**Szakmai gyakorlat dokumentáció**

***07.01.***

Kick off: A szakmai gyakorlat felvezetése, annak feladatának ismertetése. Előzetes érdeklődési körök felmérése. Ismerkedés az SAP HANA rendszerrel.

***07.02.***

Tanári oktatás az alapokról. Nehézségek akadtak a szerverrel, ezért csak korlátozottan lehetett feladatokat végezni. Önálló tanulás az SAP HANA tutorialok segítségével. Feladatok megoldására tett kísérletek, leginkább kudarcokkal, vagy szerverhiba vagy pedig a segítségek elavultsága miatt.

<https://blogs.sap.com/2017/09/04/xs-advanced-for-not-so-dummies/>

<https://blogs.sap.com/2017/09/05/xs-advanced-for-not-so-dummies-pt-2-multi-target-applications/>

<https://blogs.sap.com/2017/09/22/xs-advanced-for-not-so-dummies-pt-3-microservices/>

<https://blogs.sap.com/2018/02/16/xs-advanced-for-not-so-dummies-routing/>

***07.03.***

Internetes anyagok böngészése, önálló feldolgozása:

SAP HANA Express Edition install:

<https://developers.sap.com/tutorials/hxe-database-server.html>

SAP HANA XSA development:

<https://developers.sap.com/mission.xsa-get-started.html>

<https://developers.sap.com/group.hana-xsa-get-started.html>

<https://developers.sap.com/group.scp-8-odata-service.html>

<https://developers.sap.com/group.scp-5-node.html>

<https://developers.sap.com/group.cp-frontend-angular-1.html>

<https://developers.sap.com/group.cp-frontend-react-1.html>

SAP HANA Core Data Services:

<https://developers.sap.com/tutorials/xsa-ml-e2e-create-cds-db.html>

***07.04.***

Önálló ismeretszerzés. Megkaptuk a feladatokhoz szükséges csv fájlt, amivel el is kezdtünk dolgozni. További szerverhibák, a feltöltés sikertelensége miatt adattisztítás.

***07.05.***

A saját szerverünk működőképes lett, így az előző napi problémák nagy része megoldódott. Létrehoztuk a táblázatokat, a csv fájlt is tudtuk importálni. A maradék időben az adatbázisunkkal foglalkozás.

Az importálás menete:

* A saját szerverünkre bejelentkezünk: <https://oktnb132.inf.elte.hu:53075/watt/index.html>
* Új Multi Target Applicationt hozun„New Project From Template”k létre.
* Jobb klikk a mappánkra: „New Project From Template”
* „New SAP HANA database module”
* Az így létrejött src mappára jobb klikk: „New HDB CDS Artifact”
* Ezt megnyitva egy kontextust látunk, és azon belül hozzunk létre egy entityt, melyeknek tetszőleges neveket adhatunk.
* Az entityt megnyitva létrehozzuk a fejléceket, és beállítjuk a típusokat. A fejlécek megfelelnek a kapott csv fájl fejléceivel.
* Buildeljük a projektet
* Database explorerben létrehozunk egy új adatbázist „Add a database to the Database Explorer” lehetőségre, és a felugró listából kiválasztjuk az MTA projectünk nevét
* Ezután a table fülön belül megtaláljuk a létrehozott adatbázis modellünket, amire jobb gombbal kattintva felugrik az „Import Data” lehetőség
* A lehetőségek közül kiválasztjuk a csv fájlunkat
* Megnézzük, hogy az attribútumokat jól ismert-e fel, ha nem, akkor ezt kézzel korrigáljuk.

**Első hét összegzése:**

Leginkább önálló tanulásról szólt, és az SAP HANA-val való ismerkedéssel. A szerver hibái miatt korlátozottan tudtunk bármit is csinálni, ami csak a hét végére oldódott meg.

***07.08.-07.12.***

Tanári engedéllyel szabadságon voltam a héten.

A diáktársaim által érdeklődtem, hogy miket csináltak, és igyekeztem képben maradni a dolgokkal.

***07.15.***

Megkezdtem a felzárkózást a csoporttársakhoz. A többiek segítőkészek voltak, és sokat számított ez a segítség. OData létrehozása az elmondott instrukciók alapján.

* tarsas-ba „New Node.js Module” (fontos az „Enable XSJS support” kipipálása, a modult

odata-nak neveztem el)

* odata lib mappájába „New File” (myodata.xsodata), ebbe a következő kód került:

service{

"table.data" as "egesztabla";

}

* server.js fájl 11. sorában a redirectUrl-t átírtam „/myodata.xsodata”-ra
* build után kaptunk egy linket (https://oktnb132.inf.elte.hu:51069/myodata.xsodata), ennek

végére különböző lekérdezéseket illesztve kapjuk meg a kívánt adatokat (pl.

<https://oktnb132.inf.elte.hu:51069/myodata.xsodata/$metadata>)

***07.16.***

Altáblázatok létrehozása, melyek a kalkulációs nézethez fognak kelleni. Ezek „kereteit” a tarsas/db/src/table.hdbcds fájl vizuális nézetében csinálom meg. Mikor végeztem a létrehozással, a Database Explorer konzolablakában, SQL utasításokkal töltöm fel az altáblázatokat a megfelelő adatokkal.

A következő altáblázatokat hoztam létre:

* TARSKODOK(TARS\_ROV\_NEV, TARS\_HOSZ\_NEV, CIM\_EGYBEN, TARS\_TIPUS\_KOD, ADOSZAM,

GAZD\_FORM\_KOD, CEGALL\_KOD, NEMGAZD\_AG\_KOD, NEMGAZD\_AGAZAT\_KOD,

NEMGAZD\_SZAKAGAZAT\_KOD, JEGYZ\_TOKE\_ERT\_HUF, ORSZAG\_KOD, REGIO\_KOD,

MEGYE\_KOD, TELEPULES\_KOD, ASZ\_EVE)

* TARSTIPUS(TARS\_TIPUS\_MEGNEV, TARS\_TIPUS\_KOD)
* GAZD(GAZD\_FORM, GAZD\_FORM\_KOD)
* CEGALL(CEGALL, CEGALL\_KOD)
* NEMAG(NEMGAZD\_AG\_MEGNEV, NEMGAZD\_AG\_KOD)
* NEMAGAZAT(NEMGAZD\_AGAZAT\_MEGNEV, NEMGAZD\_AGAZAT\_KOD)
* NEMSZAKAGAZAT(NEMGAZD\_SZAKAGAZAT\_MEGNEV, NEMGAZD\_SZAKAGAZAT\_KOD)
* ORSZAG(ORSZAG, ORSZAG\_KOD)
* REGIO(REGIO, REGIO\_KOD)
* MEGYE(MEGYE, MEGYE\_KOD)
* TELEPULES(TELEPULES, TELEPULES\_KOD)

***07.17.***

Hallgató társ segítségével Angular telepítése, működésre bírása. Kaptam egy tömörített mappát, amely keretül szolgálhat saját Angularos kódom megírására.

Calculation View létrehozása egy hallgató kolléga segítségével. Lépései:

* tarsas src mappájának .hdiconfig fájljában plugin\_version átírása 2.0.30.0-ra (csak ezzel
* működik a kalkulációs nézet)
* szintén src mappába „New Calculation View” (calcview.hdbcalculationview)
* Create Join-nal behúzok egy joint a szerkesztőbe, amit „data”-ra nevezek át
* „Add Data Source” funkcióra kattintás után megadom az altáblázatokat forrásként
* ezután dupla kattintás a „data”-ra – Join Definition fülben az inner joinok meghatározása az
* adattagok összekötésével
* ezt követően a Mapping fülbe az adattagok behúzása (mindegyiket csak egyszer, tehát mivel
* a MEGYE-ben szerepel a MEGYE és a MEGYE\_KOD, a TARSKODOK-ból már nem húzom be a
* MEGYE\_KOD-ot még egyszer)
* szerkesztő bezárása, dupla kattintás az Aggregationre
* Mapping fülben itt is minden adattagot behúzunk
* build után működik a Data Preview

***07.18.***

Az Angularos projekt befejezése, véglegesítése. A többiek segítsége nagyban megkönnyítette a feladatomat, gyorsan be tudtam hozni a lemaradásomat. GitHubos felhasználót hoztunk létre, majd mindenki feltöltötte a saját forráskódjait és dokumentációját. Innentől kezdve teljes erőbedobással tudtam készülni a következő projektre, ami a heti megbeszélésen választottak szerint python környezetben lévő adatbányászat lett.

***07.19***

Mivel már régen programoztam python nyelvben ezért kicsit újra fel kellett eleveníteni a korábbi ismereteimet, de nem tartott túl sokáig, hogy visszajöjjön a tudásom. Az ismétlés után az adatbányászatnak néztem utána. Hogy néz ki ez a folyamat. Milyen adatbányászatok vannak, miért jó ez az egész.

Amikről olvastam:

* Cluster Analysis: Hasonló adatok csoportosítása.
* Classification: Adatok közti kapcsolat vizsgálata.
* Regression analysis: Függöség, függetlenség vizsgálata.
* Association: Nagyobb adathalmazok közti kapcsolat vizsgálata.
* Time Series: Adatok vizsgálata időrendben.
* Data preprocessing: Adatgyűjtés után, adatok tisztítása, normalizálása.
* Social Network: Szociális háló vizsgálata.
* Recommender system: Reklámok elhelyezése, megjóslása annak, hogy adott terméket kik vennének.
* Miscellaneous: Egy képből másik kép készítése, képek vizsgálata.

**A hét összegzése:**

Ez a hét a pótlás, és a tovább haladás közötti váltakozásról szólt. A többi diák segítőkészsége miatt gyorsan tudtam felzárkózni, aztán pedig haladni tovább.

***07.22. – 07.26.***

Ezen a héten is szabadságon voltam, mivel már a nyár előtt megvoltak a programjaim, a szakmai gyakorlatról pedig csak ez után értesültem. Igyekeztem amennyire csak lehet a távollétemben is lépést tartani a többiekkel. Azt az információt kaptam, hogy főleg dokumentációk és tutorialok olvasásával töltötték az időt. Mivel a sikertelen próbálkozásaikat nem tudtam én is kipróbálni, ezért a szabadidőmben ezekben a témákban böngésztem.

***07.29. – 07.30.***

Próbálkoztam megvalósítani az eddig olvasottak alapján az adatbányászatot python és SAP környezetben is. Eljárásokat próbáltam lefuttatni inkább kevés sikerrel, a többi hallgatónak se volt nagyobb szerencséje, de a végére egy kezdetleges klaszterezeés már ment nekik, amit én is próbáltam megérteni.

***07.31.***

Beszámolót tartottunk az eddigi gyakorlati munkánkról, és az elért eredményekről, és a további lehetőségeket is megbeszéltük. Vincellér Zoltán tanár úr felvetett egy új projektötletet, egy szenzorokkal foglalkozó feladatot. Ez az ötlet nekem nagyon megtetszett főleg az okos otthonnal való foglalkozás. Mivel az adatbányászatban nem voltak nagy sikereim, és annyira nem is tetszett ezért szívesen elvállaltam ezt az új szenzoros feladatot.

***08.01. – 08.02.***

Ezt a két napot legfőképpen a szenzorok tanulmányozásával töltöttem, utánanéztem milyen fajtáik vannak, melyek kellenek egy smart home megvalósításához, és hogyan alkalmazhatók még. Tanulmányoztam ezen kívül az IoT rendszereket is, és az ehhez kapcsolódó témaköröket.

***08.05.***

Újabb megbeszélésen megkaptuk a tényleges szenzoros feladat felvázolását. Kaptunk egy adatbázist, amiből az adatokat egy köztes szabadon választott típusú köd felhasználásával fel kell tölteni az adatokat, úgy mint, hogy ha valós idejű feltöltés lenne.

***08.06. – 08.07.***

Az adatbázist próbáltuk meg feldolgozni azokkal, akik szintén becsatlakoztak ebbe a projektbe. Felbontottuk az adatokat, hogy a feltöltés elvégezhető legyen, a lehető legkedvezőbb módon. Felbontottuk a feladatot szenzorszimulátor-fog és frontend részre. A parsert alkalmassá tettük az SAP HANA rendszerbe való feltöltésre.

***08.08. – 08.09.***

A többiek tovább foglalkoztak ezzel az adatbázis szenzoros feladattal, és még jobban feldolgozták azt, adatbányászatot is végeztek rajta. Mivel nekem még a hiányzásaim után is pótolnom kellett ezért kértem még magamnak feladatot. Szerettem volna továbbra is szenzorokkal foglalkozni, és felvetettem, hogy van egy saját raspbery pi számítógépünk, amihez vannak szenzorok is, és ehhez tudok-e valami feladatot csináni. Béleczki András tanár úr azt javasolta csinálhatok egy real-time adatbázist, és erre használjam a Google FireBase nevű internetes alkalmazását.

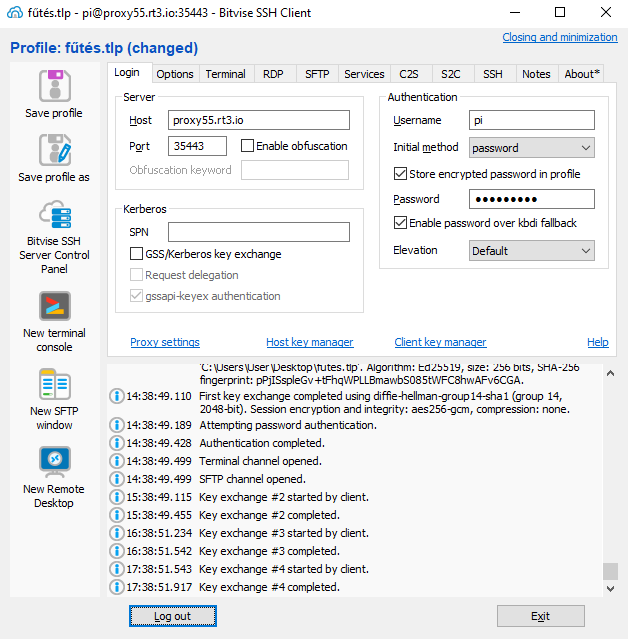
**A hét összegzése:**

Áttértem egy szenzorokkal foglakozó feladatra, amin a többiekkel közösen dolgoztunk, majd kaptam egy önálló feladatot is amin következő hetekben dolgozhatok.

***08.12. – 09. 12.***

Ezekben a hetekben már otthonról dolgoztam, mivel már a többieknek nem kellett bejárni az egyetemre, csak esetleg annak aki szintén pótólt mint ahogy én is, de a tanárok megengedték, hogy egyedül nem kell bemennem dolgozni, hanem ezt végezhetem home officeban is.

Az otthoni raspberry pi számítógépen dolgoztam, ezen csináltam mindent. A raspberryhez való csatlakozást a Bitwise SSH Client nevű alkalmazással oldottam meg, de amikor nem otthonról akartam programozni akkor a remote.it Connect nevű programot is igénybe kellett vennem.



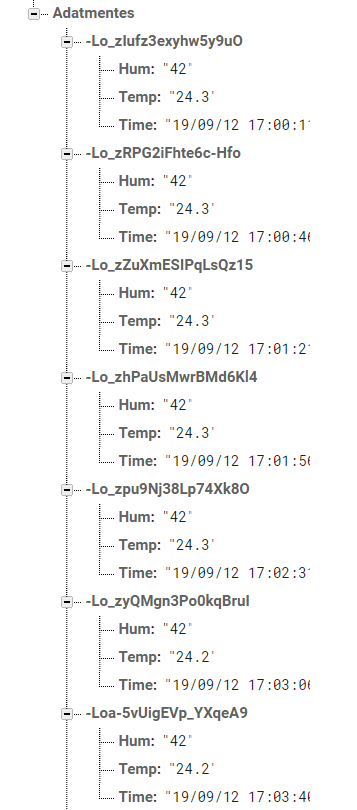
Először el kellett érnem, hogy le tudjam kérdezni a szenzoroktól az adatokat, ami hosszadalmasnak bizonyult. Először sikerült megcsinálnom, hogy a házban lévő érzékelő hőmérséklet adatait le tudtam kérdezni. Mivel házon belül szabályozva van a hőmérséklet ezért ott szinte csak tized fok eltérések voltak, és így az adataim nem nagyon tértek volna el egymástól. Ezért inkább egy kinti hőmérséklet szenzort választottam, ami már így eltérőbb adatokat produkált. Miután ezekkel megszenvedtem rájöttem, hogy van egy hőmérséklet állomásunk is otthon, amiből kis nehézségek árán, de le tudtam kérni a hőmérsékletet, és a páratartalmat is.

Ezekután tanulmányoztam a FireBase adatbázis rendszert, több tutorialt is elolvastam, YouTube videókat néztem ezzel kapcsolatban, és igyekeztem elsajátítani az egész környezetet.

A tanulmányozások után sikerült feltölteni az adatokat a valós idejű adatbázisba. Az adatok feltöltését egy végtelen ciklusba tettem, és először úgy gondoltam, hogy 5 másodpercenként töltöm majd fel az adatokat. Ezzel az volt a baj, hogy ezek IC szenzorok, azaz integrált áramkörűek és ezért ha nagyon sokat használják rövid időközönként, akkor hőt bocsájtanak ki, és így a hőmérséklet mérési adatai pontatlanok lesznek. Így 30 másodpercesre állítottam az adatok lekérdezését, és beállítottam egy ellenőrzőt, hogy ha nem tudja lekérni az adatokat az érzékelőtől, vagy esetleg nincs internet és nem tudja feltölteni az adatokat az adatbázisba akkor se romoljon el a program.

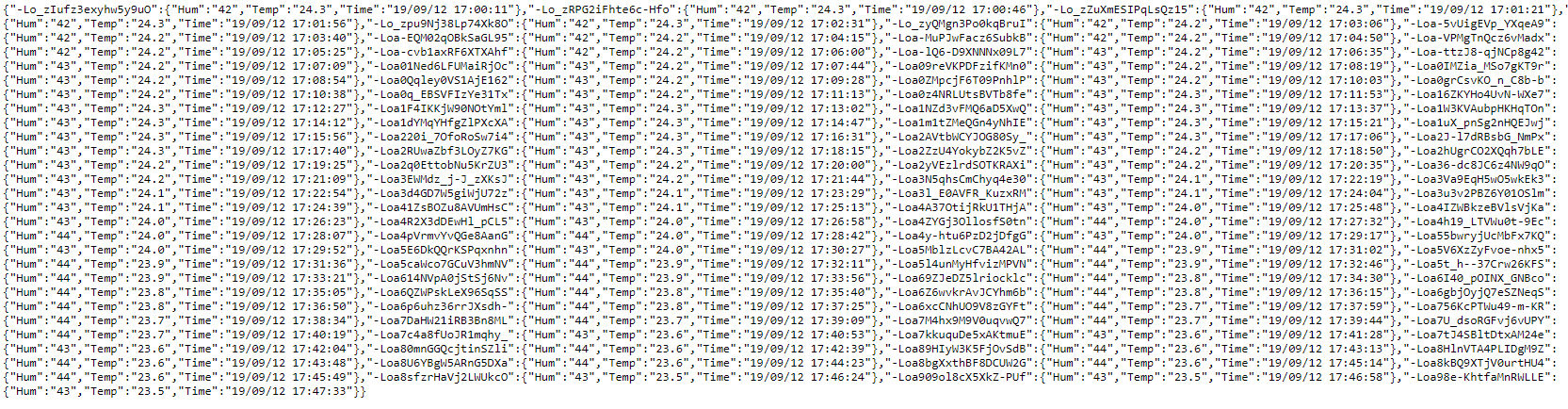
Az adatfeltöltő fájlt egy service segítségével futtattam le, hogy mindig menjen.

Így látszanak az adatok a FireBase-be ha be vagyok lépve.



Ezt kívülről nem lehet elérni, de a json fájlt mindenki le tudja kérdezni a következő linkkel: <https://tempdata-4be31.firebaseio.com/Adatmentes.json>

Ekkor egy ilyet kapunk:



Az adatokat feltöltés után óránként elemezzük. Számolunk az adatokkal, majd az így kapott adatokat egy új helyre mentjük. Először úgy terveztem, hogy a FireBase egy különálló FireStore Cloud-jába töltöm fel az elemzett adatokat, de sokszori sikertelen próbálkozás, egy tucat videó megnézése, és sok olvasás után feladtam ezt az ötletet, és inkább ugyan úgy a real time adatbázisba töltöm fel ezeket az adatokat is.

Miután az elemzett adatokat feltöltöttem, az eredeti adatbázis teljes tartalmát kitörlöm.

Az adatelemzésre is ugyan úgy python kódot használok, amit aztán egy cron segítségével minden óra nulladik percében meghív.

A gyakorlat befejezése

A gyakorlat utolsó napjaiban már nem tettem hozzá pluszt a feladatokhoz. A kódok átnézésével, hibakereséssel, szépítésével és az egész dokumentálásával telt. Ezen a dokumentáción kívül még egy 1 oldalas összefoglalót is el kellett készíteni az ELTE-Soft számára.

Összefoglalás:

A szakmai gyakorlat valamilyen szinten felkészített a későbbi munkánkra, hiszen megtanultuk, hogyan kezeljük a nehézségeket főként egyedül, vagy olykor felsőbb segítséggel. Önálló munkára tanított, mégis megtapasztalhattuk a csapatmunkát is a hallgatótársakkal együtt dolgozva.