

Foundations of Databases A.Y. 2021-2022
Homework 2 – Conceptual and Logical Design

Master Degree in Computer Engineering
Master Degree in Cybersecurity
Master Degree in ICT for Internet and Multimedia

Deadline: November 26, 2021

Team acronym	TAGMS	
Last Name	First Name	Student Number
Arslan	Tunahan	2023640
Esposito	Vittorio	2005795
Basso	Marco	2005796
Giuliani	Amedeo	2005797
Collado	Martin	2039907
Quiroz	Giannina	2041427
Zanini	Samuele	2019038

Conceptual Design

Variations to the Requirement Analysis

There are no relevant variations to Requirements Analysis.

An employee must necessarily have one and only one role. A role is typically associated with multiple employees but may not have any. For example, the "intern" role is created but without having hired anyone in that position yet.

An employee must work in at least one department. Each department can have multiple workers, but none if the department is new and newly created.

A manager can stipulate one or more contracts with many suppliers. A supplier can stipulate many contracts with the manager as well, but a new supplier can also not have stipulated anything yet.

Each contract specifies one or more items (e.g., ingredients, packaging materials, ...) and the respective quantity. An item can be specified in at least one contract, so it can be provided by many different suppliers. Each item belongs to only one item category, and to an item category can belong zero or many products.

A product, which is a finished good ready to be sold, is made up of one or more items (e.g., one glass bottle, a hundred grams of sugar, a hundred milliliters of water, ...) with the respective quantities. Consequently, an item can be utilized in many products (also none if, for example, the item is brand new). Each product belongs to one and only one product category, that is used to distinguish them. To a product category can belong zero or many products.

A package is composed by one or more packaging materials (i.e., an item used for packaging, such as a box, a meter of plastic tape, a kilogram of polystyrene, ...) with the respective quantities. Consequently, an item (e.g., packaging material in this case) can be utilized in many packages (also none if, for example, the item is brand new). Each package belongs to one and only one package category, that is used to distinguish them. To a package category can belong zero or many packages.

In a lot they can be stocked one or more products (i.e., only one type of product in a certain quantity) and one or more packages (i.e., only one type of package in a certain quantity). Each product can be stocked in many lots (in none if, for example, the product is brand new) and the same holds true for packages.

The customer decides with the seller regarding the products to be bought. The salesman, then, after communicating the products (with respective quantity) to the warehouse worker, will place the order only when all lots are ready. A seller can place zero or many orders for a customer, so a customer can make many order (none if, for example, the customer is new). An order can be place by only one salesman for only one customer. An order includes one or more lots. Each lot can be included by only one order (none if the lot is produced in advanced and waiting to be ordered). When the order is ready, a worker will ship it: a worker can ship zero or many orders, and an order can be shipped by at most one worker (none if the order is waiting to be shipped).

Entity-Relationship Schema

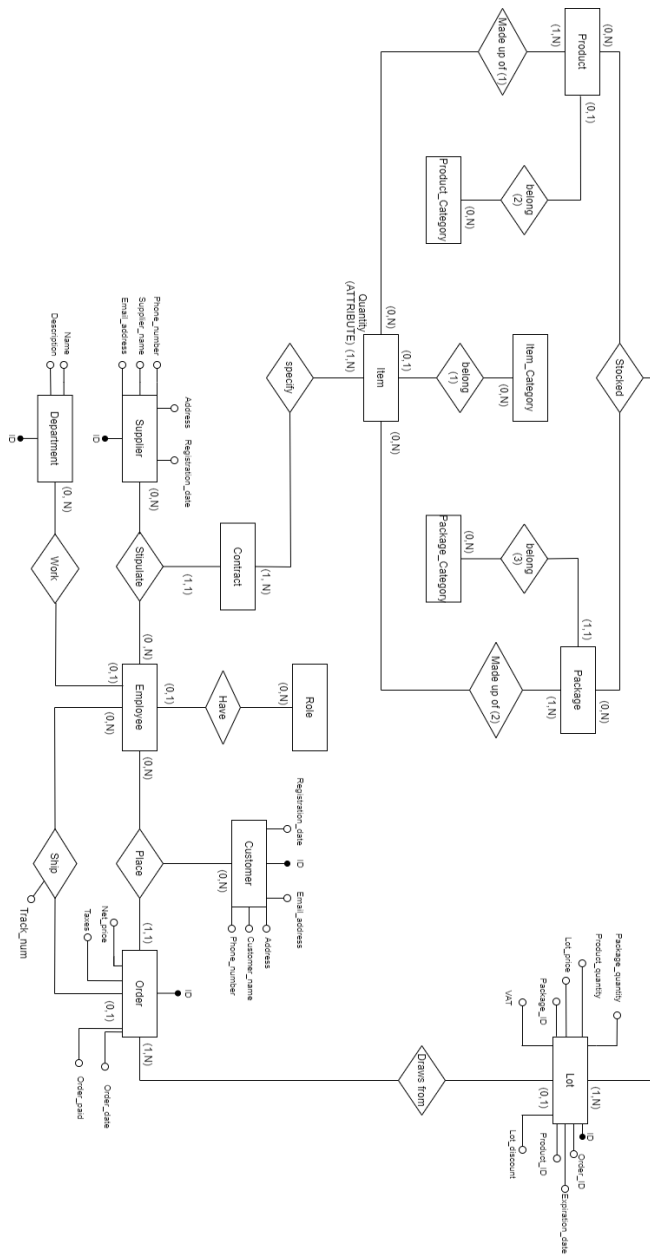


Figure 1: Entity-Relationship schema.

Data Dictionary

Entities Table

Entity	Description	Attributes	Identifier
Employee	Represents data of an employee who works in the company and needs access to the system	<ul style="list-style-type: none">• TAX_number: TAX code of the employee as badge ID, text• First_name: name of the employee, text• Last_name: surname of the employee, text• Phone_number: phone number of the employee, int• Email_address: email address of the employee, text• Gender: gender of the employee, boolean• Birth_date: birth date of the employee, datetime• Hiring_date: hiring date of the employee, text• Role_ID: role identifier, int• Department_ID: department identifier, int	TAX_number
Role	Represents data on the role of employees who work in the company	<ul style="list-style-type: none">• ID: role ID of the employee, serial• Name: name of the role, text• Description: technical description of the role, text	ID

Department	Represents data on the departments in which employees work	<ul style="list-style-type: none"> • ID: department ID of the company, serial • Name: name of the department, text • Description: description of the department's function, text 	ID
Customer	Represents data about a customer of the company	<ul style="list-style-type: none"> • ID: ID of the customer that can be the TAX code or the VAT number, text • Customer_name: name of the customer, text • Phone_number: phone number of the customer, int • Email_address: email address of the customer, text • Address: billing address of the customer, text • Registration_date: customer registration date in the database, datetime 	ID

Contract	Represents data about a contract stipulated between a supplier and a manager for the supply of items	<ul style="list-style-type: none"> • ID: contract ID, serial • Type: type of the contract, text • Description: description of the contract, text • Contract_date: date of signature of the contract with the supplier, datetime • Supplier_ID: VAT as supplier ID, int • Employee_ID: employee ID, text 	ID
Supplier	Represents data about a supplier of the company	<ul style="list-style-type: none"> • ID: VAT number as supplier ID, int • Supplier_name: name of the supplier company, text • Phone_number: phone number of the supplier company, int • Email_address: email address of the supplier company, text • Address: address of the supplier company, text • Registration_date: recording date of the supplier company, datetime 	ID

Order	Represents the order placed by a salesman for a customer	<ul style="list-style-type: none"> • ID: order ID, serial • Order_date: date in which the order has been processed, datetime • Order_paid: status of the payment, boolean • Net_price: total amount including discount(VAT excluded), float • Taxes: total amount of taxes to be payed, float 	ID
Lot	Represents the inventory of the company, containing products and items	<ul style="list-style-type: none"> • ID: lot ID, serial • Order_ID: order ID, serial • Product_ID: product ID, serial • Product_quantity: quantity of the produced product, int • Package_ID: package ID related to the specific lot, serial • Package_quantity: lot packaging quantity, int • Expiration_date: expiration date of the specific product, datetime • Lot_price: lot price without the discount, float • Lot_discount: total discount of a specific lot, int • VAT: value added tax at the time o sale, int 	ID

Product	Represents the final product that is marketed	<ul style="list-style-type: none"> • ID: product ID, serial • Name: name of the product, text • Energy: energy nutritional value, float • Fat: fat nutritional value, float • Saturated fatty acids: saturated fatty acids nutritional value, float • Carbohydrates: carbohydrates nutritional value, float • Sugar: sugar nutritional value, float • Protein: protein nutritional value, float • Salt: salt nutritional value, float • Net_weight: net weight of the product, float • Package_weight: gross weight of the product, float • Production_cost: cost of the production for a product, float • Inflation: price increase factor, float • Category_ID: category of the product, serial 	ID
---------	---	---	----

Item	Represents materials provided by suppliers from which the final products will be produced	<ul style="list-style-type: none"> • ID: ID of the item, int • Description: description of the item, text • Category_ID: ID of the category group, serial • Quantity: quantity of the specific item, int 	ID
Package	Represents packaging of finished products which are made up of boxes, tapes, and other packaging materials	<ul style="list-style-type: none"> • ID: ID of the package, serial • Description: description of the package, text • Weight: weight dimension of the package, int • Height: height dimension of the package, int • Width: width dimension of the package, int • Depth: depth dimension of the package, int • Package_Category_ID: ID of the package category, serial 	ID
Product_Category	Represents the category of a product, an item or a package	<ul style="list-style-type: none"> • ID: ID of the product category, int • Name: name of the product category, text • Description: description of the product category, text 	ID

Item_Category	Represents the category of a product, an item or a package	<ul style="list-style-type: none"> • ID: ID of the item category, int • Name: name of the item category, text • Description: description of the item category, text 	ID
Package_Category	Represents the category of a product, an item or a package	<ul style="list-style-type: none"> • ID: ID of the package category, int • Name: name of the package category, text • Description: description of the package category, text 	ID

Relationships Table

Relationship	Description	Component Entities	Attributes
Have	Relates each employee to a role	<ul style="list-style-type: none"> • Employee (0,1) • Role (0,N) 	None
Work	Assigns each employee to a department	<ul style="list-style-type: none"> • Employee (0,1) • Department (0,N) 	None
Stipulate	Links the supplier with the company and the contract stipulated	<ul style="list-style-type: none"> • Supplier (0,N) • Employee (0,N) • Contract (1,1) 	None

Place	Links the order made by the employee	<ul style="list-style-type: none"> • Employee (0,N) • Order (1,1) • Customer (0,N) 	None
Ships	Relates the employee shipping the order with the order itself and the shipment details	<ul style="list-style-type: none"> • Employee (0,N) • Order (0,1) 	Tracking number
Specify	Describes which items are provided by a contract	<ul style="list-style-type: none"> • Contract (1,N) • Item (1,N) 	None
Belongs to (1)	Links item to the category	<ul style="list-style-type: none"> • Item (0,1) • Item_Category (0,N) 	None
Belongs to (2)	Product item to the category	<ul style="list-style-type: none"> • Product (0,1) • Product_Category (0,N) 	None
Belongs to (3)	Package product to the category	<ul style="list-style-type: none"> • Package (0,1) • Package_Category (0,N) 	None
Made up of (1)	Describes which items are involved into creation to the product	<ul style="list-style-type: none"> • Item (0,N) • Product (1,N) 	None
Made up of (2)	Describes which items are involved into creation to the package	<ul style="list-style-type: none"> • Item (0,N) • Package (1,N) 	None

Stocked	Specifies the items and product stocked in the inventory	<ul style="list-style-type: none"> • Package (0,N) • Product (0,N) • Lot (1,N) 	None
Draws from	Associates the order and the lot	<ul style="list-style-type: none"> • Order (1,N) • Lot (0,1) 	None

External Constraints

- Employees can only insert and modify transactions in their particular department and role (i.e workers assigned in raw materials cannot add finished products transactions)
- Customers coordinate with Salesmen thus, only Salesmen can create Customers' profile, orders and payment and track status
- Employees can only ship orders once payment of Customer has been confirmed
- Products and Items belong in a specific category and must be added correctly by the Employees

Functional Requirements Satisfaction Check

The DBMS has to be able to:

- **store all the details of the employees, customers and suppliers in the organization:** Entities Employee and Role store data related to the employees. Entity Customer has details about the customers and entity Supplier has data related to the Supplier.
- **allow the employees to update their personal information:** Entity Employee has some attributes as Email_address, Password or Phone_number which can be changed. Employees can access the system using their credentials which are Email_address and Password and change this data.
- **store details of all on-hand products in the inventory such as item code, item description, quantity and expiration date:** The concept of inventory is implemented by the entities Item and Lot. Attributes ID_Product, Description, Expiration_Date, Product_Cost from entity Product and ID_Item, Description from entity Item show the data regarding items in the stock. The amount of each item is shown in attribute Item_Quantity of entity Item. Assembled products are organized in lots whose information is contained in Lot entity, in attributes ID_Lot, ID_Product (the product that lot is composed of), Quantity (the amount of elements of product the lot is composed of), Expiration_Date (common for all elements of the lot) and Order (the order the lot is assigned to, if NULL determines a not-yet-assigned lot).

- **allow the employees to log into the system and enter the inbound items they received with information item code, item description, quantity, expiration date and supplier:** With attributes Email_address and Password employees log in the database and insert this data in the entity Item, that could be inserting a new instance or updating an existing one.
- **show and generate the list of inbound and outbound transactions:** the inbound transactions can be derived from instances of the Contract entity, the outbound transactions can be derived from instances of the Order entity.
- **allow the employees to log into the system and enter the outbound transaction needed for the issuance of the products in the production and shipment to the customers:** Salesmen are responsible of entering outbound transactions as instances of the Order entity with proper attributes values, specifically Product_List, Quantities_List (has at index i the quantity of purchased items for product at index i of Product_List), Address and ID_Order which identifies the instance, Cost and Tax_Percentage. The system will calculate the not-yet-assigned lots that satisfy the requirements and collect their ID in the attribute Lots_List of Order (and also update Order attribute in Lot entity). Cost is a derived attribute, compute automatically given the composition of the lot and the product cost.
- **inventory stocks will be automatically updated whenever there are inbound and outbound transactions:** the update is executed automatically when an inbound transaction happens by inspecting each ID_Item x and associated Quantity Q_x in the relative Contract, then for each said x its Item_Quantity attribute in the entity Item is increased by Q_x (if x is not in the inventory?). For outbound transactions, since each ID_Product involved and its Quantity attribute value are extracted from the relative Order and the correspondent value is decreased in the relative Item instance.
- **show and generate the current inventory balance or stock inquiries:** The entity Item has data related to the quantity of each item stored. The entity Lot has data about the quantity of products each lot contains, and which lots are in the stock or not-shipped yet. This latter information can be retrieved checking if there exist a relationship "Ship" that involves the order a lot is associated to, since this relationship is created once the order is actually shipped to the customer. An employee can access the instances of these entity and specifically the attributes Item_quantity of Item and Product_Quantity of Lot to determine the exact current content of the stock.
- **receive and process the Customers order, specifying which products they want and respective quantity:** Salesmen are able to access the database and enter an instance of the Order entity reporting the desired list of products and the respective quantities. The system will assign appropriate lots if they already exist, and if they do not exist the Salesman will wait and check periodically. As soon as they are ready, the order will be entered and will be uniquely determined by ID_Order.
- **modification and cancellation of orders:** the salesman can change or cancel the order accessing it through the Attribute ID_Order of Entity Order. The order will be canceled after a certain period of time after the customer does not pay for the order.
- **allow users to view order and shipment status of finished products:** With the unique tracking number the attribute ID_Shipment of the Ship relationship, and the unique Id_order attribute of the Order entity the users can get information about the order and shipment.

- **generate invoice whenever payment has been made:** When a Salesman creates an order, it also sets to False a boolean attribute "Payed" that determined a not-payd-yet order. When the customer pays the order, the information about the payment is inserted as attributes of the Place relationship. The invoice document can be generated instantly extracting the information from the related instance of the Order entity. The subtotal is computed given the attributes Cost and Tax_Percentage.
- **permit transfer of items and products:** Entity Customer has an attribute Address which refers to where the ordered products are going to be shipped. When the customer pays the order and so the attribute Payed of the Order entity is set to True, the Order information is forwarded to the workers that access to the related Address, set up the shipment and create a Ship relationship with ID_Shipment
- **grant Cycle Counting in order to validate the accuracy of inventory:** Cycle counting is meant as a periodic check by the Managers on the items and products in the warehouse. After collecting the real values of Item_Quantity attribute for each item and the list of lots in the stock, the Manager updates manually the values of such attributes in the database to the correct value.
- **re-ordering the previous orders is allowed:** The system allows salesman to reorder orders. that means, the inventory system allows customers to save their orders the Attribute ID_Order of Entity Order and access them again through the salesman, then the Attribute Order_date will be updated by new order's date.
- **create tracking code for orders:** Attribute ID_Order of Entity Order store an unique identifier of each order. It is shown too in the relationship Ships between Order, Employee and Shipment.

Logical Design

Transformation of the Entity-Relationship Schema

Redundancy Analysis

Choice of Principal Identifiers

Analysis of Database Load

Relational Schema

Data Dictionary

Relation	Attribute	Description	Domain	Constraints
Relation 1				
Relation 1				

External Constraints

Group Members Contributions

Conceptual Design

- **Variations to the Requirement Analysis:**
- **Entity-Relationship Schema:**
- **Entities Table:**
- **Relationships Table:**
- **External Constraints:**
- **Functional Requirements Satisfaction Check:**

Logical Design

- **Redundancy Analysis:**
- **Choice of Principal Identifiers:**
- **Analysis of Database Load:**
- **Relational Schema:**
- **Data Dictionary:**
- **External Constraints:**