Data base Restaurant Report

Aurélien GARNIER

Adam NOURI

Dan LELLOUCHE

Médéric ROYER

Table des matières

1) Student checklist

Task			Achieved
You must have a copy of assessment one paperwork uploaded to blackboard – this is for the external assessor, who can override my marks or fail you if it is not there for him/ her to see.			
Using your database provide the following information			
50%	1) Customer Names in alphabetic order	5%	
	2) Contact information of customers (Name and Contact)	5%	
	3) Create a booking	5%	
	4) Update a booking	5%	
	5) Add a new meal	5%	
	6) Update the cost of a meal	5%	
	7) Number of bookings for one working day	10%	
	8) SQL Complexity and Structure	10%	
Create user stories for the case study and include the SQL requirements above		10%	
MoSCoW Assessment for a) the system and b) the database		10%	
UseCase diagram to reflect the above requirements		10%	
UseCase documentation to reflect the above requirements		10%	
ERD diagram to include above requirements		10%	
A fully working database which achieves all the above requirements and contains sufficient data to demonstrates its functionality			
Everyone in your group has a full working database on their university Oracle account. If this is not achieved the external assessor can override my marks and can fail you.			
Everyone in your group must upload a copy of the paperwork to blackboard – this is for the external assessor, who can override my marks or fail you if it is not there for him/ her to see.			
Presentation			
Give me a printed handout at the start of the presentation			
A brief description of the following:			
• MoSCoW assessment			
• User Stories			
• Requirements (and assumptions) 10 or more			
• ERD and Usecase			
• How your database achieves the three normal forms of database design			
• How your database achieves simple data integrity			
• Explain the ACID principle in relation to your database			
• Explain how the collected data is governed by the GPDR principles			
• Demonstrate your working SQL			

2) List of User Stories For Each Actor

Customer

1) As a customer I want to tell the restaurant how many we are so that we can have the good number of seats.
2) As a customer I want to reserve tables for later so that I don't have to come just right now.
3) As a customer I want to change my booking so that I can change my mind about the hour or number of people.
4) As a customer I want to cancel my booking so that I can change my mind about eating in this restaurant.
5) As a customer I want to book a table without any reservation.
6) As a customer I want to be notice by mail or by phone that my table is booked so that it's more convenient.
7) As a customer I want to have access to the menu so that I don't waste time.
8) As a customer I want to know the price of the meals so that I know how much to bring.
9) As a customer I want to make a booking for more than 4 people so that I can invite more friends.
10) As a customer I want to know the restaurant opening hours so that I don't waste time because looking for an opened restaurant is a big loss of time so I want the restaurant to be clear about it.

1) As a restaurant employee I want to assign people to a table so that I can fill the restaurant.
2) As a restaurant employee I want to see what tables are free or not so that I can manage my bookings easier.
3) As a restaurant employee I want to see what tables are reserved for later so that I will not have two bookings at the same table.
4) As a restaurant employee I want to see all the tables ON THE SCREEN so that I have a good visibility of the restaurant.
5) As a restaurant employee I want to change someone's place or arrival time so that the customer can change his mind.
6) As a restaurant employee I want to cancel a booking so that a table gets free when a customer changes his mind.
7) As a restaurant employee I want to be able to call the customer in case of any issue for his booking so that I can free a table.
8) As a restaurant employee I want to assign a meal to a booking so that we have an historic of what people like to eat (and what they don't like).
9) As a restaurant employee I want to add the price of each booking so that I have a visibility of the finance of the restaurant.
10) As a restaurant employee, I want to be able to free a table when a customer finished so that new customers can eat at this table.

3)   Use Case Diagram

Use Case Diagram Restaurant

Mederic Royer | October 28, 2019
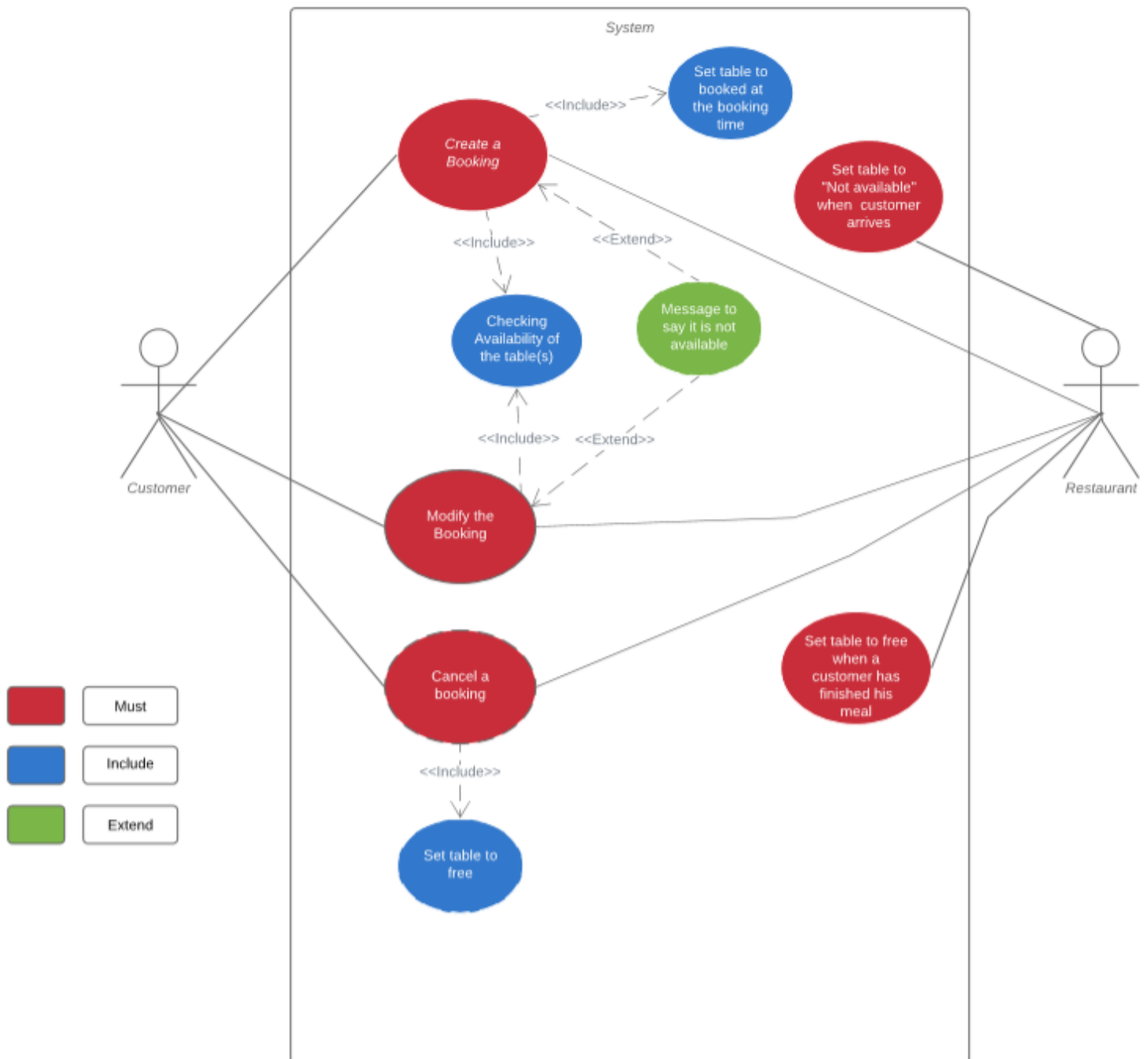
4) Use Case Documents

a) Use Case name: Reservation

<u>As a customer I want to reserve tables for later so that I don't have to come just right now.</u>

Actors:

- Customer
- Restaurant

Triggers:

The customer asks for a table at a given time.

Preconditions:

The customer has called the restaurant to ask for a table

Post conditions:

The booking has been validated and inserted into the system

The customer gives his personal details

The restaurant gives the booking information's to the customer

Normal Flow:

1) The customer asks for a booking with date, time and a number of people
2) The restaurant checks the availability of the request in the system
3) The system shows all the available tables at given times
4) The restaurant confirms the availability of the request
5) The customer validates his reservation
6) The restaurant saves the table booked into the system

Alternate Flows:

3A1: the requested time is not available, so the restaurant proposes other options that the client accepts.

1) The system shows all the available tables at given times
2) The table is not available at the time requested, the restaurant proposes alternative choices
3) The client picks another booking time from the alternative choices
4) The Restaurant confirm the choice
5) The customer validates his reservation
6) The restaurant saves the table booked in the system

3A2: **:** the requested time is not available, so the restaurant proposes other options that the client does not accept

1) The system shows all the available tables at given times
2) The table is not available at the time requested, the restaurant proposes alternative choices
3) The customer does not find other suitable options and declines his booking

Conclusion:

This use case example shows how a use case can be developed with increasing detail, starting with a simple reservation, mentioning every alternative case. First, the client asks for a booking, then the restaurant checks if the booking can be accepted or not.

<u>As a customer I want to change my booking so that I can change my mind about the hour or number of people.</u>

Actors:

- Customer
- Restaurant

Triggers:

The customer cancels his booking

Preconditions:

The customer has already booked a table

The customer has called the restaurant to cancel his booking

Post conditions:

The booking has been canceled with success

The restaurant frees the table booked to receive new customers.

Normal Flow:

7) The customer wants to cancel his booking
8) The restaurant asks for the customer booking details
9) The restaurant frees the old table booked
10) The restaurant validates the canceling

Alternate Flows:

No alternative flows

<u>**Conclusion:**</u>

This use case example shows how a customer can cancel a booking. First, the client asks for canceling, then the restaurant validates the reservation cancellation.

As a customer I want to book a table without any reservation

Actors:

- Customer
- Restaurant

Triggers:

The customer asks for a table on site without any reservation.

Preconditions:

The customer is in front of the restaurant and asks for a table.

Post conditions:

The restaurant gives a free table.

The table has been booked into the system.

Normal Flow:

1) The customer asks for a table without any reservation
2) The restaurant checks if there is a free table for his request.
3) The system shows all the available tables at given times
4) The restaurant confirms the availability of the request
5) The restaurant welcomes the customer to the restaurant
6) The restaurant saves the table booked in the system

Alternate Flows:

3A1: The restaurant is full for this day.

1) The system shows all the available tables at this time
2) The table is not available at any time on this day
3) The restaurant declines the request

3A2: The restaurant is full for this time, so the restaurant proposes other options that the client does not accept

1) The system shows all the available tables at given times
2) The table is not available at the time requested, the restaurant proposes alternative choices
3) The customer does not find other suitable options and declines his request.

3A122123: The restaurant is full for this time, so the restaurant proposes other options that the client accepts

1) The system shows all the available tables at given times
2) The table is not available at the time requested, the restaurant proposes alternative choices
3) The customer finds a suitable option and confirms his booking.
4) The restaurant saves the table booked in the system

3A122123: The restaurant is available for this time, but with conditions that the client accepts.

1) The system shows all the available tables at given times
2) The table is available at the time requested, but shows that other clients have booked this table for later
3) The restaurant proposes the condition to the customer (that he has to free the table at a given time)
4) The customer finds this option good enough.
5) The restaurant welcomes the customer
6) The restaurant saves the table booked in the system

3A122123: The restaurant is available for this time, but with conditions that the client declines.

1) The system shows all the available tables at given times
2) The table is available at the time requested, but shows that other clients have booked this table for later
3) The restaurant proposes the condition to the customer (that he has to free the table at a given time)
4) The customer declines the option.

Conclusion:

This example of use shows how a customer can book a table on site without a reservation. First, the client asks for a table. Then, the restaurant checks whether the reservation can be accepted or not based on the reservations that the system has already recorded.

d) Use Case name: Modify a reservation

As a customer I want to change my booking so that I can change my mind about the hour or number of people

Actors:

- Customer
- Restaurant

Triggers:

The customer modifies his booking for another time/number of people

Preconditions:

The customer has already booked a table

The customer has called the restaurant to modify his booking

Post conditions:

The booking has been modified with success

The restaurant gives a new booking information to the customer

Normal Flow:

7) The customer asks for another booking with different time/number of people
8) The restaurant asks for the customer booking details
9) The restaurant checks the availability of the request in the system
10) The system shows all the available tables at given times
11) The restaurant confirms the availability of the new request
12) The customer validates his new reservation
13) The restaurant frees the old table booked

Alternate Flows:

3A1: the new requested time/number of people is not available, so the restaurant proposes other options that the client can accept

4) The system shows all the available tables at given times
5) The table is not available at the time requested, the restaurant proposes alternative choices
6) The client picks another booking time from the alternative choices
7) The Restaurant confirm the choice
8) The customer validates his reservation
9) The restaurant frees the old table booked

3A2: : the new requested booking is not available, so the restaurant proposes other options that the client does not accept

4) The system shows all the available tables at given times
5) The table is not available at the time requested, the restaurant proposes alternative choices
6) The customer does not find other suitable options and declines his booking
7) The restaurant frees the table booked

5A1: The new requested booking is not available, so the restaurant proposes other options that the client does not accept, and the client keeps his first booking

1) The system shows all the available tables at given times
2) The table is not available at the time requested, the restaurant proposes alternative choices
3) The customer does not find other suitable options and goes back to his first booking

<u>**Conclusion:**</u>

This use case example shows the different client options when he wants to modify his booking. First, the client asks for a new booking, then the restaurant checks if the booking can be accepted or not.

5) One Example of a Sequence Diagram
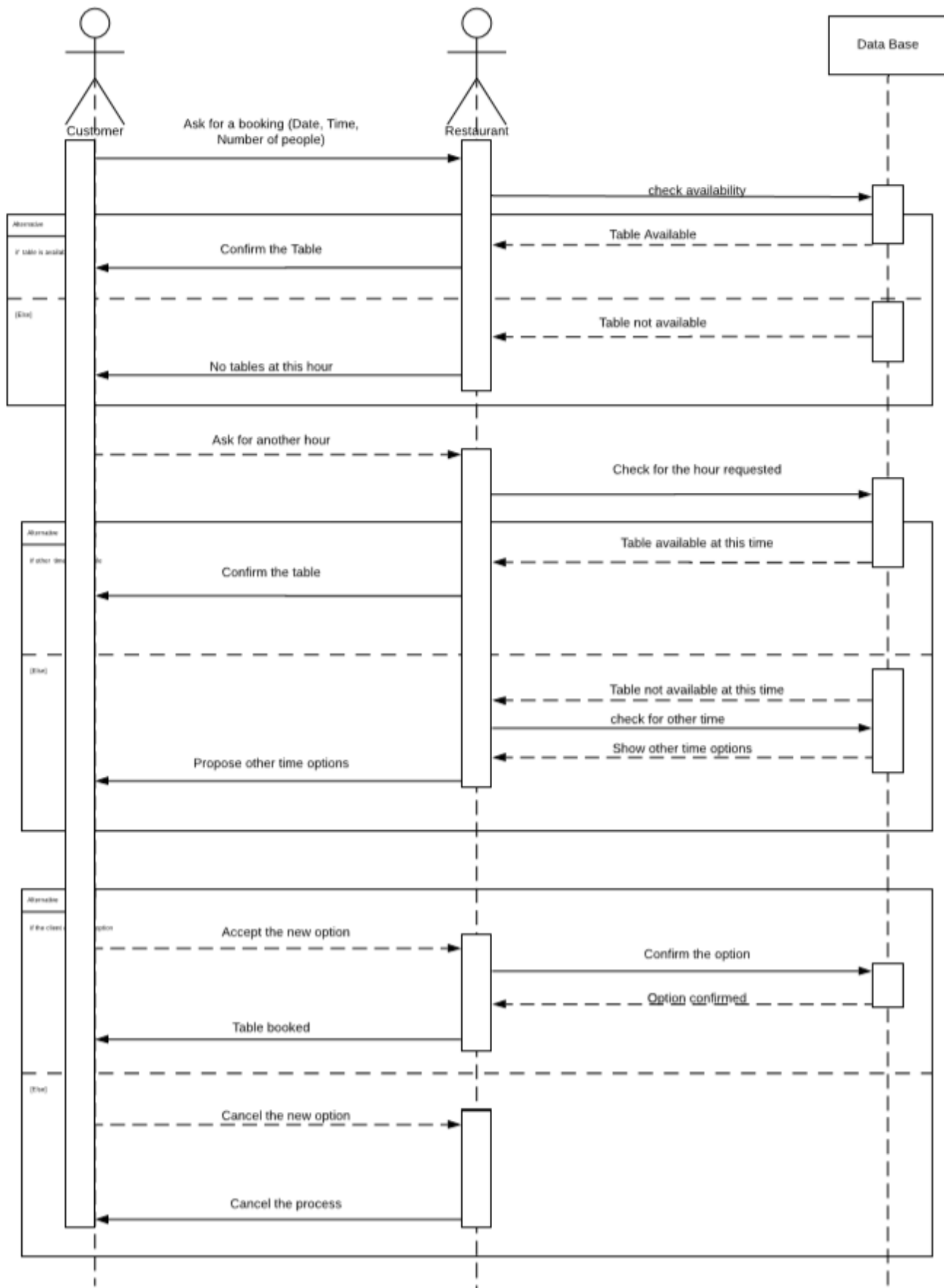
6)   Assumptions

Database:
-    Need First Name, Last name and phone number to make a booking (in order to contact the customer if it is needed)
-    The birth date is not necessary (unless the customer wants to organize his birthday, then the manager would act consequently)
-    for each meal, we offer two shifts of service
     o   for lunch: from 11:30 to 13:00, then from 13:00 to 14:30
     o   for dinner: from 7pm to 8:30pm, then from 9pm to 10.30pm
-    We ask for the start time in order to organize the shifts
-    The tables are randomly assigned to the different clients (unless they make an on-site request)

System:
-    The restaurant sends two reminders for the customer: one the day before, one 20mn before the booking time
-    The reservations can only be made during the restaurant opening hours (the customers might as well send an email or leave a message)
-    The servers or anyone in charge of welcoming the clients can be the decision-maker in making a reservation, modification or cancellation.
-    If a client wants a table, he can ask to the managing server, who can choose to accept the request or not (regarding the availability of the different tables)

7)   GPDR Requirements

-    Ask first name, last name and number phone of the customer. The name of the customer allows us to verify his identity once he comes at the restaurant and claim have a reservation. Then, the number phone allows the restaurant to send messages to the customer to remind him he has a reservation. And if needed, call the customer if the restaurant itself has an issue. So, the restaurant asks only for the required data in order to let the system work.
-    The customer can ask the restaurant to erase his personal details.
-    The restaurant isn't allowed to give personal details about a customer (phone number, name)
-    The data base is kept updated. After 2 years, the customer is called to check his personal details.
-    During the staff's formation, the confidentiality of data is explained. So that the staff knows what can happen if data is stolen or diffused illegally.

8)   The ACID principle

<u>Atomicity:</u>

- Definition: Transactions are all or nothing.
- Example in our data base:
    - Every information has to be filled in order to validate a booking.

<u>Consistency:</u>

- Definition: Only valid data is saved.
- Example in our data base:
    - A booking can be made only if a table is available, otherwise the waitress will propose another date or cancel the demand.
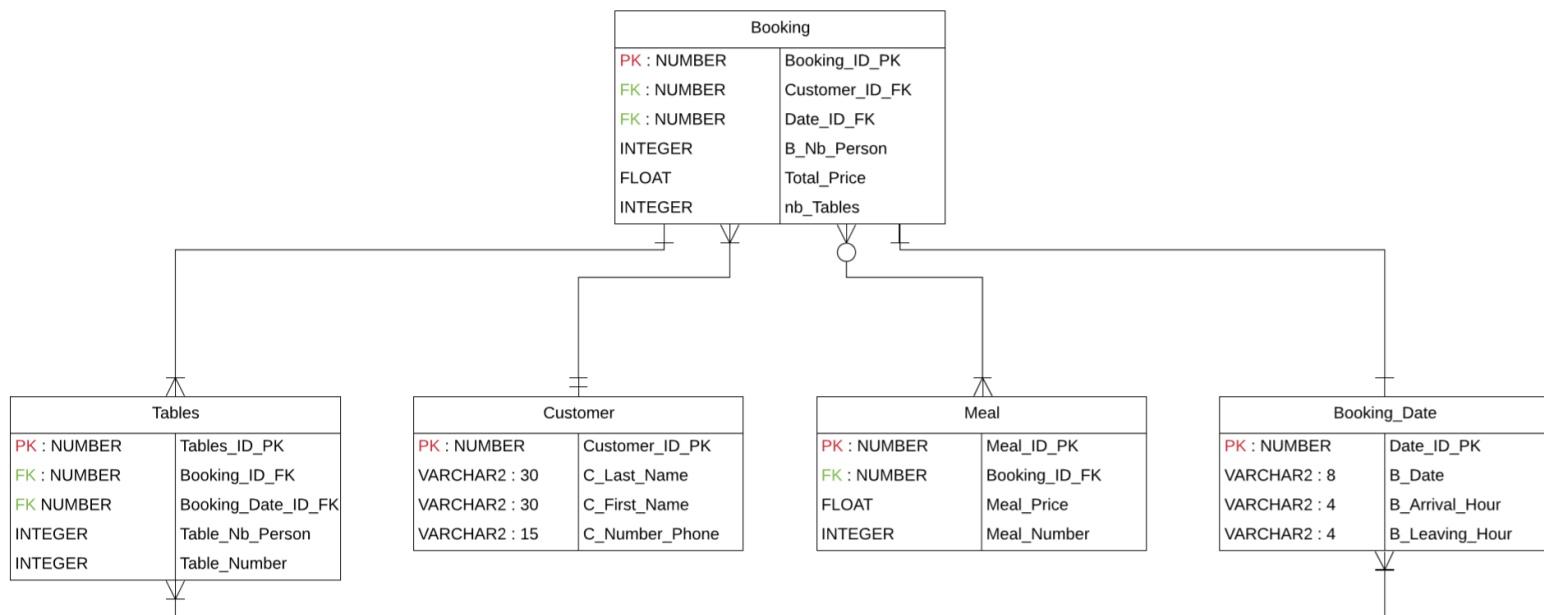
<u>Isolation:</u>

- Definition: The intermediate state of a transaction is invisible to other transactions.
- Example in our data base:
    - While all the data are not full filled, we can't see the booking in the data base.

<u>Durability:</u>

- Definition: Written data will not be lost.
- Example in our data base:
    - Data is durably stored in the different entities.

9) ERD Diagram

10) Three Normal Forms of Database Design & Data Integrity

First Normal Form (1NF):

For a table to be in the First Normal Form, we needed to follow the 3 rules:
1) It must have only attributes/columns of simple (atomic) value.
2) The values stored in a column must have been of the same domain.
3) All columns in a table needed to have a unique name.

In this form, the order in which the data is stored did not matter.

Second normal form (2NF):

For a table to be in the second normal form:

1) It should be in the First Normal form
2) And, there should be no partial dependence.
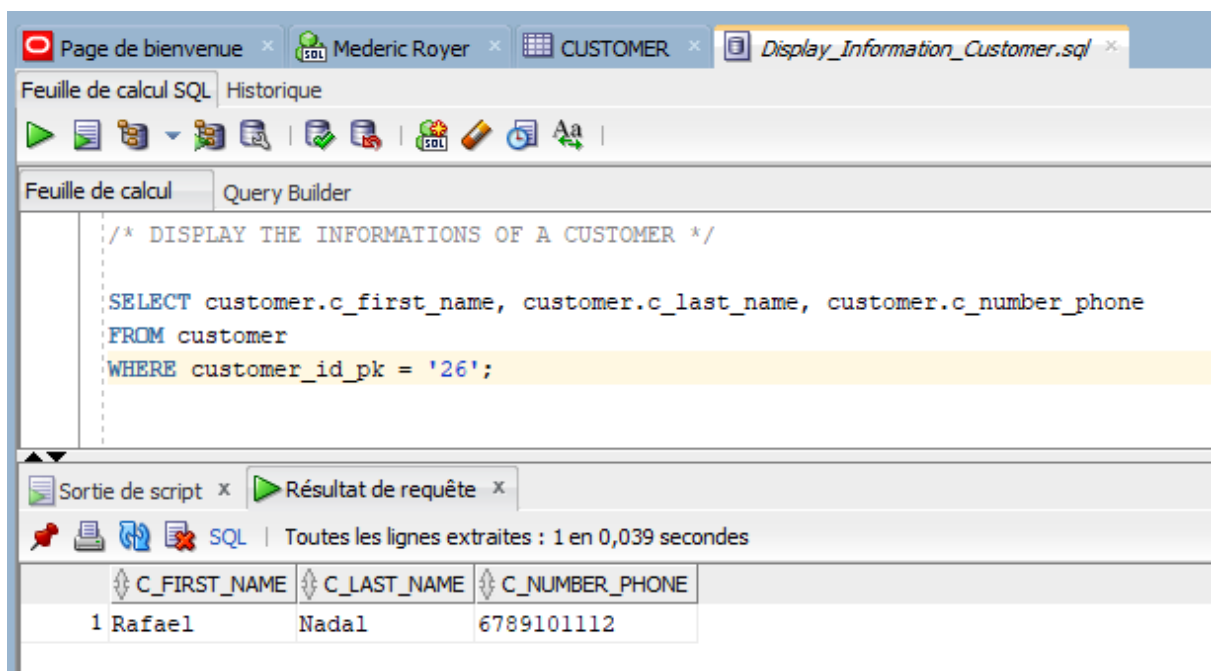
Third Normal Form (3NF)

For a table to be in the third normal form:

1) It should be in the second Normal form
2) It contains only columns that are non-transitively dependent on the primary key
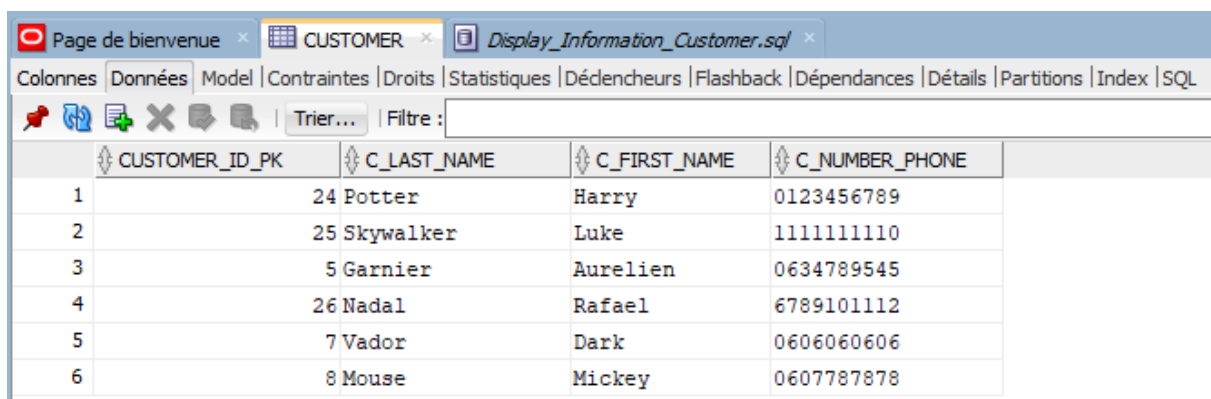
11)   Our Working SQL

Command to display a customer's details.

Command to display booking's information

Display the names in the alphabetic order

Display the different tables taken at a certain time.


```sql
/* DISPLAY THE TABLES TAKEN AT A CERTAIN TIME */

SELECT tables.table_number, booking_date.b_date, booking_date.b_arrival_hour
FROM tables
JOIN booking_date ON tables.booking_date_id_fk = booking_date.date_id_pk
WHERE booking_date.b_date = '20-12-19' AND booking_date.b_arrival_hour = '2100';
```

Résultat de requête ×

SQL | Toutes les lignes extraites : 3 en 0,042 secondes

	TABLE_NUMBER	B_DATE	B_ARRIVAL_HOUR
1	1	20-12-19	2100
2	5	20-12-19	2100
3	6	20-12-19	2100