OmaAutokanta

Työryhmän jäsenet

Teemu Kytölä, TTC2020-3021, TTV22S2

Leevi Kauranen, TTC2020-3021, TTV22S2

Lamin Ndow, TTC2020-3021, TTV22S2

Vaatimusmäärittely

Johdanto

OmaAutokanta on tietokanta, joka on suunniteltu autokorjaamoille helpottamaan heidän päivittäistä toimintaansa. Päätimme tehdä harjoitustyön kolmen hengen ryhmässä. Ryhmäämme kuului Teemu Kytölä, Leevi Kauranen ja Lamin Ndow. Tietokannan tavoitteena on tallentaa tietoja asiakkaista, heidän autoistaan, varaamistaan huolloista ja korjauksista sekä varastossa olevista tuotteista. Tietokanta on suunniteltu autokorjaamoiden henkilökunnalle. Sen avulla he voivat myös seurata ja suunnitella töiden tekemistä.

Aiheen valintaa helpotti se, että ryhmästämme löytyi yksi autoharrastaja. Hänen tietämyksestään oli apua tietokannan järkevän rakenteen suunnittelussa. Palvelua tuottavan liikkeen tietokannan suunnittelu oli myös mielenkiintoista, koska ryhmämme jäsenillä oli kiinnostusta tulevaisuudessa perustaa omaa liiketoimintaa.

Yleiskuvaus

Järjestelmän liittymät ympäristöön ovat asiakkaat, autojen omistajat ja autokorjaamoiden henkilökunta. Käyttäjät ovat autokorjaamoiden työntekijät ja sen asiakkaat. Autokorjaamontyöntekijät tallentavat ja hallinnoivat tietokannan tietoja. Käyttöympäristö on autokorjaamo. Tietokannan yleiset rajoitteet ovat, että sen käyttö edellyttää tietokantaosaamista ja tietokoneen käyttötaitoja. Tietokannan käyttö rajoittuu myös toistaiseksi Suomeen.

Tämä autokanta on toimiva, koska se on suunniteltu huolellisesti ja ottaen huomioon tärkeät tekijät, kuten normalisoinnin, eheysrajoitteet ja taulujen suhteet. Tämä mahdollistaa sen, että tietokanta on tehokas ja helppokäyttöinen.

Tietokannan taulut on normalisoitu välttämään päällekkäistä tietoa ja parantamaan tietokannan tehokkuutta. Eheysrajoitteita on käytetty varmistamaan, että tietokantaan tallennettu tieto on oikeellista ja johdonmukaista. Indeksit parantavat tietojen hakunopeutta ja taulujen väliset suhteet on määritelty selkeästi ja loogisesti, jotta tietojen hakeminen ja päivittäminen olisi helppoa.

Toiminnot

Tietokannalta vaaditaan seuraavia toimintoja:

Asiakkaiden ja heidän autojensa tallentaminen tietokantaan (pakollinen)

Huoltojen varausten tallentaminen tietokantaan (tärkeä)

Korjaustöiden tallentaminen tietokantaan (tärkeä)

Tuotteiden tallentaminen tietokantaan (tärkeä)

Auton ja korjaustyön välisen suhteen tallentaminen tietokantaan (olisi kiva saada)

Työntekijöiden tallentaminen tietokantaan (tärkeä)

Työntekijän ja korjaustyön välisen suhteen tallentaminen tietokantaan (olisi kiva saada)

Ulkoiset liittymät

Järjestelmän liittymät ympäristöön ovat asiakkaat, autojen omistajat ja autokorjaamoiden henkilökunta. Käyttöliittymän tarkempi kuvaus on, että se on tietokantajärjestelmä, joka vaatii käyttäjän syöttämään tietoja manuaalisesti tietokantaan. Nettisivujen luominen tietokantaa varten on tärkeä osa modernia liiketoimintaa. Ammattilaistemme pyynnöstä loimme tehokkaat ja käyttäjäystävälliset nettisivut, jotka hyödyntävät tietokantaamme. Nyt asiakkaamme voivat helposti löytää tarvitsemansa tiedot ja tehdä tilauksia suoraan sivuiltamme. Järjestelmällä ei ole liityntöjä muihin järjestelmiin, oheislaitteisiin tai tietoliikenneyhteyksiin, vielä.

Muut ominaisuudet

Järjestelmän vaadittavat ei-toiminnalliset ominaisuudet ovat:

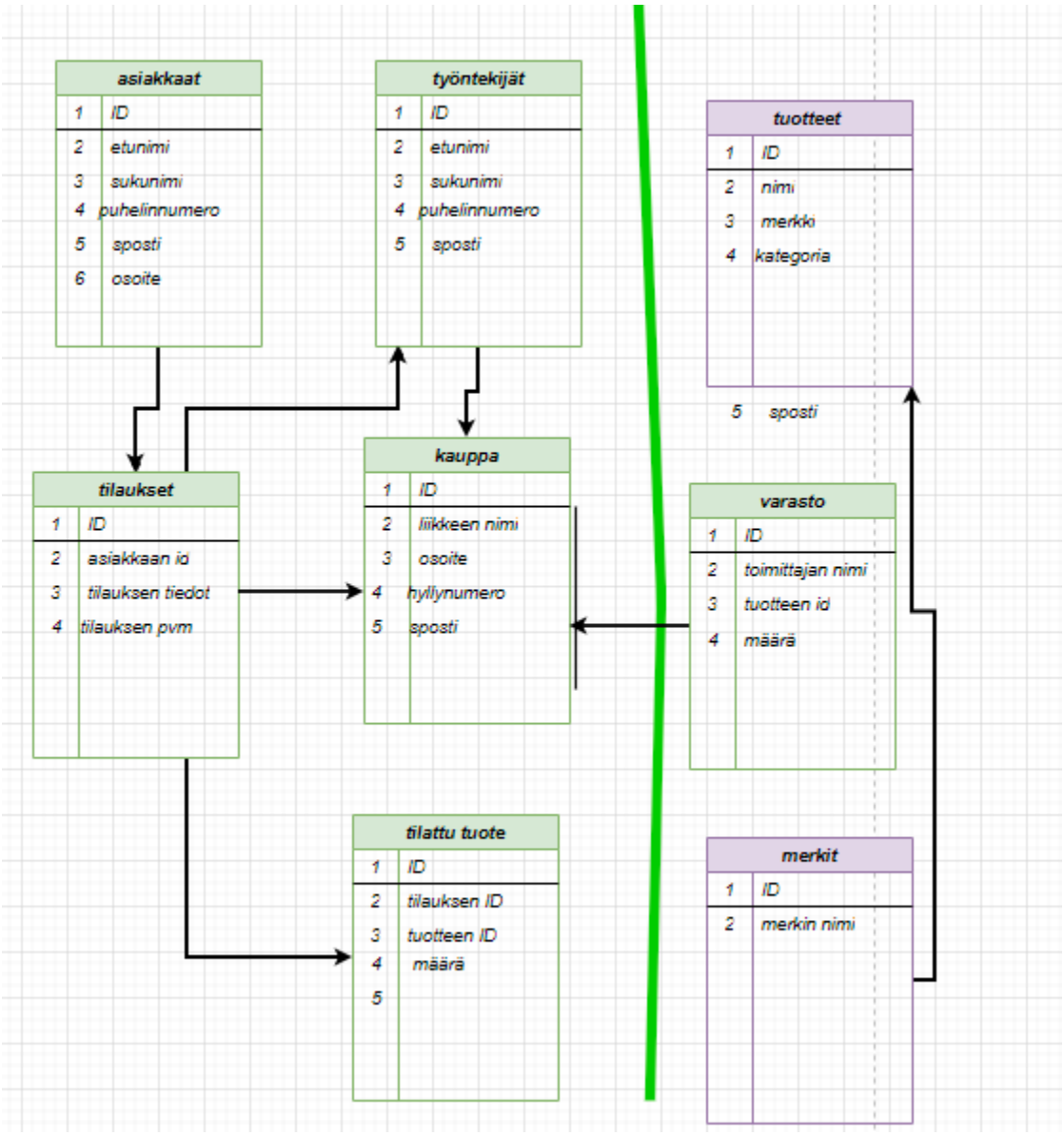
käyttöliittymä verkossa jossa asiakas pystyy varaamaan itselleen huoltoon ajan sekä selata tietokannasta löytyviä tuotteita. Kehitteillä olisi vielä että käyttäjä voi lisätä koriin tuotteet ja tilata ne.

Suorituskyky

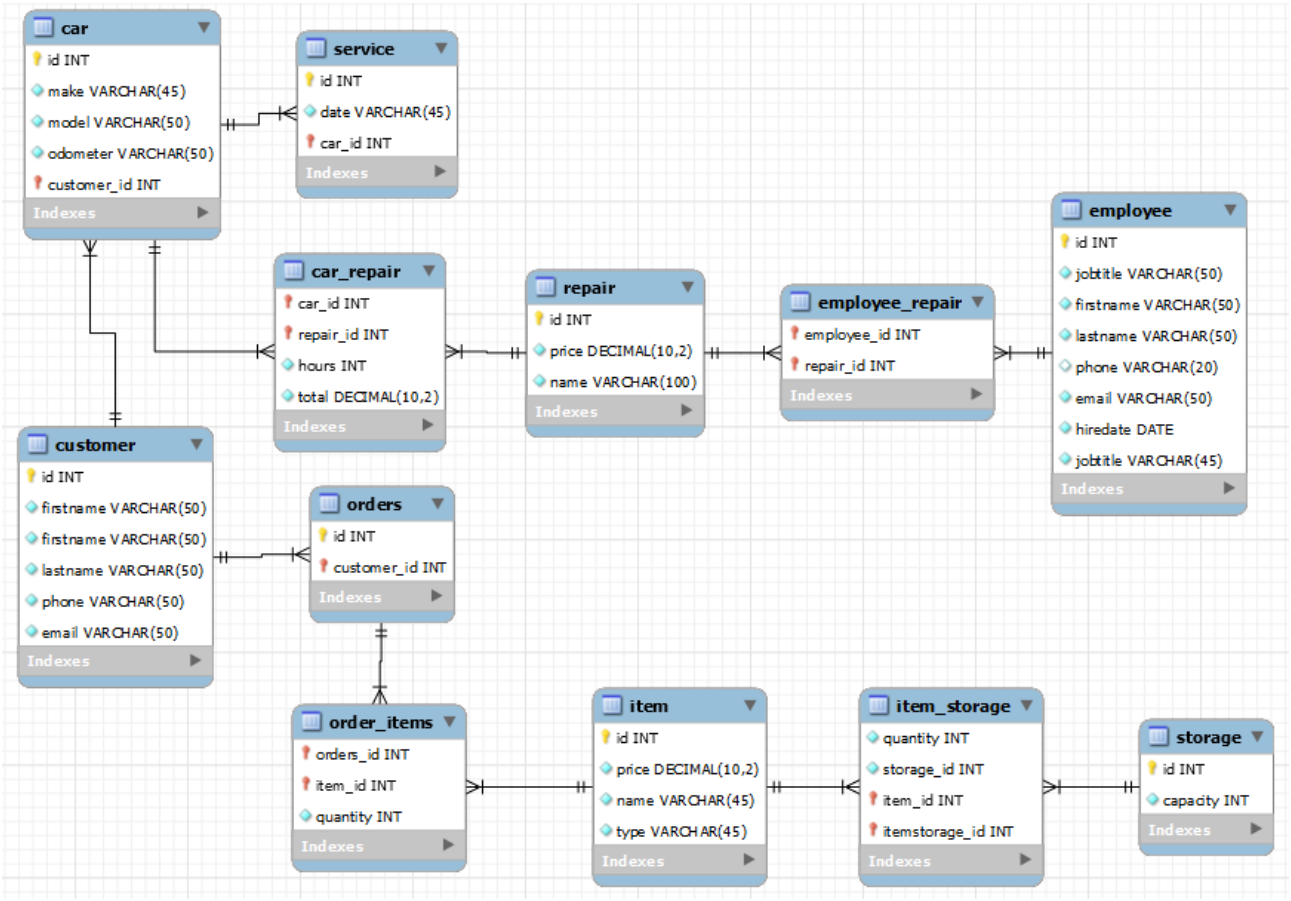
Tietokannan tulee olla nopea ja käytännöllinen.

Käsitemalli

Luonnostelimme tietokannan rakennetta alustavalla käsitemallilla. Piirsimme ennen tätä mallia ensimmäisen vedoksen valkotaululle. Osa tauluista oli vaihdettava tai poistettava kokonaan, kun käsityksemme autokorjaamon tietokantarakenteesta vahvistui. Tietokannan keskiössä ollut kauppa poistettiin ja tietokannan rakenne muuttui enemmän lineaariseksi. Kaupan sijasta loimme varastolle oman taulun. Varsinaiselle korjaamolle / kaupalle ei ollut tarvetta.


asiakkaat
1 ID
2 etunimi
3 sukunimi
4 puhelinnumero
5 sposti
6 osoite
työntekijät
1 ID
2 etunimi
3 sukunimi
4 puhelinnumero
5 sposti
tuotteet
1 ID
2 nimi
3 merkki
4 kategoria
5 sposti
tilaukset
1 ID
2 asiakkaan id
3 tilauksen tiedot
4 tilauksen pvm
kauppa
1 ID
2 liikkeen nimi
3 osoite
4 hyllynumero
5 sposti
varasto
1 ID
2 toimittajan nimi
3 tuotteen id
4 määrä
tilattu tuote
1 ID
2 tilauksen ID
3 tuotteen ID
4 määrä
5
merkit
1 ID
2 merkin nimi

Relaatiokaavio



SQL tietokantakoodi

```
-- Luodaan taulu asiakkaille
CREATE TABLE customer (
  id INT PRIMARY KEY,
  firstname VARCHAR(50) NOT NULL,
  lastname VARCHAR(50) NOT NULL,
  phone VARCHAR(20),
  email VARCHAR(50)
);

-- Luodaan taulu autoille
CREATE TABLE car (
  id INT PRIMARY KEY,
  make VARCHAR(50) NOT NULL,
  model VARCHAR(50) NOT NULL,
  odometer INT NOT NULL,
  customer_id INT NOT NULL,
  FOREIGN KEY (customer_id) REFERENCES customer(id)
);

-- Luodaan taulu huoltojen varaamiselle
CREATE TABLE service (
  id INT PRIMARY KEY,
  date DATE NOT NULL,
```

```
  car_id INT NOT NULL,
  FOREIGN KEY (car_id) REFERENCES car(id)
);

-- Luodaan taulu korjaustöille
CREATE TABLE repair (
  id INT PRIMARY KEY,
  price DECIMAL(10,2) NOT NULL CHECK (price >=0),
  name VARCHAR(100) NOT NULL
);

-- Luodaan taulu autojen ja korjaustöiden väliselle suhteelle
CREATE TABLE car_repair (
  car_id INT NOT NULL,
  repair_id INT NOT NULL,
  hours INT NOT NULL CHECK (hours >=0),
  total DECIMAL(10,2) NOT NULL CHECK (total >=0),
  PRIMARY KEY (car_id, repair_id),
  FOREIGN KEY (car_id) REFERENCES car(id),
  FOREIGN KEY (repair_id) REFERENCES repair(id)
);

-- Luodaan taulu työntekijöille
CREATE TABLE employee (
  id INT PRIMARY KEY,
  firstname VARCHAR(50) NOT NULL,
  lastname VARCHAR(50) NOT NULL,
  phone VARCHAR(20),
  email VARCHAR(50),
  hiredate DATE NOT NULL,
  jobtitle VARCHAR(50) NOT NULL
);

-- Luodaan taulu työntekijöiden ja korjaustöiden väliselle suhteelle
CREATE TABLE employee_repair (
  employee_id INT NOT NULL,
  repair_id INT NOT NULL,
  PRIMARY KEY (employee_id, repair_id),
  FOREIGN KEY (employee_id) REFERENCES employee(id),
  FOREIGN KEY (repair_id) REFERENCES repair(id)
);

-- Luodaan taulu varastossa oleville tuotteille
CREATE TABLE item (
  id INT PRIMARY KEY,
  price DECIMAL(10,2) NOT NULL CHECK (price >=0),
  name VARCHAR(100) NOT NULL,
  type VARCHAR(50) NOT NULL
);
```

```sql
-- Luodaan taulu varaston tilalle
CREATE TABLE itemstorage (
  id INT PRIMARY KEY,
  capacity INT NOT NULL
);

-- Luodaan taulu tuotteiden ja varaston väliselle suhteelle
CREATE TABLE item_storage (
  item_id INT NOT NULL,
  storage_id INT NOT NULL,
  quantity INT NOT NULL CHECK (quantity >=0),
  PRIMARY KEY (item_id, storage_id),
  FOREIGN KEY (item_id) REFERENCES item(id),
  FOREIGN KEY (storage_id) REFERENCES itemstorage(id)
);


CREATE TABLE orders (
  id INT PRIMARY KEY,
  customer_id INT NOT NULL,
  FOREIGN KEY (customer_id) REFERENCES customer(id)
);

CREATE TABLE order_items (
  order_id INT NOT NULL,
  item_id INT NOT NULL,
  quantity INT NOT NULL CHECK (quantity > 0),
  FOREIGN KEY (order_id) REFERENCES orders(id),
  FOREIGN KEY (item_id) REFERENCES item(id)
);
```

Testidatan lisääminen

```
Voidaan lisätä kaksi uutta asiakasta customer
tauluun ja heidän autonsa car tauluun käyttämällä
INSERT INTO lausetta.
Esimerkiksi:

-- Lisätään kaksi uutta asiakasta
INSERT INTO customer (id, firstname, lastname, phone, email)
VALUES
(11, 'Uusi', 'Asiakas1', '0401111111', 'uusi.asiakas1@example.com'),
(12, 'Uusi', 'Asiakas2', '0502222222', 'uusi.asiakas2@example.com');
```

```sql
-- Lisätään kaksi uutta autoa
INSERT INTO car (id, make, model, odometer, customer_id)
VALUES
(11, 'Uusi', 'Auto1', 100000, 11),
(12, 'Uusi', 'Auto2', 200000, 12);



-- Lisätään testidataa asiakkaille
INSERT INTO customer (id, firstname, lastname, phone, email)
VALUES
(1, 'Matti', 'Meikäläinen', '0401234567', 'matti.meikalainen@example.com'),
(2, 'Sanna', 'Virtanen', '0509876543', 'sanna.virtanen@example.com'),
(3, 'Jari', 'Korhonen', '0412345678', 'jari.korhonen@example.com'),
(4, 'Liisa', 'Lahtinen', '0518765432', 'liisa.lahtinen@example.com'),
(5, 'Pekka', 'Mäkinen', '0423456789', 'pekka.makinen@example.com'),
(6, 'Minna', 'Niemi', '0527654321', 'minna.niemi@example.com'),
(7, 'Juha', 'Laine', '0434567890', 'juha.laine@example.com'),
(8, 'Anna', 'Koskinen', '0536543210', 'anna.koskinen@example.com'),
(9, 'Timo', 'Heikkinen', '0445678901', 'timo.heikkinen@example.com'),
(10, 'Maria', 'Lehto', '0545432109', 'maria.lehto@example.com');

-- Lisätään testidataa autoille
INSERT INTO car (id, make, model, odometer, customer_id)
VALUES
(1, 'Toyota', 'Corolla', 150000, 1),
(2, 'Volkswagen', 'Golf', 120000, 2),
(3, 'Ford', 'Focus', 90000, 3),
(4, 'Nissan', 'Micra', 80000, 4),
(5, 'Honda', 'Civic', 110000, 5),
(6, 'Skoda', 'Octavia', 130000, 6),
(7, 'Volvo', 'V40', 140000, 7),
(8, 'Hyundai', 'i20', 70000, 8),
(9, 'Kia', 'Rio', 60000, 9),
(10, 'Renault', 'Clio', 100000, 10);

-- Lisätään testidataa huoltojen varaamiselle
INSERT INTO service (id, date, car_id)
VALUES
(1, '2023-04-20', 1),
(2, '2023-04-21', 2),
(3, '2023-04-22', 3),
(4, '2023-04-23', 4),
(5, '2023-04-24', 5),
(6, '2023-04-25', 6),
(7, '2023-04-26', 7),
(8, '2023-04-27', 8),
(9, '2023-04-28', 9),
(10, '2023-04-29', 10);
```

```sql
-- Lisätään testidataa korjaustöille
INSERT INTO repair (id, price, name)
VALUES
(1, 50.00, 'Öljynvaihto'),
(2, 100.00, 'Jarrupalat'),
(3, 400.00, 'Renkaat'),
(4, 300.00, 'Tuulilasi'),
(5, 200.00, 'Jakohihna'),
(6, 150.00, 'Akku'),
(7, 250.00, 'Ilmastointi'),
(8, 200.00, 'Pakoputki'),
(9, 300.00, 'Kytkin'),
(10, 400.00, 'Iskunvaimentimet');

-- Lisätään testidataa autojen ja korjaustöiden väliselle suhteelle
INSERT INTO car_repair (car_id, repair_id, hours, total)
VALUES
(1 ,1 ,1 ,175.00),
(1 ,2 ,2 ,350.00),
(2 ,3 ,4 ,900.00),
(2 ,4 ,3 ,675.00),
(3 ,5 ,2 ,450.00),
(3 ,6 ,1 ,275.00),
(4 ,7 ,3 ,625.00),
(4 ,8 ,2 ,450.00),
(5 ,9 ,4 ,900.00),
(5 ,10 ,5 ,1100.00),
(6 ,1 ,1 ,175.00),
(6 ,3 ,4 ,900.00),
(7 ,2 ,2 ,350.00),
(7 ,4 ,3 ,675.00),
(8 ,5 ,2 ,450.00),
(8 ,7 ,3 ,625.00),
(9 ,6 ,1 ,275.00),
(9 ,8 ,2 ,450.00),
(10 ,9 ,4 ,900.00),
(10 ,10 ,5 ,1100.00);
```

```sql
-- Lisätään testidataa työntekijöille
INSERT INTO employee (id, firstname, lastname, phone, email, hiredate, jobtitle)
VALUES
(1, 'Eero', 'Aho', '0401111111', 'eero.aho@example.com', '2020-01-01', 'Mekaanikko'),
(2, 'Sari', 'Saari', '0502222222', 'sari.saari@example.com', '2020-02-02', 'Mekaanikko'),
(3, 'Antti', 'Ahonen', '0413333333', 'antti.ahonen@example.com', '2020-03-03', 'Mekaanikko'),
(4, 'Leena', 'Lehtinen', '0514444444', 'leena.lehtinen@example.com', '2020-04-04', 'Mekaanikko'),
(5, 'Markku', 'Mäki', '0425555555', 'markku.maki@example.com', '2020-05-05', 'Mekaanikko'),
(6, 'Hanna', 'Hakala', '0526666666', 'hanna.hakala@example.com', '2020-06-06', 'Myyjä'),
(7, 'Jussi', 'Jokinen', '0437777777', 'jussi.jokinen@example.com', '2020-07-07', 'Myyjä'),
(8, 'Riitta', 'Ranta', '0538888888', 'riitta.ranta@example.com', '2020-08-08', 'Myyjä'),
(9, 'Tomi', 'Tuominen', '0449999999', 'tomi.tuominen@example.com', '2020-09-09', 'Päällikkö'),
(10, 'Laura', 'Laine', '0540000000', 'laura.laine@example.com', '2020-10-10', 'Päällikkö');

-- Lisätään testidataa työntekijöiden ja korjaustöiden väliselle suhteelle
INSERT INTO employee_repair (employee_id, repair_id)
VALUES
(1 ,1),
(1 ,2),
(2 ,3),
(2 ,4),
(3 ,5),
(3 ,6),
(4 ,7),
(4 ,8),
(5 ,9),
(5 ,10),
(6 ,1),
(6 ,3),
(7 ,2),
(7 ,4),
(8 ,5),
(8 ,7),
(9 ,6),
(9 ,8),
(10 ,9),
(10 ,10);



-- Lisätään testidataa varastossa oleville tuotteille
INSERT INTO item (id, price, name, type)
VALUES
(1, 10.00, 'Öljy', 'Neste'),
(2, 15.00, 'Jarruneste', 'Neste'),
(3, 20.00, 'Jäähdytysneste', 'Neste'),
(4, 25.00, 'Jarrupalat', 'Osat'),
(5, 100.00, 'Renkaat', 'Osat'),
```

```
(6, 75.00, 'Tuulilasi', 'Osat'),
(7, 50.00, 'Jakohihna', 'Osat'),
(8, 37.50, 'Akku', 'Osat'),
(9, 62.50, 'Ilmastointi', 'Osat'),
(10, 50.00, 'Pakoputki', 'Osat');

-- Lisätään testidataa varaston tilalle
INSERT INTO itemstorage (id, capacity)
VALUES
(1, 1000);

-- Lisätään testidataa tuotteiden ja varaston väliselle suhteelle
INSERT INTO item_storage (item_id, storage_id, quantity)
VALUES
(1 ,1 ,100),
(2 ,1 ,50),
(3 ,1 ,75),
(4 ,1 ,25),
(5 ,1 ,40),
(6 ,1 ,10),
(7 ,1 ,15),
(8 ,1 ,20),
(9 ,1 ,30),
(10 ,1 ,35);
```

SQL-kyselyt

```
-- Esimerkki 1: Matti Meikäläinen tilasi 5 litraa öljyä
SELECT c.firstname, c.lastname, c.phone, c.email, i.name, i.price, i.type, 5 AS quantity, i.price * 5 AS total
FROM customer c
JOIN item i ON i.id = 1
WHERE c.id = 1;
```

	firstname	lastname	phone	email	name	price	type	quantity	total
▶	Matti	Meikäläinen	0401234567	matti.meikalainen@example.com	Öljy	10.00	Neste	5	50.00

```
-- Esimerkki 2: Sanna Virtanen tilasi 4 uutta rengasta
SELECT c.firstname, c.lastname, c.phone, c.email, i.name, i.price, i.type, 4 AS quantity, i.price * 4 AS total
FROM customer c
JOIN item i ON i.id = 5
WHERE c.id = 2;
```

	firstname	lastname	phone	email	name	price	type	quantity	total
▶	Sanna	Virtanen	0509876543	sanna.virtanen@example.com	Renkaat	100.00	Osat	4	400.00

```
SELECT c.firstname, c.lastname, i.name AS item_name, i.price AS item_price, SUM(i.price) OVER
(PARTITION BY c.id) AS total_price
FROM customer c
JOIN car ca ON c.id = ca.customer_id
JOIN car_repair cr ON ca.id = cr.car_id
JOIN repair r ON cr.repair_id = r.id
```

```
JOIN item i ON r.name = i.name
WHERE c.id = 1 -- vaihda asiakkaan id haluamaksesi
LIMIT 3;
```

	firstname	lastname	item_name	item_price	total_price
▶	Matti	Meikäläinen	Jarrupalat	25.00	25.00

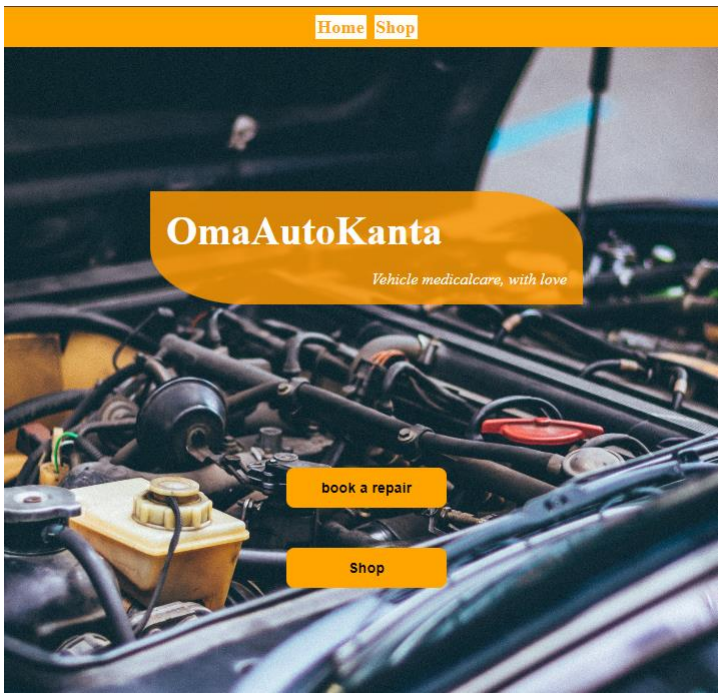
Tässä on esimerkki triggeristä, joka suoritetaan aina kun uusi tilaus lisätään orders-tauluun. Tämä triggeri päivittää item_storage-taulun quantity-saraketta vähentämällä siitä tilauksen tuotteiden määrän:

```
CREATE TRIGGER new_order

AFTER INSERT ON orders

FOR EACH ROW

BEGIN

   UPDATE item_storage

   SET quantity = quantity - (SELECT quantity FROM order_items WHERE order_id = NEW.id AND item_id = item_storage.item_id)

   WHERE storage_id = 1;

END;
```

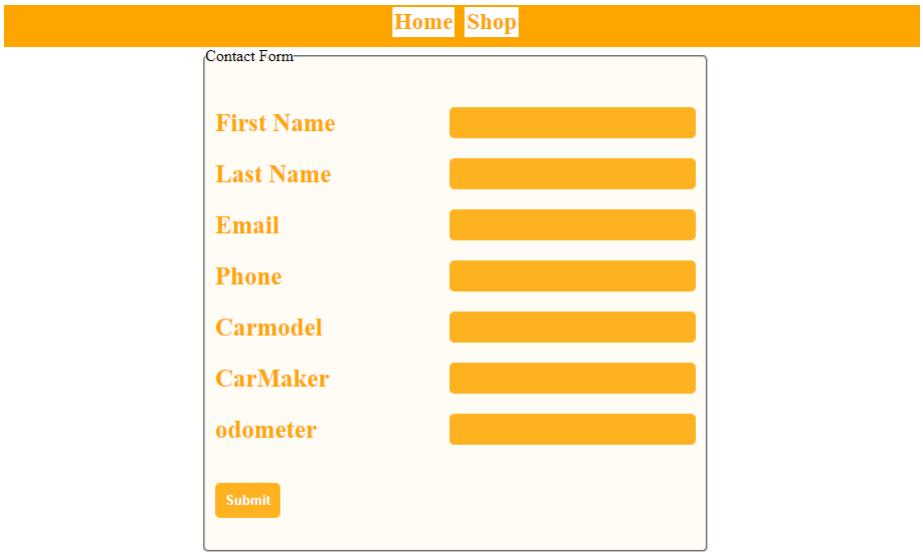
Kuvia Käyttöliittymästä

Toteutimme erittäin pelkistetyn käyttöliittymän, ulkonäköön ei ole nähty vaivaa.

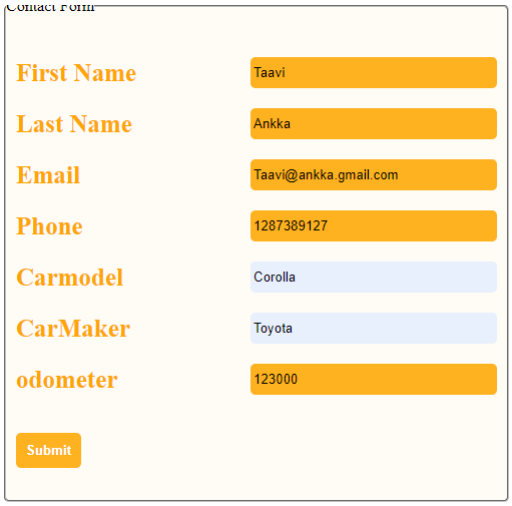
Tässä sivuston etusivu:

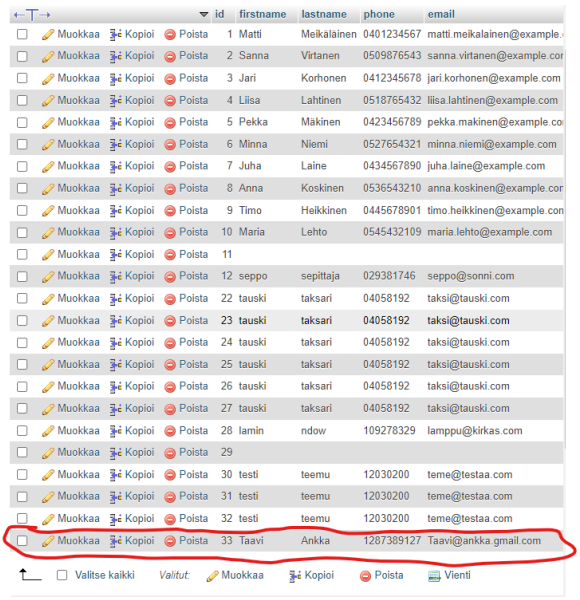


Sivustolla voi varata itselleen ajan huoltoon:



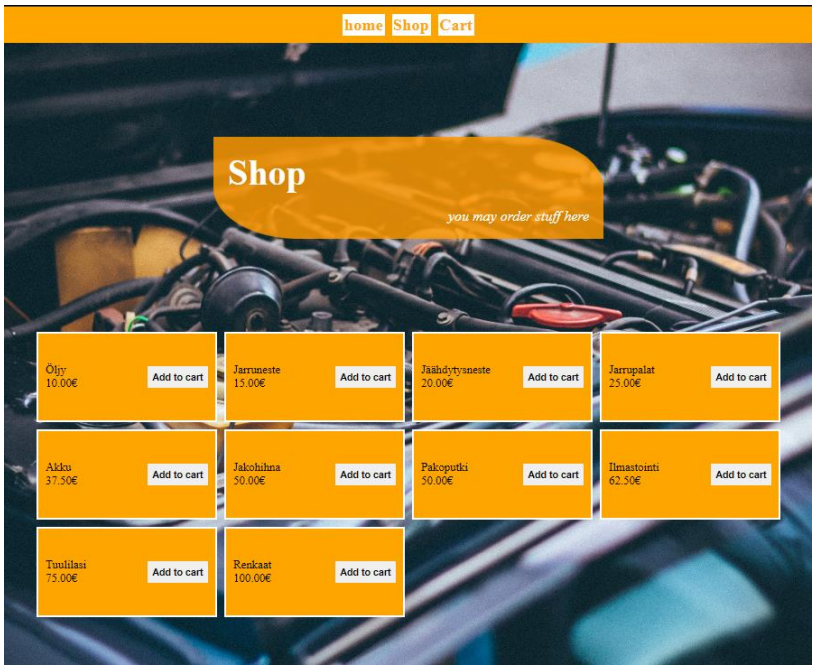
Tähän syötetyt tiedot siirtyvät tietokantaan.





	id	firstname	lastname	phone	email
Muokkaa Kopioi Poista	1	Matti	Meikalainen	0401234567	matti.meikalainen@example.
Muokkaa Kopioi Poista	2	Sanna	Virtanen	0509876543	sanna.virtanen@example.cor
Muokkaa Kopioi Poista	3	Jari	Korhonen	0412345678	jari.korhonen@example.com
Muokkaa Kopioi Poista	4	Liisa	Lahtinen	0518765432	liisa.lahtinen@example.com
Muokkaa Kopioi Poista	5	Pekka	Mäkinen	0423456789	pekka.makinen@example.co
Muokkaa Kopioi Poista	6	Minna	Niemi	0527654321	minna.niemi@example.com
Muokkaa Kopioi Poista	7	Juha	Laine	0434567890	juha.laine@example.com
Muokkaa Kopioi Poista	8	Anna	Koskinen	0536543210	anna.koskinen@example.cor
Muokkaa Kopioi Poista	9	Timo	Heikkinen	0445678901	timo.heikkinen@example.con
Muokkaa Kopioi Poista	10	Maria	Lehto	0545432109	maria.lehto@example.com
Muokkaa Kopioi Poista	11				
Muokkaa Kopioi Poista	12	seppo	sepittaja	029381746	seppo@sonni.com
Muokkaa Kopioi Poista	22	tauski	taksari	04058192	taksi@tauski.com
Muokkaa Kopioi Poista	23	tauski	taksari	04058192	taksi@tauski.com
Muokkaa Kopioi Poista	24	tauski	taksari	04058192	taksi@tauski.com
Muokkaa Kopioi Poista	25	tauski	taksari	04058192	taksi@tauski.com
Muokkaa Kopioi Poista	26	tauski	taksari	04058192	taksi@tauski.com
Muokkaa Kopioi Poista	27	tauski	taksari	04058192	taksi@tauski.com
Muokkaa Kopioi Poista	28	lamin	ndow	109278329	lamppu@kirkas.com
Muokkaa Kopioi Poista	29				
Muokkaa Kopioi Poista	30	testi	teemu	12030200	teme@testaa.com
Muokkaa Kopioi Poista	31	testi	teemu	12030200	teme@testaa.com
Muokkaa Kopioi Poista	32	testi	teemu	12030200	teme@testaa.com
Muokkaa Kopioi Poista	33	Taavi	Ankka	1287389127	Taavi@ankka.gmail.com

Valitse kaikki Valitut: Muokkaa Kopioi Poista Vienti



Shop sivu hakee tuotteet tietokannasta ja tulostaa ne divin kera.

	id	price	name	type
Muokkaa Kopioi Poista	1	10.00	Öljy	Neste
Muokkaa Kopioi Poista	2	15.00	Jarruneste	Neste
Muokkaa Kopioi Poista	3	20.00	Jäähdytysneste	Neste
Muokkaa Kopioi Poista	4	25.00	Jarrupalat	Osat
Muokkaa Kopioi Poista	5	100.00	Renkaat	Osat
Muokkaa Kopioi Poista	6	75.00	Tuulilasi	Osat
Muokkaa Kopioi Poista	7	50.00	Jakohihna	Osat
Muokkaa Kopioi Poista	8	37.50	Akku	Osat
Muokkaa Kopioi Poista	9	62.50	Ilmastointi	Osat
Muokkaa Kopioi Poista	10	50.00	Pakoputki	Osat

Niin kuin näkee, tässä tietokannan tuotteet

Käyttäjä voi lisätä tuotteita ostoskoriin ja ostoskorin pääsee näkemään sisältöineen cart osiossa.

Shopping Cart

Item Name	Price
Jarrupalat	25.00
Jarruneste	15.00
Öljy	10.00
Akku	37.50
Jakohihna	50.00
Jakohihna	50.00
Jakohihna	50.00
Jäähdytysneste	20.00
Jarruneste	15.00
Akku	37.50
Total:	310

Continue Shopping

sivuston koodit löytyvät gitlabista alla olevasta urlista:

https://github.com/Karvakaula/My_Projects/tree/main/Databases%20Course/harjoitusty%C3%B6/codes

Perusteluja ratkaisuista

Tarkoituksena oli luoda käytännöllinen tietokanta pienen ja paikallisen autokorjaamon käyttöön.

customer-taulu sisältää asiakkaiden tiedot.

car-taulu sisältää autojen tiedot. customer_id-sarakkeessa onviittaus customer-taulun id-sarakkeeseen, mikä mahdollistaa sen, että voit hakea tietoa siitä, minkä asiakkaan auto on kyseessä.

service-taulu sisältää huoltojen varaustiedot. car_id-sarakkeessa on viittaus car-taulun id-sarakkeeseen, mikä mahdollistaa sen, että voit hakea tietoa siitä, minkä auton huolto on kyseessä.

repair-taulu sisältää korjaustöiden tiedot.

car_repair-taulu sisältää autojen ja korjaustöiden välisen suhteen tiedot. car_id-sarakkeessa on viittaus car-taulun id-sarakkeeseen ja repair_id-sarakkeessa viittaus repair-taulun id-sarakkeeseen. Tämä mahdollistaa sen, että voit hakea tietoa siitä, mitkä korjaukset on tehty tietylle autolle.

employee-taulu sisältää työntekijöiden tiedot.

employee_repair-taulu sisältää työntekijöiden ja korjaustöiden välisen suhteen tiedot. employee_id-sarakkeessa on viittaus employee-taulun id-sarakkeeseen ja repair_id-sarakkeessa viittaus repair-taulun id-sarakkeeseen. Tämä mahdollistaa sen, että voit hakea tietoa siitä, mitkä työntekijät ovat tehneet tietyt korjaukset.

item-taulu sisältää varastossa olevien tuotteiden tiedot.

itemstorage-taulu sisältää varaston tilan tiedot.

item_storage-taulu sisältää tuotteiden ja varaston välisen suhteen tiedot. item_id-sarakkeessa on viittaus **item**-taulun id-sarakkeeseen ja storage_id-sarakkeessa viittaus itemstorage

orders-taulu sisältää tilausten tiedot. customer_id-sarakkeessa on viittaus customer-taulun id-sarakkeeseen, mikä mahdollistaa sen, että voit hakea tietoa siitä, minkä asiakkaan tilaus on kyseessä.

---

order_items-taulu sisältää tilausten ja tuotteiden välisen suhteen tiedot. order_id-sarakkeessa on viittaus orders-taulun id-sarakkeeseen ja item_id-sarakkeessa viittaus item-taulun id-sarakkeeseen. Tämä mahdollistaa sen, että voit hakea tietoa siitä, mitkä tuotteet kuuluvat tiettyyn tilaukseen.

Nämä taulut ovat hyödyllisiä tietokannassa, koska ne mahdollistavat tietojen tallentamisen ja hakemisen järjestelmällisesti ja tehokkaasti. item-taulu sisältää varastossa olevien tuotteiden tiedot, itemstorage-taulu sisältää varaston tilan tiedot ja item_storage-taulu sisältää tuotteiden ja varaston välisen suhteen tiedot. Tämä mahdollistaa sen, että voit hakea tietoa siitä, mitä tuotteita on varastossa ja kuinka paljon niitä on.

Olisi mahdollista tehdä tilaukset ilman **order_items**-taulua, mutta se rajoittaisi tietokantamme joustavuutta. Ilman **order_items**-taulua meidän tulisi esittää jokainen tilauksen tuote erillisenä rivinä **orders**-taulussa. Tämä vois toimii, jos jokaisessa tilauksessa on aina sama määrä tuotteita, mutta vaihtelevaa määrää tuotteita sisältävien tilausten esittäminen olisi vaikeaa.

Order_items-taulun käyttö mahdollistaa tilausten esittämisen minkä tahansa määrän tuotteilla ja helppottaa tuotteiden lisäämistä tai poistamista tilauksesta. Se myös helpottaa tietokannan kyselyjä tilauksista ja niiden tuotteista. Yleisesti ottaen **order_items**-taulun käyttö on joustavampi ja skaalautuvampi tapa esittää useita tuotteita sisältäviä tilauksia.

Itsearviot

Teemu Kytölä

Osallistuin harjoitustyöhön yhteistyössä muiden jäsenten kanssa. Olin mukana suunnittelemassa tietokantaa ja piirtämässä sen rakennetta. Lisäksi työskentelin itsenäisesti dokumentaation ja. Keskustelimme harjoitustyön suunnittelusta myös Discord-kanavalla ja kävimme läpi erilaisia ideoita ja ratkaisuja. Osallistuin myös ER-diagrammin suunnitteluun ja piirtämiseen sekä testidatan luomiseen. Osa testidatasta luotiin tekoälyllä ja osa käsin yhdessä muiden jäsenten kanssa. Työskentelymme oli tehokasta ja tuloksellista. Antaisin osuudelleni arvosanan 5.

Aikaa kulutettu

noin 5h suunnitteluun,

normalisointi ja eheysrajoitteiden suunnittelu 3h

Noin 7h taulujen ja yhteyksien suunnitteluun(foreign ja primary keyt

eer kaavion toteutus 3h

Tietokannan toiminnallisuuksien suunnittelu 2h

Tietokannan luonti ja testausdatan suunnittelu 5h

Käyttöliittymän toteutus 1h

Testausdatan syöttö 2h

Tietokannan testaus 4h

Dokumentointiin noin 1,5h

Itsearvointi 0,5h

Leevi Kauranen

Toteutimme harjoitustyön enimmäkseen työskentelemällä joko discordin välityksellä tai paikan päällä porukassa. Suunnittelu toteutettiin suurimmilta osin yhdessä. Itsenäisesti työskentelin käyttöliittymän toteutuksessa ja php-kielen opettelussa. Jokainen osallistui työhön tasavertaisesti ja lähetimme omia

ehdotuksiamme discordiin ja yhdessä otimme niitä mukaan toteutukseen. Työskentely oli tehokasta ja yhteistyö pelasi moitteettomasti. Antaisin itselleni arvosanaksi työstä 5.

noin 4h suunnitteluun,

normalisointi ja eheysrajoitteiden suunnittelu 3h

Noin 6h taulujen ja yhteyksien suunnitteluun(foreign ja primary keyt

eer kaavion toteutus 1h

Tietokannan toiminnallisuuksien suunnittelu 1h

Tietokannan luonti ja testausdatan suunnittelu 4h

Käyttöliittymän toteutus 6h

Testausdatan syöttö 2h

Tietokannan testaus 3h

Dokumentointiin noin 3,5h

Itsearvointi 0,5h

Lamin Ndow

Osani harjoitustyöhön tein pääosin yhteistyössä vuorovaikutteisesti. Käytännössä molempien jäsenien kanssa porukassa tai kahdestaan. Olin mukana tietokannan suunnittelussa ja luonnostelussa. Piirsin ja jäsentelin tietokantaa. Testailimme MySQL workbench sovelluksessa tietokannan rakennetta ja loogisuutta. Itsenäisemmin työskentelin dokumentaation luonnostelussa ja vaatimusmäärittelyn toteutuksessa. Harjoitustyön suunnittelua kävimme myös tapaamisien välillä yhteisellä Discord-kanavalla.

Harjoitustyölle osaltani antaisin arvosanan 5. Saimme tehtyä käytäntöön perustuvan tietokannan pienen autokannan käyttöön.

noin 5h suunnitteluun,

normalisointi ja eheysrajoitteiden suunnittelu 5h

Noin 8h taulujen ja yhteyksien suunnitteluun (foreign ja primary keyt)

eer kaavion toteutus 0.5h

Tietokannan toiminnallisuuksien suunnittelu 1h

Tietokannan luonti ja testausdatan suunnittelu 4h

Käyttöliittymän toteutus 0,5h

Testausdatan syöttö 1h

Tietokannan testaus 2h

Dokumentointiin noin 5h

Itsearvointi 0,5h