

Review of Database Systems I

Jesús Manuel Juárez Pasillas

August 19, 2021

1 Introduction

In this practice, a review of what was seen in "Database Systems I" will be carried out, this in order to remember what was learned in this matter. To avoid errors, entity-relationship diagrams, relational models and logical models are used with which the entities and their attributes are considered (defining the data types, length, etc.), as well as the relationships between the entities and how they are each of these behave. Based on these diagrams, the DDL statements are created with which the tables will be created, the attributes that will be saved, as well as their data type and the length they will have.

2 Developing

2.1 Activity 1:

This activity consists of creating an entity-relationship diagram which fulfills everything requested in a statement. After creating the diagram, it is asked to make the relational model in the sqldeveloper tool.

1. Generate an entity-relationship model that satisfies the request (Figure 1).

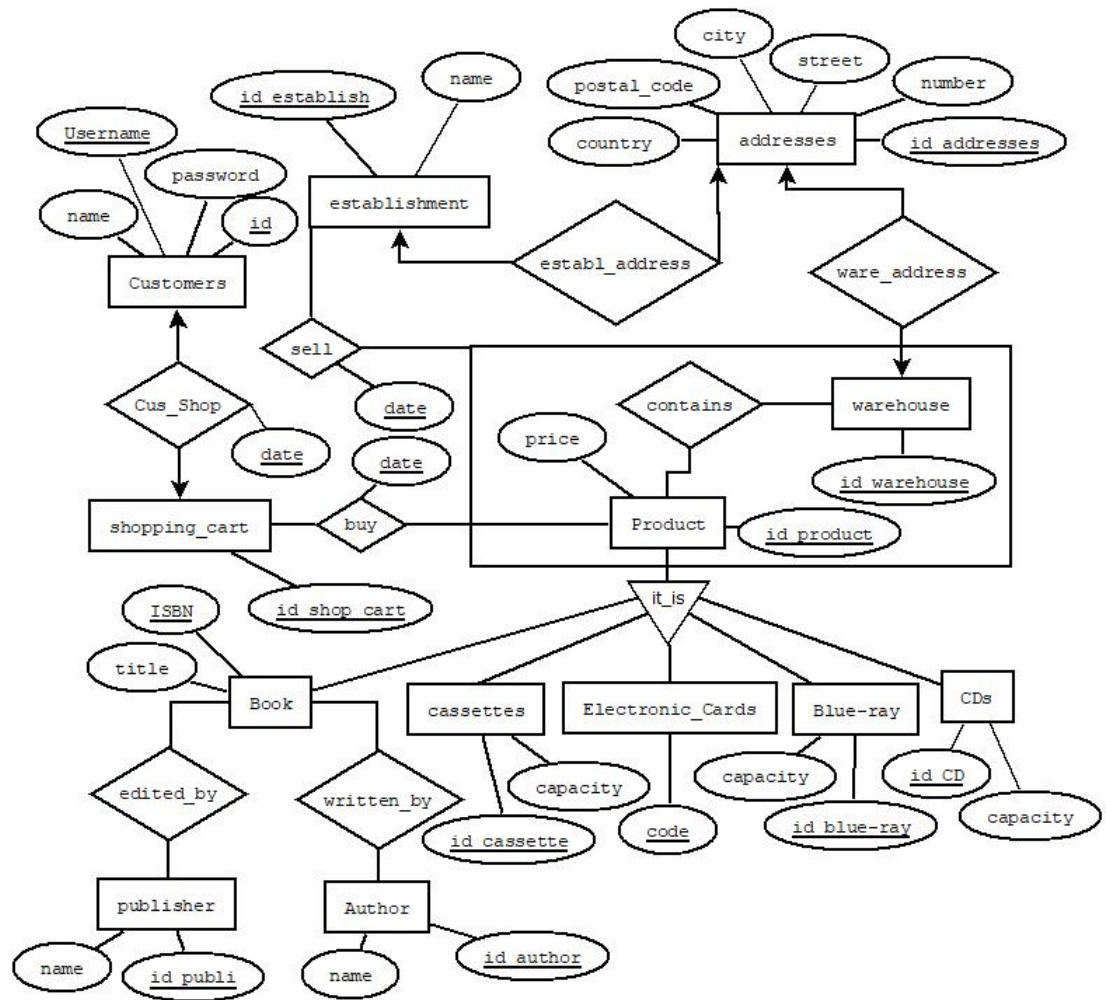


Figure 1: Entity-relationship model

2. Generate the corresponding relational model in Sqldeveloper (Figure 2).

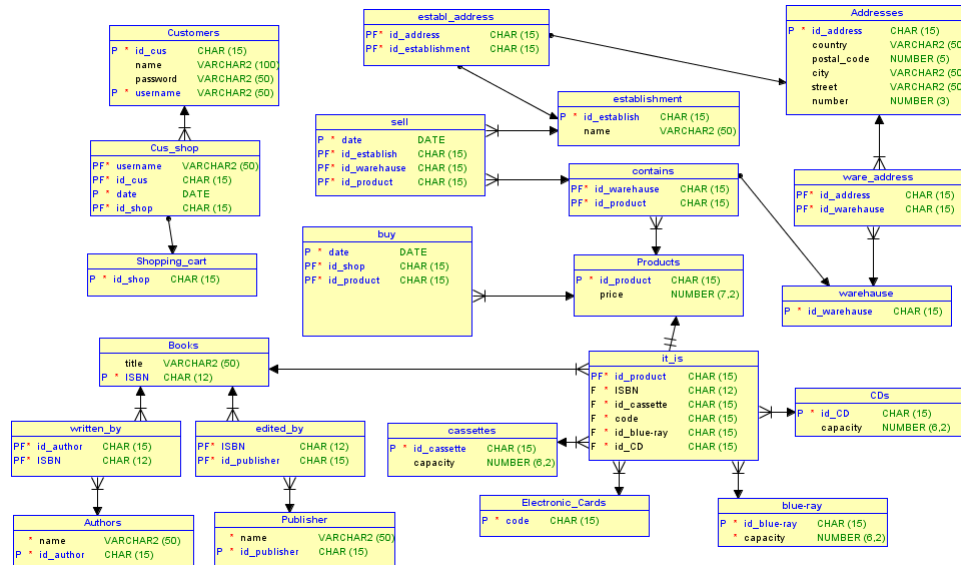


Figure 2: Relational model

2.2 Activity 3:

Activity 3: To create this activity, it is requested to pass a relational model to an entity-relationship diagram (Figure 3).

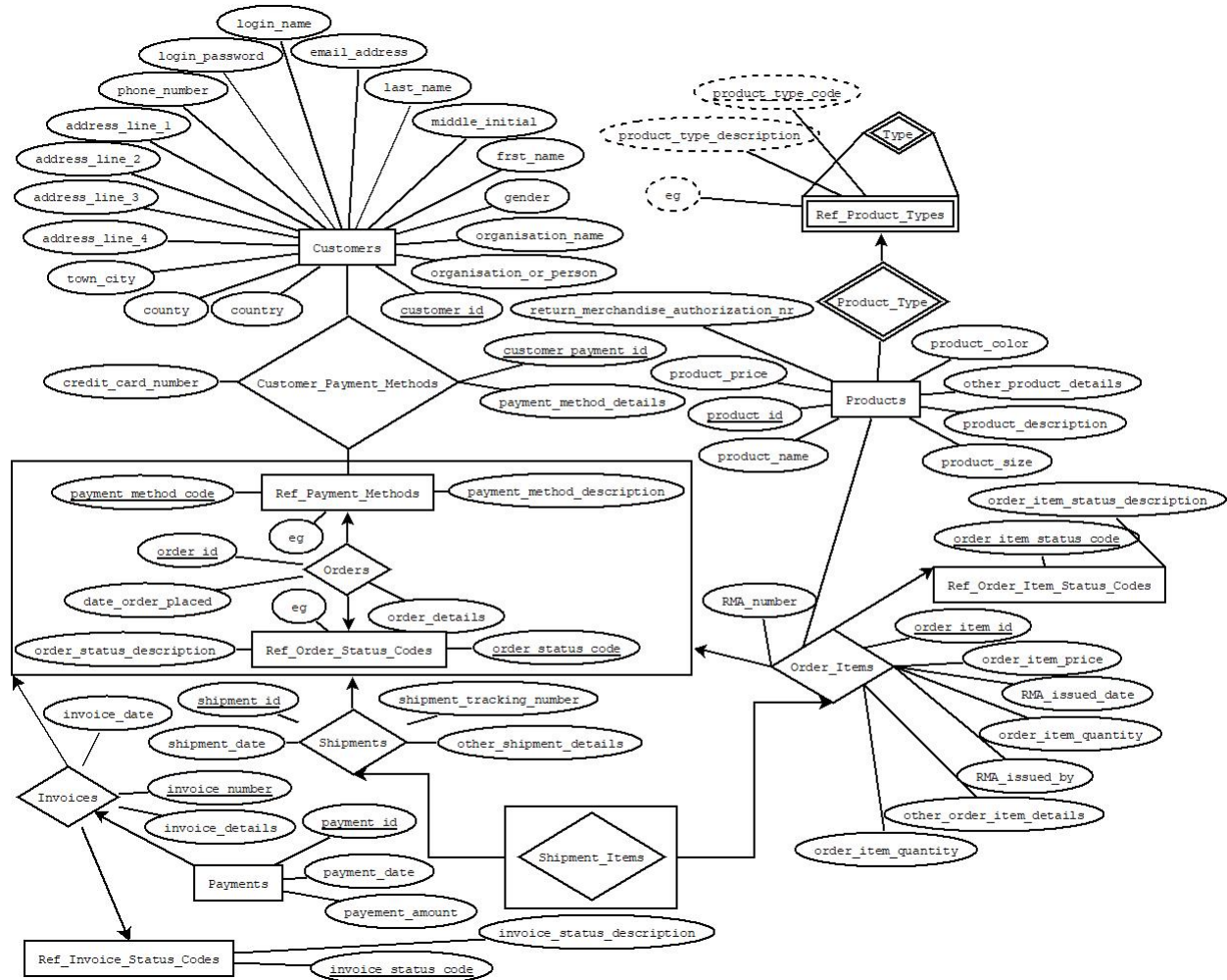


Figure 3: Entity-relationship model

2.3 Activity 4:

For this last activity you need to create the DDL statements in Oracle, once the tables have been generated create the relational model in sqldeveloper and from this relational model use reverse engineering to create the logical model.

1. Generate the corresponding DDL Oracle statements.

- create table Asset_Types(
asset_type_code number(8) constraint asset_types_pk primary key,
asset_type_description varchar2(100) constraint ass_ty_des_nn not null
);
- create table IT_Assets(
asset_id number(8) constraint it_asset_pk primary key,
asset_type_code number(8),
description varchar2(100),
other_details varchar2(100),
constraint asset_type_it_asset_fk foreign key (asset_type_code) references Asset_Types(asset_type_code)
);
- create table IT_Asset_Inventory(
it_asset_inventory_id number(8) constraint it_asset_inven_pk primary key,
asset_id number(8) constraint it_as_as_nn not null,
inventory_date Date constraint asset_inven_inven_nn not null,
number_assigned number(8) constraint asset_inven_num_assig_nn not null,
number_in_stock number(8) constraint asset_inven_num_in_stock_nn not null,
other_details varchar2(100),
constraint it_asset_it_asset_inven_fk foreign key (asset_id) references IT_Assets(asset_id)
);
- create table Employee(
employee_id number(8) constraint employee_pk primary key,
first_name varchar(50) constraint emplo_fir_na_nn not null,
last_name varchar(50) constraint emplo_las_na_nn not null,
departament varchar(50) constraint emplo_dep_nn not null,
extension varchar(50) constraint emplo_ext_nn not null,
cell_mobile char(15) constraint emplo_cell_mo_nn not null,
email_address varchar(50) constraint emplo_ema_add_nn not null,
other_details varchar(100)
);
- create table Employee_Assets(
asset_id number(8),
employee_id number(8),
date_out date,
date_returned varchar(50),
condition_out varchar(50),
condition_returned varchar(50),
other_details varchar(100),
constraint emplo_emplo_asse_fk foreign key (employee_id) references Employee(employee_id),

constraint asse_emplo_asse_fk foreign key (asset_id) references IT_Assets(asset_id),
 constraint employee_asset_pk primary key (asset_id,employee_id,date_out)
);

2. Generate automatically the corresponding relational model in Sqldeveloper (Figure 4).

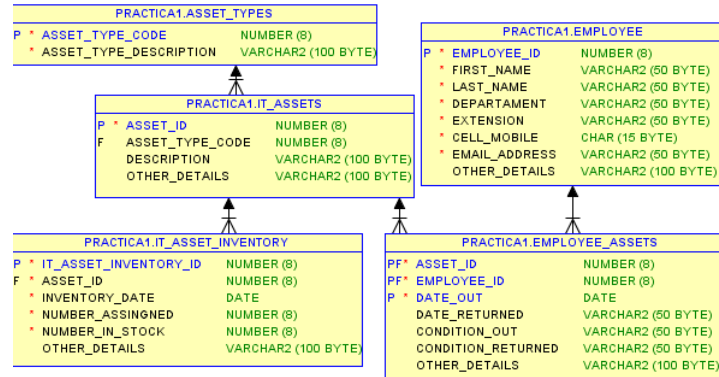


Figure 4: Relational model

3. Compare the results with the provided in this exercise.
 - When creating the database from the relational model, it is very easy to create the DLL statements, it is also easy to identify the attributes that each table will contain and if these are primary keys or foreign keys, identifying the table where they come from and with this identifying the restrictions that each table will have.
4. Generate automatically the logical model by using the reverse engineering.

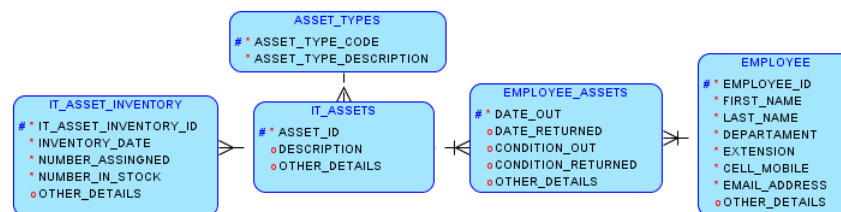


Figure 5: Logical model

2.4 Pre-Assessment:

- Practices pre-Assessment for Database Systems Laboratory II

Practice	Pre-Assessment
COMPLIES WITH THE REQUESTED FUNCTIONALITY	
HAS THE CORRECT INDENTATION	X
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	X
HAS A REPORT WITH IDC FORMAT	X
REPORT INFORMATION IS FREE OF SPELLING ERRORS	X
DELIVERED IN TIME AND FORM	X
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	80

Table 1: Pre-Assessment.

2.5 Conclusion:

All these models are created to define what the database tables will have, as well as each column in each table, and establish the data types and lengths that will be allowed to be saved in each table. Remembering how these diagrams, models and dll sentences are made will allow us to have better control over the database that is going to be created.