24303 DATABASES
Lab02 - Northwind Supermarket

Goals: The lab will cover the following aspects:

a. Understand a given Relational Model
b. Implement queries in SQL (Data Manipulation Language – DML)
c. Implement PL/SQL requirements: Procedures, Triggers and Events

Probably, you should have to use additional study time to complete this lab. Students are encouraged to check all the material in detail before submission.

Note: The University has provided the required software to develop this practice, which is properly installed in the computers of the laboratory room classes. You will always have the option of working on the university computers. Furthermore, the professors have already provided installation guides and manuals to facilitate the installation in student's personal computers which are under student's responsibility.

Methodology:

This Lab02 is divided into three class sessions.

The statement of the lab must be read in order, following the instructions given step by step.

Session 1:

- Understand a given database design relational model (**northwind_relational_model.pdf**).
- Create and populate the database using the scripts for creation of the database tables and data population (**creates_northwind.sql** and **inserts_northwind.sql**).
- Create a new SQL file for queries (**queries_northwind.sql**) and write your own queries based on the lab statement.

Session 2:

- Provide improvements to the database with a new Relational Model and deliver it in a pdf file: **northwind_new_models.pdf** with justifications and also the matching Conceptual model of your proposal.
- Start working on the list of new requirements for the database that will consist in creating procedures, trigger and events structures. You should create an SQL file for each: (**req1_northwind.sql**, **req2_northwind.sql […] reqN_northwind.sql**). One for every new requirement.

Session 3:

- Continue working on the implementation of the new requirements (procedures, triggers and events as asked).

Submission:

You must implement and submit:

1. A pdf document with your improvements to the **Relational Model** and the argumentation of the changes.
2. Multiple SQL scripts:
    a. Solution for the proposed queries.
    b. Solutions for DB requirements (each one in a different file).

This practice will be done in groups of TWO or THREE people. The practice will be developed in two sessions and the teacher will assist you if you have technical problems for its development. **THE TEACHER WILL NOT PROVIDE ANY SOLUTION**.

Each group will submit the following files:

1. **queries_northwind.sql**
2. **northwind_new_models.pdf**
3. **reqN_northwind.sql** ***(one for every requirement)***

All in a ZIP file. The ZIP file should be called with the NIA of each member of the group (for example, NIA1_NIA2.ZIP for groups of 2 people).
ONLY A MEMBER OF THE GROUP WILL BE RESPONSIBLE FOR RETURNING THE WORK and always after the third session of Lab02.

EACH WORKING GROUP WILL DELIVER * ONE VERSION *** OF THE PRACTICE**

It is really important to follow each point of the submission instructions. Otherwise, the practice will not be properly qualified, that is, it will be evaluated with a 0. **Please, before your submission check the following requirements:**

- All your scripts work BEFORE your submission
- You have defined your tables and fields using the names specified in this document.
- The file scripts must include the UTF-8 codification.
- The files scripts must follow the names specified in this document.

Deadline: See Aula Global.

1. Introduction

This lab02 is quite different from the previous labs and may be more difficult in the proposed exercises but the database model is easier than the one in the Lab01 since you do not need to create conceptual or relational models from scratch, but you need to understand the concepts in the provided ER and RM diagrams.

Please read the lab statement in order and work through the exercises in order.

Using the given scripts, the **first step** is to understand the database model and the **second step** is to create the model as a database and populate it.

At the **third step** is where your work starts by developing the solutions for the asked queries.

The **fourth step** is to analyse the database and find errors or optimizations and the **fifth step** will be to work on how to correct those errors only in the model. And translate your relational model to the Conceptual one.

Sixth and last step, will be to work on the new requirements in developing the PL SQL code for some Procedures, Triggers and Events.

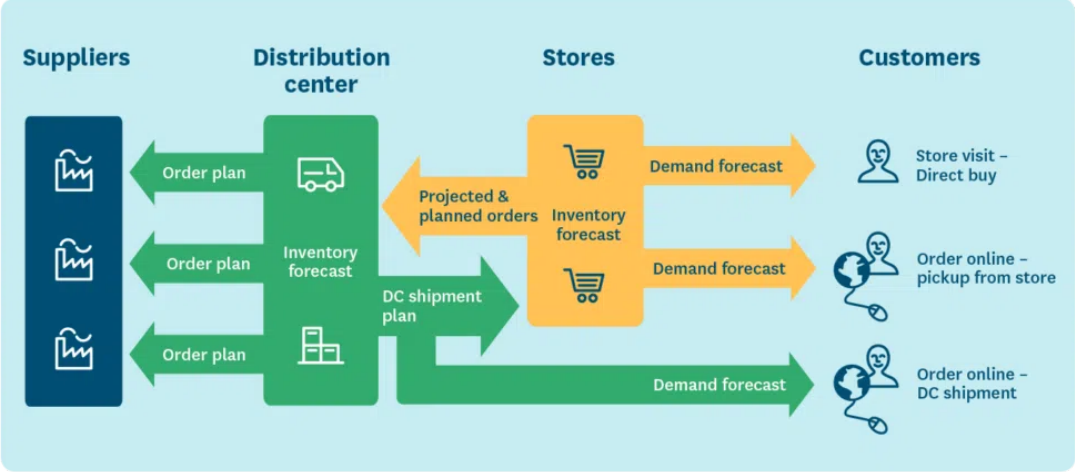
2. The database

Northwind is a company that has a small business-to-business retail that is growing. This means that Northwind is a provider of goods for other companies. They provide mainly groceries. Nowadays, his business model is based on an old-fashioned sell-by-phone and their owners would like to add an online ecommerce store to it.

Since they are not experts on the matter, Northwind hired us from UPF Campus Poblenou to develop some solutions to their business model.

They need a proposal about how its information will be properly stored and how the synchronisation between the actual offline business model and the new online channel would work. The new proposal should contain the previous design and develop the needed improvements since it should be able to maintain the old and the business model in parallel.

The keys of a retail business are **Customers**, **Products** and **Orders**. But Suppliers, Employees and Shippers are also important because without them, the rest wouldn't work and we couldn't attend to our Clients properly.



Note: *If you want to learn more about supply chain management for grocery stores, click here.*

As we mentioned, the **Customers** of Northwind are some other companies. For them they store the company name, basic contact person information (name and title), full address information, and phone and fax numbers.

Regarding Northwind' **Products**, we need to explain that some of the products are composed of several units. For instance, the customers will purchase a bundle/package of items as a complete product. It doesn't mean that you should consider every single unit in the bundle, just consider the whole bundle as a product.

Northwind database actually includes the following information about their products: name, number of units that the product contains, price, units of the products in stock, on order and their reorder level, and whether the product is discontinued or not.

Products are organised into categories. For each category, Northwind needs to store the name, a description and an image (just the image filename). The image file name cannot be duplicated. Each product has only one category.

In the case of **Orders**, it is important to store information about the date of creation, the date requested by the final customer, the date of shipment, the complete information of the shipping address and total weight of the order.

Is important to mention that Orders contain Products. It is required to store the different products that the purchase order contains and for each product: unit price, quantity and discount. We are aware that the same product can only appear once per each Order.

For the **Suppliers**, Northwind is storing the same information as for their Customers plus the company's website; even though the information of Suppliers and Customers should be stored separately.

Furthermore, each product can only be provided by a specific supplier. But a supplier can provide more than one product.

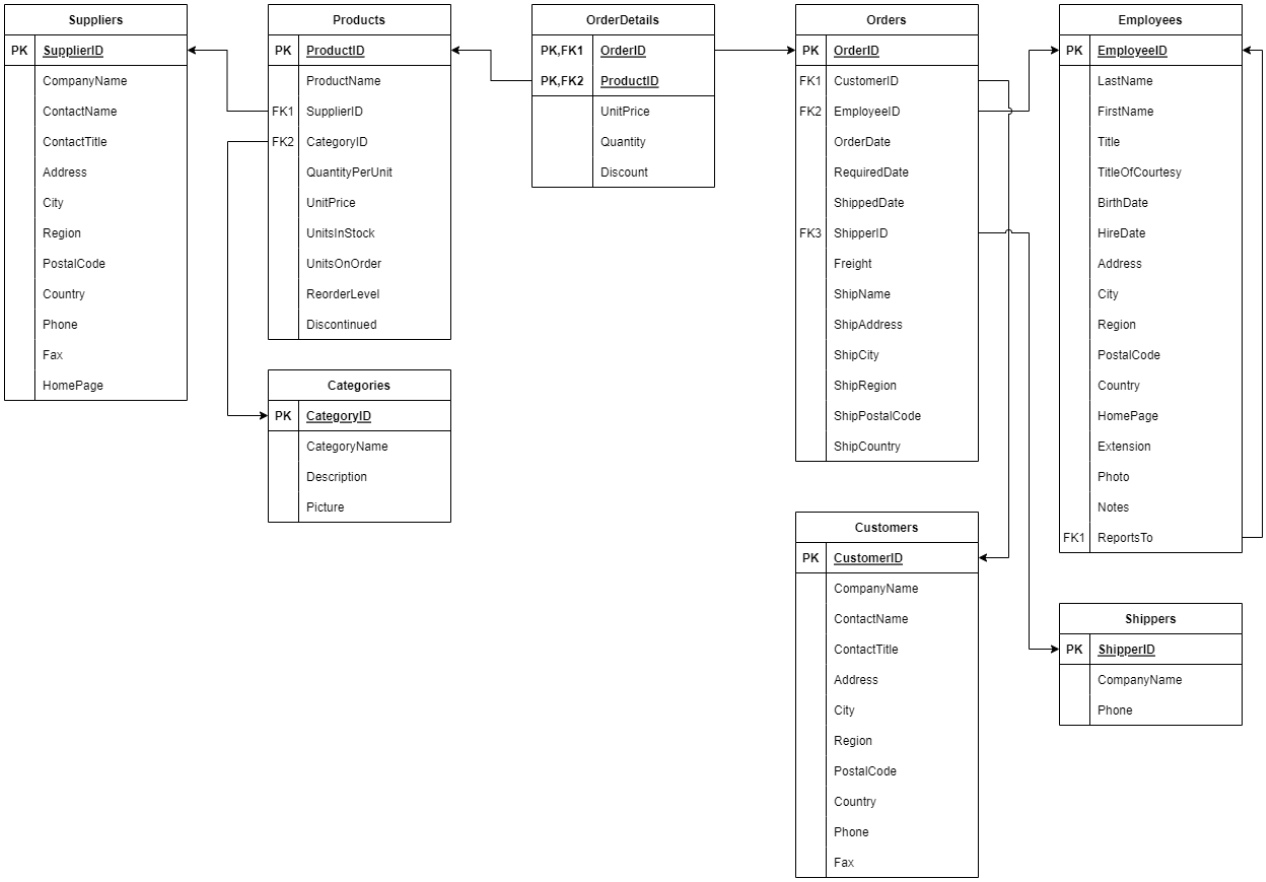
Another important key point of the business is the **Employees**. They are responsible for receiving the purchase order from Customers and entering it into the system. In order to optimise the system, a given employee uses to insert orders for the same certain group of customers.

In depth, it is necessary to store first and last name (separately), title and courtesy title (also separately), date of birth, date of hire, full address information, website, extension, photo and generic notes.

Also, the customer's business hierarchy requires that each employee has a supervisor. All but one, the boss who wouldn’t have any supervisor above him/her. Note that several employees may report to the same supervisor but they won’t have more than one supervisor at a time.
About **Shippers**, Northwind only needs to store the company name and the phone number.

3. Understanding the previous design

See below the actual beta 1 version of the Relational Model of Northwind's database:



4. Creating and populating the Database

Now, you are at the point where you understand the model and you need to create it and fill it with information.

On the Aula Global you will find two files (**creates_northwind.sql** and **inserts_northwind.sql**), download and execute them on your local machine through Workbench or terminal.

Check the following information that details the expected row count for each table of the model:

Table	COUNT of rows
Shippers	3
Categories	8
Customers	91
Employees	9
Suppliers	29
Products	77
Orders	830
OrderDetails	2155

5. Queries (40 points)

In this phase you will need to create a new sql file (**queries_northwind.sql**). You are asked to implement SQL queries for the following statements *(note that in some cases subqueries are needed to solve them)*. For each query you deliver, create an associated view. Everything should be informed into the view sql code: filters, operations, joins, etc.
For instance, for the first query:

```
DROP VIEW IF EXISTS query_1;

CREATE VIEW query_1 AS
      SELECT whatever
       FROM dummy
       WHERE lorep = ipsum;
```

1. Get the supplier with the biggest amount of different provided products. Return the supplier name and the count of different provided products.

2. Get the first five employees with the highest number of orders. Show the Employee's ID's FirstName, LastName and their count of orders. Sort the result by the count of orders descending.

3. Display all information stored at the database for all the products of the Supplier named "Grandma Kelly's Homestead". Use a single query (with subquery if needed).

4. Show the count of different served Products for each Order, also show the name and the last name of the Employee who placed the Order.

5. Show the product name, quantity and average unit price for each product sold.

6. Show complete information for orders placed in 1997 and shipped in the same year by German customers. Sort the result by customer and order date ascending.

7. Return the order identifier and the date for all the Orders which contain Products that belong to the "Beverages" Category. Sort the results by date descending. Do not use the Category Identifier, use "Beverages".

8. Locate the Order with OrderID 10255 and calculate its total cost adding the UnitPrice of all the contained products inside it.

9. Show the complete information for orders containing Products from Japanese Suppliers. Sort the result by customer and order date ascending.

10. Show the cheapest and the most expensive product(s) *(Use only a single query)*.

11. Return the address, city, postal code and country of all Clients. All fields in the same return field, that is, the previous four fields in a single column (find a function to help you concatenate those values in a single field). Also show the client's identifier and name in two other single columns.

12. We want to know who are the employees with more processed orders than the employee with id number 8. Show their personal information.

13. Show the orders with more than three different products inside.

14. Return the orders of customers located in London and for the suppliers in New Orleans.

15. Show the name of the products, their category and the price, only when its price is above 20 and they are linked to the category with the minimum number of registered products.

16. Show the complete name of the employees who processed the highest number of orders. (Note: There could be more than one employee with the same record, so deliver a query that accepts more than one result).

17. Return VIP customers (those with more than 25 registered orders and worked also with more than 2 shippers and with 4 or more employees).

18. Show the names of the employees who worked with all the registered shipping companies.

19. Return the order ID's that are required more than 7 days after their order date to process. You should find a function to help you calculate between two dates.

20. For every order detail, return the product name, supplier name, category name, employee first and last name, shipper company name, and customer company name. Order all the results depending on the OrderID ascending, and then on the ProductName also ascending.

5.1. Correction criteria

The correct implementation of all queries will be evaluated **with 40 points**.
Queries 1 to 20 will count 2 points

In this context, 3 scenarios are possible:

- The query is correct and it shows the expected results. You get the maximum qualification for this query.
- The query is not correct enough but it does not return any error. That is, the output is close to the expected result. In this case, you get half the maximum qualification. For example:
    - Some fields are missing.
    - The sorting is not correct.
- If the query presents syntax errors or is not able to be executed, you do not get any mark for this query.

6. Analysing the database (10 points)

The database has two problems. One in an undetected entity hierarchy and the other one are repeated attributes that can be optimised with some new entities that promote data reuse.

Get the **northwind_models.drawio** (AG) where you will find the relational diagram and make the corrections.

You must send a PDF file (**northwind_new_models.pdf**) with the explanations/justifications about your correction proposals and the new conceptual and relational schemes. You do not have to apply those changes on the database.

6.1. Correction criteria

This part will be evaluated **with 10 points**. On this terms:

- New relational model **2 points**.
- Conceptual model based on your new relational model **3 points**.
- Justifications and explanations of the changes applied **5 points**.

7. New requirements for the database (50 points)

(…Procedures, Triggers and Events coming soon…)