Universidad Autónoma de Zacatecas

Academic Unit of Electrical Engineering

Software Engineering Academic Program

Group: 5B - Semester: 2022-5º

Practice Number: 1

Practice Name: Review of Database Systems I

Date: 18/August/2022

professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Índice

Review of Database Systems I

August 18th, 2022

1. Introduction

An entity-relationship diagram, also known as an entity-relationship model or ERD, is a type of flowchart that illustrates how .[en]tities,"such as people, objects, or concepts, relate to each other within a system. ER diagrams are often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. Also known as ERDs or ER models, they employ a defined set of symbols, such as rectangles, diamonds, ovals, and connecting lines, to represent the interconnectedness of entities, relationships, and their attributes. They are a reflection of the grammatical structure and use entities as nouns and relationships as verbs.

2. Practice objective

Review the creation of entity-relationship models (ER model) from several data sources (problem statements).

3. Developing

To carry out this first practice I had to make a small reminder of the entity-relationship diagrams and the diagrams seen in the subject of database systems one.

Exercise 1: For this activity number 1, the entity-relationship diagram and the relational diagram of a scenario planted in practice were carried out. These diagrams can be seen below:

In this activity number one we were asked to make an entity-relationship diagram of a proposed scenario. To make this diagram, various entities were made, which are:

Customer: This entity represents a customer of the bookstore.

Product: This entity represents a product of the library; For example: book, CDs, Blue-ray, electronic files, etc.

Type Product: This entity represents the type of product that is sold in the bookstore, which can be books, multimedia files, electronic files, etc. This entity is related to the Product entity since the products can be classified into different types, this relationship has a carnality of many to one since a product can only be of one type and a product type can have many products.

Store: This entity represents the store or establishment where the products are stored. This entity is related to the products entity and has a one-to-many cardinality since a store can have many products and a product can belong to only one store.

A product-customer relationship was also added to the diagram, this relationship is called shopping cart and it was done since a customer can add products to the shopping cart. And said process was put into an aggregation since this process could generate a sale of one or several products.

In **figure number one** you can see the entity-relationship diagram of activity one.

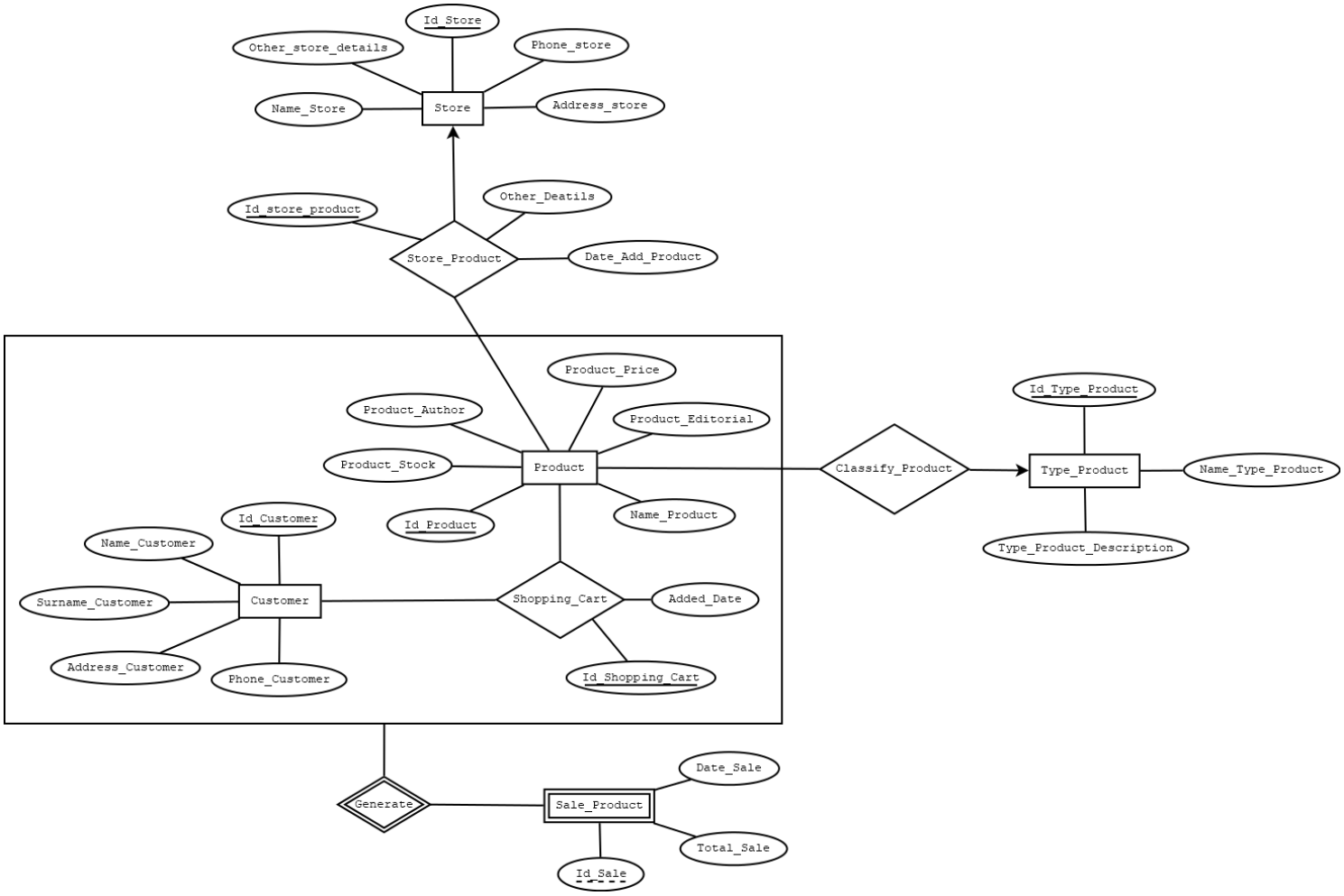

Figura 1: *Entity-relationship diagram activity number 1.*

In **figure 2** you can see the relational diagram of activity one.

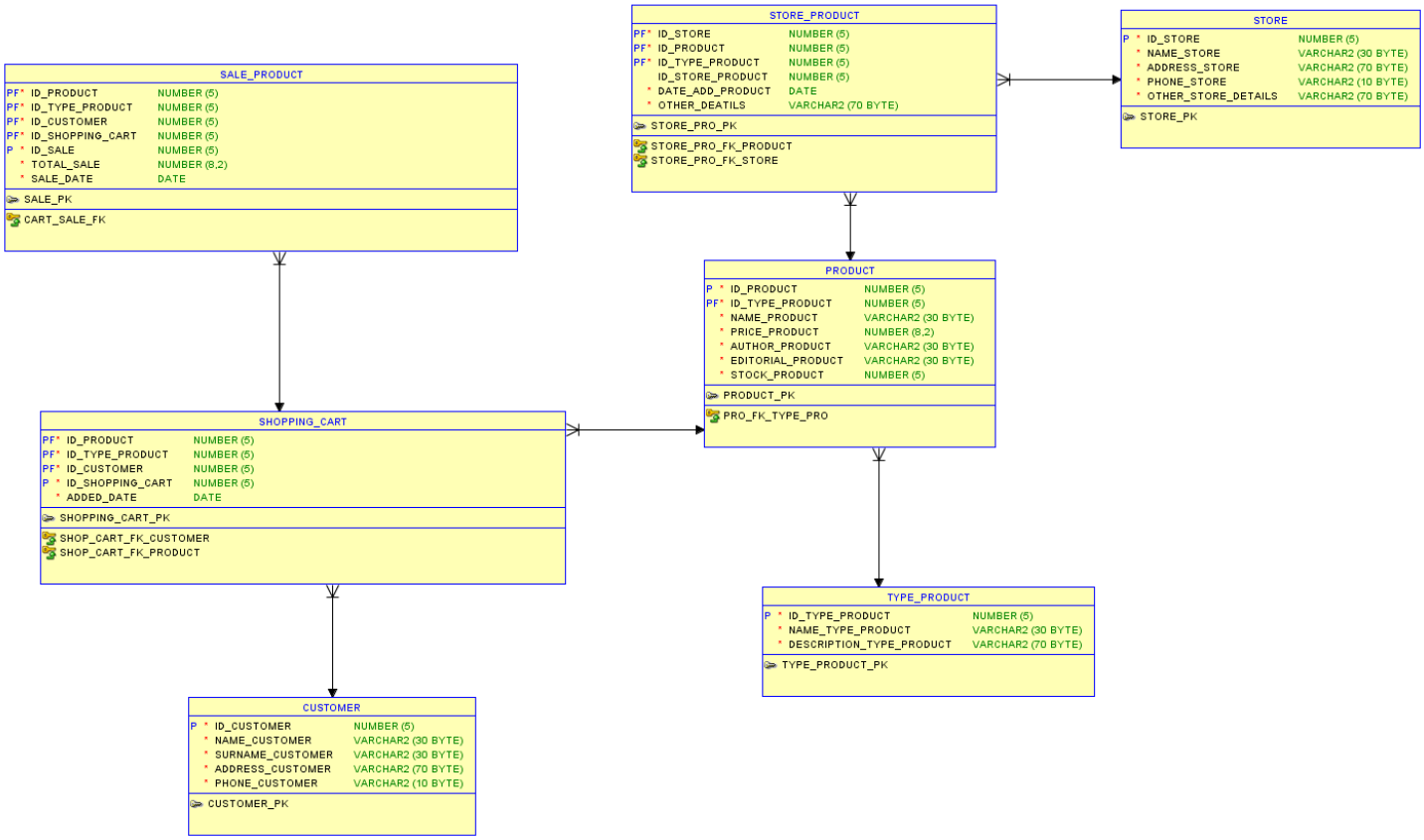

Figura 2: *Relational diagram of activity 1.*

Exercise 2: Activity number two was not finished.

Exercise 3: In this activity number three we were asked to convert a relational model to an entity-relationship diagram where a result like the one shown in **figure 3** was obtained.

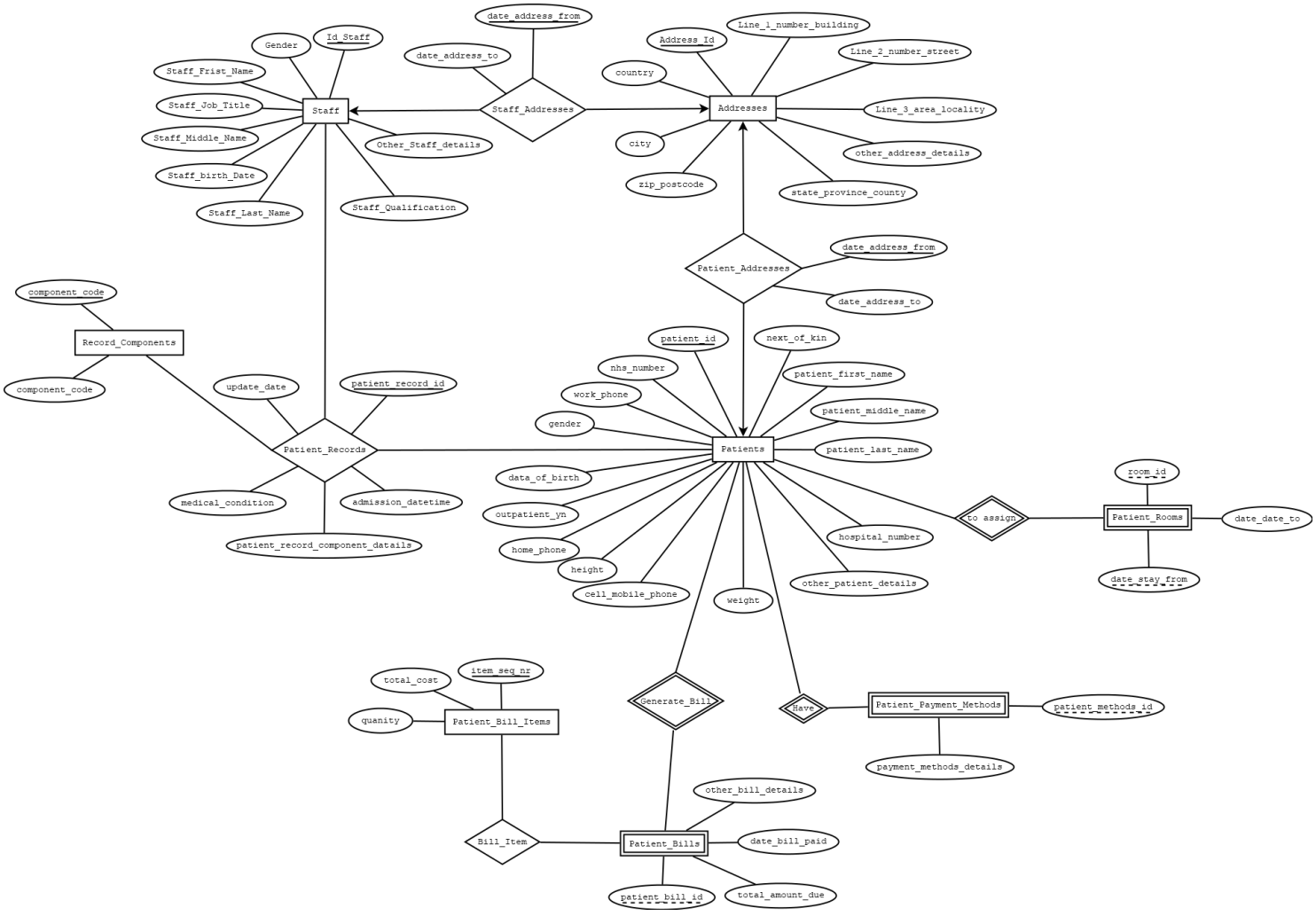

Figura 3: *Entity-relationship diagram of activity three*

Exercise 4: In activity number four we were asked to generate the Oracle DDL statements corresponding to the diagram shown in **figure 4**.

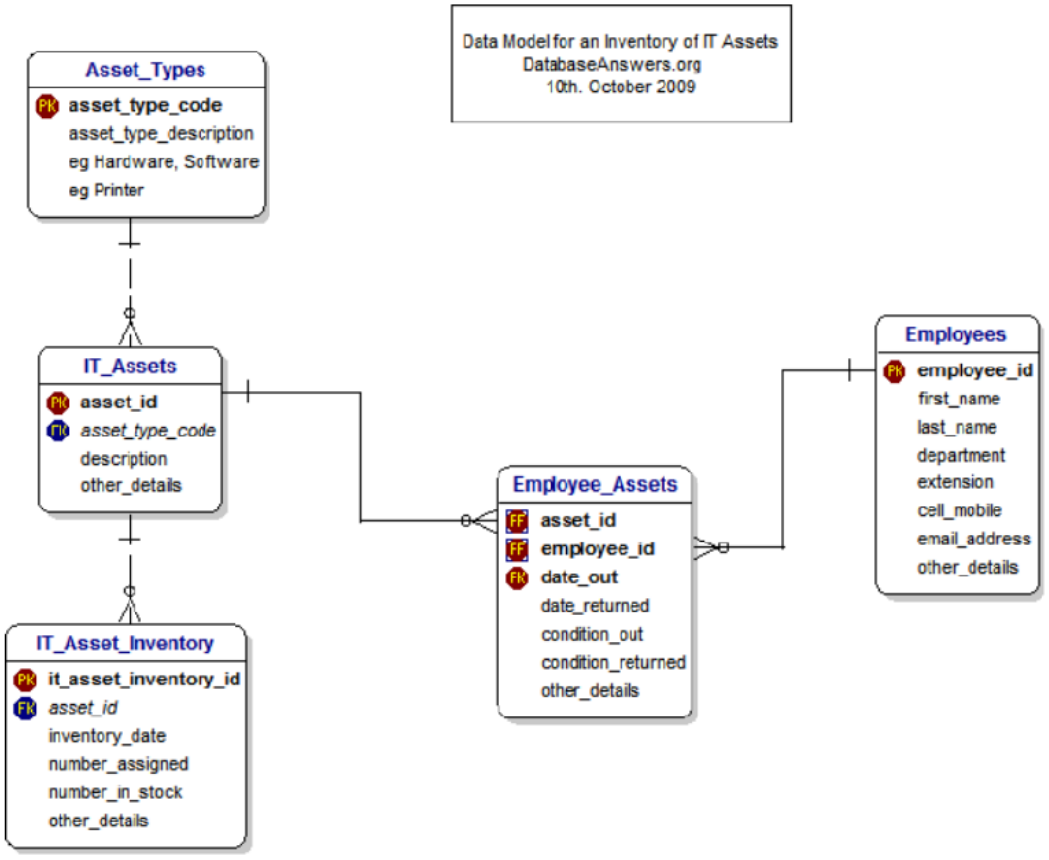

Figura 4: *Relational diagram activity 4.*

The following figures show the DDL statements for creating the entities and relationships in the diagram of figure 4.

```sql
CREATE TABLE ASSET_TYPES (
    ASSET_TYPE_CODE CHAR(8) CONSTRAINT ASSET_TYPE_CODE_PK PRIMARY KEY,
    ASSET_TYPE_DESCRIPTION VARCHAR2(30) CONSTRAINT ASSET_TYPE_DESCRIPTION_NN NOT NULL

);

SELECT * FROM ASSET_TYPES;

CREATE TABLE IT_ASSETS (
    ASSET_ID NUMBER(5),
    ASSET_TYPE_CODE CHAR(8),
    DESCRIPTIONS VARCHAR2(30) CONSTRAINT DESCRIPTION_NN NOT NULL,
    OTHER_DETAILS VARCHAR2(30) CONSTRAINT OTHER_DETAILS_NN NOT NULL,
    CONSTRAINT IT_ASSETS_PK PRIMARY KEY (ASSET_ID),
    CONSTRAINT ASSET_TYPE_FK FOREIGN KEY (ASSET_TYPE_CODE) REFERENCES ASSET_TYPES (ASSET_TYPE_CODE)
);

SELECT * FROM IT_ASSETS;
```

Figura 5: *DDL statement of activity four.*

```sql
CREATE TABLE IT_ASSETS_INVENTORY (
    IT_ASSETS_INVENTORY_ID NUMBER(5) CONSTRAINT IT_ASSETS_INVENTORY_ID_PK PRIMARY KEY,
    ASSET_ID NUMBER(5),
    INVENTORY_DATE DATE CONSTRAINT INVENTORY_DATE_NN NOT NULL,
    NUMBER_ASSIGNED NUMBER(5) CONSTRAINT NUM_ASSIGNED_NN NOT NULL,
    NUMBER_IN_STOCK NUMBER(5) CONSTRAINT NUM_IN_STOCK_NN NOT NULL,
    OTHER_DETAILS VARCHAR2(30) CONSTRAINT OTHER_DETAILS_INV_NN NOT NULL,
    CONSTRAINT ASSET_FK FOREIGN KEY (ASSET_ID) REFERENCES IT_ASSETS (ASSET_ID)
);

SELECT * FROM IT_ASSETS_INVENTORY;

CREATE TABLE EMPLOYEES (
    EMPLOYEE_ID NUMBER(5) CONSTRAINT EMPLOYEE_ID_PK PRIMARY KEY,
    FRIST_NAME VARCHAR2(30) CONSTRAINT FIRST_NAME_NN NOT NULL,
    LAST_NAME VARCHAR2(30) CONSTRAINT LAST_NAME_NN NOT NULL,
    DEPARTMENT VARCHAR2(45) CONSTRAINT DEPARTMENT_NN NOT NULL,
    EXTENSION NUMBER(3) CONSTRAINT EXTENSION_NN NOT NULL,
    CELL_MOBILE VARCHAR2(10) CONSTRAINT CELL_MOBILE_NN NOT NULL,
    EMAIL_ADDRESS VARCHAR2(45) CONSTRAINT EMAIL_ADDRESS_NN NOT NULL,
    OTHER_DETAILS VARCHAR2(30) CONSTRAINT OTHER_DETAILS_EMP_NN NOT NULL
);

SELECT * FROM EMPLOYEES;
```

Figura 6: *DDL statement of activity four.*

```
47  SELECT * FROM EMPLOYEES;
48
49  CREATE TABLE EMPLOYEE_ASSETS (
50      ASSET_ID NUMBER(5),
51      EMPLOYEE_ID NUMBER(5),
52      DATE_OUT DATE,
53      CONSTRAINT EMPLOYEE_ASSETS_PK PRIMARY KEY (ASSET_ID, EMPLOYEE_ID, DATE_OUT),
54      CONSTRAINT ASSET_EMP_FK FOREIGN KEY (ASSET_ID) REFERENCES IT_ASSETS (ASSET_ID),
55      CONSTRAINT EMPLOYEE_FK FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEES (EMPLOYEE_ID)
56  );
57
58  SELECT * FROM EMPLOYEE_ASSETS;
59
```

Figura 7: *DDL statement of activity four.*

Then we were asked to automatically generate the corresponding relational model in Sqldeveloper (using the graphical user interface) and once this was done we were asked to Compare the results with those provided in this exercise. **Figure 8** shows the relational model that was generated from the creation of the tables.

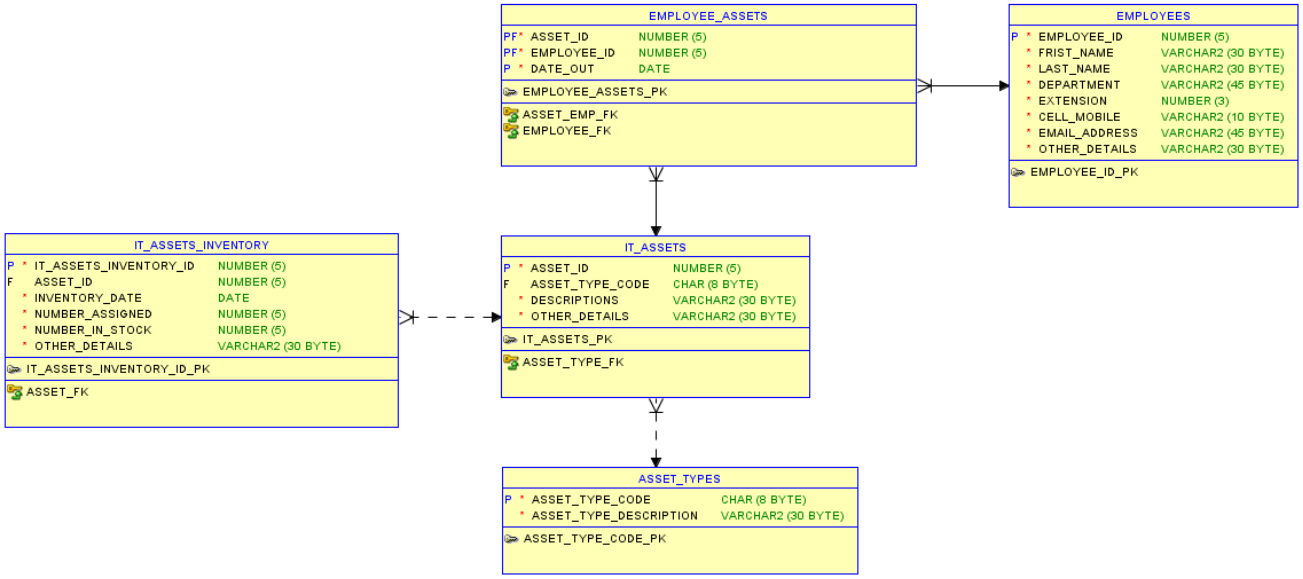

Figura 8: *Activity 4 relational model.*

Finally, we were asked to automatically generate the logic model using reverse engineering, the diagram we obtained is shown in **figure 9**.

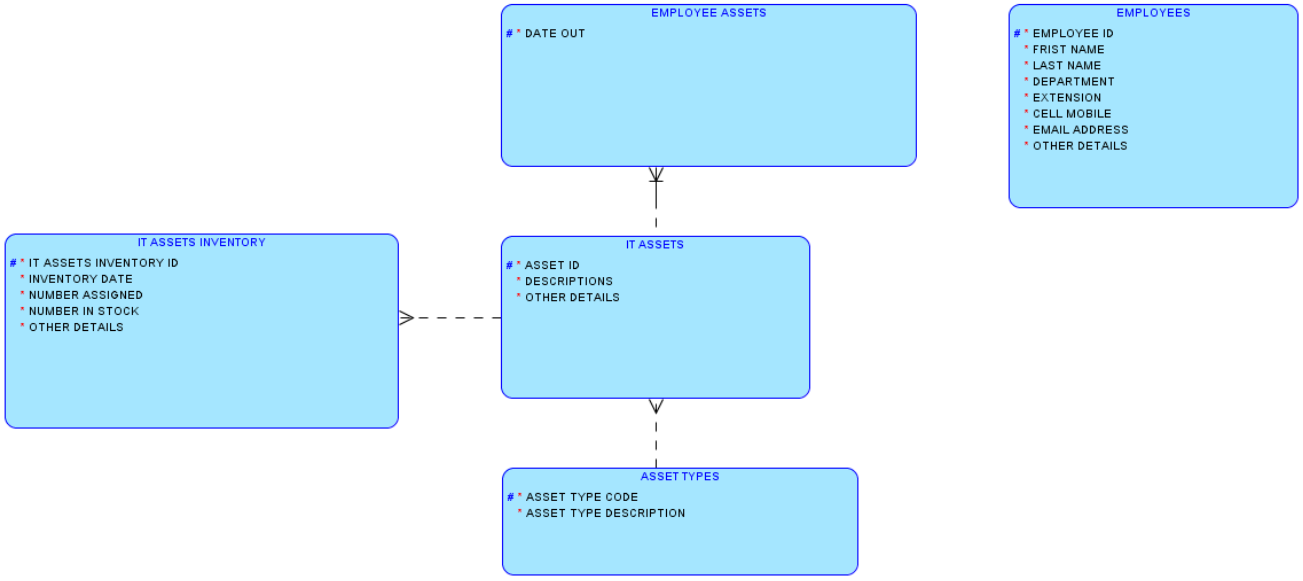

Figura 9: *Reverse Logic Model.*

4. Pre-assessment

In this section you will find the Pre-assessment

Criteria to be evaluate	Does it comply?	(%)
COMPLIES WITH THE REQUESTED FUNCTIONALITY	YES	
HAS THE CORRECT INDENTATION	YES	
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	YES	
HAS A REPORT WITH IDC FORMAT	YES	
REPORT INFORMATION IS FREE OF SPELLING ERRORS	YES	
DELIVERED IN TIME AND FORM	YES	
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	NO	90 %

5. Conclusion

To conclude with this first practice I just want to mention that it was very helpful since it helped me remember and review concepts about entity-relationship models and relational models. Although it is worth mentioning that I still have a hard time applying the modeling concepts correctly, for which I think I have to improve so that in future practices I can do a better job.