**Contents:**

**Introduction---------------------------------------------------------------------------------------------------------------------- 3**

**ER diagrams and explanations ---------------------------------------------------------------------------------------------- 3**

**Solution of given topic (Comments)--------------------------------------------------------------------------------------- 5**

**Result (suggestions)----------------------------------------------------------------------------------------------------------- 11**

**References ---------------------------------------------------------------------------------------------------------------------- 11**

**Introduction:**

In modern life every industry should be digitalized. In retail sector it is essential topic. As it is known that supermarkets should handle huge amount of orders, sales and manage all data related to internal business processes. For this reason, several technical solutions are worked by companies. One of the common problems is to keep all information and details about products, orders , sales and etc.

This project solves problems of supermarket in case of data management and information retrievals. Multiple linked tables in storemanagement schema makes jobs done easily, increase efficiency and productivity.

**General Information:**

As a database management system(DBMS) MySQL was used. 8.0.13 version of MySQL. For the project, joins, views, select, update, delete, insert, constraints, filters, alter operations used in different kind of purposes. Firstly, I made a list of my needs and started to design necessary tables and their relationship statuses. As a result, I got ER diagram that will direct me during further processes.

**ER diagram:**

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

In the given picture(Figure 1) taken from database system(detailed information will be in next paragraphs) you can see all details about table structures and primary-foreign key relationships;

**Explanation:**

**Users table** describes all information about users . **Orders table** keeps recording of details of **Users’ order** to the Supplier demanding new products for selling. **Sales table** holds basic information about **Users’ sales**(payment, date and cashier identity). **Sales\_detailed** **table** is responsible for given detailed description about Sales amount, product etc. **Categories table** is for keeping information about category types of products in market such as dairy, fruits, bakery and so on. **Status\_sales** **table** describes status of a product such as active(saleable), frozen(is not allowed to sell), deleted(not in the system). **Product\_type** table holds types of product(piece, weighted and etc). **Status\_Orders table** describes current status of given orders. For example, Status of ordered product by market can be awaiting(waiting for delivery), canceled or received(by market reception). **Supplier table** holds all information about suppliers, contracts(start and end dates). Products table is the biggest and most related table of system. It combines **Categories**, **Supplier, Status\_sales, Product\_type** and additional product information. All ids are auto increment in tables.

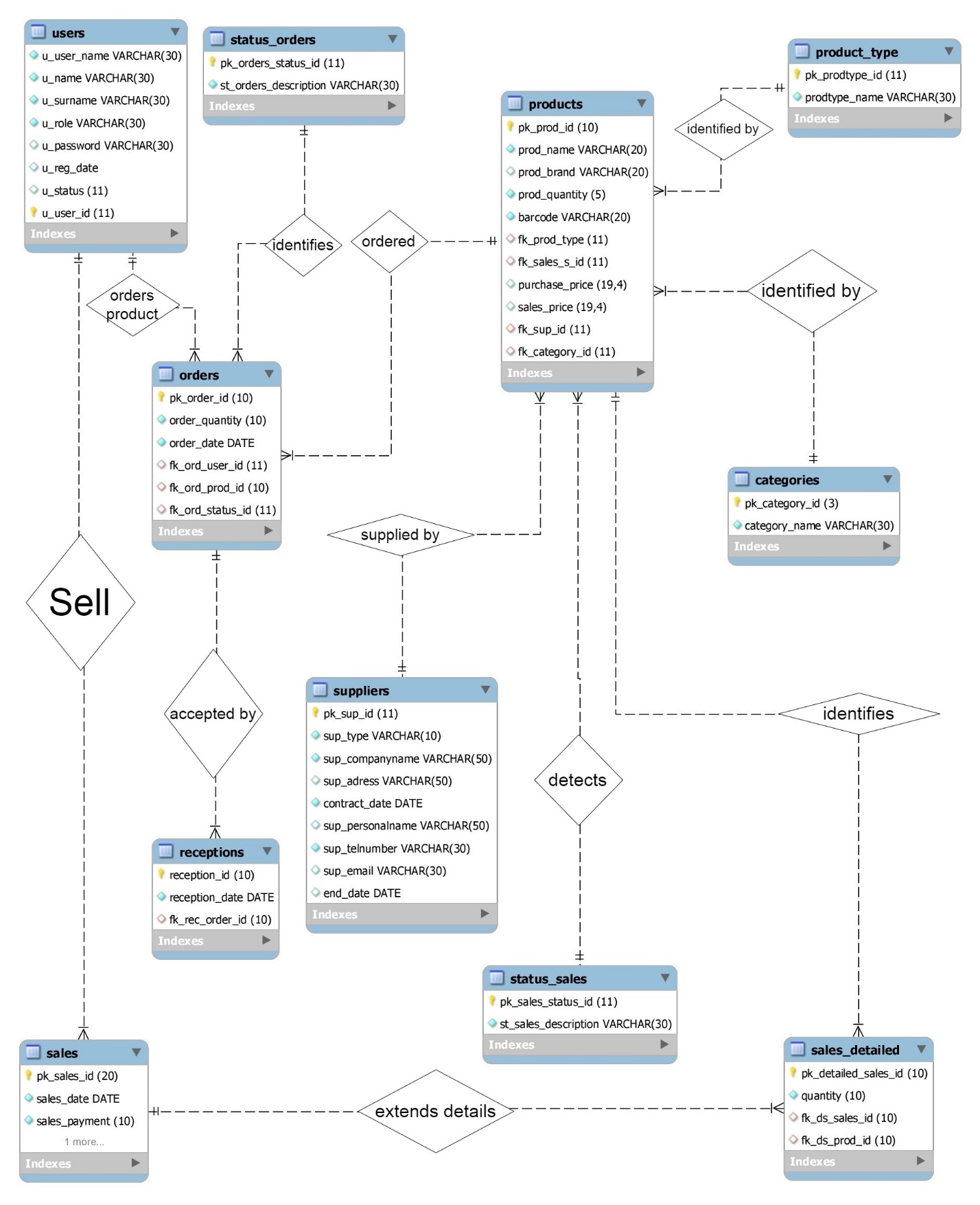


Figure 1

**Solution of given topic (Comments):**

After determining structure of system, I started to implement technical solutions. I used MySQL as DBMS and connected it with my C# WinForm Application. Most of the insert operations handled by desktop interface.   
In the following part creation and autoincerement setups will be given. Constraints such as primary,foreign key, not null, unique, default were used. For some tables I had to add additional columns during production. So, adding and modifying data type is also used.

**Create and autoincrement:**

*create database StoreManagement;*

*create table users(*

*u\_user\_name varchar(30) primary key,*

*u\_name varchar(30) not null,*

*u\_surname varchar(30) not null,*

*u\_role varchar(30) not null,*

*u\_password varchar(30) default '123456789',*

*u\_reg\_date datetime*

*);*

*alter table users add column u\_status int;*

*alter table users drop primary key;*

*alter table users add column u\_user\_id int primary key auto\_increment;*

*alter table users auto\_increment = 1000;*

*alter table users add constraint u\_user\_name unique(u\_user\_name);*

*create table CATEGORIES(*

*pk\_category\_id int(3) primary key auto\_increment,*

*category\_name varchar(30) not null*

*);*

*alter table CATEGORIES auto\_increment = 100;*

*create table PRODUCT\_TYPE(*

*pk\_prodtype\_id int primary key auto\_increment,*

*prodtype\_name varchar(30) not null*

*);*

*create table STATUS\_SALES(*

*pk\_sales\_status\_id int primary key auto\_increment,*

*st\_sales\_description varchar(30) not null*

*);*

*create table STATUS\_ORDERS(*

*pk\_orders\_status\_id int primary key auto\_increment,*

*st\_orders\_description varchar(30) not null*

*);*

*create table SUPPLIERS(*

*pk\_sup\_id int primary key auto\_increment,*

*sup\_type varchar(10) not null,*

*sup\_companyname varchar(50) not null,*

*sup\_adress varchar(50),*

*contract\_date date not null,*

*sup\_personalname varchar(50),*

*sup\_telnumber varchar(30) not null,*

*sup\_email varchar(30)*

*);*

*alter table SUPPLIERS auto\_increment = 10000000;*

*alter table storemanagement.suppliers add column end\_date date;*

*create table PRODUCTS(*

*pk\_prod\_id int unsigned primary key auto\_increment,*

*prod\_name varchar(20) not null,*

*prod\_brand varchar(20),*

*prod\_quantity smallint unsigned not null,*

*barcode varchar(20) unique not null,*

*fk\_prod\_type int,*

*fk\_sales\_s\_id int,*

*purchase\_price int not null check(purchase\_price >= 0),*

*sales\_price int not null check(sales\_price >= 0),*

*fk\_sup\_id int,*

*fk\_category\_id int,*

*foreign key (fk\_prod\_type)*

*references product\_type(pk\_prodtype\_id),*

*foreign key (fk\_sales\_s\_id)*

*references status\_sales(pk\_sales\_status\_id),*

*foreign key (fk\_sup\_id )*

*references suppliers(pk\_sup\_id),*

*foreign key (fk\_category\_id)*

*references categories(pk\_category\_id)*

*);*

*alter table products auto\_increment = 1000000000;*

*alter table storemanagement.products*

*modify purchase\_price decimal(19,4),*

*modify sales\_price decimal(19,4);*

*create table ORDERS(*

*pk\_order\_id int unsigned primary key auto\_increment,*

*order\_quantity int unsigned not null,*

*order\_date date not null,*

*fk\_ord\_user\_id int,*

*fk\_ord\_prod\_id int unsigned,*

*fk\_ord\_status\_id int,*

*foreign key (fk\_ord\_user\_id)*

*references users(u\_user\_id),*

*foreign key (fk\_ord\_prod\_id)*

*references products(pk\_prod\_id),*

*foreign key (fk\_ord\_status\_id)*

*references status\_orders(pk\_orders\_status\_id)*

*);*

*alter table ORDERS auto\_increment = 10000000;*

*create table RECEPTIONS(*

*reception\_id int unsigned primary key auto\_increment,*

*reception\_date date not null,*

*fk\_rec\_order\_id int unsigned,*

*foreign key (fk\_rec\_order\_id)*

*references orders(pk\_order\_id)*

*);*

*alter table receptions auto\_increment = 1000000000;*

*create table SALES(*

*pk\_sales\_id int(20) unsigned primary key auto\_increment,*

*sales\_date date not null,*

*sales\_payment int unsigned not null,*

*fk\_sales\_user\_id int,*

*foreign key (fk\_sales\_user\_id)*

*references users(u\_user\_id)*

*);*

*create table SALES\_DETAILED(*

*pk\_detailed\_sales\_id int unsigned primary key auto\_increment,*

*quantity int unsigned not null,*

*fk\_ds\_sales\_id int unsigned,*

*fk\_ds\_prod\_id int unsigned,*

*foreign key (fk\_ds\_sales\_id)*

*references sales(pk\_sales\_id),*

*foreign key (fk\_ds\_prod\_id)*

*references products(pk\_prod\_id));*

**Creating views**

//detailes about products

*create view view\_products as*

*select pk\_prod\_id "Product ID on system",prod\_name "Name", prod\_quantity "Quantity",*

*sales\_price "Sales price",*

*purchase\_price "Purchase price",*

*st\_sales\_description "Sales status",*

*prodtype\_name "Product type",*

*barcode "Barcode", prod\_brand "Brand name",*

*category\_name "Category name",*

*sup\_companyname "Supplier name"*

*from products*

*left join categories*

*on products.fk\_category\_id = categories.pk\_category\_id*

*left join suppliers*

*on products.fk\_sup\_id = suppliers.pk\_sup\_id*

*left join status\_sales*

*on status\_sales.pk\_sales\_status\_id = products.fk\_sales\_s\_id*

*left join product\_type*

*on products.fk\_prod\_type = product\_type.pk\_prodtype\_id;*

// Details about all received product

*create view view\_Reception as*

*select pk\_order\_id "OrderID", prod\_name "Product name",*

*order\_quantity "Ordered Quantity",order\_date "Ordered Date",*

*st\_orders\_description "Status of Order",*

*concat(u\_name,u\_surname) "Ordered by",*

*category\_name "Category name",*

*sup\_companyname "Supplier"*

*from orders*

*left join users*

*on orders.fk\_ord\_user\_id = users.u\_user\_id*

*left join products*

*on products.pk\_prod\_id = orders.fk\_ord\_prod\_id*

*left join status\_orders*

*on status\_orders.pk\_orders\_status\_id = orders.fk\_ord\_status\_id*

*left join categories*

*on categories.pk\_category\_id = products.fk\_category\_id*

*left join suppliers*

*on suppliers.pk\_sup\_id = products.fk\_sup\_id;*

//View for products that haven’t received yet

*create view view\_Reception\_active as*

*select pk\_order\_id "OrderID", prod\_name "Product name",*

*order\_quantity "Ordered Quantity",order\_date "Ordered Date",*

*st\_orders\_description "Status of Order",*

*concat(u\_name,u\_surname) "Ordered by",*

*category\_name "Category name",*

*sup\_companyname "Supplier"*

*from orders*

*left join users*

*on orders.fk\_ord\_user\_id = users.u\_user\_id*

*left join products*

*on products.pk\_prod\_id = orders.fk\_ord\_prod\_id*

*left join status\_orders*

*on status\_orders.pk\_orders\_status\_id = orders.fk\_ord\_status\_id*

*left join categories*

*on categories.pk\_category\_id = products.fk\_category\_id*

*left join suppliers*

*on suppliers.pk\_sup\_id = products.fk\_sup\_id*

*where orders.fk\_ord\_status\_id != 2;*

//view for received products

*create view view\_Received as*

*select pk\_order\_id "OrderID", prod\_name "Product name",*

*order\_quantity "Ordered Quantity",order\_date "Ordered Date",*

*st\_orders\_description "Status of Order",*

*concat(u\_name,u\_surname) "Ordered by",*

*category\_name "Category name",*

*sup\_companyname "Supplier"*

*from orders*

*left join users*

*on orders.fk\_ord\_user\_id = users.u\_user\_id*

*left join products*

*on products.pk\_prod\_id = orders.fk\_ord\_prod\_id*

*left join status\_orders*

*on status\_orders.pk\_orders\_status\_id = orders.fk\_ord\_status\_id*

*left join categories*

*on categories.pk\_category\_id = products.fk\_category\_id*

*left join suppliers*

*on suppliers.pk\_sup\_id = products.fk\_sup\_id*

*where orders.fk\_ord\_status\_id = 2;*

**Inserting data to tables**

*insert into users values( 'admin0','admin ad','admin soyad','admin','admin1234',sysdate());*

*insert into users values( 'cashier2','Seyiddd','Agayev','cashier','cashier1234',sysdate());*

*insert into categories(category\_name) values ('Bakery');*

*insert into product\_type(product\_type.prodtype\_name) values ('piece');*

*insert into product\_type(product\_type.prodtype\_name) values ('weighted');*

*insert into status\_sales(st\_sales\_description) values ('Active');*

*insert into status\_sales(st\_sales\_description) values ('Frozen');*

*insert into status\_sales(st\_sales\_description) values ('Deleted');*

*insert into status\_orders(st\_orders\_description) values ('Awaiting');*

*insert into status\_orders(st\_orders\_description) values ('Received');*

*insert into suppliers*

*values (10000000,'local','veyseloglu','baki',date(sysdate()),'Akif Akifli','+994555555555',null);*

*insert into storemanagement.products (prod\_name, prod\_brand, barcode, prod\_quantity,*

*fk\_prod\_type, purchase\_price, sales\_price, fk\_sup\_id,*

*fk\_category\_id,fk\_sales\_s\_id ) values ('Apple','Dayinin magazasi','55155155151',*

*'3','1','1','2','10000000','100',1) ;*

*insert into storemanagement.orders(order\_quantity,order\_date,*

*fk\_ord\_user\_id,fk\_ord\_prod\_id,fk\_ord\_status\_id)*

*values('2',date(sysdate()),'1006','1000000002',1);*

**Select command for multiple tasks**

//getting supplier id based on company name

*select pk\_sup\_id from storemanagement.suppliers where sup\_companyname = 'ARAZ';*

//getting category id based on category name

*select pk\_category\_id from storemanagement.categories where category\_name = 'BAKERY';*

//getting product type id based on type name

*select pk\_prodtype\_id from storemanagement.product\_type where prodtype\_name = 'piEce';*

// getting category, product type and company names from their own tables

*select upper(storemanagement.categories.category\_name) from storemanagement.categories;*

*select upper(storemanagement.product\_type.prodtype\_name) from storemanagement.product\_type;*

*select upper(storemanagement.suppliers.sup\_companyname) from storemanagement.suppliers;*

*//Data for describing in desktop application*

*select \* from storemanagement.categories order by category\_name;*

*//Suppliers that have active contract*

*select pk\_sup\_id from storemanagement.suppliers where sup\_companyname = 'ARAZ' and end\_date is null;*

*//Product and their status joined together*

*select \* from products*

*left join status\_sales*

*on status\_sales.pk\_sales\_status\_id = products.fk\_sales\_s\_id;*

*//getting sales status id base on status name*

*select pk\_sales\_status\_id from storemanagement.status\_sales where st\_sales\_description = 'active';*

*//view selections*

*select \* from view\_Reception;*

*select \* from view\_received;*

*select \* from view\_products;*

*select \* from view\_reception\_active;*

*//Select specified user*

*select u\_name as NAME,u\_surname as SURNAME, u\_user\_name as USERNAME, u\_role as ROLE, u\_reg\_date as REGISTIRATION\_DATE from storemanagement.users where true and u\_name ='Seyid' and u\_surname ='Agayev' and u\_user\_name ='cashier0' ;*

**Updates:**

//deactivating user from system

*update users set storemanagement.users.u\_status = 1 where users.u\_status is null;*

*//changing password of selected user*

*update storemanagement.users set u\_password = '123' where u\_user\_name = 'cas';*

*//ending contract*

*update storemanagement.suppliers set end\_date = date(sysdate()) where pk\_sup\_id = 10000000;*

*//changing status(active,frozen,…) of given product*

*update products set fk\_sales\_s\_id = '{0}' where pk\_prod\_id = '{1}';*

**Result:**

Database was suitable for C# program work without errors. Related tables protects data lose. It can be used as a prototype market management system for mini stores. Automation task, reporting and statistical reports can be easily generated.

**References:**

<https://dev.mysql.com/doc/>

<https://www.tutorialspoint.com/mysql/>

<http://www.mysqltutorial.org/mysql-views-tutorial.aspx>

<https://www.w3schools.com/sql/sql_join.asp>