Nyári gyakorlat első két hét beszámoló

Kezdetek (első hét hétfő - ~csütörtök)

Az első hét első napjai főként az előzőlegesen e-mail-ben kiküldött segédanyagok olvasgatásából állt. Bár elérhető volt már a trial licenszes SAP platform, erőforrások tekintetében a teremnyi használat önmagában közel használhatatlanná tette (ezen ügyben kaptunk egy saját szerverre ígéretet), emellett feladat híján nem sok ötletem volt, mit lehetne próbálgatni benne, így viszonylag hamar háttérbe szorult. Személy szerint engem főként a PAL és az adatbányászat érdekelt, azonban mivel nem tűnt úgy, hogy ezeket gyorsan módomban állna kipróbálni, és konkrét feladat se volt még, az elmélyedés helyett inkább több téren próbáltam felületesebb ismereteket szerezni, és az adattárházak fogalmaival/nodeJS-sel ismerkedtem, utóbbiból létrehoztam egy lokális projektet és – az előzőleges JS ismeretek alapján – próbáltam megérteni a működését.

Adatbázis (első hét ~csütörtök – második hét kedd)

Amikor végül megkaptuk a társaságok bejegyzéseit, és fel lett vázolva nagy vonalakban a feladat, először az adatok importálását és normalizálását vettem célul. Rövid tanulmányozás után létrehoztam egy relációsémát, amire szerintem a .csv-ben tárolt adatok illeszkedni fognak. Ezen a ponton sokaknál (köztük nálam is) akadtak nehézségek, ugyanis az ingyenes HANA SAP nehezen bírta a teremnyi hallgató build request-jeit, emellett problémát okoztak még a nem egyedi sémaelnevezések, illetve a legújabb database modul verzióval se tudtuk a sémát buildelni, és akiknek valahogyan mégis sikerült, a csv adat import nem volt elérhető. Ezek miatt nem túl sok eredménnyel telt el a csütörtök nagy része. A nehézségek miatt az eredeti csv-ből inkább egy json exportot csináltam, amit utána JS-sel dolgoztam volna fel, azonban ekörül vége lett az aznapi időnek.

Pénteken a hamarosan elérhető saját szerver ígérete miatt inkább az akkortájt 2. adag tutorial anyagot olvasgattam, mert ez a további haladás feltételének tűnt, azonban végül a nap végéig a szerver nem lett elérhető, úgyhogy sok produktuma nem volt a napnak.

Az elkövetkező héten hétfőn már elkészült a lokális szerver, így megtörténhetett a séma létrehozása és az adatok importálása. Ehhez több körben újra kellett csinálni a sémát, mert egyes társaságok nevei 500 karakternél is hosszabbak voltak. A mezőhosszok igazítása után is maradt egy sor, amely hibás volt, így az simán nem került felvitelre. Ezek után elkezdtem megtervezni, hogy hogyan kéne a táblákat szétszedni. A legelső, triviális felbontásban csak a név-kód mező párok lettek volna kivéve, az eredeti táblában csak a kódokat meghagyva, azonban az adatok tanulmányozása után úgy gondoltam, hogy az ország-régió-megye-település és nemgazd_ag-nemgazd_agazat-nemgazd_szakagazat mezőket hierarchikusan egymásra lehet rakni. Ekkor megkezdtem az alap adathalmazból az kód-név mezők külön táblákba exportálását `CREATE TABLE TBL AS SELECT DISTINCT `KÓD`, `NÉV` FROM `TÁBLA` ORDER BY `KÓD`; ALTER TABLE TBL ADD PRIMARY KEY(`KÓD`);` struktúrájú lekérdezésekkel. A legtöbbnél a kulcsok kiosztásakor hibába botlottam, mert több esetben is egy kódhoz több név is tartozott (rendszerint 1 karakter elgépeléséből). Ezeket `UPDATE` lekérdezésekkel kiküszöböltem, így az egyszerűbb kód-név táblákat sikerült kiemelni. Az előbb említett általam hierarchikusnak gondolt mezőkhöz érve az elképzelés az volt, hogy a legfelső szint (ország/nemgazd_ag) csak egy kód-név párból áll, kód kulccsal, míg az alatta levők kód-név-eggyel_fentebbi_szint_kód hármasokból így az eredeti táblában a legalsó szint kulcsából el lehetne jutni a legfelső szintig. Itt problémába ütköztem, ugyanis az ország-régió-megye-település tábláknál a régiók és megyék között is szerepelt ’egyéb’, amely egyebek viszont több

országhoz is tartoztak, így, amely települések több országhoz/régióhoz is tartoztak, nem voltak egyértelműen országig felvezethetőek csak az egyel feljebbi szint kódjának hozzáadásával. Ekkor gondolkodtam, hogy újratervezem a hierarchiát, esetleg egy 'HELY' tábla bevezetésével, amely ország-régió-megye-település kódnégyesekből állna, aminek a település már valid kulcsa, azonban ebben a megoldásban sem voltam teljesen biztos, hogy helyes lenne, és inkább elnapoltam a normalizálást, és a Bing Maps API használatát próbálgattam egy egyszerű html oldallal, amin a táblából szúrópróba-szerűen vett címekkel tesztelgettem. A címekből az API geokódoló paraméterezhető URL-jének query-ként megadva először long-lat koordinátákat szerezem, majd utána egy térképen pushpineket helyeztem el az így megszerzett pozíciókra. Ezek mellett a nap folyamán Béleczki tanár úr útmutatása mellett a szerveren levő homokozón létre lett hozva egy odata service, egyelőre csak a nyers adatokat közvetítve.

Kedden visszatértem a normalizáláshoz. Kezdetnek kaptam János csoporttársamtól egy valamennyi duplikátumtól/hibás bejegyzéstől mentesített .JSON file-t, ezt importáltam, majd a hierarchiát egyelőre elhagyva simán kód-név táblákra szedtem szét az adatokat a már korábban említett struktúrájú sql utasításokkal. A visszamaradt fő kapcsolótáblából első körben a teljesen megegyező sorokat küszöböltem ki egy `SELECT DISTINCT *…..` lekérdezéssel, kulcsának végül összetetten a `TARS_HOSZ_NEV, TARS_ROV_NEV, ASZ_EVE` mezőket adtam meg, ez alapján kezdetben ~16 megegyező kulcsú pár volt, ezeket egyesével, célzott lekérdezésekkel küszöböltem ki (a probléma az összesnél az volt, hogy a címeknél a budapesti irányítószám mellett még római számmal a kerület is meg volt jelenítve, a kerületet is tartalmazó címekhez tartozó sorokat töröltem). Ezen kívül még korábbi lekérdezések során szemet szúrt, hogy pár Budapesti teljes címmel rendelkező társaságnál településnek nem Budapest szerepel, ezeket a sorokat szintén töröltem, bár ez nem volt túl következetes részemről, mert más településekkel hasonló jellegű problémákat nem kerestem. A gyakorlatra kijelölt idő lejártával a nap hátralevő részében internetes segédanyagok felhasználásával próbáltam az általam megfelelőnek tartott táblákból egy calculation view-t készíteni. Kezdetben hibákba ütköztem, ugyanis az importált táblákat a rendszer build közben nem találta meg, mert a database artifact-ba ezek sémái nem lettek felvéve, hanem csak lekérdezésekkel lettek létrehozva. Minden relációhoz létrehoztam egy-egy entity-t, megadtam az egymáshoz való kapcsolatukat, a saját és foreign kulcsokat, majd az így kapott táblákba átmásoltam a korábbiak tartalmát. Így már az alábbi (https://data-flair.training/blogs/calculation-view-in-sap-hana/) tutorial alapján sikeresen összeraktam a calculation view-t, majd utána odata-ként elérhetővé tettem.

A hét hátralevő részében nem tudtam megjelenni a gyakorlaton, és a feladattal sem tudtam haladni.

Frontend (harmadik hét hétfő - kedd)

Egy olyan felületet terveztem elkészíteni, ami táblázatként pár oszlopnyi adatot ad meg soronként, illetve az adott sorhoz tartozik még egy gomb, amivel egy térképre a társasághoz tartozó címre elhelyez egy pint. Ehhez először rendelkezésemre állt még előző hét hétfőről a Bing Maps API tesztelős html oldal. Ebbe akartam importálni egy XMLHttpRequest-tel az odata-ból elérhető JSON-t, de ekkor problémába ütköztem, ugyanis nem volt az elérhető más hostról (CORS request denied). Pár órányi keresés után se sikerült ezt megoldani, úgyhogy a már adat JSON forrásként megnevezett Jánoshoz fordultam, mivel előző hét kedden ő nodeJS-ből küzdött hasonló problémával. Neki volt egy megoldása nodeJS-re és Angular keretrendszerre is, előbbi (saját felfogásom szerint) biztonsági funkciók kikapcsolásával, utóbbi proxy-val. Végül az utóbbi mellett döntöttem. E-mail-ben keringett egy kiindulási alapnak megfelelő mintaprojekt, ezt próbáltam először működésre bírni, azonban értés nélkül kaotikus volt, indításra a lokális szerver hibákat dobott, és végül inkább az Angular dokumentációjának Getting Started részével nulláról kezdtem egy saját projektet. Így, hogy

valamennyi fogalmam már volt a keretrendszer struktúrájáról beimportáltam a korábbi mintaprojektből az odata service-t és a proxy-t. Lekérdeztem az első 100 bejegyzést, mindegyik után generáltam egy táblázatbeli sort, bennük a TARS_ROV_NEV, CÍM_EGYBEN, ASZ_EVE sorokat megjelenítve, illetve soronként egy extra gombbal egy térképre való pin hozzáadáshoz.

Kedden miután ezek működtek, próbáltam implementálni egy Bing Map-et. Ezzel a vártnál sokkal több nehézség volt, ugyanis míg html felületen ez kb 2 script és a hozzájuk tartozó div elhelyezéséből áll, Angularban fogalmam se volt, hogy mit hol kell elhelyezni, hogy utólag a mapról legyen referenciám a későbbi pinek felhelyezéséhez. Végül találtam egy megoldást, ami ígéretesnek tűnt, implementálva meg is jelenítette a térképet, egy furcsa viselkedést váltott ki. Először is, az oldal első indítására build failed volt, az oldal nem jelent meg, azonban (az `ng serve` paranccsal futtatott szerver érzékeli a forrásfájlok változásait, ekkor újrabuildeli a projektet) mindenféle változtatás nélkül (pl. szóköz hozzáírás majd törlés, mentés) minden továbbira a hibaüzenetek megmaradtak, azonban az oldal megjelent, és látszólag rendesen működött is, erre a jelenségre nem sikerült magyarázatot találni. Odáig nem jutottam el, hogy megpróbáljak pint elhelyezni rá, nem akartam a látszólag hibás megoldásra építeni, ezért inkább a Bing Maps API-ból elérhető statikus, képként megjelenített térképre hagyatkoztam, ami scriptek helyett csak egy <img> tag src attribútuma. Létrehoztam GeocoderService néven egy service-t, ami az egyik függvényével query-ből long-lat koordinátákat állít elő, illetve a másikkal long-lat koordinátákból a koordináták középpontú, ott egy pinnel rendelkező API által előállított kép url-jét adja vissza. Ekkor abba a hibába ütköztem, hogy a long-lat koordinátás résznél a HttpRequest eredménye undefined, de csak első híváskor. Ezt végül a függvény aszinkron megvalósításával sikerült kiküszöbölni (előbb próbáltunk olvasni az objektumból, semminthogy az betöltődött volna, ami az itt csak egyszer, init-en lekért odata-nál nem volt problémás, de a gombnyomásra lekért adat első hívásánál igen). Végül az 'Ismeretlen' egybecímmel rendelkező társaságok miatt bekerült egy ellenőrzés, hogy kezeljük az eseteket, amikor nem kapunk vissza koordinátákat.