

TIETA7 HARJOITUSTYÖ

DIVARI DATABASE

ALBERT IHO

RYHMÄ 32

Ohjelman ominaisuudet

Ohjelma on jaettu kahdeksaan luokkaan, jotka toimivat vesiputoustyyliin. Ylempi luokka kutsuu alempaa, joka tarvittaessa kutsuu vielä alempaa. Ohjelman kahdeksasta luokasta yksi käynnistää ohjelman, kaksi käsittelevät siirtymiä, neljä komentoa käsittelevää luokkaa sekä apuluokka. Siirtymiä käsittelevät luokat ovat "Kayttoliittymä" ja "User." Kayttoliittymässä on ohjelmaan kirjautumisen käyttöliittymä ja kirjautumiseen liittyvät operaatiot (checkUserInfo, createNewUser) ovat Login-luokassa jota kutsutaan kayttoliittymästä.

Käyttäjän kirjaututtua sisään onnistuneesti Login-luokka välittää käyttäjän kirjautumisen ja käyttötason (admin vai normaalikäyttäjä) User-luokalle joka vastaanottaa käyttäjän komennot Divari Database-ohjelmassa. (jos harjoitustyön arviointivaiheessa sisäänkirjautumisen ja käyttäjän luomisen kanssa tulee ongelmia, voi sisäänkirjautumisen kiertää antamalla syötteiksi "login" > "override" jolloin ohjelma kirjaa käyttäjän sisään suoraan ylläpitäjänä. Tällöin ei tosin ostoskärryn luominen onnistu, ja sen myötä normaalin käyttäjän toiminnallisuuksia ei voi testata overriding avulla.) User luokka kutsuu Commands luokasta löytyviä komennot suorittavia operaatioita käyttäjän syötteen mukaan. Commands luokassa on toteutettu 16 komentoa joista 9 on normaalikäyttäjän käytettävissä ja 7 ovat ylläpitäjien tukena.

Komennot listattuna:

1. help
2. newCart
3. search
4. add
5. remove
6. cart
7. cashout
8. discard
9. exit
10. query
11. addItem
12. promote
13. demote
14. reset
15. triggerTest
16. r2

Harjoitustyön toiminnallisuuden kannalta komennot 0-9 ovat oleellisia, ylläpitäjien tueksi luodut komennot olivat pitkälti myös minua varten harjoitustyön laatijana, en osannut avata postgresql komentoikkunaa niin sovelsin tekemällä query-komennon joka käytännössä vastaa komentoikkunaa, SQL-lauseet ajetaan JDBC avulla. Komennot 11, 12 ja 13 olivat hetkellisenä apuna ohjelman toiminnallisuuden kokeilussa. 14, 15 ja 16 ovat harjoitustyön tarkistuksen tukena, 14. nollaa taulut "tila.sql" dokumentissa mainittuun muotoon, 15. todistaa harjoitustyössä toimivan tapahtumat **t6** ja **t7**. Komento 16 on puhtaasti harjoitustyön tehtävänannossa annettu **r2** raportti/kysely.

Commands luokkaa tukevat aliluokat Order ja Delivery. Komennon luonteesta riippuen kutsutaan joko Order-luokan (tilauksiin liittyvissä komennossa) komentoja jotka suorittavat tarkentavaa tiedonpalautusta tai mahdollistavat esimerkiksi useamman kirjan lisäämisen tilaukseen kutsumatta komentoa useasti. Delivery-luokan operaatiot käsittelevät tilauksen lähettämiseen liittyvää kirjojen lähetyksiin jakamista ja niiden hintaan sekä määrään liittyviä asioita.

Viimeisenä luokkana on Help-luokka jossa tärkeimpänä operaationa on yhteyden luominen tietokantaan. Help luokassa on myös syötteen merkkien laillisuuden tarkistava operaatio jota kutsutaan tilanteissa missä voisi olla mahdollista toteuttaa SQL-injektio. Niiden lisäksi muutama virheilmoitus on toteutettu help-luokan toimintona, joka nopeuttaa palautteen muodon muuttamista esimerkiksi SQL-virheilmoitusten suhteen. Help-luokan viimeisenä operaationa on reset, joka palauttaa harjoitustyön "tila.sql" tiedoston mukaiseen tilaan asiakas-taulua lukuunottamatta, asiakas-taulun tyhjennys tulee suorittaa käsin jos sen haluaa tehdä. Reset-operaation päätarkoitus oli auttaa ohjelman debuggaamisessa, sekä harjoitustyön arvioinnissa. Kaikki toiminnallisuus on helppo testata kun on joku tietty "vakioitu" lähtötaso josta tauluja lähdetään muuttamaan.

Ohjelman käyttö

Ohjelma on Javalla toteutettu komentorivipohjainen ohjelma. Ohjelman valmisteluun se pitäisi ladata tietokoneelle ja luoda yhteys paikkaan josta tietokannat löytyvät. Muuten ohjelma pitkälti ohjeistaa käyttäjää joka askeleella kuinka edetä, ja ohjelman pitäisi antaa palautetta käyttäjälle tapauksissa joissa käyttäjän syöte on virheellinen.

Muutokset aiemman suunnitelman suhteen

Suurimpina muutoksina on varmasti Tilaustiedot sekä Lahetys-taulujen lisääminen työhön. Tilaustiedot käsittelevät yksittäiseen tilaukseen liittyviä teoksia ja Lahetykset käsittelevät teosten ostajille toimittamiseen liittyviä tietoja. Kirjat- näkymään lisättiin teos_id-sarake sillä päädyin ohjelmassa käsittelemään teoksia teos_id'n avulla. Kirjat_lkm- näkymään laskettavaksi tiedoksi tuli yksittäiseen isbn:ään liittyvät teos_id't joidenka tarkoituksena on ilmaista sen kirjan myynnissä olevia teoksia.

Oma arvio

Työn aloittaminen oli selvästikin vaikein osa-alue, koin aloittamisen olevan suuri askel kurssin eteen tekemäni työn vuoksi. Pelkäsin harjoitustyön olevan teknisesti liian vaikea sillä SQL-kielen logiikka ja operaatiot ovat aikaisemmin olleet minulle haastavia. Työn aloitettuani huomasin olevan mahdollista toteuttaa kaikki tietokantayhteys hyvin yksinkertaisilla kyselyillä ja suorittaa tietojenkäsittely ja tulosten muokkaaminen javan puolella. Innostuin kovasti ja harjoitustyöhön alkoi ilmestyä operaatioita ja toiminnallisuuksia myös tehtävänannon ulkopuolelta, joka venytti harjoitustyön myöhästymisen pyydetyn lisääjän lisäksi kolmella päivällä.

Selviä puutteita työssä on scanner luokan virheet, ohjelma saattaa kaatua kun scanner odottaa saavansa INT-arvon, mutta syötetty arvo ei ole numeerinen. Ohjelma ei myöskään päivitä yksittäisen divarin teostietoja ostotapahtuman jälkeen. Ajattelin ratkaisevani asian lisäämällä teos-tauluun divari_id ominaisuuden jonka avulla teoksen omistavan divarin tiedot päivitetäisiin, mutta se olisi viivästänyt harjoitustyön palautusta edelleen. Harjoitustyöstä myös puuttuu raportit/kyselyt r3 sekä r4. Uskon että olisin saanut tehtyä molemmat mutta aloitin yksinkertaisesti harjoitustyön työstämisen myöhään ja pitkäksi venynyt kevätflunssa/räkätauti nakersi aikaansaavuutta vapun kanssa.

Selvennyksenä tein harjoitustyön yksin, sillä unohdin ilmoittaa tarvitsevani ryhmän kun ryhmä mihin minun piti mennä olikin saanut kolmannen jäsenen jota en ymmärtänyt ottaa huomioon. Harjoitustyö itsessään oli sopivan haastava tehdä yksin, koen että en pysynyt asiassa vaan lähdin

Albert Iho (iho.albert.k@student.uta.fi)
ai425377

rönsyilemään koodin suhteen ja tästä syystä harjoitustyö saattaa sisältää ~500 riviä koodia jotka eivät ole toiminnallisuuden kannalta elintärkeitä.