Semester and Year

PROJECT TITLE

COMP/MIS-302 DATABASE MANAGEMENT SYSTEMS

GROUP PARTICIPANT(S)

NOTE: This documentation report should be submitted within the project deadline in a compressed file along with the database file that proves the implementation of the database.

Table of Contents

THE CASE AND APPLICATION REQUIREMENTS

1.1 The Case/Business

CASE: A database for a dairy company

“DairyCY Inc.“ (anonymous) is a dairy company that is located in Cyprus. It runs a long-established business in Cyprus and one of the most famous dairy companies in Cyprus.

Its main operations are to produce, import, sell and export dairy products such as milk, cheese, ice cream, yogurt and so forth. The company owns its factories to produce the daily products. At the same time, however, it has a partnership with companies all over the world. Thus, some of the products are imported, not produced in the factories. The products which are either produced in its own factories or imported are sold to not only local people in Cyprus, but also exported to countries in the world where there is an partnership agreement.

In the project, considering the limited amount of time, the focus is put especially on the development of the products and logistics of them. Therefore, the data model and ad-hoc queries developed in this project only contains entities and activities regarding the logistics.

1.2 Application Requirements

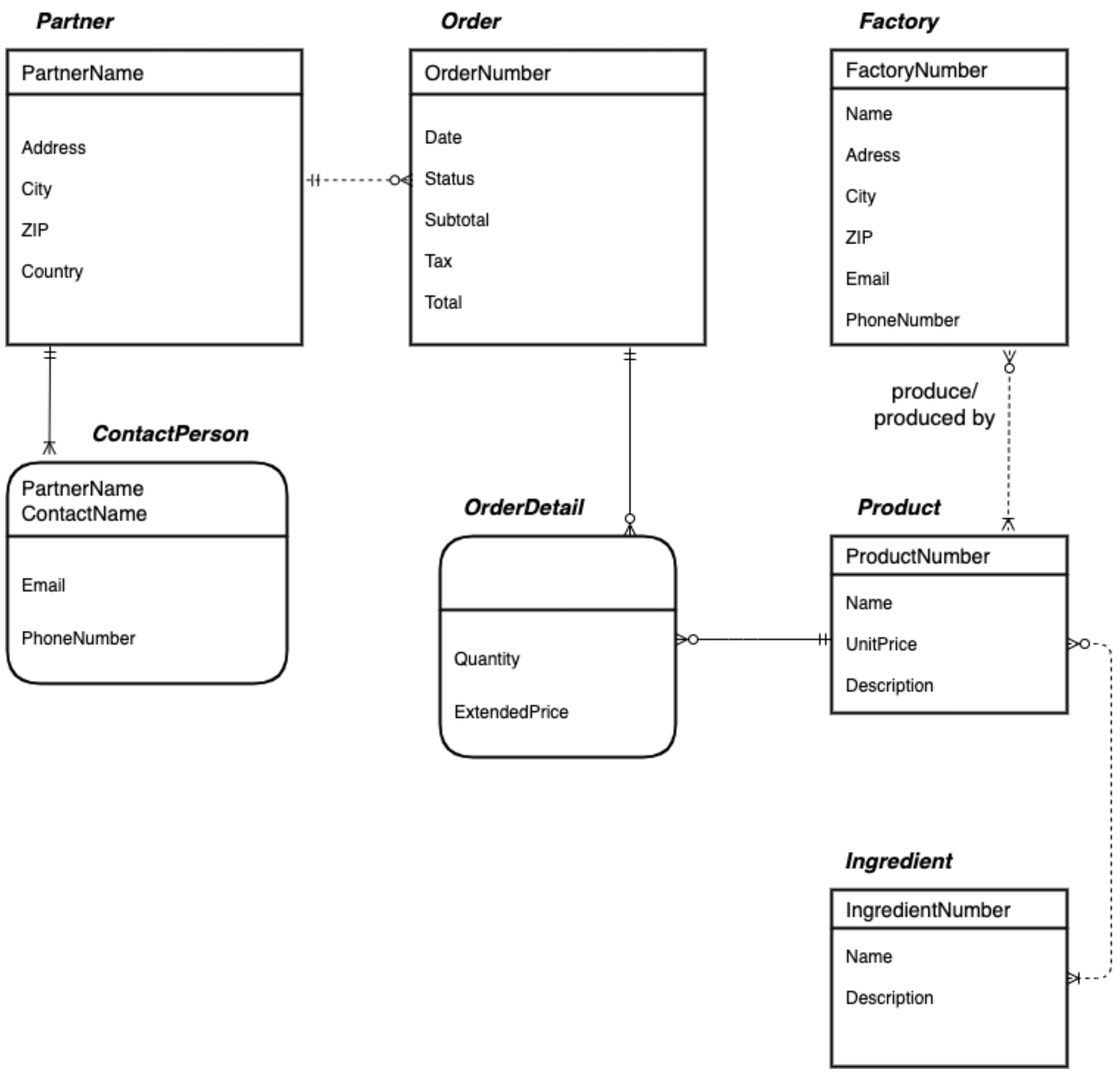
Based on the description above, the possible application of the database would be following;

<u>Querying the information regarding;</u>

- The specific partner companies such as its address and its contact persons.
- The owned factories such as the address.
- The owned products such as its unit price, description and associated ingredients.
- The order details regarding buying and selling activities such as the subtotal, tax and so on, and the calculated total amount from the recorded data.
- The detailed information regarding the specific order such as the kinds and quantity of the products which is bought or sold.

STEP 1 – CREATE DATA MODEL FROM APPLICATION REQUIREMENTS

2.1 Data Model



2.2 Supportive Documentation

Based on the description above, the possible entities for the database are assumed as follows; ***Factory, Product, Ingredient, Order, OrderDetail,Partner****,* and ***ContactPerson***.

[***Factory***]

This is the entity to store the details regarding the factories owned by the company such as its name, address and so forth.

[***Product*** and ***Ingredient***]

Then, the next entity is the *Product*. In the entity, the data regarding the products such as product name, unit price and so forth are going to be stored. Then, the *Ingredient* entity is another entity which can store the data regarding the ingredients of each product.

One thing important to consider here is that all the products of the company are not produced only by the owned factories. Some of them are imported from partner companies all over the world. Therefore, the *Product* entity should not have a relationship only with the *Factory* entity, but also with another entity, the *Partner* entity.

[***Partner*** and ***ContactPerson***]

The *Partner* entity is the entity which stores the date regarding the partner companies such as name, location and so on. Assuming that there could be more than one contact person in a partner company, a weak entity called *ContactPerson* is placed.

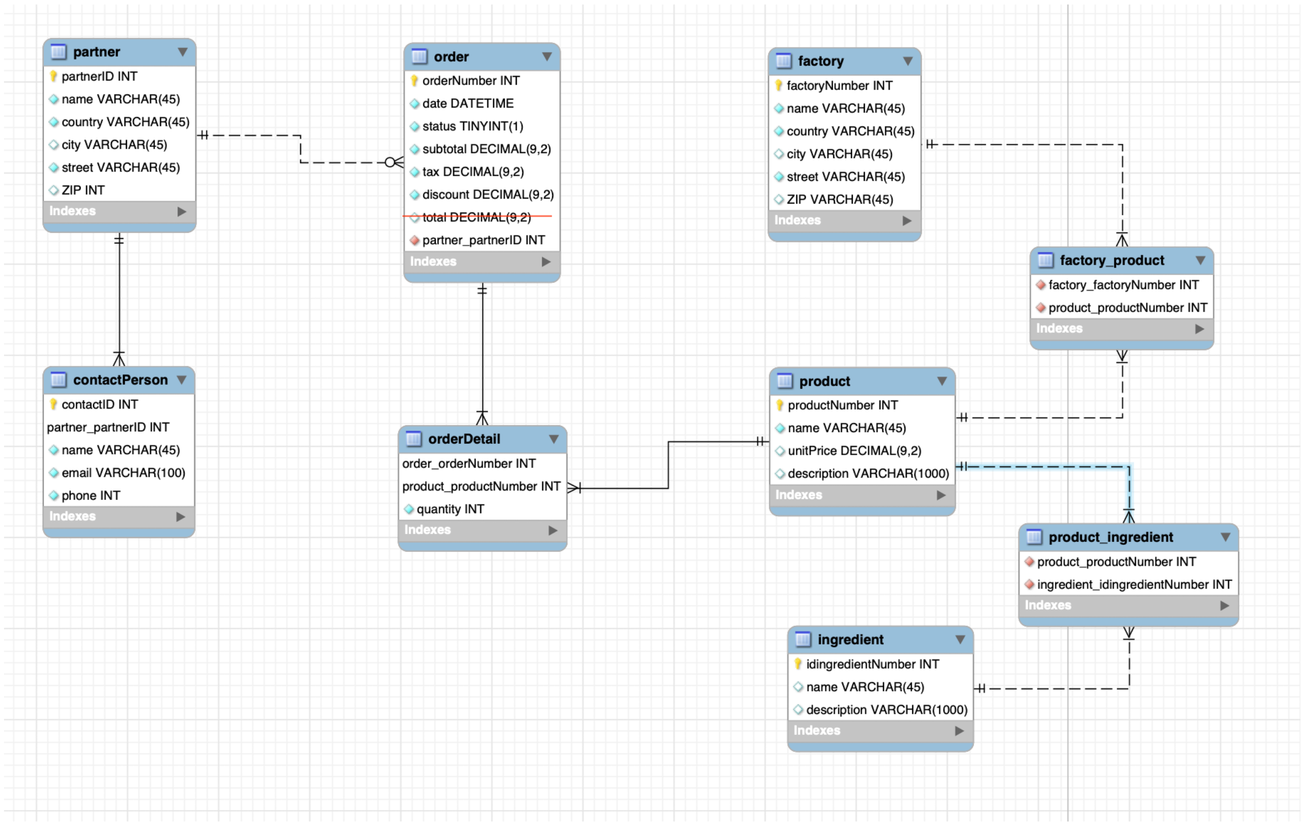
In order to store the data for the trading between the company and the partner companies, other entities *Order* and *OrderDetail* are placed between the *Product* and the *Partner* entities.

[***Order*** and ***OrderDetail***]

The *Order* and the *OrderDetail* are entities which store data regarding the trade specifically for each company and each country. Assuming that there is not only an import of the products, but also an export of the products between the company and the partner companies, those entities are designed to store both importing and exporting order information and to be easily differentiated.

STEP 2 – TRANSFORM DATA MODEL INTO DATABASE DESIGN

2.1 Database Design



2.2 Normalization

As it is striked through with the red line in the database design above, “total” attribute is deleted because there is a partial dependency between “total” and other attributes such as “subtotal”, “tax” and “discount,” which violates the 2NF. Thus, “total” is deleted and it is going to be calculated upon the queries.

2.3 Data Dictionary

Partner

Column Name	Type	Key	NULL status	Remarks
PartnerID	Int	Primary	NOT NULL	Surrogate Key
PartnerName	Varchar(45)	No	NOT NULL	
Street	Varchar(45)	No	NOT NULL	
City	Varchar(45)	No	NULL	
ZIP	Int	No	NULL	
Country	Varchar(45)	No	NOT NULL	

ContactPerson

Column Name	Type	Key	NULL status	Remarks
PartnerID	Int	Foreign and Primary key	NOT NULL	
ContactID	Int	Primary key	NOT NULL	Surrogate key
ContactName	Varchar(45)	No	NOT NULL	
Email	Varchar(100)	No	NOT NULL	
PhoneNumber	Int	No	NOT NULL	

Order

Column Name	Type	Key	NULL status	Remarks
OrderNumber	Int	Primary key	NOT NULL	Surrogate key
Date	Date	No	NOT NULL	
Status	TinyInt	No	NOT NULL	0: buy from partner 1: sell to partner
Subtotal	Decimal(9,2)	No	NOT NULL	
Tax	Decimal(9,2)	No	NOT NULL	
Discount	Decimal(9,2)	No	NOT NULL	
PartnerID	Int	Foreign key	NOT NULL	

OrderDetail

Column Name	Type	Key	NULL status	Remarks
OrderNumber	Int	Foreign and primary key	NOT NULL	
ProductNumber	Int	Foreign and primary key	NOT NULL	
Quantity	Int	No	NOT NULL	

Product

Column Name	Type	Key	NULL status	Remarks
ProductNumber	Int	Primary key	NOT NULL	Surrogate key
Name	Varchar (45)	No	NOT NULL	
UnitPrice	Decimal(9,2)	No	NOT NULL	
Description	Varchar (1000)	No	NULL	

Factory

Column Name	Type	Key	NULL status	Remarks
FactoryNumber	Int	Primary key	NOT NULL	Surrogate key
Name	Varchar (45)	No	NOT NULL	
street	Varchar(45)	No	NOT NULL	
City	Varchar(45)	No	NULL	
ZIP	Int	No	NULL	

Factory_and_Product

Column Name	Type	Key	NULL status	Remarks
FactoryNumber	Int	Foreign and primary key	NOT NULL	
ProductNumber	Int	Foreign and primary key	NOT NULL	

Ingredient

Column Name	Type	Key	NULL status	Remarks
IngredientNumber	Int	Primary key	NOT NULL	Surrogate key
Name	Varchar (45)	No	NOT NULL	
Description	Varchar (1000)	No	NULL	

Product_and_Ingredient

Column Name	Type	Key	NULL status	Remarks
ProductNumber	Int	Foreign and primary key	NOT NULL	
IngredientNumber	Int	Foreign and primary key	NOT NULL	

2.4 Minimum Cardinality Enforcement

Partner to ContactPerson (M:M)

ContactPerson Is required child	Action on Partner (Parent)	Action on ContactPerson (Child)
Insert	INSERT trigger on Parent to create row in Child. Child will be given all the data since all the columns cannot be NULL.	Will be created by INSERT trigger on Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	Prohibit - business rule.

Partner to Order (M:O)

Partner Is required parent	Action on Partner (Parent)	Action on Order (Child)
Insert	None.	Get Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	None.

Order to OrderDetail (M:O)

Order Is required parent	Action on Order (Parent)	Action on OrderDetail (Child)
Insert	None.	Get Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	None.

Product to OrderDetail (M:O)

Product Is required parent	Action on Product (Parent)	Action on OrderDetail (Child)
Insert	None.	Get Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	None.

Factory to Factory_and_Product (M:O)

Factory Is required parent	Action on Factory (Parent)	Action on Factory_and_Product (Child)
Insert	None.	Get Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	None.

Product to Factory_and_Product (M:M)

Factory_and_Product Is required child	Action on Product (Parent)	Action on Factory_and_Product (Child)
Insert	INSERT trigger on Parent to create row in Child.	Will be created by INSERT trigger on Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	Prohibit - business rule. Instead, put comments in remarks.

Product to Product_and_Ingredient (M:O)

Product Is required parent	Action on Product (Parent)	Action on Product_and_Ingredient (Child)
Insert	None.	Get Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	None.

Ingredient to Product_and_Ingredient (M:M)

Product_and_Ingredient Is required child	Action on Ingredient (Parent)	Action on Product_and_Ingredient (Child)
Insert	INSERT trigger on Parent to create row in Child.	Will be created by INSERT trigger on Parent.
Modify key or Foreign key	Prohibit - surrogate key.	Prohibit - Child must always refer to Parent associated with it.
Delete	Prohibit - business rule. Instead, put comments in remarks.	Prohibit - business rule. Instead, put comments in remarks.

STEP 3 – DATABASE IMPLEMENTATION

3.1 Database Creation

```sql
-- 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_FO
R_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema DairyCY
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema DairyCY
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `DairyCY` DEFAULT CHARACTER SET utf8 ;
USE `DairyCY` ;

-- -----------------------------------------------------
-- Table `DairyCY`.`partner`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`partner` (
  `partnerID` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `country` VARCHAR(45) NOT NULL,
  `city` VARCHAR(45) NULL,
  `street` VARCHAR(45) NOT NULL,
  `ZIP` INT NULL,
  PRIMARY KEY (`partnerID`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `DairyCY`.`contactPerson`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`contactPerson` (
  `contactID` INT NOT NULL AUTO_INCREMENT,
  `partner_partnerID` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `phone` INT NOT NULL,
  PRIMARY KEY (`contactID`, `partner_partnerID`),
  INDEX `fk_contactPerson_partner_idx` (`partner_partnerID` ASC) VISIBLE,
  CONSTRAINT `fk_contactPerson_partner`
    FOREIGN KEY (`partner_partnerID`)
    REFERENCES `DairyCY`.`partner` (`partnerID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DairyCY`.`order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`order` (
  `orderNumber` INT NOT NULL,
  `date` DATETIME NOT NULL,
  `status` TINYINT(1) NOT NULL,
  `subtotal` DECIMAL(9,2) NOT NULL,
  `tax` DECIMAL(9,2) NOT NULL,
  `discount` DECIMAL(9,2) NOT NULL DEFAULT 0,
  `total` DECIMAL(9,2) NOT NULL,
  `partner_partnerID` INT NOT NULL,
  PRIMARY KEY (`orderNumber`),
  INDEX `fk_order_partner1_idx` (`partner_partnerID` ASC) VISIBLE,
  CONSTRAINT `fk_order_partner1`
    FOREIGN KEY (`partner_partnerID`)
    REFERENCES `DairyCY`.`partner` (`partnerID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

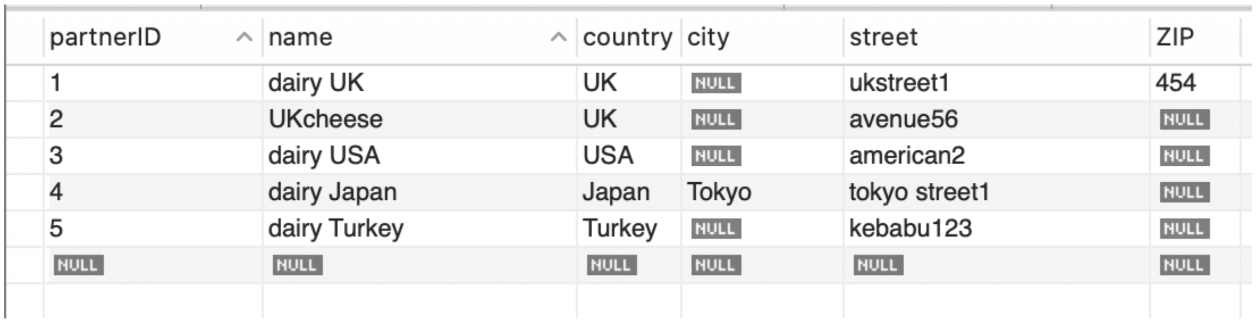
```
-- -----------------------------------------------------
-- Table `DairyCY`.`product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`product` (
  `productNumber` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `unitPrice` DECIMAL(9,2) NOT NULL,
  `description` VARCHAR(1000) NULL,
  PRIMARY KEY (`productNumber`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DairyCY`.`orderDetail`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`orderDetail` (
  `order_orderNumber` INT NOT NULL,
  `product_productNumber` INT NOT NULL,
  `quantity` INT NOT NULL,
  PRIMARY KEY (`order_orderNumber`, `product_productNumber`),
  INDEX `fk_orderDetail_product1_idx` (`product_productNumber` ASC) VISIBLE,
  CONSTRAINT `fk_orderDetail_order1`
    FOREIGN KEY (`order_orderNumber`)
    REFERENCES `DairyCY`.`order` (`orderNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_orderDetail_product1`
    FOREIGN KEY (`product_productNumber`)
    REFERENCES `DairyCY`.`product` (`productNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DairyCY`.`ingredient`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`ingredient` (
  `idingredientNumber` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `description` VARCHAR(1000) NULL,
  PRIMARY KEY (`idingredientNumber`))
ENGINE = InnoDB;
```

```sql
-- -----------------------------------------------------
-- Table `DairyCY`.`factory`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`factory` (
  `factoryNumber` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `country` VARCHAR(45) NOT NULL,
  `city` VARCHAR(45) NULL,
  `street` VARCHAR(45) NOT NULL,
  `ZIP` VARCHAR(45) NULL,
  PRIMARY KEY (`factoryNumber`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DairyCY`.`factory_product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`factory_product` (
  `factory_factoryNumber` INT NOT NULL,
  `product_productNumber` INT NOT NULL,
  INDEX `fk_factory_product_factory1_idx` (`factory_factoryNumber` ASC) VISIBLE,
  INDEX `fk_factory_product_product1_idx` (`product_productNumber` ASC) VISIBLE,
  CONSTRAINT `fk_factory_product_factory1`
    FOREIGN KEY (`factory_factoryNumber`)
    REFERENCES `DairyCY`.`factory` (`factoryNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_factory_product_product1`
    FOREIGN KEY (`product_productNumber`)
    REFERENCES `DairyCY`.`product` (`productNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```sql
-- -----------------------------------------------------
-- Table `DairyCY`.`product_ingredient`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DairyCY`.`product_ingredient` (
  `product_productNumber` INT NOT NULL,
  `ingredient_idingredientNumber` INT NOT NULL,
  INDEX `fk_product_ingredient_product1_idx` (`product_productNumber` ASC) VISIBLE,
  INDEX `fk_product_ingredient_ingredient1_idx` (`ingredient_idingredientNumber` ASC)
VISIBLE,
  CONSTRAINT `fk_product_ingredient_product1`
    FOREIGN KEY (`product_productNumber`)
    REFERENCES `DairyCY`.`product` (`productNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_product_ingredient_ingredient1`
    FOREIGN KEY (`ingredient_idingredientNumber`)
    REFERENCES `DairyCY`.`ingredient` (`idingredientNumber`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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

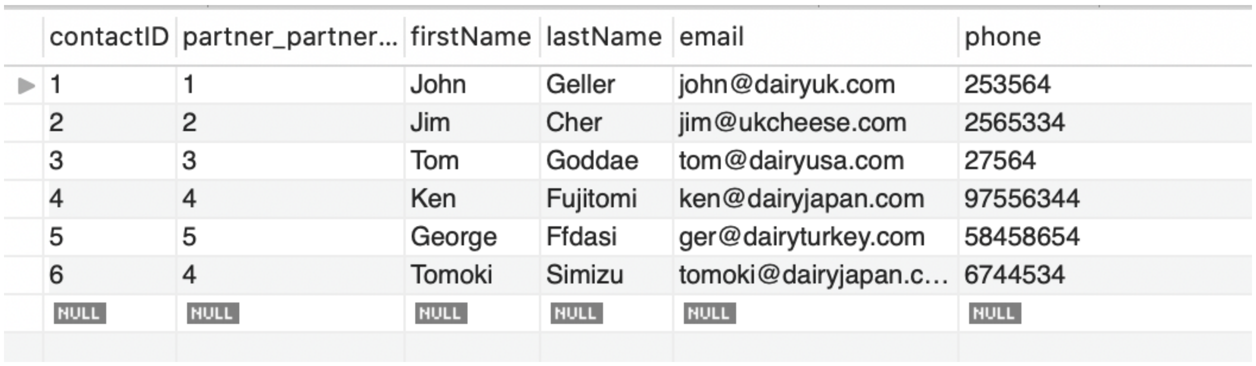
3.2 Insertion of Sample/Test Data

[partner]



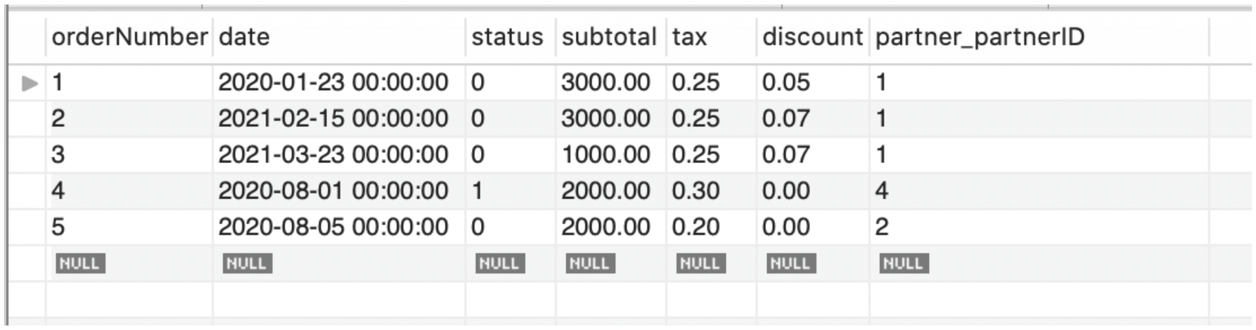
partnerID	name	country	city	street	ZIP
1	dairy UK	UK	NULL	ukstreet1	454
2	UKcheese	UK	NULL	avenue56	NULL
3	dairy USA	USA	NULL	american2	NULL
4	dairy Japan	Japan	Tokyo	tokyo street1	NULL
5	dairy Turkey	Turkey	NULL	kebabu123	NULL
NULL	NULL	NULL	NULL	NULL	NULL

[contactPerson]



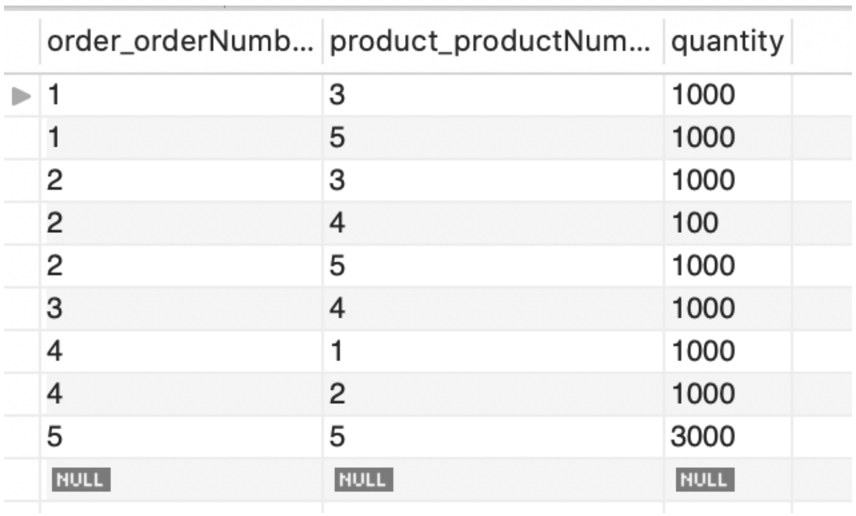
contactID	partner_partner...	firstName	lastName	email	phone
1	1	John	Geller	john@dairyuk.com	253564
2	2	Jim	Cher	jim@ukcheese.com	2565334
3	3	Tom	Goddae	tom@dairyusa.com	27564
4	4	Ken	Fujitomi	ken@dairyjapan.com	97556344
5	5	George	Ffdasi	ger@dairyturkey.com	58458654
6	4	Tomoki	Simizu	tomoki@dairyjapan.c...	6744534
NULL	NULL	NULL	NULL	NULL	NULL

[order]



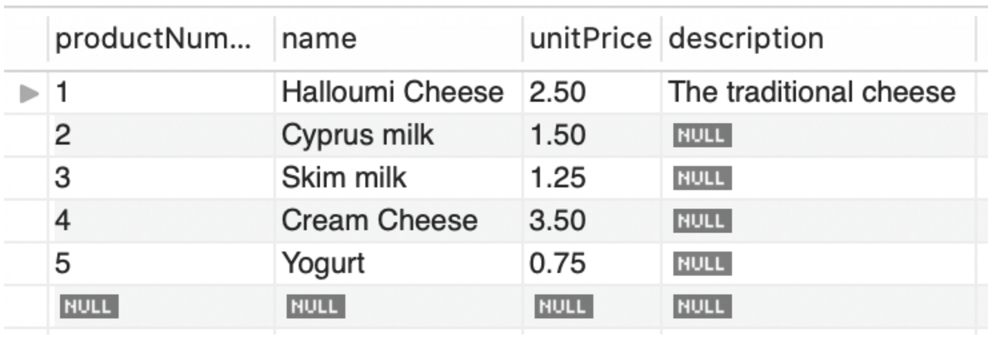
orderNumber	date	status	subtotal	tax	discount	partner_partnerID
1	2020-01-23 00:00:00	0	3000.00	0.25	0.05	1
2	2021-02-15 00:00:00	0	3000.00	0.25	0.07	1
3	2021-03-23 00:00:00	0	1000.00	0.25	0.07	1
4	2020-08-01 00:00:00	1	2000.00	0.30	0.00	4
5	2020-08-05 00:00:00	0	2000.00	0.20	0.00	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

[orderDetail]



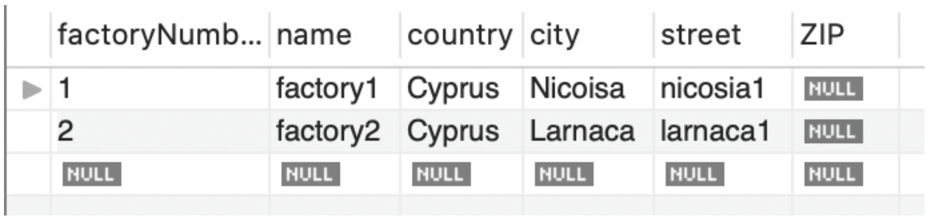
order_orderNumb...	product_productNum...	quantity
1	3	1000
1	5	1000
2	3	1000
2	4	100
2	5	1000
3	4	1000
4	1	1000
4	2	1000
5	5	3000
NULL	NULL	NULL

[product]



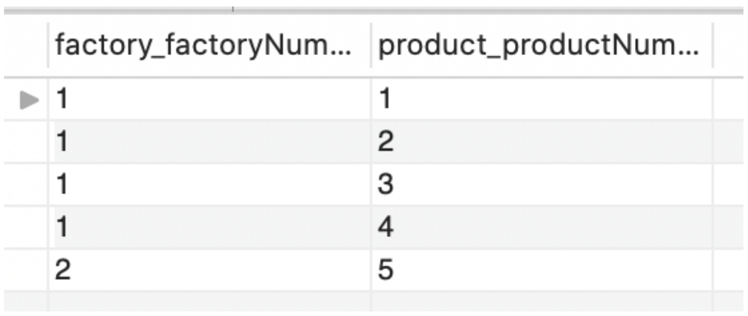
productNum...	name	unitPrice	description
1	Halloumi Cheese	2.50	The traditional cheese
2	Cyprus milk	1.50	NULL
3	Skim milk	1.25	NULL
4	Cream Cheese	3.50	NULL
5	Yogurt	0.75	NULL
NULL	NULL	NULL	NULL

[factory]



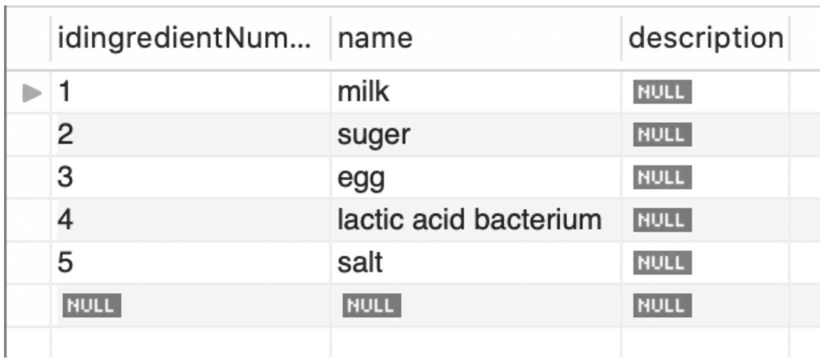
factoryNumb...	name	country	city	street	ZIP
1	factory1	Cyprus	Nicoisa	nicosia1	NULL
2	factory2	Cyprus	Larnaca	larnaca1	NULL
NULL	NULL	NULL	NULL	NULL	NULL

[factory_product]



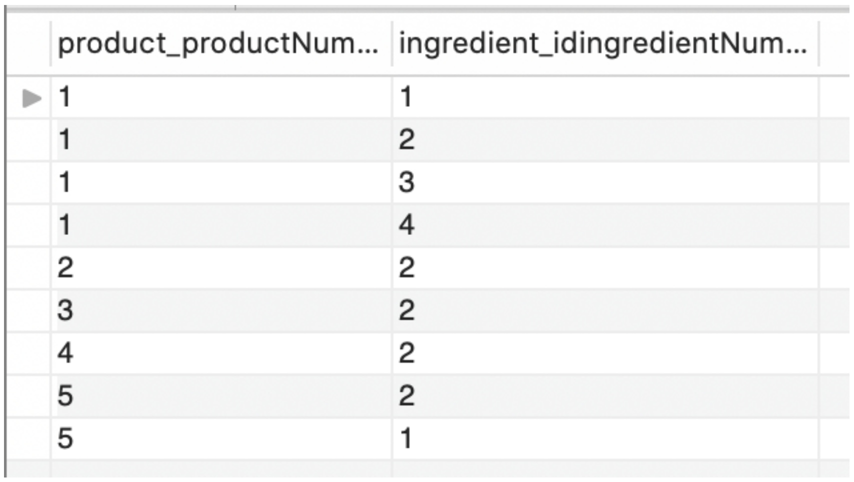
factory_factoryNum...	product_productNum...
1	1
1	2
1	3
1	4
2	5

[ingredient]



idingredientNum...	name	description
1	milk	NULL
2	suger	NULL
3	egg	NULL
4	lactic acid bacterium	NULL
5	salt	NULL
NULL	NULL	NULL

[product_ingredient]



product_productNum...	ingredient_idingredientNum...
1	1
1	2
1	3
1	4
2	2
3	2
4	2
5	2
5	1

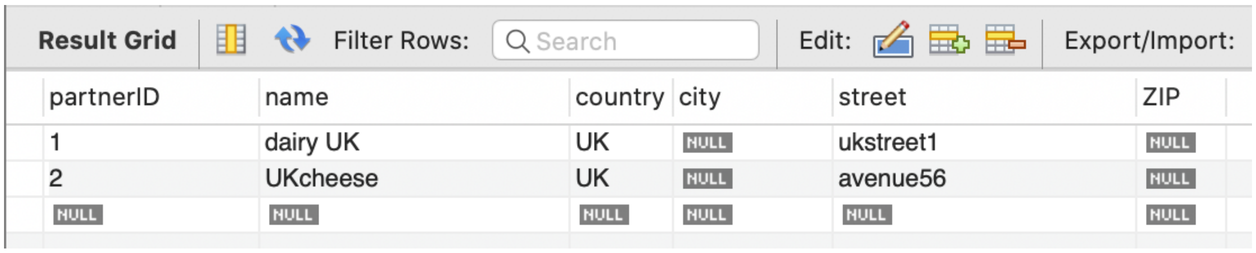
STEP 4 – QUERY PROCESSING

4.1 Query Implementation

```sql
/* ----- query1: Find partners in UK ----- */
SELECT *
FROM partner
WHERE country = 'UK';

/* ----- query2: show contact persons of 'dairy Japan' ----- */
SELECT
  p.partnerID,
  p.name,
  p.country,
  cp.firstName,
  cp.lastName,
  cp.email,
  cp.phone
FROM partner p
JOIN contactPerson cp USING(partnerID)
WHERE p.partnerID = 4

/* ----- query3: calculate the total amount for each order ----- */
SELECT
  orderNumber,
  (subtotal + (subtotal*tax))*(1-discount) AS 'Total amount'
FROM `order`

/* ----- query4: showing the product and quantity of the order No.1 ----- */
SELECT
  o.orderNumber,
  p.name AS 'product name',
  od.quantity,
  pa.name AS 'buying from'
FROM partner pa
JOIN `order` o USING (partnerID)
JOIN orderDetail od USING (orderNumber)
JOIN product p USING (productNumber)
WHERE o.orderNumber = 1;
```

```sql
/* ----- query5: showing the product whcich contains sugar as ingredient ----- */
SELECT
  p.name,
  p.unitPrice,
  i.name
FROM product p
JOIN product_ingredient pi USING (productNumber)
JOIN ingredient i USING (ingredientNumber)
WHERE ingredientNumber = 2

/* ----- query6: calculating the total amount for buying products ----- */
SELECT
  SUM((subtotal + (subtotal*tax))*(1-discount)) AS 'Total amount bought'
FROM `order`
WHERE status = 0;

/* ----- query7: showing the product information produced by factory No.1 ----- */
SELECT *
FROM product
WHERE productNumber IN
(
  SELECT productNumber
  FROM factory_product
  WHERE factoryNumber = 1
)

/* ----- query8:  showing product whose name contains word 'cheese' ----- */
SELECT *
FROM product
WHERE name REGEXP 'cheese'

/* ----- query9: showing partners with ZIP code----- */
SELECT *
FROM partner
WHERE ZIP IS NOT NULL

/* ----- query10: orders with dicount----- */
SELECT *
FROM `order`
WHERE discount <> 0
```

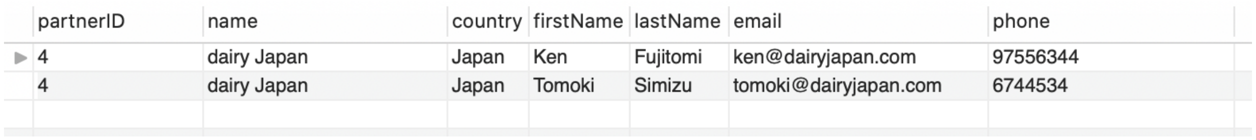
[query1]



Result Grid | Filter Rows: Search | Edit: | Export/Import:

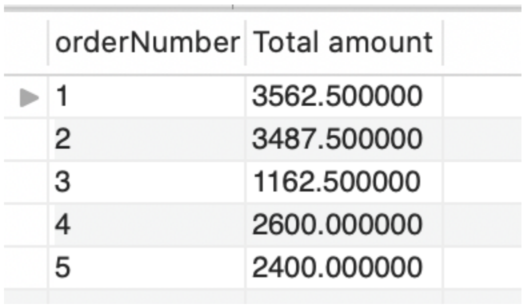
partnerID	name	country	city	street	ZIP
1	dairy UK	UK	NULL	ukstreet1	NULL
2	UKcheese	UK	NULL	avenue56	NULL
NULL	NULL	NULL	NULL	NULL	NULL

[query2]



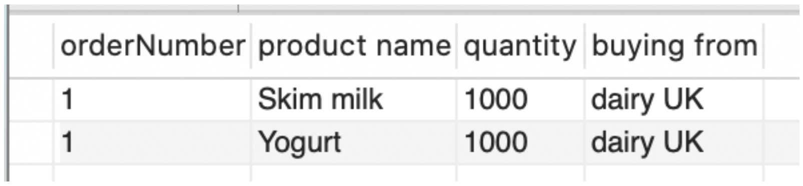
partnerID	name	country	firstName	lastName	email	phone
4	dairy Japan	Japan	Ken	Fujitomi	ken@dairyjapan.com	97556344
4	dairy Japan	Japan	Tomoki	Simizu	tomoki@dairyjapan.com	6744534

[query3]



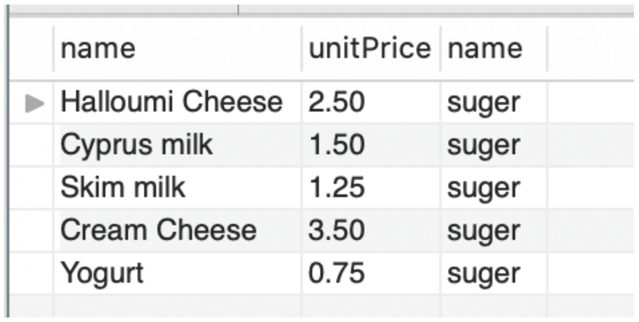
orderNumber	Total amount
1	3562.500000
2	3487.500000
3	1162.500000
4	2600.000000
5	2400.000000

[query4]



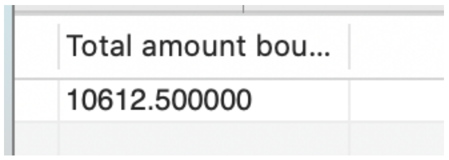
orderNumber	product name	quantity	buying from
1	Skim milk	1000	dairy UK
1	Yogurt	1000	dairy UK

[query5]



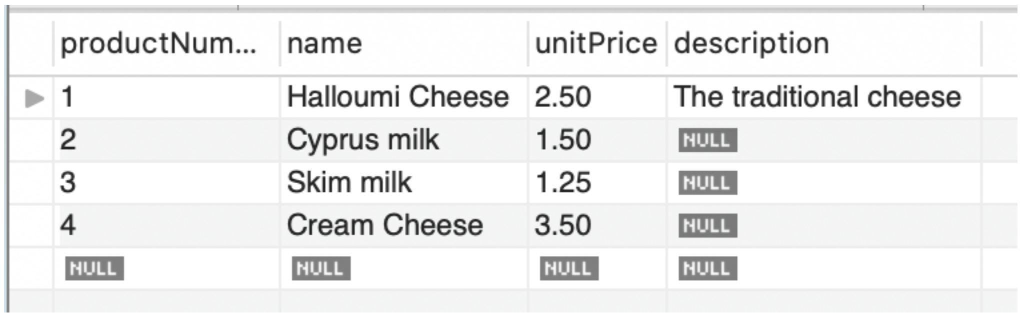
name	unitPrice	name
Halloumi Cheese	2.50	suger
Cyprus milk	1.50	suger
Skim milk	1.25	suger
Cream Cheese	3.50	suger
Yogurt	0.75	suger

[query6]



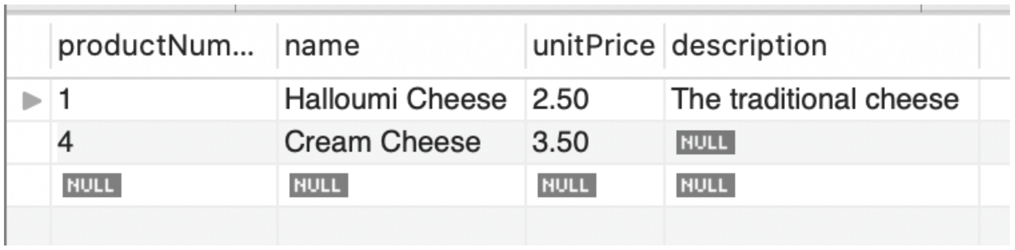
Total amount bou...
10612.500000

[query7]



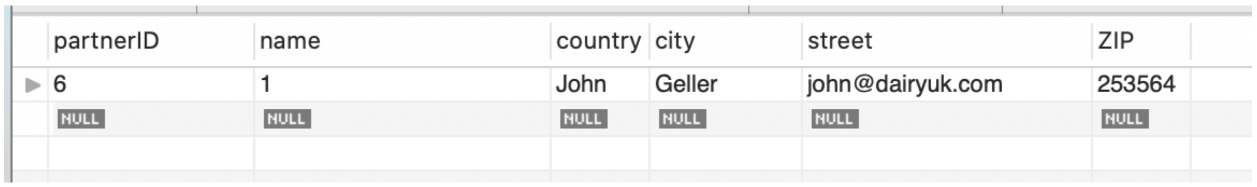
productNum...	name	unitPrice	description
1	Halloumi Cheese	2.50	The traditional cheese
2	Cyprus milk	1.50	NULL
3	Skim milk	1.25	NULL
4	Cream Cheese	3.50	NULL
NULL	NULL	NULL	NULL

[query8]



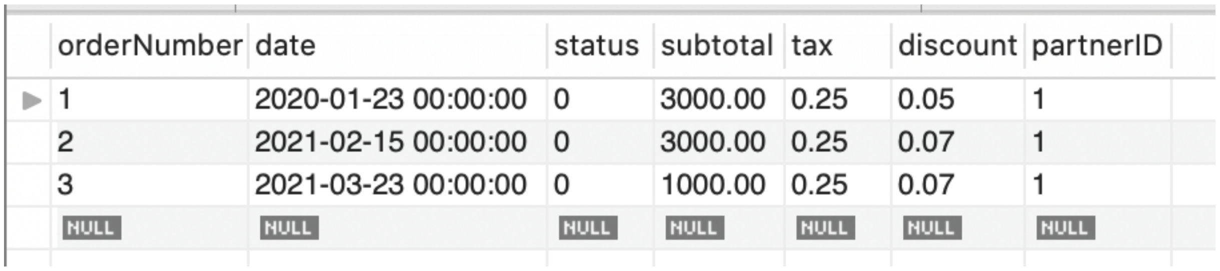
productNum...	name	unitPrice	description
1	Halloumi Cheese	2.50	The traditional cheese
4	Cream Cheese	3.50	NULL
NULL	NULL	NULL	NULL

[query9]



partnerID	name	country	city	street	ZIP
6	1	John	Geller	john@dairyuk.com	253564
NULL	NULL	NULL	NULL	NULL	NULL

[query10]



orderNumber	date	status	subtotal	tax	discount	partnerID
1	2020-01-23 00:00:00	0	3000.00	0.25	0.05	1
2	2021-02-15 00:00:00	0	3000.00	0.25	0.07	1
3	2021-03-23 00:00:00	0	1000.00	0.25	0.07	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL