

Harjoitustyö

CS-A1150 - 9.4.2021

Eetu Mustonen - Nea Nieminen - Julia Sippala

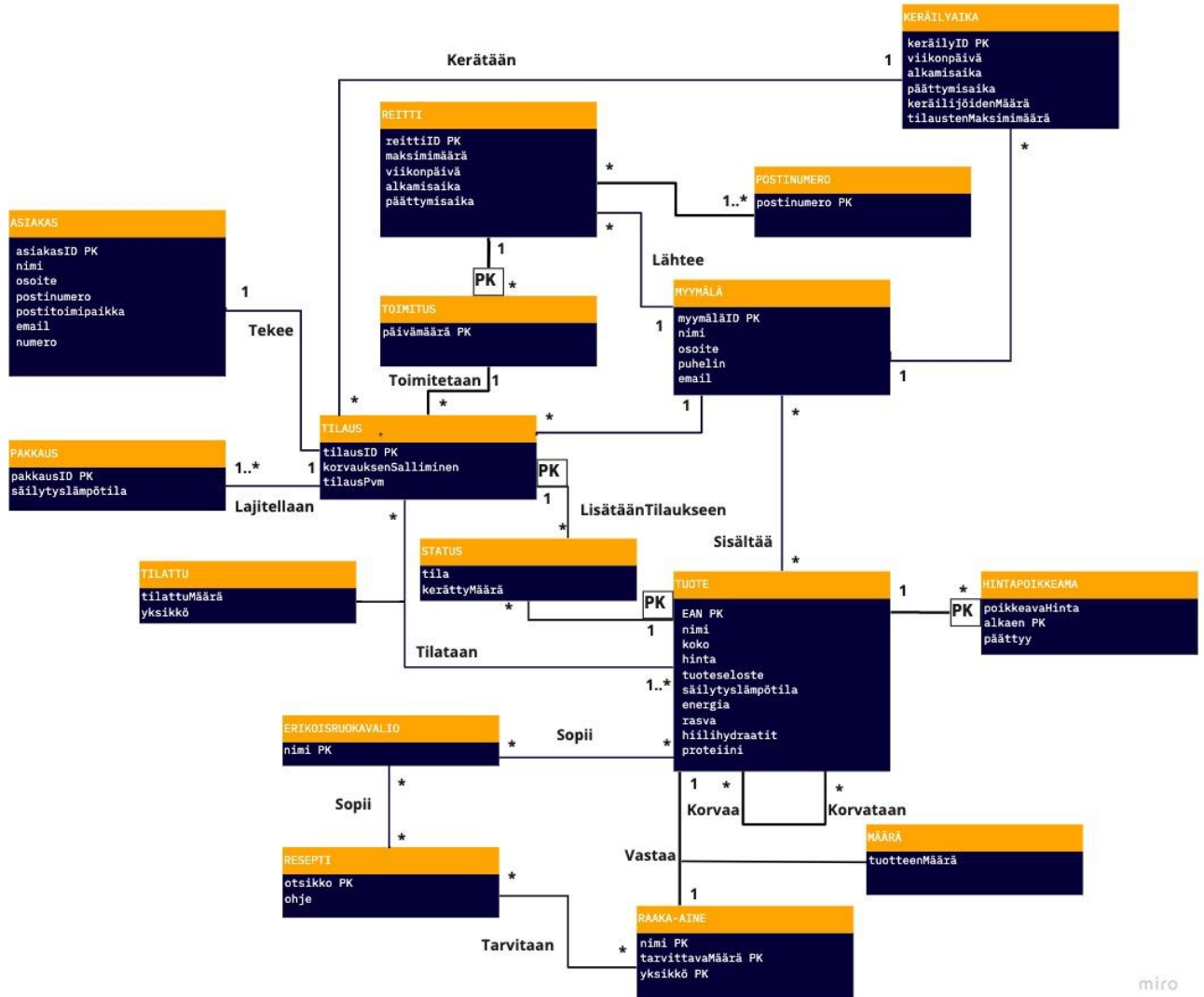
eetu.mustonen@aalto.fi

nea.nieminen@aalto.fi

julia.sippala@aalto.fi

Harjoitustyön ensimmäinen osa

1.UML-kaavio



2. Relaatiokaavio

Asiakas(asiakasID, nimi, osoite, postinumero, postitoimipaikka, email, puhNro)

Myymälä(myymäläID, nimi, osoite, puhNro, email)

Tilaus(tilausID, korvauksenSalliminen, tilausPvm, asiakasID, päivämäärä, reittiID, keräilyID, myymäläID)

Status(tilausID, EAN, tila, kerättyMäärä)

Pakkaus(pakkausID, säilytyslämpötila, tilausID)

Toimitus(päivämäärä, reittiID)

Reitti(reittiID, maksimimäärä, viikonpäivä, alkamisaika, päättymisaika, myymäläID)

kattaaNumerot(reittiID, postinumero)

Keräilyaika(keräilyID, viikonpäivä, alkamisaika, päättymisaika, keräilijöidenMäärä, tilaustenMaksimimäärä, myymäläID)

Tuote(EAN, nimi, koko, hinta, tuoteseloste, säilytyslämpötila, energia, rasva, hiilihydraatit, proteiini)

KuuluuTilaukseen(tilausID, EAN, tilattuMäärä, yksikkö)

VoiKorvata(korvattavaEAN, korvaavaEAN)

Hintapoikkeama(EAN, alkaen, päättyy, poikkeavaHinta)

Resepti(otsikko, ohje)

Raaka-aine(nimi, tarvittavaMäärä, yksikkö, EAN, tuotteenMäärä)

KuuluuReseptiin(otsikko, nimi, tarvittavaMäärä, yksikkö)

Erikoisruokavalio(nimi)

ReseptinErikoisruokavalio(otsikko, nimi)

TuotteenErikoisruokavalio(EAN, nimi)

3. Selostus ratkaisusta

3.1. Ratkaisujen perustelut

Asiakas

UML-mallissa on asiakkaiden tietojen säilyttämiselle oma luokkansa, sillä se on oma selkeä kokonaisuutensa. Tietokannassa pitää myös olla mahdollista lisätä uusia asiakkaita, ja tämän voi ratkaista selkeimmin sillä, että Asiakas-luokkaan lisätään vain uusi rivi, josta löytyvät asiakkaan uniikki asiakasID, hänen nimensä sekä osoite- ja yhteystiedot. AsiakasID on luokan avain, sillä nimiä ja osoitteita voi olla samoja, mutta jokaisella asiakkaalla on oma asiakasID. Asiakkaat voivat tehdä mielivaltaisen määrän tilauksia ja kaikilla tilauksilla on oltava tasan yksi asiakas, joten Asiakas-luokalla on yhdestä moneen -assosiaatio Tilaus-luokkaan.

Tilaus

Tilauksia pitää tietysti voida luoda lisää, kun asiakas tilaa netistä ruokaa. Täten Tilauksen on oltava oma luokkansa. Tilaukselle määräytyy myös oma tilausID, joka on tilauksen avain, sillä se on uniikki jokaiselle tilaukselle. Tehtävänannon perusteella oletetaan, että tilauksen toimitusosoite saadaan suoraan Asiakas-luokan tiedoista, eikä asiakas voi jokaisen tilauksen kohdalla erikseen valita toimitusosoitetta. Asiakas valitsee tilausta tehdessään, salliiko hän korvaukset tilauksessaan, kuten tehtävänannossa on määritetty, ja täten määrittyy korvauksenSalliminen-attribuutin arvoksi kyllä tai ei. Tilauksen teon päivämäärä tallettuu tilausPvm-attribuuttiin. Tilausta tehdessä asiakas valitsee, mistä myymälästä hän haluaa tilata (tehtävänannon mukaisesti). Myymälällä voi kuitenkin olla mielivaltaisen määrä tilauksia. Näiden ehtojen vuoksi Tilaus-luokka yhdistyy monesta yhteen-assosiaatiolla Myymälä-luokkaan.

Reitti

Reitti lähtee aina jostain tietystä myymälästä, mutta myymälällä voi olla mielivaltaisen määrä toimituksia. Tämän vuoksi Reitti-luokka yhdistyy Myymälä-luokkaan monesta yhteen-assosiaatiolla. Reitti-luokalla on avaimena reittiID, joka on uniikki kaikille Toimitus-luokan instansseille. Tehtävänannon vaatimusten mukaisesti toimitusreiteillä on maksimimäärä tilauksia, mitä toimitukseen mahtuu. Tätä merkitsee Reitti-luokan maksimimäärä-attribuutin arvo. Lisäksi toimituksella on tietoina viikonpäivä, aikaikkuna ja postinumerot, joihin toimitetaan (erillisenä luokkana). Nämä pysyvät samoina viikosta toiseen. Aikaikkuna on

tehty tiedonhaun yksinkertaistamiseksi kahtena attribuuttina, alkamisaikana ja päättymisaikana.

Postinumerot

Mikäli postinumerot sijoitettaisiin Toimitus-luokkaan, niiden pitäisi olla merkkijonolistana, joka ei olisi optimaalista tietokannan toiminnan vuoksi. Tämän vuoksi olemme tehneet postinumeroille oman luokan, jotta attribuuttien arvot pysyvät atomisina.

Postinumero-luokalla on vain yksi attribuutti, postinumero, sillä muuta tietoa ei tarvita ja se riittää tekemään arvoista uniikkeja, joten se toimii myös luokan avainattribuuttina.

Postinumero-luokka yhdistyy Toimitus-luokkaan monesta moneen -assosiaatiolla, mutta kuitenkin siten, että Toimitus-luokan instansseilla pitää olla ainakin yksi postinumero johon ne yhdistyvät (ei voi olla toimitusta joka ei toimita mihinkään). Yksi postinumero voi kuitenkin olla monella eri reitillä, ja toisaalta voi olla postinumeroita, jotka eivät ole millään reitillä tietyllä ajanhetkellä. Tämän vuoksi Postinumero-luokan instanssilla voi olla mielivaltainen määrä assosiaatioita Toimitus-luokan instansseihin.

Toimitus

Koska reitti aikoiheen, postinumeroiheen ja maksimimäärineen on sama kaikilla viikoilla tehtävänannon mukaisesti, olemme toisteisuuden välttämiseksi tehneet sille oman luokansa Reitti. Yksilöllisen toimituskerran reitillä spesifioi kuitenkin luokka Toimitus, jossa on attribuuttina vain päivämäärä. Päivämäärä yksin ei riitä tekemään tilauskerrasta uniikkia, eikä se siten riitä avaimeksi, minkä takia Toimitus-luokalla on avainattribuuttina päivämäärä-attribuutin lisäksi myös Reitti-luokan reittiID. Toimitus-luokan avulla saadaan tieto tietystä toimituskerrasta reitillä, jota tarvitaan tilauksen yhteyteen, koska pitää olla mahdollista saada lista kaikista toimituksen tilauksista sekä tarkastella tilauksen tietoja jälkikäteen. Toimitus-luokka liittyy Reitti-luokkaan monesta yhteen assosiaatiolla, koska tietyllä toimituskerralla on oltava tasan yksi reitti, mutta reitillä voi olla mielivaltainen määrä toimituksia. Toimitusta valittaessa tilauksen yhteydessä asiakas voi valita vain sellaisen toimituksen, jonka reitillä on assosiaationa Postinumero-luokasta sama postinumero-attribuutti kuin asiakkaan tiedoissa oleva postinumero

Keräily

Keräilyaika-luokalla on oma keräilyID, koska muut luokan attribuutit (viikonpäivä, alkamisaika, päättymisaika, keräilijöidenMäärä ja tilaustenMaksimimäärä) eivät riitä tekemään instansseista uniikkeja (eri myymälöillä voi olla samat arvot). Kuitenkin tilausta tehdessä on tärkeää, että eri myymälöiden instanssit ovat erikseen (tästä lisää seuraavassa kappaleessa). Attribuuttien arvot pysyvät tehtävänannon määritelmien perusteella samana

viikosta toiseen, joten tarkempaa päivämäärää ei tarvita, vaan samat instanssit pätevät viikosta toiseen. Keräilyaika-luokan instanssilla on aina yksi myymälä, mihin se liittyy, mutta myymälällä voi olla mielivaltainen määrä keräilyaikoja. Täten Keräilyaika-luokka yhdistyy Myymälä-luokkaan monesta yhteen -assosiaatiolla. Päädyimme siihen, että keräilijöille ei tehtävänannon vaatimusten mukaan tarvita muuta tietoa tietokannassa kuin sen, kuinka monta heitä on milloinkin, jotta sen perusteella voidaan määrittää tilausten maksimimäärä kyseisessä aikaikkunassa.

Toimitusajan valitseminen

Tilausta tehdessä asiakas valitsee toimitusajan siten, että hän voi valita vain aikaikkunan, joka toteuttaa seuraavan: Reitti-luokalla on olemassa instanssi, jossa alkamisaika- ja päättymisaika-attribuutit muodostavat halutun aikaikkunan. Reitti-luokan instanssissa postinumerot-attribuutin tulee sisältää tilauksen postinumero. Toimitus-luokan instanssissa, joka on halutulla päivämäärällä ja assosioitunut haluttuun Reitti-luokan instanssiin, ei vielä saa olla assosiaationa Reitti-luokan maksimimäärä-attribuutin suuruista määrää Toimitus-luokan instansseja. Ehtona tilauksen tekemiselle kyseiseen aikaikkunaan on myös, että on olemassa Keräilyaika-luokan instanssi, joka on samana päivänä kuin toimitusaikaikkuna, siten että Keräilyaika-luokan instanssin päättymisaika on ennen Toimitus-luokan instanssin alkamisaika-attribuuttia. Keräilyaika-luokan instanssilla ei myöskään saa vielä olla tilausten Maksimimäärä-attribuutin arvon suuruista määrää assosiaatioita Tilaus-luokan instansseihin. Nämä perustuvat tehtävänannon vaatimuksiin.

Pakkaus

Kerääjä pakkaa tuotteita eri pakkauksiin tilauksen ja säilytyslämpötilan mukaan. Tehtävänannon mukaan pitää pystyä "tehdä lista, jossa on määrättyyn kuljetukseen kuuluvat tilaukset ja pakkaukset, joita kuhunkin tilaukseen kuuluu". Pakkaus on oma luokkansa, koska tilaukseen voi kuulua monta eri pakkausta, joilla tulee kaikilla olla omat tietonsa (pakkausID ja säilytyslämpötila), jotta niistä voidaan tehdä lista. Tilauksen tuotteet lajitellaan pakkauksiin, joten tilauksella voi olla mielivaltainen määrä pakkauksia (kuitenkin vähintään yksi). Yhdessä pakkauksessa voi olla vain yhden tilauksen tuotteita, joten jokaisella pakkauksella on tasan yksi tilaus. Täten Pakkaus-luokan ja Tilaus-luokan välillä on monesta yhteen -assosiaatio.

Myymälä ja valikoima

Myymälä -luokan attribuutteja ovat pääavaimena toimivan myymäläID:n lisäksi myymälän nimi, osoite ja yhteystiedot. Koska myymälä on oma luokkansa, uusia myymälöitä on helppoa lisätä. Myymälän valikoima koostuu tuotteista, ja luokkien välillä on monesta moneen -assosiaatio, sillä valikoimassa on monia tuotteita ja tuotteet voivat kuulua moneen

valikoimaan. Relaatiomallissa assosiaatiota vastaa relaatio KuuluuValikoimaan, jonka avulla tuotteita voidaan myös lisätä tai poistaa valikoimista.

Tuotteet ja tilaukset

Tuote-luokan attribuutteina ovat tuotteen perustietojen (nimi, koko, hinta, säilytyslämpötila) lisäksi tuoteseloste sekä ravintosisältö. Päätimme sisällyttää ravintosisällön tuote-luokkaan, sillä voimme olettaa, että kaikilla verkkokaupan tuotteilla on ravintosisältö, jolloin emme tarvitse tähän erillistä luokkaa. Tuoteseloste on merkkijono, josta selviää tuotteen raaka-aineet siinä järjestyksessä, että ensimmäisenä on raaka-aine, jota on eniten. Tuotteen pääavaimena toimii EAN-koodi, sillä se on uniikki jokaisen tuotteen kohdalla. Tuote-luokkaan voi myös helposti lisätä uusia tuotteita.

Tuote-luokassa on itseassosiaatio, sillä tuotteet voivat korvata toisia tuotteita. Tuote voi samaan aikaan korvata useampaa tuotetta, ja tuotteella voi olla monta korvaavaa tuotetta, joten kyseessä on monesta-moneen -assosiaatio. Relaatiomallissa tätä kuvaa relaatio VoiKorvata, jonka avulla voidaan etsiä korvaavia tuotteita.

Tuotteiden kohdalla voi olla hintapoikkeamia, kuten alennuksia, jotka ovat voimassa tietyn ajan. Nämä löytyvät Hintapoikkeama-luokasta, jonka pääavaimena toimivat poikkeavan hinnan alkamispäivämäärä sekä tuotteen pääavaimena toimiva EAN-koodi. Tähän luokkaan tallentuvat myös kaikki vanhat hintapoikkeamat ja niiden voimassaoloajat, jotta ne voidaan tarkistaa myöhemmin tarpeen tullen. Hintapoikkeamat liittyvät tuotteisiin monesta yhteen -assosiaatiolla, sillä yhdellä tuotteella voi olla monta hintapoikkeamaa, mutta yksi hintapoikkeama voi koskea vain yhtä tuotetta.

Tilauksen ja tuotteen välillä on monesta moneen -assosiaatio, sillä tuotteet voivat kuulua moneen tilaukseen ja tilauksiin voi kuulua monta tuotetta. Tilauksessa on kuitenkin oltava vähintään yksi tuote. Luokkien välillä on myös assosiaatioluokka Tilattu, jonka avulla tuotteita voidaan lisätä tilaukseen. Attribuutit tilattuMäärä ja yksikkö kertovat, kuinka paljon tuotetta on tilattu (esim. 2 kappaletta tai 400 grammaa).

Lisäksi tuotteen ja tilauksen välillä on Status-luokka, jonka attribuutteina on tuotteen sen hetkinen tila sekä kerätty määrä. Sen avaimina toimivat Tilaus- ja Tuote-luokkien avaimet, sillä niin voidaan linkittää status tiettyyn tilaukseen ja tuotteeseen. Tilatun tuotteen status voi olla *odottaa keräystä*, *korvattu*, *korvaava*, *poistettu* tai *ok*. Jos status on *korvattu*, kyseinen tuote on ollut keräilyn aikana loppu ja se on korvattu jollain toisella. Jos tilauksessa on valittu, että tuotetta ei saa korvata ja tuote on loppu, sitä ei voida kerätä tilaukseen ja status

on *poistettu*. Status auttaa pitämään keräyksen aikana kirjaa siitä, mitkä tuotteet on kerätty ja mitkä odottavat vielä keräystä. Lisäksi statuksen avulla voidaan tarkistaa, mitkä tuotteet on korvattu, mitkä tuotteet korvaavat alkuperäisiä tuotteita ja mitkä ovat jääneet kokonaan pois tilauksesta. Näiden avulla voidaan laskea tilauksen alkuperäisen hinnan lisäksi lopullinen hinta keräilyn jälkeen, sekä tehdä listat siitä, mitkä tuotteet on korvattu ja mitkä tuotteet korvaavat alkuperäisiä tuotteita sekä mitkä tuotteet on jätetty kokonaan pois.

Reseptit

Resepti-luokassa attribuutteina ovat reseptin otsikko sekä ohje merkkijonona. Otsikko toimii luokan pääavaimena, sillä voimme olettaa, että samalla nettisivulla ei ole kahta täysin samannimistä reseptiä. Resepti-luokkaan liittyy Raaka-aine -luokka, jonka attribuutteina ovat nimi, reseptiin tarvittava määrä sekä yksikkö (esim. kpl tai g). Nämä kaikki tarvitaan luokan avaimiksi. Luokassa voi olla esimerkiksi samannimisiä raaka-aineita, joita tarvitaan eri määriä, mutta tarvittavaMäärä-attribuutti voi olla sama, vaikka yksikkö on eri. Reseptit ja raaka-aineet liittyvät toisiinsa monesta moneen -assosiaatiolla, sillä jokaisessa reseptissä voi olla monta raaka-ainetta, ja jokainen raaka-aine voi kuulua moneen reseptiin. Tätä kuvaa relaatiomallissa relaatio KuuluuReseptiin.

Jokaista reseptin raaka-ainetta vastaa yksi tuote, jotka voidaan kerralla lisätä tilaukseen verkkokaupassa. Näin ollen Raaka-aine - ja Tuote-luokat on liitetty toisiinsa yhdestä yhteen -assosiaatiolla. Näiden välillä on assosiaatioluokka Määrä, joka kertoo, kuinka paljon tuotetta pitää lisätä tilaukseen. Jos tuotetta tarvitaan esimerkiksi 300 grammaa ja sitä myydään 200 gramman pakeeteissa, määrä pyöristyy ylöspäin ja tilaukseen lisätään kaksi pakettia.

Erikoisruokavaliot

Erikoisruokavalioille päätimme tehdä oman luokkansa, jonka attribuuttina on vain ruokavalion nimi (esim. laktoositon). Koska erikoisruokavaliot ovat oma luokkansa, uusia erikoisruokavalioita voi lisätä helposti. Luokka on liitetty monesta moneen -assosiaatiolla Tuote- ja Resepti-luokkiin, sillä sekä tuotteet että reseptit voivat sopia moneen erityisruokavalioon ja erityisruokavaliot voivat sopia moneen tuotteeseen ja reseptiin. Relaatiomallissa näitä riippuvuuksia vastaavat relaatiot ReseptinErikoisruokavalio sekä TuotteenErikoisruokavalio. Näiden avulla voidaan lisätä uusia tuotteita tai reseptejä erikoisruokavalioon ja hakea johonkin erikoisruokavalioon kuuluvia tuotteita ja reseptejä.

3.2. Riippuvuudet

FUNKTIONAALISET RIIPPUVUUDET

Jos relaation funktionaalisten riippuvuuksien kaikki vasemmat puolet ovat yliavaimia, eli määrittävät kaikki muut relaation attribuutit, niin relaatio on Boyce-Codd -normaalimuodossa.

Asiakas(asiakasID, nimi, osoite, postinumero, postitoimipaikka, email, puhNro):

asiakasID -> nimi, osoite, postinumero, postitoimipaikka, email, puhNro

On Boyce-Codd -normaalimuodossa

Myymlä(myymäläID, nimi, osoite, yhteystiedot, valikoimaID):

myymäläID -> nimi, osoite, yhteystiedot, valikoimaID

On Boyce-Codd -normaalimuodossa

Tilaus(tilausID, korvauksenSalliminen, tilausPvm, asiakasID, reittiID, päivämäärä, keräilyID, myymäläID):

tilausID -> korvauksenSalliminen, tilausPvm, asiakasID, reittiID, päivämäärä, keräilyID, myymäläID

Relaatio on Boyce-Codd -normaalimuodossa.

Status(tilausID, EAN, tila, kerättyMäärä)

tilausID, EAN -> tila, kerättyMäärä

Relaatio on Boyce-Codd -normaalimuodossa.

Pakkaus(pakkausID, säilytyslämpötila, tilausID):

pakkausID -> säilytyslämpötila, tilausID

Relaatio on Boyce-Codd -normaalimuodossa.

Toimitus(päivämäärä, reittiID)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

Reitti(reittiID, maksimimäärä, viikonpäivä, alkamisaika, päättymisaika, myymäläID):

reittiID -> maksimimäärä, viikonpäivä, alkamisaika, päättymisaika, myymäläID

Relaatio on Boyce-Codd -normaalimuodossa.

Postinumero(postinumero)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

kattaaNumerot(reittiID, postinumero)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

Keräilyaika(keräilyID, viikonpäivä, alkamisaika, päättymisaika, keräilijöidenMäärä, tilaustenMaksimimäärä, myymäläID):

keräilyID -> viikonpäivä, alkamisaika, päättymisaika, keräilijöidenMäärä, myymäläID

keräilijöidenMäärä -> tilaustenMaksimimäärä

Relaatio on Boyce-Codd -normaalimuodossa.

Tuote(EAN, nimi, koko, hinta, tuoteseloste, säilytyslämpötila, energia, rasva, hiilihydraatit, proteiini):

EAN -> nimi, koko, hinta, tuoteseloste, säilytyslämpötila, energia, rasva, hiilihydraatit, proteiini

Relaatio on Boyce-Codd -normaalimuodossa.

KuuluuTilaukseen(tilausID, EAN, tilattuMäärä, yksikkö)

tilausID, EAN -> tilattuMäärä, yksikkö

Relaatio on Boyce-Codd -normaalimuodossa.

VoiKorvata(korvattavaEAN, korvaavaEAN):

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

Hintapoikkeama(EAN, alkaen, päättyy, poikkeavaHinta):

EAN, alkaen -> päättyy, poikkeavaHinta

Relaatio on Boyce-Codd -normaalimuodossa.

Resepti(otsikko, ohje)

otsikko -> ohje

Relaatio on Boyce-Codd -normaalimuodossa.

Raaka-aine(nimi, tarvittavaMäärä, yksikkö, EAN, tuotteenMäärä)

Nimi, tarvittavaMäärä, yksikkö -> EAN, tuotteenMäärä

Relaatio on Boyce-Codd -normaalimuodossa.

KuuluuReseptiin(otsikko, nimi, tarvittavaMäärä, yksikkö)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

Erikoisruokavalio(nimi)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

ReseptinErikoisruokavalio(otsikko, nimi)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

TuotteenErikoisruokavalio(EAN, nimi)

Ei epätriviaaleja riippuvuuksia

Relaatio on Boyce-Codd -normaalimuodossa.

1. UML-kaavio



2. Relaatiokaavio

Asiakas(asiakasID, nimi, osoite, postinumero, postitoimipaikka, email, puhNro)

Myymälä(myymäläID, nimi, osoite, puhNro, email)

KuuluuValikoimaan(myymäläID, EAN)

Tilaus(tilausID, korvauksenSalliminen, tilausPvm, asiakasID, toimitusPvm, reittiID, keräilyID, myymäläID)

Korvaus(korvausID, tilausID, alkuperäisenEAN, korvaavanEAN)

Pakkaus(pakkausID, säilytyslämpötila, tilausID)

Toimituskerta(toimitusPvm, reittiID)

Toimitus(reittiID, maksimimäärä, viikonpäivä, alkamisaika, päättymisaika, myymäläID)

KattaaNumerot(reittiID, postinumero)

Postinumero(postinumero)

Keräilyaika(keräilyID, viikonpäivä, alkamisaika, päättymisaika, keräilijöidenMäärä, tilaustenMaksimimäärä, myymäläID)

Tuote(EAN, nimi, koko, säilytyslämpötila, energia, rasva, hiilihydraatit, proteiini)

KuuluuTilaukseen(tilausID, EAN, tilattuMäärä, yksikkö, kerätty)

VoiKorvata(korvattavaEAN, korvaavaEAN)

Hinta(EAN, hinta, alkaen, päättyy)

Raaka-aine(aineID, nimi)

TuotteenAineet(EAN, aineID, monesko)

Resepti(otsikko, ohje)

ReseptinOsa(nimi, tarvittavaMäärä, yksikkö, EAN, tuotteenMäärä)

KuuluuReseptiin(otsikko, nimi, tarvittavaMäärä, yksikkö)

Erikoisruokavalio(nimi)

ReseptinErikoisruokavalio(otsikko, nimi)

TuotteenErikoisruokavalio(EAN, nimi)

3. Selostus tehdyistä muutoksista

Korvasimme Hintapoikkeama-luokan Hinta-luokalla, jonka attribuutteina ovat hinta, alkamispäivämäärä (toimii pääavaimena yhdessä Tuote-luokan EAN-koodin kanssa) ja päättymispäivämäärä. Ensimmäisessä versiossa tuotteen "normaalihinta" oli tallennettuna Tuote-luokassa ja hintapoikkeamat kuten alennukset erillisessä Hintapoikkeama-luokassa. Hinta-attribuutti poistettiin Tuote-luokasta kokonaan, ja sen sijaan voimassaolevan hinnan voi löytää Hinta-luokasta, johon tallentuu myös kaikki menneet hinnat. Tämä yksinkertaistaa hinnan etsimistä, sillä näin voidaan hakea suoraan voimassaoleva tai tiettynä aikana voimassa ollut hinta, eikä tarvitse aina erikseen tarkistaa, onko tuotteella hintapoikkeamaa.

Ensimmäisen palautuksen palautteessa sanottiin: "Tiedot korvatuista tuotteista tarvitsevat oman luokan. Kun kerääjä huomaa jonkin tuotteen olevan lopussa, hän tekee valinnan usean mahdollisen korvaavan tuotteen välillä. Tämä valinta tallennetaan omaan luokkaansa yhdessä korvatusen tuotteen kanssa. Tällä korvattujen tuotteiden luokalla on kaksi assosiaatiota Tuote-luokkaan. Ensimmäisen kautta saadaan korvattavan tuotteen avainattribuutti ja toisen kautta korvaavan tuotteen avainattribuutti. Lisäksi tarvitaan assosiaatio tilausluokkaan.". Teimme tämän palautteen perusteella Korvaus-luokan. Poistimme kokonaan Status-luokan, josta selvisi tuotteen tila (*odottaa keräystä, korvattu, korvaava, poistettu* tai *ok*) ja kerätty määrä, koska se vaikutti palautteen perusteella tehtyjen muutosten jälkeen turhalta. Status-luokassa aiemmin ollut attribuutti *kerätty* siirrettiin Tilattu-assosiaatioluokkaan, sillä se vaikutti siihen sopivalta.

Mielestämme ensimmäisen osan Toimitus- ja Reitti-luokat olivat kaikista yksinkertaisin ja toimivin tapa kuvata viikoittain toistuvia toimitusreittejä niin, että tehtävänannon vaatimukset täyttyvät. Muutimme kuitenkin luokkien nimet paremmin luokkien käyttötarkoitusta kuvaaviksi Toimitus- ja Toimituskerta-luokiksi. Toimitus-luokan instanssit kuvaavat viikoittain toistuvia toimituksia, joilla on tietty aikaväli, viikonpäivä ja maksimimäärä. Tähän liittyy Postinumero-luokka, jonka assosiaatiot kuvaavat toimituksen maantieteellistä reittiä (eli mihin postinumeroille toimituksessa voidaan toimittaa). Sama reitti voi toistua useana

viikonpäivänä tai aikana, mutta ne kuvataan Toimitus-luokassa omina instansseinaan. Toimituspäivä yksilöi tietyn toimitusajan jonka asiakas valitsee.

Muutimme Tuotteen ja Myymälän välillä olevan Sisältää-assosiaation nimeksi KuuluuValikoimaan, sillä koimme sen olevan paremmin kuvaava. Lisäksi lisäsimme tätä assosiaatiota vastaavan relaation relaatiomalliimme, josta se aiemmin puuttui.

Tuotteesta poistettiin attribuutti "tuoteseloste" ja se korvattiin omalla luokalla "Raaka-aine", jossa on attribuutteina aineID (pääavain) ja raaka-aineen nimi, sillä palautteen perusteella Raaka-aineen kuuluu olla oma luokkansa. Assosiaatioluokka "Järjestys", kertoo monenneksiko eniten kyseistä raaka-ainetta tuote sisältää. Alkuperäinen Raaka-aine -luokka kuvasi reseptissä käytettyjä raaka-aineita, joten muutimme tämän luokan nimeksi "Reseptin osa".

<https://drive.google.com/drive/folders/1e99GeGOTTKRlpMmlAyJI8OgFxvbDt2Sp?usp=sharing>

Harjoitustyön toinen osa

1. SQL

1.1 Tietokannan luominen

Kaikissa attribuuteissa, joiden arvot olivat selkeästi merkkijonoja, käytimme SQLiten TEXT-tietotyyppiä (esim nimi, osoite jne). Käytimme puhelinnumeroille, postinumeroille, EAN:lle ja tietokannan eri ID:ille TEXT-tietotyyppiä, sillä niitä ei tarvitse käyttää laskutoimituksissa ja ne voivat alkaa nollalla. Esimerkiksi Tilaus-tilun korvauksen Salliminen-attribuutissa käytimme tietotyyppinä INTEGERiä, määritellen mahdollisiksi arvoiksi 1 ja 2, sillä SQLitessä ei ole tietotyyppiä totuusarvoille. Päivämäärille ja ajoille käytimme TEXT-tietotyyppiä, sillä ne ovat merkkijonoja (esim 2021-05-05 ja 23.15). Ajankohtien arvoille laitoimme ehdoksi, että ne ovat suurempia tai yhtä suuria kuin 00.00 ja pienempiä kuin 24.00 (merkkijonon vertailu). Viikonpäivälle käytimme TEXT-tietotyyppiä sillä ehdolla, että arvo on ma, ti, ke jne. Koko, energia, tilausten Maksimimäärä jne ovat REAL tai INTEGER tietotyyppiä riippuen siitä, tarvitaanko desimaalilukua (koko, energia,...) vai kokonaislukua (maksimimäärät,...).

Tietokannan luomiseen käytetyt käskyt:

```
CREATE TABLE Asiakas (  
    asiakasID TEXT PRIMARY KEY,  
    nimi TEXT NOT NULL,  
    osoite TEXT NOT NULL,  
    postinumero TEXT NOT NULL,  
    postitoimipaikka TEXT NOT NULL,  
    email TEXT NOT NULL,  
    puhNro TEXT NOT NULL  
);
```

```
CREATE TABLE Myymälä(  
    myymäläID TEXT PRIMARY KEY,  
    nimi TEXT NOT NULL,
```

```
        osoite TEXT NOT NULL,  
        email TEXT NOT NULL,  
        puhNro TEXT NOT NULL  
);
```

```
CREATE TABLE KuuluuValikoimaan(  
        myymäläID TEXT REFERENCES Myymälä(myymäläID),  
        EAN TEXT REFERENCES Tuote(EAN),  
        PRIMARY KEY (myymäläID, EAN)  
);
```

```
CREATE TABLE Tilaus (  
        tilausID TEXT PRIMARY KEY,  
        korvauksenSalliminen INTEGER DEFAULT 1  
        CHECK (korvauksenSalliminen = 0 OR  
        korvauksenSalliminen = 1),  
        tilausPvm TEXT NOT NULL,  
        asiakasID TEXT,  
        toimitusPvm TEXT,  
        toimitusID TEXT,  
        keräilyID TEXT,  
        myymäläID TEXT,  
        FOREIGN KEY (  
        asiakasID  
        )  
        REFERENCES Asiakas (asiakasID),  
        FOREIGN KEY (  
        toimitusPvm,  
        toimitusID  
        )  
        REFERENCES Toimituskerta (toimitusPvm,  
        toimitusID),  
        FOREIGN KEY (  
        keräilyID  
        )  
        REFERENCES Keräilyaika (keräilyID),  
        FOREIGN KEY (  
        myymäläID  
        )  
        REFERENCES Myymälä (myymäläID)
```

```
myymäläID
)
REFERENCES Myymälä (myymäläID)
);
```

```
CREATE TABLE Korvaus(
    korvausID TEXT PRIMARY KEY,
    tilausID TEXT REFERENCES Tilaus(tilausID),
    alkuperäisenEAN REFERENCES Tuote(EAN),
    korvaavanEAN REFERENCES Tuote(EAN) DEFAULT NULL
);
```

```
CREATE TABLE Toimitus(
    toimitusID TEXT PRIMARY KEY,
    maksimimäärä INTEGER,
    viikonpäivä TEXT,
    alkamisaika TEXT,
    päättymisaika TEXT,
    myymäläID TEXT REFERENCES Myymälä(myymäläID)
    CHECK (viikonpäivä = 'ma' OR viikonpäivä = 'ti' OR viikonpäivä = 'ke' OR viikonpäivä
= 'to' OR viikonpäivä = 'pe' OR viikonpäivä = 'la' OR viikonpäivä = 'su'),
    CHECK (alkamisaika > 00.00 AND alkamisaika < 24.00),
    CHECK (päättymisaika > 00.00 AND alkamisaika < 24.00)
);
```

```
CREATE TABLE Toimituskerta(
    toimitusPvm TEXT NOT NULL,
    toimitusID TEXT REFERENCES Toimitus(toimitusID),
    PRIMARY KEY (toimitusPvm, toimitusID)
);
```

```
CREATE TABLE Postinumero(
    postinumero TEXT PRIMARY KEY
);
```

```
CREATE TABLE KattaaNumerot(  
    toimitusID TEXT REFERENCES Toimitus(toimitusID),  
    postinumero TEXT REFERENCES Postinumero(postinumero),  
    PRIMARY KEY (toimitusID, postinumero)  
);
```

```
CREATE TABLE Pakkaus(  
    pakkausID TEXT PRIMARY KEY,  
    säilytyslämpötila INTEGER,  
    tilausID TEXT REFERENCES Tilaus(tilausID)  
);
```

```
CREATE TABLE Keräilyaika(  
    keräilyID TEXT PRIMARY KEY,  
    viikonpäivä TEXT NOT NULL,  
    alkamisaika TEXT CHECK (alkamisaika > 00.00 AND alkamisaika < 24.00),  
    päättymisaika TEXT CHECK (päättymisaika > 00.00 AND päättymisaika < 24.00),  
    keräilijöidenMäärä INTEGER NOT NULL,  
    tilaustenMaksimimäärä INTEGER NOT NULL,  
    myymäläID TEXT REFERENCES Myymälä(myymäläID)  
);
```

```
CREATE TABLE Tuote(  
    EAN TEXT PRIMARY KEY,  
    nimi TEXT NOT NULL,  
    koko REAL,  
    säilytyslämpötila REAL NOT NULL,  
    energia REAL NOT NULL,  
    rasva REAL NOT NULL,  
    hiilihydraatit REAL NOT NULL,  
    proteiini REAL NOT NULL  
);
```

```
CREATE TABLE VoiKorvata (  
  korvattavaEAN REFERENCES Tuote (EAN),  
  korvaavaEAN REFERENCES Tuote (EAN) DEFAULT NULL,  
  PRIMARY KEY (korvattavaEAN, korvaavaEAN)  
);
```

```
CREATE TABLE KuuluuTilaukseen (  
  tilausID TEXT REFERENCES Tilaus (tilausID)  
  NOT NULL,  
  EAN TEXT REFERENCES Tuote (EAN)  
  NOT NULL,  
  tilattuMäärä REAL NOT NULL,  
  yksikkö TEXT NOT NULL,  
  kerätty INTEGER CHECK (kerätty IN (0, 1) )  
  DEFAULT 0,  
  PRIMARY KEY (tilausID, EAN)  
);
```

```
CREATE TABLE Hinta(  
  EAN TEXT REFERENCES Tuote(EAN),  
  hinta REAL NOT NULL,  
  alkaen TEXT,  
  päättyy TEXT NOT NULL,  
  PRIMARY KEY (EAN, alkaen)  
);
```

```
CREATE TABLE RaakaAine(  
  aineID TEXT PRIMARY KEY,  
  nimi TEXT NOT NULL  
);
```

```
CREATE TABLE TuotteenAineet(  
  EAN TEXT REFERENCES Tuote(EAN),  
  aineID TEXT REFERENCES RaakaAine(aineID),  
  monesko INTEGER NOT NULL,
```

```
PRIMARY KEY (EAN, aineID)  
);
```

```
);
```

```
CREATE TABLE Resepti (  
    otsikko TEXT PRIMARY KEY,  
    ohje TEXT  
);
```

```
CREATE TABLE KuuluuReseptiin(  
    otsikko TEXT REFERENCES Resepti(otsikko),  
    nimi TEXT REFERENCES ReseptinOsa(nimi),  
    tarvittavaMäärä REAL REFERENCES ReseptinOsa(tarvittavaMäärä),  
    yksikkö TEXT REFERENCES ReseptinOsa(yksikkö)  
);
```

```
CREATE TABLE Erikoisruokavalio(  
    nimi TEXT PRIMARY KEY  
);
```

```
CREATE TABLE ReseptinErikoisruokavalio(  
    otsikko TEXT REFERENCES Resepti(otsikko),  
    nimi TEXT REFERENCES Erikoisruokavalio(nimi),  
    PRIMARY KEY (otsikko, nimi)  
);
```

```
CREATE TABLE TuotteenErikoisruokavalio(  
    EAN TEXT REFERENCES Tuote(EAN),  
    nimi TEXT REFERENCES Erikoisruokavalio(nimi),  
    PRIMARY KEY (EAN, nimi)  
);
```

```
CREATE TABLE ReseptinOsa(  
    nimi TEXT NOT NULL,  
    tarvittavamäärä REAL NOT NULL,
```

yksikkö TEXT NOT NULL,
EAN TEXT REFERENCES Tuote(EAN) NOT NULL,
tuotteenMäärä REAL NOT NULL,
PRIMARY KEY (nimi, tarvittavaMäärä, yksikkö)
);

1.2 Kyselyt ja hakemistot

Hakemisto

Hakemisto asiakkaista postinumeroitain kannattaa tehdä, jotta löydetään tehokkaasti samalla alueella asuvien asiakkaiden tiedot esimerkiksi viestintää varten. Asiakas-taulussa on todella paljon rivejä, jotka ovat jakautuneet usean postinumeron mukaan, jolloin ei kannata käydä läpi koko taulua etsiessä asiakkaita yhden postinumeron perusteella. Postinumero on tieto, jota ei kovin usein tarvitse hakemistoon ja tauluun päivittää.

```
CREATE INDEX AsiakkaatPostinumeroitain ON Asiakas(postinumero)
```

Hakemisto asiakkaista asiakasID:n mukaan kannattaa tehdä, jotta tilauksen tehneen asiakkaan tiedot voidaan hakea tehokkaasti. Asiakkaita on paljon ja veisi paljon aikaa löytää yksittäisen asiakkaan tiedot.

```
CREATE INDEX AsiakasID ON Asiakas(asiakasID)
```

Hakemisto tuotteen EAN:n mukaan kannattaa tehdä, jotta yksittäisen tuotteen tietoja voidaan päivittää tehokkaasti. Tuote-taulussa on todella paljon monikoita ja yhden tuotteen etsimiseen kuluisi todella paljon aikaa.

```
CREATE INDEX EAN ON Tuote(EAN)
```

Näkymät

Näkymä tuotteen tiedoista kannattaa tehdä, jotta asiakas voi tarkastella valikoimaa ja tuotteiden tietoja ilman pääsyä itse taulujen tietoihin. Näkymällä yhdistetään myös tuotteen

oleelliset tiedot eli EAN, nimi, koko, säilytyslämpötila, ravintoarvot, hinta, sekä erikoisruokavalio.

```
CREATE VIEW TuotteenTiedot AS
```

```
    SELECT Tuote.EAN, Tuote.nimi, koko, säilytyslämpötila, energia, rasva, hiilihydraatit,  
    proteiini, hinta, TuotteenErikoisruokavalio.nimi AS Erikoisruokavalio
```

```
    FROM Tuote, Hinta, TuotteenErikoisruokavalio
```

```
    WHERE Tuote.EAN = Hinta.EAN AND Hinta.EAN = TuotteenErikoisruokavalio.EAN
```

Käyttötapauksia

Reseptit ja niiden erikoisruokavaliot

Reseptit on hyvä jakaa eri erikoisruokavalioiden perusteella.

```
SELECT resepti.otsikko, ohje, nimi
```

```
FROM Resepti, ReseptinErikoisruokavalio
```

```
WHERE Resepti.otsikko = ReseptinErikoisruokavalio.otsikko;
```

	otsikko	ohje	nimi
1	Banaanipirtelö	Sekoita	G

Myymälöiden tuotteiden keskimääräinen hinta

Voidaan vertailla eri myymälöiden tuotteiden keskimääräistä hintaa ja huomata niissä poikkeamia.

```
SELECT myymäläID, myymälä.nimi, AVG(hinta.hinta)
```

```
FROM myymälä, kuuluuvalikoimaan, tuote, hinta
```


WHERE myymälä.myymäläID = kuuluuvalikoimaan.myymäläID AND
 kuuluuvalikoimaan.EAN = tuote.EAN AND tuote.EAN = hinta.EAN

GROUP BY myymälä.myymäläID;

	myymäläID	nimi	AVG(hinta.hinta)
1	3333	Marketti-Tapiola	1.03333333333333

Postinumerot joihin myymälä voi toimittaa

Voidaan tarkistaa, mille alueille myymälästä toimitetaan tilauksia.

SELECT myymälä.myymäläID, myymälä.nimi, postinumero.postinumero

FROM myymälä, toimitus, kattaaNumerot, postinumero

WHERE myymälä.myymäläID = toimitus.myymäläID AND toimitus.toimitusID =
 kattaaNumerot.toimitusID AND kattaaNumerot.postinumero = postinumero.postinumero;

	myymäläID	nimi	postinumero
1	3333	Marketti-Tapiola	00001
2	3333	Marketti-Tapiola	00002
3	3333	Marketti-Tapiola	00003

Millä jonkin tuotteen voi korvata

Esimerkkikysely kertoo, millä tuotteella banaanin voi korvata. Esimerkiksi keräilijät voivat tarvita tietoa siitä, millä tuotteella jonkin loppuneen tuotteen voi korvata.

SELECT nimi, korvattavaEAN ,korvaavaEAN

FROM VoiKorvata, Tuote

WHERE VoiKorvata.korvattavaEAN = Tuote.EAN AND VoiKorvata.korvattavaEAN = "1111";

	nimi	korvattavaEAN	korvaavaEAN
1	Banaani	1111	1112

Gluteenittomat tuotteet

Esimerkkikyselyssä etsitään kaikki gluteenittomat tuotteet.

```
SELECT nimi, EAN
FROM Tuote, TuotteenErikoisruokavalio
WHERE TuotteenErikoisruokavalio.nimi = "G" AND Tuote.EAN =
TuotteenErikoisruokavalio.EAN;
```

	nimi	EAN
1	Banaani	1111
2	LuomuBanaani	1112
3	Maito	1113

Myymälät joiden valikoimaan jokin tuote kuuluu

Esimerkkikyselyssä tarkastetaan minkä myymälän valikoimiin banaani kuuluu.

```
SELECT nimi, myymäläID
FROM Myymälä
WHERE EXISTS
( SELECT *
FROM KuuluuValikoimaan
WHERE EAN=1111 AND KuuluuValikoimaan.myymäläID = Myymälä.myymäläID
);
```

	nimi	myymäläID
1	Marketti-Tapiola	3333

Kerääjää varten lista

Mukana listassa ovat tilaukseen kuuluvat tuotteet, niiden määrät sekä tieto siitä, mihin lämpötilaan kukin tuote kuuluu. Niiden tietojen avulla kerääjä voi kerätä tuotteet.

```
SELECT nimi, Tuote.EAN, tilattuMäärä, yksikkö, säilytyslämpötila
FROM Tuote, KuuluuTilaukseen
WHERE KuuluuTilaukseen.tilausID = "6666" AND KuuluuTilaukseen.EAN = Tuote.EAN;
```

	nimi	EAN	tilattuMäärä	yksikkö	säilytyslämpötila
1	Banaani	1111	10	kpl	18
2	Maito	1113	2	litraa	6

Tuotteet jotka keräämättä

Kerääjien täytyy pystyä tarkistamaan, mitkä tuotteet on vielä keräämättä.

```
SELECT nimi, EAN
FROM Tuote, KuuluuTilaukseen
WHERE kerätty = 0 AND Tuote.EAN = KuuluuTilaukseen.EAN;
```

	nimi	EAN
1	Banaani	1111
2	Maito	1113

Tuotteet jotka korvattu

Voidaan tarkistaa, mitkä tuotteet tilauksesta on korvattu, ja mikä tuote on valittu korvaajaksi.

```
SELECT A1.nimi, A1.EAN, B1.nimi, B1.EAN
FROM Korvaus, Tuote AS A1, Tuote AS B1
WHERE alkuperäisenEAN = A1.EAN AND korvaavanEAN = B1.EAN;
```

	nimi	EAN	nimi:1	EAN:1
1	Banaani	1111	LuomuBanaani	1112

Tietyn päivän toimitusajat

Voidaan katsoa, mitä toimitusaikavälejä tiettynä päivänä on tarjolla.

```
SELECT alkamisaika, päättymisaika
FROM Toimitus, Toimituskerta
WHERE Toimitus.toimitusID = Toimituskerta.toimitusID AND Toimituskerta.toimitusPvm =
"2021-05-29" ;
```

	alkamisaika	päättymisaika
1	09.00	11.00
2	11.00	13.00

Montako tilausta on toimitettava tiettynä toimituskertana

Voidaan tarkistaa, montako tilausta tiettynä toimituskertana (päivä, aikaväli) täytyy toimittaa.

```
SELECT toimitusID, COUNT(*) as Määrä
FROM Toimituskerta, Tilaus
WHERE Toimituskerta.toimitusID = Tilaus.toimitusID;
```

	toimitusID	Määrä
1	4445	2

Kauppalista reseptin pohjalta (tuotteen nimi, määrä ja kokonaishinta tuotteelle)

Voidaan käyttää verkkokaupassa, jos asiakas valitsee "lisää reseptin ainesosat koriin" tms, tai kauppalistana asiakkaalle.

```
SELECT nimi, tuotteenMäärä, (hinta * tuotteenMäärä) AS Summa
FROM Tuote, Hinta, (SELECT tuotteenmäärä, EAN
FROM KuuluuReseptiin, ReseptinOsa
WHERE otsikko = "Banaanipirtelö" AND KuuluuReseptiin.nimi = ReseptinOsa.nimi AND
KuuluuReseptiin.tarvittavaMäärä = ReseptinOsa.tarvittavaMäärä AND
KuuluuReseptiin.yksikkö = ReseptinOsa.yksikkö) AS C1
WHERE Tuote.EAN = Hinta.EAN AND C1.EAN = Tuote.EAN ;
```

	nimi	tuotteenMäärä	Summa
1	Banaani	10	5
2	Maito	1	2

Tilaukset per asiakas

Tuottaa listan asiakkaista, ja siitä kuinka monta tilausta heillä on ollut. Voidaan käyttää esim markkinointia suunnitellessa.

```
SELECT asiakasID, COUNT (DISTINCT asiakasID)
FROM Tilaus
GROUP BY asiakasID;
```

	asiakasID	COUNT(DISTINCT asiakasID)
1	2222	1
2	2223	1

Tuotteiden hinnat keskimäärin

Tuottaa listan tuotteista ja niiden keskimääräisestä hinnasta (ottaa mukaan kaikki hinnat, joita kyseiselle tuotteelle on tallennettu menneisyydessä).

```
SELECT EAN, AVG(hinta)
FROM Hinta
GROUP BY EAN;
```

	EAN	AVG(hinta)
1	1111	0.5
2	1112	0.6
3	1113	2

Määrä aktiivisia asiakkaita postinumeroittain

Otetaan mukaan vain asiakkaat, jotka ovat tilanneet ainakin kerran. Voidaan tarvita esim reittien ja aikataulujen suunnittelussa.

```
SELECT postinumero, COUNT(DISTINCT Asiakas.asiakasID)
FROM Asiakas, Tilaus
WHERE Asiakas.asiakasID = Tilaus.asiakasID
GROUP BY postinumero;
```

	postinumero	COUNT(DISTINCT Asiakas.asiakasID)
1	00001	1
2	00002	1

Kaikkien asiakkaiden sähköpostit

Voidaan käyttää markkinointitarkoituksiin.

```
SELECT email
FROM Asiakas;
```

	email
1	erä.jorma@email.com
2	puupää.pekka@email.com