

Harjoitustyön tehtävänanto

Yleistä harjoitustyöstä

Harjoitustyössä suunnitellaan ja toteutetaan relaatiotietokanta kuvitteelliselle loppukäyttäjälle. Voit valita aiheen itse, kuten esimerkiksi arkkupakastin-inventaario, myyntireskontra, urheilujoukkue-tietokanta tai tuoteinventaarior. Tietokannalle toteutetaan esimerkki-luonteinen Python-ohjelma, joka demonstroi viittä tietokantahakua, yhtä päivitystä, yhtä syötettä ja yhtä visualisointia. Työ tehdään yksilötyönä. Jos epäselvyyksiä esiintyy, ota yhteyttä luennoitsijaan tai harjoituksissa kurssiassistenttiin.

Työn osa-alueet ovat

1. Kirjallinen, johon kuuluu
 1. Suunnittelu (mallinnettavan aiheen kuvaus ja työn rajaukset; tietokannan ER-käsittemalli ja eheyssäännöt)
 2. relaatiomalliksi konvertointi eli muuntaminen ja
 3. ohjelmallisen toteutuksen kuvailu.
2. Ohjelmallinen toteutus SQLite3:lla ja Pythonilla

Osa-alueet kuvataan seuraavaksi.

Kirjallinen osuus

Käytä annettua raporttipohjaa. Raporttipohjassa on suositeltu rakenne ja esimerkkejä sisällölle. Pohjan pitäisi toimia sekä Wordillä että OpenOfficella.

Kirjallisen työn rakenne ja toteutusvaiheet

1. Johdanto. Kuvaile tässä loppukäyttäjä, tietokannan käyttöympäristö ja käyttötarve.
2. Käsittemalli, käsiteanalyysi ja relaatioksi transformointi. Suunnittele käsitteet ja mallinna ne ER-kaaviolla. Sen jälkeen muunna käsittemalli (ER-mallina ilmaistuna) relaatiomuotoon (relaatiokaavoina tai UML-taulukkona ilmaistuna).
3. Tietokantatoteutus. Toteuta relaatiotietokanta SQLite3-tietokannalla ja sen jälkeen toteuta vaaditut ominaisuudet Pythonilla. Suunnittele myös mihin kohtiin olisi hyvä soveltaa indeksejä nopeuttamaan hakuja.
4. Keskustelu. Reflektio ja johtopäätökset. Tämä voi olla melko suppea normaalin työn kohdalla.

Suunnitteluosuus, vaatimukset

- Käsittemallin tulee koostua vähintään kuudesta käsitteestä (yksilötyyppi) ja täten lopullisen tietokannan tulee koostua ainakin yhtä monesta relaatiosta eli taulusta. Relaatiomalli voi myös olla laajempi (esim. M-N -transformaation takia), mutta ei suppeampi.
- Käsittemallin tulee sisältää vähintään yksi monesta - moneen suhde (M:N).
- Suunnitteluosuuden tulee sisältää sekä käsittemallin että siitä johdettu taulukuvauksen.
- Määrittele minimi- ja maksimikardinaliteetit sekä eheysrajoitteet.

- Tietokantaan tulee toteuttaa ainakin viisi kyselyä (queries) tai näkymää (views), jotka tuottavat jonkin/joidenkin loppukäyttäjien kannalta hyödyllistä tietoa. Kahta JOIN-lausetta on käytettävä ainakin yhdessä kyselyssä eli M:N-suhdetta käyttävä haku.
- Suunnittele indeksien käyttö ja perustele valinnat.
- Tietokannassa ei saa esiintyä tietorakenteita, jotka ovat relaatiomallin vastaisia. Kts. lisätietoa harjoituksissa, jotka liittyvät ER- ja relaatiomalli-suunnitteluun.

Ohjelmallinen osuus

Ohjelmallisessa toteutuksessa ensin tehdään tiedostoon tallennettu SQLite3-tietokanta (ei sarja SQL-komentoja, vaan binäärinen SQLite3:lla tehty tietokanta) ja interaktiivinen Python-sovellus, joka demonstroi yhden osa-alueen tietokannan toiminnasta. Esimerkiksi urheilutieto-tietokannassa sovellus voi tulostaa vanhojen otteluiden tiedot ja sitten kysyä että päivitetäänkö vanhaa ottelutulosta tai syötetäänkö uusi ottelu. Koko tietokantaa ei tarvitse pystyä käyttämään sovelluksen kautta. Riittää että toteuttaa lineaarisen ohjelman joka demonstroi viisi hakua, yhden päivityksen, yhden syötteen ja yhden bokeh-kirjastolla toteutetun visualisaation. Lisätietoa bokehista saat esimerkiksi Python-[viikkotehtävästä](#). Bokeh-visualisaatio voi olla esimerkiksi pylväsdiagrammi.

Ohjelmallinen toteutus, vaatimukset

- Tietokanta on testattavissa ja käytettävissä sekä SQLite3 että Python sovelluskäyttöliittymän kautta.
- Ohjelma demonstroi miten tietokannasta voidaan lukea tietoja, päivittää olemassa olevia tietoja sekä syöttää uutta dataa. Lisäksi yksi visualisointidemonstrointi.
 - Eli: Ohjelma sisältää vähintään viisi hakua, yhden syötteen, yhden päivityksen ja yhden bokeh-visualisoinnin.
- Tietokanta toteuttaa eheysrajoitteet SQL:n CASCADE ja vastaavilla komennoilla.
- Tietokanta sisältää järkevästi suunniteltuja indeksejä.

Ohjelmallinen osuus, vaadittava sisältö

- Python-ohjelma
- Python-ohjelman käyttämä SQLite3-tietokanta
- Erillinen tekstitiedosto, joka sisältää taulun luomiseen vaadittavat SQL-komennot

Katso palautettavan ohjelman ja muiden osuuksien hakemistorakenne alta.

Palautusohje

Harjoitustyön deadline on 2.3.2019

Kaikissa tapauksissa harjoitustyö palautetaan Moodleen. Linkki on Moodle-kurssin etusivulla.

Palautukseen kuuluu

- Raportti PDF-muodossa. (Työvaihe 1)

- Tietokannan luomiseksi tarvittavat SQL-käskyt, jotka voidaan suorittaa SQLite3:lla .read -komentoa käyttäen sekä kyselyjen SELECT-lauseet. (Työvaihe 2)
- Ohjelmatoteutus, johon kuuluu sekä SQLite3-muotoinen tietokanta että sovelluskäyttöliittymä, joka hyödyntää tietokantaa. (Työvaihe 3)

Palautustiedoston rakenne

1. Yhdessä paketissa (ZIP tai TAR) kaikki työhön kuuluva materiaali. Missään tapauksessa ei RAR-muotoisia tai muita kummallisia pakkausohjelmia.
2. Pakattu tiedosto sisältää YHDEN HAKEMISTON, joka on nimetty mallilla: "Sukunimi_Etunimi.zip". Hakemiston sisältö on seuraava:
 1. Raportti.pdf
 2. SQL-komennot.sql (tekstitiedosto, SQLite3-tietokannan luomiseksi tarvittavat komennot, pitää olla ladattavissa SQLite3:n .read -komennolla)
 3. Alihakemisto, jonka nimi on Ohjelmatoteutus. Tähän hakemistoon laitetaan kaikki ohjelmakoodi ja muut tiedostot, joita sovelluksen ajamiseksi tarvitaan. Laita myös valmis versio SQLite3-muotoisesta tietokannasta mukaan, jolla ohjelmaa voi heti testiajaa. Tässä versiossa voi olla testisyötettä valmiina tietokannassa.

Arvostelukriteereitä

Lähtökohtaisesti jos kaikki vaatimukset on toteutettu ohjeiden mukaisesti ja tyydyttävästi, niin työstä saa täydet pisteet. Erityisen tärkeitä on kirjallisen osuuden, suunnittelutyön ja ohjelmallisen toteutuksen vaatimusten täyttäminen. Arvostelussa painotetaan erityisesti seuraavia asioita:

- ER-mallintamisen ymmärtäminen ja demonstroiminen
- Reaalimaailman osa-alueen ("problem domain") järkevä mallintaminen ja mallinnettujen osien valinta
- Käsitteellisen mallin muuttaminen relaatiotietokannaksi
- Suhdetyyppien (1:1, 1:N, M:N) sekä kardinaliteettien ymmärtäminen ja ymmärtämisen demonstroiminen.
- SQL-tietokantasuunnittelu käsitteellisen mallin ja mallin transformaation pohjalta.
- Eheysrajoitteet, pää- ja vierasavaimet.
- M:N -suhteen toiminnan demonstroiminen käsitteellisellä ja ohjelmatasolla.
- Suunnittelutyön ja ohjelmallisen toteutuksen hyvä dokumentointi. Raportointipohjasta on hyvä lähteä etenemään.
- Virheetön Python-ohjelma, joka demonstroi tietokannan käyttöä.
- Vertaisarvioinnin suorittaminen tai suorittamatta jättäminen.

Puutteista vähennetään 1pts, 5pts tai 10pts puutteiden vakavuudesta riippuen.

Vertaisarviointi ei vaikuta arvioitavan työn arvosanaan ja arviointi on vain sanallista. Kurssin luennoitsija käy läpi työt itsenäisesti ja pisteyttää jokaisen työn suoraan. Hyvä vertaisarviointi tai arvioinnin tekemättä jättäminen vaikuttaa kuitenkin vertaisarvioijan arvosanaan.

Erityisiä aikaisempien vuosien palautuksissa esiintyneitä ongelmakohtia jotka kannattaa tarkastaa ennen palautusta

- Mallintamiseen liittyvät ongelmat ja väärin kaavioiden laittaminen väärin mallinnusvaiheisiin

- Kardinaliteettien väliin jättäminen mallintamisessa
- Eheyssääntöjen käyttämättä jättäminen SQLite-toteutuksessa (CASCADE, CHECK, ym.)
- M:N -suhteen puuttuminen (tai sitä demonstroivan haun puuttuminen)
- Python-ohjelman kaatuminen testatessa
- Taulunluontikomentojen puuttuminen erillisenä tiedostona tai ne eivät toimi .read-komennolla kokeiltaessa.

Last modified: Sunday, 17 February 2019, 11:49 PM