P.PORTO ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Disciplina de Bases de Dados

Ano Letivo de 2023/2024

Greenhouse Management

Rodrigo Alves

André Ribeiro

Junho, 2024

Data de Receção	
Responsável	
Avaliação	
Observações	

Greenhouse Management

Rodrigo Alves

André Ribeiro

Junho, 2024

Acknowledgments

I would like to express my sincere gratitude to Professor Vasco Santos for guiding us through the course. Your dedication to teaching and your passion for the subject matter have been crucial in shaping our understanding of database systems. Your guidance has been invaluable, and I am truly grateful for the opportunity to learn under your mentorship.

Summary

This report describes the creation of a comprehensive database system for greenhouse management. The system aims to automate and optimize the recording and querying of cultivation and distribution data for plants, vegetables, and fruits. The project's objectives included developing a reliable data structure for recording planting and harvesting activities, as well as auxiliary product consumption. The methodology involved creating conceptual, logical, and physical models, and implementing integrity constraints to maintain data quality. The database system supports various user transactions, providing insights into harvested products, planting quantities, and section-specific distributions. It enables efficient querying and data analysis, facilitating better decision-making in greenhouse operations. The project faced challenges related to managing complex relationships and ensuring thorough data validation, which were addressed through careful design and constraint implementation. The developed database system significantly enhances greenhouse management by providing a structured and efficient means of data handling. It establishes a strong foundation for future enhancements, such as advanced analytics and reporting capabilities, to further support greenhouse operational efficiency.

Index

Figure Index

Table Index

1. Introdução

1.1 Contextualization

The manager of a greenhouse company wants to streamline all registration and information retrieval processes by creating a database. The main goal is to record information about the plants, vegetables, and fruits managed and distributed by the company using an information system. The following basic specifications are considered:

- Greenhouses can be divided into different sections (Aromatic Plants, Vegetables, and Fruits) to facilitate monitoring of product growth for marketing. Each section can hold a maximum of 10 different products.
- Information about planting and harvesting of products needs to be documented.
- It's important to track the quantity of seeds used for each planting and the use of other products that promote growth (e.g., fertilizer, water, pesticides, etc.).
- Once a product's harvest is complete, this information should be stored

.1.2 Presentation of the Case Study

The management of greenhouses involves various processes, such as planting, harvesting, and monitoring plant growth. To streamline these processes and improve efficiency, the company aims to develop an information system that can handle all aspects of greenhouse management. The proposed system will allow for the recording of crucial data related to plantations, harvests, and auxiliary product consumption. Through this system, the company seeks to enhance its operational capabilities and make informed decisions based on accurate and up-to-date information.

1.3 Motivation and Objectives

The company is developing a greenhouse management system to modernize and streamline its operations. The system aims to improve data accuracy, enhance operational efficiency, facilitate decision-making, and ensure compliance with regulatory requirements and industry standards. The specific objectives of the project include designing and implementing a relational database, developing user-friendly interfaces, implementing querying and reporting functionalities, and testing the system for reliability and performance.

1.4 Report Structure

The report is organized according to the following structure:

Introduction: This section provides an overview of greenhouse management and outlines the report's objectives.

Conceptual Model: This part identifies various types of relationships and documents the attributes, domains, and keys of the entities involved in greenhouse management.

Logical Model: It establishes relations for the logical data model and validates them using normalization techniques. The relations are further verified through user transactions.

Physical Model: This model includes the design of base relations for the database tables, such as Greenhouse, Section, Product, AuxiliaryProduct, Plantation, Harvest, and AuxiliaryConsumption.

Conclusion: This section summarizes the findings of the project and highlights key insights into greenhouse management.

Acknowledgments: Recognition is given to individuals who contributed to the completion of the project.

References: A list of sources referenced in the report is provided for further reading and validation.

Next, we will delve into the “Addressing the Issue” section, where we will discuss the conceptual and logical models, integrity constraints, physical model, and design considerations in more detail.

2. Addressing the issue

To effectively address the challenges of greenhouse management, we use a systematic approach that includes conceptual modeling, logical modeling, and physical modeling.

Conceptual Model

We start by creating a conceptual model that depicts the key entities and their relationships in greenhouse management. This model offers a high-level view of the system's structure and functionality, laying the foundation for the subsequent stages of database design.

Logical Model

Guided by the conceptual model, we move on to the logical modeling phase, where we establish relationships based on the identified entities. The logical model presents the database schema in a structured format, ready for implementation in a relational database management system (RDBMS). We validate the relationships using normalization techniques to ensure data integrity and reduce redundancy.

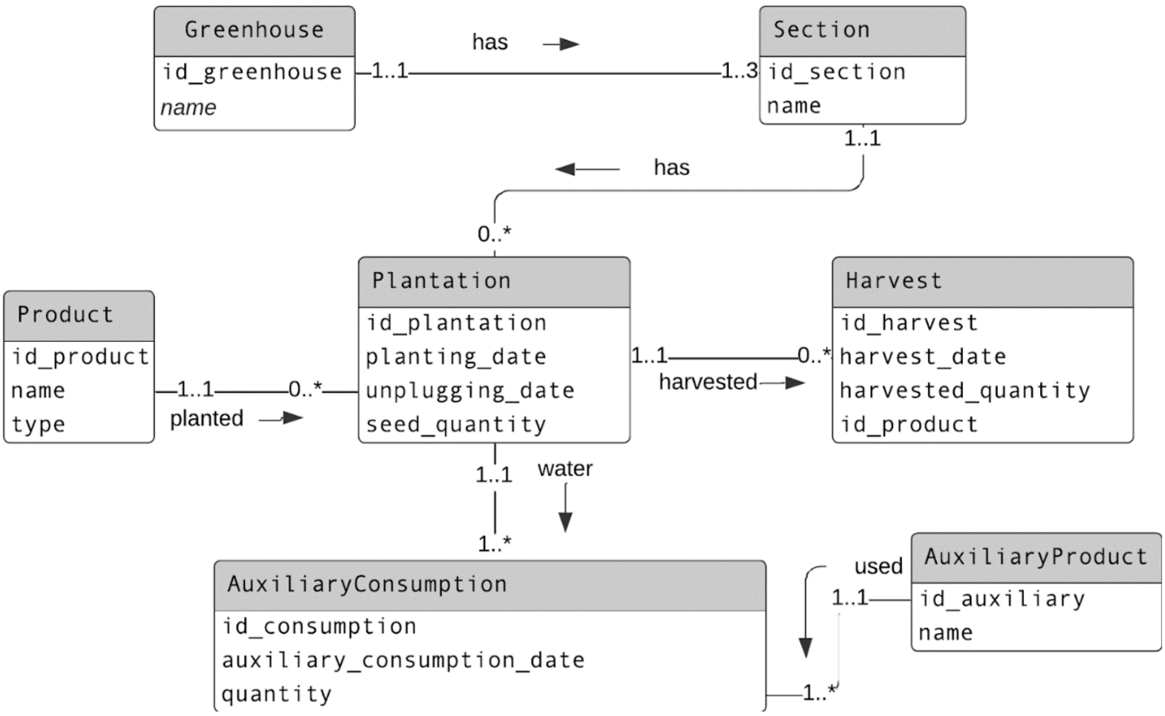
Physical Model

In the final phase of our approach, we develop the physical model by transforming the logical model into base relations. Each base relation corresponds to a table in the database, with defined attributes and constraints. We also specify general constraints to enforce data integrity and business rules. Additionally, we create views to offer customized perspectives of the data to different users.

By following this systematic approach, our goal is to build a strong and scalable database system that meets the specific requirements of greenhouse management. This approach ensures that the database design is well-structured, efficient, and capable of supporting the diverse needs of greenhouse operations.

3. Conceptual Model

Figure 1 – conceptual model



3.1 Identify Relationship Types

In the greenhouse management system, understanding the types of relationships is essential for comprehending how different entities interact with each other. The primary entities for this system are Product, Section, Plantation, Harvest, Auxiliary Product, and Auxiliary Consumption, and their relationships define the structure and behavior of the system.

According to Connolly, T., & Begg, C. (2021). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed., p. 510-511).[Pearson].
"The next step is to identify relationships between entities in the user requirements specification. We can look for nouns to identify entities and use the grammar to find relationships indicated by verbs or verbal expressions.

Here are the main types of relationships identified:

1. Greenhouse-Section Relationship:
   Each greenhouse can be associated with up to 3 sections, with each being of a different type (herbs, vegetables, or fruits). In each greenhouse, there cannot be repeated sections for different products simultaneously. This forms a many-to-many relationship.

1. Section-Plantation Relationship:
   Each section can host multiple plantations, but each plantation is associated with only one section. However, a section can host a maximum of up to 10 plantations of different products. This forms a one-to-many relationship.

2. Product-Plantation Relationship:
   Each product can be involved in multiple plantations, but each plantation involves only one product. This forms a one-to-many relationship.

3. Plantation-Harvest Relationship:
   Each plantation can result in multiple harvests, but each harvest is associated with only one plantation. This forms a one-to-many relationship.

4. AuxiliaryProduct-AuxiliaryConsumption Relationship:
   Each auxiliary product can be involved in multiple auxiliary consumptions, but each auxiliary consumption involves only one auxiliary product. This forms a one-to-many relationship.

5. Plantation-Auxiliary Consumption Relationship:
   Each plantation can involve multiple auxiliary consumptions, but each auxiliary consumption is associated with only one plantation. This forms a one-to-many relationship.

3.2 Table Identify Relationship Types

Entity Name	Description	Aliases	Occurrence
Greenhouse	Represents a greenhouse where the plants are grown	Represents a greenhouse where the plants are grown	The greenhouse contains up to 3 sections
Section	Represents a section within a greenhouse. Each section belongs to a single greenhouse	Herbs vegetables fruit	The section has up to 10 plantations
Product	Represents the products that are planted and harvested within the system	Crop, yield, produce	Product to be planted to be harvested in the future
Plantation	Represents a specific plantation activity. Each plantation is associated with a product and involves planting seeds and managing the growth of the plants.	planting activity, crop cycle	Total harvest, partial harvest
Harvest	Represents the process of collecting mature crops from the plantation, recording the quantity and date of the harvested product.	total harvest, partial harvest	Multiple harvests can occur from a single plantation
AuxiliaryConsumption	Represents the consumption of auxiliary products (such as fertilizers, pesticides, etc.) used on the plantation. Each record follows the date of consumption and quantity of the auxiliary product.	consumption record, usage	Multiple auxiliary consumptions per plantation
AuxiliaryProduct	Represents auxiliary products that are consumed during the plantation process.	water natural fertilizer synthetic fertilizer	Auxiliary products are tracked by type and usage

Table 1 - Identify Relationship Types

3.3 Document Attributes

Entity name	Attributes	Description	Data Type & Length	Nulls	Multi-valued
Greenhouse	Id_greenhouse name	Represents a greenhouse where the plants are grown	INT (PK) VARCHAR (100)	No No	No No
Section	Id_section name	Represents a section within a greenhouse. Each section belongs to a single greenhouse	INT (PK) VARCHAR (1)	No No	No No
Product	Id_product name type	Represents the products that are planted and harvested within the system	INT (PK) VARCHAR (100) VARCHAR (1)	No No No	No No No
Plantation	Id_plantation planting_date unplugging_date seed_quantity	Represents a specific agricultural activity, involving planting seeds and possibly harvesting.	INT (PK) DATE DATE INT	No No No No	No No No No
Harvest	Id_harvest Harvest_date harvested_quantity id_product	Harvest represents the cultivated and harvested produce within the system	INT (PK) DATE INT	No No No No	No No No No
AuxiliaryConsumption	Id_consumption auxiliary_consumption_date quantity	Tracks the consumption of auxiliary products on the plantation, recording the date and quantity used.	INT (PK) DATE INT	No No No	No No No
AuxiliaryProduct	Id_auxiliary name	Represents auxiliary products that are consumed during the plantation process	INT (PK) VARCHAR (100)	No No	No No

Table 2 – Document Attributes

3.4 Identify and associate attributes with entity or relationship types

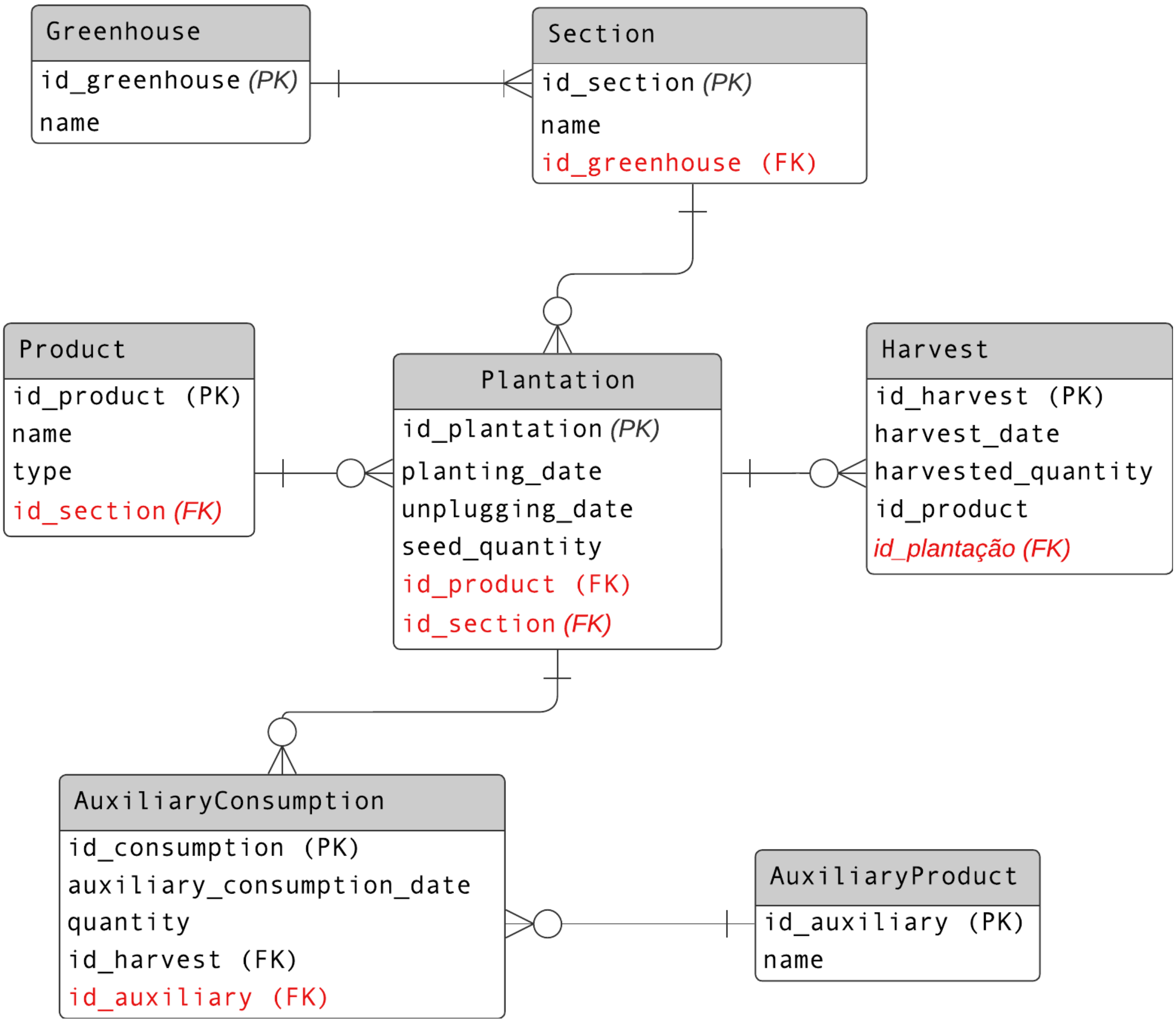
"In the next step, we identify facts about chosen entities and relationships for the database by looking for nouns or noun phrases in the user requirements. Attributes are identified when the noun or noun phrase represents a property, quality, identifier, or characteristic of an entity or relationship."

Entity Name	Multiplicity	Relations	Multiplicity	Name of Entity
Greenhouse	1..1	Has	1..3	Section
Section	1..1	Has	0..*	Plantation
Product	1..1	Planted	0..*	Plantation
Plantation	1..1	Harvested	0..*	Harvest
Plantation	1..1	Watered	0..*	AuxiliaryConsumption
AuxiliaryProduct	1..1	Used	0..*	AuxiliaryConsumption

Table 3 - Identify and associate attributes with entity or relationship types

4. Logical Model

Figure 2 – logical model



4.1 Validate relations using normalization

Normalization ensures that database tables are structured efficiently, minimizing redundancy and dependency.
The normalization process aims to achieve the following:

- Minimal attributes required to support data requirements.
- Logical grouping of attributes with functional dependencies within the same relation.
- Reduction of redundancy, ensuring each attribute is represented only once, except for those forming foreign keys necessary for relational joins.

Greenhouse	
Id_greenhouse	name
1	Greenhouse 1
2	Greenhouse 2
3	Greenhouse 3

Table 4 – Greenhouse normalization

Section	
Id_section	name
1	a
2	v
3	f

Table 5 – Section normalization

AuxiliaryProduct	
Id_auxiliary	name
1	water
2	Organic fertilizer
3	Inorganic fertilizer

Table 6 – AuxiliaryProduct normalization

Harvest			
Id_harvest	Harvest_date	Harvested_quantity	Id_product
1	05-06-2024	35	1
2	05-05-2024	55	2
3	05-042024	40	3

Table 7 – Harvest normalization

Plantation			
Id_plantation	planting_date	unplugging_date	Seed_quantity
1	05-06-2024	05-06-2025	35
2	05-05-2024	05-05-2025	55
3	05-04-2024	05-04-2025	40

Table 8 – Plantation normalization

Product		
Id_product	name	type
1	Product 1	A
2	Product 2	F
3	Product 3	V

Table 9 – Product normalization

AuxiliaryConsumption		
Id_consumption	Auxiliary_consumption_date	quantity
1	05-06-2024	35
2	05-05-2024	55
3	05-04-2024	40

Table 10 – AuxiliaryConsumption normalization

1. First Normal Form (1NF):

- Ensure each table has a primary key.
- Ensure all attributes contain only atomic (indivisible) values.
- Ensure each attribute contains values of a single type.

➢ All derived relations are already in 1NF as they have primary keys, atomic values, and single-type attributes.

2. Second Normal Form (2NF):

- Ensure the relation is in 1NF.
- Ensure all non-key attributes are fully functionally dependent on the primary key.

➢ Greenhouse: Already in 2NF.
➢ Section: Already in 2NF.
➢ Product: Already in 2NF.
➢ Plantation: Already in 2NF.
➢ Harvest: Already in 2NF.
➢ AuxiliaryProduct: Already in 2NF.
➢ AuxiliaryConsumption: Already in 2NF.

3. Third Normal Form (3NF):

- Ensure the relation is in 2NF.
- Ensure all non-key attributes are non-transitively dependent on the primary key.

➢ Greenhouse: Already in 3NF.
➢ Section: Already in 3NF.
➢ Product: Already in 3NF.
➢ Plantation: Already in 3NF.
➢ Harvest: Already in 3NF.
➢ AuxiliaryProduct: Already in 3NF.
➢ AuxiliaryConsumption: Already in 3NF.

After applying the normalization process to the derived relations, it is confirmed that all relations comply with the 3NF rules. This ensures that they are well-structured, reduces redundancy, and improves data integrity, thereby providing a robust logical data model foundation for the greenhouse management system.

4.2 Verify relations through user transactions

To ensure that the logical model supports the necessary user transactions, we validate the relations against common queries that users might perform. Here are the details of how the model supports specific transactions:

1. What products were harvested last month?
   - **Query**: Retrieve the product names and harvest dates for harvests that occurred in the previous month.
   - **SQL:**

Figure 3 – query 1

```
SELECT p.name, h.harvest_date
FROM Harvest h
JOIN Product p ON h.id_product = p.id_product
WHERE h.harvest_date >= DATEADD(month, -1, GETDATE())
      AND h.harvest_date < GETDATE();
```

2. What's the quantity of the planted trimester by product in the last year?
   - **Query**: Calculate the total quantity of seeds planted for each product in each trimester of the past year.
   - **SQL**

Figure 4 – query 2

```
SELECT p.name,
DATEPART(QUARTER, pl.planting_date) AS trimester,
SUM(pl.seed_quantity) AS total_quantity
FROM Plantation pl
JOIN Product p ON pl.id_product = p.id_product
WHERE pl.planting_date >= DATEADD(year, -1, GETDATE())
GROUP BY p.name, DATEPART(QUARTER, pl.planting_date);
```

3. List of products with the most quantity harvested in the year of 2018?
   - **Query**: Identify products that had the highest harvested quantity in 2018.
   - **SQL**:

Figure 5 – query 3

```
SELECT p.name, SUM(h.harvested_quantity) AS total_harvested FROM
Harvest h JOIN Product p ON h.id_product = p.id_product WHERE
YEAR(h.harvest_date) = 2018 GROUP BY p.name ORDER BY
total_harvested DESC;
```

4. How much of each produce was harvested?
   - **Query**: Calculate the total quantity harvested for each product.
   - **SQL**:

Figure 6 – query 4

```sql
SELECT p.name,
       SUM(h.harvested_quantity) AS total_harvested
FROM Harvest h
JOIN Product p ON h.id_product = p.id_product
GROUP BY p.name;
```

By validating these transactions, we demonstrate that the logical model can support essential operations and provide meaningful data insights. The structured approach ensures that the database can handle various queries efficiently, helping users manage greenhouse operations effectively.

4.4 Integrity Constraints

The greenhouse management system relies on integrity constraints to maintain the accuracy and consistency of its data. These constraints enforce rules and conditions to prevent the entry of invalid or inconsistent data.
Here are the key integrity constraints for the system:

Entity Integrity:

Each primary key attribute must have a non-null value and be unique within its respective table.
Example: The id_product attribute in the Product table serves as the primary key, ensuring that each product has a unique identifier.

Referential Integrity:

Foreign key constraints ensure that values entered into a foreign key column must exist in the referenced primary key column of another table.
Example: The id_product column in the Plantation table references the id_product column in the Product table, ensuring that only valid product identifiers can be inserted.

Domain Integrity:

Domain constraints define the allowable values for attributes within a table.
Example: The type attribute in the Product table is constrained to accept only specific values ('V' for vegetable, 'F' for fruit, or 'A' for aromatic herb).

Check Constraints:

Check constraints enforce specific conditions on data entered into a column.
Example: A check constraint on the harvested_quantity column in the Harvest table ensures that the quantity harvested is always non-negative.

Entity Relationship Constraints:

These constraints ensure that relationships between entities are maintained correctly.
Example: In the Plantation table, each plantation must be associated with a valid section, ensuring that plantations are always assigned to existing sections.

Business Rules:

Additional business rules specific to the greenhouse management system may be enforced using triggers or stored procedures.
Example: A trigger can be implemented to enforce a maximum limit of 10 distinct products per section in the Plantation table.

Implementing these integrity constraints helps the greenhouse management system maintain data integrity and consistency, providing users with reliable and accurate information for decision-making and analysis.

5. Physical Model

5.1 Design base relations

The physical model transforms the logical design into specific implementation details, which include defining tables, columns, indexes, and other database objects. In the context of the greenhouse management system, the physical model describes the arrangement of the database tables and their relationships.

Greenhouse Table:

This table stores information about the greenhouses.
Attributes:

- *id_greenhouse*: Primary Key, uniquely identifies each greenhouse.
- *name*: Stores the name of the greenhouse.

Section Table:

This table represents the sections within each greenhouse.
Attributes:

- *id_section*: Primary Key, uniquely identifies each section.
- *name*: Indicates the type of section (e.g., herbs, vegetables, fruits).
- *id_greenhouse*: Foreign Key, references the greenhouse to which the section belongs.

Product Table:

This table contains details about the products grown in the sections.
Attributes:

- *id_product*: Primary Key, uniquely identifies each product.
- *name*: Stores the name of the product.
- *type*: Indicates the type of product (e.g., vegetable, fruit, herb).
- *id_section*: Foreign Key, indicates the section where the product is grown.

AuxiliaryProduct Table:

This table stores information about auxiliary products.
Attributes:

- *id_auxiliary*: Primary Key, uniquely identifies each auxiliary product.
- *name*: Stores the name of the auxiliary product.

Plantation Table:

This table records details of the plantations within sections.
Attributes:

- *id_plantation*: Primary Key, uniquely identifies each plantation.

- *planting_date*: Indicates the date when the plantation was initiated.
- *unplugging_date*: Indicates the date when the plantation was terminated.
- *seed_quantity*: Stores the quantity of seeds planted.
- *id_product*: Foreign Key, references the product planted in the plantation.
- *id_section*: Foreign Key, indicates the section where the plantation is located.

Harvest Table:

This table tracks information about harvested products.
Attributes:

- *id_harvest*: Primary Key, uniquely identifies each harvest record.
- *harvest_date*: Indicates the date when the harvest occurred.
- *harvested_quantity*: Stores the quantity of product harvested.
- *id_product*: Foreign Key, references the product harvested.
- *id_plantation*: Foreign Key, indicates the plantation from which the harvest was made.

AuxiliaryConsumption Table:

This table records consumption of auxiliary products associated with plantations.
Attributes:

- *id_consumption:* Primary Key, uniquely identifies each consumption record.
- *auxiliary_consumption_date*: Indicates the date of auxiliary product consumption.
- *quantity*: Stores the quantity of auxiliary product consumed.
- *id_plantation*: Foreign Key, indicates the plantation where consumption occurred.
- *id_auxiliary*: Foreign Key, references the auxiliary product consumed.

6. Conclusion and Future Work

After a structured approach, we have successfully completed the design and development of the greenhouse management database system. We followed a structured approach that included conceptual, logical, and physical modeling phases.

Throughout this process, we addressed various aspects such as identifying entity types, establishing relationships, deriving logical relations, validating through normalization, and designing base relations in the physical model. Each step was meticulously executed to ensure the integrity, efficiency, and effectiveness of the database system.

By adhering to sound database design principles and considering the specific requirements of greenhouse management, we have created a robust and scalable system capable of handling the complexities associated with greenhouse operations. The database schema incorporates appropriate constraints, relationships, and optimizations to facilitate seamless data management, retrieval, and analysis.

However, there are some limitations that are important to mention. In the harvest table, the presence of the product_id is insignificant and meaningless, since each plantation is only associated with one product at a time. In other words, it is not necessary for the stored procedure to receive the product ID as a parameter.

In conclusion, the successful development of the greenhouse management database marks a significant milestone in the pursuit of efficient and technology-driven agricultural management solutions, poised to drive innovation and growth in the greenhouse industry. The completion of this project has been greatly influenced by the teachings of "Database Systems: A Practical Approach to Design, Implementation, and Management" by Thomas Connolly and Carolyn Begg.

Bibliography

Connolly, T., & Begg, C. (2021). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed., p.). [Pearson].

List of Acronyms and Abbreviations

BD	Base de Dados
1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
PK	Primary Key
FK	Foreign Key