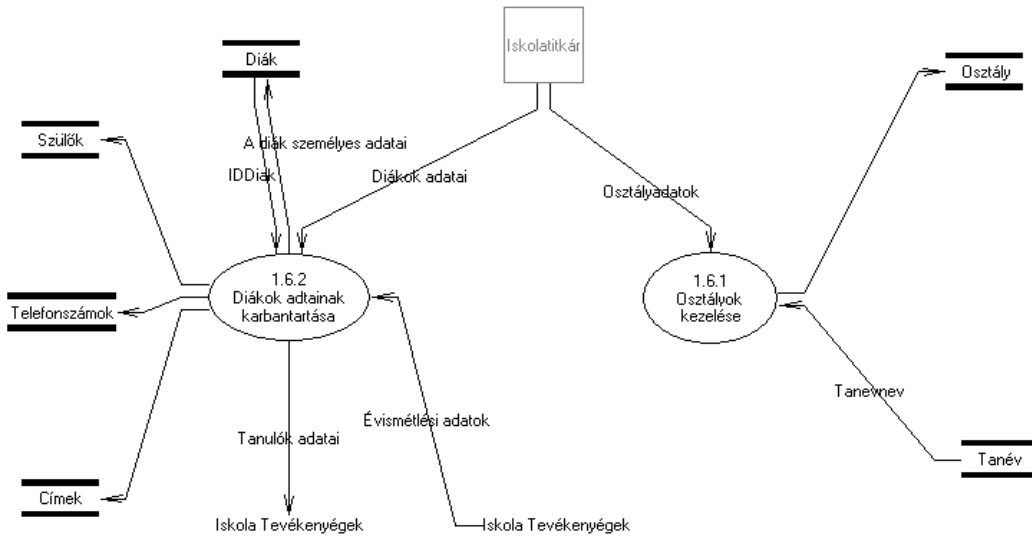
[2.1]



5.1.5. -1.6-Osztályok és diákok karbantartása (2.Szint)

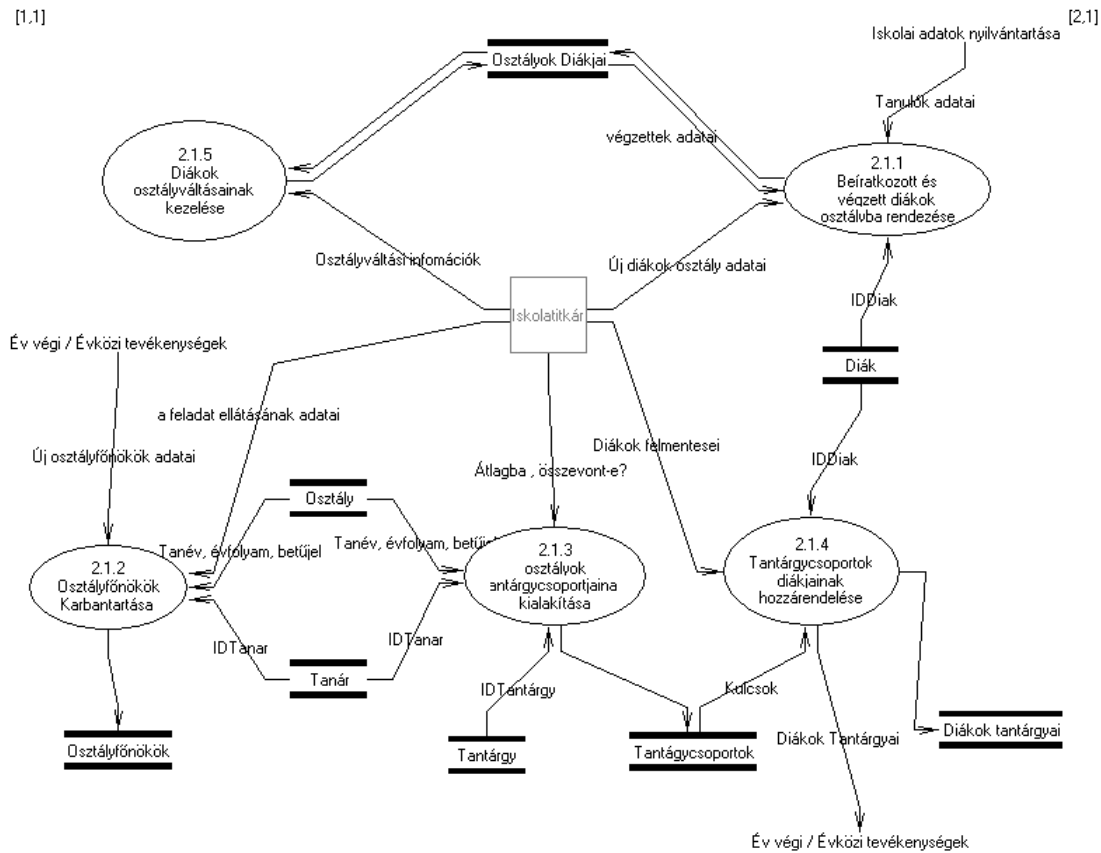
[1.1]

[2.1]



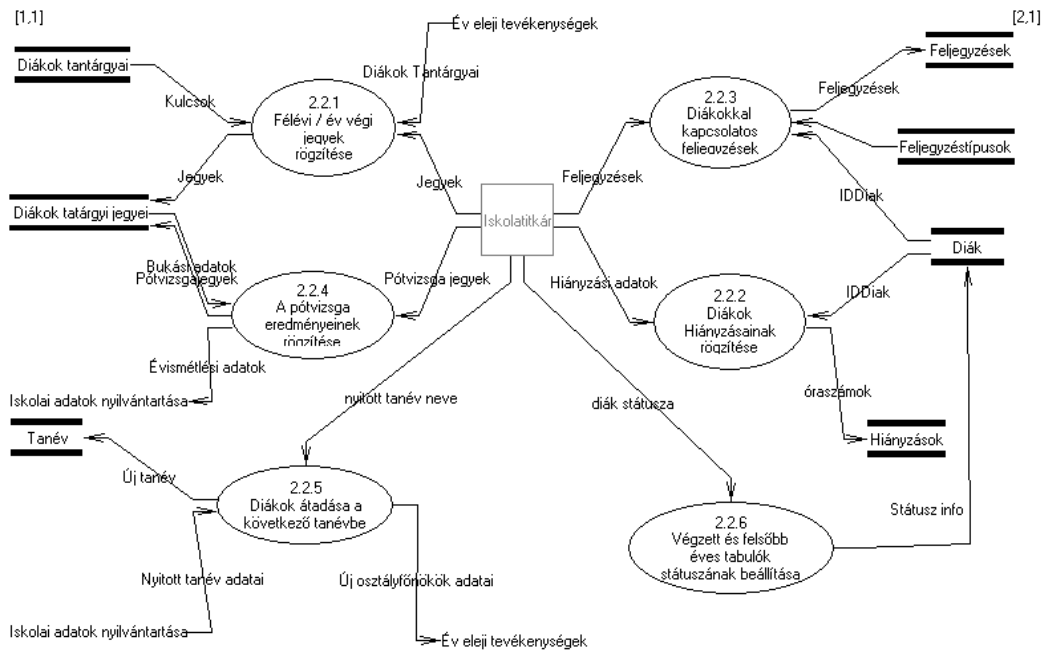
5.1.6. -2.1-Év eleji tevékenységek (2.Szint)

[1.1]



5.1.7. -2.2- Év végi / Évközi tevékenységek (2.Szint)

[1.1]



5.2. Folyamatok

5.2.1. Fő folyamat

Folyamatlista

Folyamatszám	Folyamatnév	Legalacsonyabb szint	Külső	Folyamat	Alfolyamatszám
1	Iskolai adatok nyilvántartása	NEM	NEM	Fő folyamat	7
2	Iskolai tevékenységek	NEM	NEM	Fő folyamat	2
3	Félévi / év végi statisztikák	NEM	NEM	Fő folyamat	3

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	NEM
Szint	1	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

Folyamat 'Iskolai tevékenységek'

Folyamatszám	2		
Folyamatnév	Iskolai tevékenységek	Külső	NEM
Szint	1	Legalacsonyabb szint	NEM
Specifikáció	A tanév során elvégzendő tevékenységek		
Leírás	Az iskolai napi és időszakos tevékenységei		

Folyamat 'Félévi / év végi statisztikák'

Folyamatszám	3		
Folyamatnév	Félévi / év végi statisztikák	Külső	NEM
Szint	1	Legalacsonyabb szint	NEM
Specifikáció	Félévi / év végi statisztikák automatizált elkészítése a bevitt adatok segítségével		
Leírás	A félévi és év végi statisztikák automatikus elkészítése		

5.2.2. Folyamat 'Iskolai adatok nyilvántartása'

Folyamatlista

Folyamatszám	Folyamatnév	Legalacsonyabb szint	Külső	Folyamat	Alfolyamatszám
1.1	Iskola adatainak karbantartása	IGEN	NEM	Iskolai adatok nyilvántartása	-----
1.2	Tantárgyak karbantartása	IGEN	NEM	Iskolai adatok nyilvántartása	-----
1.3	Tanárok karbantartása	IGEN	NEM	Iskolai adatok nyilvántartása	-----
1.4	Tanévek karbantartása	IGEN	NEM	Iskolai adatok nyilvántartása	-----
1.5	Feljegyzéstípusok karbantartása	IGEN	NEM	Iskolai adatok nyilvántartása	-----
1.7	Általános Lekérdezések	IGEN	NEM	Iskolai adatok nyilvántartása	-----
1.6	Osztályok és diákok karbantartása	NEM	NEM	Iskolai adatok nyilvántartása	2
2	Iskolai tevékenységek	NEM	IGEN	Iskolai adatok nyilvántartása	0
2	Iskolai tevékenységek	NEM	IGEN	Iskolai adatok nyilvántartása	0
2	Iskolai tevékenységek	NEM	IGEN	Iskolai adatok nyilvántartása	0

Folyamat 'Iskola adatainak karbantartása'

Folyamatszám	1.1		
Folyamatnév	Iskola adatainak karbantartása	Külső	NEM
Szint	2	Legalacsonyabb szint	IGEN
Specifikáció	Az iskola, mint intézmény, azonosító adatainak karbantartása		

Folyamat 'Tantárgyak karbantartása'

Folyamatszám	1.2		
Folyamatnév	Tantárgyak karbantartása	Külső	NEM
Szint	2	Legalacsonyabb szint	IGEN
Specifikáció	Tantárgyak karbantartása		

Folyamat 'Tanárok karbantartása'

Folyamatszám	1.3		
Folyamatnév	Tanárok karbantartása	Külső	NEM
Szint	2	Legalacsonyabb szint	IGEN
Specifikáció	Tanárok karbantartása		

Folyamat 'Tanévek karbantartása'

Folyamatszám	1.4		
Folyamatnév	Tanévek karbantartása	Külső	NEM
Szint	2	Legalacsonyabb szint	IGEN
Specifikáció	Tanévek meghatározó adatainak, időpontjainak karbantartása		

Folyamat 'Feljegyzéstípusok karbantartása'

Folyamatszám	1.5		
Folyamatnév	Feljegyzéstípusok karbantartása	Külső	NEM
Szint	2	Legalacsonyabb szint	IGEN
Specifikáció	A tanulók tanulmányi előmenetelében nyilvántartani kívánt események, eredmények típusainak karbantartása		

Folyamat 'Általános Lekérdezések'

Folyamatszám	1.7		
Folyamatnév	Általános Lekérdezések	Külső	NEM
Szint	2	Legalacsonyabb szint	IGEN
Specifikáció	Az iskola szereplőit érintő általános lekérdezések		

Folyamat 'Osztályok és diákok karbantartása'

Folyamatszám	1.6		
Folyamatnév	Osztályok és diákok karbantartása	Külső	NEM
Szint	2	Legalacsonyabb szint	NEM
Leírás	Osztályok és diákok karbantartását végző folyamatok		

Folyamat 'Iskolai tevékenységek'

Folyamatszám	2		
Folyamatnév	Iskolai tevékenységek	Külső	IGEN
Szint	2	Legalacsonyabb szint	NEM
Specifikáció	A tanév során elvégzendő tevékenységek		
Leírás	Az iskolai napi és időszakos tevékenységei		

Folyamat 'Iskolai tevékenységek'

Folyamatszám	2		
Folyamatnév	Iskolai tevékenységek	Külső	IGEN
Szint	2	Legalacsonyabb szint	NEM
Specifikáció	A tanév során elvégzendő tevékenységek		
Leírás	Az iskolai napi és időszakos tevékenységei		

Folyamat 'Iskolai tevékenységek'

Folyamatszám	2		
Folyamatnév	Iskolai tevékenységek	Külső	IGEN
Szint	2	Legalacsonyabb szint	NEM
Specifikáció	A tanév során elvégzendő tevékenységek		
Leírás	Az iskolai napi és időszakos tevékenységei		

5.2.3. Folyamat 'Osztályok és diákok karbantartása'

Folyamatlista

Folyamatszám	Folyamatnév	Legalacsonyabb szint	Külső	Folyamat	Alfolyamatszám
1.6.1	Osztályok kezelése	IGEN	NEM	Osztályok és diákok karbantartása	-----
1.6.2	Diákok adatainak karbantartása	IGEN	NEM	Osztályok és diákok karbantartása	-----
2	Iskolai tevékenységek	NEM	IGEN	Osztályok és diákok karbantartása	0
2	Iskolai tevékenységek	NEM	IGEN	Osztályok és diákok karbantartása	0

Folyamat 'Osztályok kezelése'

Folyamatszám	2001-06-01		
Folyamatnév	Osztályok kezelése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	A tanév osztályainak karbantartása		

Folyamat 'Diákok adatainak karbantartása'

Folyamatszám	1.6.2		
Folyamatnév	Diákok adatainak karbantartása	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	A diákok személyes adatainak, szülői adatok és elérhetőségek karbantartása		

Folyamat 'Iskolai tevékenységek'

Folyamatszám	2		
Folyamatnév	Iskolai tevékenységek	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Specifikáció	A tanév során elvégzendő tevékenységek		
Leírás	Az iskolai napi és időszakos tevékenységei		

Folyamat 'Iskolai tevékenységek'

Folyamatszám	2		
Folyamatnév	Iskolai tevékenységek	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Specifikáció	A tanév során elvégzendő tevékenységek		
Leírás	Az iskolai napi és időszakos tevékenységei		

5.2.4. Folyamat 'Iskolai tevékenységek'

Folyamatlista

Folyamatszám	Folyamatnév	Legalacsonyabb szint	Külső	Folyamat	Alfolyamatszám
2.1	Év eleji tevékenységek	NEM	NEM	Iskolai tevékenységek	5
2.2	Év végi / Évközi tevékenységek	NEM	NEM	Iskolai tevékenységek	6
1	Iskolai adatok nyilvántartása	NEM	IGEN	Iskolai tevékenységek	0
1	Iskolai adatok nyilvántartása	NEM	IGEN	Iskolai tevékenységek	0
1	Iskolai adatok nyilvántartása	NEM	IGEN	Iskolai tevékenységek	0

Folyamat 'Év eleji tevékenységek'

Folyamatszám	2.1		
Folyamatnév	Év eleji tevékenységek	Külső	NEM
Szint	2	Legalacsonyabb szint	NEM
Leírás	Év eleji tevékenységek		

Folyamat 'Év végi / Évközi tevékenységek'

Folyamatszám	2.2		
Folyamatnév	Év végi / Évközi tevékenységek	Külső	NEM
Szint	2	Legalacsonyabb szint	NEM
Leírás	Év végi / évközi tanulmányi munkát érintő tevékenységek		

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	IGEN
Szint	2	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	IGEN
Szint	2	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	IGEN
Szint	2	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

5.2.5. Folyamat 'Év eleji tevékenységek'

Folyamatlista

Folyamatszám	Folyamatnév	Legalacsonyabb szint	Külső	Folyamat	Alfolyamatszám
2.1.1	Beiratkozott és végzett diákok osztályba rendezése	IGEN	NEM	Év eleji tevékenységek	-----
2.1.2	Osztályfőnökök Karbantartása	IGEN	NEM	Év eleji tevékenységek	-----
2.1.3	osztályok tantárgycsoportjainak kialakítása	IGEN	NEM	Év eleji tevékenységek	-----
2.1.4	Tantárgycsoportok diákjainak hozzárendelése	IGEN	NEM	Év eleji tevékenységek	-----
2.1.5	Diákok osztályváltásainak kezelése	IGEN	NEM	Év eleji tevékenységek	-----
1	Iskolai adatok nyilvántartása	NEM	IGEN	Év eleji tevékenységek	0
2.2	Év végi / Évközi tevékenységek	NEM	IGEN	Év eleji tevékenységek	0
2.2	Év végi / Évközi tevékenységek	NEM	IGEN	Év eleji tevékenységek	0

Folyamat 'Beiratkozott és végzett diákok osztályba rendezése'

Folyamatszám	2.1.1		
Folyamatnév	Beiratkozott és végzett diákok osztályba rendezése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Beiratkozott és végzett diákok osztályba rendezése		

Folyamat 'Osztályfőnökök Karbantartása'

Folyamatszám	2.1.2		
Folyamatnév	Osztályfőnökök Karbantartása	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Osztályfőnökök kijelölése, munkájuk kezdetének, és befejezésének rögzítése		

Folyamat 'osztályok tantárgycsoportjainak kialakítása'

Folyamatszám	2.1.3		
Folyamatnév	osztályok tantárgycsoportjainak kialakítása	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Osztályok tantárgycsoportjainak kialakítása, paraméterek beállítása. Az átlagba beszámít-e, összevont tantárgy-e?		

Folyamat 'Tantárgycsoportok diákjainak hozzárendelése'

Folyamatszám	2.1.4		
Folyamatnév	Tantárgycsoportok diákjainak hozzárendelése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Diákok tantárgycsoportokhoz rendelése, felmentések beállítása		

Folyamat 'Diákok osztályváltásainak kezelése'

Folyamatszám	2.1.5		
Folyamatnév	Diákok osztályváltásainak kezelése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Diákok év közbeni osztályváltásainak kezelése		

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

Folyamat 'Év végi / Évközi tevékenységek'

Folyamatszám	2.2		
Folyamatnév	Év végi / Évközi tevékenységek	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Leírás	Év végi / évközi tanulmányi munkát érintő tevékenységek		

Folyamat 'Év végi / Évközi tevékenységek'

Folyamatszám	2.2		
Folyamatnév	Év végi / Évközi tevékenységek	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Leírás	Év végi / évközi tanulmányi munkát érintő tevékenységek		

5.2.6. Folyamat 'Év végi / Évközi tevékenységek'

Folyamatlista

Folyamatszám	Folyamatnév	Legalacsonyabb szint	Külső	Folyamat	Alfolyamatszám
2.2.1	Félévi / év végi jegyek rögzítése	IGEN	NEM	Év végi / Évközi tevékenységek	-----
2.2.2	Diákok Hiányzásainak rögzítése	IGEN	NEM	Év végi / Évközi tevékenységek	-----
2.2.3	Diákokkal kapcsolatos feljegyzések	IGEN	NEM	Év végi / Évközi tevékenységek	-----
2.2.4	A pótvizsga eredményeinek rögzítése	IGEN	NEM	Év végi / Évközi tevékenységek	-----
2.2.5	Diákok átadása a következő tanévbe	IGEN	NEM	Év végi / Évközi tevékenységek	-----
2.2.6	Végzett és felsőbb éves tanulók státuszának beállítása	IGEN	NEM	Év végi / Évközi tevékenységek	-----
2.1	Év eleji tevékenységek	NEM	IGEN	Év végi / Évközi tevékenységek	0
1	Iskolai adatok nyilvántartása	NEM	IGEN	Év végi / Évközi tevékenységek	0
1	Iskolai adatok nyilvántartása	NEM	IGEN	Év végi / Évközi tevékenységek	0
2.1	Év eleji tevékenységek	NEM	IGEN	Év végi / Évközi tevékenységek	0

Folyamat 'Félévi / év végi jegyek rögzítése'

Folyamatszám	2.2.1		
Folyamatnév	Félévi / év végi jegyek rögzítése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Félévi / év végi jegyek rögzítése		

Folyamat 'Diákok Hiányzásainak rögzítése'

Folyamatszám	2.2.2		
Folyamatnév	Diákok Hiányzásainak rögzítése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Diákok Hiányzásainak rögzítése		

Folyamat 'Diákokkal kapcsolatos feljegyzések'

Folyamatszám	2.2.3		
Folyamatnév	Diákokkal kapcsolatos feljegyzések	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Diákokkal kapcsolatos feljegyzések készítése		

Folyamat 'A pótvizsga eredményeinek rögzítése'

Folyamatszám	2.2.4		
Folyamatnév	A pótvizsga eredményeinek rögzítése	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	A pótvizsga eredményeinek rögzítése, ennek megfelelően az évismételő státusz beállítása		

Folyamat 'Diákok átadása a következő tanévbe'

Folyamatszám	2.2.5		
Folyamatnév	Diákok átadása a következő tanévbe	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Diákok átadása a következő tanévbe		

Folyamat 'Végzett és felsőbb éves tanulók státuszának beállítása'

Folyamatszám	2.2.6		
Folyamatnév	Végzett és felsőbb éves tanulók státuszának beállítása	Külső	NEM
Szint	3	Legalacsonyabb szint	IGEN
Specifikáció	Végzett és felsőbb éves tanulók státuszának beállítása		

Folyamat 'Év eleji tevékenységek'

Folyamatszám	2.1		
Folyamatnév	Év eleji tevékenységek	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Leírás	Év eleji tevékenységek		

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

Folyamat 'Iskolai adatok nyilvántartása'

Folyamatszám	1		
Folyamatnév	Iskolai adatok nyilvántartása	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Specifikáció	Az iskola alapvető adatainak, mint: Diákok, Tanárok, Tantárgyak, Tanévek, Osztályok, Osztály Diákjai ... stb. karbantartása		
Leírás	Az iskola tanárainak, tanéveinek, diákjainak, osztályainak ... stb. nyilvántartása		

Folyamat 'Év eleji tevékenységek'

Folyamatszám	2.1		
Folyamatnév	Év eleji tevékenységek	Külső	IGEN
Szint	3	Legalacsonyabb szint	NEM
Leírás	Év eleji tevékenységek		

5.3. Logikai adatmodellezés

Az adatmodellek három elem segítségével adhatók meg. Ezen elemek közül az első az *egyed*, vagy entitás, amely a valós világ egy jól definiált más objektumoktól megkülönböztethető eleme. A második a *tulajdonság* vagy attribútum, mellyel a egyedeket írjuk le. A harmadik pedig a *kapcsolat*, amely a egyedek között fennálló viszonyokat adja meg. Mindhárom elem esetén szokás az elemeket a típus utótaggal említeni, kiemelve az absztrakt jellegét. Ebben az elemek konkrét értékét előfordulásnak nevezzük.

Egy rendszer vizsgálata során a logikai adatmodellezésnek az a célja, hogy a megtaláljuk a rendszer egyedeit az őket leíró tulajdonságokat, és feltárjuk a közöttük lévő kapcsolatokat. Konkrétabban a intézmény jelen esetben az iskola, működéséhez szükséges adatok belső szerkezetének meghatározása, leírása célunk, függetlenül a későbbi feldolgozási követelmények, valamint a feldolgozó és tarolóeszközök lehetőségeitől függetlenül.

A rendszer kezelésére választott adatbázis kezelő a Microsoft Access, mint korábban már említette relációs adatbázis kezelő rendszer, melyben az egyedeket táblázatokkal adjuk meg. A táblázatokban az oszlopok a tulajdonságoknak, sorai pedig az egyed előfordulásoknak felelnek meg. A tanulói nyilvántartás adatmodelljének kialakítása során a relációs adatmodellnél alkalmazott normalizálás módszerének lépéseit alkalmaztuk.

A normalizálás során az adatmodellt valamilyen normálformára hozzuk. Ez lehet első (1NF), második (2NF), harmadik(3NF), Boyce-Codd (BCNF), Negyedik, (4NF) vagy ötödik (5NF) normálforma akár, a valós rendszerek megvalósításánál általában elegendő az adatmodellt harmadik normálformára hozni. Mivel ez a modell már mentes a törlési módosítási, és bővítési anomáliáktól és a a tranzitív függéstől is. A reláció:

- 1 NF-ban van: ha minden attribútum elemi azaz minden előfordulásra egy értéket vesz fel, az attribútumok nem lehetnek többértékűek, és összetettek sem. A táblázatok minden sorában és oszlopában pontosan egy érték állhat.
- 2 NF-ban van: ha a reláció 1 NF-ban van és ha egyetlen leíró attribútum sem függ részlegesen a reláció elsődleges kulcsától.

A leíró attribútum olyan attribútum, amely a reláció egyik kulcsának sem része.

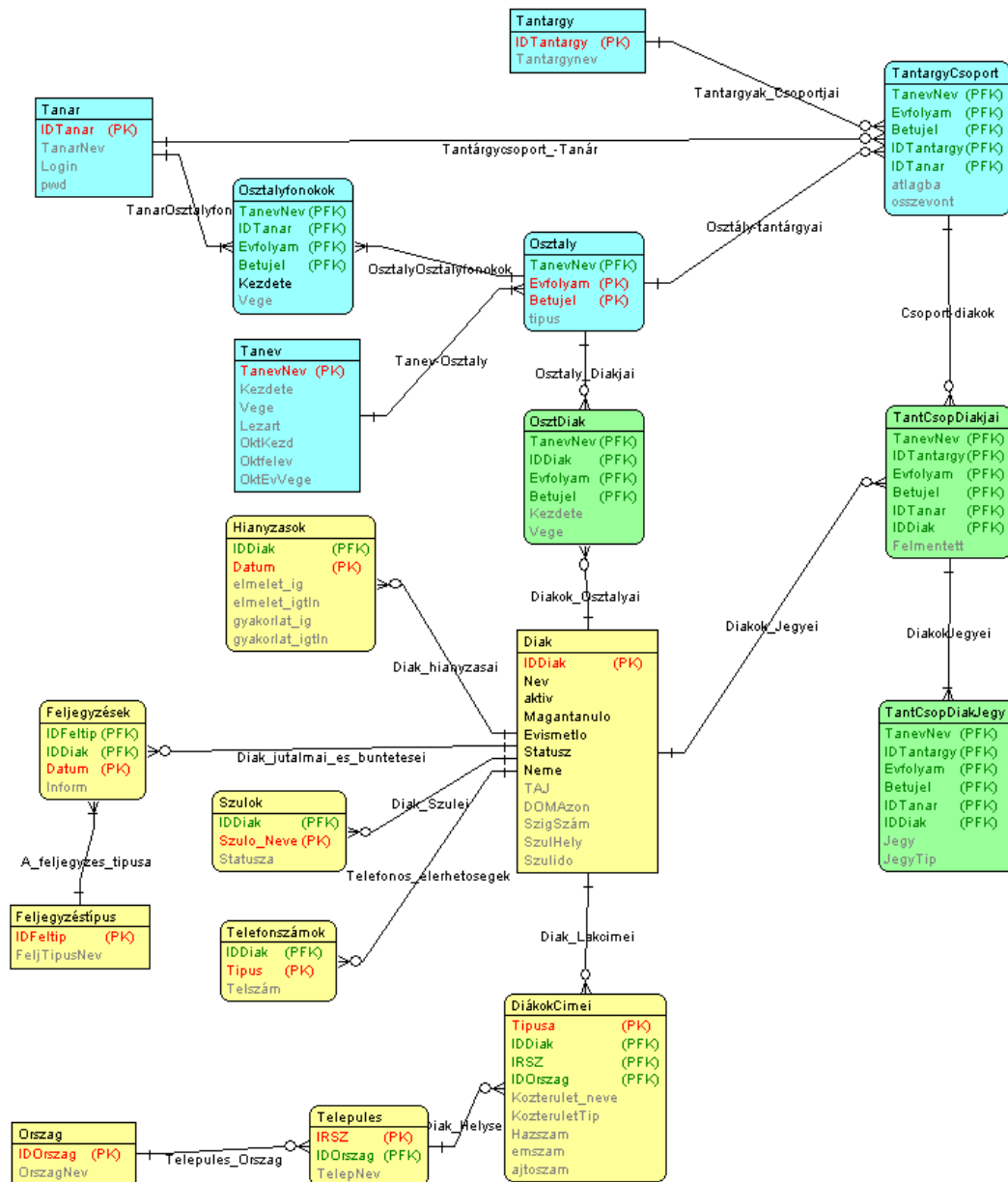
- 3 NF-ban van: ha 2 NF-ban van és ha egyetlen leíró attribútum sem függ tranzitíven a reláció elsődleges kulcsától.

Mint látható a normálformák hierarchikusan egymásra épülő rendszert alkotnak.

5.4. Egyed-Kapcsolat diagram:

[1.1]

[2.1]



5.5. Entitás lista

Entitásnév	Entitás típus	Elsődleges kulcs
Diak	Független	IDDiak
DiákokCimei	Függő	Tipusa, IDDiak, IRSZ, IDOrszág
Feljegyzések	Függő	IDFeltip, IDDiak, Datum
Feljegyzéstípus	Független	IDFeltip
Hianyások	Függő	IDDiak, Datum
Ország	Független	IDOrszág
Osztály	Függő	TanevNev, Evfolyam, Betujel
Osztályfonokok	Függő	TanevNev, IDTanar, Evfolyam, Betujel
OszDiak	Függő	TanevNev, IDDiak, Evfolyam, Betujel
Szülök	Függő	IDDiak, Szulo_Neve
Tanar	Független	IDTanar
Tanev	Független	TanevNev
Tantargy	Független	IDTantargy
TantargyCsoport	Függő	TanevNev, Evfolyam, Betujel, IDTantargy, IDTanar
TantCsopDiakjai	Függő	TanevNev, IDTantargy, Evfolyam, Betujel, IDTanar, IDDiak
TantCsopDiakJegy	Függő	TanevNev, IDTantargy, Evfolyam, Betujel, IDTanar, IDDiak
Telefonszámok	Függő	IDDiak, Tipus
Település	Függő	IRSZ, IDOrszág

5.6. Entitások

Entitás 'Diak'					
Entitás típus:		Független			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PK	IDDiak	AutoNumber LI	NEM	NEM	A tanuló azonosítója
	Nev	Text (50)	IGEN	NEM	A tanuló neve
	aktiv	Yes/No	IGEN	NEM	A tanuló az iskolába jár-e
	Magantanulo	Yes/No	IGEN	NEM	A tanulo magantanuló-e
	Evismetlo	Yes/No	IGEN	NEM	A tanulo évismétlő-e
	Statusz	Text (10)	IGEN	NEM	A tanuló Létszámcsoportaja (Debreceni, Bejáró, Kollégista)
	Neme	Text (5)	IGEN	NEM	A tanulo neme
	TAJ	Text (12)	NEM	NEM	A tanuló TAJ száma
	DOMAzon	Text (11)	NEM	NEM	A tanuló oktatási azonosítója
	SzigSzám	Text (10)	NEM	NEM	A tanuló személyi igazolvány száma
	SzulHely	Text (30)	NEM	NEM	A tanuló születési helye

Entitás 'Diákok Címei'					
Entitás típus: <i>Függő</i>					
Attribútumok					
<i>Kulcs</i>	<i>Attribútum-/szerepnév</i>	<i>Adattípus</i>	<i>Nem nulla</i>	<i>Egyedi</i>	<i>Leírás</i>
PK	Tipusa	Text (12)	NEM	NEM	A lakcím típusa : ideiglenes, állandó
PFK	IDDiak	Long Integer	NEM	NEM	A diák azonosítója
PFK	IRSZ	Text (10)	NEM	NEM	A lakcím irányító száma
PFK	IDOrszag	Text (5)	NEM	NEM	A lakcím országkódja
	Kozterulet_neve	Text (50)	NEM	NEM	A lakcím közterületének neve
	KozteruletTip	Text (5)	NEM	NEM	A lakcím közterületének típusa
	Hazszam	Text (5)	NEM	NEM	A lakcím házszáma
	emszam	Integer	NEM	NEM	A lakcím házszáma
	ajtoszam	Integer	NEM	NEM	A lakcím emelet száma

Entitás 'Feljegyzések' Általános információ					
Entitás típus: <i>Függő</i>					
Attribútumok					
<i>Kulcs</i>	<i>Attribútum-/szerepnév</i>	<i>Adattípus</i>	<i>Nem nulla</i>	<i>Egyedi</i>	<i>Leírás</i>
PFK	IDFeltip	Long Integer	NEM	NEM	A feljegyzéstípus azonosítója
PFK	IDDiak	Long Integer	NEM	NEM	A tulajdonos diák azonosítója
PK	Datum	Date/Time	NEM	NEM	A feljegyzés dátuma
	Inform	Text (50)	NEM	NEM	A feljegyzés pontos leírása, paraméterei

Entitás 'Feljegyzéstípus'					
Entitás típus: <i>Független</i>					
Attribútumok					
<i>Kulcs</i>	<i>Attribútum-/szerepnév</i>	<i>Adattípus</i>	<i>Nem nulla</i>	<i>Egyedi</i>	<i>Leírás</i>
PK	IDFeltip	AutoNumber LI	NEM	NEM	Feljegyzéstípus azonosítója
	FeljTípusNev	Text (50)	NEM	NEM	Feljegyzéstípus megnevezése

Entitás 'Hiányzasok'					
Entitás típus: <i>függő</i>					
Attribútumok					
<i>Kulcs</i>	<i>Attribútum-/szerepnév</i>	<i>Adattípus</i>	<i>Nem nulla</i>	<i>Egyedi</i>	<i>Leírás</i>
PFK	IDDiak	Long Integer	NEM	NEM	A hiányzó diák azonosítója
PK	Datum	Text (50)	NEM	NEM	A rögzítés dátuma
	elmélet_ig	Integer	NEM	NEM	elméleti igazolt órák száma
	elmélet_igtln	Integer	NEM	NEM	elméleti igazolatlan órák száma
	gyakorlat_ig	Integer	NEM	NEM	gyakorlati igazolt órák száma
	gyakorlat_igtln	Integer	NEM	NEM	gyakorlati igazolatlan órák száma

Entitás 'Ország'					
Entitás típus: <i>Független</i>					
Attribútumok					
<i>Kulcs</i>	<i>Attribútum-/szerepnév</i>	<i>Adattípus</i>	<i>Nem nulla</i>	<i>Egyedi</i>	<i>Leírás</i>
PK	IDOrszag	Text (5)	NEM	NEM	Az ország hámbetűs kódja
	OrszagNev	Text (50)	NEM	NEM	Az ország neve

Entitás 'Osztaly' Általános információ					
Entitás típus:		függő			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	TanevNev	Text (10)	IGEN	NEM	A tanév azonosítója
PK	Evfolyam	Integer	NEM	NEM	Az osztály évfolyama
PK	Betujel	Text (1)	NEM	NEM	Az osztály betűjele
	tipus	Text (50)	NEM	NEM	Az osztálytípusa (gimnáziumi, szakközépiskolai, szakképző)

Entitás 'Osztalyfonokok'					
Entitás típus:		Függő			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	TanevNev	Text (10)	IGEN	NEM	Az osztályfőnökség tanév azonosítója
PFK	IDTanar	Long Integer	NEM	NEM	Az osztályfőnök tanár azonosítója
PFK	Evfolyam	Integer	NEM	NEM	Az osztály évfolyama
PFK	Betujel	Text (1)	NEM	NEM	Az osztály betűjele
	Kezdet	Date/Time	IGEN	NEM	Az osztályfőnökség kezdete
	Vege	Date/Time	NEM	NEM	Az osztályfőnökség vége

Entitás 'OsztDiak'					
Entitás típus:		Függő			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	TanevNev	Text (10)	IGEN	NEM	Az osztálytagság tanévének azonosítója
PFK	IDDiak	Long Integer	NEM	NEM	A diák azonosítója
PFK	Evfolyam	Integer	NEM	NEM	Az osztály évfolyama
PFK	Betujel	Text (1)	NEM	NEM	Az osztály betűjele
	Kezdet	Date/Time	NEM	NEM	Osztálytagság kezdet
	Vege	Date/Time	NEM	NEM	Osztálytagság vége

Entitás 'Szulok' Általános információ					
Entitás típus:		függő			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	IDDiak	Long Integer	NEM	NEM	A diák azonosítója
PK	Szulo_Neve	Text (50)	NEM	NEM	A szülő neve
	Statusza	Text (25)	NEM	NEM	státusz (édes v nevelő, apa, vagy anya)

Entitás 'Tanar'					
Entitás típus:		Független			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PK	IDTanar	AutoNumber LI	NEM	NEM	A tanár azonosítója
	TanarNev	Text (50)	NEM	NEM	A tanár neve

Entitás 'Tanev'					
Entitás típus:		Független			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PK	TanevNev	Text (10)	IGEN	NEM	A tanév azonosítója formátuma 2007/2008
	Kezde	Date/Time	NEM	NEM	Tanév kezdete
	Vege	Date/Time	NEM	NEM	A tanév vége
	Lezart	Yes/No	NEM	NEM	A tanév állapotát jelzi lezárt/nem lezárt
	OktKezd	Date/Time	NEM	NEM	Az oktatás kezdete: ált. 09.01.
	Oktfelev	Date/Time	NEM	NEM	Félév vége: ált. 02.01.
	OktEvVege	Date/Time	NEM	NEM	Az oktatás vége: ált. 06.15.

Entitás 'Tantargy'					
Entitás típus:		Független			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PK	IDTantargy	AutoNumber LI	NEM	NEM	A tantárgy azonosítója
	Tantargynev	Text (50)	NEM	NEM	A tantárgy neve

Entitás 'TantargyCsoport'					
Entitás típus:		Függő			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	TanevNev	Text (10)	IGEN	NEM	A tantárgycsoport tanévének azonosítója
PFK	Evfolyam	Integer	NEM	NEM	A tantárgycsoport évfolyama
PFK	Betujel	Text (1)	NEM	NEM	A tantárgy csoport betűjele
PFK	IDTantargy	Long Integer	NEM	NEM	A tanított tantárgy azonosítója
PFK	IDTanar	Long Integer	NEM	NEM	A tantárgyat tanító tanár azonosítója
	atlagba	Yes/No	NEM	NEM	átlagszámításnál figyelembe kell-e venni a jegyeket

Entitás 'TantCsopDiakjai'					
Entitás típus:		Függő			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	TanevNev	Text (10)	IGEN	NEM	A tantárgycsoport tanévének azonosítója
PFK	IDTantargy	Long Integer	NEM	NEM	A tantárgycsoport évfolyama
PFK	Evfolyam	Integer	NEM	NEM	A tantárgy csoport betűjele
PFK	Betujel	Text (1)	NEM	NEM	A tanított tantárgy azonosítója
PFK	IDTanar	Long Integer	NEM	NEM	A tantárgyat tanító tanár azonosítója
PFK	IDDiak	Long Integer	NEM	NEM	A diák azonosítója
	Felmentett	Yes/No	NEM	NEM	A diák felmentett-e a tárgyból

Entitás 'TantCsopDiakJegy'					
Entitás típus:					
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	TanevNev	Text (10)	IGEN	NEM	A tantárgycsoport tanévének azonosítója
PFK	IDTantargy	Long Integer	NEM	NEM	A tantárgycsoport évfolyama
PFK	Evfolyam	Integer	NEM	NEM	A tantárgy csoport betűjele
PFK	Betujel	Text (1)	NEM	NEM	A tanított tantárgy azonosítója
PFK	IDTanar	Long Integer	NEM	NEM	A tantárgyat tanító tanár azonosítója
PFK	IDDiak	Long Integer	NEM	NEM	A diák azonosítója
	Jegy	Byte	NEM	NEM	A tantárgycsoportból a diák által szerzett jegy
	JegyTip	Text (2)	NEM	NEM	A tantárgycsoportból a diák által szerzett jegy típusa FE- félévi EV-év végi JV -javítóvizsga

Entitás 'Telefonszámok'					
Entitás típus:		<i>függő</i>			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PFK	IDDiak	Long Integer	NEM	NEM	A diák azonosítója
PK	Tipus	Text (7)	NEM	NEM	Telefonszám típusa (Diák száma, Anya Száma, Apa száma)
	Telszám	Text (20)	NEM	NEM	Telefonszám

Entitás 'Telepules'					
Entitás típus:		<i>függő</i>			
Attribútumok					
Kulcs	Attribútum-/szerepnév	Adattípus	Nem nulla	Egyedi	Leírás
PK	IRSZ	Text (10)	NEM	NEM	A település irányítószáma
PFK	IDOrszag	Text (5)	NEM	NEM	A település országának azonosítója
	TelepNev	Text (50)	NEM	NEM	A település neve

6. Felhasználói dokumentáció

Az alkalmazások készítésének egyik fontos lépése a dialógusok terveinek, elkészítése, a kialakításukkor tartalmi összetartozást és a felhasználói igényeket egyaránt figyelembe kell venni. A tervezés során megadjuk, hogy milyen vezérlőelemeket, hol és mire használunk a párbeszédablakokban.

6.1. Bejelentkezés az alkalmazásba

A Tanulói nyilvántartásba való bejelentkezés felhasználói név és jelszó páros segítségével történik, mely alapján a rendszer eldönti, hogy igazgató iskolatitkár vagy osztályfőnök jelentkezett be. A bejelentkező képernyő az ACCESS szabványos bejelentkezési felülete.



Az név és jelszó megadása után, a felhasználó részére előzetesen beállított jogosultságoknak megfelelő, fér hozzá az objektumokhoz a nyilvántartásban.

6.2. Munkakörnyezet beállítása

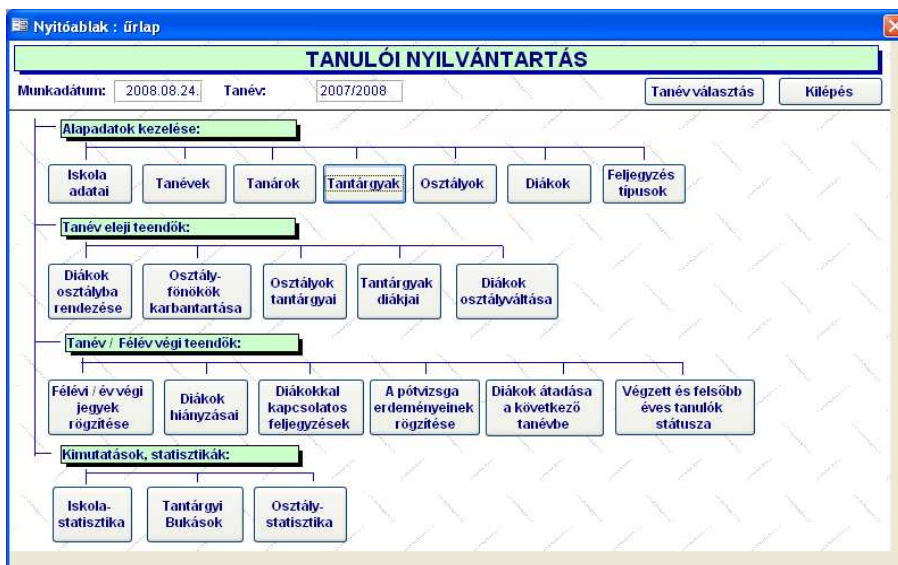
A bejelentkezés után ki a felhasználó kiválasztja azt a tanévet és munkadátumot, amivel a rendszerben akar dolgozni. A nyitott tanévekben, a tanévhez rendelt intervallumban tetszőlegesen nappal, míg lezárt tanév esetén a tanév legutolsó napjával léphet be a nyilvántartásba.



Első belépéskor a rendszer észleli az és automatikusan létrehozza a rendszer dátumnak megfelelő tanévet, amiben azután egyből be lehet lépni.

6.3. Az iskolatitkár funkciói a programba

Az iskolatitkár rendszergazdai jogokkal bír, és minden objektumhoz (adatbázis, tábla, űrlap, jelentés lekérdezések ...) teljes jogosultsággal fér hozzá. A munka környezet beállítása után, a fő ablak jelenik meg.



Minden funkció, innen érhető el a funkciónak megfelelő gombra kattintással, és a párbeszédablakok bezárása után ide is térünk vissza.

Felépítése van az adatfolyam diagram folyamatait követi le. Minden egyes menü egy folyamatot valósít meg.

6.4. Alapadatok karbantartása

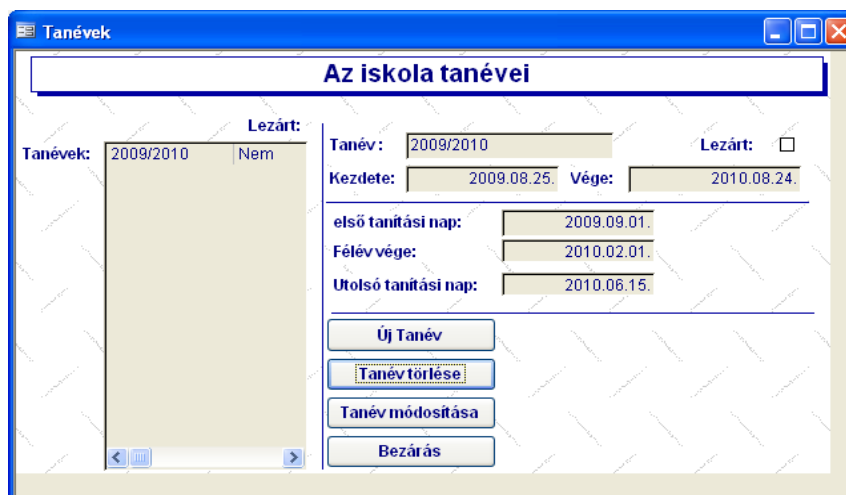
Ez a funkciók első nagy csoportja, amely az iskolai oktatás szerelőihez, és objektumaihoz kapcsolódó adatok karbantartási funkcióit gyűjti össze.

6.4.1. Iskola adatai:

Az iskola, mint intézmény, azonosító adatainak megadását és későbbi módosítását végezheti, az iskolatitkár, ebben a menüpontban. Az itt megadott adatok szerepelnek a

nyilvántartás adatiból készített félévi és év végi statisztikai adatokat tartalmazó és egyéb az oktatáshoz kapcsolódó jelentéseken. Az első bejelentkezést követően kell feltölteni adatokkal.

6.4.2. Tanévek



A tanévek, mint ahogyan a az első bejelentkezéskor is, automatikusan generálódnak, amikor a diákokat átléptetjük a következő tanévbe. Ezen későbbi pontosítására, új tanév nyitására és nem használt tanévek törlésére van lehetőség ebben a menüpontban.

6.4.3. Tanárok, Tantárgyak, Feljegyzéstípusok

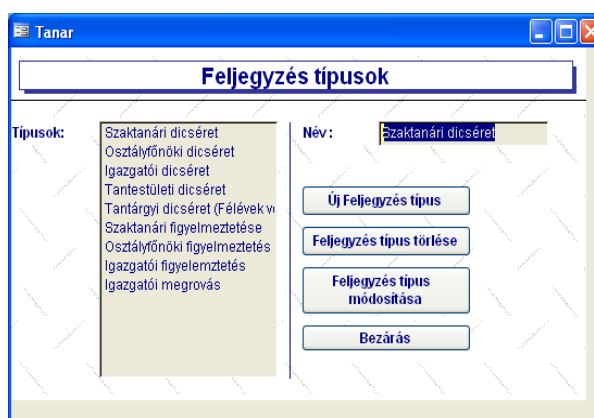
A fenti három menüpontot, hasonló szerkezetük és a terjedelem miatt együtt mutatom, be. Mindhárom menüpontra igaz, hogy jelen állapotukban, csak a legszükségesebb, információk megadását és módosítását teszik lehetővé, a fenti objektumokhoz kapcsolódóan. Ez az egyik továbbfejlesztésre váró része a nyilvántartásnak.



A menüpontban, jelen állapotában, a tanárok neveinek karbantartását végezheti az iskolatitkár. De a későbbiekben ez továbbfejleszhető, akár a tanárok teljes személyi anyagának kezelésének megvalósításával is.



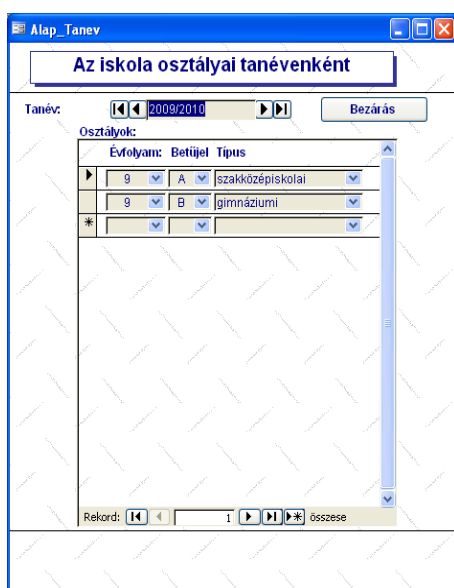
Az előzőhöz hasonlóan ez a menüpont is kiegészíthető lesz a tantárgya rövid leírásával, esetlegesen, a követelményekkel.



A tanulók tanulmányi előmenetelében nyilvántartani kívánt események, eredmények, ... stb. fajtáit adhatjuk meg itt ebben a menüpontban.

6.4.4. Osztályok

Az iskola osztályainak tanévenkénti kialakítását és későbbi kezelését a fenti menüben



végezheti a titkár. Az osztályok évfolyamának, betűjelének és típusának megadásával, a tanév menüpontban vagy automatikusan létrehozott tanévekben.

6.4.5. Diákok

The screenshot shows a window titled 'Diák' with a sub-header 'Diák legfontosabb adatai'. On the left, there is a list of student names. In the center, there are buttons for 'Új Diák', 'Diák törlése', 'Diák módosítása', and 'Bezárás'. Below this, there is a form with several fields:

<input checked="" type="checkbox"/> aktív	Diák Státusza: bejáró	TAJ: 457214386,34	Születési hely: Berettyóújfalu
<input type="checkbox"/> magántanuló	Diák Neme: Fű	OMAzonosító: 79269203838	Születési idő: 1991.12.07.
<input type="checkbox"/> évismételő		IG. szám: 375523CA	

A diákok személyes, mint születési adatok, egyedi azonosítók és egyéb adatai, mint például elérhetőségek, állandó és ideiglenes lakcímek, szülők neve és elérhetősége, ezen a menüben keresztül kezelhetők. Beiratkozáskor itt rögzíthetők az új tanulók adatai.

6.5. Tanév eleji teendők

A tanévkezdéshez kapcsolódó tevékenységek, folyamatok, kezelését biztosító menük tartoznak a második csoportba.

6.5.1. Diákok osztályba rendezése

Az újonnan beiratkozott, vagy most végzett de az iskolában szakmát tanuló diákok, - *szabad diákok* -, (bal oldali lista) osztálytagságának beállítása ebben a menüben lehetséges. A jobb oldali legördülő menükben kiválasztjuk, a tanévben induló osztályokat. A kiválasztás

The screenshot shows a window titled 'UjDiakokOsztybaRak : űrlap' with a sub-header 'Új és végzett, aktív státuszú tanulók osztályhoz rendelése'. It includes a form for 'Munkadátum' (2009.08.25) and 'Tanév' (2009/2010). Below this, there are dropdowns for 'Évfolyam' (9) and 'Bettíjet' (A). The main area contains two tables:

Új Osztályban nem sorolt diákok		
ID	Név	Születési dátum
11	Séber Sándor	1991.06.21.
12	Fodor Sándor	1994.06.23.
13	Fodor János	1992.11.01.
14	Lovas Imre	1995.10.26.
15	Papp Miklós	1996.01.25.
16	Kiss Péter	1993.06.26.
17	Szilágyi György	1993.12.01.
18	Györki Balázs	1994.06.09.
19	Molnár Tamás	1995.10.04.
20	Serdült Klaudia	1996.07.14.
21	Gyöngyösi Tamás	1995.03.20.

Az osztály diákjai			
ID	Név	Születési dátum	Születési hely
1	Bordák Szabolcs	1991.12.07.	Berettyóújfalu
2	Molnár János	1996.06.12.	Debrecen
3	Vámosi Dóra	1993.04.26.	Debrecen
4	Lakfalvi Dóra	1992.06.10.	Debrecen
5	Lengyel Éva	1996.12.04.	Debrecen
6	Jáger Tamás	1993.07.11.	Debrecen
7	Borbély Adrienn	1995.04.10.	Debrecen
8	Benedek Marcell T.	1992.12.30.	Debrecen
9	Elek Norbert	1994.02.12.	Debrecen
10	Jakabóczi Gábor	1996.05.17.	Debrecen

után megjelennek a jobb oldali listában az osztályba már korábban felvett, vagy

automatikusan átvett tanulók, A két gombbal értelemszerűen behelyezhetünk új illetve törölhetünk felvett tanulókat.

6.5.2. Osztályfőnökök karbantartása



A tanévben kialakított osztályok osztályfőnökeit, illetve osztályfőnök váltásai a fenti párbeszédablakban adhatjuk meg. Új osztályfőnök esetén kezdés dátuma a munkadátum, amelyet a bejelentkezés után adtunk meg. Osztályfőnökváltáskor szintén a munkadátum lesz az előző osztályfőnök tevékenységi idejének vége is.

6.5.3. Osztályok tantárgyai

Az osztályban tanított tantárgyak és a tantárgyat tanító tanárok megadása ebben a menüpontban tehető meg. A megadáskor a fentiek mellett meg kell adni azt is, hogy az adott tárgyból szerzett jegy az átlagba számít-e, (pl.: Magatartás, szorgalom) Illetve, hogy a



tantárgy összevont-e (pl.: Az elektronika alapjai c. tárgy, ami a szakmacsoportos alapozó oktatás az elektronika- elektrotechnika szakmacsoportra c. tárgy részét képezi, a diák jegyet kap belőle, meg is bukhat, de az átlagba csak a tartalmazó tárgy jegye számít.) A csoportbontásban tanított tárgyak (pl.: általában idegen nyelvi órák) megadásának első lépés is itt lehetséges oly módon, hogy a tantárgyat kétszer minkét tanárral felvesszük az osztályhoz.

6.5.4. Tantárgyak diákjai

Itt rendeli hozzá az iskolatitkár az osztály diákjait, az osztály tantárgyaihoz. Itt fejeződik be a csoportbontások kezelése is, ugyanis az osztály kiválasztása (évfolyam + betűjel) után a tantárgyak legördülő listában a tantárgy a tanító tanárral együtt jelenik meg, így csak azokat a diákokat szükséges berakni a tantárgycsoportot tanuló diákok közé, akik tényelegetesen az adott tanárnál tanulják a tárgyat. A jobb oldali listából lehetőség van egy, több vagy minden tanuló tanulócsoportba helyezésére és az onnan történő törlésére is. Valamint a második „Diákok felmentései” c. lapon meg lehet adni tantárgyanként, hogy mely tanulók felmentettek az adott tárgyból a csoport tanulói közül. Ez azért fontos mert a későbbi félévi és év végi jegybeíráskor a felmentéssel rendelkező diákoknak nem történik jegybeírás.

Osztály diákjainak tantárgyai és felmentései

Munkadátum: 2009.08.25. Tanév: 2009/2010

Évfolyam: 9
Betűjel: A
Tantárgy: Irodalom - Gilányiné Gyarmati Júlia Ol.

Diákok Tantárgyai | Diákok Felmentései

A Tantárgy tanuló diákok		
ID	Név	Felmentett
1	Bordák Szabolcs	0
2	Molnár János	0
3	Vámosi Dóra	0
4	Lakfalvi Dóra	0
5	Lengyel Éva	0
6	Jäger Tamás	0
7	Borbély Adrienn	0
8	Benedek Marcell T.	0
9	Elek Norbert	0
10	Jakabóczi Gábor	0

Az osztály diákjai			
ID	Név	Szüldo	SzüHely
1	Bordák Szabolcs	1991.12.07.	Beregyórfalu
2	Molnár János	1999.06.12.	Debrecen
3	Vámosi Dóra	1993.04.26.	Debrecen
4	Lakfalvi Dóra	1992.06.10.	Debrecen
5	Lengyel Éva	1996.12.04.	Debrecen
6	Jäger Tamás	1993.07.11.	Debrecen
7	Borbély Adrienn	1995.04.10.	Debrecen
8	Benedek Marcell T.	1992.12.30.	Debrecen
9	Elek Norbert	1994.02.12.	Debrecen
10	Jakabóczi Gábor	1996.05.17.	Debrecen

Osztály diákjainak tantárgyai és felmentései

Munkadátum: 2009.08.25. Tanév: 2009/2010

Évfolyam: 9
Betűjel: A
Tantárgy: Irodalom - Gilányiné Gyarmati Júlia Ol.

Diákok Tantárgyai | Diákok Felmentései

Felmentések:	
Név	Felmentett
Bordák Szabolcs	<input type="checkbox"/>
Molnár János	<input type="checkbox"/>
Vámosi Dóra	<input type="checkbox"/>
Lakfalvi Dóra	<input type="checkbox"/>
Lengyel Éva	<input type="checkbox"/>
Jäger Tamás	<input type="checkbox"/>
Borbély Adrienn	<input type="checkbox"/>
Benedek Marcell T.	<input type="checkbox"/>
Elek Norbert	<input type="checkbox"/>

6.5.5. Diákok osztályváltása

Diákok osztályváltása : űrlap

Diákok osztályváltása

Munkadátum: 2009.08.25 Tanév: 2009/2010

Évfolyam: 9 Betűjel: A

Évfolyam: 9 Betűjel: B

Az osztály diákjai

IDDial	Nev	Szulido
1	Bordák Szabolcs	1991.12.07.
2	Molnár János	1996.06.12.
3	Vámosi Dóra	1993.04.26.
4	Lakfalvi Dóra	1992.06.10.
5	Lengyel Éva	1996.12.04.
6	Jáger Tamás	1993.07.11.
7	Borbély Adrienn	1995.04.10.
8	Benedek Marcell T.	1992.12.30.
9	Elek Norbert	1994.02.12.
10	Jakabóczy Gábor	1996.05.17.

Egyik --->

< --- Másik

Az osztály diákjai

IDDial	Nev	Szulido	SzulHely
11	Séber Sándor	1991.06.21.	Debrecen
12	Fodor Sándor	1994.06.23.	Debrecen
13	Fodor János	1992.11.01.	Debrecen
14	Lovas Imre	1995.10.26.	Derecske
15	Papp Miklós	1996.01.25.	Berettyóújfalú
16	Kiss Péter	1993.06.26.	Berettyóújfalú
17	Szilágyi György	1993.12.01.	Berettyóújfalú
18	Györki Balázs	1994.06.09.	Debrecen
19	Molnár Tamás	1995.10.04.	Debrecen
20	Serdült Klaudia	1996.07.14.	Debrecen

A diákok év közbeni osztályváltását az alábbi dialógus ablakon keresztül lehet végrehajtani a forrás és a cél osztályok kiválasztása után, az osztály diákjai megjelennek a két listában és az osztályok között szabályozott módon válthatnak osztályt. A listákban a többszörös kijelölés lehetősége biztosított.

6.6. Tanév / Félév végi teendők

A félév és a tanév befejezéséhez kapcsolódó tevékenységeket, folyamatokat, megvalósító funkciók tartoznak a harmadik nagy csoportba.

6.6.1. Félévi /Évvégi jegyek rögzítése

A tanulók jegyeit rögzíthetjük a tanév osztályaiban, félévente, tanulónként,. Az osztály és a jegytípus kiválasztása után a a jegybeírás gombra klikkelve az osztály összes tanulójának, a korábban beállított tárgyakból, jegy írható be. Ezt a műveletet osztályonként meg kell ismételni, az osztályok között a léptető gombok segítségével navigálhatunk.. A jegybeíráshoz ki kell választani, tanuló neve alatt a bevinni kívánt jegyek, típusát, majd a jegybeírás tanulónként elvégezhető

A jegybeírást követően az ablak bezárása után visszajutunk az alkalmazás fő ablakába, ahol a következő lépésben a tanulók hiányzásait rögzíthetjük.

Diákok félévi - év végi jegyei

Munkadátum: 2007.08.25. Tanév: 2007/2008

Osztály: 9.A

Név: Benedek Marcell T.

Jegybeírás Típus: félévi

Tantárgy:	Jegy:	Jegy Típus:
Magatartás	2	FE
Szorgalom	2	FE
Irodalom	4	FE
Magyar nyelv	2	FE
Angol nyelv	2	FE
Matematika	5	FE
Fizika	5	FE
Testnevelés és sport	5	FE

Rekord: 1 összesen 8

6.6.2. Diákok hiányzásai

A diákok hiányzásainak rögzítése mindig a munkadátummal történik és a statisztikák készítése előtt a program göngyöli az eddigi hiányzásokat. A hiányzásokat osztályonként rögzíthetjük, mindig a kiválasztott osztály hiányzásait rögzíthetjük az ablakban található hiányzások rögzítése felíratú gomb segítségével.

Diákok félévi / év végi hiányzásai

Munkadátum: 2007.08.25. Tanév: 2007/2008

Osztály: 9.A

Hiányzások rögzítése

Név:	Igazolt		Igazolatlan		Dátum:
	elmélet	gyakorlat	elmélet	gyakorlat	
Teszt Elek	67	67	76	67	2008.01.15.
Jakabóczy Gábor	36	151	124	26	2008.01.15.
Elek Norbert	51	96	138	129	2008.01.15.
Benedek Marcell T.	47	38	60	59	2008.01.15.
Borbély Adrienn	9	129	85	105	2008.01.15.
Jáger Tamás	65	53	39	29	2008.01.15.
Lengyel Éva	126	72	144	14	2008.01.15.
Lakfalvi Dóra	15	122	12	39	2008.01.15.
Vámosi Dóra	24	57	102	133	2008.01.15.
Molnár János	149	63	35	107	2008.01.15.
Bordák Szabolcs	65	56	27	115	2008.01.15.

Rekord: 1 összesen 10

6.6.3. Diákokkal kapcsolatos feljegyzések

A menüpont megnevezésének megfelelően a diákokhoz kapcsolódó megjegyzések rögzíthetők, a feljegyzés típus, név és dátum megadása után, rövid leírással egészíthetjük ki a feljegyzést. A következő feljegyzés rögzítéséhez, a bal alsó részén lévő navigátor gombok segítségével nyílik lehetőség.

Rekord: 1 összesen 1

6.6.4. Pótvizsga eredmények rögzítése

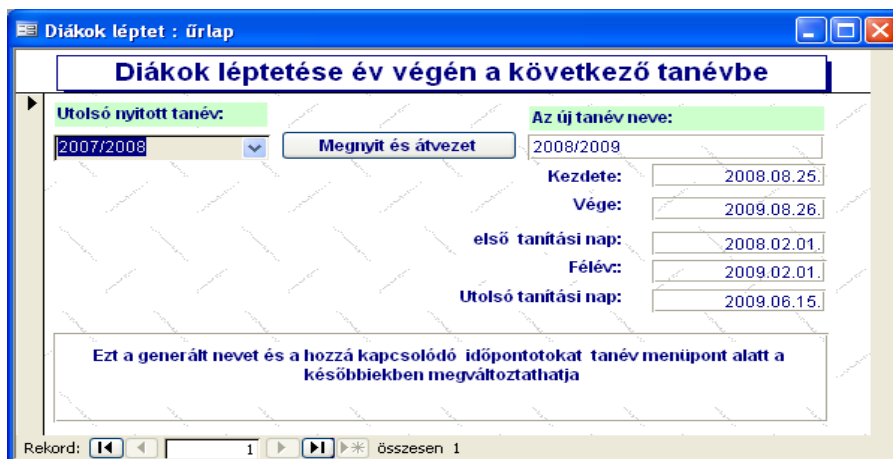
Az tanév végén bukott tanulók pótvizsgán elért eredményeinek rögzítése, az ablak jobb felső részén látható gomb használata után lehetséges. A funkció aktiválása után a tanév végén elkészül a fenti osztályozó ív, melybe rögzíthető a pótvizsgák eredményei.

Név	Osztály	Tantárgy	Tanár	Típus	Jegy
Molnár János	9.A	Magatartás	Alföldi Ferenc	JV	3
Fodor János	9.B	Magatartás	Kőrösi Sándor	JV	3
Orbázi Gyula	10.A	Magatartás	Komódi István	JV	3
Fedics Lilla	10.A	Magatartás	Komódi István	JV	3
Szabó Zsolt	10.B	Magatartás	Bujdosó Sándor	JV	3
Badár Ferenc	10.B	Magatartás	Bujdosó Sándor	JV	3
Olajos Zsolt	11.B	Magatartás	Bálintné Bohács Anikó	JV	3
Szűcs Bettina	11.B	Magatartás	Bálintné Bohács Anikó	JV	3
Szabó László M.	12.A	Magatartás	Beke Béla	JV	3
Békés H. István G.	12.A	Magatartás	Beke Béla	JV	3
Budaházy Csaba	12.A	Magatartás	Beke Béla	JV	2
Mészáros Norbert	12.B	Magatartás	Biró Attila	JV	2
Csulák Andrea	12.B	Magatartás	Biró Attila	JV	2
Máté Tibor	12.B	Magatartás	Biró Attila	JV	2
Veszprémi Tibor	13.A	Magatartás	László Tibor	JV	2

Rekord: 1 összesen 134

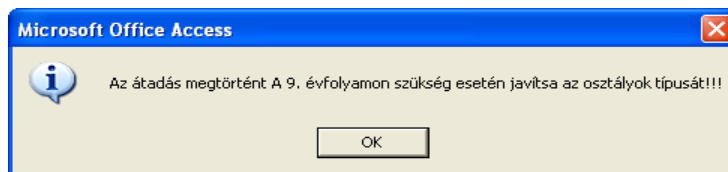
6.6.5. Diákok átadása a következő tanévbe

A párbeszédablakban a nyitott tanévet kiválasztva, megjelennek az új tanév adatai. A megnyit és átvezet feliratú gomb segítségével elindítható a 9-11. évfolyamos osztályok



tanulók, osztályfőnökök átléptetése az új tanévbe. A folyamat lezárja az eddigi osztálytagságokat, az osztályfőnök működési idejét az aktuális tanévben.

Az átadás átvétel sikerességéről a fenti párbeszédablak tájékoztat.



6.6.6. Végzett és felsőbb éves tanulók státuszának beállítása

Név:	Évfolyam:	Betűjel:	Születési idő:	Születési hely:	Befejezés:	Aktiv:
László István	12	B	1993.03.23.	Debrecen	2008.08.24.	☑
Mészáros Norbert	12	B	1995.06.24.	Debrecen	2008.08.24.	☑
Dobrán Alexandra	12	B	1995.11.01.	Debrecen	2008.08.24.	☑
Csulák Andrea	12	B	1995.03.17.	Debrecen	2008.08.24.	☑
Lovas Gergely	12	B	1996.07.11.	Debrecen	2008.08.24.	☑
Máté Tibor	12	B	1992.09.20.	Debrecen	2008.08.24.	☑
Tóth Gábor	12	B	1996.05.25.	Derecske	2008.08.24.	☑
Horváth Ferenc	12	B	1991.04.03.	Berettyóújfalu	2008.08.24.	☑
Hutóczki Mihály	12	B	1991.03.13.	Berettyóújfalu	2008.08.24.	☑
Goda Zsuzsanna	13	B	1991.08.07.	Berettyóújfalu	2008.08.24.	☑
Tikász Áron	13	B	1991.02.24.	Debrecen	2008.08.24.	☑
Kiss Krisztina	13	B	1991.07.14.	Debrecen	2008.08.24.	☑

Mivel a 12. évfolyamtól felfelé a diákok változatos képzési formákat választhatnak, mint továbbtanulás felsőfokú intézményekbe, szakmaválasztása az iskolában, ... stb. Ezért

ezek az osztályok tanulói nem lépnek automatikusan a következő tanévben, hanem a státuszuk tanulónkénti állítása után, a továbbra is aktív státuszú diákokat a diákok osztályba rendezése menüpontban rendelhetjük az esetleges új osztályokhoz,

6.7. Kimutatások, Statisztikák

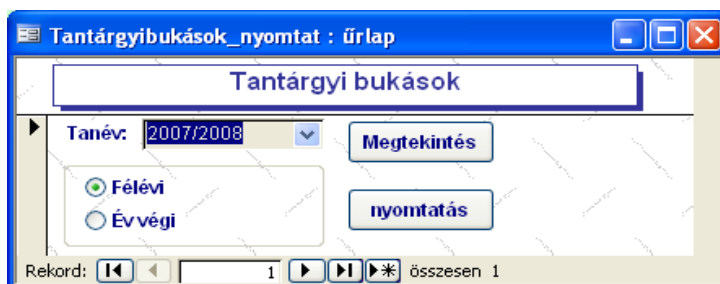
Az utolsó csoportba az iskola jelentési kötelezettségeit segítő többnyire automatikusan elkészülő statisztikák kimutatások készítését megvalósító funkciók tartoznak

6.7.1. Iskolastatisztika



A tanév és a készítési időszak megadása után automatikusan elkészíthető a kötetelményekben megfogalmazott adatokat és az ezekből nyer információkat tartalmazó iskolastatisztika.

6.8. Tantárgyi Bukások



A tanév és a készítési időszak megadása után automatikusan elkészíthető a tantárgyak Bukási statisztikája.

7. Összefoglalás

A tanulmányi rendszer rendszertervének elkészítésére a munkahelyem mindennapjai inspiráltak, és a munka elején nem tűnt nagy feladatnak. Az általam ismertnek vélt folyamatok és az iskola adatainak mélyebb elemzése során, egy nagy és igen bonyolult rendszer bontakozott ki számtalan egyeddel és összetett kapcsolatokkal, mint például a Osztály-Tanár-Tantárgy-Tanuló-Jegy kapcsolat. A rendszer méreteiből fakadóan, az alkalmazás készítésekor csak a nyilvántartó rendszer egy részletének megvalósítását tűztem ki célul. Ez pedig iskolastatisztika automatikus létrehozásának megvalósítása volt.

A feladat megvalósítása során, új kihívás volt számomra az Access Visual Basicben való programozása, mivel a nyelvet a dolgozat megírása előtt nem használtam.

Az elkészült alkalmazás az iskolatitkárok és az iskola mindennapi munkáját segíti. Biztosítva a gyors és pontos adatszolgáltatást a fenntartó és egyéb szervek, intézmények irányába.

8. Irodalomjegyzék

- **Jeffrey D. Ullman – Jennifer Widom: Adatbázisrendszerek Alapvetés**
Panem, 1998
- **Stolniczki Gyula : SQL kézikönyv**
ComputerBooks, 1995
- **Bana István: Az SSADM rendszertervezési módszertan**
LSI Oktatóközpont, 2000
- **Czenky Márta: ADATMODELLEZÉS * SQL ÉS ACCES ALKLAMAZÁS**
ComputerBooks, 1995
- **Denise Gosnell: Beginning Access 2003 VBA**
Wiley Publishing Inc., 2004
- **Patricia Cardoza, Teresa Henning, Graham Seach, Armen Stein: Acces 2003 VBA**
Programmer's Reference
Wiley Publishing Inc., 2004
- **Peter G. Aitken: Programozás Visual Basic 6 nyelven Kék könyv**
Kiskapu, 1999
- **Sályi János -Szelezsán János: ADATBÁZISOK**
SZÁMALK Kiadó, 1991

9. Mellékletek

9.1. -1.- számú melléklet – Az adatbázis szerkezetét kialakító kód

```
' Created          2009.05.29.
' Modified         2009.05.29.
' Project
' Model           Tanulói nyilvántartás
' Company
' Author          Beke Béla
' Version
' Database        Access 2000

Public dbs As DAO.Database
Public tdf As DAO.TableDef
Public idx As DAO.Index
Public rel As DAO.Relation

Sub Main()

Set dbs = CurrentDb()

On Error GoTo ErrorHandler

    Call DropTables
    Call DropIndexes
    Call CreateTables

    Call CreatePrimaryKeys
    Call CreateIndexes

    Call CreateRelations

    MsgBox "Script successfully processed.", vbInformation
    Exit Sub

ErrorHandler:
    Select Case Err.Number
        Case 3010
            MsgBox "Table " & tdf.Name & " already exist!", vbInformation
            Err.Clear
        Case 3284
            MsgBox "Index " & idx.Name & " for table " & tdf.Name & " already exist!",
vbInformation
            Err.Clear
        Case Else
            MsgBox Err.Description, vbCritical
    End Select
End Sub

' Drop indexes
'=====

Sub DropIndexes()

'Drop all non unique indexes

End Sub

' Drop tables
'=====

Public Sub DropTables()
```

```

'Drop relations
Call DropRelation("TelepulesDiákokCimei")
Call DropRelation("TantCsopDiakjaiTantCsopDiakJegy")
Call DropRelation("TantargyCsoportTantCsopDiakJegy")
Call DropRelation("TantargyTantargyCsoport")
Call DropRelation("TanevOsztyaly")
Call DropRelation("TanarOsztyalyfonokok")
Call DropRelation("TanarTantargyCsoport")
Call DropRelation("OsztyalyOsztyalyfonokok")
Call DropRelation("OsztyalyOsztyDiak")
Call DropRelation("OsztyalyTantargyCsoport")
Call DropRelation("OrszagTelepules")
Call DropRelation("A_feljegyzes_tipusa")
Call DropRelation("DiakDiákokCimei")
Call DropRelation("DiakFeljegyzesek")
Call DropRelation("DiakHianyzasok")
Call DropRelation("DiakOsztyDiak")
Call DropRelation("DiakSzulok")
Call DropRelation("Diakok_Jegy")
Call DropRelation("DiakTelefonszamok")

```

```

'Drop tables
Call DropTable("Telepules")
Call DropTable("Telefonszamok")
Call DropTable("TantCsopDiakJegy")
Call DropTable("TantCsopDiakjai")
Call DropTable("TantargyCsoport")
Call DropTable("Tantargy")
Call DropTable("Tanev")
Call DropTable("Tanar")
Call DropTable("Szulok")
Call DropTable("OsztyDiak")
Call DropTable("Osztyalyfonokok")
Call DropTable("Osztyaly")
Call DropTable("Orszag")
Call DropTable("Hianyzasok")
Call DropTable("Feljegyzestipus")
Call DropTable("Feljegyzesek")
Call DropTable("DiákokCime")
Call DropTable("Diak")

```

End Sub

```

' Create tables
'=====

```

```

Sub CreateTablees()

```

```

Call CreateTable1 'Diak
Call CreateTable2 'DiákokCime
Call CreateTable3 'Feljegyzesek
Call CreateTable4 'Feljegyzestipus
Call CreateTable5 'Hianyzasok
Call CreateTable6 'Orszag
Call CreateTable7 'Osztyaly
Call CreateTable8 'Osztyalyfonokok
Call CreateTable9 'OsztyDiak
Call CreateTable11 'Szulok
Call CreateTable12 'Tanar
Call CreateTable13 'Tanev
Call CreateTable14 'Tantargy
Call CreateTable15 'TantargyCsoport
Call CreateTable16 'TantCsopDiakjai
Call CreateTable17 'TantCsopDiakJegy
Call CreateTable18 'Telefonszamok
Call CreateTable19 'Telepules
End Sub

```

```

'=== Create table Diak =====

```

```

Sub CreateTable1()

```

```

Set tdf = dbs.CreateTableDef( "Diak" )

Call AddFieldToTable("IDDiak", dbLong, 0, dbAutoIncrField, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Nev", dbText, 50, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("aktiv", dbBoolean, 0, 0, "Yes", "", "", TRUE, FALSE )
Call AddFieldToTable("Magantanulo", dbBoolean, 0, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Evismetlo", dbBoolean, 0, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Statusz", dbText, 10, 0, ""debreceni"", "", "", TRUE, FALSE )
Call AddFieldToTable("Neme", dbText, 5, 0, ""Fiú"", "", "", TRUE, FALSE )
Call AddFieldToTable("TAJ", dbText, 12, 0, "", "", "", FALSE, TRUE )
Call AddFieldToTable("DOMAzon", dbText, 11, 0, "", "", "", FALSE, TRUE )
Call AddFieldToTable("SzigSzám", dbText, 10, 0, "", "", "", FALSE, TRUE )
Call AddFieldToTable("SzulHely", dbText, 30, 0, "", "", "", FALSE, TRUE )
Call AddFieldToTable("Szulido", dbDate, 0, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

Call AddPropertyToField( "Statusz","Caption","Diák Státusza",dbText)
Call AddPropertyToField( "Neme","Caption","Diák Neme",dbText)
Call AddPropertyToField( "TAJ","InputMask","000\~000\~000;0;_" ,dbText)
Call AddPropertyToField( "TAJ","UnicodeCompression",true,dbBoolean)
Call AddPropertyToField( "DOMAzon","InputMask","0000000000" ,dbText)
Call AddPropertyToField( "DOMAzon","UnicodeCompression",true,dbBoolean)
Call AddPropertyToField( "SzigSzám","InputMask","aa\ 00000;0;_" ,dbText)
Call AddPropertyToField( "SzigSzám","UnicodeCompression",true,dbBoolean)
Call AddPropertyToField( "SzulHely","UnicodeCompression",true,dbBoolean)
Call AddPropertyToField( "Szulido","InputMask","0000/00/00/;0;_" ,dbText)

End Sub

'=== Create table DiákokCimeei =====

Sub CreateTable2()

Set tdf = dbs.CreateTableDef( "DiákokCimeei" )

Call AddFieldToTable("Tipusa", dbText, 12, 0, ""állandó"", "", "", FALSE, FALSE )
Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IRSZ", dbText, 10, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDOrszag", dbText, 5, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Kozterulet_neve", dbText, 50, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("KozteruletTip", dbText, 5, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Hazszam", dbText, 5, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("emszam", dbInteger, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("ajtoszam", dbInteger, 0, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Feljegyzesek =====

Sub CreateTable3()

Set tdf = dbs.CreateTableDef( "Feljegyzesek" )

Call AddFieldToTable("IDFeltip", dbLong, 0, 0, "1", "", "", TRUE, FALSE )
Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Datum", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Inform", dbMemo, 0, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Feljegyzestipus =====

Sub CreateTable4()

Set tdf = dbs.CreateTableDef( "Feljegyzestipus" )

```

```

Call AddFieldToTable("IDFeltip", dbLong, 0, dbAutoIncrField, "", "", "", FALSE, FALSE )
Call AddFieldToTable("FeljTipusNev", dbText, 255, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

Call AddPropertyToTable( "Description", "Jutalmak"+Chr(13)+Chr(10) _
+ "Buntetesek"+Chr(13)+Chr(10) _
+ "egyéb", dbText)

End Sub

'=== Create table Hianyzasok =====

Sub CreateTable5()

Set tdf = dbs.CreateTableDef( "Hianyzasok" )

Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Datum", dbText, 50, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("elmelet_ig", dbInteger, 0, 0, "0", "", "", FALSE, FALSE )
Call AddFieldToTable("elmelet_igtln", dbInteger, 0, 0, "0", "", "", FALSE, FALSE )
Call AddFieldToTable("gyakorlat_ig", dbInteger, 0, 0, "0", "", "", FALSE, FALSE )
Call AddFieldToTable("gyakorlat_igtln", dbInteger, 0, 0, "0", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Orszag =====

Sub CreateTable6()

Set tdf = dbs.CreateTableDef( "Orszag" )

Call AddFieldToTable("IDOrszag", dbText, 5, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("OrszagNev", dbText, 50, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Osztaly =====

Sub CreateTable7()

Set tdf = dbs.CreateTableDef( "Osztaly" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Evfolyam", dbInteger, 0, 0, "", "", "", "Between 9 And 15", FALSE, FALSE )
Call AddFieldToTable("Betujel", dbText, 1, 0, "", "", "", "Between "A" And "J"", FALSE,
FALSE )
Call AddFieldToTable("tipus", dbText, 50, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Osztalyfonokok =====

Sub CreateTable8()

Set tdf = dbs.CreateTableDef( "Osztalyfonokok" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Evfolyam", dbInteger, 0, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Betujel", dbText, 1, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Kezdet", dbDate, 0, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("IDTanar", dbLong, 0, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Vege", dbDate, 0, 0, "", "", "", FALSE, FALSE )

```

```

dbs.TableDefs.Append tdf

Call AddPropertyToField( "Kezdete","Format","Short Date",dbText)
Call AddPropertyToField( "Kezdete","InputMask","0000/00/00/;0;_",dbText)
Call AddPropertyToField( "Vege","InputMask","0000/00/00/;0;_",dbText)

End Sub

'=== Create table Osztdiak =====

Sub CreateTable9()

Set tdf = dbs.CreateTableDef( "Osztdiak" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Evfolyam", dbInteger, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Betujel", dbText, 1, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Kezdete", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Vege", dbDate, 0, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Szulok =====

Sub CreateTable11()

Set tdf = dbs.CreateTableDef( "Szulok" )

Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Szulo_Neve", dbText, 50, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Statusza", dbText, 25, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Tanar =====

Sub CreateTable12()

Set tdf = dbs.CreateTableDef( "Tanar" )

Call AddFieldToTable("IDTanar", dbLong, 0, dbAutoIncrField, "", "", "", FALSE, FALSE )
Call AddFieldToTable("TanarNev", dbText, 50, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table Tanev =====

Sub CreateTable13()

Set tdf = dbs.CreateTableDef( "Tanev" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Kezdete", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Vege", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("OktKezd", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("OktFelev", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("OktEvVege", dbDate, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Lezart", dbBoolean, 0, 0, "No", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

Call AddPropertyToField( "Kezdete","InputMask","0000/00/00/;0;_",dbText)
Call AddPropertyToField( "Vege","InputMask","0000/00/00/;0;_",dbText)

```

```

Call AddPropertyToField( "OktKezd", "InputMask", "0000/00/00/;0;_", dbText)
Call AddPropertyToField( "OktFelev", "InputMask", "0000/00/00/;0;_", dbText)
Call AddPropertyToField( "OktEvVege", "InputMask", "0000/00/00/;0;_", dbText)
Call AddPropertyToField( "Lezart", "Format", "Yes/No", dbText)

End Sub

'=== Create table Tantargy =====

Sub CreateTable14()

Set tdf = dbs.CreateTableDef( "Tantargy" )

Call AddFieldToTable("IDTantargy", dbLong, 0, dbAutoIncrField, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Tantargynev", dbText, 50, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table TantargyCsoport =====

Sub CreateTable15()

Set tdf = dbs.CreateTableDef( "TantargyCsoport" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("Evfolyam", dbInteger, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Betujel", dbText, 1, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDTantargy", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDTanar", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("atlagba", dbBoolean, 0, 0, "Yes", "", "", FALSE, FALSE )
Call AddFieldToTable("osszevont", dbBoolean, 0, 0, "No", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

Call AddPropertyToField( "atlagba", "Format", "Yes/No", dbText)
Call AddPropertyToField( "atlagba", "Description", "átlagszámításnál figyelembe kell-e venni a jegyeket", dbText)
Call AddPropertyToField( "osszevont", "Format", "Yes/No", dbText)
Call AddPropertyToField( "osszevont", "Description", "több résztantárgyból tevődik-e össze", dbText)

End Sub

'=== Create table TantCsopDiakjai =====

Sub CreateTable16()

Set tdf = dbs.CreateTableDef( "TantCsopDiakjai" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )
Call AddFieldToTable("IDTantargy", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Evfolyam", dbInteger, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Betujel", dbText, 1, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDTanar", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Felmentett", dbBoolean, 0, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

'=== Create table TantcsopDiakJegy =====

Sub CreateTable17()

Set tdf = dbs.CreateTableDef( "TantcsopDiakJegy" )

Call AddFieldToTable("TanevNev", dbText, 10, 0, "", "", "", TRUE, FALSE )

```

```

Call AddFieldToTable("IDTantargy", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Evfolyam", dbInteger, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Betujel", dbText, 1, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDTanar", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("JegyTip", dbText, 2, 0, "", "", "", FALSE, TRUE )
Call AddFieldToTable("Jegy", dbByte, 0, 0, "", "Az érdemjegy 1 és 5 közötti egész szám lehet csak", ">=1 And <=5", FALSE, FALSE )

dbs.TableDefs.Append tdf

Call AddPropertyToField( "JegyTip","UnicodeCompression",true,dbBoolean)

End Sub

'=== Create table Telefonszamok =====

Sub CreateTable18()

Set tdf = dbs.CreateTableDef( "Telefonszamok" )

Call AddFieldToTable("IDDiak", dbLong, 0, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Tipus", dbText, 7, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("Telszám", dbText, 20, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

Call AddPropertyToField( "Tipus","Description",""+Chr(13)+Chr(10) _
+ "Anya - Anya Száma"+Chr(13)+Chr(10) _
+ "Apa - Apa száma"+Chr(13)+Chr(10) _
+ "",dbText)
Call AddPropertyToField( "Telszám","InputMask","\ (90"") ""000\ -0009;0;_",dbText)

End Sub

'=== Create table Telepules =====

Sub CreateTable19()

Set tdf = dbs.CreateTableDef( "Telepules" )

Call AddFieldToTable("IRSZ", dbText, 10, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("IDOrszag", dbText, 5, 0, "", "", "", FALSE, FALSE )
Call AddFieldToTable("TelepNev", dbText, 100, 0, "", "", "", FALSE, FALSE )

dbs.TableDefs.Append tdf

End Sub

' Create primary keys
'=====

Sub CreatePrimaryKeys()

'=== Create primary key for table Diak =====

Set tdf = dbs.TableDefs( "Diak" )
Set idx = tdf.CreateIndex( "pk_Diak" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDDiak", False )

tdf.Indexes.Append idx

'=== Create primary key for table DiákokCimeei =====

Set tdf = dbs.TableDefs( "DiákokCimeei" )

```

```

Set idx = tdf.CreateIndex( "PrimaryKey" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "Tipusa", False )
Call AddFieldToIndex( "IDDiak", False )
Call AddFieldToIndex( "IRSZ", False )
Call AddFieldToIndex( "IDOrszag", False )

tdf.Indexes.Append idx

'=== Create primary key for table Feljegyzesek =====

Set tdf = dbs.TableDefs( "Feljegyzesek" )
Set idx = tdf.CreateIndex( "PrimaryKey" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDFeltip", False )
Call AddFieldToIndex( "IDDiak", False )

tdf.Indexes.Append idx

'=== Create primary key for table Feljegyzestipus =====

Set tdf = dbs.TableDefs( "Feljegyzestipus" )
Set idx = tdf.CreateIndex( "pk_Feljegyzestipus" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDFeltip", False )

tdf.Indexes.Append idx

'=== Create primary key for table Hianyzasok =====

Set tdf = dbs.TableDefs( "Hianyzasok" )
Set idx = tdf.CreateIndex( "PrimaryKey" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDDiak", False )
Call AddFieldToIndex( "Datum", False )

tdf.Indexes.Append idx

'=== Create primary key for table Orszag =====

Set tdf = dbs.TableDefs( "Orszag" )
Set idx = tdf.CreateIndex( "pk_Orszag" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDOrszag", False )

tdf.Indexes.Append idx

'=== Create primary key for table Osztaly =====

Set tdf = dbs.TableDefs( "Osztaly" )
Set idx = tdf.CreateIndex( "pk_Osztaly" )

idx.Primary = True
idx.Unique = True

```

```

idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )
Call AddFieldToIndex( "Evfolyam", False )
Call AddFieldToIndex( "Betujel", False )

tdf.Indexes.Append idx

'=== Create primary key for table Osztalyfonokok =====

Set tdf = dbs.TableDefs( "Osztalyfonokok" )
Set idx = tdf.CreateIndex( "PrimaryKey" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )
Call AddFieldToIndex( "Evfolyam", False )
Call AddFieldToIndex( "Betujel", False )
Call AddFieldToIndex( "Kezdete", False )

tdf.Indexes.Append idx

'=== Create primary key for table OsztDiak =====

Set tdf = dbs.TableDefs( "OsztDiak" )
Set idx = tdf.CreateIndex( "PrimaryKey" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )
Call AddFieldToIndex( "IDDiak", False )
Call AddFieldToIndex( "Evfolyam", False )
Call AddFieldToIndex( "Betujel", False )
Call AddFieldToIndex( "Kezdete", False )

tdf.Indexes.Append idx

'=== Create primary key for table Szulok =====

Set tdf = dbs.TableDefs( "Szulok" )
Set idx = tdf.CreateIndex( "pk_Szulok" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDDiak", False )
Call AddFieldToIndex( "Szulo_Neve", False )

tdf.Indexes.Append idx

'=== Create primary key for table Tanar =====

Set tdf = dbs.TableDefs( "Tanar" )
Set idx = tdf.CreateIndex( "pk_Tanar" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDTanar", False )

tdf.Indexes.Append idx

'=== Create primary key for table Tanev =====

Set tdf = dbs.TableDefs( "Tanev" )
Set idx = tdf.CreateIndex( "pk_Tanev" )

idx.Primary = True

```

```

idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )

tdf.Indexes.Append idx

'=== Create primary key for table Tantargy =====

Set tdf = dbs.TableDefs( "Tantargy" )
Set idx = tdf.CreateIndex( "pk_Tantargy" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDTantargy", False )

tdf.Indexes.Append idx

'=== Create primary key for table TantargyCsoport =====

Set tdf = dbs.TableDefs( "TantargyCsoport" )
Set idx = tdf.CreateIndex( "pk_TantargyCsoport" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )
Call AddFieldToIndex( "Evfolyam", False )
Call AddFieldToIndex( "Betujel", False )
Call AddFieldToIndex( "IDTantargy", False )
Call AddFieldToIndex( "IDTanar", False )

tdf.Indexes.Append idx

'=== Create primary key for table TantCsopDiakjai =====

Set tdf = dbs.TableDefs( "TantCsopDiakjai" )
Set idx = tdf.CreateIndex( "pk_TantCsopDiakJegy" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )
Call AddFieldToIndex( "IDTantargy", False )
Call AddFieldToIndex( "Evfolyam", False )
Call AddFieldToIndex( "Betujel", False )
Call AddFieldToIndex( "IDTanar", False )
Call AddFieldToIndex( "IDDiak", False )

tdf.Indexes.Append idx

'=== Create primary key for table TantcsopDiakJegy =====

Set tdf = dbs.TableDefs( "TantcsopDiakJegy" )
Set idx = tdf.CreateIndex( "PrimaryKey" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "TanevNev", False )
Call AddFieldToIndex( "IDTantargy", False )
Call AddFieldToIndex( "Evfolyam", False )
Call AddFieldToIndex( "Betujel", False )
Call AddFieldToIndex( "IDTanar", False )
Call AddFieldToIndex( "IDDiak", False )
Call AddFieldToIndex( "JegyTip", False )

tdf.Indexes.Append idx

```

```

'=== Create primary key for table Telefonszamok =====
Set tdf = dbs.TableDefs( "Telefonszamok" )
Set idx = tdf.CreateIndex( "pk_Telefonszamok" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IDDiak", False )
Call AddFieldToIndex( "Tipus", False )

tdf.Indexes.Append idx

'=== Create primary key for table Telepules =====
Set tdf = dbs.TableDefs( "Telepules" )
Set idx = tdf.CreateIndex( "pk_Telepules" )

idx.Primary = True
idx.Unique = True
idx.IgnoreNulls = False

Call AddFieldToIndex( "IRSZ", False )
Call AddFieldToIndex( "IDOrszag", False )

tdf.Indexes.Append idx

End Sub

' Create indexes
'=====

Sub CreateIndexes()

End Sub

' Create relations
'=====

Sub CreateRelations()
'=== Create relations between parent table Diak and child table DiákokCimeik =====

Set rel = dbs.CreateRelation("DiakDiákokCimeik")

rel.Table = "Diak"
rel.ForeignTable = "DiákokCimeik"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Diak and child table Feljegyzesek =====

Set rel = dbs.CreateRelation("DiakFeljegyzesek")

rel.Table = "Diak"
rel.ForeignTable = "Feljegyzesek"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Diak and child table Hianyzasok =====

Set rel = dbs.CreateRelation("DiakHianyzasok")

```

```

rel.Table                = "Diak"
rel.ForeignTable        = "Hianyzasok"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Diak and child table OsztDiak =====
Set rel = dbs.CreateRelation("DiakOsztDiak")

rel.Table                = "Diak"
rel.ForeignTable        = "OsztDiak"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Diak and child table Szulok =====
Set rel = dbs.CreateRelation("DiakSzulok")

rel.Table                = "Diak"
rel.ForeignTable        = "Szulok"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Diak and child table TantCsopDiakjai =====
Set rel = dbs.CreateRelation("Diakok_Jegyei")

rel.Table                = "Diak"
rel.ForeignTable        = "TantCsopDiakjai"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Diak and child table Telefonszamok =====
Set rel = dbs.CreateRelation("DiakTelefonszamok")

rel.Table                = "Diak"
rel.ForeignTable        = "Telefonszamok"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Feljegyzestipus and child table Feljegyzesek =====
Set rel = dbs.CreateRelation("A_feljegyzes_tipusa")

rel.Table                = "Feljegyzestipus"
rel.ForeignTable        = "Feljegyzesek"
rel.Attributes = 0

Call AddFieldToRelation("IDFeltip", "IDFeltip")

dbs.Relations.Append rel

```

```

'=== Create relations between parent table Orszag and child table Telepules =====
Set rel = dbs.CreateRelation("OrszagTelepules")

rel.Table                = "Orszag"
rel.ForeignTable        = "Telepules"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDOrszag", "IDOrszag")

dbs.Relations.Append rel

'=== Create relations between parent table Osztaly and child table Osztalyfonokok =====
Set rel = dbs.CreateRelation("OsztalyOsztalyfonokok")

rel.Table                = "Osztaly"
rel.ForeignTable        = "Osztalyfonokok"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("TanevNev", "TanevNev")
Call AddFieldToRelation("Evfolyam", "Evfolyam")
Call AddFieldToRelation("Betujel", "Betujel")

dbs.Relations.Append rel

'=== Create relations between parent table Osztaly and child table OsztDiak =====
Set rel = dbs.CreateRelation("OsztalyOsztDiak")

rel.Table                = "Osztaly"
rel.ForeignTable        = "OsztDiak"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("TanevNev", "TanevNev")
Call AddFieldToRelation("Evfolyam", "Evfolyam")
Call AddFieldToRelation("Betujel", "Betujel")

dbs.Relations.Append rel

'=== Create relations between parent table Osztaly and child table TantargyCsoport =====
Set rel = dbs.CreateRelation("OsztalyTantargyCsoport")

rel.Table                = "Osztaly"
rel.ForeignTable        = "TantargyCsoport"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("TanevNev", "TanevNev")
Call AddFieldToRelation("Evfolyam", "Evfolyam")
Call AddFieldToRelation("Betujel", "Betujel")

dbs.Relations.Append rel

'=== Create relations between parent table Tanar and child table Osztalyfonokok =====
Set rel = dbs.CreateRelation("TanarOsztalyfonokok")

rel.Table                = "Tanar"
rel.ForeignTable        = "Osztalyfonokok"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDTanar", "IDTanar")

dbs.Relations.Append rel

'=== Create relations between parent table Tanar and child table TantargyCsoport =====
Set rel = dbs.CreateRelation("TanarTantargyCsoport")

```

```

rel.Table = "Tanar"
rel.ForeignTable = "TantargyCsoport"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDTanar", "IDTanar")

dbs.Relations.Append rel

'=== Create relations between parent table Tanev and child table Osztaly =====
Set rel = dbs.CreateRelation("TanevOsztaly")

rel.Table = "Tanev"
rel.ForeignTable = "Osztaly"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("TanevNev", "TanevNev")

dbs.Relations.Append rel

'=== Create relations between parent table Tantargy and child table TantargyCsoport =====
Set rel = dbs.CreateRelation("TantargyTantargyCsoport")

rel.Table = "Tantargy"
rel.ForeignTable = "TantargyCsoport"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IDTantargy", "IDTantargy")

dbs.Relations.Append rel

'=== Create relations between parent table TantargyCsoport and child table TantCsopDiakjai
=====
Set rel = dbs.CreateRelation("TantargyCsoportTantCsopDiakJegy")

rel.Table = "TantargyCsoport"
rel.ForeignTable = "TantCsopDiakjai"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("TanevNev", "TanevNev")
Call AddFieldToRelation("Evfolyam", "Evfolyam")
Call AddFieldToRelation("Betujel", "Betujel")
Call AddFieldToRelation("IDTantargy", "IDTantargy")
Call AddFieldToRelation("IDTanar", "IDTanar")

dbs.Relations.Append rel

'=== Create relations between parent table TantCsopDiakjai and child table TantcsopDiakJegy
=====
Set rel = dbs.CreateRelation("TantCsopDiakjaiTantcsopDiakJegy")

rel.Table = "TantCsopDiakjai"
rel.ForeignTable = "TantcsopDiakJegy"
rel.Attributes = 0

Call AddFieldToRelation("TanevNev", "TanevNev")
Call AddFieldToRelation("IDTantargy", "IDTantargy")
Call AddFieldToRelation("Evfolyam", "Evfolyam")
Call AddFieldToRelation("Betujel", "Betujel")
Call AddFieldToRelation("IDTanar", "IDTanar")
Call AddFieldToRelation("IDDiak", "IDDiak")

dbs.Relations.Append rel

'=== Create relations between parent table Telepules and child table DiákokCimej =====

```

```

Set rel = dbs.CreateRelation("TelepulesDiákokCime")

rel.Table = "Telepules"
rel.ForeignTable = "DiákokCime"
rel.Attributes = dbRelationUpdateCascade+0

Call AddFieldToRelation("IRSZ", "IRSZ")
Call AddFieldToRelation("IDOrszag", "IDOrszag")

dbs.Relations.Append rel

End Sub

' Drop relation
'=====

Sub DropRelation(RelName As String)

Set rel = Nothing
On Error Resume Next
Set rel = dbs.Relations(RelName)
On Error GoTo 0

If Not rel Is Nothing Then dbs.Relations.Delete ( RelName )

End Sub

' Drop table
'=====

Sub DropTable(TableName As String)

Set tdf = Nothing
On Error Resume Next
Set tdf = dbs.TableDefs(TableName)
On Error GoTo 0

If Not tdf Is Nothing Then dbs.TableDefs.Delete ( TableName )

End Sub

' Drop index
'=====

Sub DropIndex(TableName As String, IndexName As String)

Set tdf = Nothing
Set idx = Nothing
On Error Resume Next
Set tdf = dbs.TableDefs(TableName)
Set idx = tdf.Indexes(IndexName)
On Error GoTo 0

If (Not tdf Is Nothing) And (Not idx Is Nothing) Then tdf.Indexes.Delete ( IndexName )

End Sub

' Add fields to table
'=====

Sub AddFieldToTable(FieldName As String, DataType As String, SizeCol As Integer, Attributes As Long, DefaultValue As Variant, ValText As String, ValRule As String, NotN As Boolean, ZeroLength As Boolean)

Dim fld As DAO.Field

Set fld = tdf.CreateField( FieldName, DataType )

If SizeCol <> 0 Then fld.Size = SizeCol
If Attributes <> 0 Then fld.Attributes = Attributes

```

```

fld.Required = NotN
If DataType = dbText or DataType = dbMemo Then fld.AllowZeroLength = ZeroLength
fld.DefaultValue = DefaultValue
fld.ValidationRule = ValRule
fld.ValidationText = ValText

tdf.Fields.Append fld

End Sub

' Add properties to table
'=====

Sub AddPropertyToTable( PropertyName As String, Value As Variant, DataType As String)

Dim prp As DAO.Property

Set prp = tdf.CreateProperty(PropertyName, DataType, Value)

tdf.Properties.Append prp

End Sub

' Add properties to field
'=====

Sub AddPropertyToField( FieldName As String, PropertyName As String, Value As Variant,
DataType As String)

Dim prp As DAO.Property
Dim fld As DAO.Field

Set fld = tdf.Fields( FieldName )
Set prp = fld.CreateProperty(PropertyName, DataType, Value)

fld.Properties.Append prp

End Sub

' Add fields to index
'=====

Sub AddFieldToIndex( FieldName As String, Descending As Boolean )

Dim fld As DAO.Field

Set fld = idx.CreateField( FieldName )

If Descending = True Then fld.Attributes = dbDescending

idx.Fields.Append fld

End Sub

' Add fields to relation
'=====

Sub AddFieldToRelation( PKField As String, FKField As String )

Dim fld As DAO.Field

Set fld = rel.CreateField( PKField )

fld.ForeignName = FKField

rel.Fields.Append fld

End Sub

```

9.2. -2.- számú melléklet : Iskolastatisztika és Bukási statisztika jelentések