



INTELLIGENS RENDSZEREK II. (ÜZLETI INTELLIGENCIA) LABORATÓRIUM

Adattárházak építése 3. laborgyakorlat feladatai

Készítette:

Kovács Dániel László
(dkovacs@mit.bme.hu)

Méréstechnika és Információs Rendszerek Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

2008, október.

1. Áttekintés

A gyakorlat célja: Megismertetni a hallgatókat az adattárházak felépítésének alapjaival.

A gyakorlat tartalma: A gyakorlat során a hallgatók különböző adatkinyerési és transzformációs lépések útján adattárházat hoznak létre, amit web-alkalmazásként is közzétesznek. A gyakorlat a következő főbb elemekből fog állni:

- Adattárház projektek létrehozása az IBM DB2 Data Warehouse Edition Design Studio 9.1.2-es verziójával.
- Data Flow adatfolyamok létrehozása, és szerkesztése
- Szöveges adatfájlok tartalmának adatbázis táblákba mozgatása adatfolyammal
- Adatminőség javítása további adatforrások hozzáadásával (az staging adatbázis táblák tartalmának automatikus ellenőrzése referenciák által)
- Adatfolyamok végrehajtása
- Végrehajtási Terv Gráfok (EPG-k) generálása, és értelmezése
- Vezérlési folyamatok (Control Flows) létrehozása, és adatfolyamok integrációja
- Iteráció beépítése a vezérlési folyamatokba
- Vezérlési folyamatok adatfolyam-alkalmazásokba csomagolása
- Websphere Application Server (WAS) alkalmazás-szerver üzembe helyezése
- WAS web-alkalmazásaihoz tartozó adatforrások megadása
- Adatfolyam-alkalmazás közzététele WAS web-alkalmazásként, és a közzétett web-alkalmazás folyamatként való futtatása, monitorozása, és hibakeresése
- WAS web-alkalmazások eltávolítása, és az alkalmazás-szerver leállítása
- Adatbázis CubeViews számára történő előkészítése (adatbázis engedélyezés)

2. Kritériumok

A gyakorlat során a hallgatóknak 3,5 órnyi tiszta munkaidő áll a rendelkezésére. Ezalatt az idő alatt kell teljesíteniük a következő szakaszban felsorolt feladatokat. A gyakorlat során végzett munka a gyakorlat végén kerül megítélésre (teljesítmény, és minőség tekintetében).

3. Feladatok

1. Indítsuk el a „C:\VMWare-images\oktatas\VIMIA430_-_Intelligens_Rendszerek_2_labor_-_DW350ERC4_-_3rd_exercise” könyvtárban található „3rd_exercise.vmx” virtuális gép „MeresElott” nevű snapshot-ját (**dupla kattintás a VMX fájlra, VM → Snapshot → MeresElott → Yes, Start this virtual machine, View → Full Screen**).
2. A virtuális gépen lépünk be a Windows-ba „db2admin” felhasználóként, „ibm2blue” jelszóval (és várjuk meg, amíg minden rendesen betöltődik).

A következőkben az **SQL Warehousing Tool (SQW)** nevezetű eszközzel fogunk megismerkedni, amely az IBM DB2 adatbázis-technológiájára épülő adattárházak belső menedzselésére szolgál (adatok mozgatására és transzformációjára). Ezzel az eszközzel egységes, üzleti folyamatainkhoz illeszkedő adattárházba integrálhatjuk a különböző, heterogén adatforrásokból (relációs adatbázisokból, fájlokból, stb) származó adatokat.

3. Indítsuk el a „Design Studio”-t az asztalon, és miután elindult, csatlakozzunk a **DWESAMP** adatbázis szerverhez (röviden: adatbázishoz). A Database Explorer részen (bal oldalt lent) kattintsunk a jobb egérgombbal a **DWESAMP** kapcsolatra, és válasszuk a **Reconnect** opciót. Csatlakozzunk „db2admin” néven és „ibm2blue” jelszóval.
4. A következőkben hozzunk létre egy új adattárház projektet (**File → New → Data Warehouse Project**).
5. A projekt neve legyen: **DW350 Warehouse Project**. Kattintsunk a **Next** gombra. Az adattárház projekt most hozzá szeretne fűzni a forrás és cél adatbázis táblákhoz, így hát ezeket meg kell adnunk a Data Project részben. Egy, vagy több Data Project-et is kapcsolhatunk egy-egy Warehouse Project-hez. Most a **DW350 Data Project** nevű Data Project-et válasszuk ki, és kattintsunk a **Finish** gombra.
6. Látható, hogy a Data Project Explorer részen (bal oldalt fent) megjelent az imént létrehozott adattárház projektünk. Bontsuk is ki, majd bontsuk ki benne a **Databases** részt, és azon belül kattintsunk duplán a **DW350 Data Model.dbm** adatmodellre. Ennek hatására megnyílik az adatmodell (és egyben a vásznon is megjelenik – ne zárjuk le). Látható, hogy a modell a **DWESAMP** adatbázisra hivatkozik, ám azon belül csak a **DWH** és **MARTS** sémára.
7. Bontsuk ki tehát a Database Explorer részben (bal oldalt lent) a megfelelő, immár zölden világító adatbázis kapcsolat sémáit (**DWESAMP → DWESAMP → Schemas**). Itt látszik, hogy a DWH és MARTS nevezetű adatbázis sémák mellett még számos egyéb séma is megjelenik. Egyszerűsítsük le a dolgokat. Kattintsunk jobb egérgombbal a **Schemas** feliratra, és válasszuk a **Filter** opciót, majd kattintsuk ki a **Disable filter** jelölődobozt, válasszuk ki a **Selection** rádiógombot, és jelöljük ki a **DWH** és **MARTS** sémákat, majd kattintsunk a **Finish** gombra. Ezek után, ha újra kibontjuk a Schemas részt, már csak az előbbi két sémát fogjuk látni benne.

A következőkben rátérünk az adattárház projektünk létrehozásának lényegi okára: olyan **adatfolyamokat (Data Flows)** fogunk építeni, melyek különböző DB2-es adattáblák között fogják mozgatni, és transzformálni az adatokat.

8. A Data Project Explorer részben (bal oldalt fent) a DW350 Warehouse Project nevű adattárház projektünkön belül kattintsunk jobb egérgombbal a **Data Flows**-ra, majd **New → Data Flow**. A létrehozni kívánt adatfolyam neve legyen „**Load DWH Fact**”, majd **Finish**. Látható, hogy jobb oldalt megjelenik a felhasználható adatfolyam-operátorok palettája, amik az adatok mozgatásáért, transzformációjáért, és tisztításáért felelősek.
9. A következőkben adjuk meg az SQW-nek, hogy mely DB2-es adatbázist használja a végrehajtás során. Alul, a **Properties** részben kattintsunk az **SQL execution schema** ikonra. Itt vagy konkrét sémát adunk meg, vagy egy változónak a nevét. Az utóbbi megoldás az adattárház alkalmazás közzétételekor (deployment) nagyobb rugalmasságot biztosíthat, ámde mi egyelőre maradjunk a fix értéknél. Legyen ez a **DWH** séma.
10. Hasonlóképp adjuk meg az **SQL execution database**-t is. A legördülő menüből válasszuk ki a **DWESAMP** adatbázist.
11. A következőkben két szövegfájl adattartalmát fogjuk „befolyatni” az adattárházunkba. A jobb oldalt látható adatfolyam-operátor palettából ragadjuk ki a **File Import** névre keresztelt operátor típust, és – folyamatosan nyomva tartva az egér bal gombját – mozgassuk a „Load DWH Fact” névre keresztelt adatfolyamunkhoz tartozó vászon (középen) bal oldalára. Az adatfolyamok mindig balról jobbra haladjanak!
12. A feljövő ablakban böngésszük ki a megfelelő fájlnevet (**C:\DW35\SQW\DWH_ITM_TXN_1.txt**), majd kattintsunk az **Open**-re. Ezek után kattintsunk a **Next**-re, hogy beállíthassuk az imént megadott fájl formátumát (szintaxisát). Ehhez kattintsunk a **Load from File Format** gombra, és válasszuk ki a megfelelő formátum-leíró fájlt (**C:\DW35\SQW\dwh_itm_txn.fileformat**), majd **Open**, **Next**, és **Finish**.

Látható, hogy a formátum-leíró fájl egy meta-adat, amelyet valaki manuálisan hozott létre ahhoz, hogy az általa leírt adatforrásokból automatikusan kinyerhetővé váljanak a szükséges adatok. Esetünkben tehát adott felépítésű DB2-es adatbázis táblába fognak kerülni az importált fájlokban lévő adatok.

13. Az előbbiekhöz hasonlóan egészítsük ki adatfolyamunkat egy újabb **File Import** operátorral, amely a **C:\DW35\SQW\DWH_ITM_TXN_2.txt** fájlt olvassa be. Itt azonban, miután az operátort a palettáról a vászonra mozgattuk (az előbbi File Import operátor alá), nyomban zárjuk is le **piros X** gombbal a felugró dialógus ablakot, és a létrehozott/kiválasztott operátor **Properties** részében (lent) adjuk meg a megfelelő információkat, azaz itt böngésszük ki az importálandó fájl nevét, illetve a formátum-leíró fájlt is.
14. A következőkben mindkét File Import operátor mellé tegyünk egy-egy **Distinct** operátort, és kössük össze a megfelelő File Import operátorok kimenetét (output port-ját) a megfelelő Distinct operátorok bemenetével (input port-jával).

15. Mindkét Distinct operátor **Properties** részében (lent), azon belül a **Column Select** rész **Selected Columns** mezőjében válasszuk ki (a CTRL gombot lenyomva tartva, bal egérgombbal kattintva) az **MKT_BSKT_TXN_ID**, **PD_ID**, és **ITM_TXN_TMS** oszlopokat, majd kattintsunk a **kis sárga balra nyílra**.

Ennek hatására a Distinct operátorok **result** kimenetén a megfelelő, beimportált fájlok adott 3 mezőben együttesen különböző adatai fognak ismétlődés nélkül megjelenni, míg a **discard** kimeneten azok az adatok jelennek meg, melyek eredetileg többször is előfordultak (az adott 3 mező együttes értékét tekintve).

16. A duplikált adatokat írjuk ki egy-egy külön fájlba! Ehhez először is mozgassunk egy **File Export** operátort a vászonra, a fentebbi Distinct operátortól jobbra, és picit lentebb, majd X-eljük ki a felugró ablakot.
17. Az elhelyezett File Export operátor Properties részében, azon belül a General fülben kattintsunk a **File Name** mellett lévő gombra (amin a kis gyémánt, és A1-es felirat szerepel), majd válasszuk ki a **Use Variable** opciót.

Itt tehát egy fokkal rugalmasabbak szeretnénk lenni, és változóként szeretnénk definiálni annak a fájlnak a nevét, amelybe a duplikátumok kerülnek.

18. Kattintsunk tehát most a File Name melletti beviteli mező jobb oldalán a „...” gombra, aminek hatására feljön egy **Variable Selection** ablak.
19. Hozzunk létre egy változó-csoportot! A **Variable Groups** mező alatt kattintsunk a **New** gombra, és **DW350** néven hozzuk létre a csoportot.
20. Az imént létrehozott változó-csoporton belül definiáljunk most két változót (egymás után, egyenként)! A **Variables** listától jobbra fent kattintsunk a **New** gombra, és adjuk meg a változó nevét, értékének típusát, értékét, és azt, hogy legkésőbb mikor lehet ezen az értéken változtatni (tervezési időben, futási időben, stb). Az egyik változó neve **discard1**, a másiké pedig **discard2** legyen. Mindkettő típusa **File**. Értékük rendre **c:\temp\discard1.txt**, és **c:\temp\discard2.txt**. A **Final phase for value changes** mező értéke pedig **EXECUTION_INSTANCE** legyen.
21. Miután létrejött mindkét változó, és láthatóak a Variables listában, kattintsunk a **discard1**-re, és nyomjuk meg a **Replace** gombot. Ennek hatására File Export operátorunk a **discard1** változóval azonosított fájlt fogja ezek után generálni.
22. Hozzunk létre egy másik File Export operátort is (az alsó Distinct operátortól jobbra, picit lentebb), az előzőhöz hasonlóképp, ámde az exportált fájlra immár a **discard2** változó hivatkozzon.
23. Most pedig kössük össze a megfelelő Distinct operátorok **discard** kimenetét a megfelelő File Export operátorok **input** bemenetével.
24. Ezek után kombináljuk össze a két Distinct operátor kimenetét! Húzzunk egy **Union** operátort a vászonra, és kössük az **input** bemeneteire a File Export operátorok **result** bemeneteit, majd bizonyosodjunk meg arról, hogy Union operátorunk **Properties**-ének **Set Details** fülében a **set operation** beállítás **UNION**-ra van állítva.

25. Ahhoz, hogy adatfolyamunk eddig (vagy bármely lépésben) előállított kimenetét vizsgálni tudjuk (pl. ellenőrizni), egy **Data Stage**-et kell létrehoznunk a folyamaton belül, a megfelelő helyen. Mozgassunk most tehát egy **Data Station** operátort a vászonra (a Union-tól jobbra), és kössük az **input** bementére a Union **result** kimenetét, és a **Ctrl+S** billentyűkombináció lenyomásával mentsük eddigi munkánkat.
26. A Data Station operátor azonban egyelőre még működésképtelen, mivel egy valódi adatbázis táblára (egy úgynevezett **Staging Table**-re) van szüksége ahhoz, hogy az adott lépésben előállt adatokat el tudja helyezni. A Data Project Explorer-ben (balra fent) bontsuk ki a következőket: **DW350 Warehouse Project → Databases → DW350 Data Model.dbm → DWESAMP → DWH**
27. Mivel esetünkben a Staging Table az **ITM_TXN** táblával megegyező lesz, ezért másoljuk le ezt a táblát. A Data Project Explorer-ben kattintsunk rá jobb gombbal, majd **Copy**, és kattintsunk jobb gombbal a **DWH** sémára, majd **Paste**.
28. A létrejött, **Copy of ITM_TXN** nevű tábla **Properties** részében írjuk át a tábla nevét **ITM_TXN_STAGE**-re (és pl. a Label mezőre átkattintva érvényesítsük a változtatásokat).
29. Új táblánk egyelőre „csak” egy modell, nincs még fizikai manifesztációja. Kattintsunk tehát jobb egérgombbal az **ITM_TXN_STAGE** táblára a Data Project Explorer-ben, és válasszuk a **Generate DDL** opciót. Kattintsunk most a **Deselect All** gombra, és miután minden jelölődoboz inaktívvá lett, kattintsuk be a **Fully qualified names**, és **CREATE statements** jelölődobozokat, majd **Next**, aztán megint **Deselect All**, jelöljük ki a **Tables**-t, és megint **Next**.
30. A generátor SQL script neve (**File Name**) legyen **ITM_TXN_STAGE.sql**. Kattintsuk be továbbá a **Run DDL on server** opciót, és **Next**.
31. Ezen a képernyőn válasszuk a **Use an existing connection** opciót, és a **DWESAMP** kapcsolatot, majd **Next**, és **Finish**. Ennek hatására tehát bizonyos kényszerek, és egyéb sajátosságok nélkül, sikerrel létrejött az ITM_TXN fizikai adattábla másolata, az ITM_TXN_STAGE (ami immár a Database Explorer-ben is látható kell, hogy legyen).
32. Mentsük le minden eddigi munkánkat: **File → Save All**
33. Most pedig, miután végre létrejött egy valódi Staging Table, kattintsunk az adatfolyamunk vásznán a **Data Station** operátorra, és vegyük szemügyre a **Properties** részt (lent). Először is a Station type **PERSISTANT_TABLE** kell, hogy legyen.
34. A **Staged database table** beviteli mezőtől jobbra található „...” gombra kattintva böngésszük ki a **DWH.ITM_TXN_STAGE** táblát. Ezzel tehát megadtuk, hogy a Data Station adatfolyam-operátor kimenete konkrétan hová kerüljön.
35. Kattintsunk az adatfolyamunk vásznára, és nyomjunk egy **Ctrl+S** billentyű kombinációt (ezzel lementve az imént tovább módosított adatfolyamunkat).

A következőkben validálni fogjuk a Staging Table-ben lévő adatokat. Ehhez referencia gyanánt két további adatforrást iktatunk a folyamba.

36. A vászon jobb oldalán látható palettáról mozgassunk egy **Table Source** operátor példányt a Data Station alá az adatfolyamban.
37. Mivel a forrás tábla a DWESAMP adatbázisban van, továbbá az Execution Database (az adattárház tekintetében) ugyancsak a DWESAMP adatbázis, ezért a Table Source operátort Properties részében a **Location** opció maradhat most **SQL execution database**.
38. Böngésszük most ki azt a forrás táblát (amit referencia gyanánt fogunk használni ahhoz, hogy ellenőrizzük a Data Station kimenetén keletkező adatok helyességét)! Kattintsunk a **Source database table** beviteli mezőtől jobbra lévő „...” gombra, és böngésszük ki a **DWH.PD** adattáblát.
39. Ezek után illesszük be a folyamba a másik referencia forrás-táblát! Az előbbiekhöz hasonlóan helyezzünk el egy Table Source operátort az előbbi alá, és forrástábla gyanánt böngésszük ki a **DWH.MKT_BSKT_TXN** táblát. **Finish**.
40. Ezen után helyezzünk el egy **Key Lookup** operátort a vásznon a Data Station, és Table Source operátoroktól jobbra, majd kattintsunk a **lookup** port-ja alatti kis ikonra ahhoz, hogy még egy lookup port jöjjön létre rajta (**lookup1**). Kössük a Data Station **result** kimenetét a Key Lookup **data** bemenetére. A PD Table Source **output**-ját kössük a Key Lookup **lookup**-jára, míg az MKT_BSKT_TXN Table Source **output**-ját a **lookup1**-re.
41. A következőkben konfiguráljuk kicsit a Key Lookup operátort! A Properties részben válasszuk ki a **Condition List** fület. A lookup port-nak megfelelő sorban kattintsunk bele a **Value** mezőbe, és utána a megjelenő „...” gombra, majd alkossuk meg a következő SQL állítást: **DATA_018.PD_ID = LOOKUP_018.PD_ID**

Lehet, hogy 018 helyett más szám fog szerepelni (ha például előzőleg töröltünk valamiféle operátort, majd újra létrehoztuk azt), de ez a lényegen nem változtat...

42. Ha kész az SQL kifejezés, **OK**, majd hozzuk létre hasonlóképp a lookup1 port-hoz is a következő kifejezést: **DATA_018.MKT_BSKT_TXN_ID = LOOKUP1_018.MKT_BSKT_TXN_ID**
43. Most kattintsunk a Key Lookup operátor Properties-ének **Select List** fülére, és ott kattintsunk a **Results Columns** táblázat első sorába, majd a **Shift** gomb lenyomása mellett az utolsó sorába is. Ennek hatására kijelölődik a teljes táblázat. Ekkor kattintsunk a **piros X** gombocskára a táblázat fejléce felett (ahhoz, hogy töröljük a táblázat tartalmát). Megjegyzés: ezt egyébként a táblázat tetején elhelyezkedő **dupla/szürke X** gomb lenyomásával is megtehettük volna mindenféle kijelölés nélkül.
44. Ezek után az **Available Columns** részben jelöljük ki a **DATA_018**-hoz tartozó összes oszlopot, és kattintsunk a **sárga jobbra nyílra** (ahhoz, hogy átmozgassuk őket a Result Columns-ba).

45. Most pedig, miután megoldottuk a Key Lookup operátor által végzett összehasonlítást/validálást, próbáljunk meg azokkal az adatbázis bejegyzésekkel (azaz sorokkal) is kezdeni valamit, amik nem mentek át a szűrőn, avagy nem egyeztek a referencia-adatokkal. Ehhez először is mozgassunk egy **File Export** operátort a Key Lookup operátortól jobbra. A felugró képernyőn a következő fájlnevet adjuk meg: **c:\temp\discards_item_txn.txt**, majd **Finish**, és kössük a Key Lookup **discard** port-ját az új File Export operátor **input** port-jához.
46. Mit kezdjünk viszont azokkal az adatokkal, amik átmentek a szűrőn (match)? – Hát próbáljuk meg őket egy adatbázis táblába tölteni. Ehhez helyezzünk el egy **Bulk Load Target** operátort a Key Lookup operátortól jobbra, a File Export fölé. A **Load into database table** mezőtől jobbra lévő „...” gombra kattintva böngésszük ki a **DWH.ITM_TXN** táblát, majd állítsuk a **Load mode**-ot **REPLACE**-re, és **Finish**.
47. Ha ez kész, kössük a Key Lookup **match** port-ját a Bulk Load Target operátor **input** port-jára. A felugró listából válasszuk a **Connect by name** opciót (mivel a Key Lookup match kimenetén előálló virtuális tábla struktúrája gyakorlatilag megegyezik Bulk Load Target-ben megadott DWH.ITM_TXN tábláéval).
48. Ellenőrizzük, hogy megfelelő-e a mezők összeillesztése! Ehhez kattintsunk a Key Lookup, majd pedig Bulk Load Table operátorok jobb felső sarkában található **+** jelre. Ha minden stimmel, akkor a **-** jellel újra elrejthetjük a részleteket.
49. Mentsük le eddigi munkánkat: kattintsunk az adatfolyam vásznára, és **Ctrl+S**.
50. Elkészült tehát az adatfolyam. Ellenőrizzük, hogy jó-e: a fenti menüből válasszuk ki a **Data Flow → Validate** opciót, aztán **OK**.
51. Ha az imént stimmelt az adatfolyam, úgy hajtsuk most végre! Kattintsunk a **Data Flow → Execute** menüpontra. A felugró képernyőn gyakorlatilag minden (Diagnostic, Resources, és Variables) megfelelő most. De azért kattintsunk csak rá a **Variables** fülre, és nézzük meg, hogy az adatfolyamunkban jelenleg szereplő 2 változónak aktuálisan éppen most mi az értéke.
52. Miután mindent szemrevételeztünk, és megfelelőnek találtunk, kattintsunk az **Execute** gombra. Ha minden rendben lefutott (Execution succeeded), akkor **OK**. *(Megjegyzés: elő szokott fordulni, hogy annak ellenére, hogy mindent jól csináltunk, a végrehajtás során névütközésre hivatkozva hibát jelez a rendszer (bug). Ekkor töröljük a megfelelő Distinct operátort a folyamból, és ugyanúgy hozzuk létre még egyszer. Ezt addig ismételjük, amíg végül már gond nélkül lefut a folyam. Az operátor result port-jának Union-os input-ra kötésekor a felugró menüből a **Connect by name** opciót válasszuk.)* Az eredményt pedig úgy mintavételezhetjük, ha a Database Explorer-ben az **ITM_TXN_STAGE** és **ITM_TXN** táblákra jobb egérgombbal kattintva kiválasztjuk a **Data → Sample Contents** opciót (aminek hatására az alsó ablakban láthatóvá válik a **Data Output**).

Felmerülhet a kérdés, hogy valójában, fizikai szinten hogyan is zajlik egy ilyen adatfolyam végrehajtása. A válasz az **EPG (Execution Plan Graph)** végrehajtási-terv gráfokban rejlik. Az EPG gráfok az adatfolyamhoz képest olyanok, mint a bytecode (.class) a Java forráskódhoz képest: a JVM (Java Virtual Machine), avagy most igazából az SQW keretrendszer ezeket képes végrehajtani, az adatfolyamokat önmagukban nem, csak fordítás után. Ez többek közt nagyfokú platform függetlenséget jelent.

53. Ahhoz, hogy megtekintsük az adatfolyamunkhoz tartozó EPG gráfot, generálnunk kell a kódját: legyen kiválasztva az adatfolyamunk (pl. úgy, hogy a vászonra kattintunk), majd a menüből válasszuk ki a **Data Flow → Generate Code** opciót. Rövid áttekintést követően le is zárhatjuk a kigenerált script-et...
54. A kigenerált script-nél ugyanis van az EPG gráfoknak egy jóval szemléletesebb, adatfolyamokéhoz hasonló megjelenítése is. Ahhoz, hogy ezt megnézhessük, legyen kiválasztva az adatfolyamunk (Load DWH Fact), és a menüből válasszuk a következő opciót: **Data Flow → View Generated EPG → Runtime EPG**
55. A megjelenített gráf gyakorlatilag már egy **Control Flow** vezérlési folyamat (rövidesen erről is szót ejtünk bővebben). Ahhoz, hogy ebben a viszonylag komplex folyamatban hibát tudjunk keresni, tegyük a következőt: győződjünk meg arról, hogy a Runtime EPG van aktuális kiválasztva (pl. a vásznon a megfelelő fül kék), és kattintsunk a menüsor (amiben a File, Edit, Navigate, ... menük vannak) alatt elhelyezkedő ikonok közül a **kis zöldes bogárkára** (amely a méretezéstől közvetlenül jobbra van). Ha a megfelelő ikon fölé visszük az egérkurzort, akkor a következő felirat kell, hogy megjelenjen: **Debug the execution plan graph (EPG)**. Kattintsunk tehát erre az ikonra.
56. A hibakeresés azonnali megkezdéséhez kattintsunk a **Debug** gombra, majd kattintsunk a kis zöld bogárka ikontól balra megjelent **sárga lefelé nyílra**. Ha fölé visszük az egérkurzort, a következő felirat jelenik meg: **Execute the current node and move to the next** – ebből is látszik, hogy a gomb egymás után történő lenyomásával lépésenként haladhatunk végig az EPG-n. 8-9 lenyomás után például a Console-on meg kell, hogy jelenjen egy SQL-es üzenet, mivel végrehajtásra került egy **SQLCodeUnit**. Tovább nyomkodva a nyilacskát újabb SQLCodeUnit-ok végére érhetünk (amit újabb Console üzenetek jeleznek).
57. A hibakeresés befejezéséhez kattintsunk a sárga lefelé nyíl melletti **kis piros kockára**. A kocka funkciója a következő: **Cancel debug session**.
58. Ezek után zárjuk le az EPG gráfot, és a „Load DWH Fact” adatfolyamot is (a vásznon lévő fülek **kis X** gombjaival).

Az imént létrehozott adatfolyam nagyszerűen működött, azonban önmagában nem alkalmazható futási (**runtime**) környezetben. Azaz például nem tehetjük önmagában közzé (**deploy**), mint egy **WAS** (Websphere Application Server) alkalmazás szerveren futó web-alkalmazást. Ehhez először is egy **Control Flow** vezérlési folyamatba kell ágyazni (egy, vagy több) adatfolyamunkat, és utána a vezérlési folyamatot, mint folyamatot (**process**) futtatni a WAS szerveren, persze csakis miután közzétettük. A következőkben tehát ebbe az irányba fogunk elindulni. Ennek során meg fogjuk érteni, hogy mi a különbség a vezérlési folyamatok, és az adatfolyamok között, és miért van szükség az utóbbiak előbbiekre való beágyazására.

59. Hozzunk létre egy új adattárház projektet, amely egy részleges projektre épül! Kattintsunk a **File → New → Other** opcióra. Ezen belül **Data Warehousing → Samples → SQL Warehousing Sample – Partial**, és kattintsunk a **Next**-re.
60. A projekt neve legyen **DW350 Partial Project**. Kattintsunk a **Finish**-re.

61. Zárjuk le a megjelent **README**-t, illetve minden mást is a vásznon, továbbá zsugorítsuk össze a Data Project Explorer-ben is az esetleg kibontott Warehouse Project-ünket (az áttekinthetőség kedvéért).
62. A Data Project Explorer-ben jobb egérgombbal kattintsunk a **DW350 Partial Project**-re, és válasszuk a **Project References** opciót. Válasszuk ki a **DW350 Data Project**-et, és **OK**.
63. A Data Project Explorer-ben bontsuk ki a **DW350 Partial Project → Data Flows** részt. Látható, hogy ehhez a projekthez több adatfolyam is tartozik. Ezek közül nyissuk is meg a **marts-store** nevezetű adatfolyamot dupla-kattintással.
64. A vásznon egy üres részre kattintva ellenőrizzük, hogy a Properties-ben (lent) az SQL execution schema értéke **MARTS**, és az SQL execution database értéke **DWESAMP**.
65. Az utolsó Table Join operátorral valami nem stimmel. Ezt jelzi a bal felső sarkában látható **piros X** is. Kattintsunk rá erre. Ahhoz, hogy orvosoljuk a problémát, **OK**, aztán az operátor **Properties** részének **Condition** fülében a „...” gombra kattintva egészítsük ki az SQL kifejezést a következő sorral: **AND IN_023.OU_TP_ID = IN1_023.CL_ID**. Ennek hatására nagyjából a következő SQL kifejezésnek kell adódnia:
- ```
IN3_023."parentkey" = IN2_023."memberkey" AND
IN4_023."parentkey" = IN3_023."memberkey" AND
IN5_023."parentkey" = IN4_023."memberkey" AND
IN_023.OU_IP_ID = IN5_023."memberkey" AND
IN_023.OU_IP_ID = IN6_023.OU_IP_ID AND
IN_023.OU_TP_ID = IN1_023.CL_ID
```
66. Ha ez kész, kattintsunk az **OK** gombra az SQL Condition Builder-ben.
67. Most mozgassunk egy **Order By** operátort a vászonra a Table Join-tól jobbra. Mivel inner join műveletet szeretnénk, ezért a Table Join **inner** port-ját csatlakoztassuk az Order By **input**-jára.
68. Az Order By operátor Properties részében, a **Sort Key Ordering** fülben a **Sort Columns**-hoz adjuk hozzá az **STR\_IP\_ID** oszlopot.
69. Tegyük egy **Table Target** operátort az Order By operátortól jobbra.
70. A **Target database table** értékét böngészés útján állítsuk **MARTS.STORE**-ra, az **SQL operation**-t hagyjuk **Insert**-en. Kattintsunk a **Finish**-re.
71. Csatlakoztassuk az Order By operátor **result** port-ját a Table Target **input**-jához.
72. Kattintsunk az egér jobb gombjával az imént létrehozott kapcsolatra (a két operátor port-jai között), és válasszuk az **Edit → Edit Column Connections** opciót. Ellenőrizzük, hogy a kimeneten és a bemeneten rendre ugyanazok az oszlopok szerepelnek-e (azaz megfelelő-e a mapping/leképezés).

73. Az **OK**-ra kattintva zárjuk le a Connections ablakot, mentjük le eddigi munkánkat (pl. **Ctrl+S**), és validáljuk a kiegészített adatfolyamot: **Data Flow → Validate**

Látható, hogy az előbbi beavatkozások hatására eltűnt a Table Join operátornál hibát jelzőt piros X jel. Adatfolyamunk tehát immár megfelelő.

74. A Data Project Explorer-ben most nyissuk meg a **marts-fact** adatfolyamot (dupla-kattintás), és keressük meg benne az egyetlen **Group By** operátort, majd menjünk a **Properties** részének a **Select List** fülébe. A **Result Columns** listában az **Expression** nevezetű oszlopot szélesítsük ki úgy, hogy minden, ami benne szerepel, láthatóvá váljon.
75. Kattintsunk bele az Expression oszlop azon mezejébe, amelyben a következő szerepel: **INPUT\_014.SUM\_ITM\_STM\_PRC \* INPUT\_014.SUM\_NBR\_ITM**
76. Kattintsunk ezek után a mezőben megjelenő „...” gombra (ahhoz, hogy szerkeszthessük az SQL kifejezést).
77. A **Functions** részben bontsuk ki az **AGGREGATION**-t, és kattintsunk duplán a **SUM(DOUBLE)**-ra. Az **SQL text** részben korrigáljuk az SQL kifejezést (pontosabban a zárójeleket) úgy, hogy a következő adódjon: **SUM(INPUT\_014.SUM\_ITM\_STM\_PRC \* INPUT\_014.SUM\_NBR\_ITM)**
78. Ha kész, **OK**. Ugyanígy alakítsuk át az **INPUT\_014.CG\_PD\_DEPT\_NM** kifejezést a következőre: **MIN(INPUT\_014.CG\_PD\_DEPT\_NM)**
79. Ha ezzel elkészültünk, kattintsunk a módosított marts-fact adatfolyam vásznára, és mentjük a változtatásokat (**Ctrl+S**), majd **validáljuk az adatfolyamot**.

A következőkben Control Flows vezérlési folyamatokba fogjuk ágyazni a Data Flows adatfolyamokat. Lényegében tehát az adatfolyamok a vezérlési folyamatok egyes „lépései/operátorai” lesznek. A különbség abban lesz, hogy a vezérlési folyamatban nem adatok folydogálnak, hanem konkrétan a vezérlés (mint pl. egy IF-THEN-ELSE struktúrában).

80. A Data Project Explorer-ben, a DW350 Partial Project alatt kattintsunk az egér jobb gombjával a **Control Flows**-ra, és válasszuk a **New → Control Flow** opciót.
81. A kiválasztott data warehouse project legyen a **DW350 Partial Project**. A vezérlési folyamat neve legyen **Marts Control Flow. Finish**.

Látható, hogy a vásznon megjelent az újonnan létrehozott vezérlési folyamat gráfja, ami momentán egyetlen egy **Start** operátorból áll.

82. Tegyük a Start operátor mögé egy **Data Flow** operátort. A Properties részben állítsuk át a **Label**-jét **Marts\_Time**-ra (és érvényesítsük a változtatást pl. úgy, hogy átkattintunk a Description mezőbe), a **Data Flow** részben ugyanígy pedig böngésszük ki a **marts-time** adatfolyamot.

83. Ismételjük meg az előbbi műveletet még háromszor! Hozzunk létre egy **Marts\_Product**, **Marts\_Store**, és **Marts\_Fact** Data Flow operátort a vezérlési folyamunkba (egymás után, sorban), melyek rendre a **marts-product**, **marts-store**, és **marts-fact** adatfolyamokra építenek.

84. A Start **start**-kimenetét (**kis kék jobbra nyíl**) kössük a Marts\_Time operátor **input**-jára (**kis kék jobbra nyíl**), utána pedig sorra az **On\_Success** kimeneteket (**kis zöld pipa**) a közvetlen rákövetkező operátor **input**-jára. Így tehát 5 operátorunk lesz sorba kötve: **Start → Marts\_Time → Marts\_Product → Marts\_Store → Marts\_Fact**

Ezzel tehát sorba kötöttük (daisy chain) a 4 létező adatfolyamunkat (sikeres végrehajtást feltételezve).

85. Mozgassunk most egy **Command** operátort a legjobboldalibb, Mart\_Fact nevű operátorunktól jobbra, és kössük össze őket **On\_Success**-szel.

86. A Command operátor Properties részében a Command type maradjon DB2 SQL Script, and **SQL script location** mezőbe viszont böngésszük ki a következőt: **C:\IBM\DWES\samples\data\countMartTables.sql**. A **DB2 connection**-be pedig válasszuk ki a **DWESAMP** adatbázist.

87. Fejezzük be vezérlési folyamunkat azáltal, hogy egy **End** operátort teszünk a végére, a Command-tól jobbra. A Command operátort **unconditional** kimenetét (**kis kék jobbra nyíl**) kössük az End **end**-bemenetére (**kis kék jobbra nyíl**).

88. Ezzel akár már készen is lehetnénk, de felmerülhet a kérdés, hogy mi történjen akkor, ha bármelyik adatfolyam végrehajtása kudarcot vall? Ebben az esetben az adott adatfolyamoknak megfelelő Data Flow operátorok **On\_Failure** port-jával (**kis piros X**) kellene kezdenünk valamit, például küldeni egy értesítő email-t az illetékesnek. Ezt Email operátorok vezérlési folyamba való illesztésével tudjuk megoldani. Mindazonáltal, ha az illetékes minden esetben ugyanaz, akkor szükségtelen minden egyes operátorhoz külön-külön Email operátort tenni. Elég tehát, ha most a Start operátortól jobbra, lentebb elhelyezünk egy **Email** operátort, és a Start **On\_Failure** port-ját (**kis piros X**) az Email operátor **input**-jára kötjük.

89. Rakjunk még az Email operátor mögé egy **End**-et is, és kössük őket össze **unconditional** alapon.

90. Az Email operátor Properties részében adjunk meg 1-1 email címet a **Sender**-hez, és a **Receiver**-hez is (itt ügyeljünk arra, hogy ne változóként, hanem **Fixed Value**-ként adjuk meg őket), illetve a **Subject**-hez írjuk be azt, hogy **Failure**.

91. Ezek után kattintsunk a vászonra, mentjük munkánkat (**Ctrl+S**), és validáljuk a vezérlési folyamatot (**Control Flows → Validate**). Ha minden rendben lezajlott, **X-eljük ki** a vezérlési folyamatot, és a két adatfolyamot is a vászonra.

A következőkben megismerkedünk a vezérlési folyamatokba iktatható **iterációkkal**. Ezek lényegében a FOR-ciklusokhoz hasonlítanak. Konkrétan adatfolyamok változó paraméterekkel történő iterált végrehajtásával fogunk foglalkozni. Ehhez az előzőleg létrehozott „Load DWH Fact” nevű adatfolyamunk módosított változatát fogjuk felhasználni.

92. A Data Project Explorer-ben zsugorítsuk most össze a **DW350 Partial Project**-et, és bontsuk ki a **DW350 Warehouse Project**-et, azon belül a Data Flows-t, ott pedig nyissuk meg a **Load DWH Fact** adatfolyamot.
93. Kattintsunk a **File → Save As** menüpontra, és **Load DWH Fact Update.dataflowxml** néven, újra mentjük el az adatfolyamot.
94. Módosítsuk most az adatfolyam elején található második File Import (**File Import\_02**) operátort! A Properties részében a **File Name**-et állítsuk át **Use Variable**-re (a kis gyémántos A1-es ikonra kattintva), majd kattintsunk a tőle jobbra lévő „...” gombra, hozzunk létre egy új változót **itm\_txn\_update\_file** néven, aminek a típusa **File**, értéke **C:\DW35\sqw\DWH\_ITM\_TXN\_1.txt**, és a „Final phase for value changes” pedig szokás szerint **EXECUTION\_INSTANCE** legyen, majd **OK**.
95. Válasszuk ki az újonnan létrejött változót a Variables listában, és kattintsunk a **Replace** gombra (ahhoz, hogy a második File Import\_02 operátorban megfelelőképp frissüljön a File Name mező).
96. Folytassuk most tovább az adatfolyam módosítását! Először is állítsuk át a **Union** operátor Properties részében, a **Set Details** fülben a **set operation**-t **INTERSECTION**-re.
97. Ezek után válasszuk ki a **Data Station** operátort, és a Properties-ének a legalján kattintsuk be a **Pass through** jelölődobozt. Ennek hatására a Data Station operátor végrehajtása során a Staging Table-be nem fognak bejegyzések kerülni (hiszen az adott, DWH.ITM\_TXN táblában az előzőek folytán már amúgy is vannak adatok, fölösleges volna őket most törölni). Mindebből továbbá az is látható, hogy a Data Station operátorokat a tervezés/fejlesztés/tesztelés után, élesben (a deployment során) akár ki is vehetjük az adatfolyamból.
98. Most mentjük a változtatásokat, és validáljuk (de végre ne hajtuk!!) az adatfolyamot.

Most pedig, a módosított adatfolyamra építve, létrehozzuk azt a vezérlési folyamatot, amely immár iterációt is tartalmaz. Az iterációt úgy fogjuk megvalósítani, hogy lesz egy szövegfájlunk, amelynek tartalma szövegfájlok neve és elérése. Az iteráció során ezen a listán végighaladva fogjuk az egyes elemeket (a fájlneveket, amik adatforrásokat azonosítanak) az imént módosított adatfolyamunk megfelelő (**itm\_txn\_update\_file**) változójával egyesíteni, és így az adatfolyamot ismételten végrehajtani.

99. Először is a **DW350 Warehouse Project**-hez hozzunk létre egy új vezérlési folyamatot **ITM\_TXN\_Control** néven.
100. Tegyük egy **Iterator** operátor-példányt a Start-tól jobbra, és csatlakoztassuk a Start **start**-kimenetét az Iterátor **input**-jához, majd az Iterator Properties-ében az **Iteration action**-t állítsuk át **Repeat look for each delimited data item in a file**-ra, a **File Name**-hez böngésszük ki a **C:\DW35\sqw\DWH\_ITM\_TXN\_update\_files.txt** fájlt, az **Iteration delimiter**-t állítsuk vesszőre (azaz legyen „,”), és az **Iteration variable** mezőnél lévő „...” gombra kattintva válasszuk ki az **itm\_txn\_update\_file**-t, majd **Select**.

101. Az Iterator operátor beállítását követően tegyünk egy **Data Flow** operátort a vászonra, az Iterator-tól jobbra, fentebb. A Data Flow operátor Properties részében adatfolyam gyanánt a **Load DWH Fact Update**-et adjuk meg.
102. Csatlakoztassuk most az Iterator kimenő **loop** port-ját a Data Flow **input**-jához, és a Data Flow **unconditional** kimenetét pedig az **EndIterator input**-jához.
103. Tegyük továbbá egy **End** operátort az EndIterator-tól jobbra, és csatlakoztassuk az EndIterator **unconditional** kimenetét az End operátort **input** port-jára.
104. Ezek után kattintsunk a vászonra, mentjük eddigi munkánkat (**Ctrl+S**), és **validáljuk a vezérlési folyamat** (de még semmiképp se hajtjuk végre!!).
105. Először is, ha „mindez volt, ahogy írva”, nyissuk meg az adatforrások listáját tartalmazó **C:\DW35\sqw\DWH\_ITM\_TXN\_update\_files.txt** fájlt Notepad-del. A fájl tartalma a következő kell, hogy legyen (írjuk át, ha nem ilyen):

C:\DW35\sqw\DWH\_ITM\_TXN\_update\_1.txt,C:\DW35\sqw\DWH\_ITM\_TXN\_update\_2.txt

106. **Mentsük le, és zárjuk be** a szövegfájlt. Hajtuk végre (**execute**) a vezérlési folyamatot. Bizonyosodjunk meg arról, hogy a végrehajtás gond nélkül végbement (**succeeded**), és végezetül **zárjuk le** a vásznon az adatfolyamot, és a vezérlési folyamatot.

A következőkben megismerkedünk majd azokkal a megoldásokkal, amik lehetővé teszik, hogy az SQL Warehousing Tools-szal eddig elkészített vezérlési folyamainkat web-alkalmazásokként is közzétegyük. Ehhez a **WAS (Websphere Application Server)** alkalmazás szervert fogjuk igénybe venni. Erre a szerverre fogjuk deploy-olni a web-alkalmazásunkat (a megfelelőképp becsomagolt vezérlési folyamo(ka)t), és ezen is fogjuk futtatni őket. A futási (runtime) környezet menedzselésére a **DWE Administration Console** web-alkalmazás való.

107. A Data Project Explorer-ben jobb egérgombbal kattintsunk a **DW350 Warehouse Project**-re, és ott válasszuk a **New → Data Warehouse Application** opciót, majd **Next**. A **Profile Name** legyen **DW350\_Load\_Profile**, majd megint **Next**. Az **Available Control Flows** részről tegyük át az ebben az adattárház projektben található egyetlen vezérlési folyamunkat (**ITM\_TXN\_Control**) a **Selected Control Flows** részre, majd **Next**. A Resource Profiles részen ne változtassunk, inkább kattintsunk a **Next**-re, majd a változókon se változtassunk (hiszen mindhárom hivatkozva van a vezérlési folyamban), így megint **Next**, majd mivel a futási környezetben nincs most szükségünk DDL-re, ezért megint **Next**, aztán kattintsunk a **Next**-re a Saving Application Profile, és Code Generation oldalakon is, ám a Package Generation oldalon állítsuk be a **Zip file directory**-t a **c:\temp**-re, és **Finish**, és **OK**.
108. Ezek után lezárhatjuk a Design Studio-t (**File → Exit**), és indítsuk el a WAS alkalmazás szervert (**Start menü → All Programs → IBM WebSphere → Application Server 6 → Profiles → default → Start the Server**). Várjuk meg, amíg a felugró parancssori ablak magától bezáródik (ez 1-2 percig is eltarthat).

109. Most pedig nyissuk meg a WAS szerveren alapértelmezésben futó **dweadm** web-alkalmazást. Ehhez kattintsunk duplán az asztalon lévő **DWE Administration Console** ikonra, majd a feljövő browser-ablakban lépünk is be (**Log in**) **db2admin** felhasználóval, és **ibm2blue** jelszóval a web-alkalmazásba.

A következőkben adatforrásokhoz fogunk csatlakozni, másképp ugyanis a WAS szerver nem lesz képes hozzájuk férni.

110. A bal oldali menüben bontsuk ki a **DWE SQL Warehousing → Resources** részt, és kattintsunk a **Create Data Sources**-ra (majd várjunk egy picit, amíg betöltődik a megfelelő oldal).
111. **JNDI Name**-nek a **jdbc/dwesamp**-ot adjuk, az **Attached to WAS** opció legyen **No**, írjuk be a megfelelő **felhasználó nevet**, és **jelszót**, majd a **Display Name** legyen **DWESAMP**, a **Database Type** maradjon **DB2**, a **Database Name** pedig legyen a már megszokott **DWESAMP**.
112. A felkonfigurált adatkapcsolat leteszteléséhez kattintsunk a **Test connection** gombra, és türelmesen várjuk meg az eredményt. Ezek után, ha minden rendben volt, kattintsunk a **Finish**-re.

A laborgyakorlat (avagy az egyszerűség) kedvéért a forrás és cél adatbázis táblák most egyazon adatbázisban helyezkednek el. A gyakorlatban azonban ez nyilván nem mindig van így, és ezért legtöbbször külön adatkapcsolatot kell létrehozni mind a forrás, mind a cél adatbázisok számára.

113. A bal oldali menüben bontsuk ki a **Data Warehouse Applications** részt, és ott kattintsunk a **Deploy Data Warehouse Applications**-re.
114. Bizonyosodjunk meg arról, hogy a **Location of .zip File on Web Client** opció van kiválasztva, majd böngésszük ki az előzőekben létrehozott ZIP fájlt (**c:\Temp\DW350\_Load\_Profile.zip**), majd kattintsunk a **Next** gombra, aztán az áttekintő oldalon is kattintsunk a **Next**-re.
115. Változtassuk a web-alkalmazás nevét (**Application Name**) **DW350WHA**-ra, az **Application Home Directory** legyen **c:/DW350WHA/Home**, a **Log Directory** legyen **c:/DW350WHA/log**, a **Working Directory** legyen **c:/DW350WHA/Work**, majd kattintsunk a **Next**-re.
116. Mivel egyetlen adatforrásunk van csak, ezért nincs sok választásunk, kattintsunk a **Next**-re. Mivel rendszer erőforrásokat sem definiáltunk (FTP szerverek, stb), ezért újfent kattintsunk a **Next**-re.
117. A változóink esetében minden alapértelmezett érték megfelelő, így hát kattintsunk a **Finish**-re. Vegyük észre, hogy a deploy-olt web-alkalmazás automatikusan üzembe is lép (**Enabled**).
118. Kattintsuk be a **DW350WHA** melletti Select-jelölődobozt, majd kattintsunk az **Update** gombra. Válasszuk ki a **Variables** fület, és tekintsük át (itt még akár most is változtathatnánk az értékeken).



119. Ezek után bontsuk ki a bal oldali menüben a **Processes** fület, és kattintsunk a **Run Processes** linkre.
120. Itt válasszuk ki egyetlen folyamatunkat (**Select** checkbox), és kattintsunk a **Start**-ra. A következő képernyőn a web-alkalmazás adott, futtatandó példányának nevét adhatnánk meg, de mi kattintsunk inkább a **Next**-re, és hagyjuk, hogy automatikusan generálódjon egy alapértelmezett név.
121. A változók értékén továbbra sincs értelme változtatni, habár, mivel EXECUTION\_INSTANCE-ra állítottuk megváltoztatásuk végső fázisát, ezért most újfent lehetőségünk volna rá. Kattintsunk a **Finish**-re.
122. A web-alkalmazás e példányának állapota kezdetben **Started**, majd **Running**, majd pedig **Finished** lesz. Ha a **Refresh** gombot kellő gyakorisággal nyomkodjuk, úgy mindegyik állapotot láthatjuk. Mikor az állapot **Finished**-re váltott, kattintsuk be a **Select** jelölődobozt, és kattintsunk a **Monitor** gombra.
123. A bal oldali menüben kattintsunk a **Monitor Process Instances**-ra, majd kattintsunk alkalmazás-példányunk nevére (**Instance name**), és nézzük meg a HTML log fájlt a **Process Log** sorban. Ha bármilyen hiba adódott a futtatás során, azt itt láthatjuk.
124. Folytassuk az adminisztrációs konzol megismerését a **History and Statistics** résszel (ezt bontsuk most ki a bal oldali menüben).
125. Kattintsunk a **View Deployment History**-ra. Itt nincs sok látnivaló.
126. Kattintsunk a **View Statistics for Process Instances**-ra (bal oldalt), és kattintsunk a folyamat-példány nevére. Nézzük végig a különböző fűleket, amikben különböző statisztikai információkat láthatunk folyamatunk e példányáról.
127. A bal oldali menüben még megnézhetjük a **Logs and Traces** részt is, ahonnan gyakorlatilag ugyancsak hozzáférhetünk az előbbi információkhoz, továbbá a szerverre vonatkozó statisztikákat is megtekinthetjük.
128. Folytassuk most alkalmazásunk uninstallálásával! A bal oldali menüben kattintsunk a **DWE SQL Warehousing → Data Warehouse Applications → Manage Data Warehouse Applications** linkre.
129. Itt **válasszuk ki** web-alkalmazásunkat, kattintsunk az **Uninstall** gombra, bizonyosodjunk meg arról, hogy a **Yes** van kiválasztva, és **Finish**.

Egyelőre, ezen a laborgyakorlaton még nem foglalkozunk a Cube Views-szal, vagy az Intelligent Miner-rel. Mindazonáltal mindkettő számára szükséges az, hogy engedélyezzük (enable) a megfelelő adatbázist, amire alapoznak. Ez lényegében azt jelenti, hogy az említett alkalmazásoknak az adott adatbázisban adott táblákra, tárolt eljárásokra, felhasználó által definiált funkciókra, stb, van/lehet szüksége. Az engedélyezés során tehát ezek az objektumok jönnek létre. Természetesen nem csak a DWE Administration Console-on engedélyezhetünk adatbázisokat, hanem a Design Studio-ból is, ámde az utóbbival inkább fejlesztői és/vagy teszt adatbázisokat szokás engedélyezni, míg az előbbivel éles verziókat is.

130. Hozzunk először is létre egy adatbázis profilt, amit engedélyezhetünk! Kattintsunk a **DWE Common → Database Profile** linkre, majd a **Create** gombra.
131. A **profil neve** legyen **CVProf**. A **Profile Create Option** legyen **Create Profile with a New Data Source**. Kattintsunk a **Next**-re.
132. Az új képernyőn a **JDBC Provider** legyen **DB2 Universal JDBC Driver**, a **Data Source Name** legyen **CVSAMPLE**, a **JNDI Name** legyen **jdbc/dwe/cvsample**, a **Database Name** legyen **CVSAMPLE**, és kattintsunk a **Next**-re, majd írjuk be jelszavunkat, és **Create**.
133. **Válasszuk ki** a létrejött profilt, és kattintsunk a **Test** gombra.
134. Ha minden rendben volt, akkor kattintsunk most a bal oldali menüben a **DWE Common → Database Enablement** linkre, majd jelöljük ki mind a **Mining**, mind pedig az **OLAP** jelölődobozokat, kattintsunk a **Check Enable Status** gombra. Láthatjuk, hogy az adatbázis nincs még engedélyezve egyik szempontból sem, így kattintsunk a **Back** gombra, és kattintsuk ki a **Mining** checkbox-ot, és kattintsunk az **Enable Database** gombra, majd az **Enable**-re.
135. Most pedig, hogy ezzel készen vagyunk (és előkészítettük a következő mérésre a CVSAMPLE adatbázist) – és a gyakorlatvezetőnek is bemutattuk eddigi munkánkat –, jelentkezzünk ki a jobb felső sarokban lévő kis **Logout** linkre kattintva, aztán kattintsunk a **Logout** gombra, és **zárjuk le** a böngészőt.
136. Most pedig állítsuk le a WAS szerveret (**Start menü → All Programs → IBM WebSphere → Application Server 6 → Profiles → default → Stop the Server**).