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
Basso	Marco	2005796
Cimarosto	Pietro	2027173
Collado	Martin	2039907
Esposito	Vittorio	2005795
Giuliani	Amedeo	2005797
Quiroz	Giannina	2041427
Zanini	Samuele	2019038

Conceptual Design

Variations to the Requirement 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 can be 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. Our system, therefore, is able to also store information relating to suppliers with whom the company has not yet entered into agreements.

Each contract specifies one or more items (e.g., ingredients, packaging materials, etc) 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 items. The stock quantity of each item is tracked. It will therefore be increased upon receipt of the items, after purchase from one or more suppliers, and decreased upon the preparation of lots of products that are made up of the items. The delivery date of the items is specified in each contract, so that the respective quantity in stock is automatically updated only upon arrival of the goods. The company does not use periodic delivery contracts, so every time it becomes necessary to purchase certain ingredients or packaging material, a new contract is signed with the supplier.

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, etc) with the respective quantities. For example, a product called "Coke J" can consist of one aluminum can, 50ml of water, 10g of sugar, etc. Another product, called "Coke B" for example, may have the same ingredients as the previous example but can be packaged with a glass bottle. The expiration date of a product is specified in the various batches (that include that particular product) and it may differ in each lot. The stock quantity of a product is not explicitly specified, but it can be obtained by checking the specified product quantities in each batch not yet sold or shipped. 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. The expiration date of the ingredients (items) is not tracked as the company guarantees to keep them stored for a short period of time because the ingredients are used shortly after their purchase and a FIFO policy is being implemented.

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, etc) with the respective quantities. For example, a package named "PK1" can consist of 4 boxes of dimensions 30cm x 30cm x 10cm, 2 meters of plastic tape and 200 g of polystyrene. A "PK2" package can consist of 6 boxes of dimensions 30cm x 30cm x 10cm, 4 meters of plastic tape and 300 g of polystyrene. An item (e.g., packaging material in this case) can be utilized in many packages (also none if, for example, the package is 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 there can be stocked many 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). The number of products in each batch depends both on the dimensions of the package of the individual product (e.g., bottle of glass) and on the features of the package (in particular, the size of the box and the number of boxes that make up the package). 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. Each lot is characterized by an expiration date. As some lots may be produced in advance

to reduce lead times, some of them may not sell on time and therefore expire. The data analyst will perform analysis in this regard to reduce waste.

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 the lots included in the order 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 placed 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). The invoice will be automatically generated by the application linked to the DBMS as soon as the order is placed. 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). The cancellation of the order is not accepted since the goods can be produced on commission and the company wants to minimize the waste caused by the expiry of the products. The customer, which is a business, must necessarily pay within 60 days and can be informed about the status of the order by contacting the seller.
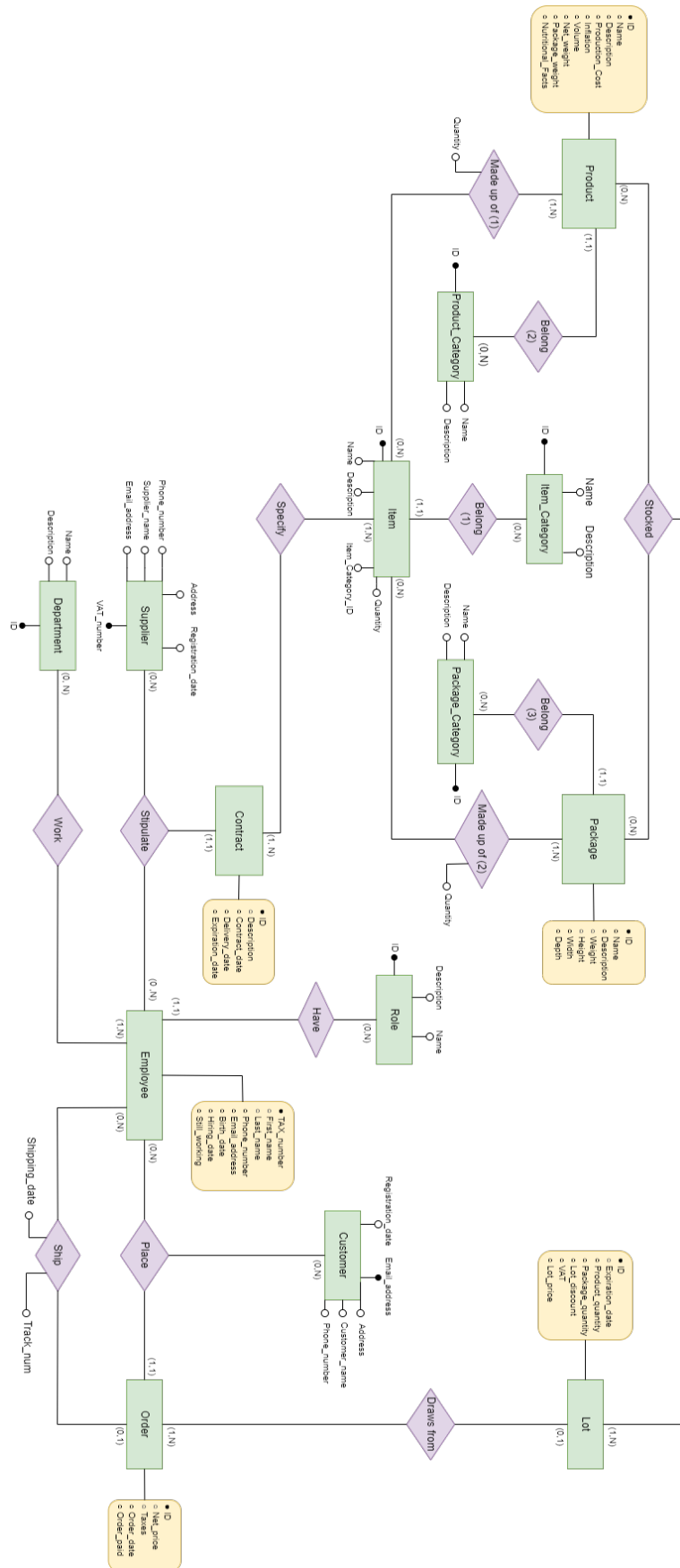
Entity-Relationship Schema

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	• TAX_number: TAX code of the employee, text • First_name: name of the employee, text • Last_name: surname of the employee, text • Phone_number: phone number (prefix included) of the employee, text • Email_address: email address of the employee, text • Birth_date: birthdate of the employee, datetime • Hiring_date: hiring date of the employee, datetime • Still_working: flag used to know if employee is still working for the company, boolean	TAX_number
Role	Represents data on the role of employees who work in the company	• ID: role identifier, 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	• ID: department identifier 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	• VAT_number: VAT number of the customer, int • Customer_name: name of the customer, text • Phone_number: phone number (prefix included) of the customer, text • Address: billing address of the customer, text • Registration_date: customer registration date in the database, datetime	VAT_number
Contract	Represents data about a contract stipulated between a supplier and a manager for the supply of items	• ID: contract identifier, serial • Description: description of the contract, text • Contract_date: date of signature of the contract with the supplier, datetime • Expiration_date: expiration date of the contract with the supplier, datetime	ID
Supplier	Represents data about a supplier of the company	• VAT_number: VAT number of the supplier company, int • Supplier_name: name of the supplier company, text • Phone_number: phone number (prefix included) of the supplier company, text • 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	VAT_number

Order	Represents the order placed by a salesman for a customer	- ID: order identifier, serial - Order_date: date in which the order has been processed, datetime - Order_paid: status of the payment, boolean - Net_price: total net 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	- ID: lot identifier, serial - Product_quantity: quantity of the included product, int - Package_quantity: lot packaging quantity, int - Expiration_date: expiration date of the included products, datetime - Lot_price: lot price (without the discount) at the time of sale, float - Lot_discount: percentage of discount of a lot, int - VAT: value added tax (percentage) at the time of sale, int	ID

Product	Represents the final product that is marketed	• ID: product identifier, serial • Name: name of the product, text • Description: description of the product, text • Nutritional_Facts: description of nutritional facts of the specific product, text • Volume: volume of the product in milliliters, int • Net_weight: net weight of the product in grams, int • Package_weight: package weight of the product in grams, int • Production_cost: cost of the production for a product, float • Inflation: price increase factor, float	ID
Item	Represents materials provided by suppliers from which the final products will be produced	• ID: identifier of the item, serial • Name: name of the item, text • Description: description of the item, text • Quantity: stock quantity of the item (automatically updates), int	ID

Package	Represents packaging of finished products which are made up of boxes, tapes, and other packaging materials	- ID: identifier of the package, serial - Name: name of the package, text - Description: description of the package, text - Weight: weight dimension of the package in grams, int - Height: height dimension of the package in centimeters, int - Width: width dimension of the package in centimeters, int - Depth: depth dimension of the package in centimeters, int	ID
Product_Category	Represents the category of a product, an item or a package	- ID: identifier of the product category, serial - Name: name of the product category, text - Description: description of the product category, text	ID
Item_Category	Represents the category of an item	- ID: identifier of the item category, serial - Name: name of the item category, text - Description: description of the item category, text	ID
Package_Category	Represents the category of a package	- ID: identifier of the package category, serial - 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	• Employee (1,1) • Role (0,N)	None
Work	Assigns each employee to a department	• Employee (1,N) • Department (0,N)	None
Stipulate	Links the supplier with the company and the contract stipulated	• Supplier (0,N) • Employee (0,N) • Contract (1,1)	None
Place	Links the order made by the employee	• 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	• Employee (0,N) • Order (0,1)	• Track_num • Shipping_date
Specify	Describes which items are provided by a contract	• Contract (1,N) • Item (1,N)	• Quantity
Belong (1)	Links items to the category	• Item (1,1) • Item_Category (0,N)	None
Belong (2)	Links products to the category	• Product (1,1) • Product_Category (0,N)	None
Belong (3)	Links packages to the category	• Package (1,1) • Package_Category (0,N)	None
Made up of (1)	Describes what items are involved in creating the product	• Item (0,N) • Product (1,N)	• Quantity

Made up of (2)	Describes what items are involved in creating the package	• Item (0,N) • Package (1,N)	• Quantity
Stocked	Specifies the products and packages stocked in the lots	• Package (0,N) • Product (0,N) • Lot (1,N)	None
Draws from	Associates the lots to an order	• Order (1,N) • Lot (0,1)	None

External Constraints

- The company decides the total discount to apply to a specific lot. This discount expresses a percentage and is a number between 0 and 100. Furthermore, the company is able to take into account changes in VAT.

- The units of measure used by the company are specified in detail in the entity table.

- Employees can only operate in their own department of competence with the role that belongs to them.

- Sellers create a profile for new customers. They can also keep track of the order status on their own, also updating the payment status once the bill is settled.

- Only the Manager can insert new contracts, so the Employee who takes part in the "stipulate" relationship must have the role equal to "Manager". Only the Salesman can insert new orders, so the Employee who takes part in the "place" relationship must have the role equal to "Seller". Only the Worker can ship orders, so the Employee who takes part in the "ship" relationship must have the role equal to "Worker".

- A product must consist of one or more items all having as Item_Category "ingredient" or "container". A package must consist of one or more items all having as Item_Category "box", "plastic tape", "polystyrene", or "other packaging material".

- The quantities of products contained in a lot belong to a finite set (e.g. 25, 50, 100), so the customer cannot order an arbitrary quantity of products.

- Only Items ordered and specified in the Contract for the month will be accepted during delivery. Items not ordered will be rejected 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:** Employee and Role entities store data related to the employees. Customer entity has details about the customers and Supplier entity has data related to the Supplier.

- **allow the employees to update their personal information:** Employee entity has some attributes as Email_address, Password or Phone_number that can be changed. Employees can access the system using their credentials 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, Product and Lot, whose attributes show the data regarding the stock. The amount of each item is shown in attribute Quantity of entity Item. Assembled products are organized in lots whose information is contained in Lot entity in the attributes ID, Product_ID (the product that lot is composed of), Product_quantity (the amount of elements of product the lot is composed of), Expiration_date (common for all elements of the lot) and Order_ID (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:** Employees can 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, that are instances of the Order entity with proper values for the attributes. The salesman will check for not-yet-assigned lots that satisfy the requirements and initialize a relationship between lot and order by updating the Order attribute in Lot entity. Net_price is a derived attribute, computed 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 $\mathbf{x}$ and associated Quantity Q_x in the relative Contract, then for each said $\mathbf{x}$ its Item_Quantity attribute in the entity Item is increased by Q_x. For outbound transactions, lots with NULL value in the order attribute are in the stock and unassigned, and lots associated to an order that is not in a Ship relationship are the ones that are waiting in the stock to be shipped. Therefore, these two sources of information allow to determine the lots in the stock at every time.

- **show and generate the current inventory balance or stock inquiries:** The Item entity has data related to the quantity of each item stored. The Lot entity 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 assigned to, since this relationship is created only once the order is actually shipped to the customer. An employee can access the instances of these entity and specifically the attributes Quantity of Item entity and Product_quantity of Lot entity (eventually, of all lots containing a certain product) 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 Salesman will find and assign appropriate lots if they already exist, and if they do not exist it will notify the Production department which will notify back when they are ready. As soon as this happens, the order will be entered and will be uniquely determined by ID.

- **modification and cancellation of orders:** the salesman can change or cancel the order accessing it through the Attribute ID of Order entity.

- **allow users to view order and shipment status of finished products:** With the unique tracking number the attribute "Tracking number" of the Ship relationship, and the unique ID 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 sets automatically to False a boolean attribute Order_paid that determines a not-payed-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. Final_Price is computed given the attributes Net_Price and Taxes.

- **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 Order_paid of the Order entity is set to True, the Order information is forwarded to the workers that inspect the related Address, set up the shipment and create a Ship relationship with Tracking number

- **grant Cycle Counting in order to validate the accuracy of inventory:** Cycle counting is define as a periodic check by the Managers or by a Worker Supervisor on the items and products in the warehouse. After acquiring the actual quantities for each item and the list of lots in the stock, it will be reconciled with the system quantities. The Manager can update (if necessary) the values of such attributes in the database to the correct value.

- **re-ordering the previous orders is allowed:** The system allows salesmen to access past orders using the ID attribute and retrieve information about the lots, the products, their quantities and all the necessary data to set up a new order with the same content.

- **create tracking code for orders:** Attribute Tracking number of Ship relationship stores an identifier, provided by the third-party company responsible for the shipping, that uniquely identifies the shipment.

Logical Design

Transformation of the Entity-Relationship Schema

Redundancy Analysis

Choice of Principal Identifiers

Analysis of Database Load

The load analysis is divided in two parts: the first, regarding whether to use "inflation" and "production_cost" and calculating the final product price on-demand; the second, to decide whether to store "lot_price" into the "Lot" entity, or computing it when necessary via the relationship "Stocked".

Operation	Description	Frequency	Type
O_1 : Insert new lot	Store data about a newly packaged lot.	25/week	Online
O_2 : Compute order price			Online

Table 4: Without redundancy

Concept	Construct	Access	Type	Average Access
Product	Entity	1	R	1 x 25 x 1 = 25
Lot	Entity	1	W	1 x 25 x 2 = 25
Stocked	Relationship	1	W	1 x 25 x 2 = 25
Total Access				**75**

read product (compute price), write lot, write stocked

Relational Schema

Data Dictionary

Table 5: With redundancy

Concept	Construct	Access	Type	Average Access
Product	Entity	1	R	1 x 25 x 1 = 25
Lot	Entity	1	W	1 x 25 x 2 = 25
Stocked	Relationship	1	W	1 x 25 x 2 = 25
Total Access			**75**	

Relation	Attribute	Description	Domain	Constraints
Product	ID	Identifier of a product	Serial	PRIMARY KEY
	Name	Name of a product	Text	NOT NULL
	Description	Description of a product	Text	NOT NULL
	Production_Cost	Cost of producing a product	Float	NOT NULL
	Inflation	Price increase factor of a product	Float	NOT NULL
	Volume	Volume of a product in milliliters	Int	NOT NULL
	Net_Weight	Net weight of a the product in grams	Int	NOT NULL
	Package_weight	Package weight of a the product in grams	Int	NOT NULL
	Nutritional Facts	Description of nutritional facts of a product	Text	NOT NULL
	Product_Category_ID	Identifier of a Product Category	Serial	Foreign Key that refers to ID of Product_Category
Product_Category	ID	Identifier of a Product Category	Serial	PRIMARY KEY
	Name	Name of a Product Category	Text	NOT NULL
	Description	Description of a Product Category	Text	NOT NULL
Item	ID	Identifier of an item	Serial	PRIMARY KEY
	Name	Name of an item	Text	NOT NULL
	Description	Description of an item	Text	NOT NULL
	Quantity	Stock quantity of an item	Int	NOT NULL
	Item_Category_ID	Identifier of an Item Category	Serial	Foreign Key that refers to ID of Item_Category
Item_Category	ID	Identifier of an Item Category	Serial	PRIMARY KEY
	Name	Name of an Item Category	Text	NOT NULL
	Description	Description of an Item Category	Text	NOT NULL
Package	ID	Identifier of a package	Serial	PRIMARY KEY
	Name	Name of a package	Text	NOT NULL
	Description	Description of a package	Text	NOT NULL
	Weight	Weight of a package in grams	Int	NOT NULL
	Height	Height of a package in centimeters	Int	NOT NULL
	Width	Width of a package in centimeters	Int	NOT NULL
	Depth	Depth of a package in centimeters	Int	NOT NULL
	Package_Category_ID	Identifier of a Package Category	Serial	Foreign Key that refers to ID of Package_Category

Package_Category	ID	Identifier of a Package Category	Serial	PRIMARY KEY
	Name	Name of a Package Category	Text	NOT NULL
	Description	Description of a Package Category	Text	NOT NULL
Lot	ID	Identifier of a lot	Serial	PRIMARY KEY
	Expiration_date	Expiration date of the included products	Datetime	NOT NULL
	Product_quantity	Amount of a product in each lot	Int	NOT NULL
	Package_quantity	Amount of a package in each lot	Int	NOT NULL
	Lot_Discount	Percentage of discount of a lot	Int	NOT NULL
	VAT	Value added tax (percentage) at the time of sale	Int	NOT NULL
	Lot_price	Lot price without discount	Float	NOT NULL
	Order_ID	Identifier of an Order of the order payment	Serial	Foreign Key that refers to ID of Order
Order	ID	Identifier of an order	Serial	PRIMARY KEY
	Net_price	Total net amount including discount (VAT excluded)	Float	NOT NULL
	Taxes	Amount of taxes to be paid	Float	NOT NULL
	Order_date	Date in which the order has been processed	Datetime	NOT NULL
	Order_paid	Status of the order payment	Boolean	NOT NULL
	Employee_ID	identifier of the employee that places the order	Text	Foreign key that refers to VAT_Number
Made up of (1)	Quantity	Amount of each item in a product	Int	
	ID	Identifier of a product	Serial	
	ID	Identifier of an item	Serial	
Made up of (2)	Quantity	Amount of each item in a package	Int	
	ID	Identifier of a package	Serial	
	ID	Identifier of an item	Serial	
Ship	ID	Identifier of a lot	Serial	
	ID	Identifier of an order	Serial	
	Shipping_date	Identifier of an order	Datetime	
	Track_num	Identifier of an order	Int	
Employee	TAX_number	TAX code of the employee	Text	PRIMARY KEY
	First_name	name of the employee	Text	NOT NULL
	Last_name	surname of the employee	Text	NOT NULL
	Phone_number	phone number (prefix included) of the employee	Text	NOT NULL
	Email_address	email address of the employee	Text	NOT NULL
	Birth_date	birthdate of the employee	Datetime	NOT NULL
	Hiring_date	hiring date of the employee	Datetime	NOT NULL
	Still_working	flag used to know if employee is still working for the company	Boolean	NOT NULL
	Role_ID	Identifier of the role that an employee has	Serial	Foreign Key that refers to Id of Role
Role	ID	role identifier	Serial	PRIMARY KEY
	Name	name of the role	Text	NOT NULL
	Description	technical description of the role	Text	NOT NULL
Department	ID	department identifier of the company	Serial	PRIMARY KEY

	Name	name of the department	Text	NOT NULL
	Description	description of the department's function	Text	NOT NULL
Supplier	VAT_number	VAT number of the supplier company	Int	PRIMARY KEY
	Supplier_name	name of the supplier company	Text	NOT NULL
	Phone_number	phone number (prefix included) of the supplier company	Text	NOT NULL
	Email_address	email address of the supplier company	Text	NOT NULL
	Address	address of the supplier company	Text	NOT NULL
	Registration_date	recording date of the supplier company	Datetime	NOT NULL

External Constraints

Group Members Contributions

Conceptual Design

- **Variations to the Requirement Analysis:** Esposito

- **Entity-Relationship Schema:** Esposito, Zanini, Collado, Giuliani

- **Entities Table:** Esposito,

- **Relationships Table:** Zanini, Giuliani

- **External Constraints:** Quiroz, Collado

- **Functional Requirements Satisfaction Check:** Cimarosto, Collado, Arslan

Logical Design

- **Redundancy Analysis:**

- **Choice of Principal Identifiers:**

- **Analysis of Database Load: Giuliani**

- **Relational Schema:**

- **Data Dictionary:** Collado, Arslan

- **External Constraints:**