MDE 515 - Project 1
Fall 2014

Student: Anastasios Antoniadis
Email: anantoni@di.uoa.gr
RN: M1381

SECTION 1 - Database Schema

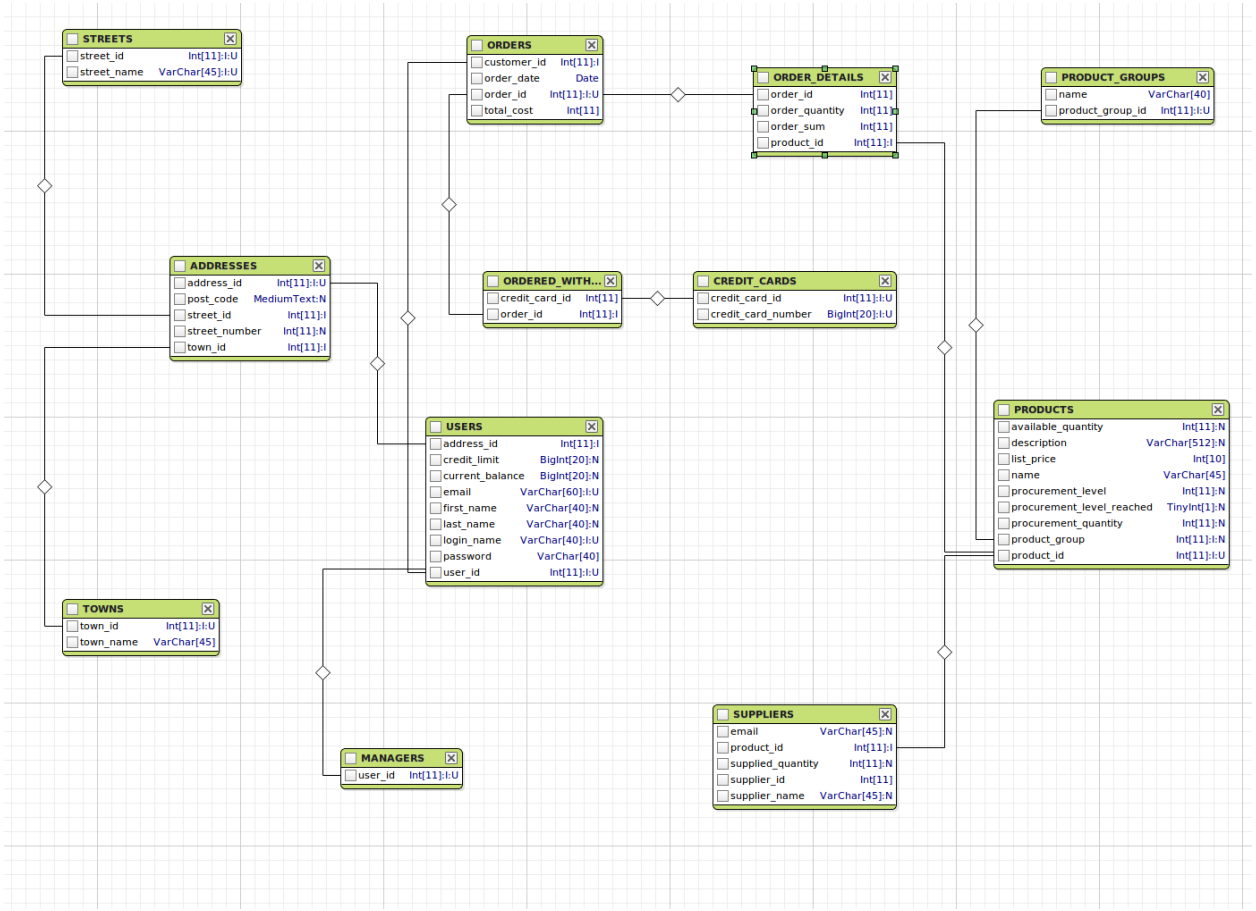

Illustration 1: Database Schema

- Regarding the users I decided one table for the streets and one for the town which are references by ADRESSES to avoid some duplicates
- I added a table for the users which are managers (MANAGERS).
- I connected the credit cards with the orders in a non-optimal way because i was undecided as to whether it was the best relationship choice. In the end I didn't modify it due to time constraints.
- I added one more field to PRODUCTS (procurement_level_reached). When it is set to 1 the manager needs to reorder products and update the counters in the database.
- I didn't modify the SUPPLIERS table, although I should have.

SECTION 2 - Stored Procedures

6. Product Browsing

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `searchProducts`(IN
product_name VARCHAR(45), IN product_description VARCHAR(512), IN
product_group_name VARCHAR(40), IN supplier_name VARCHAR(45), IN
order_choice VARCHAR(50))
BEGIN
	SET @s = 'SELECT * FROM PRODUCTS as p, PRODUCT_GROUPS AS pg';

	IF supplier_name IS NOT NULL THEN
		SET @s = CONCAT(@s, ', SUPPLIERS AS s');
	END IF;

	SET @s = CONCAT(@s, ' WHERE p.product_group = pg.product_group_id');

	IF supplier_name IS NOT NULL THEN
		SET @s = CONCAT(@s, ' AND p.product_id = s.product_id AND
s.supplier_name = ', QUOTE(supplier_name));
	END IF;
	IF product_group_name IS NOT NULL THEN
		SET @s = CONCAT(@s, ' AND pg.name = ',
QUOTE(product_group_name));
	END IF;
	IF product_name IS NOT NULL THEN
		SET @s = CONCAT(@s, ' AND p.name LIKE \'%', product_name, '%\'');
	END IF;
	IF product_description IS NOT NULL THEN
		SET @s = CONCAT(@s, ' AND p.description LIKE \'%',
product_description, '%\'');
	END IF;
	IF order_choice IS NOT NULL THEN
		SET @s = CONCAT(@s, order_choice);
	END IF;

	PREPARE stmt FROM @s;
  EXECUTE stmt;
  DEALLOCATE PREPARE stmt;
END
```

7a

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `orderSumMinMaxPerProduct`()
BEGIN
	SELECT p.product_id AS product_id, p.name AS product_name, description,
list_price, available_quantity, procurement_level, procurement_quantity,
			procurement_level_reached, MIN(o.order_sum) AS
min_order_sum, MAX(o.order_sum) AS max_order_sum
   FROM PRODUCTS AS p, ORDER_DETAILS AS o
   WHERE p.product_id = o.product_id
   GROUP BY p.product_id;
END
```

7b

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `daysGreaterThan10k`(IN
selected_month INT, IN selected_year INT)
BEGIN
	SELECT DAY(o.order_date) AS found_day
	FROM ORDERS AS o
	WHERE YEAR(o.order_date) = selected_year AND MONTH(o.order_date) =
selected_month
   GROUP BY o.order_date
   HAVING SUM(o.total_cost) > 10000;
END
```

7c

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `bestSellingProducts`()
BEGIN
	SELECT p.product_id, p.name AS product_name, description, list_price,
pg.name AS product_group,
		available_quantity, procurement_level, procurement_quantity,
procurement_level_reached, SUM(o.order_sum) AS order_sum
   FROM ORDER_DETAILS AS o, PRODUCTS AS p, PRODUCT_GROUPS AS pg
	WHERE o.product_id = p.product_id AND p.product_group = product_group_id
   GROUP BY o.product_id
   ORDER BY SUM(o.order_sum) DESC
   LIMIT 1;
```

```
END
```

7d

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`mostExpensiveProductPerGroup`()
BEGIN
	SELECT p1.product_id AS product_id, p1.name AS product_name,
p1.description AS description, p1.list_price AS list_price, p1.available_quantity AS
available_quantity,
		p1.procurement_quantity AS procurement_quantity,
p1.procurement_level AS procurement_level, p1.procurement_level_reached AS
procurement_level_reached
   FROM PRODUCTS AS p1, PRODUCT_GROUPS AS pg
   WHERE p1.product_group = pg.product_group_id AND
		p1.list_price =
		(SELECT MAX(p2.list_price) AS max_list_price
			FROM PRODUCTS AS p2
			WHERE p1.product_group = p2.product_group)
	GROUP BY p1.product_group;
END
```

7e

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `neverOrderedProducts`()
BEGIN
	SELECT *
   FROM PRODUCTS AS p
	WHERE p.product_id NOT IN(
		SELECT DISTINCT o.product_id
		FROM ORDER_DETAILS AS o);
END
```

7f

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`productsNotOrderedInMonthOfYear`(IN selected_month INT, IN selected_year INT)
BEGIN
	SELECT *
```

```
    FROM PRODUCTS AS p
        WHERE p.product_id NOT IN(
                SELECT DISTINCT od.product_id
                FROM ORDER_DETAILS AS od, ORDERS AS o
       WHERE od.order_id = o.order_id AND MONTH(o.order_date) = selected_month
AND YEAR(o.order_date) = selected_year);
END
```

8. Buying Suggestions

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `suggestProducts`( IN product_id
INT )
BEGIN
        SELECT p.product_id AS product_id, p.name AS product_name, p.description
AS description, p.list_price AS list_price, p.available_quantity AS available_quantity,
                        p.procurement_level AS procurement_level,
p.procurement_quantity AS procurement_quantity, p.procurement_level_reached AS
procurement_level_reached, COUNT(*) AS times_ordered
        FROM ORDER_DETAILS AS od1, ORDERS AS o1, ORDERS AS o2,
ORDER_DETAILS AS od2, PRODUCTS AS p
        WHERE o1.customer_id = o2.customer_id
                        AND o1.order_id = od1.order_id
                        AND o2.order_id = od2.order_id
                        AND od1.product_id != od2.product_id
                        AND od2.product_id = p.product_id
                        AND od1.product_id = product_id
                        GROUP BY od2.product_id
                        ORDER BY times_ordered DESC;
END
```

9. Six Degrees of Separation

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sixDegreesOfSeparation`(IN
supplier_id1 INT, IN supplier_id2 INT, OUT degree INT)
BEGIN
        DECLARE found INT;
        DROP TABLE IF EXISTS CURRENT_DEGREE_SUPPLIERS;
    DROP TABLE IF EXISTS PREVIOUS_DEGREE_SUPPLIERS;
```

```
   SET degree = 0;
        IF supplier_id1 != supplier_id2 THEN
                CREATE TABLE CURRENT_DEGREE_SUPPLIERS AS (
                        SELECT s2.supplier_id
                        FROM SUPPLIERS AS s1, SUPPLIERS AS s2
                        WHERE s1.product_id = s2.product_id AND s1.supplier_id =
supplier_id1 AND s1.supplier_id != s2.supplier_id
                        GROUP BY s1.supplier_id, s2.supplier_id);
      SET found = EXISTS(SELECT * FROM CURRENT_DEGREE_SUPPLIERS AS d1
WHERE d1.supplier_id = supplier_id2);
      SET degree = 1;

      IF found != 1 THEN
                        CREATE TABLE PREVIOUS_DEGREE_SUPPLIERS AS
(SELECT * FROM CURRENT_DEGREE_SUPPLIERS);
                        REPEAT
            TRUNCATE CURRENT_DEGREE_SUPPLIERS;
                                INSERT INTO CURRENT_DEGREE_SUPPLIERS (
                                        SELECT DISTINCT(s2.supplier_id)
                                        FROM PREVIOUS_DEGREE_SUPPLIERS AS d1,
SUPPLIERS AS s1, SUPPLIERS AS s2
                                        WHERE d1.supplier_id = s1.supplier_id AND
s1.product_id = s2.product_id AND s1.supplier_id != s2.supplier_id AND s2.supplier_id
NOT IN (SELECT * FROM PREVIOUS_DEGREE_SUPPLIERS));

            SET degree = degree + 1;
            SET found = EXISTS(SELECT * FROM CURRENT_DEGREE_SUPPLIERS
AS d1 WHERE d1.supplier_id = supplier_id2);
            TRUNCATE PREVIOUS_DEGREE_SUPPLIERS;
            INSERT INTO PREVIOUS_DEGREE_SUPPLIERS (SELECT * FROM
CURRENT_DEGREE_SUPPLIERS);
                        UNTIL degree > 6 OR found = 1
         END REPEAT;
                END IF;
        END IF;
   DROP TABLE IF EXISTS CURRENT_DEGREE_SUPPLIERS;
   DROP TABLE IF EXISTS PREVIOUS_ORDER_SUPPLIERS;
END
```

11. Statistics

- most popular products this week

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `mostPopularProducts`(IN
cur_date DATE, IN lim INT)
BEGIN
        SELECT od.product_id AS product_id,
                        p.name AS product_name,
         p.description AS description,
         p.list_price AS list_price,
         p.available_quantity AS available_quantity,
         p.procurement_level AS procurement_level,
         p.procurement_quantity AS procurement_quantity,
         p.procurement_level_reached AS procurement_level_reached,
         SUM(od.order_quantity) AS order_quantity,
         o.order_date AS order_date
   FROM PRODUCTS AS p, ORDERS AS o, ORDER_DETAILS AS od
   WHERE p.product_id = od.product_id
                AND o.order_id = od.order_id
      AND WEEK(cur_date) = WEEK(o.order_date)
      AND YEAR(cur_date) = YEAR(o.order_date)
        GROUP BY od.product_id
   ORDER BY order_quantity DESC
   LIMIT lim;
END
```

- most popular suppliers

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `mostPopularSuppliers`(IN lim
INT)
BEGIN
        SELECT supplier_id, supplier_name, email, SUM(s.supplied_quantity) AS
total_amount_supplied
   FROM SUPPLIERS AS s
   GROUP BY s.supplier_id
   ORDER BY SUM(s.supplied_quantity) DESC
   LIMIT lim;
END
```

- most popular post codes

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `mostPopularPostCodes`(IN lim
INT)
BEGIN
        SELECT a.post_code
   FROM USERS AS u, ADDRESSES AS a, ORDERS AS o
   WHERE a.address_id = u.address_id AND
                        u.user_id = o.customer_id
   GROUP BY a.post_code
   ORDER BY COUNT(DISTINCT o.order_id) DESC
   LIMIT lim;
END
```

12. Awards

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `topClients`(IN lim INT)
BEGIN
        SELECT o.customer_id AS customer_id, SUM(o.total_cost) AS total_cost
   FROM ORDERS AS o
   GROUP BY o.customer_id
   ORDER BY SUM(o.total_cost) DESC
   LIMIT lim;
END
```

3 - Create Schema

```
CREATE DATABASE  IF NOT EXISTS `fd_schema` /*!40100 DEFAULT CHARACTER
SET utf8 */;
USE `fd_schema`;
-- MySQL dump 10.13  Distrib 5.6.19, for linux-glibc2.5 (x86_64)
--
-- Host: 127.0.0.1    Database: fd_schema
-- ------------------------------------------------------
-- Server version      5.5.40-0ubuntu1

--
```

```
-- Table structure for table `PRODUCT_GROUPS`
--

DROP TABLE IF EXISTS `PRODUCT_GROUPS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `PRODUCT_GROUPS` (
  `product_group_id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(40) NOT NULL,
  PRIMARY KEY (`product_group_id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `PRODUCT_GROUPS`
--

LOCK TABLES `PRODUCT_GROUPS` WRITE;
/*!40000 ALTER TABLE `PRODUCT_GROUPS` DISABLE KEYS */;
INSERT INTO `PRODUCT_GROUPS` VALUES
(1,'V'),(2,'M'),(3,'C'),(4,'I'),(5,'B'),(6,'G'),(7,'H');
/*!40000 ALTER TABLE `PRODUCT_GROUPS` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `STREETS`
--

DROP TABLE IF EXISTS `STREETS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `STREETS` (
  `street_id` int(11) NOT NULL AUTO_INCREMENT,
  `street_name` varchar(45) NOT NULL,
  PRIMARY KEY (`street_id`),
  UNIQUE KEY `street_name_UNIQUE` (`street_name`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
CREATE TABLE `TOWNS` (
  `town_id` int(11) NOT NULL AUTO_INCREMENT,
  `town_name` varchar(45) NOT NULL,
  PRIMARY KEY (`town_id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;


DROP TABLE IF EXISTS `SUPPLIERS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `SUPPLIERS` (
  `supplier_id` int(11) NOT NULL AUTO_INCREMENT,
  `product_id` int(11) NOT NULL,
  `supplier_name` varchar(45) DEFAULT NULL,
  `email` varchar(45) DEFAULT NULL,
  `supplied_quantity` int(11) DEFAULT NULL,
  PRIMARY KEY (`supplier_id`,`product_id`),
  KEY `fk_suppliers_products1_idx` (`product_id`),
  CONSTRAINT `fk_suppliers_products1` FOREIGN KEY (`product_id`) REFERENCES
`PRODUCTS` (`product_id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;



--
-- Table structure for table `ADDRESSES`
--

DROP TABLE IF EXISTS `ADDRESSES`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ADDRESSES` (
  `address_id` int(11) NOT NULL AUTO_INCREMENT,
  `town_id` int(11) NOT NULL,
  `street_id` int(11) NOT NULL,
  `street_number` int(11) DEFAULT NULL,
```

```
  `post_code` mediumtext,
  PRIMARY KEY (`address_id`),
  KEY `fk_towns_has_streets_streets1_idx` (`street_id`),
  KEY `fk_towns_has_streets_towns_idx` (`town_id`),
  CONSTRAINT `fk_towns_has_streets_streets1` FOREIGN KEY (`street_id`)
REFERENCES `STREETS` (`street_id`) ON UPDATE CASCADE,
  CONSTRAINT `fk_towns_has_streets_towns` FOREIGN KEY (`town_id`)
REFERENCES `TOWNS` (`town_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `CREDIT_CARDS`
--

DROP TABLE IF EXISTS `CREDIT_CARDS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `CREDIT_CARDS` (
  `credit_card_id` int(11) NOT NULL AUTO_INCREMENT,
  `credit_card_number` bigint(20) NOT NULL,
  PRIMARY KEY (`credit_card_id`),
  UNIQUE KEY `unique_credit_card_id` (`credit_card_id`),
  UNIQUE KEY `unique_credit_card_number` (`credit_card_number`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Temporary view structure for view `FULL_USER_PROFILE`
--

DROP TABLE IF EXISTS `FULL_USER_PROFILE`;
/*!50001 DROP VIEW IF EXISTS `FULL_USER_PROFILE`*/;
SET @saved_cs_client     = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `FULL_USER_PROFILE` AS SELECT
 1 AS `user_id`,
 1 AS `login_name`,
 1 AS `first_name`,
```

```
 1 AS `last_name`,
 1 AS `email`,
 1 AS `credit_limit`,
 1 AS `current_balance`,
 1 AS `town_name`,
 1 AS `street_name`,
 1 AS `street_number`,
 1 AS `post_code`,
 1 AS `password`*/;
SET character_set_client = @saved_cs_client;

--
-- Table structure for table `MANAGERS`
--

DROP TABLE IF EXISTS `MANAGERS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `MANAGERS` (
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`user_id`),
  CONSTRAINT `fk_managers_users` FOREIGN KEY (`user_id`) REFERENCES
`USERS` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `ORDERED_WITH_CREDIT_CARD`
--

DROP TABLE IF EXISTS `ORDERED_WITH_CREDIT_CARD`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ORDERED_WITH_CREDIT_CARD` (
  `order_id` int(11) NOT NULL DEFAULT '0',
  `credit_card_id` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`credit_card_id`,`order_id`),
  KEY `lnk_ORDERED_WITH_CREDIT_CARD_ORDERS` (`order_id`),
```

```
  CONSTRAINT `lnk_ORDERED_WITH_CREDIT_CARD_CREDIT_CARDS` FOREIGN
KEY (`credit_card_id`) REFERENCES `CREDIT_CARDS` (`credit_card_id`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `lnk_ORDERED_WITH_CREDIT_CARD_ORDERS` FOREIGN KEY
(`order_id`) REFERENCES `ORDERS` (`order_id`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `ORDERS`
--

DROP TABLE IF EXISTS `ORDERS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ORDERS` (
  `order_id` int(11) NOT NULL AUTO_INCREMENT,
  `order_date` date NOT NULL,
  `customer_id` int(11) NOT NULL,
  `total_cost` int(11) NOT NULL,
  PRIMARY KEY (`order_id`),
  KEY `fk_orders_users1_idx` (`customer_id`),
  CONSTRAINT `fk_orders_users1` FOREIGN KEY (`customer_id`) REFERENCES
`USERS` (`user_id`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;



--
-- Table structure for table `ORDER_DETAILS`
--

DROP TABLE IF EXISTS `ORDER_DETAILS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ORDER_DETAILS` (
  `order_id` int(11) NOT NULL,
  `product_id` int(11) NOT NULL,
```

```
  `order_quantity` int(11) NOT NULL DEFAULT '1',
  `order_sum` int(11) NOT NULL,
  PRIMARY KEY (`order_id`,`product_id`),
  KEY `fk_order_details_products1_idx` (`product_id`),
  CONSTRAINT `fk_order_details_orders1` FOREIGN KEY (`order_id`) REFERENCES
`ORDERS` (`order_id`) ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT `fk_order_details_products1` FOREIGN KEY (`product_id`)
REFERENCES `PRODUCTS` (`product_id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `PRODUCTS`
--

DROP TABLE IF EXISTS `PRODUCTS`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `PRODUCTS` (
  `product_id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `description` varchar(512) DEFAULT NULL,
  `list_price` int(10) NOT NULL,
  `product_group` int(11) DEFAULT NULL,
  `available_quantity` int(11) DEFAULT NULL,
  `procurement_level` int(11) DEFAULT '0',
  `procurement_quantity` int(11) DEFAULT NULL,
  `procurement_level_reached` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`product_id`),
  KEY `fk_products_product_groups1_idx` (`product_group`),
  CONSTRAINT `fk_products_product_groups1` FOREIGN KEY (`product_group`)
REFERENCES `PRODUCT_GROUPS` (`product_group_id`) ON DELETE NO ACTION
ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=822 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `USERS`
--
```

```
CREATE TABLE `USERS` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `login_name` varchar(40) NOT NULL,
  `password` varchar(40) NOT NULL,
  `email` varchar(60) NOT NULL,
  `first_name` varchar(40) DEFAULT NULL,
  `last_name` varchar(40) DEFAULT NULL,
  `address_id` int(11) NOT NULL,
  `credit_limit` bigint(20) DEFAULT '0',
  `current_balance` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `email_UNIQUE` (`email`),
  UNIQUE KEY `login_name_UNIQUE` (`login_name`),
  KEY `fk_users_addresses1_idx` (`address_id`),
  CONSTRAINT `fk_users_addresses1` FOREIGN KEY (`address_id`) REFERENCES
`ADDRESSES` (`address_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=27 DEFAULT CHARSET=utf8;
```