

## Kordamisküsimusi eksamiks

Tutvuda tuleb loomulikult kogu kursuse käigus loengutes vaadeldud materjaliga, aga järgnevalt olulised punktid mida üle vaadata. Teemadest 8 ja 15 pole ühtegi küsimust. Ülejäänutest on – mõnest vähem/mõnest rohkem. Teadmiste saamise eesmärgil soovitan siiski lugeda ka teemade 8 ja 15 materjale.

- Erinevat tüüpi andmemudelite (hierarhiline, relatsiooniline, objekt-orienteeritud) ja vastavate andmebaasisüsteemide väljatöötamise kronoloogiline järjekord (kõigepealt hierarhilisel mudelil põhinevad andmebaasisüsteemid, seejärel relatsioonilisel mudelil põhinevad ja kõige viimaks objekt-orienteeritud andmebaasisüsteemid).
- Andmebaaside valdkonnas tuntud inimesed ja millega nad on end ajalukku jäädvustanud – E. F. Codd (relatsioonilise mudeli "isa"), P. Chen (olemi-suhte diagrammi väljamõtleja), C. J. Date ja H. Darwen (Kolmanda Manifesti autorid).
- Kuidas nimetatakse aastal 1995 avaldatud dokumenti, milles esitatakse C.J. Date ja H. Darwen poolt relatsioonilise mudeli täiendatud ja parandatud kirjelduse? (The Third Manifesto – Kolmas Manifest)
- Milline on Kolmandas Manifestis kirjeldatud andmebaasikeele nimi? (D)
- Mida tähendavad akronüümid ERD, UML, CASE ja SQL? (Entity-Relationship Diagram, Unified Modeling Language, Computer Aided/Assisted Software/System Engineering ja Structured Query Language).
- Võtmed relatsioonilises mudelis: primaarvõti, kandidaatvõti, alternatiivvõti, lihtvõti, liitvõti, supervõti, intelligentne võti, sisulise tähendusega võti, kattuv-võti, välisvõti.
- Relvari (relatsioonilise muutuja) supervõtmete arvu leidmine.
- Relatsioonilise mudeli põhimõisted: relatsiooniline muutuja (relvar), relatsioon, tüüp e domeen.
- Milline on ainus skalaarne tüüp, mida iga relatsiooniline andmebaasisüsteem igal juhul peab toetama? (BOOLEAN)
- Mida tähendab, et andmebaasisüsteemis on tüüp/operatoor süsteemi-definieeritud? (see on loodud andmebaasisüsteemi loojate poolt)
- Relatsiooni aste ja võimsus. (aste on atribuutide arv ja võimsus on korteežide arv)
- Millise andmemudeli alusel loodud andmebaas on ja milline ei ole navigatsiooniline andmebaas? (nt relatsiooniline ei ole)
- Olemi terviklikkuse reegel ja viidete terviklikkuse reegel.
- Kuidas moodustuvad relatsioonilises mudelis andmete vahelised seosed?
- Suletud maailma eeldus. (Iga antud ajahetkel relatsioonis asuv korteež, esitab sellel ajahetkel tõest väidet. Kui antud ajahetkel võiks relatsioonis sisalduda mingi korteež, kuid see korteež seal tegelikult ei sisaldu, siis järelikult see korteež esitab vale väite.)
- Informatsiooni ühtse esitamise printsiip. (kogu relatsioonilises andmebaasis hoitav informatsioon esitatakse vaid ühel viisil – relatsiooni atribuutide väärtustena)

- Relatsioonialgebra põhimõisted.
- Relatsioonialgebra operatsioonid: projektsioon, piirang, lõige, vahe, hulgateoreetiline summa, ühendamine, otsekorrutis, jagamine.
- Relatsioonialgebra operatsioonide kommutatiivsuse ja assotsiatiivsuse omadus. (Vahe leidmise operatsioon ei ole kommutatiivne ja assotsiatiivne)
- Unaarsed ja binaarsed relatsioonialgebra operatsioonid. (projektsioon ja piirang on unaarsed)
- Identiteedi projektsioon – projektsioon, mille tulemuses on kõik algse relatsiooni atribuudid.
- Täiendavad relatsioonialgebra operatsioonid – poolühendamine, poolvahe leidmine, vasakpoolne välisühendamine, laiendamine, kokkuvõtmine, koostamine, grupeerimine, mässimine.
- Relatsioonialgebra operatsioonide realiseerimine SQL keele abil. Peab oskama määrata, millist relatsioonialgebra operatsiooni realiseerib üks või teine SQL keeles kirjutatud päring.
- SQL keel ja tema süntaks: Milline SQL lause on õige või väär? Millise SQL lause abil pakutud variantidest, saab kirjeldatud ülesannet lahendada? (Enda teadmiste testimiseks sobib [http://www.w3schools.com/sql/sql\\_quiz.asp](http://www.w3schools.com/sql/sql_quiz.asp))
- SQL lause põhjal tuleb osata aru saada, millist ülesannet see lahendab.
- Milliseid funktsioone tuleb SQL standardi alusel kasutada hetke kuupäeva, kellaaja ning kuupäeva + kellaaja leidmiseks (CURRENT\_DATE, CURRENT\_TIME, CURRENT\_TIMESTAMP).
- Mis aastal avaldati esimene SQL standardi versioon? (1986)
- Milline on hetkel kehtiv SQL standardi versioon? (SQL:2011)
- Stringide konkatenatsioon. Milline on standardses SQLis selle operaatori esitamiseks kasutatav sümbol? (||)
- Väärtuste mittevõrdsus. Milline on standardses SQLis selle operaatori esitamiseks kasutatav sümbol? (<>)
- LIKE predikaat ja selles SQL standardi järgi kasutatavad mustri sümbolid. (% ja \_)
- Alampäringud: Üks kord täidetav alampäring, korreleeruv alampäring, skalaarne alampäring.
- Mitmevalentset loogikat kasutab relatsiooniline mudel? (kahevalentset)
- Mitmevalentset loogikat kasutab SQL? (kolmevalentset)
- Andmebaasiobjektide nimetamine ISO SQL standardi järgi. (nimi on maksimaalselt 128 märki pikk). Nimi ei tohi alata numbriga, sisaldada tühikut ja olla reserveeritud sõna, kui tegu pole just piiritletud identifikaatoriga.
- Mis asi on "piiritletud identifikaator"? (identifikaator on jutumärkides).
- Millised on SQL standardis nimetatud andmetüübid? (nt CURRENCY, MONEY, AUTONUMBER, SERIAL ei kuulu SQL standardisse)
- Kuupäeva ja kellaaja formaat SQL standardi järgi. ('YYYY-MM-DD HH24:MI:SS')
- Millist tüüpi objektide loomise võimaluse näeb ette SQL standard (nende loomiseks on CREATE lause)? (SCHEMA, TABLE, VIEW, DOMAIN, TYPE, ASSERTION, ROLE, TRIGGER, PROCEDURE, SEQUENCE, ...)
- Millist tüüpi objekte SQL standard ei kirjelda (nende loomiseks ei ole CREATE lauset)? (DATABASE, INDEX, USER, TABLESPACE, ...)

- Millisest SQL standardi versioonist alates on ühe või teise andmebaasiobjekti loomise võimalus kirjeldatud SQL standardis? (Protseduur, funktsioon – Alates SQL:1992 täiendusest (aastast 1996) PSM96; Trigger – Alates SQL:1999; Kasutaja-definieeritud tüüp – Alates SQL:1999; Roll – Alates SQL:1999; Arvutada generaator – Alates SQL:2003)
- Tabelite loomisel kirjeldatavad erinevad kitsendused e piirangud. Millised need on, kuidas neid kirjeldada ja kuidas nad mõjutavad tabelisse andmete lisamist ja andmete muutmist.
- Veerutaseme kitsendused vs. tabelitaseme kitsendused. (Tabelitaseme kitsendust tuleb kasutada, kui kitsendus hõlmab rohkem kui ühte veergu).
- Kompenseerivad tegevused, mida saab määrata välisvõtme kitsenduses ja mis määravad andmebaasisüsteemi käitumise viidete terviklikkuse vea korral. Kuidas nad toimivad, millises olukorras nende kontroll käivitatakse ja millises mitte?
  - ON UPDATE/DELETE NO ACTION (**NB!! See ei tähenda, et viidete terviklikkust ei kontrollita. See tähendab hoopis, et andmebaasisüsteem ei teosta viidete terviklikkuse vea korral ette kompenseerivat tegevust ja vea tekitanud andmemuudatus lükatakse tagasi!**)
  - ON UPDATE/DELETE RESTRICT
  - ON UPDATE/DELETE CASCADE
  - ON UPDATE/DELETE SET NULL
  - ON UPDATE/DELETE SET DEFAULT

Need määrangud määravad andmebaasisüsteemi käitumise viidete terviklikkuse vea korral. Kui viidete terviklikkuse viga ei teki, pole vaja ka kompenseerivaid tegevusi läbi viia ja andmemuudatus täidetakse ilma andmebaasisüsteemi poolse vahelesegamiseta.

- Deklaratiivne vs. protseduurne terviklikkuse reeglite tagamine (Deklaratiivsed kitsendused esitatakse CREATE TABLE või ALTER TABLE lauses, kuid protseduurne terviklikkuse reeglite tagamine tähendab triggerite ja andmebaasiserveris talletatud rutiinide loomist).
- Mida tähendab RESTRICT või CASCADE määrang andmebaasiobjekti kustutamise lauses? (RESTRICT – objekti ei kustutata, kui on sellest sõltuvaid objekte; CASCADE – objekt kustutatakse koos sõltuvate objektidega)
- Vaated, nende omadused ja kasutusvõimalused. Läbi milliste vaadete saab baastabelites andmeid muuta?
- Millise lausega luuakse SQLis virtuaalseid tabeleid? (CREATE VIEW)
- Millised on rahvusvahelise klassifikaatori "ISO 5218 – Information technology -- Codes for the representation of human sexes" kirjeldatavad võimalikud väärtused? (0, 1, 2, 9)
- Mida tähendab WITH CHECK OPTION määrangu kasutamine vaadete kirjeldamise juures? (andmemuudatused läbi vaadete peavad vastama vaate alampäringu tingimustele).
- Milliseid SQL lauseid kasutatakse õiguste/rollide jagamiseks ja äravõtmiseks kasutajatelt/rollidelt? (GRANT/REVOKE)

- Mida tähendavad WITH GRANT OPTION ja WITH ADMIN OPTION määrangud GRANT lauses? (õiguse/rolli saaja saab seda teistele kasutajatele või rollidele edasi anda)
- Infosüsteemi arendamise etappide järjekord klassikalises (kaskaadses) infosüsteemide arendamise protsessis. Milline etapp eelneb ja järgneb *vahetult* millisele teisele etapile. (strateegiline analüüs, detailanalüüs, disain, ehitamine, rakendamine, hooldamine)
- Mille poolest erineb SQL keel ja selle aluseks olev andmemudel relatsioonilisest andmemudelist?
- Milline on relatsioonilise mudeli ja SQLi vaheline seos? (SQL andmebaasikeele väljatöötamisel on lähtutud relatsioonilisest mudelist, kuid tulemuseks saadud andmebaasikeel ei järgi täies mahus relatsioonilise mudeli nõudmisi)
- Strateegiline analüüs: eesmärgid, tegevused, dokumendid, mis selle käigus luuakse (vt näidisprojekt), tulemused.
- Detailanalüüs: eesmärgid, tegevused, dokumendid, mis selle käigus luuakse (vt näidisprojekt), tulemused.
  - Peab teadma, millised dokumendid koostatakse mingi allsüsteemi kirjeldamisel (pädevusala, funktsionaalne allsüsteem, register), mingis infosüsteemi arendamise etapis. Tuletage meelde oma näiteprojekti sisukord.
  - Peab teadma iseseisvas töös koostatud dokumentide kohta, mis informatsiooni nad edasi annavad ja milleks neid kasutatakse (kasutusjuhtude mudel, kontseptuaalne andmemudel, seisundidiagramm, tegevusdiagramm, andmebaasioperatsioonide lepingud, CRUD maatriks) ning milline mudel on aluseks millise teise mudeli loomisele (näiteks põhiobjektide järgi leian funktsionaalsed allsüsteemid ja registrid; seisundidiagrammi järgi leian paljud kasutusjuhud ja paljude andmebaasioperatsioonide nimed).
  - Peab ka tunda UML klassi-, kasutusjuhtude-, seisundi- ja tegevusdiagrammidel kasutatavate põhiliste sümbolite tähendust.
- Kuidas saab UML keelt kasutada andmebaasi projekteerimise juures?
- Millist UMLi diagrammi tüüpi saab kasutada olemi-suhte diagrammi ja andmebaasi diagrammi koostamiseks? (klassidiagrammi)
- Olemi-suhte diagramm ja temaga seotud mõisted.
- Peab oskama leida lausendeid, mis kirjeldavad kõige täpsemalt etteantud olemi-suhte diagrammi fragmenti (tuleb tunda võimsustikke ning osata neid sõnaliselt interpreteerida).
- Loogiline disain: eesmärgid, tegevused, dokumendid, mis selle käigus luuakse (vt näidisprojekt), tulemused.
- Millised on hea relatsiooni võtme omadused? (tuttavlikkus, lihtsus, stabiilsus, ei sisalda kodeeritud informatsiooni)
- Mis on reaalsed kasutusjuhud, millal ja milleks neid luuakse? (luuakse DISAINI käigus)
- Peab aru saama *Database Design Language* kirjeldustest:

Tabel(veerg1 NOT NULL, veerg2 NULL, veerg3 NULL) – veergude nimekiri (NOT NULL – veerg on kohustuslik; NULL – veerg on mittekohustuslik) <b>Primaarvõti</b> (veerg1) – primaarvõti e esmasvõti
--

**Alternatiivvõti** (veerg2)

**Välisvõti** (veerg3) **Viitab** Tabel2(veerg1)

– alternatiivvõti

– välisvõti

- Normaliseerimine.
  - Atribuutide hulkade vaheliste sõltuvuste tüübid (funktsionaalne sõltuvus, multiväärtuslik sõltuvus, ühendamisõltuvus).
  - Funktsionaalse sõltuvuse determinant ( $A \Rightarrow B - A$  on determinant).
  - Täielik funktsionaalne sõltuvus.
  - Tegevused igale normaalkujule üleminekul.
  - Kõigi normaalkujude definitsioonid.
  - Nõudmised, millele peab relvar vastama peale teatud normaalkujule viimist.
  - Milline on kõige madalam normaalkuju, mille omamise korral võib relvari kohta öelda, et see on "normaliseeritud"? (esimene normaalkuju)
  - Mida tähendab, et relatsiooniline andmebaas on täielikult normaliseeritud? (kõik selles olevad relvarid on vähemalt viiendal normaalkujul)
  - Võivad olla antud definitsioonid, mille põhjal tuleb öelda, millise normaalkuju kohta need käivad.
  - Kas normaalkujul  $N$  olev relatsioon on alati ka normaalkujul  $N+1$ ? (ei)
  - Kas normaalkujul  $N$  olev relatsioon on alati ka normaalkujul  $N-1$ ? (jah)
  - Heathi ja Fagini teoreemid.
  - Lihtsad reeglid normaalkuju määramiseks.
    1. Kui relvar on Boyce/Coddi normaalkujul ja mõni tema kandidaatvõti on lihtne, siis on see relvar ka neljandal normaalkujul (aga ei pruugi olla viiendal normaalkujul).
    2. Kui relvar on kolmandal normaalkujul ja iga tema kandidaatvõti on lihtne, siis on see relvar ka viiendal normaalkujul.
- Kuidas aitab andmebaasi disaini parandada ortogonaalse andmebaasi disaini printsiibi rakendamine? (vähendab andmete liiasust üle erinevate tabelite/relatsiooniliste muutujate)
- Kas normaliseerimise ja ortogonaalse printsiibi rakendamine aitab vabaneda igasugusest andmete liiasusest? (ei)
- Kas hea andmebaasi disaini põhimõtete kohaselt peab igasugune andmete liiasus olema kontrollitud või kontrollimata? (kontrollitud)
- Kontseptuaalse andmemudeli teisendusreeglid loogilise disaini andmemudeliks (1:1, 1:M ja M:N seosetüübid, rekursiivne seosetüüp, üldistussuhted, kaar). Küsimuses kirjeldatud olukorras peab oskama valida mitme soovitusse seast õige.
- Mis vahe on loogilisel ja füüsilisel andmete sõltumatusel (mõisted andmebaasisüsteemi arhitektuurist)?
- Millised on töölaua- ja serveri andmebaasisüsteemide sarnasused ja erinevused?
- Mida saab ja ei saa teha andmebaasisüsteemis MS Access (2013), kus on kasutusel traditsiooniline Jet andmebaasimootor? Kas seal saab või ei saa luua tabeleid, vaateid, indekseid, trigereid, salvestatud protseduure? Kas seal saab või ei saa luua ekraanivorme, trükiseid, makroid? Näiteks ei saa MS Accessis (2013) (kui kasutatakse andmebaasimootorit "Jet") kasutada CREATE TRIGGER lauset SQL trigerite loomiseks. Samas on alates MS Access (2013) võimalik siduda tabelitega andmete makrosid, mis võimaldavad lahendada samu probleeme nagu SQL trigerid. Samuti ei saa seal luua eraldi arvujada generaatori objekti (*sequence generator*) nagu nt.

Oracles või PostgreSQLis. Tuleb kasutada *autonumber* andmetüüpi. MS Accessis pole võimalik luua uusi *tüüpe* ja *domeene* (süsteem ei toeta CREATE TYPE ja CREATE DOMAIN lauseid). MS Accessis saab luua baastabeleid, vaateid, indekseid. MS Accessis saab ka luua CREATE PROCEDURE lausega lihtsaid protseduure, mis sisaldavad ühte SQL lauset.

- Kas *andmebaas* ja *andmebaasisüsteem* on sünonüümid? (ei)
- Andmebaasisüsteemi süsteemikataloog – kuidas seda kasutada ja kuidas seal andmed uuenevad? (andmebaasisüsteem uuendab seal andmeid automaatselt, kohe peale andmekirjelduskeele lause käivitamist)
- SQL lause töötlemine enne täitmist – millistest sammudest see koosneb, millises järjekorras neid samme läbitakse ning kes või mis neid samme läbi viib?
- Milline on üldine strateegia loogilise täitmisplaani optimeerimiseks? (andmete hulka piiravad operatsioonid (projektsioon, piirang) üritatakse teha enne ühendamise (*joini*) operatsiooni)
- Millist nime kannab SQL standardi järgi skeem, mis sisaldab süsteemikataloogi põhjal tehtud vaateid? (INFORMATION\_SCHEMA)
- Millised on tuntud andmebaasisüsteemid (kaubamärgid)?
- Andmebaasisüsteemide ühisnimetajad (nt millised on avaliku/salajase lähtekoodiga, töölaua/ülemineku/serveri süsteemid).
- Füüsiline disain: eesmärgid, tegevused, dokumendid, mis selle käigus luuakse (vt näidisprojekt), tulemused.
- Korteži loogiline aadress (relvari nimi, kandidaatvõtme hõlmatud atribuutide nimed, kandidaatvõtme väärtus).
- Indekseerimine. Erinevad indeksite tüübid: B-puu, Bitmap, ... Millises olukorras on neid võimalik ja õige kasutada ja millises mitte? (B-puu indeksi loomist võib kaaluda veergude korral, kus on suhteliselt palju unikaalseid väärtuseid ning mida kasutatakse sageli päringute tingimustes; Bitmap indeksit on kõige parem kasutada suure andmemahuga (kuna indeksi andmemahut kujuneb väikeseks), harva muudetavates tabelites. Bitmap indeksit tuleks kasutada veergudel, kus on vähe erinevaid väärtuseid.)
- Kas andmebaasisüsteem peaks alati eelistama päringule vastamiseks indeksi kasutamist? (ei, andmebaasisüsteem peaks eelistama tabeli läbiskaneerimist kui: (a) tabel on väga väike – indeksi lugemine ei vähenda oluliselt loetavate plokkide arvu; (b) tabel on suur. Päring tagastab suure hulga tabeli ridadest – indeksi lugemise tõttu loetakse ühte plokki korduvalt.)
- Mida tähendab fraas "B-puu", kui räägime B-puu indeksitest? (tasakaalustatud puu)
- Millistele veergudele loob enamik andmebaasisüsteeme indeksi automaatselt? (primaarvõtme või unikaalsuse kitsenduse poolt hõlmatud)
- Kas indeksi loomine suurendab andmebaasi salvestamiseks vajaliku kettaruumi hulka? (jah)
- Kuidas kiirendada suure hulga ridade lisamist tabelisse? (kustutada tabeliga seotud indeksid, lisada read, luua indeksid uuesti)
- Mida tähendab ridade migreerumine andmebaasi sisemisel tasemel ja kas see on töökiiruse seisukohast hea või halb? (halb)
- Kuidas mõjutab andmete muutmise sagedus seda, kui tihedalt tuleb andmeplokkidesse sisemisel tasemel ridu paigutada?

- Millisesse andmebaasi tasemesse (skeemi) kuuluvad erinevad andmebaasiobjektid? (vaated välisesse, baastabelid kontseptuaalsesse, indeksid sisemisse)
- Kuidas (millise käsuga) saab Oracles luua arvujada generaatorit (CREATE SEQUENCE) ja trigerit (CREATE TRIGGER)?
- Aktiivne, sündmustele reageeriv andmebaas – millise andmebaasiobjekti olemasolul võib sellest rääkida? (triger, deklaratiivsed kitsendused)
- Trigerid (Oracle näitel). Konkreetse näite põhjal peab oskama määrata, kas ja kui mitu korda käivitab mingi SQL lause kirjeldatud trigeri kehendis esitatud tegevusi. Peab aru saama, kuidas eristada tabeli- ja reataseme trigereid ning sellest, kuidas määrata tingimusi (WHEN klausel), mille täidetuse korral triger käivitub. Samuti peab teadma, kuidas kirjutada trigerit mis käivitub siis, kui muudetakse andmeid mingis kindlas tabeli veerus.
- Millist tüüpi trigereid saab/ei saa Oracles kasutada? Milliste sündmuste tulemusel triger käivitatakse ja millist tüüpi sündmustega ei saa trigerit seostada? (Pole olemas SELECT trigereid ega DURING trigereid)
- Oracles trigeritega seotud "mutating table" probleem. (reataseme trigeri protseduuris ei saa SQL lauseid kasutades lugeda/muuta andmeid tabelis, mille muutmine trigeri käivitas)
- Millise fraasiga algab Oracles salvestatud protseduuri/trigeri/arvujada generaatori loomise lause? (CREATE PROCEDURE / CREATE TRIGGER / CREATE SEQUENCE)
- Kuidas nimetatakse andmebaasisüsteemis Oracle kasutatavat protseduuride keelt? (PL/SQL)
- Millised on hea disaini printsiibid pakettide (nagu neid saab näiteks luua andmebaasisüsteemis Oracle) loomiseks ja neile vastutuste jagamiseks? Tegemist on üldiste hea tarkvara disaini printsiipidega (madal sõltuvus – *low coupling*, kõrge kokkukuuluvus – *high cohesion*)
- Milline on kõige olulisem dokument, mille alusel leitakse andmebaasiserveris talletatud rutiinid, mis tuleks andmebaasis luua? (andmebaasioperatsioonide lepingud)
- Millised on transaktsiooni neli põhilist omadust? (ACID – atomaarsus, terviklikkus, isoleeritus, kestvus/jätkuvus)
- Milliste käskudega toimub SQLis transaktsioonide kinnitamine ja tühistamine? (COMMIT ja ROLLBACK)
- Millistele tingimustele vastavad tabelid on kõige paremad denormaliseerimise kandidaadid? (andmeid küsitakse sageli, muudatusi tehakse harva)
- Milliseid tegevusi tehakse andmetabelite denormaliseerimise käigus ja mis on selle tulemus? Milleks denormaliseerimist ette võetakse? (mõningate päringute kiiremaks muutmiseks)
- Mitmendale normaalkujule tuleks tabelid viia enne, kui kaaluda mõne tabeli denormaliseerimist? (viiendale normaalkujule)
- Millised on denormaliseerimise ohud? (põhjustab andmete muutmise anomaaliate tekke; võib suurendada andmete muutmiseks kuluvat aega; suurendab andmemahutu; võib põhjustada vastuoluliste andmete sattumist andmebaasi; andmebaasi kontseptuaalne skeem muutub kasutaja jaoks ebaselgemaks)