

TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Tarkvaratehnika õppekava

Irma Sauga

Andmeait Tartu Ülikooli Aruandlussüsteemi näitel

Magistritöö (30 EAP)

Juhendaja: Vambola Leping

TARTU 2014

Andmeait Tartu Ülikooli Aruandlussüsteemi näitel

Lühikokkuvõte:

Antud magistritöö eesmärk on anda lugejale sisuline ülevaade andmeaitadest ja sellega seonduvast – terminoloogia, arhitektuur, protsessid. Oluline valdkond on andmeaida rutiinne tööprotsess – info eraldamine andmeallikatest, töötlemine ja laadimine andmeaita. Teise teemana kaetakse andmeaida arhitektuuri ja modelleerimisega seonduv. Töö praktilise osana disainitakse ja luuakse prototüüp Tartu Ülikooli Aruandlussüsteemi jaoks, mis võimaldab hakata koostama aruandeid üliõpilaste õppetulemuste kohta.

Võtmesõnad: andmeait, andmevakk, arendusprotsess, Aruandlussüsteem

Data Warehouse as explained by Tartu University's Reporting System

Abstract:

This master's thesis gives a broad overview of data warehouses and its terminology, architecture and processes. Another important subject is design workflow and its subprocesses. To give an example of theory, a prototype datamart is designed and built for Tartu University's Reporting System to provide reports about students academic performance.

Keywords: data warehouse, data mart, design process, Reporting System

Sisukord

Sissejuhatus	5
1 Andmeait kui analüüsivahend	6
1.1 Andmeait - tähendus	6
1.2 Andmeait - ajalugu ja tänapäev	7
1.3 Andmeait - ülesehitus	8
1.3.1 Operatiivsüsteem	9
1.3.2 Andmete teisenduskiht	9
1.3.3 Andmeait ja andmevakad	10
1.3.4 Metaandmed ja operatiivne andmehoidla	12
1.3.5 Klientrakendused	13
2 Andmeaida arendusprotsess	15
2.1 Võtmeteemad	15
2.2 Andmeaida kasutuselevõtu otstarbekuse hindamine	16
2.2.1 Tugevad toetajad juhtivalt poolelt	16
2.2.2 Kaalukas äriiline motivatsioon	17
2.2.3 IT ja äri partnerlus	17
2.2.4 Olemasolev analüütiline kultuur	17
2.2.5 Teostatavus	17
2.3 Andmeaida arendustsükkel	18
2.4 Andmeaida teisenduskihi protsessid ja nende väljatöötamine	20
3 Andmevakk Aruandlussüsteemi jaoks - prototüüp	22
3.1 Idee	22
3.2 ÕIS	23
3.3 Aruandlussüsteem	24
3.4 Lahendus	26

3.4.1	Disainiprotsess	26
3.4.2	Prototüüp	27
3.4.3	Ülevaade kasutatud töövahenditest	30
3.4.4	Andmemudel	30
3.4.5	Abilahendused	32
3.4.6	Ajahaldus andmete laadimisel tabelitesse	32
3.5	Tööprotsess	33
3.5.1	Tabelite loomine	33
3.5.2	Eel-laadimisala täitmine	34
3.5.3	Dimensioonitabelite täitmine	34
3.5.4	Faktitabeli täitmine	34
3.5.5	Lõpptegevused andmeaidas	34
3.5.6	Prototüübi koostamine	34
	Kokkuvõte	37
	Viited	38
	Lisad	41
	Lisa 1. Disainidokument „Üliõpilaste edetabelile“	41
	Lisa 2. Andmevaka disain	45
	Lisa 3. Prototüübiga seotud SQL laused	64
	Litsents	75

Sissejuhatus

Igas infosüsteemis tekib säilitatavaid andmeid ning aja möödudes nende hulk suureneb ning hallatavus väheneb. Infosüsteemi kasvades võib lisanduda allüksusi, milledes kasutatakse hoiustamiseks üksteisest erinevaid andmestruktuure. Kui nüüd soovida koostada üle kogu süsteemi aruandeid ning analüütilisi päringuid, on abiks andmeait – keskne andmekogu, mis pakub võimalust koguda allüksustest perioodiliselt infot, töödelda seda vastavalt ühtsele standardile, hoiustada ning luua kogutu põhjal soovitud aruandeid.

Antud magistritöö eesmärk on anda lugejale sisuline ülevaade andmeaitadest, sellega seonduvast (terminoloogia, arhitektuur, protsessid) ning läbi mitmete arendusetappide luua prototüüp üliõpilaste õppeedukuse põhjal aruannete koostamiseks. Nimetatud arendusetappides kavandati esmalt täiendus olemasolevale Aruandlussüsteemile, loodi andmemudel ning tehniline disain ja viimaks pandi kokku prototüüp.

Esimeses peatükis selgitatakse lugejale, mis on andmeait, mis on selle otstarve ning tutvustatakse ajalugu, arhitektuuri ja seonduvaid protsesse. Teises peatükis tuuakse esmalt välja võtmetegurid, mille järgi hinnata mingi arendusprojekti valmisolekut andmelao kasutuselevõtuks ning seejärel esitatakse lühiülevaade andmeaia arendustsüklist.

Kolmandas peatükis kirjeldatakse magistritöö praktilise osana Tartu Ülikooli Aruandlussüsteemi jaoks valminud prototüüpi, mis võimaldab hakata koostama aruandeid üliõpilaste õppetulemuste kohta. Praegusel hetkel kogutakse ja hoitakse kogu üliõpilastega seonduv info Õppeinfosüsteemi (edaspidi ÕIS) andmebaasis, kuid selles puuduvad põhjalikumad statistika ja andmekaeve võimalused õppetulemuste hindamiseks. Seevastu Aruandlussüsteem on Tartu Ülikooli andmeladu, milles töödeldakse erinevatest Ülikooli andmeallikatest pärinevat infot (kaasa arvatud ÕIS), kuid mille õppijaid puudutavad aruanded piirduvad tudengite arvu statistikaga. Mõlemas süsteemis puudub praegu võimalus demonstreerida üliõpilaste edukust ning seetõttu pakub antud magistritöö välja lahenduse taolise funktsionaalsuse jaoks, disainides kasutajaliidese, andmemudeli ning vastavad aruanded.

Lisadena on magistritöös esitatud prototüübi valmimise käigus loodud kasutajaliidese disainidokument, mis sisaldab kasutajalugusid, tehnilise disaini dokument, mis sisaldab andmemudelit ning andmeaia tabelite struktuuri, ning nimekiri töös kasutatud SQL lausetest.

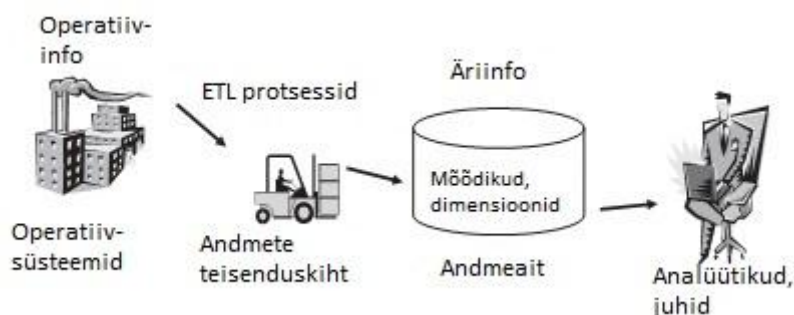
1 Andmeait kui analüüsivahend

Andmeait kui analüüsivahend peatükis selgitatakse lugejale, mis on andmeait, mis on selle otstarve ning tutvustatakse ajalugu, arhitektuuri ja seonduvaid protsesse.

1.1 Andmeait - tähendus

Iga ettevõtte või organisatsioon omab edukaks tegutsemiseks andmebaasisüsteemi, milles hoitakse igapäevatööks vajalikke andmeid. Algushetkel on andmemudel lihtsalt loetav ning sisaldab ehk paarkümmend tabelit, kuid ettevõtte kasvades andmemahud suurenevad ning suureneb ka osapoolte arv, kes seda kasutavad. Ettevõtete ühinemisel võib tekkida olukord, kus operatiivandmed paiknevad erinevatel tehnoloogilisel platvormidel. Kui soovida nüüd koostada sellises keerukas süsteemis ettevõtte käekäiku analüüsivaid päringuid, satutakse takistuste ette – iga aruande loomine on ühekordselt kasutatav aeganõudev protsess, mille käigus tuleb läbi töötada mitmeid andmebaase ning mille lõpptulemuse väärtust kahandavad pidevalt muutuvad andmed. Et probleemi lahendada, on võimalik kasutusele võtta andmeait.

Andmeait on üldistatult ettevõtte või organisatsiooni eraldiseisev andmebaasisüsteem, millesse regulaarselt lisatakse töödeldud operatiivandmeid ning milles nende andmete põhjal on võimalik koostada aruandeid. Andmeait ise uut informatsiooni nõ. juurde ei tooda, vaid selles sisalduv pärineb ja põhineb kõik kasutusel olevatel operatiivsüsteemidel (joonis 1.1.1).



Joonis 1.1.1: Andmeaida osalus protsessis [1].

Andmeaitadele omapärane kiirus ja efektiivsus põhinevad üldisel protsessil – esimeses etapis kogutakse erinevatest allikatest info kokku, seejärel andmete teisenduskihis („*staging area*“) töödeldakse ning kolmandas etapis laetakse andmeaita. Säilitatavad andmed on

puhastatud, kooskõlalised ning organiseeritud kergesti kättesaadavaks. Et andmeait tsentraliseerib kogu info, piisab kasutajatele võimalusest ligi pääseda ainult ühele süsteemile ning kaob ära vajadus hallata keerulisemat ja ebaturvalisemat kasutajaõiguste lahendust, mis tagas eraldi juurdepääsu igale operatiivandmebaasile. Võrreldes operatiivsüsteemidega on andmeaidas lihtne ka prognooside koostamine tunnustatud statistiliste meetoditega – ajaloolise info olemasolu aitab sellele kaasa.

Andmeaidale rakenduvad spetsiifilised reeglid [2, 3]:

1. ajateisendlik – andmeait sisaldab pika perioodi jooksul kogutud infot nõnda, et iga olemasolevat kirjet on võimalik seostada selle lisamisajaga ning võimalik on liikuda „ajas tagasi“;
2. püsiv – andmeaita juba lisatud andmed on lõplikud ning ei muutu;
3. teemakeskne – andmeait on ehitatud analüüsima kindlat operatiivsüsteemide valdkonda, mitte aga tegevust (näiteks müüki, mitte aga tellimuste vastuvõttu);
4. integreeritud – andmed on ühilduvad, samas vormingus ja ühetähenduslikud.

Lisaks terminile „andmeait“ on kohati kasutusel ka väljend „andmeladu“ – eestikeelse kirjanduse eeskujul käsitletakse selles töös neid sünonüümidena.

1.2 Andmeait - ajalugu ja tänapäev

Eestis on andmeaidad kasutusel juba vähemalt eelmise aastakümne algusest, tegemist ei ole kindlasti haruldase nähtusega. Mitmed ettevõtted – Adepta Solutions OÜ [4], OÜ Resta [5] – on spetsialiseerunud nende arendamisele. Andmeaidade lahendused on kasutusel näiteks sellistes suuremates ettevõtetes ning riigiasutustes nagu AS Tallinna Vesi [6], AS Estonian Air [7], Tere AS [8], Politseiamet [9] ja Eesti Post [10].

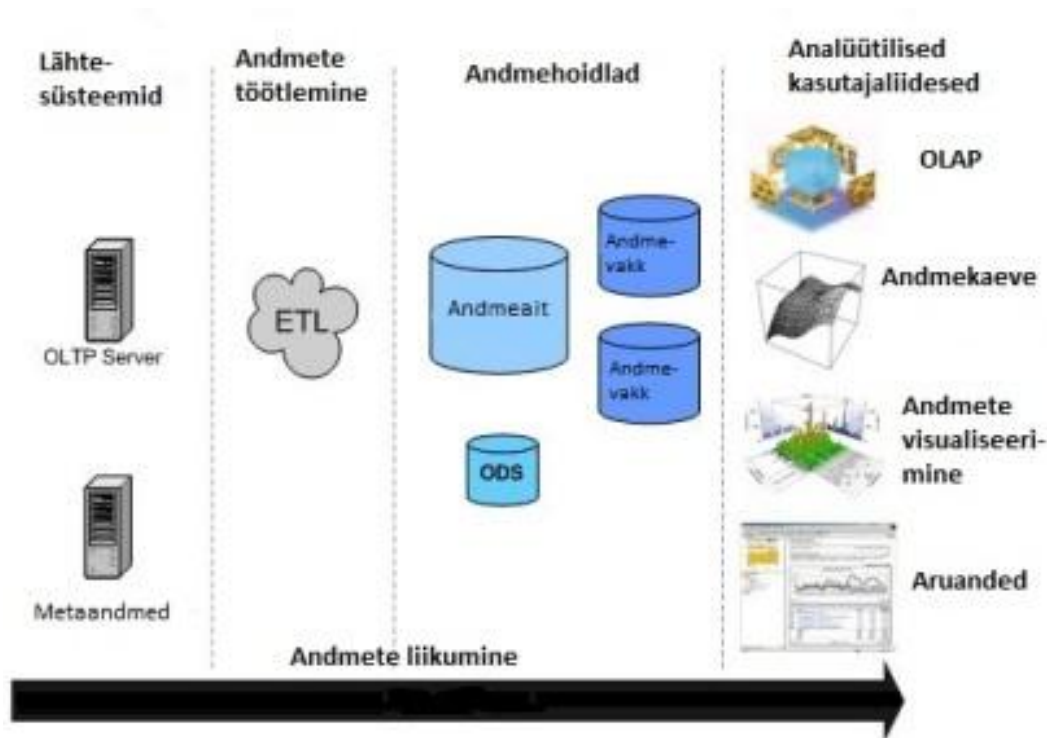
Kui Eestis on selles valdkonnas tegutsenud suhteliselt lühiajaliselt – ligikaudu viisteist aastat – siis mujal maailmas, eriti USA Silicon Valleys, tuldi esimeste sellisuunaliste ideedega välja juba 1970. aastatel, kui Bill Inmon võttis kasutusele termini „andmeladu“. 1980. aastatel andsid IBM-i teadlased välja teadusartikli äriandmeladude teemal ning 1990. aastate algusest hakati aktiivselt arendama spetsiaaltarkvara [11]. Esmalt oli tegemist kohmakate süsteemidega, mis vajasisid palju ressursse, olid andmete töötlemisel aeglased ning kus andmeid laeti juurde reeglina kord nädalas. Aja möödudes tuldi välja efektiivsemate

tehnoloogiliste lahendustega, mis muutsid andmete lisamise ja kasutamise ligilähedaseks reaalajaliste protsessidega.

Praegusel kümnendil on andmelaondus kaasa läinud populaarsemate tehnoloogiatega – „*big data*“ (suured andmed), pilvetehnoloogia jne – konsolideerimine pilve pakub näiteks võimalust maksimeerida paindlikkust ning vähendada kulusid. Globaliseerumisega on esile tõusnud vajadus pääseda ligi andmevahenditele ükskõik kus – internet on muutunud lahutamatuks osaks andmelaost. 2014. aastaks ennustatakse andmelaonduses olulisemateks trendideks suurte andmehulkade protsessimise lihtsustumist tänu Hadoopile [12], jõudluse suurenemist tänu sisemälu-kesksema tehnoloogia („*in-memory technologies*“) kasutuselevõtule ning täis-komplekse riist- ja tarkvaralahenduste (näiteks Oracle Big Data Appliance [13]) laiemat levikut [14].

1.3 Andmeait - ülesehitus

Kuigi termin „andmeait“ tähistab otsesõnu ainult andmebaasi, milles hoitakse analüüside ja aruannete tarbeks andmeid, räägitakse sellega seoses alati ka ümbritsevast infrastruktuurist. Viimase võib laias laastus jaotada neljaks osaks – operatiivsüsteem lähteandmetega, andmete teisenduskiht, andmeait ning klientrakendused analüüside jaoks [3] (joonis 1.3.1).



Joonis 1.3.1: Andmeaida infrastruktuur [15].

1.3.1 Operatiivsüsteem

Operatiiv- ehk OLTP-süsteemid („*OnLine Transactional Processing database*“) toetavad igapäevaseid äriprotsesse, seetõttu on rõhk asetatud väga kiirele päringute töötlemisele (efektiivsust mõõdetakse päringute arvuga sekundis). Kõrge normaliseerimise tulemusena on andmebaasides palju tabeleid ning neis hoitakse ainult andmete „jooksvat“ seisut. Sõltuvalt ettevõttest on võimalik, et ühe andmelao operatiivsüsteemid kasutavad erinevaid tarkvarasid või operatsioonisüsteeme – näiteks raamatupidamine toetub andmebaasitarkvarale MySQL [16], laosüsteem on ülesehitatud aga Oracle [17] peale [1, 18].

Lisaks OLTP-süsteemidele võivad lähteandmed pärineda ka arhiivifailidest või välistest allikatest – esimene võimaldab paremat ülevaadet minevikust ning teine võimaldab kõrvutada ennast ülejäänud tööstusharu tulemustega.

1.3.2 Andmete teisenduskiht

Andmete teisenduskiht on oluline vahelüli operatiivsüsteemide ja andmeaida vahel. Info transportimiseks on kihis kirjeldatud 3 peamist ülesannet:

1. ekstraktimine
2. töötlemine
3. laadimine

Andmete ekstraktimisel („*extract*“) eraldatakse operatiivsüsteemidest andmeid ning konverteeritakse need ühtsesse, töötlemiseks sobivasse, formaati. Ühe ettevõtte piires võivad vajalikud andmed asuda erinevates süsteemides ning mitmesugusel erineval kujul.

Andmete töötlemisel („*transform*“) korrastatakse eelmisel sammul eraldatud andmed laadimiseks sobivale kujule. Esmalt teostatakse puhastamine, mille käigus eemaldatakse nii trükivead ja duplikaadid kui ka lisatakse puuduolevad vaikeväärtused. Standardiseerimisega ühtlustatakse erinevatest allikatest pärinev info ning tagatakse, et konkreetne definitsioon oleks ühene üle kogu süsteemi. Jälgida tuleb näiteks seda, et kui ühes süsteemis on sugude tähistamiseks kasutatud „M“ ja „N“ vormi ning teises „mees“ ja „naine“ varianti, siis tuleks ühele andmeallikatest rakendada teisendamist. Lisaks puhastusfunktsioonidele on selles etapis võimalik rakendada agregeerimist – andmete summeerimist – kindlaksmääratud parameetrite abil, näiteks kuu, päeva ja aasta kaupa [19]. Nõnda on võimalik vähendada märgatavalt aega, mis kulub hiljem lõppkasutajal madala detailsusega suurte andmehulkade pärimiseks.

Andmete laadimisel („*load*“) liigutatakse töödeldud andmed aita. Siin on võimalik eristada kahte tüüpi laadimisi – esmane ja korduv. Esimesel korral on transporditav andmehulk suur ning võtab palju aega. Edaspidiselt on tegu rutiinse protsessiga ning kogused ja ajakulu vähenevad. Sõltuvalt disainist võivad korduvad laadimised toimuda kord päevas, kuus või kvartalis.

Protsessi lihtsustamiseks ja automatiseerimiseks pakuvad tarkvaratootjad erinevaid vahendeid, populaarsemad on näiteks IBM InfoSphere Information Server [20], Informatica PowerCenter [21] ja SAP BusinessObjects Data Integrator [22].

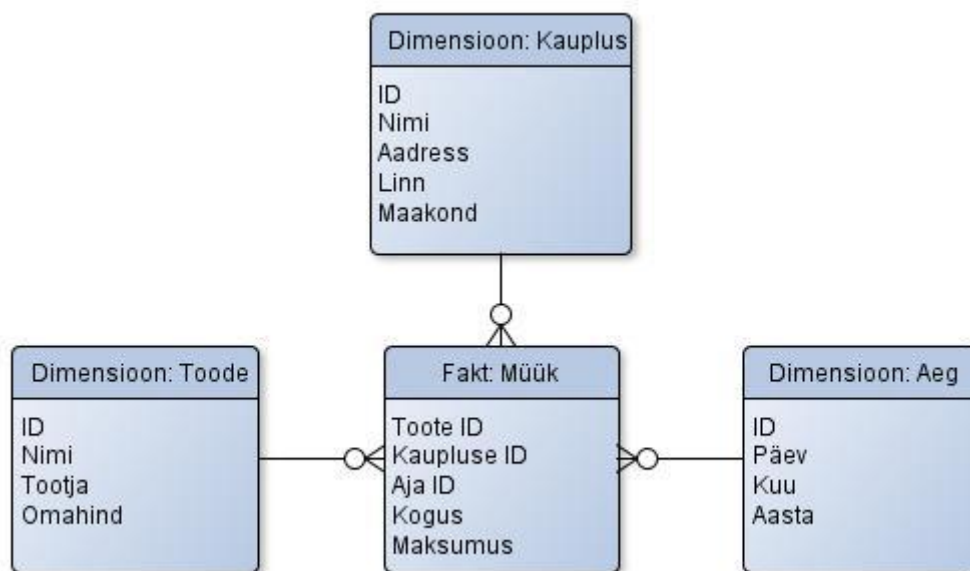
1.3.3 Andmeait ja andmevakad

Andmeaitade valdkonnas eristatakse kahte tüüpi lähenemist disainis: "ülevalt alla" („*top-down*“) ja „alt üles“ („*bottom-up*“). Neist esimene pärineb Bill Inmonilt ning sõnastab, et andmeait on keskne hoidla kogu ettevõttele, kus andmeid säilitatakse kõige madalamal detailsusastmel ning normaliseeritud kujul. Sõltuvad andmevakad („*dependent data mart*“)

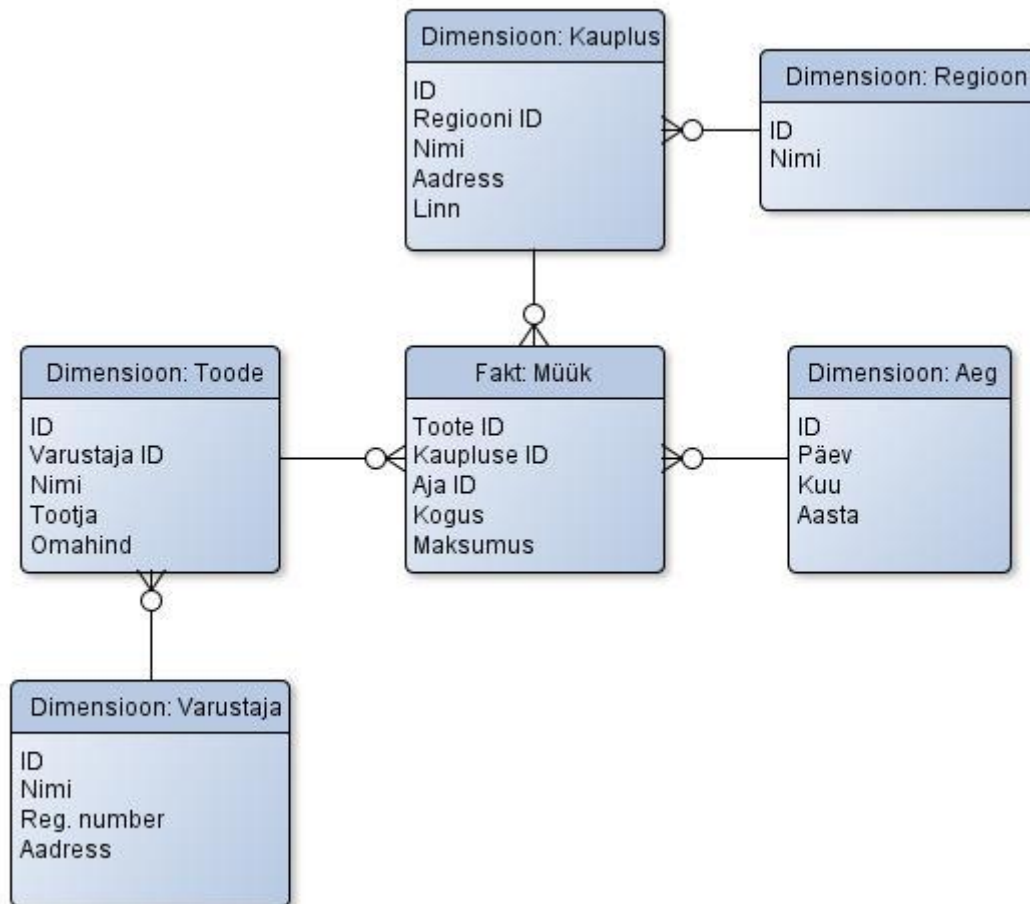
saavad oma lähteandmed andmeaidast, denormaliseerimine ja summeerimine toimub andmevaka tasemel.

Teise definitsiooni autor on Ralph Kimball ning see sõnastab, et andmeait ei ole midagi rohkemat kui erinevate andmevakkade (sõltumatud andmevakad) kogum. Selle põhimõtte järgi alustatakse arendust üksiku andmevaka loomisest, mis rahuldab mingi konkreetse osakonna või äriprotsessi vajaduse. Andmevakk kirjeldab üht ettevõtte konkreetset äriprotsessi ning koosneb dimensioonidest ja faktidest. Dimensioon sisaldab kõiki ühe subjekti (näiteks toode, isik) kohta olemasolevaid andmeid, faktid võivad olla sõltuvalt vajadusest nii detailsed kui ka summeritud.

Andmevakas on tabelid denormaliseeritud ning omavahel seostatud kas täht- (joonis 1.3.3.1) või lumehelbeskeemi (joonis 1.3.3.2) alusel.



Joonis 1.3.2.1: Tähtskeem.



Joonis 1.3.3.2: Lumehelbeskeem.

Tähtskeemi korral on keskmeks faktitabel ning sellest kiirguvad välja dimensioonitabelid. Reeglina on tähtskeemis faktitabel kolmandal normaalkujul ning dimensioonitabeleid on denormaliseeritud. Päringute loetavus tuleneb skeemi lihtsast ülesehitusest ning kiirus läbi efektiivsuse, mida pakub vajadus ühendada omavahel SQL päringutes ainult mõned tabelid. Lumehelbeskeem on tähtskeemi variatsioon, kus dimensioonitabeli kirjade ja andmemahtude vähendamise nimel on see viidud normaalkujule.

1.3.4 Metaandmed ja operatiivne andmehoidla

Andmeaida oluline osa on metaandmed – andmed andmete kohta. Kõige paremini saaks seda kirjeldada kui sõnastikku või kataloogi, mis aitab hallata kõiki komponente ning aitab lõppkasutajal mõista sisulist poolt. Metaandmeid on olemas mitut erinevat tüüpi ning sõltuvalt autorist ja rakendajast võidakse neid liigitada ja grupeerida nii kasutamise kui ka kasutajatüüpide järgi.

Ralph Kimball liigitab metaandmeid kolme osaliselt kattuvasse kategooriasse [23]:

1. tehnilised andmed kirjeldavad süsteemis leiduvaid objekte ja protsesse tehnilisest vaatepunktist, näiteks andmestruktuuri kaudu tabelleid, indekseid ja dimensioone;
2. ärilised andmed kirjeldavad süsteemi kasutajasõbralike terminitega ning on ennekõike suunatud lõppkasutajatele;
3. protsessiandmed kirjeldavad erinevate andmeaidas läbiviidavate operatsioonide tulemusi.

Metaandmete olemasolu on kriitiline nii andmeaida kasutamiseks, ehitamiseks kui ka administreerimiseks – ilma selleta on keeruline või lausa võimatu mõista andmeaida sisu.

Operatiivse andmehoidla ODS („*operational data store*“) on osa andmeaidast, milles toimub kohene analüüs kehtivate andmetega. Erinevalt andmevakkadest saab see andmehoidla oma sisu lähteandmetest või teisenduskihist, aga mitte aidast.

1.3.5 Klientrakendused

Lõppkasutajatele on andmeaitades ja –vakkades sisalduva info kättesaamiseks olemas mitmeid erinevaid võimalusi – OLAP, andmekaeve, aruandlus, andmete visualiseerimine. Sõltuvalt kasutajast pakuvad need rohkem või vähem kõrgtaseme ligipääsu ning võimalust iseseisvalt andmeaida sisu põhjal analüüsi koostada.

OLAP („OnLine Analytical Processing“) on reaalajas analüütiline töötlemise viis, mille abil on võimalik hetke- ja ajalooliste andmete põhjal koostada prognoose ning trendianalüüsi. Tehnoloogiliselt võimaldavad selle multidimensionaalne andmemudel ning andmete summeerimise tehnikad analüüsida kiirelt suuri andmehulki. OLAP ise jaotub kolme erinevasse kategooriasse:

- mitmemõtteline OLAP ehk MOLAP on tavapäraseim vorm, kus andmed on mitmedimensioonilistes vaadetes;
- relatsiooniline OLAP ehk ROLAP põhineb relatsioonilisel andmebaasil;
- hübriid OLAP ehk HOLAP on kompromiss kahe eelneva lahenduse vahel, kus summeeritud andmete jaoks kasutatakse MOLAP lähenemist, aga detailide saamiseks pööratakse tagasi relatsioonilise andmebaasi poole.

Andmekaeve abil on võimalik avastada andmetes erinevaid mustreid, peamiselt kasutatakse seda „peidetud“ seoste otsimiseks ning prognooside koostamiseks.

Aruandlust toetavad töövahendid võimaldavad kontrollida andmete korrektsust seoses äriiga ning jälgida võtmeindikaatoreid („*key performance indicator*“).

Andmete visualiseerimise vahendid on tihti kombineeritud nii OLAP kui ka andmekaevega. Ennekõike on eesmärgiks kuvada kasutajale andmeaidas sisalduvat võimalikult arusaadavalt – joonistatud kujul tulevad mustrid ja seosed paremini esile.

2 Andmeida arendusprotsess

Reeglina ei arvesta ükski alustav ettevõtte oma arhitektuuri koostamisel andmeidaga, võimalik, et nad isegi ei tea sellest midagi. Aja möödudes ettevõtte kasvab, andmemahud suurenevad ning analüüsid muutuvad keerulisemaks – kõik see nõuab järjest läbimõeldumaid ja efektiivsemaid lahendusi. Algsest andmebaasilahendusest enam ei piisa, hakatakse vaatama järjest kaugemale – andmeait võiks olla järgmine samm. Järgneva peatüki eesmärk on esmalt pakkuda kriteeriume, mille suhtes hinnata valmisolekut andmelao kasutuselevõtuks ning seejärel pakkuda ülevaadet andmeida arendustsüklist.

2.1 Võtmeteemad

Üks esimesi samme andmeida projektis on läbi mõelda nõ. võtmeteemad – sisulised ja asjakohased valdkonnad, mis mängivad projektiplaneerimises olulist rolli. Raamatus „*Data Warehousing Fundamentals for IT Professionals*“ [1] on neid välja toodud 5 – ootused andmelaole, riskianalüüs, disainiline lähenemine, töövahendite päritolu ning tarnijate arvukus.

Ootused andmelaole on üheselt seotud tema eeldatav tulevase väärtusega – ettevõtte jaoks on oluline, et tehtav investeering tasuks ennast ära. Siin tuleks mõelda eesmärkidest lähtudes, kas andmeladu on antud ettevõtte jaoks kõige sobivam lahendus ning seejärel selgitada välja, missugust kasu suudaks see pakkuda.

Riskianalüüsi eesmärk on ennekõike hinnata projekti maksumust ning selgitada välja võimalikud kaod ebaõnnestumise korral. Samas ei tohi jätta tähelepanuta ka riske, mis tekivad, kui andmeladu ei võeta kasutusele – mida võidakse kaotada, millisest võimalustest ilma jääda jne.

Disainiline lähenemine, millest oli lähemalt juttu peatükis 1.3.3, tähistab vajadust mõelda, millisel viisil soovitakse andmeladu üles ehitada – kas alustada kõikehõlmavast kesksest andmelaost, üksikust andmevakast või lähtuda nende hübriidist. „Alt üles“ lähenemisega vajab ettevõtte piisavalt ressursi, et disainida ja ehitada kogu ettevõtet kattev andmeladu, ning mahukuse tõttu võtab realiseerimine kaua aega, kuid samas tagatakse nõnda ühtne ülevaade kõigist andmetest. „Ülevalt alla“ lähenemise eelis on aga kiirus, millega on võimalik

esimesed andmevabad disainitud ja toimima saada, kuid selle arvelt on ühtlust keerulisem saavutada.

Töövahendite päritolu küsimuses on oluline selgitada välja, kui suurt osa protsessidest ja tarkvarast on võimalik ja tasuv katta väliste lahendustega ning mida tuleks ise ehitada. Andmete teisenduskihi funktsioonid on hea näide – võimalik on vajalik funktsionaalsus ehitada nii ettevõttesiseselt kui kasutada ka kolmanda osapoole lahendusi. Ühelt poolt säilib iseehitamisega kontroll kogu toimimise üle, samas võib väline tarkvara aidata kaasa kiirele teostamisele.

Tarnijate arvukus kajastub küsimuses, kas kasutada ühe või mitme erineva tarkvaraettevõtte tooteid. Kuigi ühe pakkuja täislahenduse eelisteks on komponentide sujuv integratsioon, ühtlus ja keskne haldus, on probleemiks selliste lahenduste vähesus. Teisalt on erinevate tarnijate kasutamise eeliseks võimalus panna kokku just konkreetsele ettevõttele sobiv komplekt ning valida mingi funktsiooni jaoks parim võimalik tarkvara. Samas tuleb sel juhul jälgida, et valitud tooted kindlasti omavahel ka kokku sobiks.

2.2 Andmeaida kasutuselevõtu otstarbekuse hindamine

Andmeait võib tunduda kui lahendus kõigile probleemidele, kuid ei tohi jätta arvestamata, et tegemist on siiski suuremahulise arendusprojektiga. Kui alustada seda ilma hindamiseta, võib juhtuda, et lõpptulemus ei vasta ootustele või halvemal juhul ei jõua projekt valmimiseni.

Ralph Kimball sõnastab oma raamatus „*The Data Warehouse Lifecycle Toolkit*“ [23] 5 peamist võtmetegurit – tugevad toetajad juhtivalt poolelt, kaalukas äriiline motivatsioon, IT ja äri partnerlus, olemasolev analüütiline kultuur ning teostatavus - mida kasutada, kui on soov hinnata organisatsiooni valmisolekut andmeaida kasutuselevõtuks. Neid faktoreid võib võtta kui vundamenti, millele toetub edukas andmeaida projekt. Kui tundub, et mõnel kriteeriumil puudub kate, tasub projekti jätkamisse suhtuda ettevaatlikumalt ning leida puuduv toetus.

2.2.1 Tugevad toetajad juhtivalt poolelt

Üks olulisemaid tegureid õnnestumises on tugev ja sihikindel juhtfiguur, kellel on organisatsioonis ka mõjuvõimu, et projekti läbi suruda ning sellele arendustegevuses

prioriteete saada. Temal (või neil) peab olema kindel visioon lõpptulemusest ning selle mõjust ülejäänud ettevõttele, samas aga peab säilima realistlik arusaam, et tulla toime lühiajaliste tagasilöökidega.

2.2.2 Kaalukas äriline motivatsioon

Üks sage ühine näitaja õnnestunud andmeaida projektides on tugev motivaator äriliselt poolelt. Selleks võivad olla nii konkurendid, sisemised kriisid kui ka nägemus võimalusest turul läbi lüüa. Sobivaks kandidaadiks on ka klientidele orienteeritud organisatsioonid – andmeait võimaldab analüüsida käitumis- ja tarbimismustreid ning selle põhjal pakkuda sobilikumaid teenuseid. Kui andmeait põhineb või toetub mõnele märkimisväärsele ärilisele motivatsioonile või huvile, toetab see projekti õnnestumist ning võimaldab kergemini leida ressursse.

2.2.3 IT ja äri partnerlus

Selleks, et andmeaida projekt õnnestuks, on vaja, et IT ja äriline pool töötaksid käsi-käes. Ilma äriteadmisteta ei paku andmeaida projekt piisavat tuge protsessidele ning ilma IT poolse panuseta on keeruline saada tööle andmeaidaga kaasnevaid tehnilisi protsesse.

2.2.4 Olemasolev analüütiline kultuur

Andmeait võimaldab analüüsida ja aruannete kaudu toetada äriliste otsuste tegemist. Kui organisatsioon lähtub juba praegu faktipõhistest analüüsides, on andmeait sobilik täiendus ning rakendatakse koheselt kasutajate poolt töösse. Vastasel juhul tuleb arvestada, et lisatööna on vaja muuta ka organisatsiooni sisekultuuri ning mõtteviisi.

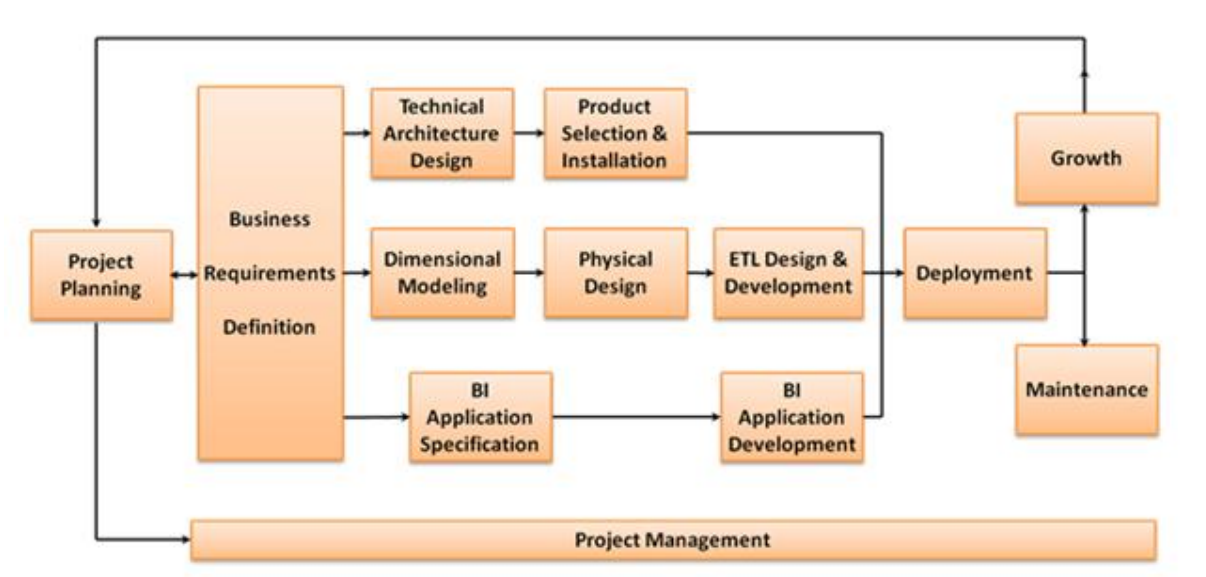
2.2.5 Teostatavus

Andmeaida planeerimisel tuleb tähelepanu pöörata lähteandmetele – kas ja kuidas on need kättesaadavad. Probleemseks võivad osutuda nii igasuguse töötlemiseta, esmaseks arenduseks liigselt keerukas süsteemis paiknevad või halvimal juhul isegi kogumata andmed. Teisalt võib

pidurdavaks teguriks lugeda ka seda, kui üle kogu organisatsiooni ei ole määratletud ühiseid ärireegleid ja –definiitsioone.

2.3 Andmeaida arendustsükkel

Andmeaida projektid on suuremas ulatuses sarnased teistele tarkvaraarendusprojektidele. Erinevused tekivad nõuete kogumise viisist ning sõltuvusest teistest rakendustest pärinevatest andmebaasidest ja failidest [24]. Arendustsükli läbiviimiseks võib kasutada mitmeid erinevaid meetodikaid, aga üks enamtuntud on Kimballi elutsükkel („*Kimball Lifecycle*“) [23, 25], mis töötati välja juba 1990. aastate alguses. Hoolimata tehnoloogia suurtest edusammudest vahepealsel ajal on põhitõed kehtivad ka praegusel ajal ning kasutatavad paljudes projektides.



Joonis 2.3.1: Kimballi elutsükkel [23, 25].

Joonisel 2.3.1 on kujutatud kõrgetasemeliste ja üldjooneliste etappide kaupa andmeaida arendusprojekti elutsükkel Kimballi Grupi ettekujutuses. Koosneb see osaliselt paralleelsetest tööülesannetest, mida kokku on 13:

1. Projekti planeerimine („*Project Planning*“) – fookuses on projekti käimalükkamine, skoobi määratlemine ning töögrupi komplekteerimine
2. Projekti juhtimine („*Project Management*“) – antud ettevõtmine on oluline kogu projekti sujuvaks edenemiseks ning jätkub kogu elutsükli vältel

3. Ärinõuete defineerimine („*Business Requirements Definition*“) – võimaldab määrata organisatsiooni jaoks olulisimad mõjutavad tegurid, keskendudes ärikasutajate tänastele ja tulevastele soovidele. Mõjukamad ideed valitakse välja, organiseeritakse vastavalt väärtuslikkusele ja teostatavusele ning seejärel koostatakse juba täpsed nõuded arendustegevuse jaoks.
4. Tehnilise arhitektuuri disainimine („*Technical Architecture Design*“) – märgitakse ära tehnilised nõuded süsteemile. Koos järgmise tööülesandega moodustavad need „tehnoloogilise raja“ („*Technology Track*“).
5. Toote valimine ja installeerimine (*Product Selection & Installation*) – eelmise sammu jätkuna valitakse sobivad nõudeid täitvad tarkvaralahendused ning seadistatakse need üles
6. Dimensionaalne modelleerimine („*Dimensional Modelling*“) – „andmete rajal“ („*Data Track*“) (punktid 6, 7 ja 8) disainitakse esmalt andmelao jaoks andmemudel, millega tagatakse lihtne ja mugav kasutajakogemus ning päringute kiirus.
7. Füüsiline disain („*Physical Design*“) – selle tööülesande raames saavutab andmemudel füüsilise kuju, kus võetakse arvesse jõudlust parandavaid lahendusi.
8. Teisenduskihi disain ja arendus („*ETL Design ja Development*“) – viimase sammuna lisatakse andmete rajal andmeaita teisenduskiht koos oma eraldus-, puhastus-, laadimis- ja haldusprotsessidega
9. Äriteabe rakenduse määratlemine („*BI Application Specification*“) – koos järgmise sammuga moodustavad need „äriteabe raja“ („*Business Intelligence Track*“), kus esmalt üritatakse ära määratleda, milliseid aruandeid, päringuid, andmekaeve vahendeid jne. on otstarbekas kasutusele võtta.
10. Äriteabe rakenduse arendamine („*BI Application Development*“) – selle tööülesande raames arendatakse valmis eelmises punktis välja valitud vahendid
11. Juurutamine („*Deployment*“) – ühendab omavahel tehnoloogilise külje, andmed ning äriteabe loogika rakendused
12. Haldus („*Maintenance*“) – on vajalik loodud rakenduse töö toetamiseks edaspidiselt
13. Kasvamine („*Growth*“) – viitab andmeaida arendamise järgmisele iteratsioonile, milles kogu elutsüklil algab uuesti projekti planeerimise etapist.

2.4 Andmeaida teisenduskihi protsessid ja nende väljatöötamine

Oluline komponent eelnevas punktis nimetatud arendustsüklis on andmete teisenduskiht ja selle funktsionaalsus. Käesoleva magistritöö praktilise osa parema selgitamise eesmärgil kirjutatakse selle etapi sisemised protsessid täpsemalt lahti.

Selle tarbeks, et lähtesüsteemidest saaksid korrektsed ja valiidsed kirjed andmeaita, tuleb disainida protsess, mis suudab järjepidevalt ilma andmeid dubleerimata valida ja vajadusel töödelda täpsete kriteeriumite järgi algallikatest valitud andmeid. Protsessi väljatöötamist lihtsustab üldine soovituslik nimekiri:

1. **Andmete eel-laadimisala loomine.** Nimetatud vaheetappi kasutatakse abivahendina andmete toimetamisel lähtesüsteemidest andmeaita. Selle abil on võimalik välistada mahukate päringute sooritamist operatiivsüsteemides ning tagada, et juhul kui andmeaida sisemises laadimisprotseduuris tekib viga, siis pole vaja uuesti algallikast andmeid laadida (eeldusel, et viga ei tekkinud eel-laadimisala täitmisel).
2. **Andmete valimine.** Andmeaidas on soovituslik kasutada metaandmete tabelit, milles hoitakse iga andmete algallika kohta infot, näiteks ajatempli viimase laadimishetke kohta. Kasutades sellist abiinfot, on võimalik valida lähtesüsteemist suurim hulk sobivaid kirjeid, mis kattuvad vähimal määral juba andmelattu sisestatud faktidega (näe. kattuvad kirjed on sel juhul ainult need, mis vajavad uuendamist). Et ajatempli kasutamine oleks aga võimalikult lihtne, on soovituslik välistada andmete valimist näe. laadimispäeva kirjed.
3. **Andmete sisestamine eel-laadimisalasse.** Punktis 2 valitud andmed sisestatakse tabelitesse, mida on soovituslik eristada näiteks sufiksiga _STAGE, et tagada parem ülevaade andmelao sisemisest struktuurist.
4. **Andmete töötlemine.** Iga dimensioonitabeli jaoks defineeritakse ja hiljem rakendatakse kindel hulk töötlemisreegleid, mis aitavad andmeid puhastada, korrastada ja ühtlustada. Lisaks võib märkida iga kirje juurde, millisest andmeallikast see pärineb, et hiljem soovi korral lihtsustada andmete tagasilinkimist algallikatega.
5. **Andmete salvestamine dimensioonitabelitesse.** Kui eel-laadimisalas olevad andmed on töödeldud, siis täiendatakse esmalt dimensioonitabeleid – olemasolevate kirjete korral tegutsetakse vastavalt dimensioonitüübile (kas uuendatakse kirjet või luuakse sellest uus versioon) või uued sissekanded lisatakse ja neile genereeritakse võtmed.

6. **Andmete salvestamine faktitabelitesse.** Kui dimensioonitabelid on laetud ja täiendatud, korratakse sarnast operatsiooni lähtesüsteemidest kogutud faktidega – kõik viited dimensioonitabelitele asendatakse vastavate välisvõtmetega ning seejärel kas lisatakse uus kirje või uuendatakse juba andmelaos olevat.
7. **Lõpptegevused.** Kui dimensioonid ja faktitabelid on eel-laadimisalasse kogutud infoga täiendatud ning protseduuride käigus pole ühtegi viga üles kerkinud, on võimalik läbi viia lõpptegevused – eel-laadimisala tabelid tühjendatakse andmetest ning metaandmete tabeli kasutamise korral uuendatakse sealset sisu, näiteks ajatemplit, vastavalt laadimises osalenud lähtesüsteemidele.

Töötades välja (ja hiljem läbi viies) andmeaida teisenduskihi protsesse, on eelpool toodud nimekiri üks variant, kuidas saavutada vähemalt minimaalset süsteemi pakkuv tugistruktuur. Selline raamistik tagab, et on paigas andmete liikumine lähtesüsteemidest andmeaita.

3 Andmevakk Aruandlussüsteemi jaoks - prototüüp

Käesolevas peatükis tutvustatakse lugejale lähemalt antud magistritöö praktilist osa ning selgitatakse selle lähtesüsteeme ning disainiprotsessi.

3.1 Idee

Tänapäevases IT-ühiskonnas on haridusvaldkonnas infosüsteemide kasutamine hädavajalik, võimaldades hallata erinevaid õppekorraldusega seotud andmeid. Esmane tasand on andmete kogumine, säilitamine ja taasesitamine – seda suudavad pakkuda pea kõik süsteemid. Samm edasi on aga analüüsitud tulemuste, aruannete ja prognooside koostamise toetamine. Kui näiteks esimese tasandi süsteemis kuvatakse õppurile ainult tema tulemused, pakkudes ülevaadet isiklikust edasiminekest, siis järgmise tasandi süsteemis oleks võimalik anda üliõpilasele tagasisidet kujul, mis suudab võtta arvesse ka temaga seonduvad kaasõppurid. Võrdluselement pakub mitmekülgsemat tagasisidet õppimisele, võimaldades tulemusi hinnata üldise taseme suhtes.

Soovist pakkuda informatiivsemat tagasisidet tekkis idee antud magistritöö praktilise osa jaoks. Nimelt, Tartu Ülikooli Matemaatika-informaatikateaduskonna talvisel koosolekul arutati, et tudengitele saaks õppimist muuta atraktiivsemaks nn. „Üliõpilaste edetabeli“ kaudu, mis oleks kõigile kättesaadav ning kus õppurid asetuksid järjekorras vastavalt oma õppe edukusele ja õppekava läbimisele. Sellest arenes omakorda koostöös juhendajaga mõte, kuidas täiendada praegust operatiivset õppeinfosüsteemi andmebaasi andmelaoga, mis oleks orienteeritud taoliste info pakkumisele. Andmeladu on selle ülesande jaoks sobiv keskkond ennekõike seetõttu, et niigi aeglase õppeinfosüsteemi kasutuskoormust ei suurendata keeruliste päringutega ning andmeida dimensiooniline andmemudel võimaldab koostada lihtsamaid ja loetavamaid päringuid kui õppeinfosüsteemis.

Esialgelt eeldati töös, et Tartu Ülikoolil puudub keskne andmeladu ning magistritöö raames disainitakse ja ehitatakse see valmis vastavalt disainidokumendile. Vestlustest Õppeosakonna Õppeinfosüsteemi- ja analüüsitalitluse ning Infotehnoloogia osakonna töötajatega selgus aga, et Tartu Ülikoolil on vastav infosüsteem juba olemas ning kasutusel mõningate aruannete jaoks. Seetõttu keskendus magistritöö ümber ennekõike olemasoleva andmelao täiendamisele disainidokumendi nõudeid katva andmevaka kaudu.

Töö raames disainitud prototüüp hõlmab küll ainult üliõpilaste õppetulemuste edetabeli ning sellele baseeruvad erinevad statistilised aruanded, kuid disainimise käigus tekkis ka uusi täiendavaid ideid. Näiteks saaks andmelao ja andmekaeve abil otsida ühiseid mustreid õpingute poolelijätjate tulemustes. Tulevikus aitaks selline tagasiside hinnata õppekava keerukust ning vajadusel viia ellu korrekture katkestajate protsendi vähendamiseks.

Magistritöö raames disainitud prototüüp on loodud ajutise katsetusena, et vältida vajadust selle plaaniväliseks toetamiseks Infotehnoloogia osakonna poolt, kuid Õppeosakond on viidanud, et edasise huvi korral võidakse antud disain arendusprojekti raames täieõigusliku osana realiseerida.

3.2 ÕIS

Tartu Ülikoolis on ametlikuks õppekorralduse infovahetuskeskkonnaks Õppeinfosüsteem ehk ÕIS. See võeti kasutusele 6. novembril 2001. aastal ning selle kasutamist reguleerib „Õppeinfosüsteemi korraldus Tartu Ülikoolis“. [26, 27]

ÕIS seob tervikuks ligi kümme eri funktsiooniga eri andmeid sisaldavat moodulit, igapäevase õppetöö korraldamise seisukohalt on neist olulisemad esimesed 7:

1. **õppekavade moodul** sisaldab õppekavade üldandmeid ja õppekavade kõigi aastate versioone;
2. **õppeainete moodul** sisaldab õppeainete üldandmeid ja ainekavasid; õppeaine tunnuseks moodulis on ainekood, mis luuakse ÕISi poolt automaatselt õpetava struktuuriüksuse koodi alusel, kasutuses olnud ainekoode ei anta uutele ainetele;
3. **tunniplaanide moodul**;
4. **õppetööle registreerimise moodul** sisaldab registreerimiseks avatud aineid; moodulis toimub ainetele ja eksamitele registreerimine;
5. **õpitulemuste moodul** sisaldab eksamite ja arvestuste protokolle, varasemate õpingute ja töökogemuse arvestamise (VÕTA) otsuseid ja lõputööde kaitsmise protokolle;
6. **täienduskoolitusprogrammide moodul**;
7. **üliõpilasolekute moodul** sisaldab üliõpilaste andmeid (isikuandmed, õppekoht, edasijõudmine õppetöös); üliõpilase tunnuseks ÕISis on matriklinumber, mis luuakse

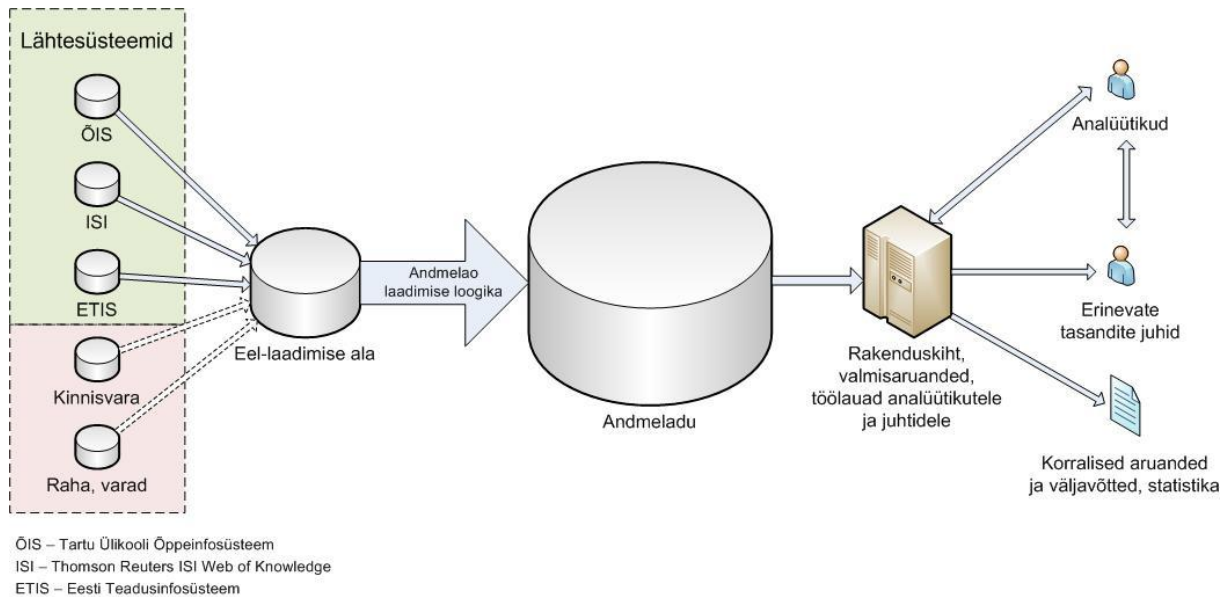
automaatselt immatrikuleerimise või eksterniks või külalisüliõpilaseks arvamise korralduse registreerimisel üliõpilasolekute moodulis;

8. **õppetoetuste arvestamise moodul** sisaldab andmeid õppetoetust saavate üliõpilaste, toetuse suuruse ja perioodi ning väljamaksete kohta;
9. **vastuvõtumoodul** sisaldab vastuvõtuks avatud õppekavasid, vastuvõtutulemuste arvutamise aluseks olevaid kriteeriume ning andmeid üliõpilaskandidaatide ja nende vastuvõtutulemuste kohta;
10. **kasutajate haldamise moodul** sisaldab ülikooli isikute andmebaasi kantud kasutajate ning kasutusõiguste kohta käivaid andmeid;
11. vastavalt vajadusele loodud **täiendav(ad) moodul(id)**, mis sisaldavad muid andmeid.

ÕIS võimaldab küll üliõpilastele või õppejõule ligipääsu konkreetsetl temaga (ja tema õppeainega) seotud infole, kuid piirab isikuandmete konfidentsiaalsuse tõttu võimalust tutvuda teiste isikutega seotud infoga. Lisaks on ÕISi probleemiks selle süsteemi aeglus – juba praegu on tudengitele tunda, kuidas suurte koormustega perioodidel (näiteks uue semestri ainetele registreerimise võimaluse avamisel) süsteemi laadimisajad suurenevad kordades ning halvimal juhul võib ligipääs olla ajutiselt võimatu. Seetõttu ei ole otstarbekas ÕISi süsteemi lisada keerukaid analüütilisi protsesse ja päringuid.

3.3 Aruandlussüsteem

Tartu Ülikooli andmeladu, antud töös viidatud kui Aruandlussüsteem, on Tartu Ülikooli infosüsteem ja seega universaalne keskne andmekogu aruannete ja analüütiliste päringute jaoks. See seob omavahel mitmed erinevad lähtesüsteemid nii ülikoolisiselt kui ka väljastpoolt (joonis 3.3.1): Õppeinfosüsteem ÕIS, publikatsioonide infosüsteem ISI, Eesti Teadusinfosüsteem ETIS, Tartu Ülikooli finantssüsteem AXAPTA koos rahaveebiga ja kinnisvara haldussüsteem. Arendatud on seda alates 2007. aastast tellimustööna peamiselt AS Resta ja AS Logica (nüüdne CGI) poolt [28]. Andmelao kasutajaliideseks on Oracle Answers, mis on omakorda väiksem osa Oracle-i poolt pakutavast Oracle Business Intelligence Enterprise Editionist [29].



Joonis 3.3.1: Tartu Ülikooli andmelao ülesehitus.

Andmelao eesmärkideks on:

- koondada suur osa aruandlusest ühte infosüsteemi
- vähendada käsitööd aruannete koostamisel
- võimaldada aruandlust üle erinevate valdkondade
- töötada välja standardsed mõisted ja aruanded
- toetada organisatsiooniüleste mõõdikute loomist ja kasutamist
- teha standardsed aruanded laiemalt kättesaadavaks

Hetkel on võimalik sealtkaudu saada näiteks ülevaadet õppijate arvust õppeastme järgi nii õppevaldkonniti, teaduskonniti kui ka päritolumaa järgi. Lisaks on seal olemas ülevaade ülikooli töötajate jaotumisest ning publikatsioonide, juhendamiste ja ESI indikaatorite infot koondavad aruanded.

Andmelao laadimine toimub automaatselt. Laadimiste graafik lähtesüsteemide kaupa:

- igal ööl: ÖIS, Axapta, Rahaveeb, Archibus
- igal pühapäeval: ETIS
- iga kuu 25ndal: ESI
- iga kvartali viimase kuu 25ndal: ISI publikatsioonid
- iga kvartali viimase kuu 29ndal: ISI tsiteeringute arvud

3.4 Lahendus

Vastavalt peatükis 3.1 välja toodud ideele seisneb antud magistritöö praktilise osa lahendus Tartu Ülikooli andmelaole uue andmevaka disainimises. Selle eesmärgi saavutamiseks läbiti osaliselt muudetud versioon andmeaida arendustsüklist (peatükk 2.3). Et esiteks oli eesmärgiks luua ainult prototüüp ning teiseks oli antud juhul tegu olemasoleva andmelao täiendamisega, põhjustasid mõlemad faktorid muudatused arendustsükliis.

3.4.1 Disainiprotsess

Arendustsükli esimeseks sammuks oli projekti planeerimine, mille käigus määratleti skoop – üliõpilaste õppetulemustel põhineva edetabeli prototüüp – ning pandi paika edasised sammud. Järgmises etapis defineeriti skoobist tulenevad ärinõuded, mis suudaksid igakülgset katta kasutajate eeldatavaid soove, ning koostati nende põhjal koos kasutuslugudega disainidokument (Lisa 1). Esmalt toimus see protsess koostöös juhendajaga, hiljem pöörduti ka Õppeosakonna Õppeinfosüsteemi- ja analüüsitalitluse ning Infotehnoloogia osakonna Infosüsteemide talitluse poole, kes pakkusid omaltpoolt sisulist ja tehnilist kompetentsi, olles vastavalt nii ÕISI kui ka andmelao haldajad. Kontaktisikuteks olid vastavalt Ivika Puusepp ja Björn Dahl.

Mitmete kokkusaamiste tulemusena suudeti piiritleda skoobist lähtuv ülesanne ning tuvastada andmeolemid, mida oli lähtesüsteemidest ja andmelaost vaja töö eesmärgi saavutamiseks. Et tänu varasematele andmelao arendustele olid juba mitmete ÕISI moodulite andmed Aruandlussüsteemis kättesaadavad, ei olnud vajadust kõiki andmeid õppeinfosüsteemist pärida. Antud praktilise osa raames suurendati andmelao katvust ÕISI suhtes õppeainete, õppetööle registreerimise ja õppetulemuste moodulite ulatuses.

Juriidiliseks takistuseks osutus esmalt isikuandmete kaitse – nii ÕIS kui ka andmeladu sisaldavad isikustatud andmeid, mis on konfidentsiaalsed ning millele Tartu Ülikooli seaduste järgi on täismahus ligipääs lubatud ainult vastavalt ametikirjelduses fikseeritud tööülesannetele. Nõnda on välistatud nii üliõpilased, õppejõud kui ka avalikud kasutajad. Seetõttu saavutati ülikoolipoolsete kontaktidega kokkulepe, et üleantavast andmekogust eemaldatakse kõik isikustatud andmed ning iga tudengi kirje on esindatud ainult selle sisemise identifikaatori kaudu andmelaos. Sel viisil on otsene tuvastamine välistatud ning oht kaudseks tuvastamiseks õppeainete läbimise kaudu minimaalne. Teise sammuna sõlmiti Tartu

Ülikooli ja magistrandi vahel „Andmete kasutamise leping“, millega määratleti ära mõlema poole õigused ja kohustused ning pandi üheselt kirja, millistele andmetele ligipääs võimaldatakse.

Olles omandanud ligipääsu andmekogule, oli võimalik koostada andmemudel ning TÜ andmelao disainidokumendi põhjal valmis kirjutada sarnane dokument ka antud andmevaka jaoks (Lisa 2).

3.4.2 Prototüüp

Esmane visuaalne prototüüp valmis suhteliselt arendustsükli alguses, et kohtumistel Õppe- ja Infotehnoloogia osakonna inimestega oleks võimalik paremini selgitada magistritöö eesmärke ning määratleda täpsemalt vajaminevaid andmekogumeid. Nimetatud prototüüp loodi ainult HTML ja CSS keeli kasutades ning sisaldas 4 vaadet. Õppetulemustest parema visuaalse ülevaate saamiseks on igale hindele määratud kindel värvitoon rohelisest punaseni – parim tulemus A on tähistatud rohelise tooniga ning kehvim tulemus F punasega. Vahepealsed hinded B, C, D ja E on tähistatud seguga nendest toonidest. „Arvestatud“ ja „Mittearvestatud“ hinded on tähistatud vastavalt hinnete A ja F toonidega. Kui üliõpilasel puudub tulemus antud aines (tegemist on näiteks vabaainega õppekavas ning ta ei ole sellele registreerunud), siis kuvatakse tabelis sellel kohal valge kast.

Minu tulemused

<u>Programmeerimine</u>	A	A	A	A	B
<u>Matemaatiline analüüs</u>	B	B	D	C	E
<u>Algebra</u>	B	B	D	E	
<u>Süsteemihaldus</u>	B	C	C	E	F

Joonis 3.4.2.1: "Minu tulemused" vaade.

Esimesel vaatel „Minu tulemused“ (joonis 3.4.2.1) kuvatakse üliõpilase hinded võrdluses kõigi temaga sama aasta õppekavale sisseastunutega. Õppeained ja tudengid sorteeritakse vastavalt hinnetele nõnda, et kõige paremate tulemustega aine on esimeses reas ning kõige paremate tulemustega õppur esimeses hindeveerus. Sellisel viisil on võimalik kuvada

üliõpilaste edetabelit vastavalt õppeedukusele. Tulemusi vaatava isiku ehk „Mina“ hinded on eristuvad sinise piirjoonega tulbana. Iga õppeaine nimi on ühtlasi ka link, mis viitab vastava aine tulemustele aastate lõikes „Minu ained“ vaates.

Minu ained

Näita:

Matemaatiline analüüs

2012	A	B	B	D	F
2011	B	B	C	C	D
2010	A	B	B	D	F
2009	B	B	C	E	E

Joonis 3.4.2.2: "Minu ained" vaade.

Teise vaate „Minu ained“ (joonis 3.4.2.2) eesmärk on kuvada üliõpilastele täpsem ülevaade õppetulemustest aastate lõikes kõikides õppeainetes, milles antud tudeng on omandanud tulemusi. Hinded on kuvamisel sorteeritud iga aasta jaoks gruppideks parimast kõige kehvemani suunaga vasakult paremale. Õppija isiklik tulemus on esile tõstetud sinise raamiga kastina. Rippmenüüst on võimalik valida kõiki teisi läbitud aineid.

Õppetulemused õppekava järgi

Teaduskond:

Õppeaste:

Õppekava:

Eetika alused	A	A	A	A	B
Esteetika	A	B	B	C	E
Euroopa ideede ajalugu	B	B	C	D	
Semiootika alused	B	B	C	D	F

Joonis 3.4.2.3: "Õppetulemused õppekava järgi" vaade.

Kolmas vaade „Õppetulemused õppekava järgi“ (joonis 3.4.2.3) tutvustab võimalust kuvada õppetulemusi õppekavade kaupa. Konkreetse õppekava valimiseks on vaja täpsustada mitmed kitsendavad tingimused, antud joonisel on näiteks toodud teaduskond ja õppeaste. Üliõpilase isiklikku tulemust selles vaates hinnete seas ei tähistata ning seetõttu ei organiseerita tulemusi ka õppijate järgi, vaid sorteeritakse hinnete järgi. Seega on parimate tulemustega õppeaine esimeses reas ning veergudes on hinded tähe järgi grupeerituna järjestatud parimast kõige kehvemani suunaga vasakult paremale.

Õppetulemused aine järgi

Teaduskond:

Instituut:

Õppeaine:

Semiootika ajalugu

2012	B	B	C	D	F
2011	B	C	C	C	D
2010	A	B	B	D	F/C
2009	B	B	E	E	E

Joonis 3.2.4: "Õppetulemused aine järgi" vaade.

Neljas vaade „Õppetulemused aine järgi“ (joonis 3.4.2.4) võimaldab sarnast ülevaadet hinnetest aastate lõikes nagu vaade „Minu ained“, kuid siin ei ole nimekiri piiratud ainult üliõpilase poolt läbitud ainetega. Konkreetse aine valimiseks on vaja täpsustada mitmed kitsendavad tingimused, antud joonisel on näiteks toodud ainult teaduskond ja instituut. Ka selles vaates ei tooda õppija isiklikku tulemust esile ning hinded on organiseeritud gruppidesse vasakult paremale nõnda, et parim hinne A on esimene ning kehvim tulemus F viimane. Antud joonisel on näha ka viis, kuidas plaanitakse kuvada olukorda, kus esmalt omandati aines puudulik hinne F ning seejärel „parandati“ see hindele C.

Kõik need 4 vaadet pärinevad esialgest visuaalsest prototüübist ning ei ole töö käigus loodud andmemudeli tulemid.

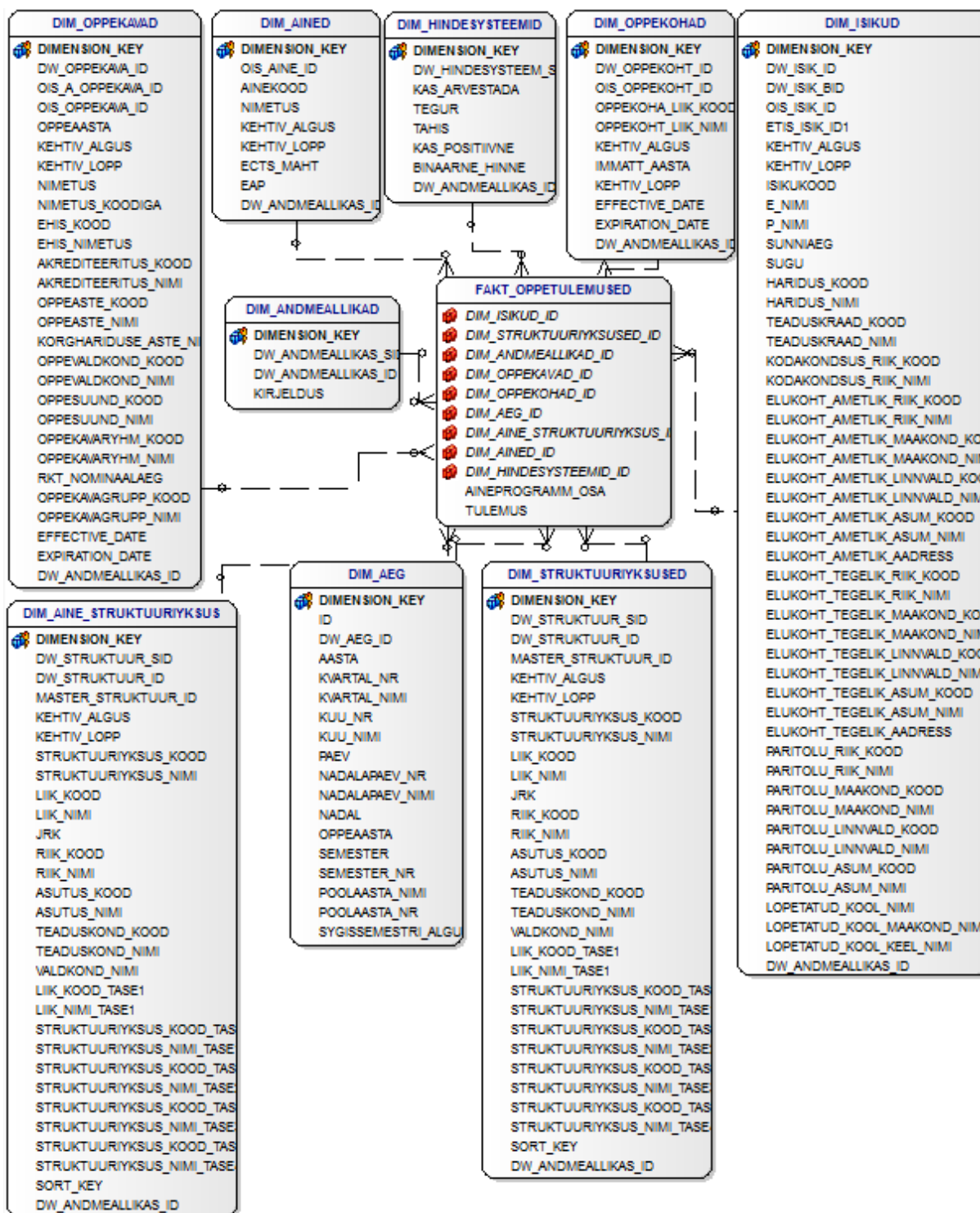
3.4.3 Ülevaade kasutatud töövahenditest

Disainiprotsessi käigus olid kasutusel mitmed erinevad tarkvaralised töövahendid. Andmelaole ligipääsemiseks osutus parimaks Oracle SQL Developer [30], millest töös kasutati versiooni 3.2. Tegemist on tasuta kättesaadava SQL arenduskeskkonnaga, mis on orienteeritud lihtsustamiseks Oracle-i andmebaaside ehitamist ja haldamist.

Andmemudeli koostamiseks kasutati Datanamic Solutions-i tarkvara DeZign for Databases versiooni V7.3.5 [31]. Tegemist on andmebaaside disainimiseks ja modelleerimiseks mõeldud töövahendiga, mis toetab enamlevinuid andmebaasiplatvorme (Oracle, MySQL, PostgreSQL jt.). Tegemist ei ole küll vabavaraga, kuid esmaseks testimiseks on olemas 30-päevane tasuta prooviperiood. Antud tarkvara eeliseks oli intuiitiivne ja mugav kasutajaliides, tänu millele kulus tundmaõppimisele märkimisväärselt vähe aega ning disainimisega oli võimalik kiirelt alustada.

3.4.4 Andmemudel

Prototüübi andmemudel (joonis 3.4.3.1) on disainitud, lähtudes dimensionaalise modelleerimise põhimõtetest, et toetada paremini andmelaot funktsioneerimist. Dimensionid DIM_OPPEKOHAD, DIM_ISIKUD, DIM_OPPEKAVAD, DIM_AEG, DIM_ANDMEALLIKAD ja DIM_STRUKTUURIYKSUSED olid andmelaos juba olemas varasematest arendustest. Dimensioon DIM_AINE_STRUKTUURIYKSUS omab identset struktuuri tabeliga DIM_STRUKTUURIYKSUSED, vajadus selle järgi tuleneb tingimusest, et faktitabel saab olla sama dimensioniga seotud vaid ühe korra. Selle prototüübiga on andmelattu lisanduvateks dimensionideks DIM_HINDESYSTEEMID, DIM_AINED ning faktitabeliks FAKT_OPPETULEMUSED. Täpsem info, milliseid andmeid antud tabelid hoiavad ning kuidas seostuvad lähtesüsteemidega, on kirjas dokumendis „Andmevaka disain“ (Lisa 2).



Joonis 3.4.3.1: Prototüübi andmemudel – dimensiooni- ja faktitabelid.

3.4.5 Abilahendused

Lisaks dimensiooni- ja faktitabelitele sisaldab andmeladu eel-laadimisala- ja metaandmete tabeleid. Esimesena nimetatud struktuuridesse kogutakse andmed lähtesüsteemidest vastavalt ajakriteeriumile ning nendes toimub ka andmete (ümber)töötlemine vastavalt reeglitele. Metaandmete tabel sisaldab aga andmelao toimimiseks vajalikku korralduslikku infot – iga kasutusel oleva lähtesüsteemi kohta on seal ajatempliga kirje, mis tähistab viimast laadimishetke. Selleks, et metaandmete tabelis hoitav info oleks ajakohane, määratakse pärast edukat andmete laadimisprotsessi lõppu iga kasutatud lähtesüsteemi kirje ajatempliks SQL-i UPDATE lausega käesolev ajahetk (Lisa 3, SQL 21).

Veel on andmelaos kasutusel Oracle-i automaatsed jadad („*sequence*“), et tagada kirjete identifikaatorite unikaalsus ja idempotentsus. Iga dimensioonitabeli jaoks on loodud jada, mis genereerib primaarvõtme („*primary key*“), mis on kasutusel ka faktitabelis kirjete linkimiseks vastavatesse dimensioonidesse.

3.4.6 Ajahaldus andmete laadimisel tabelitesse

Õppeinfosüsteemi andmelao laadimisel eristatakse alglaadimist (andmete esmakordne laadimine, kus andmelattu laaditakse dimensioonide algseis ja kõik enne alglaadimise kuupäeva registreeritud faktid) ja regulaarlaadimist (kõik järgnevad andmelao laadimised, kus andmelattu laaditakse pärast viimast andmelao laadimist tehtud muudatused dimensioonides ja uued registreeritud faktid).

Andmelao regulaarlaadimise algoritmides kasutatakse mõisteid „andmelao (käesoleva) laadimise kuupäev” ja „andmelao eelmise laadimise kuupäev”. Andmelao (käesoleva) laadimise kuupäeva all mõistetakse vastavale laadimise alustamise kuupäevale eelneva kuupäeva viimast sekundit ja andmelao eelmise laadimise kuupäeva all mõistetakse vastava kuupäeva esimest sekundit. Sellest lähtuvalt laetakse regulaarlaadimisel faktid, mis on registreeritud enne andmelao laadimise kuupäeva, ja jäetakse laadimata faktid, mis on registreeritud andmelao laadimise kuupäeval (kuna kõik selle päeva tegevused ei ole veel registreeritud).

Kui andmelao laadimisel ei kasutata reaajas toimivat lähteandmebaasi, vaid selle regulaarselt tehtavat koopiat, siis tuleb mõiste „andmelao laadimise kuupäev” all mõista lähteallikast koopia tegemise kuupäeva.

Eelpool nimetatud regulaarlaadimise algoritmid lähtuvad andmete valimisel algtabelite väljast „Aeg“, kus hoitakse iga kirje kohta selle lisamise hetke tähistavat ajatemplit. Kui kirjega toimub muudatus, uuendatakse vastavalt ka eelpoolnimetatud välja väärtust. Juhul kui ühe tabeli väärtus lisatakse võõrvõtme („foreign key“) kaudu kõrvaltabelist, siis tuleks laadimisalgoritmide huvides uuendada ka võõrvõtme välja „Aeg“.

3.5 Tööprotsess

Prototüübi lahenduse täideviimist saab jagada mitmesse järjestikusesse etappi, sealhulgas siis nii töö lähtesüsteemide ja andmelaoga kui ka prototüübi toimiva lahenduse kirjutamine kasutades programmeerimiskeeli. Kõik järgmistes etappides kasutatud SQL-laused on ära toodud Lisas 3 ning täpsed viited on etappides välja toodud.

3.5.1 Tabelite loomine

Esimese sammuna sai peatükis 3.4.4 defineeritud andmemudelid realiseeritud CREATE SQL lausetega uued dimensioonitabelid dim_ained (SQL 1), dim_aine_struktuuriyksus (SQL 3), dim_hindesüsteemid (SQL 2) ja faktitabel fakt_oppetulemused (SQL 4). Kuigi välisvõtmete kasutamine suurendab INSERT, UPDATE ja DELETE SQL lausete protsessimisel töömahtu, on antud juhul eelistatud andmete kvaliteedi säilimine ja seetõttu on need viited sisse jäetud.

Järgmiseks loodi eelpoolnimetatud tabelitele samuti CREATE SQL käsku kasutades eel-laadimisala tabelid: dim_hindesüsteemid_stage (SQL 5), dim_ained_stage (SQL 6) ja fakt_oppetulemused_stage (SQL 7). Et dimensioon dim_aine_struktuuriyksus on üks-ühele koopia juba andmelaos olemasolevast tabelist dim_struktuuriyksused, siis selle laadimine käib vastavalt kopeeritava dimensiooni reeglitele ning prototüübis selle jaoks eraldi eel-laadimisala ei planeeritud.

Viimase sammuna loodi selles etapis peatükis 3.4.5 kirjeldatud abilahendused: metaandmete tabel (SQL 8) ning jadad seq_dim_hindesys_primary_key (SQL 9) hinnete dimensiooni ja seq_dim_ained_primary_key (SQL 10) ainete dimensiooni tarbeks.

3.5.2 Eel-laadimisala täitmine

Kui eel-laadimisala tabelid on loodud, täidetakse need ÕIS-ist kogutud infoga: `dim_hindesüsteemid_stage` (SQL 11), `dim_ained_stage` (SQL 12) ja `fakt_oppetulemused_stage` (SQL 13). Dimensioonide kirjade töötlemisreeglid on ära toodud tehnilise disaini dokumendis Lisa 2.

3.5.3 Dimensioonitabelite täitmine

Töödeldud ja eel-laadimisalasse salvestatud andmed kantakse edasi dimensioonidesse, igale uuele kirjele lisatakse külge primaarvõti `dimension_key`. Dimensioonid täidetakse vastavalt SQL lausetega: `dim_hindesüsteemid` (SQL 14), `dim_ained` (SQL 15). Mõlemal juhul lisatakse kõigile kirjetele väljale `dw_andmeallikas_id` väärtus 2, mis tähistab andmeallikana ÕIS-i.

3.5.4 Faktitabeli täitmine

Kui dimensioonide laadimine on lõpetatud, tõstetakse kirjed ümber ka faktitabeli eel-laadimisalast (SQL 16). Kõik viited dimensioonitabelite kirjetele vahetatakse vastavate kirjete ID-d (`dimension_key`) vastu.

3.5.5 Lõpptegevused andmeaidas

Kui dimensiooni- ja faktitabelite laadimine andmetega on edukalt lõpule viidud, uuendatakse metaandmete tabelis vastavat lähtesüsteemi kirje ajatemplit (SQL 20) vastavaks antud ajahetkega. Lisaks tühjendatakse kõik eel-laadimisala tabelid sinna kogutud kirjetest (SQL 21).

3.5.6 Prototüübi koostamine

Kui vajalikud testandmed said eelpoolnimetatud SQL käskudega andmeaita sisse kantud, oli võimalik asuda prototüüpi realiseerima. Kui peatükis 3.4.2 loodud lahendus kasutas ainult HTML-i ja CSS-i, siis antud etapis loodud prototüüp on kirjutatud juba skriptimiskeeles PHP ning teostab päringuid andmeaida tabelite vastu. Et peatükis 3.4.1 nimetatud ligipääsu Tartu

Ülikooli andmekogule ei olnud 2014. aasta kevadel enam võimalik kasutada, siis on prototüüp ajutiselt üles ehitatud internetis tasuta pakutavale MySQL andmebaasi veebiteenusele FreeSQLDatabase [32]. Kõik punktides 3.5.1 – 3.5.5 kasutatud Oracle-i spetsiifilised SQL laused kohandati arendamise käigus sobivaks MySQL andmebaasile. Tänu veebiteenusele on võimalik prototüüpi testida ka mujal kui ainult magistritöö autori isikliku arvuti lokaalmasinas.

Esimese ülesandena realiseeriti prototüübist „Minu tulemused“ vaade, kus küsiti andmeaidast kõik otsitud isikuga sama aasta õppekaval olevate tudengite tulemused [33]. Töö lihtsustamiseks eeldati, et on teada õppija ID (tabelis *dim_isikud.ois_isik_id*) ning leiti selle põhjal esmalt tema aasta õppekava ID ning siis kõik tulemused, mille õppekava ID on sama (SQL 17). Et saavutada peatükis 3.4.2 demonstreeritud sorteeritud vaadet, kasutati järgnevat algoritmi:

1. Arvutati igale üliõpilasele keskmine hinne: tema tulemuste tegurid liideti kokku ja jagati kõigi õppeainete arvuga.
2. Arvutati igale õppeainele keskmine hinne: selle tulemuste tegurid liideti kokku ja jagati kõigi leitud õppurite arvuga.
3. Mõlemate keskmiste hinnete jadad sorteeriti nõnda, et järjestus oli suurimast numbrist väikseimani.
4. Viimaks käidi sorteeritud õppeainete jada elemendi kaupa läbi nõnda, et igale reale tekkis esmalt antud õppeaine nimetus ning siis vastavalt tudengite keskmise hinde jadale ka nende tulemused.
5. Et kõigil andmeaidast saadud tulemustel oli küljes ka üliõpilase identifikaator ÕIS-is (mis oli samal ajal ka prototüübis teada), oli võimalik teiste tulemuste hulgast eristada nõ. päringut teinud isiku hindeid.

Teise ülesandena realiseeriti prototüübist „Minu ained“ vaade [34], kus küsiti andmebaasist esmalt kõik õppeained, milles teadaoleva ID-ga õppija on tulemuse kirja saanud (SQL 18). Seejärel valiti neist ainetest üks ning päriti omakorda andmeaidast kõik selle õppeaine tulemused erinevatel õppeaastatel. Kergema organiseerimise eesmärgil sorteeriti need tulemused esmalt *dim_oppekavad.oppeaasta* ning seejärel *dim_hindesysteemid.tegur* järgi

kahanevalt (SQL 19). Et saavutada peatükis 3.4.2 demonstreeritud sorteeritud vaadet, kasutati järgnevat algoritmi:

1. Andmeidast tagastatud sorteeritud andmed kirjutatakse jadasse nõnda, et esimesel tasemel tekib iga õppeaasta kohta üks võti („*key*“) ning selle väärtused („*value*“) on sellel õppeaastal välja antud hinded.
2. Punktis 1 koostatud jada käiakse läbi nõnda, et iga õppeaasta kohta tekib tabelisse üks rida, kus esimeses veerus on õppeaasta nimi ning selle järel paremusjärjestuses tulemused.
3. Kõigil andmeidast saadud tulemustel on küljes üliõpilase identifikaator, mis on kasutusel ÕIS keskkonnas, võimaldades seeläbi esitatud tulemuste hulgast eristada nõ. päringut teinud isiku hinnet.

Ülejäänud 2 vaadet „Õppetulemused õppekava järgi“ ja „Õppetulemused õppeaine järgi“ jäid prototüübis realiseerimata, kuid teostamisidee oleks sarnane variandiga „Minu ained“, kus esmalt leitakse filtrite tarbeks õppekavade ja õppeainete nimekirjad ning siis vastavalt valikutele filtreeritakse välja sobivad tulemused. Need omakorda järjestatakse ning seejärel kuvatakse PHP skripti abil tabeliformaadis välja. Erinevust on nõnda palju, et mõlema nimetatud vaate korral ei tooda kuvatavate tulemuste seast eraldi välja tudengile määratud tulemust.

Kokkuvõte

Andmeaitate tähtsus on tänapäeva infoühiskonnas märkimisväärne, võimaldades ettevõtetel ja organisatsioonidel koostada mitmekülgseid ja edasiviivaid analüüse ning aruandeid enda valduses oleval info. Tänu sellele on võimalik avastada seoseid, mis aitavad edu saavutada või suurendada.

Käesolevas magistritöös tutvustati andmeaitasid, nende ajalugu ja ülesehitust. Lisaks anti lugejale ülevaade nii olulistest võtmeteguritest, mida jälgida andmeaida planeerimisel, kui ka erinevatest sammudest andmelao arendustsüklis disainist lõpplahenduse haldamiseni. Selle info kaudu oli lugejal võimalik saada põgus ülevaade antud valdkonna toimimisest ja olulisusest.

Töö praktilises osas valmis üliõpilaste edetabeli prototüüp Tartu Ülikooli Aruandlussüsteemi tarbeks, mis võimaldab koostada aruandeid tudengite õppeedukuse kohta ning seeläbi pakub õppijatele enda kohta mitmekülgsemat tagasisidet. Tööprotsessi käigus kavandati visuaalne prototüüp, määratleti skoop ning läbi mitmetearendustsüklite kirjeldati kasutuslood, andmemudel ning pandi kokku töötav prototüüp.

Eesmärgiks seatud prototüüp realiseeriti valdavas osas plaanitust – valmis nii vajalik dokumentatsioon kui ka planeeritud 4-st vaatest jõuti valmis kirjutada 2 („Minu tulemused“ ja „Minu ained“), mis mõlemad kasutavad andmeid loodud andmelao struktuurist. Kaitsmise järel edastatakse loodud dokumentatsioon ka Õppeosakonnale tutvumiseks, et neil oleks võimalik kaaluda välja pakutud funktsionaalsuse lisamist olemasolevale lahendusele.

Tehtud töö pakub edasiarendamise võimalusi ennekõike üliõpilaste õppetulemuste aruannete koostamise valdkonnas – antud info põhjal on võimalik koostada põhjalikke analüüse ning hinnata näiteks õppeainete taset lähtuvalt tudengite tulemusest aastate lõikes. Tänu nendele aruannetele on Tartu Ülikoolil huvi korral võimalik üritada välja selgitada mitmesuguseid mustreid üliõpilaste õppete katkemises ja saadud info puhul sobivaid muudatusi ette võtta.

Viited

- [1] P. Ponniah, *Data Warehousing Fundamentals for IT Professionals*, Second edition toim., John Wiley & Sons, Inc., 2010.
- [2] P. Adriaans ja D. Zantinge, *Data Mining*, Addison Wesley Longman, 1996.
- [3] R. Öpik, *Andmeaidad, ERP ja CRM-süsteemid. IDU0010. 2.loeng*, 2012.
- [4] „Business intelligence solutions - Adepta,“ [Võrgumaterjal]. Available: <http://adepta.ee/>. [Kasutatud 21 Aprill 2013].
- [5] „Resta,“ [Võrgumaterjal]. Available: <http://www.resta.ee>. [Kasutatud 21 Aprill 2013].
- [6] „AS Tallinna Vee koduleht,“ [Võrgumaterjal]. Available: <http://www.tallinnavesi.ee/>. [Kasutatud 12 Mai 2013].
- [7] „AS Estonian Air koduleht,“ [Võrgumaterjal]. Available: <http://estonian-air.com/et/>. [Kasutatud 12 Mai 2013].
- [8] „AS Tere koduleht,“ [Võrgumaterjal]. Available: <http://www.tere.eu/et/>. [Kasutatud 12 Mai 2013].
- [9] „Politsei- ja Piirivalveameti koduleht,“ [Võrgumaterjal]. Available: <http://www.politsei.ee/>. [Kasutatud 12 Mai 2013].
- [10] „Eesti Posti koduleht,“ [Võrgumaterjal]. Available: <http://www.post.ee/>. [Kasutatud 12 Mai 2013].
- [11] F. Hayes, „The story so far - Computerworld,“ 2002. [Võrgumaterjal]. Available: http://www.computerworld.com/s/article/70102/The_Story_So_Far. [Kasutatud 21 Aprill 2013].
- [12] „Apache™ Hadoop®,“ [Võrgumaterjal]. Available: <http://hadoop.apache.org/>.

- [13] „Oracle Big Data Appliance,“ Oracle, [Võrgumaterjal]. Available:
<http://www.oracle.com/us/products/database/big-data-appliance/overview/index.html>.
- [14] Forbes. [Võrgumaterjal]. Available: <http://www.forbes.com/sites/oracle/2014/03/10/the-top-10-trends-in-data-warehousing/>.
- [15] „Data Warehousing Review,“ [Võrgumaterjal]. Available:
http://dwreview.com/DW_Overview.html. [Kasutatud 21 Aprill 2013].
- [16] „MySQL,“ [Võrgumaterjal]. Available: <http://www.mysql.com/>. [Kasutatud 21 Aprill 2013].
- [17] „Oracle,“ [Võrgumaterjal]. Available: <http://www.oracle.com>. [Kasutatud 21 Aprill 2013].
- [18] „Datawarehouse4u.Info,“ [Võrgumaterjal]. Available:
<http://datawarehouse4u.info/OLTP-vs-OLAP.html>. [Kasutatud 21 Aprill 2013].
- [19] K. Haga, „Andmeait. Arvutimaailm,“ 2004. [Võrgumaterjal]. Available:
https://www.ria.ee/lib/am-2001-2005/14767_3B.HTM. [Kasutatud 21 Aprill 2013].
- [20] „IBM InfoSphere Information Server,“ [Võrgumaterjal]. Available: http://www-01.ibm.com/software/data/integration/info_server/. [Kasutatud 21 Aprill 2013].
- [21] „Informatica PowerCenter,“ [Võrgumaterjal]. Available:
<http://www.informatica.com/us/products/enterprise-data-integration/powercenter/>.
[Kasutatud 21 Aprill 2013].
- [22] „SAP Data Integrator,“ [Võrgumaterjal]. Available:
<http://www54.sap.com/solutions/tech/enterprise-information-management/software/data-integrator/index.html>. [Kasutatud 21 Aprill 2013].
- [23] R. Kimball, *The Data Warehouse Lifecycle Toolkit*, Second Edition toim., Wiley, 2008.
- [24] T. C. Hammergren ja A. R. Simon, *Data Warehouse For Dummies*, Second Edition toim., Wiley Publishing Inc., 2009.

- [25] K. Group, „Design Tip #115 Kimball Lifecycle in a Nutshell,“ 2009. [Võrgumaterjal]. Available: <http://www.kimballgroup.com/2009/08/04/design-tip-115-kimball-lifecycle-in-a-nutshell/>. [Kasutatud 02 Mai 2013].
- [26] „ÕIS - mis ja milleks?,“ [Võrgumaterjal]. Available: <https://wiki.ut.ee/pages/viewpage.action?pageId=15600355>. [Kasutatud 11 Mai 2013].
- [27] „Õppeinfosüsteemi korraldus Tartu Ülikoolis,“ [Võrgumaterjal]. Available: http://www.ut.ee/livelink_files/1194708.pdf. [Kasutatud 11 Mai 2013].
- [28] *TÜ andmelao disain*, Tartu Ülikooli Infotehnoloogia osakond, 2013.
- [29] „Oracle Business Intelligence Enterprise Editioni koduleht,“ Oracle, [Võrgumaterjal]. Available: <http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/overview/index.html>. [Kasutatud 12 Mai 2013].
- [30] „Oracle SQL Developer,“ Oracle, [Võrgumaterjal]. Available: <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>. [Kasutatud 13 Mai 2013].
- [31] „DeZign for Databases,“ Datanamic, [Võrgumaterjal]. Available: <http://www.datanamic.com/dezign/index.html>. [Kasutatud 12 Mai 2013].
- [32] „FreeSQLDatabase,“ [Võrgumaterjal]. Available: <http://www.freesqldatabase.com/>.
- [33] I. Sauga, „Prototüüp - Minu tulemused,“ [Võrgumaterjal]. Available: http://piwik.noillir.eu/proto-minu_tulemused.php.
- [34] I. Sauga, „Prototüüp - Minu ained,“ [Võrgumaterjal]. Available: http://piwik.noillir.eu/proto-minu_ained.php.
- [35] Oracle Corporation, *Top 10 Data Warehouse Trends for 2013*, 2012.

Lisad

Lisa 1. Disainidokument „Üliõpilaste edetabelile“

TARTU ÜLIKOOL

Üliõpilaste edetabel

Disainidokument

TARTU 2014

SISSEJUHATUS

Antud dokumendi eesmärgiks on kirjeldada projekti „Üliõpilaste edetabeli“ disaini. Nimetatud projekti tulemuseks on võimalus koostada aruandeid õppijate õppetulemustest erinevate grupeerimisviiside, näiteks teaduskondade ja õppeainete, järgi.

PROJEKTI KIRJELDUS

Projekti „Üliõpilaste edetabel“ eesmärk on pakkuda tudengitele informatiivsemat tagasisidet nende õppetulemuste kohta lähtudes üldisest tasemest. Hetkel on Tartu Ülikoolil olemas õppekorralduse infovahetuskeskkond, Õppeinfosüsteem ÕIS, milles hoitakse igapäevast õppega seotud operatiivinfot, kuid mis ei sobi analüütilisteks päringuteks. Lisaks on arendatud andmeladu, mis küll sobib analüüsides ja aruannete koostamiseks, kuid mille andmekogud ja funktsionaalsus on piiratud.

Projekti tulemuseks olevad aruanded kujutavad endast päringuid andmelattu, mis tagastavad hulga õppetulemusi näiteks aine, õppekava või struktuuriüksuse kohta, mis omakorda on sorteeritud vastavalt tulemustele kindlas järjekorras, võimaldades organiseeritud ülevaadet.

Aruannete „tellimise“ kohaks on ette nähtud ÕIS, sest et selles infosüsteemis on juba olemas kasutajate õiguste süsteem. Kõigis kasutuslugudes on eeldatud, et ÕIS on liidestatud andmelaoga ja/või andmevakaga. Kõik kuvatavad tulemused on isikustamata, ainsana on sõltuvalt aruandest tulemusel tähistatud selle tellinud õppija tulemus.

DOKUMENDI SÕNASTIK

ÕIS – Tartu Ülikooli õppeinfosüsteem

KASUTUSLOOD

Siin on kirjeldatud peamised kasutuslood lähtudes ÕISist.

Üliõpilase järgi õppekavaga seotud tulemuste kuvamine

Tegija: tudeng

Üliõpilane klikib ÕISis lingile „Minu tulemused“. ÕIS edastab andmelaole antud päringutüübi koos õppija indikaatoriga. Andmeladu tagastab ÕISile sorteeritud tabeli õppetulemustega, kus on kajastatud kõik tudengiga sama aasta õppekavale sisseastunute tulemused kõigis ainetes, milles nad on tulemusi saanud (nii kohustuslikud, valik- kui ka vabaained). Tabeli sisu on sorteeritud vastavalt hinnetele nõnda, et kõige paremate tulemustega aine on esimeses reas ning kõige paremate tulemustega üliõpilane esimeses hindeveerus.

Üliõpilase järgi õppeaine tulemuste kuvamine aastate lõikes

Tegija: tudeng

Üliõpilane klikib ÕISis lingile „Minu ained“. Järgneval leheküljel on õppijal võimalik valida rippmenüüst kõiki õppeaineid, milles ta on siiani ülikoolis tulemuse omandanud. Aine valiku järel edastab ÕIS andmelaole antud päringutüübi koos tudengi ja valitud õppeaine indikaatoritega. Andmeladu tagastab ÕISile sorteeritud tabeli õppetulemustega, kus on kajastatud kõik valitud õppeaine raames välja antud hinded aastate lõikes. Hinded on kuvamisel sorteeritud iga õppeaasta jaoks gruppidesse parimast kõige kehvemani suunaga vasakult paremale.

Õppetulemuste kuvamine õppekava järgi

Tegija: ÕISi kasutaja (võib olla nii üliõpilane, õppejõud kui ülikooli töötaja)

Kasutaja klikib ÕISis lingile „Õppetulemused õppekava järgi“. Järgneval leheküljel on kasutajal võimalik valida rippmenüüdest õppekava täpsustavaid kriteeriume: teaduskond,

õppeaste, õppevorm, õppekava, aasta. Kriteeriumite valimise järel edastab ÕIS andmelaole antud päringutüübi koos valitud indikaatoritega. Andmeladu tagastab ÕISile sorteeritud tabeli õppetulemustega, kus on kajastatud kõik antud õppekavaga seotud õppeained koos väljastatud hinnetega. Tulemused on sorteeritud hinnete järgi. Seega on parimate tulemustega õppeaine esimeses reas ning veergudes on hinded tähe järgi grupeerituna järjestatud parimast kõige kehvemani suunaga vasakult paremale.

Õppetulemuste kuvamine õppeaine järgi

Tegija: ÕISi kasutaja (võib olla nii üliõpilane, õppejõud kui ülikooli töötaja)

Kasutaja klikib ÕISis lingile „Õppetulemused õppeaine järgi“. Järgneval leheküljel on kasutajal võimalik valida rippmenüüdest õppeainet täpsustavaid kriteeriume: teaduskond, instituut, õppeaine. Kriteeriumite valimise järel edastab ÕIS andmelaole antud päringutüübi koos valitud indikaatoritega. Andmeladu tagastab ÕISile sorteeritud tabeli õppetulemustega, kus on kajastatud kõik valitud õppeaine raames välja antud hinded aastate lõikes. Hinded on kuvamisel sorteeritud iga õppeaasta jaoks gruppidesse parimast kõige kehvemani suunaga vasakult paremale.

Lisa 2. Andmevaka disain

TARTU ÜLIKOOL

Üliõpilaste edetabel

Andmevaka tehnilise disaini dokument

TARTU 2014

SISSEJUHATUS

Antud dokumendi eesmärgiks on kirjeldada projekti „Üliõpilaste edetabeli“ andmevaka tehnilist disaini. Nimetatud projekti tulemuseks on võimalus koostada aruandeid õppijate õppetulemustest erinevate grupeerimisviiside, näiteks teaduskondade ja õppeainete, järgi. Tehnilise infona sisaldab käesolev dokument nimekirja andmevaka andmeallikatest, dimensioonidest, faktidest ning ülekantavate andmete töötlemisreeglitest.

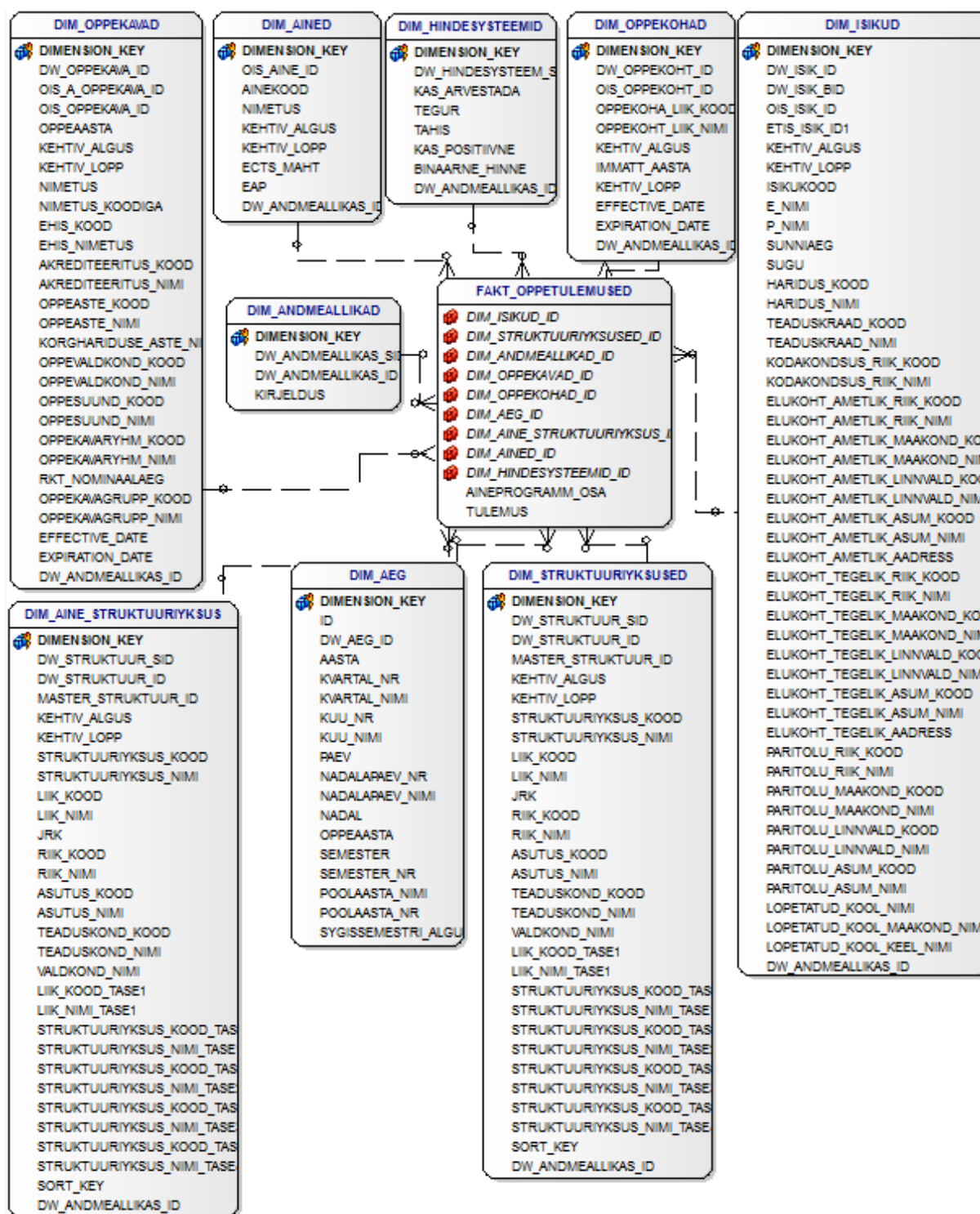
ANDMEALLIKAD

Üheks andmeallikaks on andmeladu ise, mis sisaldab juba järgnevaid dimensioone: DIM_OPPEKOHAD, DIM_ISIKUD, DIM_OPPEKAVAD, DIM_AEG, DIM_ANDMEALLIKAD ja DIM_STRUKTUURIYKSUSED. Neist saadakse andmed vastavalt õppekohtade, üliõpilaste, õppekavade (nii üldised kui ka aasta õppekavad), aja ja teaduskondade-instituutide kohta.

Andmelaos juba olemasolevate dimensioonide kohta ei esitata ülekantavate andmete töötlemisreegleid, need on olemas failis *TÜ andmelao disain.doc*. Nende tabelite kohta tuuakse välja ainult nende struktuur. Nimetatud käitumine kehtib ka dimensiooni DIM_AINE_STRUKTUURIYKSUS kohta, mis on andmelaonduse reeglist, et faktitabel võib olla ühe dimensiooniga seotud ainult ühe korra, tulenevalt koopida dimensioonist DIM_STRUKTUURIYKSUSED.

Teiseks andmeallikaks on Õppeinfosüsteem ÕIS, millest saadakse andmed õppetulemuste, hindesüsteemide, õppeainete ja ainekavade kohta. Neile vastavad dimensioonid DIM_HINDESYSTEMID ja DIM_AINED ning faktitabel FAKT_OPPETULEMUSED.

ANDMEMUDEL



DIMENSIOONID

Tabel DIM_AEG

Tabel DIM_AEG sisaldab andmelao ajadimensiooni.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti.
id		number		Ajadimensiooni identifikaator andmelaos.
dw_aeg_id	Loogiline võti	date		Aja identifikaator päeva tasemel. Võtmeväli.
aasta	I	number(4)		Neljakohtaline aastaarv.
kvartal_nr	I	number(2)		Kvartali järjekorranumber.
kvartal_nimi	I	varchar2(20)		Kvartali nimetus (näiteks I kvartal).
kuu_nr	I	number(2)		Kuu järjekorranumber (arvuline).
kuu_nimi	I	varchar2 (20)		Kuu nimetus.
paev	I	number(2)		Kuupäev (arvuline).
nadalapaev_nr	I	number(1)		Nädalapäeva järjekorranumber (1..7). Number ühele vastab esmaspäev.
nadalapaev_nimi	I	varchar2 (20)		Nädalapäeva nimetus.
nadal	I	number(2)		Kalendrinädala järjekorranumber.
oppeaasta	I	varchar2(30)		Õppeaasta nimetus (YYYY/YYYY õppeaasta).
semester	I	varchar2(10)		Semestri nimetus (Kevad, Sügis).
semester_nr	I	number		Semestri järjekorranumber (1 – Sügis, 2 –Kevad).
poolaasta_nimi	I	varchar2(20)		Poolaasta nimi (näiteks I poolaasta)
poolaasta_nr	I	number(1)		Poolaasta number
sygisselestri_algus	I	date		Sügissemestri alguskuupäev

Tabel DIM_ANDMEALLIKAD

Tabel DIM_ANDMEALLIKAD sisaldab andmeallikate dimensiooni.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti.
dw_andmeallikas_sid		number		Andmeallika identifikaator andmelaos.
dw_andmeallikas_id	Loogiline võti	integer		Andmeallika kood. 0 – puudub, -1 – teadmata, 1 – andmeladu, 2 – Õppeinfosüsteem (ÕIS), 3 – Kinnisvara haldamise süsteem (Archibus), 4 – Raamatupidamissüsteem (Axapta)
kirjeldus	I	varchar2(100)		Andmeallika kirjeldus.

Tabel DIM_ISIKUD

Tabel DIM_ISIKUD sisaldab TÜ töötajate ja üliõpilaste ja lisaks TÜ-ga mitteseotud, kuid publikatsioonidega seotud isikute andmeid.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus	Lähteallikas
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti.	
dw_isik_id		number(22)		Isiku surrogaatvõti.	
dw_isik_bid	Loogiline võti	varchar2(26)		Isiku identifikaator andmelaos.	
ois_isik_id	I	number(22)		Vastava isiku identifikaator ÕIS-is.	ois.ir.ir_isik.id_isik
etis_isik_id	I	number(22)		Vastava isiku identifikaator ETISes.	ut_dwh_stage.etis_publications.person_id
kehtiv_algus		date		Isiku versiooni kehtivuse algus.	
kehtiv_lopp		date		Isiku versiooni kehtivuse lõpp.	

isikukood	I	varchar2 (20)		Isikukood	ois.ir.ir_isik.isikukood / ut_dwh_stage.etis_public_au torid.p_id_code
e_nimi	II	varchar2 (100)		Isiku eesnimi	ois.ir.ir_isik.e_nimi / ut_dwh_stage.etis_public_au torid.p_first_name
p_nimi	II	varchar2 (100)		Isiku perekonnanimi	ois.ir.ir_isik.p_nimi / ut_dwh_stage.etis_public_au torid.p_last_name
sünniaeg	II	date		Isiku sünniaeg	ois.ir.ir_isik.synniaeg / ut_dwh_stage.etis_public_au torid.p_date_of_birth
Sugu	II	number(1)		Isiku sugu	ois.ir.ir_isik.sugu
haridus_kood	II	number(22)		Isiku haridus (kood)	saadakse ois.ir.ir_haridus.kl_liik põhjal
haridus_nimi	II	varchar2(20)		Isiku haridus (nimetus)	
teaduskraad_kood	II	number(22)		Isiku teaduskraad (kood)	saadakse ois.ir.ir_kraad.kl_kraa di_liik põhjal
teaduskraad_nimi	II	varchar2(20)		Isiku teaduskraad (nimetus)	
kodakondsus_riik_kood	II	char(3)		Isiku kodakondsusriigi kood	ois.ir.ir_maad.iso3
kodakondsus_riik_nimi	II	varchar2(50)		Isiku kodakondsusriigi nimi	ois.ir.ir_maad.nimi
elukoht_ametlik_riik_kood	II	char(3)		Isiku ametliku elukohariigi kood	ois.ir.ir_maad_iso3
elukoht_ametlik_riik_nimi	II	varchar2(50)		Isiku ametliku elukohariigi nimetus	ois.ir.ir_maad.nimi
elukoht_ametlik_maakond_kood	II	number(2)		Isiku ametliku elukoha maakonna kood	ois.ir.ir_asumikood. id_maakond
elukoht_ametlik_maakond_nimi	II	varchar2(50)		Isiku ametliku elukoha maakonna nimetus	ois.ir.ir_asumikood. nimi
elukoht_ametlik_linnvald_kod	II	number(2)		Isiku ametliku elukoha linna/valla kood	ois.ir_asumikood.id_v ald
elukoht_ametlik_linnvald_nimi	II	varchar2(100)		Isiku ametliku elukoha linna/valla nimetus	ois.ir.ir_asumikood. nimi

elukoht_ametlik_asum_kood	II	number(22)		Isiku ametliku elukoha asumi kood	ois.ir.ir_asumikood.asumikood
elukoht_ametlik_asum_nimi	II	varchar2(100)		Isiku ametliku elukoha asumi nimetus	ois.ir.ir_asumikood.nimi
elukoht_ametlik_aadress	II	varchar2(300)		Isiku ametlik aadress	ois.ir.ir_elukoht.aadress
elukoht_tegelik_riik_kood	II	char(3)		Isiku tegeliku elukohariigi kood	ois.ir.ir_maad_iso3
elukoht_tegelik_riik_nimi	II	varchar2(50)		Isiku tegeliku elukohariigi nimetus	ois.ir.ir_maad.nimi
elukoht_tegelik_maakond_kood	II	number(2)		Isiku tegeliku elukoha maakonna kood	ois.ir.ir_asumikood.id_maakond
elukoht_tegelik_maakond_nimi	II	varchar2(50)		Isiku tegeliku elukoha maakonna nimetus	ois.ir.ir_asumikood.nimi
elukoht_tegelik_linnvald_kood	II	number(2)		Isiku ametliku elukoha linna/valla kood	ois.ir.asumikood.id_vald
elukoht_tegelik_linnvald_nimi	II	varchar2(100)		Isiku ametliku elukoha linna/valla nimetus	ois.ir.ir_asumikood.nimi
elukoht_tegelik_asum_kood	II	number(22)		Isiku tegeliku elukoha asumi kood	ois.ir.ir_asumikood.asumikood
elukoht_tegelik_asum_nimi	II	varchar2(100)		Isiku tegeliku elukoha asumi nimetus	ois.ir.ir_asumikood.nimi
elukoht_tegelik_aadress	II	varchar2(300)		Isiku tegelik aadress	ois.ir.ir_elukoht.aadress
paritolu_riik_kood	II	char(3)		Isiku päritoluriigi kood	ois.ir.ir_maad.iso3
paritolu_riik_nimi	II	varchar2(50)		Isiku päritoluriigi nimi	ois.ir.ir_maad.nimi
paritolu_maakond_kood	II	number(2)		Isiku päritolukoha maakonna kood	ois.ir.ir_asumikood.id_maakond
paritolu_maakond_nimi	II	varchar2(100)		Isiku päritolukoha maakonna nimi	ois.ir.ir_asumikood.nimi
paritolu_linnvald_kood	II	number(2)		Isiku ametliku elukoha linna/valla kood	ois.ir.asumikood.id_vald
paritolu_linnvald_nimi	II	varchar2(100)		Isiku ametliku elukoha linna/valla nimetus	ois.ir.ir_asumikood.nimi
paritolu_asum_kood	II	number(22)		Isiku päritolukoha asumi kood	ois.ir.ir_asumikood.asumikood
paritolu_asum_nimi	II	varchar2(100)		Isiku päritolukoha asumi nimi	ois.ir.ir_asumikood.nimi
lopetatud_kool_nimi	II	varchar2(150)		Isiku poolt lõpetatud	ois.ir.ir_asutus.nimi

				kooli nimi	
lopetatud_kool_maakond_nimi	II	varchar2(100)		Isiku poolt lõpetatud kooli asukoha- maakonna nimi	ois.ir.ir_asumikood.nimi
lopetatud_kool_keel_nimi	II	varchar2(50)		Isiku poolt lõpetatud kooli õppekeel	
dw_andmeallikas_id	II	numeric(16)		Viide andmeallikate dimensioonile (koodile). Näitab, millisest andmeallikast isiku (vastava versiooni) andmed pärinevad.	

Tabel DIM_OPPEKAVAD

Tabel DIM_OPPEKAVAD sisaldab õppekavade andmeid.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus	Lähteallikas
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti	
dw_oppekava_id		number		Õppekava identifikaator andmelaos.	
ois_a_oppekava_id	Loogiline võti	number		Aasta õppekava identifikaator ÕISis.	ois.ok.ok_a_oppekava. id_a_oppekava
ois_oppekava_id	II	number		Õppekava püsiantmete identifikaator ÕISis.	ois.ok.ok_oppekava. id_oppekava
oppeaasta	II	number		Õppekava alguse aasta	ois.ok.ok_a_oppekava. kl_kehtivusaasta
kehtiv_algus		date		Õppekava versiooni kehtivuse algus.	
kehtiv_lopp		date		Õppekava versiooni kehtivuse lõpp.	
nimetus	II	varchar2(200)		Õppekava nimetus.	ois.ok.ok_oppekava. nimetus_ek
nimetus_koodiga	II	varchar2(200)		Õppekava nimetus koos	ois.ok.ok_oppekava.

				HM_kood_6-ga	nimetus_ek ois.ok.ok_oppekava. hm_kood_6
ehis_kood	II	varchar2(3)		EHISe tase (kood).	ois.ok.ok_oppekava. ehis_tase
ehis_nimetus	II	varchar2(100)		EHISe taseme nimetus.	ois.admin.a_klass.nimetus
akrediteeritus_kood	II	number		Akrediteerimise tunnus (kood).	ois.ok.ok_oppekava. kl_akrediteerimine
akrediteeritus_nimi	II	varchar2(50)		Akrediteerimise tunnus (nimetus).	ois.ir.ir_klassifikaator. nimetus
oppeaste_kood	II	number		Õppeastme kood	ois.ok.ok_oppekava. kl_oppeaste
oppeaste_nimi	II	varchar2(50)		Õppeastme nimetus	ois.ir.ir_klassifikaator. nimetus
korghariduse_aste_nimi	II	varchar2(50)		Kõrghariduse astme nimetus	
oppevaldkond_kood	II	number		Õppevaldkonna kood	ois.ok.ok_oppekava.kl_oppevaldkond
oppevaldkond_nimi	II	varchar2(50)		Õppevaldkonna nimetus	ois.admin.a_klass.nimetus
oppesuund_kood	II	number		Õppesuuna kood	ois.ok.ok_oppekava.kl_oppesuund
oppesuund_nimi	II	varchar2(50)		Õppesuuna nimetus	ois.admin.a_klass.nimetus
oppekavaryhm_kood	II	number		Õppekava rühma kood.	ois.ok.ok_oppekava.kl_oppekavaryhm
oppekavaryhm_nimi	II	varchar2(50)		Õppekava rühma nimetus	ois.admin.a_klass.nimetus
rkt_nominaalaeg	II	numeric		Õppekava nominaalaja pikkus aastates	ois.ok.ok_oppekava.kestus
oppekavagrupp_kood	II	number		Õppekavagrupi kood	ois.ok.ok_oppekava.kl_oppekavagrupp
oppekavagrupp_nimi	II	varchar2(50)		Õppekavagrupi nimetus	ois.admin.a_klass.nimetus
effective_date		date			
expiration_date		date			
dw_andmeallikas_id		number		Viide andmeallikate dimensioonile (koodile).	

				Näitab, millisest andmeallikast õppekavade andmed pärinevad.	
--	--	--	--	--------------------------------------------------------------	--

Tabel DIM_OPPEKOHAD

Tabel DIM_OPPEKOHAD sisaldab õppekohtade andmeid.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus	Lähteallikas
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti	
dw_oppekoht_id		number		Õppekoha identifikaator andmelaos.	
ois_oppekoht_id	Loogiline võti	number		Õppekoha identifikaator ÕISis.	ois.oh.oh_isiku_oppek oht. id_isiku_oppekoht
oppekoha_liik_kood	II	number		Õppekoha liigi kood ÕISis.	ois.oh.oh_isiku_oppek oht. kl_liik
oppekoha_liik_nimi	II	varchar2(100)		Õppekoha liigi nimetus.	ois.ir.ir_klassifikaator.nimetus
kehtiv_algus		date		Õppekoha immatrikuleerimise kuupäev	ois.oh.oh_isiku_oppek oht. algus_kp
immatt_aasta		number		Õppekoha immatrikuleerimise aasta	
kehtiv_lopp		date			ois.oh.oh_isiku_oppek oht. lopp_kp
effective_date		date			
expiration_date		date			
dw_andmeallikas_id	II	number		Viide andmeallikate dimensioonile (koodile). Näitab, millisest andmeallikast õppekoha andmed pärinevad.	

Tabel DIM_STRUKTUURIYKSUSED

Tabel DIM_STRUKTUURIYKSUSED sisaldab struktuuriüksuste dimensiooni andmeid. Tabel DIM_AINE_STRUKTUURIYKSUS on täpselt sama struktuuriga, mis tabel DIM_STRUKTUURIYKSUSED ning seda kasutatakse faktitabelis selleks, et määrata struktuuriüksust, mille alla õppeaine kuulub. (Põhjus: fakt saab olla seotud sama dimensiooniga ainult üks kord.) Samuti on mõlemas tabelis asuvad andmed identsed ning info lisamine ja muutmine toimub samade reeglite alusel.

Väli	Tüüp	Tüüp	Täitmise kohustus	Kirjeldus	Lähteallikas
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti	
dw_struktuur_sid		number		Struktuuriüksuse surrogaatvõti	
dw_struktuur_id	Loogiline võti	number		Struktuuriüksuse identifikaator andmelaos.	ois.ir.ir_struktuur.id_struktuur
master_struktuur_id		number		Viide ülemüksusele.	ois.ir.ir_struktuur.id_master_struktuur
kehtiv_algus		date		struktuuriüksuse andmete kehtivuse algus	ois.ir.ir_struktuur.algus
kehtiv_lopp		date		struktuuriüksuse andmete kehtivuse lõpp	ois.ir.ir_struktuur.lopp
struktuuriyksus_kood		varchar2(20)		struktuuriüksuse kood	ois.ir.ir_struktuur.kood
struktuuriyksus_nimi		varchar2(150)		struktuuriüksuse nimi	ois.ir.ir_struktuur.nimetus
liik_kood		number		struktuuriüksuse liik	ois.ir.ir_struktuur.liik
liik_nimi		varchar2(40)		struktuuriüksuse liigi nimi	ois.ir.ir_klassifikaator.Nimetus
jrk		number		Järjekorranumber samal tasemel sama ülemüksuse korral	ois.ir.ir_struktuur.jrk
riik_kood		varchar2(20)		Riigi kood, mille alla struktuuriüksus	

				kuulub	
riik_nimi		varchar2(100)		Riigi nimi, mille alla struktuuriüksus kuulub	
asutus_kood		varchar2(20)		Asutuse (ülikooli) kood, mille alla struktuuriüksus kuulub	
asutus_nimi		varchar2(100)		Asutuse (ülikooli) nimi, mille alla struktuuriüksus kuulub	
teaduskond_kood		varchar2(20)		teaduskonna või kolledži kood, mille alla struktuuriüksus kuulub	ois.ir.ir_struktuur.kood
teaduskond_nimi		varchar2(150)		teaduskonna või kolledži nimi, mille alla struktuuriüksus kuulub	ois.ir.ir_struktuur.nimetus
valdkond_nimi		varchar2(20)		Struktuuriüksuse kuulumine teadusvaldkona	
dw_andmeallikas_id		number		Viide andmeallikate dimensioonile (koodile). Näitab, millisest andmeallikast struktuuriüksuse andmed pärinevad.	
liik_kood_tase1		number		struktuuriüksuse esimese taseme liigi kood	
liik_nimi_tase1		varchar2(40)		struktuuriüksuse esimese taseme liigi nimi	
struktuuriyksus_kood_tase1		varchar2(20)		struktuuriüksuse esimese taseme kood	
struktuuriyksus_nimi		varchar2(150)		struktuuriüksuse	

_tase1				esimese taseme nimi	
struktuuriyksus_kood _tase2		varchar2(20)		struktuuriüksuse teise taseme kood	
struktuuriyksus_nimi _tase2		varchar2(150)		struktuuriüksuse teise taseme nimi	
struktuuriyksus_kood _tase3		varchar2(20)		struktuuriüksuse kolmanda taseme kood	
struktuuriyksus_nimi _tase3		varchar2(150)		struktuuriüksuse kolmanda taseme nimi	
struktuuriyksus_kood _tase4		varchar2(20)		struktuuriüksuse neljanda taseme kood	
struktuuriyksus_nimi _tase4		varchar2(150)		struktuuriüksuse neljanda taseme nimi	
sort_key		varchar2(100)		Sorteerimise võti 1 tase kood, 2-4 tase jrk	

Tabel DIM_HINDESYSTEEMID

Tabel DIM_HINDESYSTEEMID sisaldab hinnete taustsüsteemi andmeid. Tabeli aluseks on ÕISI tabel ER_HINNE.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus	Lähteallikas
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti	
dw_hindesysteem_sid		number		Hindesüsteemi surrogaatvõti	ois.er_hinne.id_hinne
kas_arvestada		number		Boolean-tüüpi tunnus keskmise hinde arvutamisse	ois.er_hinne.kas_arvestada

				kaasamiseks	
tegur		number		Hinde väärtus keskmise hinde arvutamisel	ois.er_hinne.tegur
tahis		varchar2(50)		Hinde kirjalik tähis visuaalsel kuvamisel kasutamiseks	ois.er_hinne.tahis
kas_positiivne		number		Boolean-tüüpi hinde positiivust/negatiivsus tähistav tunnus	ois.er_hinne.kas_positiivne
binaarne_hinne		number		Boolean-tüüp tunnus; näitab, kas tegemist on kahe või rohkema väärtusega hindegaga	er.er_hindesysteem.kl_kontrollivorm
dw_andmeallikas_id		number		Viide andmeallikate dimensioonile (koodile). Näitab, millisest andmeallikast õppevormi andmed pärinevad.	

Hindesüsteemi dimensiooni laadimine:

- 1) Tabelist ER_HINNE võetakse väljad *id_hinne*, *kas_arvestada*, *tegur*, *kas_positiivne*, *aeg*, *tahis* ja *id_hindesysteem*. Väli *aeg* nimetatakse ümber väljaks *er_hinne_aeg* ning väli *id_hinne* ümber väljaks *dw_hindesysteem_sid*.
- 2) Tabelist ER_HINDESYSTEEM võetakse väljad *id_hindesysteem*, *kl_kontrollivorm* ja *aeg*. Väli *aeg* nimetatakse ümber väljaks *er_hindesysteem_aeg* ning väli *kl_kontrollivorm* väljaks *binaarne_hinne*.
- 3) Väljale *binaarne_hinne* rakendatakse teisendusreeglit: kui *binaarne_hinne* = 2, siis *binaarne_hinne* = 1; muudel juhtudel *binaarne_hinne* = 0;
- 4) Punktis 1 saadud tabel ühendatakse punktis 2 saadud tabeliga (seos *ER_HINNE.id_hindesysteem* = *ER_HINDESYSTEEM.id_hindesysteem*) ning võetakse kirjed, kus välja *er_hinne_aeg* väärtus rahuldab tingimust (eelmine tabeli ER_HINNE

andmelattu laadimise kuupäev) <= er_hinne_aeg < (andmelao käesoleva laadimise kuupäev) või välja er_hindesysteem_aeg väärtus rahuldab tingimust (eelmine tabeli ER_HINDESYSTEEM andmelattu laadimise kuupäev) <= er_hindesysteem_aeg < (andmelao käesoleva laadimise kuupäev).

- 5) Saadud tabelile lisatakse väli dw_andmeallikas_id, mille väärtuseks saab 2 (ÕIS).
- 6) Saadud tabel ühendatakse (merge) andmelao tabeliga DIM_HINDESYSTEEMID (seos <punktis 4 saadud tabel>.dw_hindesysteem_sid = DIM_HINDESYSTEEMID.dw_hindesysteem_sid).

Hinnete taustsüsteemi dimensiooni andmelattu laadimisel kasutatakse aeglaselt muutuva dimensiooni (*Slowly Changing Dimension*) tüüpi nr 1 (andmete muutumisel kirjutatakse olemasoleva kirje väärtused üle).

Tabel DIM_AINED

Tabel DIM_AINED sisaldab õppeainete andmeid. Tabeli aluseks on ÕISi tabelid OA_REGISTER ja OA_NIMETUS.

Väli	Tüüp	Andmetüüp	Täitmise kohustus	Kirjeldus	Lähteallikas
dimension_key	Surrogaatvõti	number	kohustuslik	Automaatselt genereeritud dimensiooni võti	
ois_aine_id	Loogiline võti	number		Õppeaine identifikaator ÕISis	ois.oa_register.id_register
ainekood		varchar2(12)		Õppeaine kood	ois.oa_register.ainekood
nimetus		varchar2(255)		Õppeaine nimetus	ois.oa_nimetus.nimetus
kehtiv_algus		date		Õppeaine versiooni kehtivuse algus	ois.oa_nimetus.algus
kehtiv_lopp		date		Õppeaine versiooni kehtivuse lõpp	ois.oa_nimetus.lopp
ects_maht		number		Õppeaine maht ECTS punktides. Kasutusel kuni 30.08.2009	ois.oa_register.ects_maht
eap		number		Õppeaine maht	ois.oa_register.eap

				Euroopa ainepunktides. Kehtiv alates 31.08.2009	
dw_andmeallikas_id		number		Viide andmeallikate dimensioonile (koodile). Näitab, millisest andmeallikast õppevormi andmed pärinevad.	

Märkused:

1. Antud tabelis võib ühest õppeainest olla mitu versiooni.
2. Õppeaine uus versioon tekib siis, kui vähemalt üks õppekava atribuutidest muutub.

Aine dimensiooni laadimine:

- 1) Tabelist OA_REGISTER võetakse väljad *id_register*, *ainekood*, *ects_maht*, *eap*, *aeg*. Väli *aeg* nimetatakse ümber väljaks *oa_register_aeg* ning väli *id_register* väljaks *ois_aine_id*.
- 2) Tabelist OA_NIMETUS võetakse väljad *id_register*, *nimetus*, *algus*, *lopp* ja *aeg*. Väli *aeg* nimetatakse ümber väljaks *oa_nimetus_aeg*, väli *algus* väljaks *kehtiv_algus*, väli *lopp* väljaks *kehtiv_lopp*.
- 3) Punktis 1 saadud tabel ühendatakse punktis 2 saadud tabeliga (seos $OA_REGISTER.id_register = OA_NIMETUS.id_register$) ning võetakse kirjed, kus välja *oa_register_aeg* väärtus rahuldab tingimust (eelmine tabeli OA_REGISTER andmelattu laadimise kuupäev) $\leq oa_register_aeg < (andmelao\ käesoleva\ laadimise\ kuupäev)$ või välja *oa_nimetus_aeg* väärtus rahuldab tingimust (eelmine tabeli OA_NIMETUS andmelattu laadimise kuupäev) $\leq oa_nimetus_aeg < (andmelao\ käesoleva\ laadimise\ kuupäev)$.
- 4) Saadud tabelile lisatakse väli *dw_andmeallikas_id*, mille väärtuseks saab 2 (ÕIS).
- 5) Saadud tabel liidetakse (insert) andmelao tabeliga DIM_AINED.

Õppeainete dimensiooni andmelattu laadimisel kasutatakse aeglaselt muutuva dimensiooni (*Slowly Changing Dimension*) tüüpi nr 2 (andmete muutumisel tekib kirje uus versioon).

FAKTID

Tabel FAKT_OPPETULEMUSED

Tabel FAKT_OPPETULEMUSED sisaldab kõiki õppetulemuste fakte, iga rida tähistab ühte ÕISI sissekantud hinnet.

Väli	Tüüp	Andme- tüüp	Täitmise kohustus	Võti	Kirjeldus
dim_andmeallikad_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_ANDME ALLIKAD	Näitab, millisest andmeallikast andmed pärinevad.
dim_aeg_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_AEG	
dim_isikud_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_ISIKUD	Üliõpilase ID andmelaos.
dim_oppekohad_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_OPPEKOHAD	Õppekoha ID andmelaos.
dim_oppekavad_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_OPPEKAVAD	Õppekava ID andmelaos.
dim_struktuuriüksuse d_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_STRUKTUURI YKSUSED	Struktuuriüksuse ID andmelaos.
dim_aine_struktuuriy ksus_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_AINE_STRUKT UURIYKSUS	Õppeaine struktuuriüksuse ID andmelaos.
dim_ained_id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_AINED	Õppeaine ID andmelaos
dim_hindesüsteemid_ id	Dimensioon	number	kohustuslik	Võõrvõti tabelisse DIM_HINDESYSTEE MID	Hindesüsteemi ID andmelaos
aineprogramm_osa		number		ois.er_tulemus.osa	Aine osa, juhul kui õppeaine on jaotatud osadeks
tulemus	Eristav mõõdik/ identifikaator	number	kohustuslik	ois.er_tulemus.id_tule mus	Faktiga seotud õppetulemuse identifikaator

Õppetulemuste fakti laadimine:

1. Tabelist ER_TULEMUS võetakse väljad *tulemuse_kp*, *id_tudeng*, *id_hinne*, *aeg*, *osa*, *id_register*, *id_tulemus* ning kirjed, kus välja *aeg* väärtus rahuldab tingimust (eelmine tabeli ER_TULEMUS andmelattu laadimise kuupäev) $\leq aeg <$ (andmelao käesoleva laadimise kuupäev). Väli *id_tulemus* nimetatakse ümber väljaks *tulemus*.
2. Punktis 1 saadud tabelile lisatakse tabelist ER_TULEMUS_OH_OPPEKOHT seose ER_TULEMUS.*id_tulemus* = ER_TULEMUS_OH_OPPEKOHT.*id_tulemus* alusel väli *id_isiku_oppekoht*.
3. Punktis 2 saadud tabelile lisatakse tabelist OA_REGISTER väli *id_struktuur* seose ER_TULEMUS.*id_register* = OA_REGISTER.*id_register* alusel.
4. Punktis 3 saadud tabelile lisatakse tabelist OH_OLEK väljad *id_a_oppekava*, *id_struktuur* vastavalt seostele:
 - 4.1. ER_TULEMUS.*tulemuse_kp* BETWEEN OH_OLEK.*algus_kp* AND OH_OLEK.*lopp_kp* (valitakse tulemuse hetkel kehtinud põhiolek)
 - 4.2. ER_TULEMUS.*id_tudeng* = OH_OLEK.*id_isik*
 - 4.3. OH_OLEK.*kl_olek* < 100 AND OH_OLEK.*kl_olek* != 7 (valitakse ainult põhiolekute hulgast)
 - 4.4. <punktis 2 saadud tabel>.*id_isiku_oppekoht*= OH_OLEK. *id_isiku_oppekoht*
5. Punktis 4 saadud tabelile lisatakse (viited dimensioonitabelitele):
 - 5.1. andmelao tabelist DIM_AEG väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_aeg_id*** (seos <punktis 4 saadud tabel>.*tulemuse_kp* = DIM_AEG.*dw_aeg_id*).
 - 5.2. andmelao tabelist DIM_ISIKUD väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_isikud_id*** (seos <punktis 4 saadud tabel>.*id_tudeng* = DIM_ISIKUD.*ois_isik_id*).
 - 5.3. andmelao tabelist DIM_OPPEKOHAD väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_oppekohad_id*** (seos <punktis 4 saadud tabel>.*id_isiku_oppekoht* = DIM_OPPEKOHAD.*ois_oppekoht_id*);
 - 5.4. andmelao tabelist DIM_STRUKTUURIYKSUSED väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_struktuuriyksused_id*** (seos <punktis 4 saadud tabel>.*id_struktuur* = DIM_STRUKTUURIYKSUSED.*dw_struktuur_id*);
 - 5.5. andmelao tabelist DIM_OPPEKAVAD väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_oppekavad_id*** (seos <punktis 4 saadud tabel>.*id_a_oppekava* = DIM_OPPEKAVAD.*ois_a_oppekava_id*);

- 5.6. andmelao tabelist DIM_AINE_STRUKTUURIYKSUS väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_aine_struktuuriyksus_id*** (seos <punktis 4 saadud tabel>.id_struktuur = DIM_STRUKTUURIYKSUSED.dw_struktuur_id);
- 5.7. andmelao tabelist DIM_AINED väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_ained_id*** (seos <punktis 4 saadud tabel>.id_register = DIM_AINED.ois_aine_id);
- 5.8. andmelao tabelist DIM_HINDESYSTEEMID väli *dimension_key*, mis nimetatakse ümber väljaks ***dim_hindesysteemid_id*** (seos <punktis 4 saadud tabel>.id_hinne = DIM_HINDESYSTEEMID.dw_hindesysteem_sid);
- 5.9. Saadud tabelile lisatakse fikseeritud väärtusega väli ***dim_andmeallikad_id***, mille väärtuseks saab tabelist DIM_ANDMEALLIKAD väli *dimension_key* (seos <punktis 4 saadud tabel>.dw_andmeallikas_id = 2). Andmeallikaks on kõikjal ÕIS.
6. Saadud tabel ühendatakse (merge) andmelao tabeliga FAKT_OPPETULEMUSED kõigi dimensioonide järgi. Kui andmelaos leidub juba vastav kirje (välja *tulemus* väärtus on sama), siis uuendatakse vastavalt muutustele dimensioonide väärtusi.

Lisa 3. Prototüübiga seotud SQL laused

TABELITE LOOMINE - Dimensioonid

SQL 1. Ainete dimensioonitabel

```
CREATE TABLE dim_ained (  
    dimension_key NUMBER NOT NULL,  
    ois_aine_id NUMBER,  
    ainekood VARCHAR2(12),  
    nimetus VARCHAR2(255),  
    kehtiv_algus DATE,  
    kehtiv_lopp DATE,  
    ects_maht NUMBER,  
    eap NUMBER,  
    dw_andmeallikas_id NUMBER,  
    CONSTRAINT pk_dim_ained PRIMARY KEY (dimension_key)  
);
```

SQL 2. Hindesüsteemide dimensioonitabel

```
CREATE TABLE dim_hindesysteemid (  
    dimension_key NUMBER NOT NULL,  
    dw_hindesysteem_sid NUMBER,  
    kas_arvestada NUMBER(1),  
    tegur NUMBER(1),  
    tahis VARCHAR2(50),  
    kas_positiivne NUMBER(1),  
    binaarne_hinne NUMBER(1),  
    dw_andmeallikas_id NUMBER,  
    CONSTRAINT pk_dim_hindesysteemid PRIMARY KEY (dimension_key)  
);
```

SQL 3. Aine struktuuriüksuste dimensioonitabel

```
CREATE TABLE dim_aine_struktuuriyksus (  
    dimension_key NUMBER NOT NULL,  
    dw_struktuur_sid NUMBER,  
    dw_struktuur_id NUMBER,  
    master_struktuur_id NUMBER,
```



```

kehtiv_algus DATE,
kehtiv_lopp DATE,
struktuuriyksus_kood VARCHAR2(20),
struktuuriyksus_nimi VARCHAR2(150),
liik_kood NUMBER,
liik_nimi VARCHAR2(40),
jrk NUMBER,
riik_kood VARCHAR2(20),
riik_nimi VARCHAR2(100),
asutus_kood VARCHAR2(20),
asutus_nimi VARCHAR2(100),
teaduskond_kood VARCHAR2(20),
teaduskond_nimi VARCHAR2(150),
valdkond_nimi VARCHAR2(20),
liik_kood_tase1 NUMBER,
liik_nimi_tase1 VARCHAR2(40),
struktuuriyksus_kood_tase1 VARCHAR2(20),
struktuuriyksus_nimi_tase1 VARCHAR2(150),
struktuuriyksus_kood_tase2 VARCHAR2(20),
struktuuriyksus_nimi_tase2 VARCHAR2(150),
struktuuriyksus_kood_tase3 VARCHAR2(20),
struktuuriyksus_nimi_tase3 VARCHAR2(150),
struktuuriyksus_kood_tase4 VARCHAR2(20),
struktuuriyksus_nimi_tase4 VARCHAR2(150),
sort_key VARCHAR2(100),
dw_andmeallikas_id NUMBER,
CONSTRAINT pk_dim_aine_struk PRIMARY KEY (dimension_key)
);

```

TABELITE LOOMINE - Fakt

SQL 4. Õppetulemuste faktitabel

```

CREATE TABLE fakt_oppetulemused (
  dim_isikud_id NUMBER,
  dim_struktuuriyksused_id NUMBER,
  dim_andmeallikad_id NUMBER,
  dim_oppekavad_id NUMBER,
  dim_oppekohad_id NUMBER,
  dim_aeg_id NUMBER,
  dim_aine_struktuuriyksus_id NUMBER,

```

```

dim_ained_id NUMBER,
dim_hindesyteemid_id NUMBER,
aineprogramm_osa NUMBER,
tulemus NUMBER,
    CONSTRAINT fk_dim_isikud FOREIGN KEY (dim_isikud_id)
        REFERENCES dim_isikud (dimension_key),
    CONSTRAINT fk_dim_struktuuriyksused
        FOREIGN KEY (dim_struktuuriyksused_id)
        REFERENCES dim_struktuuriyksused (dimension_key),
    CONSTRAINT fk_dim_andmeallikad FOREIGN KEY (dim_andmeallikad_id)
        REFERENCES dim_andmeallikad (dimension_key),
    CONSTRAINT fk_dim_oppekavad FOREIGN KEY (dim_oppekavad_id)
        REFERENCES dim_oppekavad (dimension_key),
    CONSTRAINT fk_dim_oppekohad FOREIGN KEY (dim_oppekohad_id)
        REFERENCES dim_oppekohad (dimension_key),
    CONSTRAINT fk_dim_aeg FOREIGN KEY (dim_aeg_id)
        REFERENCES dim_aeg (dimension_key),
    CONSTRAINT fk_dim_aine_struk FOREIGN KEY (dim_aine_struktuuriyksused_id)
        REFERENCES dim_aine_struktuuriyksused (dimension_key),
    CONSTRAINT fk_dim_ained FOREIGN KEY (dim_ained_id)
        REFERENCES dim_ained (dimension_key),
    CONSTRAINT fk_dim_hindesyteemid FOREIGN KEY (dim_hindesyteemid_id)
        REFERENCES dim_hindesyteemid (dimension_key)
);

```

TABELITE LOOMINE – Eel-laadimisala *_STAGE

SQL 5. Dimensiooni DIM_HINDESYTEEMID eel-laadimisala

```

CREATE TABLE dim_hindesyteemid_stage (
    dw_hindesyteem_sid NUMBER NOT NULL,
    kas_arvestada NUMBER(1),
    tegur NUMBER(1),
    kas_positiivne NUMBER(1),
    tahis VARCHAR2(50),
    er_hinne_aeg DATE,
    binaarne_hinne NUMBER(1),
    er_hindesyteem_aeg DATE,
    CONSTRAINT dim_hindesyteemid_stage_pk PRIMARY KEY
        (dw_hindesyteem_sid)
);

```

SQL 6. Dimensiooni DIM_AINED eel-laadimisala

```
CREATE TABLE dim_ained_stage (  
  ois_aine_id NUMBER NOT NULL,  
  ainekood VARCHAR2(12),  
  ects_maht NUMBER,  
  eap NUMBER,  
  oa_register_aeg DATE,  
  nimetus VARCHAR2(255),  
  kehtiv_algus DATE,  
  kehtiv_lopp DATE,  
  oa_nimetus_aeg DATE,  
  CONSTRAINT dim_ained_stage_pk PRIMARY KEY (ois_aine_id)  
);
```

SQL 7. Fakti FAKT_OPPETULEMUSED eel-laadimisala

```
CREATE TABLE fakt_oppetulemused_stage (  
  dim_isikud_id NUMBER,  
  dim_struktuuriyksused_id NUMBER,  
  dim_oppekavad_id NUMBER,  
  dim_oppekohad_id NUMBER,  
  dim_aeg_id DATE,  
  dim_aine_struktuuriyksus_id NUMBER,  
  dim_ained_id NUMBER,  
  dim_hindesysteemid_id NUMBER,  
  aineprogramm_osa NUMBER,  
  tulemus NUMBER  
);
```

TABELITE LOOMINE – Metadata ja *sequence*-id

SQL 8. Meta-andmeid sisaldav Metadata tabel

```
CREATE TABLE metadata (  
  id NUMBER NOT NULL,  
  loaddate DATE,  
  andmeallikas_id NUMBER,  
  CONSTRAINT metadata_pk PRIMARY KEY (id)  
);
```

Oracle-i automaatsed jaded („*sequence*“):

SQL 9. Tabeli DIM_HINDESYSTEEMID primaavõti

```
CREATE SEQUENCE seq_dim_hindesys_primary_key
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;
```

SQL 10. Tabeli DIM_AINED primaarvõti

```
CREATE SEQUENCE seq_dim_ained_primary_key
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;
```

TABELITE TÄITMINE

SQL 11. DIM_HINDESYSTEEMID eel-laadimisala

```
INSERT INTO dim_hindesysteemid_stage (
  dw_hindesysteem_sid,
  kas_arvestada,
  tegur,
  kas_positiivne,
  tahis,
  er_hinne_aeg,
  binaarne_hinne,
  er_hindesysteem_aeg)
SELECT eh.id_hinne,
  eh.kas_arvestada,
  eh.tegur,
  eh.kas_positiivne,
  eh.tahis,
  eh.aeg,
  CASE ehs.kl_kontrollivorm WHEN 2 THEN 1 ELSE 0 END,
  ehs.aeg
FROM ER_HINNE eh, ER_HINDESYSTEEM ehs
WHERE eh.id_hindesysteem = ehs.id_hindesysteem;
```

SQL 12. DIM_AINED eel-laadimisala

```
INSERT INTO dim_ained_stage (  
    ois_aine_id,  
    ainekood,  
    ects_maht,  
    eap,  
    oa_register_aeg,  
    nimetus,  
    kehtiv_algus,  
    kehtiv_lopp,  
    oa_nimetus_aeg)  
SELECT o_r.id_register,  
    o_r.ainekood,  
    o_r.ects_maht,  
    o_r.eap,  
    o_r.aeg,  
    o_n.nimetus,  
    o_n.algus,  
    o_n.lopp,  
    o_n.aeg  
FROM OA_REGISTER o_r, OA_NIMETUS o_n  
WHERE o_r.id_register = o_n.id_register;
```

SQL 13. FAKT_OPPETULEMUSED eel-laadimisala

```
INSERT INTO fakt_oppetulemused_stage (  
    dim_isikud_id,  
    dim_struktuuriyksused_id,  
    dim_oppekavad_id,  
    dim_oppekohad_id,  
    dim_aeg_id,  
    dim_aine_struktuuriyksus_id,  
    dim_ained_id,  
    dim_hindesysteemid_id,  
    aineprogramm_osa,  
    tulemus)  
SELECT et.id_tudeng,  
    oo.id_struktuur,  
    oo.id_a_oppekava,  
    etoo.id_isiku_oppekoht,  
    et.tulemuse_kp,  
    oar.id_struktuur,
```

```

et.id_register,
et.id_hinne,
et.osa,
et.id_tulemus
FROM ER_TULEMUS et, ER_TULEMUS_OH_OPPEKOHT etoo, OA_REGISTER oar,
OH_OLEK oo, metadata md
WHERE et.id_tulemus = etoo.id_tulemus
      AND et.id_register = oar.id_register
      AND (et.tulemuse_kp BETWEEN oo.algus_kp AND oo.lopp_kp)
      AND et.id_tudeng = oo.id_isik
      AND (oo.kl_olek < 100 AND oo.kl_olek > 7)
      AND md.andmeallikas_id = 2
      AND et.aeg > md.loaddate;

```

SQL 14. Dimensioon DIM_HINDESYSTEEMID

```

INSERT INTO dim_hindesyteemid (
  dimension_key,
  dw_hindesyteem_sid,
  kas_arvestada,
  tegur,
  tahis,
  kas_positiivne,
  binaarne_hinne,
  dw_andmeallikas_id)
SELECT seq_dim_hindesys_primary_key.nextval,
       seq_dim_hindesys_fact_key.nextval,
       dhs.dw_hindesyteem_sid,
       dhs.kas_arvestada,
       dhs.tegur,
       dhs.tahis,
       dhs.kas_positiivne,
       dhs.binaarne_hinne,
       2
FROM dim_hindesyteemid_stage dhs, metadata md
WHERE (dhs.er_hinne_aeg < TRUNC(SYSDATE)
      AND md.andmeallikas_id = 2
      AND dhs.er_hinne_aeg > md.loaddate)
      OR (dhs.er_hindesyteem_aeg < TRUNC(SYSDATE)
      AND md.andmeallikas_id = 2
      AND dhs.er_hindesyteem_aeg > md.loaddate);

```

SQL 15. Dimensioon DIM_AINED

```
INSERT INTO dim_ained (  
    dimension_key,  
    ois_aine_id,  
    ainekood,  
    nimetus,  
    kehtiv_algus,  
    kehtiv_lopp,  
    ects_maht,  
    eap,  
    dw_andmeallikas_id)  
SELECT seq_dim_ained_primary_key.nextval,  
    seq_dim_ained_fact_key.nextval,  
    das.ois_aine_id,  
    das.ainekood,  
    das.nimetus,  
    das.kehtiv_algus,  
    das.kehtiv_lopp,  
    das.ects_maht,  
    das.eap,  
    2  
FROM dim_ained_stage das, metadata md  
WHERE (das.oa_register_aeg < TRUNC(SYSDATE)  
    AND md.andmeallikas_id = 2  
    AND das.oa_register_aeg > md.loaddate)  
OR (das.oa_nimetus_aeg < TRUNC(SYSDATE)  
    AND md.andmeallikas_id = 2  
    AND das.oa_nimetus_aeg > md.loaddate);
```

SQL 16. Fakt FAKT_OPPETULEMUSED

```
INSERT INTO fakt_oppetulemused (  
    dim_isikud_id,  
    dim_struktuuriyksused_id,  
    dim_andmeallikad_id,  
    dim_oppekavad_id,  
    dim_oppekohad_id,  
    dim_aeg_id,  
    dim_aine_struktuuriyksus_id,  
    dim_ained_id,  
    dim_hindesysteemid_id,  
    aineprogramm_osa,
```

```

tulemus)
SELECT isikud.dimension_key,
       yksus.dimension_key,
       allikad.dimension_key,
       kavad.dimension_key,
       kohad.dimension_key,
       aeg.dimension_key,
       aineyksus.dimension_key,
       ained.dimension_key,
       systeemid.dimension_key,
       fos.aineprogramm_osa,
       fos.tulemus
FROM dim_aeg aeg,
     dim_isikud isikud,
     dim_oppekohad kohad,
     dim_struktuuriyksused yksus,
     dim_oppekavad kavad,
     dim_aine_struktuuriyksus aineyksus,
     dim_ained ained,
     dim_hindesysteemid systeemid,
     dim_andmeallikad allikad,
     fakt_oppetulemused_stage fos
WHERE fos.dim_isikud_id = isikud.ois_isik_id
      AND fos.dim_struktuuriyksused_id = yksus.dw_struktuur_id
      AND fos.dim_oppekavad_id = kavad.ois_a_oppekava_id
      AND fos.dim_oppekohad_id = kohad.ois_oppekoht_id
      AND fos.dim_aeg_id = aeg.dw_aeg_id
      AND fos.dim_aine_struktuuriyksus_id = aineyksus.dw_struktuur_id
      AND fos.dim_ained_id = ained.ois_aine_id
      AND fos.dim_hindesysteemid_id = systeemid.dw_hindesysteem_sid
      AND allikad.dw_andmeallikas_id = 2;

```

SQL 17. „Minu tulemused“ vaate andmed

```

SELECT hinne.tahis,
       hinne.tegur,
       aine.nimetus,
       aine.ois_aine_id,
       isik.ois_isik_id
FROM fakt_oppetulemused fakt,
     dim_hindesysteemid hinne,
     dim_ained aine,
     dim_isikud isik

```



```

WHERE hinne.dimension_key = fakt.dim_hindesysteemid_id
  AND aine.dimension_key = fakt.dim_ained_id
  AND fakt.dim_isikud_id = isik.dimension_key
  AND fakt.dim_oppekavad_id = (
    SELECT DISTINCT fakt.dim_oppekavad_id
    FROM fakt_oppetulemused fakt, dim_isikud isikud
    WHERE fakt.dim_isikud_id = isikud.dimension_key
    AND isikud.ois_isik_id = <isiku id päringuvormist>);

```

SQL 18. „Minu ained“ vaate andmed – õppeainete valik

```

SELECT aine.nimetus,
       aine.ois_aine_id,
       isik.ois_isik_id
FROM fakt_oppetulemused fakt,
     dim_ained aine,
     dim_isikud isik
WHERE aine.dimension_key = fakt.dim_ained_id
  AND fakt.dim_isikud_id = isik.dimension_key
  AND isik.ois_isik_id = <isiku id päringuvormist>;

```

SQL 19. „Minu ained“ vaate andmed – õppetulemuste otsing valitud õppeaine põhjal

```

SELECT hinne.tahis,
       hinne.tegur,
       isik.ois_isik_id,
       kava.oppeaasta,
       kava.ois_a_oppekava_id
FROM fakt_oppetulemused fakt,
     dim_hindesysteemid hinne,
     dim_ained aine,
     dim_isikud isik,
     dim_oppekavad kava
WHERE hinne.dimension_key = fakt.dim_hindesysteemid_id
  AND aine.dimension_key = fakt.dim_ained_id
  AND fakt.dim_isikud_id = isik.dimension_key
  AND fakt.dim_oppekavad_id = kava.dimension_key
  AND aine.ois_aine_id = <õppeaine id päringuvormist>
ORDER BY kava.oppeaasta DESC, hinne.tegur DESC;

```

SQL 20. Metaandmete tabeli kirjete ajatempli uuendamine

```
UPDATE metadata  
SET loaddate = SYSDATE  
WHERE andmeallikas_id = <andmeallika id>;
```

SQL 21. Eel-laadimisala tabelite tühjendamine

```
TRUNCATE TABLE dim_hindesysteemid_stage;
```

```
TRUNCATE TABLE dim_ained_stage;
```

```
TRUNCATE TABLE fakt_oppetulemused_stage;
```

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Irma Sauga** (sünnikuupäev: 19.06.1987)
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Andmeait Tartu Ülikooli Aruandlussüsteemi näitel,
(*lõputöö pealkiri*)

mille juhendaja on Vambola Leping,
(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **26.05.2014**