

Foundations of Databases, A.Y. 2020/2021  
Master Degree in Computer Engineering  
Master Degree in ICT for Internet and Multimedia

Homework 2 – Conceptual and Logical Design

Deadline: November 27, 2020

Group FEDA	Project Sartoria FEDA	
Last Name Cecchinato	First Name Fabio	Student Number 1233602
Last Name Garbelotto	First Name Davide	Student Number 1234371
Last Name Grimaldi	First Name Alberto	Student Number 2026704
Last Name Lanza	First Name Enrico	Student Number 1232225

## Conceptual Design

### Variations to the Requirement Analysis

Following the suggestions proposed by the Professor, the following modifications to the Requirement Analysis have been made:

- Separation between orders related to clients and orders related to suppliers.
- Addition of an entity, "Product\_T", which specifies the type of product, to have a more fine-grained definition of it.
- The functional requirement "The customers can Monitor the phases of the process (sketch, paper pattern, prototype) to track the progress of their orders" has been changed in "The customers can monitor the phases of the process (pending, accepted, processing, delivering, completed and refunded) to track the progress of their orders".
- Products can have pictures associated with them.

### Entity-Relationship Schema

Figure 1 shows the ER schema.

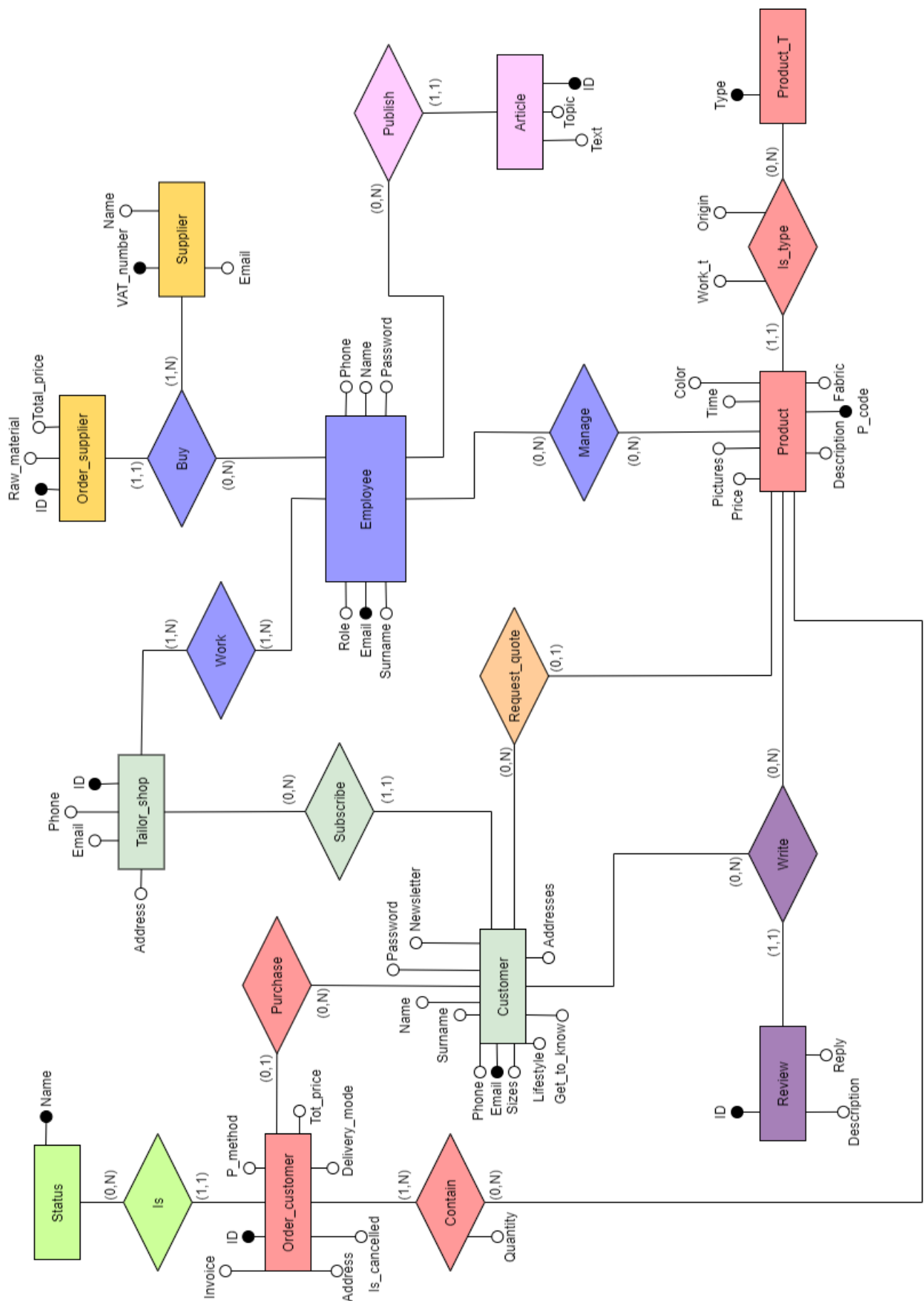


Figure 1: Entity-Relationship Schema

## Data Dictionary

### Entities Table

Entity	Description	Attributes	Identifier
Tailor_shop	General entity representing a shop	<ul style="list-style-type: none"><li>• <b>ID</b>: the identifier of the tailor's shop (serial)</li><li>• <b>Email</b>: the contact email of the shop (text)</li><li>• <b>Phone</b>: the contact phone (text)</li><li>• <b>Address</b>: the address of the shop (text)</li></ul>	<b>ID</b>
Customer	Who orders the product (client subscribed of the shop)	<ul style="list-style-type: none"><li>• <b>Email</b>: email of the customer (text)</li><li>• <b>Name</b>: name of the customer (text)</li><li>• <b>Surname</b>: surname of the customer (text)</li><li>• <b>Phone</b>: phone number of the customer (text)</li><li>• <b>Password</b>: password of the customer account (text)</li><li>• <b>Sizes</b>: sizes of the customer (text)</li><li>• <b>Lifestyle</b>: preferences of the customer (text)</li><li>• <b>Get_to_know</b>: how the customer got to know the tailor's shop (text)</li><li>• <b>Newsletter</b>: whether the customer is subscribed to the newsletter or not (boolean)</li><li>• <b>Addresses</b>: list of addresses for delivery (list of text)</li></ul>	<b>Email</b>
Employee	Tailor's shop worker	<ul style="list-style-type: none"><li>• <b>Name</b>: name of the employee (text)</li><li>• <b>Surname</b>: surname of the employee (text)</li><li>• <b>Phone</b>: phone number of the employee (text)</li><li>• <b>Email</b>: email of the employee (text)</li><li>• <b>Password</b>: password of the employee account (text)</li><li>• <b>Role</b>: role of the employee (enumeration between manager and simple employee)</li></ul>	<b>Email</b>
Supplier	Vendor of the raw material	<ul style="list-style-type: none"><li>• <b>VAT_number</b>: Value Added Tax identification number of the supplier (text)</li><li>• <b>Name</b>: name of the supplier (text)</li><li>• <b>Email</b>: email address of the supplier (text)</li></ul>	<b>VAT_number</b>
Order_supplier	Order from the tailor's shop to the supplier	<ul style="list-style-type: none"><li>• <b>ID</b>: unique identifier of the order (serial)</li><li>• <b>Raw_material</b>: raw material purchased from the suppliers (text)</li><li>• <b>Tot_price</b>: cost associated with the order (float)</li></ul>	<b>ID</b>
Product	Product of the tailor's shop	<ul style="list-style-type: none"><li>• <b>P_code</b>: identification code of the product (serial)</li><li>• <b>Pictures</b>: pictures that represents the product (image)</li><li>• <b>Time</b>: working hours to realize the final product (Int)</li><li>• <b>Description</b>: text explaining the product characteristics (realization, how to wash, ...)</li><li>• <b>Price</b>: sale price of the product (float)</li></ul>	<b>P_code</b>

		<ul style="list-style-type: none"> <li>• <b>Color:</b> colors of the product (text)</li> <li>• <b>Fabric:</b> materials to use for the product (text)</li> </ul>	
Product_T	Type of product	<ul style="list-style-type: none"> <li>• <b>Type:</b> typology of the product (enumeration between clothes and household good)</li> </ul>	<b>Type</b>
Order_customer	Order from the customer to the tailor's shop	<ul style="list-style-type: none"> <li>• <b>ID:</b> unique identifier of the order (serial)</li> <li>• <b>Tot_price:</b> cost associated with the order, that is the sum of the price of the products contained in the order (float)</li> <li>• <b>P_method:</b> how the customer has paid (enumeration between cash, credit/debit card, bank_transf, PayPal)</li> <li>• <b>Invoice:</b> invoice number associated to the order (text)</li> <li>• <b>Delivery_mode:</b> indicates whether the order has to be delivered at the physical shop or at the customer's home (enumeration between at the shop and at home)</li> <li>• <b>Address:</b> delivery address of the customer (text)</li> <li>• <b>Is_cancelled:</b> whether the order has been cancelled (boolean)</li> </ul>	<b>ID</b>
Review	Customer's evaluation of the product	<ul style="list-style-type: none"> <li>• <b>ID:</b> unique identifier for the review (serial)</li> <li>• <b>Description:</b> text of the review (text)</li> <li>• <b>Reply:</b> reply to the review by an employee (text)</li> </ul>	<b>ID</b>
Article	Post published on the blog by the company employees	<ul style="list-style-type: none"> <li>• <b>ID:</b> article identifier (serial)</li> <li>• <b>Text:</b> text of the article (text)</li> <li>• <b>Topic:</b> main topic of the article (text)</li> </ul>	<b>ID</b>
Status	Status of the customer's order	<ul style="list-style-type: none"> <li>• <b>Name:</b> name of the status (enumeration between pending, accepted, processing, delivering, completed and refunded)</li> </ul>	<b>Name</b>

#### Relationship Table

Relationship	Description	Component Entities	Attributes
Is_type	Allows to specify different types of products, i.e. clothes or household goods, and different types of work, i.e. from scratch or tailoring	<ul style="list-style-type: none"> <li>• Product (1,1)</li> <li>• Product_T (0,N)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Origin:</b> whether the idea for the product came from the tailor's shop or from the customer</li> <li>• <b>Work_t:</b> identifies the type of work to be done on the product (tailoring, from scratch)</li> </ul>
Purchase	The products purchased by a customer in a order	<ul style="list-style-type: none"> <li>• Order_customer (0,1)</li> <li>• Customer (0,N)</li> </ul>	

Contain	The order of the customer can contain several products	<ul style="list-style-type: none"> <li>Product (0,N)</li> <li>Order_customer (1,N)</li> </ul>	<ul style="list-style-type: none"> <li><b>Quantity:</b> the number of the same products in an order</li> </ul>
Request_quote	The customer requests a quote for the tailorings and for the products from scratch	<ul style="list-style-type: none"> <li>Customer (0,N)</li> <li>Product (0,1)</li> </ul>	
Subscribe	The customer can subscribe to the tailor's shop	<ul style="list-style-type: none"> <li>Customer (1,1)</li> <li>Tailor_shop (0,N)</li> </ul>	
Buy	The employee makes an order to the supplier when some raw material is lacking	<ul style="list-style-type: none"> <li>Supplier (1,N)</li> <li>Order_supplier (1,1)</li> <li>Employee (0,N)</li> </ul>	
Work	The employee works in the tailor's shop	<ul style="list-style-type: none"> <li>Employee (1,N)</li> <li>Tailor_shop (1,N)</li> </ul>	
Publish	The employee publishes articles on the online blog	<ul style="list-style-type: none"> <li>Employee (0,N)</li> <li>Article (1,1)</li> </ul>	
Write	The customer writes a review for a specific product	<ul style="list-style-type: none"> <li>Review (1,1)</li> <li>Customer (0,N)</li> <li>Product (0,N)</li> </ul>	
Manage	The employees who participates in the realization of one or more products	<ul style="list-style-type: none"> <li>Employee (0,N)</li> <li>Product (0,N)</li> </ul>	

## External Constraints

- **Chain of tailor's shops:** the system has been designed in a scalable way, such that if, in the future, the tailor's shop grows and opens other branches, they can be easily tracked inside the database. In fact, the "Tailor\_shop" entity is characterized by an ID, which univocally identifies a branch. Moreover, two different branches can have the same email address.
- **Customer Subscription:** the subscription to the system is mandatory for the online shoppers, but it's not required for the orders in the physical shop. The customer can also be registered in the system before requesting his/her first estimate or purchasing the first order. The customer can be associated with just one specific shop of the chain.
- **Physical shop customers:** in case the customer is not signed to the system, the order is still registered in the database, but it's not associated with any customer. The order can be delivered at the address saved as an attribute in "Order\_customer".
- **Employee account:** each employee has his/her own account, through which can perform all the basic functionalities of the system. Managers are also allowed to access the client info, to manage the accounts of the website and to give refunds.

- **Employees management:** when a new branch is opened, the system associates at least one employee with that shop, i.e. the responsible manager. Moreover, a certain manager can supervise the activities of more than one shop at a time.
- **Product manufacturing:** in order to better manage the realization of the products, the system registers which employee is working on each item. More employees can work on the same product at the same time, and each worker can participate in the realization of several products.
- **Product lifecycle:** a certain product exists in the database also if it has not been purchased yet, e.g. if the idea comes from the tailor who proposes the product on the online shop.

## Functional Requirements Satisfaction Check

### Three different types of users can login the system:

- **Customers:** each customer who wants to register to the tailor's shop can do it through the "Subscribe" relationship with the "Tailor\_Shop" entity. The customer information is stored inside the attributes of the "Customer" entity.
- **Employees:** each employee has his own account, through which he can sign in to the system, univocally identified by the "Email" attribute in the "Employee" entity.
- **Managers:** the managers have the same type of account of a basic employee, but the "Role" attribute specifies their role.

### The customers can:

- **Monitor the status of the order:** this information is stored inside the "Status" entity, linked to the "Order\_customer" through the relationship "Is".
- **View the products proposed by the tailor's shop:** these data can be found in the "Product" entity, by filtering for the sellable ones (using the "Origin" attribute). Indeed, only the products proposed by the tailor are added to the online store.
- **Order and purchase the existing products:** these operations can be done using the "Purchase" relationship.
- **Request an estimation:** this action can be done using the relationship "Request\_quote", only for the products that are not in the online store, because the present ones already have a price (i.e. the estimate is for the products proposed by the client).
- **Modify and/or cancel the order:** the modification of the order refers to the changing of some features of a specific product. In case the customer wants to add or remove products from his/her orders, this can be done through the cancellation of the old order and the purchase of a new one. The cancellation, which has to be approved by the employees, can be performed by setting to "True" the attribute "Is\_cancelled" of the entity "Order\_customer".
- **Review the service:** this operation can be done using the "Write" relationship.
- **View posts on the blog:** the posts are stored in the "Article" entity.
- **Request a customized service:** this operation can be done by using the "Purchase" relationship, and then assigning the right type to the products belonging to that order.

### The employees can:

- **View and publish posts on the blog:** publish an article on the blog through the "Publish" relationship.
- **Manage the process status of the orders, either changing the progress of an order or closing it:** order status information is stored in the "Status" entity, in particular in the

"Name" attribute. If an order is cancelled the information is stored in the "Order\_customer" attribute "Is\_cancelled".

- **Propose a quote of the order:** the employee can estimate the quote of an order requested by a customer through the "Request\_quote" relationship and set this amount as the price of the product.
- **Save and view orders made to the suppliers:** the employees can handle the orders stored as "Order\_supplier" instances.
- **Add and edit products on the online shop:** all the employees can add and modify products by storing the information of an item in the entity "Product".
- **Manage the newsletter. The tailor can send announcements and advertisements to the subscribed clients:** the customer can be subscribed to the newsletter and this information is stored in the "Newsletter" attribute of the "Customer" entity. In this way the employee can send the news only to those who are subscribed.
- **Reply to customer's reviews:** it's possible by modifying the "Reply" attribute in the "Review" entity.
- **Access to client's info:** it's possible by looking at the data stored in the "Customer" entity.

**Beyond all the employee's user rights, the managers are also allowed to:**

- **Access to client's info:** the manager is allowed to access the client data in the "Customer" entity.
- **Manage the accounts of the website:** the manager can handle all the accounts of the website, both the client's and the employees'.
- **Give refunds:** the manager can give refunds to the customer. The data about the amount to refund is stored in the attribute "Tot\_price" of the "Order\_customer" entity.

**If the customer is not logged in, he/she can still view the product, the blog and the comments.**

## Logical Design

### Transformation of the Entity-Relationship Schema

#### 1. Redundancy Analysis

- **Address in "Order\_customer" and "Customer":** the entity "Customer" has the attribute "Addresses" in order to save all the possible addresses. Instead in the "Order\_customer" entity the attribute "Address" contains the address of that specific order. The redundancy is kept because otherwise there would not be information about where to deliver an order if this comes from the physical shop.
- **Tot\_price:** the total price of the order can be obtained both from the "Tot\_price" attribute in the Order\_customer entity and by summing up the prices of all the products belonging to one order. The need of the "Tot\_price" attribute is therefore to be verified through the Load Analysis.

#### 2. Choice of Principal Identifiers

The schema does not contain external identification cycles and the main identifiers comply with the selection criteria.

## Analysis of Database Load

In this section, the analysis of the database load is performed to justify the presence of redundancies in the ER schema. The following operation involves the redundant attribute “Tot\_price”:

- O1: Visualize the total price and status of the order

This operation refers to the visualization of only status and total price of the order. If the client wants to view all the products contained inside the order and each of their costs, he/she has to perform a further operation.

The following table reports the description, the frequency and the type of this operation. It is an online operation since the customer needs to know on the fly the total amount of money of the order.

Operation	Description	Frequency	Type
O1: Visualize the total price and status of the order	The customer checks his/her order, e.g. tracks the status	5/day	Online

When the order is not purchased in the physical shop, but it's made online, the “Purchase” relationship and the “Customer” entity are involved too. In this kind of analysis, it makes sense to consider the worst case, i.e. when these two entities are included.

In the following tables, we present how often the involved entities and relationships are accessed in order to fulfil the operation O1. “Np” is the average number of products in one order.

### O1 with redundancy

Concept	Construct	Access	Type	Access Average
Order_customer	Entity	1	R	$1 \times 5 \times 1 = 5$
Is	Relationship	1	R	$1 \times 5 \times 1 = 5$
Status	Entity	1	R	$1 \times 5 \times 1 = 5$
Contains	Relationship	0		
Product	Entity	0		
Purchase	Relationship	1	R	$1 \times 5 \times 1 = 5$
Customer	Entity	1	R	$1 \times 5 \times 1 = 5$
Total Access				25



## O1 without redundancy

Concept	Construct	Access	Type	Access Average
Order_customer	Entity	1	R	$1 \times 5 \times 1 = 5$
Is	Relationship	1	R	$1 \times 5 \times 1 = 5$
Status	Entity	1	R	$1 \times 5 \times 1 = 5$
Contains	Relationship	1	R	$1 \times 5 \times 1 = 5$
Product	Entity	Np	R	$Np \times 5 \times 1 = 5Np$
Purchase	Relationship	1	R	$1 \times 5 \times 1 = 5$
Customer	Entity	1	R	$1 \times 5 \times 1 = 5$
<b>Total Access</b>				$30 + 5Np$

Independently from the value of “Np”, the number of accesses is greater in the case where there is no redundancy. The load analysis showed that it’s more convenient to keep the redundant attribute “Tot\_price”.

## Relational Schema

Figure 2 shows the relational schema.

To represent the relationship “Buy” between “Order\_supplier”, “Employee” and “Supplier”, the primary keys of these last two entities have been added as foreign keys on the order one. This decision is motivated by the fact that each order to the supplier is necessarily linked to specific suppliers and employees.

Instead, regarding the relationship “Request\_quote” between “Product” and “Customer”, an additional table has been added since not all the products are going to be associated with a quote. Indeed, by including the foreign key of “Customer” inside the “Product” relation, this attribute would be often a null value.

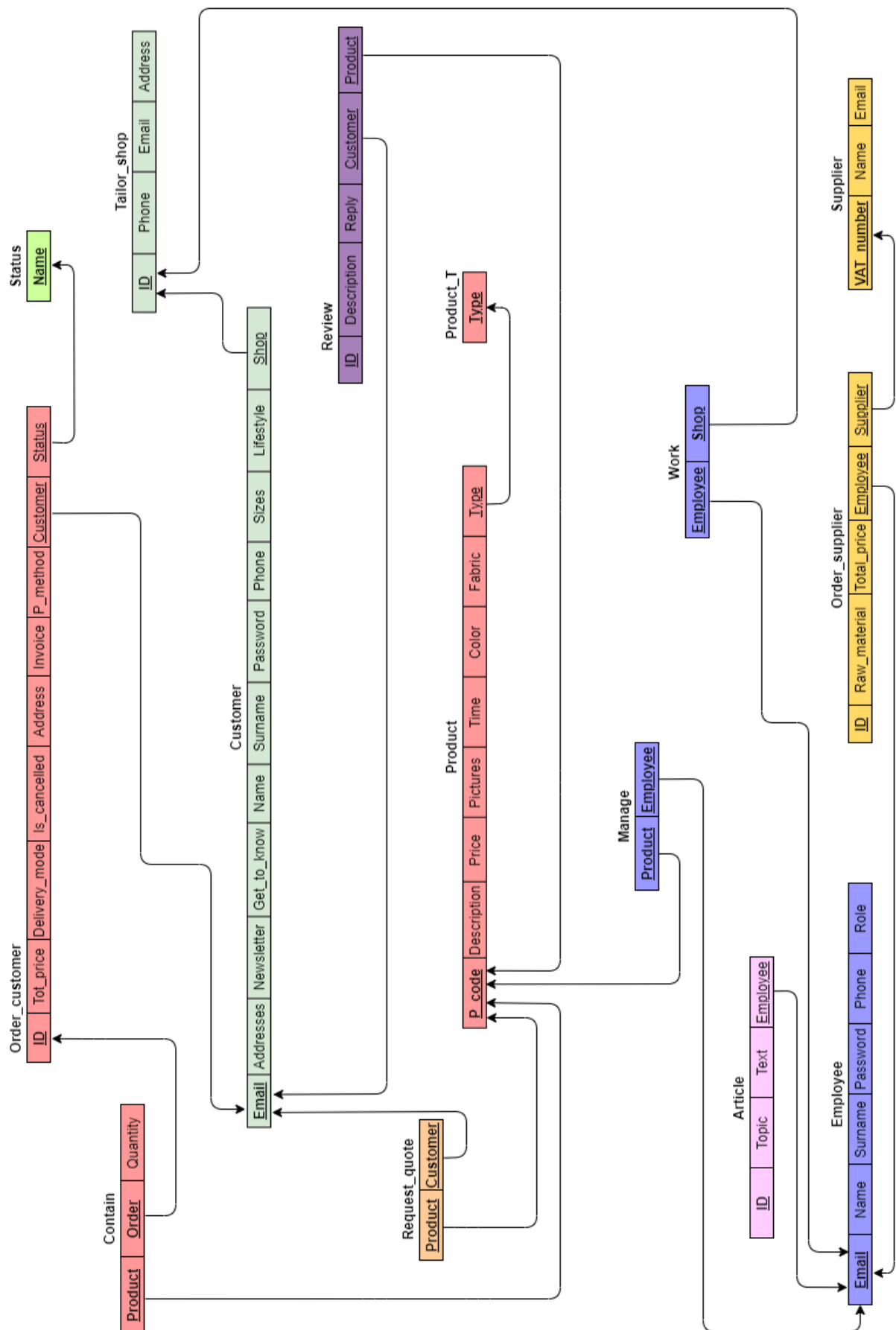


Figure 2: Relational Schema

## Data Dictionary

Relation	Attribute	Description	Domain	Constraints
Tailor_shop	ID	Identifier of the tailor's shop branch	Serial	Primary Key
	Email	Reference Email of the tailor's shop branch	Text	Not NULL
	Phone	Phone number of the tailor's shop branch	Text	Not NULL
	Address	Address of the tailor's shop branch	Text	Not NULL
Employee	Email	Email of the employee	Text	Primary Key
	Name	Name of the employee	Text	Not NULL
	Surname	Surname of the employee	Text	Not NULL
	Password	Password of the employee's account	Text	Not NULL
	Phone	Phone number of the employee	Text	Not NULL
	Role	Role of the employee	Enumeration	Enumeration {employee, manager}, not NULL
Customer	Email	Email address of the customer	Text	Primary key
	Name	Name of the customer	Text	Not NULL
	Surname	Surname of the customer	Text	Not NULL
	Password	Password of the customer account	Text	Not NULL
	Phone	Phone number of the customer	Text	

	Sizes	Sizes of the customer	Text	
	Lifestyle	Preferences of the customer	Text	
	Addresses	List of addresses for delivery	List of text	
	Newsletter	Whether the customer is subscribed to the newsletter or not	Boolean	0: no 1: yes not NULL
	Get_to_know	How the customer got to know about the tailor's shop	Text	
	Shop	Identifier of the shop where the customer is subscribed to	Serial	Foreign key to Taioir_shop
Order_customer	ID	Unique identifier of the order	Serial	Primary key
	Tot_price	Sum of the prices of the products contained in the order	Float	Not NULL
	Address	Address of the delivery	Text	
	P_method	How the customer paid (cash, credit card, ...)	Enumeration	Enumeration in {cash, credit/debit, bank_transf, PayPal}, not NULL
	Delivery_mode	Indicates whether the order has to be delivered at the physical shop or at the customer's home	Enumeration	Enumeration in {at home, in the shop}, not NULL
	Is_cancelled	Whether the order has been cancelled	Boolean	0: no 1: yes not NULL
	Invoice	Invoice number associated to the order	Text	Not NULL

	Customer	Reference to the customer who made this order	Text	Foreign key to Customer
	Status	Current status of the order	Text	Foreign key to Status
Product	P_code	Identification code of the product	Serial	Primary key
	Description	Text explaining characteristics (realization, how to wash, ...)	Text	Not NULL
	Price	Sale price of the product	Float	
	Pictures	Pictures that describe the product	Image	
	Time	Work hours to realize the product	Integer	
	Color	Color of the product	Text	
	Fabric	Material of the product	Text	
	Type	Type of the product	Enumeration	Foreign key to Product_T
Order_supplier	ID	Unique identifier of the order	Serial	Primary key
	Raw_material	Raw material purchased from the suppliers	Text	Not NULL
	Tot_price	Cost of the order	Float	Not NULL
	Employee	Reference to the employee who made the order	Text	Foreign key to Employee
	Supplier	Reference to the supplier	Text	Foreign key to Supplier
Status	Name	Name of the status	Enumeration	Primary key, Enumeration in {pending, accepted, processing, delivering, completed, refunded}

Article	ID	Article identifier	Serial	Primary key
	Topic	Main topic of the article	Text	
	Text	Text of the article	Text	Not NULL
	Employee	Email of the employee who uploaded the article	Text	Foreign key to Employee
Review	ID	Unique identifier for the review	Serial	Primary key
	Description	Actual review	Text	Not NULL
	Reply	Reply to the review made by an employee	Text	
	Customer	Customer who writes the review	Text	Foreign key to Customer
	Product	Product for which the review is written	Serial	Foreign key to Product
Product_T	Type	Typology of the product	Enumeration	Primary key, enumeration in {clothes, household good }
Supplier	VAT number	Value Added Tax identification number of the supplier	Text	Primary key
	Name	Name of the supplier	Text	Not NULL
	Email	Email address of the supplier	Text	Not NULL
Request_quote	Product	Product for which the quote is asked	Serial	Foreign key to Product
	Customer	Customer who requests the quote	Text	Foreign key to Customer
Work	Employee	Employee who works in a certain shop	Text	Foreign key to Employee

	Shop	ID of each tailor's shop	Serial	Foreign key to Tailor_shop
Manage	Product	The product ID	Serial	Foreign key to Product
	Employee	The email of employees involved in the realization of the product	Text	Foreign key to Employee
Contains	Order	The order ID	Serial	Foreign key to Order_customer
	Product	The ID of the products in each order	Serial	Foreign key to Product
	Quantity	The quantity of each product in an order	Int	Not NULL

## External Constraints

- **Chain of tailor's shops:** the "Tailor\_shop" entity is characterized by an ID, which is the primary key, by a phone number, an email and an address, which cannot assume the NULL value.
- **Customer subscription:** the "Customer", in the relational schema, is linked to a specific shop through the "Shop" attribute.
- **Physical shop customers:** when the attribute "Delivery\_mode" is set on the delivery at the shop, the "Address" attribute has to be NULL. The shop where the customer will pick up the order is obtained through the attribute "Shop" in the "Customer" relation.
- **Employee account:** each employee has his/her own account, through which is able to perform all the basic functionalities of the system. Managers are also allowed to access the client info, to manage the accounts of the website and to give refunds.
- **Employees management:** the "Work" relation connects each employee with the tailor's shops where he/she works, also in the case of managers who are linked to more than one shop.
- **Product manufacturing:** each product, represented by "Product" in the relational schema, is linked to the employees who are working on that product through the relationship "Manage", which contains the foreign key to the product and the foreign key to the employee.
- **Product lifecycle:** a certain product exists in the database also if it has not been purchased yet, e.g. if the idea comes from the tailor who proposes the product on the online shop.