Harjoitustyö – Kauneudenhoito verkkosivun tuote-/asiakastietokanta

Esipuhe

Harjoitustyön tarkoituksena on todentaa oma tämänhetkinen osaaminen ja kerrata Kevään 2023 Tietokannat TTC2020-3032 kurssin aikana opitut asiat. Ennustan oman ajankäytön olevan rajallinen, koska osa-aikainen työ ja muut kurssitehtävien vaikuttavan tämän harjoitustyön kohdalla panostukseen ajallisesti, mutta haluan esittää tällä työllä mahdollisimman lyhyesti ja ytimekkäästi oma osaaminen. Tulen kirjoittamaan omia ajatuksia tämän dokumentaation loppupuolelle, kuinka onnistuin tässä harjoitustyössä ja mahdollinen oma arvio harjoitustyöstä.

Pohdin suunnitelmaa tehdessäni missä kaikissa asioissa tietokantoja hyödynnetään. Todellisuudessa kaikki tieto tallennetaan tavalla tai toisella jonkinlaiseen tietokantaan, jotta esimerkiksi meidän arkielämämme palvelut toimisivat. (Pankki, Bussi-/Juna-aikataulut, ajanvarausjärjestelmä, puhelinluettelo kännykässä)

Suunnitelma

Mikä on harjoitustyösi aihe?

Harjoitustyöni aiheena Kauneudenhoito-alan verkkosivun tuote-/asiakastietokanta. Tietokannassa olisi tietoja tavallisten tilaajien lisäksi club-asiakkaita, jotka saavat tilatessaan alennusta tuotteista. Toimin itse harjoitustyön toteuttajana. (Anni Orilähde, AC8081)

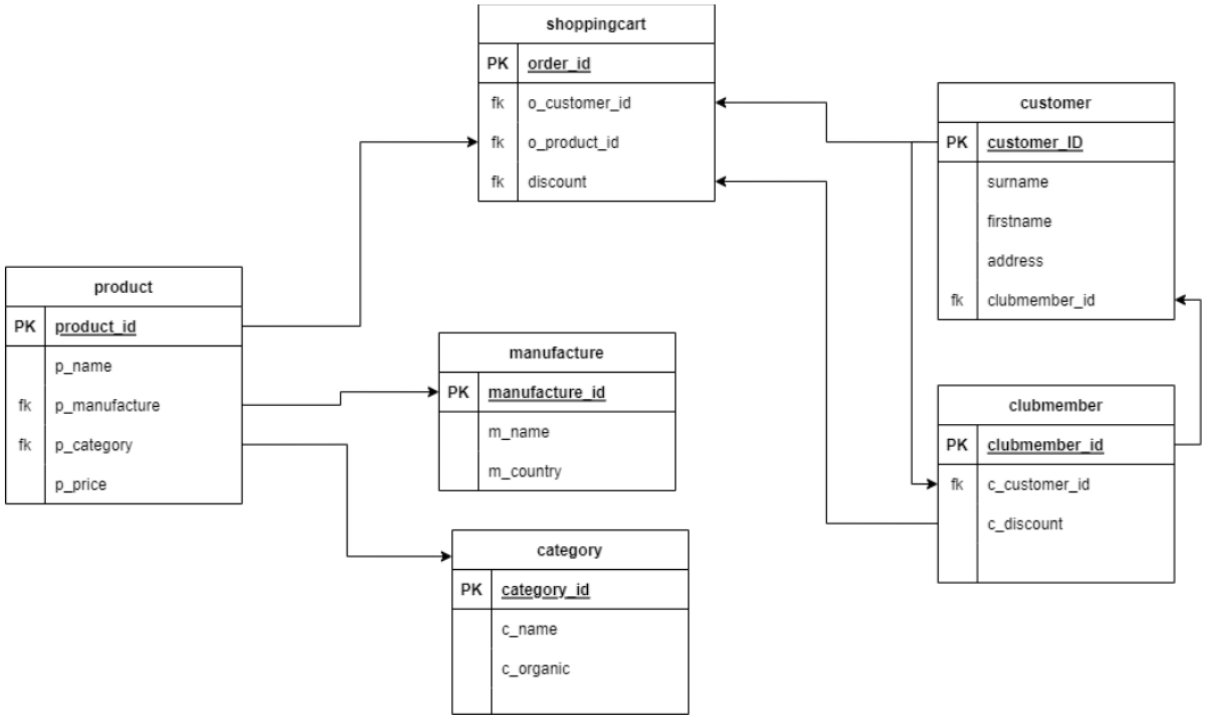
Millainen voisi olla ytimekäs muutaman virkkeen toimeksianto harjoitustyöllesi?

”Verkkokauppamme kaipaa pikaisesti päivitystä ajan tasalla olevaan asiakastietokantaan, jolla voisimme hallita asiakkaittemme tilauksia ja etuuksia.”

Mitkä ovat kaikista keskeisimmät asiat, joista tietoa tallennetaan tietokantaan?

Asiakkaiden tiedot, tuotteiden tiedot, joissa erillisellä taululla yhdistettynä hinta ja toisessa taulussa valmistaja, jolla voidaan kategorisoida tuotteita. Ja lopuksi tietysti asiakkaiden tilaamat tuotteet.

Millaiset voisivat olla ihan keskeisimmät käsitteet/taulut, joita harjoitustyössäsi tarvitaan? Voit piirtää halutessasi kaavion ihan kynälllä paperille (palautukseen kuva), draw.io:lla jne.



- 1 Asiakas voi TILATA MONTA tuotetta

- 1 tuotteen voi tilata MONTA asiakasta (ei tietenkään samaa)

- Asiakas SAATTAA kuulua clubiin

- Tuotteella on VAIN 1 valmistaja

- Tuotteella VOI OLLA USEAMPI kategoria

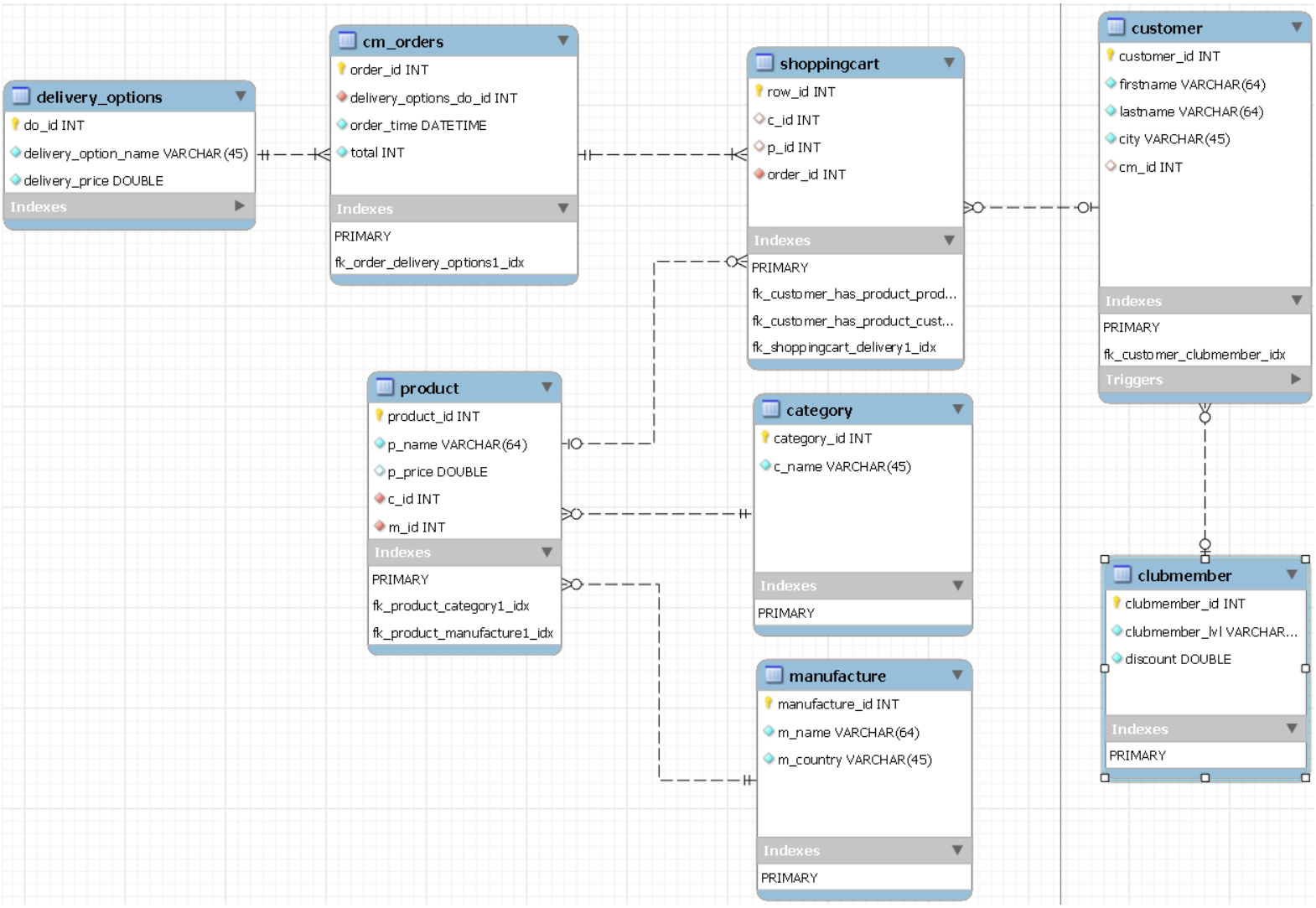
- MONTA tuotetta voi kuulua TILAUKSEEN

- Alennus tulee, JOS asiakas KUULUU clubiin

Tärkeitä avainsanoja:

- Asiakas
- Club-jäsenyys
- Tuote
- Hinta
- Tilaus

EER-Kaavio



Puretaan käsitteitä:

Tietokantaan loin MySQL Workbenchillä EER-suunnittelutyökalulla tietokannan rakenteen ja yhteydet.

”customer” -taulu:

- Yhteys ”clubmember”iin; asiakas voi kuulua (1 kpl), mutta ei ole pakko clubjäseneksi
- Yhteys ”shoppingcart”iin; asiakkaalla voi olla useampia ei yhtään tilausta ”ostoskorissa”

”clubmember” -taulu:

- Yhteys ”customer”iin; clubjäseniä voi olla useampia tai ei yhtään asiakasta

”shoppingcart” -taulu:

- Yhteys ”customer”iin: asiakkaalla voi olla vain 1 kpl uniikki ostotapahtuma (toki seuraavalla kerralla on taas uusi tyhjä ostoskori)
- Yhteys ”delivery”yn: deliveryssa voi olla useita ostoskoreja (ostotapahtumia) ja ostoskorilla on pakko olla delivery eli toimitus (ei saa olla tyhjä)
- Yhteys ”product”iin: ostoskorissa voi olla useita tuotteita, mutta 1 tuotteella on vai 1 ostoskori (toki on huomioitu, jos tilataan useampi kpl, mutta jokainen on uniikki)

”product” -taulu:

- Yhteys ”shoppingcart”iin: kts. viimeinen rivi ”shoppingcart”ista
- Yhteys ”category”yn: Tuotteella täytyy olla jokin kategoria (1 kpl)
- Yhteys ”manufature”en: Tuotteella täytyy olla valmistaja (1 kpl)

”category”-taulu:

- kategorialla voi olla useampi tai ei yhtään tuotetta

”manufacture” -taulu:

- valmistajalla voi olla useampi tai ei yhtään tuotetta.

Taulujen välillä täytyy huomioida yhteydellisyyden suhde, koska moni-moneen yhteyksiä ei sallita Relaatiotietokannoissa. Kuten tiedämme, MySQL Workbench luo automaattisesti moni-moneen taulujen välille taulun rikkomaan tämän yhteyden ja suhde säilyy 1-n.

Tietokannan luonti ja tiedon lisääminen tietokantaan

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_F
OR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema verkkokauppa
-- -----------------------------------------------------
-- Harjoitustyö Tietokanta peruskurssille Jyväskylän Ammattiopisto, Kevät 2023
DROP SCHEMA IF EXISTS `verkkokauppa` ;

-- -----------------------------------------------------
-- Schema verkkokauppa
--
-- Harjoitustyö Tietokanta peruskurssille Jyväskylän Ammattiopisto, Kevät 2023
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `verkkokauppa` ;
USE `verkkokauppa` ;

-- -----------------------------------------------------
-- Table `verkkokauppa`.`clubmember`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`clubmember` (
  `clubmember_id` INT NOT NULL AUTO_INCREMENT,
  `clubmember_lvl` VARCHAR(45) NOT NULL,
  `discount` DOUBLE NOT NULL,
  PRIMARY KEY (`clubmember_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `verkkokauppa`.`customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`customer` (
  `customer_id` INT NOT NULL AUTO_INCREMENT,
  `firstname` VARCHAR(64) NOT NULL,
  `lastname` VARCHAR(64) NOT NULL,
  `city` VARCHAR(45) NOT NULL,
  `cm_id` INT NULL,
  PRIMARY KEY (`customer_id`),
  INDEX `fk_customer_clubmember_idx` (`cm_id` ASC),
  CONSTRAINT `fk_customer_clubmember`
    FOREIGN KEY (`cm_id`)
    REFERENCES `verkkokauppa`.`clubmember` (`clubmember_id`)
    ON DELETE NO ACTION
    ON UPDATE SET NULL)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `verkkokauppa`.`category`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`category` (
  `category_id` INT NOT NULL AUTO_INCREMENT,
  `c_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`category_id`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `verkkokauppa`.`manufacture`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`manufacture` (
  `manufacture_id` INT NOT NULL AUTO_INCREMENT,
  `m_name` VARCHAR(64) NOT NULL,
  `m_country` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`manufacture_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `verkkokauppa`.`product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`product` (
  `product_id` INT NOT NULL AUTO_INCREMENT,
  `p_name` VARCHAR(64) NOT NULL,
  `p_price` DOUBLE NULL,
  `c_id` INT NOT NULL,
  `m_id` INT NOT NULL,
  PRIMARY KEY (`product_id`),
  INDEX `fk_product_category1_idx` (`c_id` ASC),
  INDEX `fk_product_manufacture1_idx` (`m_id` ASC),
  CONSTRAINT `fk_product_category1`
    FOREIGN KEY (`c_id`)
    REFERENCES `verkkokauppa`.`category` (`category_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_product_manufacture1`
    FOREIGN KEY (`m_id`)
    REFERENCES `verkkokauppa`.`manufacture` (`manufacture_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `verkkokauppa`.`delivery_options`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`delivery_options` (
  `do_id` INT NOT NULL AUTO_INCREMENT,
  `delivery_option_name` VARCHAR(45) NOT NULL,
  `delivery_price` DOUBLE NOT NULL,
  PRIMARY KEY (`do_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `verkkokauppa`.`cm_orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`cm_orders` (
  `order_id` INT NOT NULL AUTO_INCREMENT,
  `delivery_options_do_id` INT NOT NULL,
  `order_time` DATETIME NOT NULL,
  `total` INT NOT NULL,
  PRIMARY KEY (`order_id`),
  INDEX `fk_order_delivery_options1_idx` (`delivery_options_do_id` ASC),
  CONSTRAINT `fk_order_delivery_options1`
    FOREIGN KEY (`delivery_options_do_id`)
    REFERENCES `verkkokauppa`.`delivery_options` (`do_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
```

```
-- Table `verkkokauppa`.`shoppingcart`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `verkkokauppa`.`shoppingcart` (
  `row_id` INT NOT NULL AUTO_INCREMENT,
  `c_id` INT NULL,
  `p_id` INT NULL,
  `order_id` INT NOT NULL,
  PRIMARY KEY (`row_id`),
  INDEX `fk_customer_has_product_product1_idx` (`p_id` ASC),
  INDEX `fk_customer_has_product_customer1_idx` (`c_id` ASC),
  INDEX `fk_shoppingcart_delivery1_idx` (`order_id` ASC),
  CONSTRAINT `fk_customer_has_product_customer1`
    FOREIGN KEY (`c_id`)
    REFERENCES `verkkokauppa`.`customer` (`customer_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_customer_has_product_product1`
    FOREIGN KEY (`p_id`)
    REFERENCES `verkkokauppa`.`product` (`product_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_shoppingcart_delivery1`
    FOREIGN KEY (`order_id`)
    REFERENCES `verkkokauppa`.`cm_orders` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

USE `verkkokauppa`;

DELIMITER $$
USE `verkkokauppa`$$
CREATE DEFINER = CURRENT_USER TRIGGER `verkkokauppa`.`customer_BEFORE_DELETE`
BEFORE DELETE ON `customer` FOR EACH ROW
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Action is not allowed, deleting
prevented.";$$

USE `verkkokauppa`$$
CREATE DEFINER = CURRENT_USER TRIGGER `verkkokauppa`.`customer_BEFORE_INSERT`
BEFORE INSERT ON `customer` FOR EACH ROW
BEGIN
            IF NEW.lastname IS NULL OR NEW.firstname IS NULL
    THEN
    DELETE FROM customer WHERE NEW.lastname IS NULL OR NEW.firstname IS NULL;
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "The Information is required";
    END IF;
END$$


DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`clubmember`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`clubmember` (`clubmember_id`, `clubmember_lvl`,
`discount`) VALUES (1, 'Gold', 0.45);
INSERT INTO `verkkokauppa`.`clubmember` (`clubmember_id`, `clubmember_lvl`,
`discount`) VALUES (2, 'Silver', 0.35);
INSERT INTO `verkkokauppa`.`clubmember` (`clubmember_id`, `clubmember_lvl`,
`discount`) VALUES (3, 'Copper', 0.25);
```

```
INSERT INTO `verkkokauppa`.`clubmember` (`clubmember_id`, `clubmember_lvl`,
`discount`) VALUES (4, 'Sprout', 0.10);
INSERT INTO `verkkokauppa`.`clubmember` (`clubmember_id`, `clubmember_lvl`,
`discount`) VALUES (5, 'Not member', 0);

COMMIT;


-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`customer`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (1, 'Aila', 'Mäkinen', 'Jyväskylä', 5);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (2, 'Helmi', 'Haapamäki', 'Hämeenlinna', 2);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (3, 'Kaija', 'Väisänen', 'Helsinki', 5);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (4, 'Kaarina', 'Jokinen', 'Tampere', 3);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (5, 'Aino', 'Virtanen', 'Helsinki', 1);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (6, 'Soile', 'Auvila', 'Savonlinna', 4);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (7, 'Katri', 'Ojala', 'Tampere', 2);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (8, 'Kirsi', 'Koivisto', 'Hämeenlinna', 1);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (9, 'Tanja', 'Ikonen', 'Jyväskylä', 3);
INSERT INTO `verkkokauppa`.`customer` (`customer_id`, `firstname`, `lastname`,
`city`, `cm_id`) VALUES (10, 'Eeva', 'Svard', 'Jyväskylä', 2);

COMMIT;


-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`category`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`category` (`category_id`, `c_name`) VALUES (1, 'hair');
INSERT INTO `verkkokauppa`.`category` (`category_id`, `c_name`) VALUES (2, 'face');
INSERT INTO `verkkokauppa`.`category` (`category_id`, `c_name`) VALUES (3, 'body');

COMMIT;


-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`manufacture`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`manufacture` (`manufacture_id`, `m_name`, `m_country`)
VALUES (1, 'Lumene', 'Finland');
INSERT INTO `verkkokauppa`.`manufacture` (`manufacture_id`, `m_name`, `m_country`)
VALUES (2, 'XZ', 'Finland');
INSERT INTO `verkkokauppa`.`manufacture` (`manufacture_id`, `m_name`, `m_country`)
VALUES (3, 'LancÔme', 'France');
INSERT INTO `verkkokauppa`.`manufacture` (`manufacture_id`, `m_name`, `m_country`)
VALUES (4, 'Urtekram', 'Denmark');

COMMIT;
```

```
-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`product`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (1, 'Curly Mascara', 17.90, 2, 1);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (2, 'Tasapainottava Savinaamio', 23.90, 2, 1);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (3, 'Luminous Moisture Huulipuna', 13.50, 2, 1);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (4, 'Aito Omena Shampoo 250ml', 3.90, 1, 2);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (5, 'Kaura Hiusöljy 100ml ', 5.90, 1, 2);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (6, 'Teint Miracle Meikkivoide', 52.00, 2, 3);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (7, '24H Drama Liqui-Pencil Rajauskynä', 24.00, 2, 3);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (8, 'Luomu Soft Wild Rose Pink Salt Vartalokuorinta 150ml', 11.90, 3,
4);
INSERT INTO `verkkokauppa`.`product` (`product_id`, `p_name`, `p_price`, `c_id`,
`m_id`) VALUES (9, ' Luomu Wild Lemongrass Deodorantti 50ml', 5.50, 3, 4);

COMMIT;


-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`delivery_options`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`delivery_options` (`do_id`, `delivery_option_name`,
`delivery_price`) VALUES (1, 'posti', 4.90);
INSERT INTO `verkkokauppa`.`delivery_options` (`do_id`, `delivery_option_name`,
`delivery_price`) VALUES (2, 'matkahuolto', 2.90);
INSERT INTO `verkkokauppa`.`delivery_options` (`do_id`, `delivery_option_name`,
`delivery_price`) VALUES (3, 'ups', 5.80);
INSERT INTO `verkkokauppa`.`delivery_options` (`do_id`, `delivery_option_name`,
`delivery_price`) VALUES (4, 'nouto', 0);

COMMIT;


-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`cm_orders`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10011, 2, '2021-01-02', DEFAULT);
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10012, 3, '2021-02-28', DEFAULT);
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10013, 1, '2021-04-06', DEFAULT);
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10014, 1, '2022-05-11', DEFAULT);
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10015, 4, '2022-07-08', DEFAULT);
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10016, 2, '2022-06-21', DEFAULT);
INSERT INTO `verkkokauppa`.`cm_orders` (`order_id`, `delivery_options_do_id`,
`order_time`, `total`) VALUES (10017, 3, '2022-09-26', DEFAULT);

COMMIT;
```

```
-- -----------------------------------------------------
-- Data for table `verkkokauppa`.`shoppingcart`
-- -----------------------------------------------------
START TRANSACTION;
USE `verkkokauppa`;
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (1, 2, 3, 10012);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (2, 2, 4, 10012);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (3, 2, 1, 10012);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (4, 4, 8, 10013);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (5, 4, 7, 10013);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (6, 4, 7, 10013);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (7, 4, 5, 10013);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (8, 1, 8, 10011);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (9, 1, 4, 10011);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (10, 1, 3, 10011);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (11, 6, 2, 10015);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (12, 6, 9, 10015);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (13, 6, 8, 10015);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (14, 6, 3, 10015);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (15, 6, 3, 10015);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (16, 8, 1, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (17, 8, 2, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (18, 8, 3, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (19, 8, 4, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (20, 8, 5, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (21, 8, 6, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (22, 8, 7, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (23, 8, 8, 10014);
INSERT INTO `verkkokauppa`.`shoppingcart` (`row_id`, `c_id`, `p_id`, `order_id`)
VALUES (24, 8, 9, 10014);

COMMIT;
```

Taulut luotuna

clubmember

clubmember_id	clubmember_lvl	discount
1	Gold	0.45
2	Silver	0.35
3	Copper	0.25
4	Sprout	0.1
5	Not member	0
NULL	NULL	NULL

customer

customer_id	firstname	lastname	city	cm_id
1	Aila	Mäkinen	Jyväskylä	5
2	Helmi	Haapamäki	Hämeenlinna	2
3	Kaija	Väisänen	Helsinki	5
4	Kaarina	Jokinen	Tampere	3
5	Aino	Virtanen	Helsinki	1
6	Soile	Auvila	Savonlinna	4
7	Katri	Ojala	Tampere	2
8	Kirsi	Koivisto	Hämeenlinna	1
9	Tanja	Ikonen	Jyväskylä	3
10	Eeva	Svard	Jyväskylä	2
NULL	NULL	NULL	NULL	NULL

shoppingcart

row_id	c_id	p_id	order_id
1	2	3	10012
2	2	4	10012
3	2	1	10012
4	4	8	10013
5	4	7	10013
6	4	7	10013
7	4	5	10013
8	1	8	10011
9	1	4	10011
10	1	3	10011
11	6	2	10015
12	6	9	10015
13	6	8	10015
14	6	3	10015
15	6	3	10015
16	8	1	10014
17	8	2	10014
18	8	3	10014
19	8	4	10014
20	8	5	10014
21	8	6	10014
22	8	7	10014
23	8	8	10014
24	8	9	10014
NULL	NULL	NULL	NULL

cm_orders

order_id	delivery_options_do_id	order_time	total
10011	2	2021-01-02 00:00:00	0
10012	3	2021-02-28 00:00:00	0
10013	1	2021-04-06 00:00:00	0
10014	1	2022-05-11 00:00:00	0
10015	4	2022-07-08 00:00:00	0
10016	2	2022-06-21 00:00:00	0
10017	3	2022-09-26 00:00:00	0
NULL	NULL	NULL	NULL

delivery_option

do_id	delivery_option_name	delivery_price
1	posti	4.9
2	matkahuolto	2.9
3	ups	5.8
4	nouto	0
NULL	NULL	NULL

product

product_id	p_name	p_price	c_id	m_id
1	Curly Mascara	17.9	2	1
2	Tasapainottava Savinaamio	23.9	2	1
3	Luminous Moisture Huulipuna	13.5	2	1
4	Aito Omena Shampoo 250ml	3.9	1	2
5	Kaura Hiusöljy 100ml	5.9	1	2
6	Teint Miracle Meikkivoide	52	2	3
7	24H Drama Liqui-Pencil Rajauskynä	24	2	3
8	Luomu Soft Wild Rose Pink Salt Vartalok...	11.9	3	4
9	Luomu Wild Lemongrass Deodorantti 5...	5.5	3	4
NULL	NULL	NULL	NULL	NULL

category

	category_id	c_name
▶	1	hair
	2	face
	3	body
*	NULL	NULL

manufacture

manufacture_id	m_name	m_country
1	Lumene	Finland
2	XZ	Finland
3	LancÔme	France
4	Urtekram	Denmark
NULL	NULL	NULL

Kyselyt

Testataan kyselyiden avulla tietokannan toimivuus:

1. Hae kaikki asiakkaat. Jos asiakkaalla ei ole jäsenyyttä, lisää kenttään ”not clubmember ☹”

   HUOM! Tämä osio suoritettu silloin, kun ’clubmember’ taulussa ei ole ollut arvoa 5 = notmember. Jouduin myöhemmässä vaiheessa lisäämään arvon tauluun, kun en toisessa kyselyssä saanut tavoiteltua tulosta, koska kysely ei ottanut huomioon NULL – klubijäsen arvolla olevia asiakkaita

```
SELECT firstname, lastname, IFNULL(clubmember_lvl, 'Not clubmember :(') as clubmember
FROM customer
LEFT JOIN clubmember
ON customer.cm_id = clubmember.clubmember_id;
```

firstname	lastname	clubmember
Aila	Mäkinen	Not clubmember :(
Helmi	Haapamäki	Silver
Kaija	Väisänen	Not clubmember :(
Kaarina	Jokinen	Copper
Aino	Virtanen	Gold
Soile	Auvila	Sprout
Katri	Ojala	Silver
Kirsi	Koivisto	Gold
Tanja	Ikonen	Copper
Eeva	Svard	Silver

2. Hae kaikki asiakkaat, jotka kuuluvat clubjäsen luokkaan ”Silver”.

```
SELECT firstname, lastname, clubmember_lvl as clubmember, city
FROM customer
LEFT JOIN clubmember
ON customer.cm_id = clubmember.clubmember_id
WHERE clubmember_lvl = 'Silver';
```

firstname	lastname	clubmember	city
Helmi	Haapamäki	Silver	Hämeenlinna
Katri	Ojala	Silver	Tampere
Eeva	Svard	Silver	Jyväskylä

3. Hae kaikki tuotteet, joidenka valmistusmaa on ”Suomi” ja hinta on yli 15 €

```
SELECT p_name AS 'Product name', p_price AS 'Price'
FROM product
LEFT JOIN manufacture
ON product.m_id = manufacture.manufacture_id
WHERE m_country = 'Finland' AND p_price > 15;
```

	Product name	Price
▶	Curly Mascara	17.9
	Tasapainottava Savinaamio	23.9

4. Hae kaikki tilaukset taulusta ”shoppingcart” ja järjestele tilaukset asiakkaan mukaan ja laske tuotteiden yhteissumma (huomioi klubijäsenyys) *HUOM tämän kyselyn kohdalla tulin muokanneeksi tietokannan taulujen tietoja ’clubmember’-taulussa, Aila Mäkinen olisi jäännyt tuloksesta näkymättä ”NULL”-arvon vuoksi.*

ALLA LOITSU JOLLA TESTAAMME SUMMAA ILMAN JÄSENALENNUSTA:

```
SELECT customer_id, firstname, lastname, ROUND(SUM(p_price), 2) AS 'Price'
FROM customer
JOIN shoppingcart
ON customer.customer_id = shoppingcart.c_id
JOIN product
ON shoppingcart.p_id = product.product_id
GROUP BY customer_id;
```

	customer_id	firstname	lastname	Price
▶	1	Aila	Mäkinen	29.30
	2	Helmi	Haapamäki	35.30
	4	Kaarina	Jokinen	65.80
	6	Soile	Auvila	68.30
	8	Kirsi	Koivisto	158.50

Olen tarkoituksella valinnut ’shoppingcart’- tauluun asiakkaita, joista jokainen edustaa 1 klubijäsentä (jokaisella

eri alennusprosentti)

JÄSENALENNUS LASKETTUNA

```
SELECT customer_id, firstname, lastname, ROUND(SUM(p_price * (1 - clubmember.discount)), 2) AS 'Price'
FROM clubmember
JOIN customer
ON clubmember.clubmember_id = customer.cm_id
JOIN shoppingcart
ON customer.customer_id = shoppingcart.c_id
JOIN product
ON shoppingcart.p_id = product.product_id
GROUP BY customer_id;
```

customer_id	firstname	lastname	Price
1	Aila	Mäkinen	29.30
2	Helmi	Haapamäki	22.94
4	Kaarina	Jokinen	49.35
6	Soile	Auvila	61.47
8	Kirsi	Koivisto	87.18

Asiakas Aila ei kuulu klubiin, joten hänelle ei alennusta. Asiakas Helmi saa ”*Silver*”in verran alennusta eli -35% loppusummasta. Asiakas Kaarina ”*Copper*” alennus on -25% ja Asiakas Soile ”*Sprout*” -10%. Asiakas Kirsi edustaa ahkeraa tilaajaa ”*Gold*”-jäsenenä, kun loppusummasta tippuu pois -45%. (Price ei sisällä kuljetusta)

5. Hae kaikki tilaajat ja heidän toimitustapansa ja kaupunki

```
SELECT lastname, firstname, delivery_options.delivery_option_name AS delivery, city
FROM customer
JOIN shoppingcart
ON customer.customer_id = shoppingcart.c_id
JOIN cm_orders
ON shoppingcart.order_id = cm_orders.order_id
JOIN delivery_options
ON cm_orders.delivery_options_do_id = delivery_options.do_id
GROUP BY customer_id;
```

	lastname	firstname	delivery	city
▶	Mäkinen	Aila	matkahuolto	Jyväskylä
	Haapamäki	Helmi	ups	Hämeenlinna
	Jokinen	Kaarina	posti	Tampere
	Auvila	Soile	nouto	Savonlinna
	Koivisto	Kirsi	posti	Hämeenlinna

6. Otetaan lopuksi TOP 3 eniten tilatut tuotteet

```
SELECT manufacture.m_name AS Brand, product.p_name AS Product, COUNT(product.p_name) AS Amount
FROM manufacture
LEFT JOIN product
ON manufacture.manufacture_id = product.m_id
LEFT JOIN shoppingcart
ON product.product_id = shoppingcart.p_id
GROUP BY p_id
ORDER BY Amount DESC LIMIT 3;
```

	Brand	Product	Amount
▶	Lumene	Luminous Moisture Huulipuna	5
	Urtekram	Luomu Soft Wild Rose Pink Salt Vartaloku...	4
	XZ	Aito Omena Shampoo 250ml	3

7. Huomasimme, että *'cm_orders*' -taulussa on tyhjä sarake '*total*'. Lisätään sarakkeeseen tilausten summa + toimitus

Taulu ennen tietojen lisäämistä:

order_id	delivery_options_do_id	order_time	total
10011	2	2021-01-02 00:00:00	0
10012	3	2021-02-28 00:00:00	0
10013	1	2021-04-06 00:00:00	0
10014	1	2022-05-11 00:00:00	0
10015	4	2022-07-08 00:00:00	0
10016	2	2022-06-21 00:00:00	0
10017	3	2022-09-26 00:00:00	0
NULL	NULL	NULL	NULL

MySQL -komento:

```
UPDATE cm_orders o

SET o.total = (
                SELECT ROUND(SUM(p.p_price * (1 - m.discount)), 2) AS total
                FROM clubmember m
                JOIN customer c ON m.clubmember_id = c.cm_id
                JOIN shoppingcart s ON c.customer_id = s.c_id
                JOIN product p ON s.p_id = p.product_id
   WHERE s.order_id = o.order_id
                GROUP BY c.customer_id
);
```

Taulu UPDATE jälkeen:

order_id	delivery_options_do_id	order_time	total
10011	2	2021-01-02 00:00:00	29
10012	3	2021-02-28 00:00:00	23
10013	1	2021-04-06 00:00:00	49
10014	1	2022-05-11 00:00:00	87
10015	4	2022-07-08 00:00:00	61
10016	2	2022-06-21 00:00:00	0
10017	3	2022-09-26 00:00:00	0
NULL	NULL	NULL	NULL

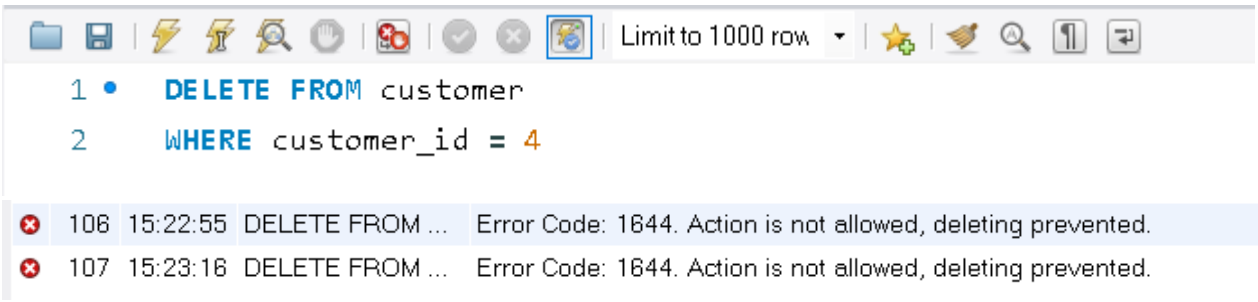
Harmikseni, en osannut summata postitusta 'total' -sarakkeeseen, mutta huomaamme, että onnistuin lisäämään tauluun komennolla hinnat. Muilla tilauksilla ei ole tuotteita, niin summaa ei tule 10016 ja 10017 tilauksille.

Samalla tuli tässä harjoitteessa testattua, miten saisin lyhyemmin tehtyä kyselyitä nimeämällä taulut kirjaimella.

Testataan triggerit:

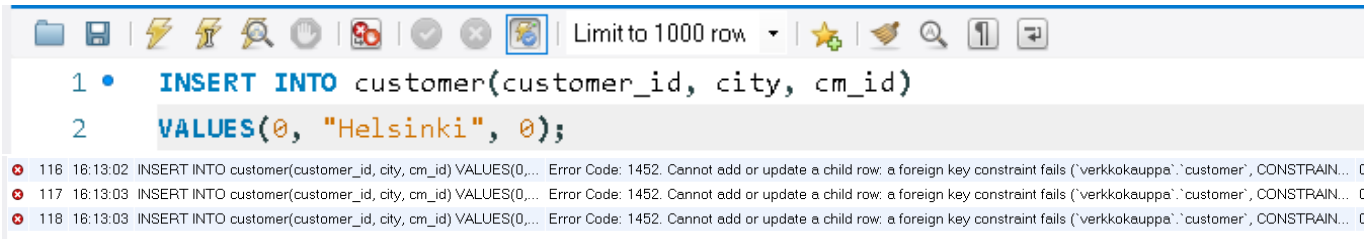
1. Asiakasta EI voi poistaa

```
CREATE DEFINER = CURRENT_USER TRIGGER `verkkokauppa`.`customer_BEFORE_DELETE`
BEFORE DELETE ON `customer` FOR EACH ROW
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Action is not allowed, deleting prevented.";
```



2. Asiakasta ei voi lisätä, jos kentässä ei ole nimeä

```
CREATE DEFINER = CURRENT_USER TRIGGER `verkkokauppa`.`customer_BEFORE_INSERT`
BEFORE INSERT ON `customer` FOR EACH ROW
BEGIN
                IF NEW.lastname IS NULL OR NEW.firstname IS NULL
  THEN
  DELETE FROM customer WHERE NEW.lastname IS NULL OR NEW.firstname IS NULL;
  SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "The Information is required";
  END IF;
END
```



Varoitusteksti antaa foreingkey restrictionin, mutta ei lisää uutta riviä tauluun. Tätä tuli testattua aikaisemmin monta kertaa ja rivejä tuli fk:sta huolimatta. Nyt riviä ei ilmesty ollenkaan eli triggeri toimii!

Pohdinta ja oma arvio

Työ oli itselleni sopivan haastava. Myönnän, että panostamalla työhön ajallisesti enemmän, olisin pystynyt lisätä lisää triggereitä ja ehtoja taulujen välille ja saada monimuotoisempia tuloksia esiteltyä tässä dokumentaatiossa. Esimerkiksi tietoa lisätessä ’*product*’- tauluun, olisi triggereillä voitu rajata, kuinka korkea hinta tuotteella saa maksimissaan olla tai tilaukseen lisäämällä tuotteita, muuttuu samalla asiakkaan loppusumma ’cm_order’-taulussa.

Ymmärrän omasta mielestäni hyvin perusteellisesti MySQL komennot. Osaan käyttää hakuehtoja keskivertoisesti, mutta vielä täytyy tulevaisuutta ajatellen harjoitella lisää JOIN-ehdoja ja alikyselyiden käyttöä oikein. Vaikeaa oli yrittää keksiä, mitä erityisesti haluamme hakea tietokannasta, mutta toivottavasti pystyin laajalti käyttämään hyväksi kaikkia tauluja. Jätin tarkoituksella ’category’-taulun, vaikka kyseistä taulua ei nyt tässä harjoituksessa käytetty. Halusin ajatella, että jos minulla olisi oma verkkokauppa, niin tämän mallinen olisi tämän tietokannan rakenne.