Practice 5: DDL

Jesús Manuel Juárez Pasillas

September 14 2021

1 Introduction

The creation of a database is not only creating the necessary sentences to do this, but it is also necessary to know which sentences are the ones that have to be created to have the best option for the database. For this, it is necessary to carry out a process where all the tables, columns and their respective data types will be defined, as well as sequences, indexes and synonyms for each table.

2 Developing

2.1 Activity 1:

You should define a problem statement about a topic of interest (a brief description).

- A car dealer wants to manage the sale of vehicles. It is required to know important information about the cars that are kept at the dealership, the customers who buy vehicles, and the employees who sell these vehicles.

2.2 Activity 2:

The problem statement of activity 1 will be passed to you (from another classmate). With this problem statement, you should be able to generate the ER diagram.

- I was provided with the following statement from my partner Marlon Alfredo García Meza, with which the requested entity-relationship diagram was created (Figure 1).
  "A hospital carries out monthly check-ups of patients, to find out the reason for admission and who was the doctor in charge. The hospital is made up of several medical departments, and the patients record the day they were admitted and the day they left ".

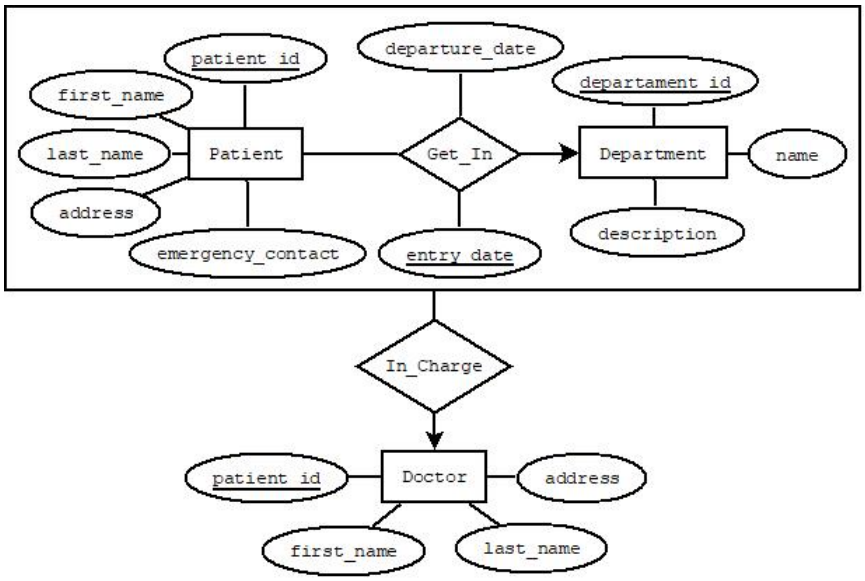

Figure 1: Entity relationship diagram.

2.3 Activity 3:

The problem statement and its corresponding ER diagram of activity 2 will be passed to you (from another classmate). With these two items, you should correct the necessary parts of the ER diagram (using your abstraction) according to its problem statement, then, you should be able to generate the relational diagram by using "Dia" software.

- The problem and the e-r diagram were provided by my colleague Rolando Aguilar Ordaz (Figure 2).
  "Problem: A company wants to store the game stats from Valorant players (matches won and lost, kills, deaths, assists, etc.) in a database so that they can be shown to people on their website".

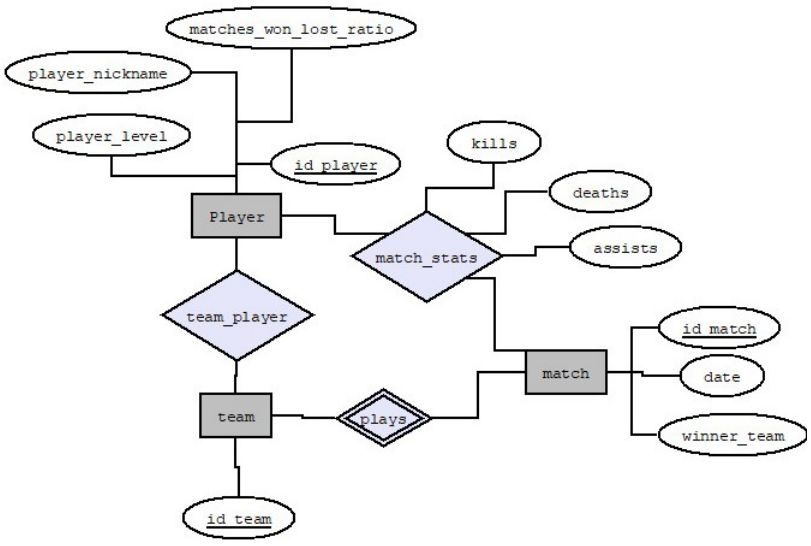

Figure 2: Rolando Aguilar Ordaz entity relationship diagram.

- The relationships specified many to many in all cases, so in some cases it was wrong. Also the relationship that had with the matches did not sound coherent to me, since I converted it to a ration of which I removed the id, and only left the date which I converted to the primary key (Figure 3).

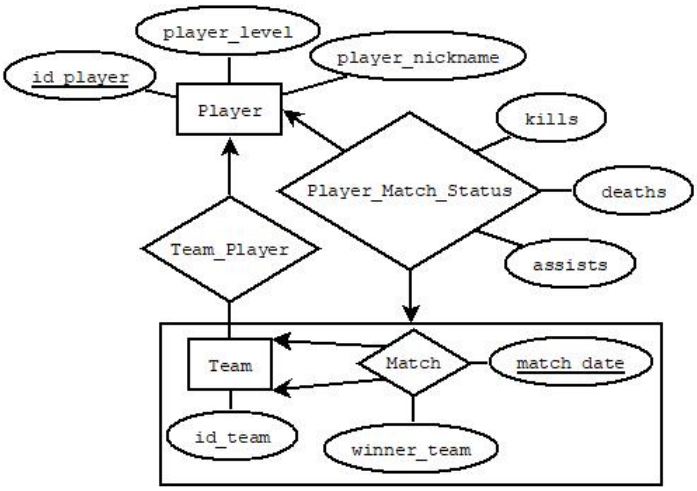

Figure 3: modified entity-relationship diagram.

- Relational diagram by using “Dia” software (Figure 4):

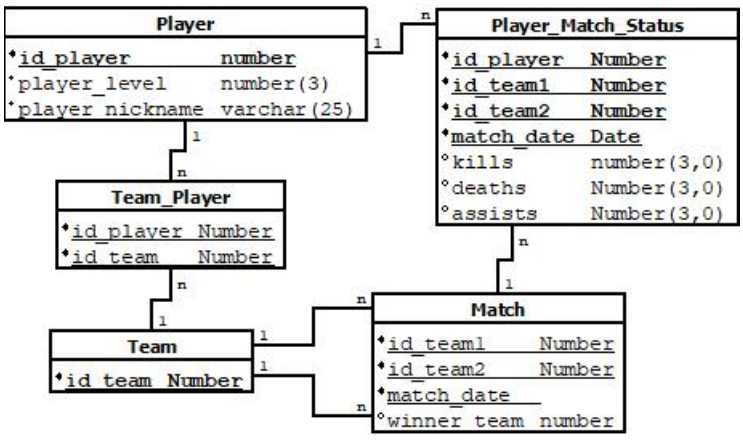

Figure 4: Relational diagram by using "Dia" software.

2.4 Activity 4:

The ER and relational diagrams of activity 3 will be passed to you (from another classmate). With just these two diagrams, you should correct the necessary parts of the relational diagram (using your abstraction) according to its ER diagram, then you should be able to generate the Oracle DDL sentences. You should add the basic indexes according to the diagram (reading the possible data to extract), the necessary sequences for the solution and the appropriate synonyms taking into account your insight.
With these tables, you should automatically generate the physical diagram in DATA MODELER (dragging the tables). Compare this diagram with the relational model made by Dia.

- The entity-relationship diagram and the relational diagram were provided by my colleague Francisco Vargas de la Llata Ibarra (Figure 5 and Figure 6).

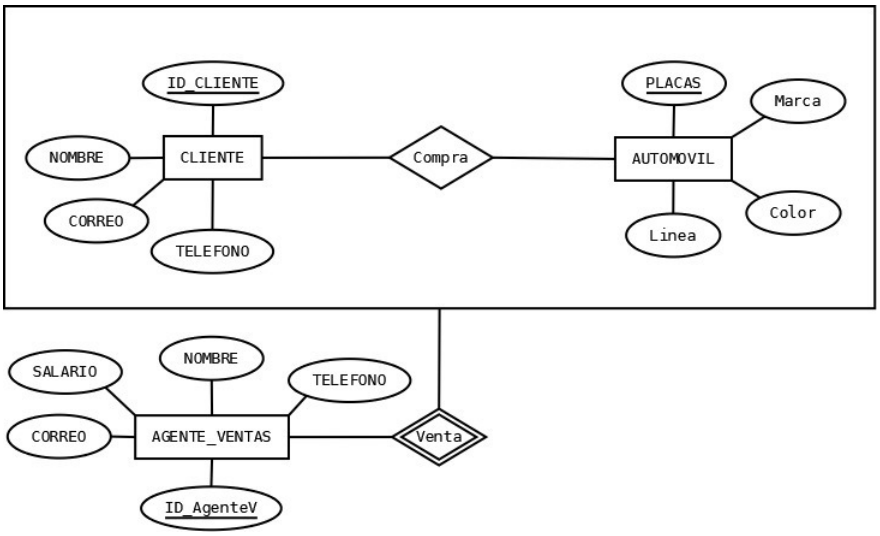

Figure 5: Francisco Vargas de la Llata Ibarra entity-relationship diagram.

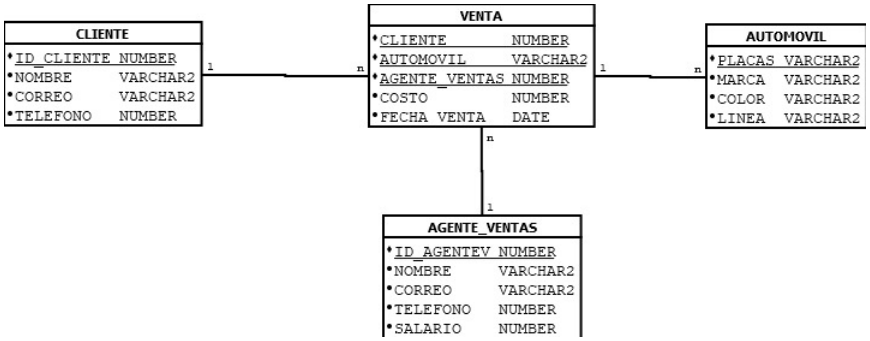

Figure 6: Relational diagram of Francisco Vargas de la Llata Ibarra.

- Only the cardinality was modified, as well as some data types and lengths were defined (Figure 7).

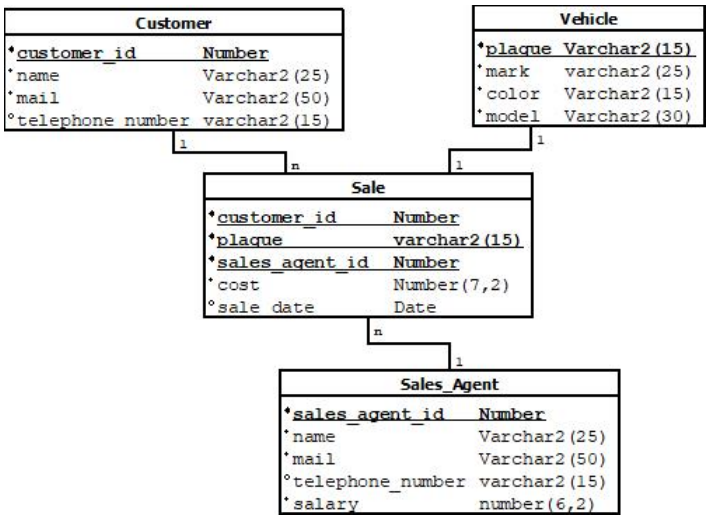

Figure 7: Modified relational diagram.

- Oracle DDL sentences:
  - create table Customers(
    customer_id number,
    name varchar2(25) constraint customer_name_nn not null,
    mail varchar2(50) constraint customer_mail_nn not null,
    telephone_number varchar2(15));
  - create table Vehicles(plaque varchar2(15),

    mark varchar(25) constraint vahicle_mark_nn not null,
    color varchar2(15) constraint vehicle_color_nn not null,
    model varchar(30) constraint vehicle_model_nn not null);
  - create table Sales_Agents(
    sales_agent_id number,
    name varchar2(25) constraint sales_ag_name_nn not null,
    mail varchar2(50) constraint sales_ag_mail_nn not null,
    telephone_number varchar2(15),
    salary number(6,2) constraint sales_ag_salary_nn not null);
  - create table Sales(
    plaque varchar2(15),
    customer_id number,

    cost number(7,2) constraint sale_cost_nn not null,
    sales_agent_id number,
    sale_date date);

- create index customer_name_idx on Customers(name);
- create index vehicle_mark_idx on Vehicles(mark);
- create bitmap index vehicle_color_idx on Vehicles(color);
- create index vehicle_model_idx on Vehicles(model);
- create index sales_ag_nam_idx on Sales_Agents(name);
- create index sales_ag_salary_idx on Sales_Agents(salary);
- create index sales_cost_idx on Sales(cost);
- create synonym Cus for Customers;
- create index sales_date_idx on Sales(sale_date);
- alter table Customers add constraint customer_pk primary key (customer_id);
- alter table Vehicles add constraint vehicle_pk primary key (plaque);
- alter table Sales_Agents add constraint sale_ag_pk primary key (sales_agent_id);
- alter table Sales add constraint cus_sales_fk foreign key (customer_id) references Customers(customer_id);
- alter table Sales add constraint sales_pk primary key (customer_id, plaque, sales_agent_id);
- alter table Sales add constraint vehi_sales_fk foreign key (plaque) references Vehicles(plaque);
- alter table Sales add constraint sal_ag_sales_fk foreign key (sales_agent_id) references Sales_Agents(sales_agent_id);
- create sequence cus_id NOMAXVALUE NOCYCLE;
- create sequence sal_ag_id NOMAXVALUE NOCYCLE;
- create synonym Vehi for Vehicles;
- create synonym Sal for Sales;
- create synonym Sal_Age for Sales_Agents;

3 Pre-Assessment:

- Practices pre-Assessment for Database Systems Laboratory II

Practice	Pre-Assessment
COMPLIES WITH THE REQUESTED FUNCTIONALITY	X
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)	100

Table 1: Pre-Assessment.

4 Conclusion:

Creating a database in conjunction with more people can be very beneficial, or very harmful, due to the fact that not everyone has the same abstraction with which to solve a problem. If all parties make mistakes or everything is well stunned from the first step, this will affect the final result, since it will be based on all the other aspects made to create the final product.