



---

# 2nd Evaluation Exam Model B

---

## NSM/ASIR. Database Management (DBM)

---

Pau Miñana

2023-2024

---

# INDEX

---

## 1. PART 1: Relational databases (9 points)

### 1.1. Queries and views (4 points)

Query 1 (1 point)

Query 2 (1 point)

Query 3 (1 point)

Query 4 (1 point)

### 1.2. Implementation of scripts (3.5 points)

Script 1 (1.75 points)

Script 2 (1.75 points)

### 1.3. User management and data security (1.5 points)

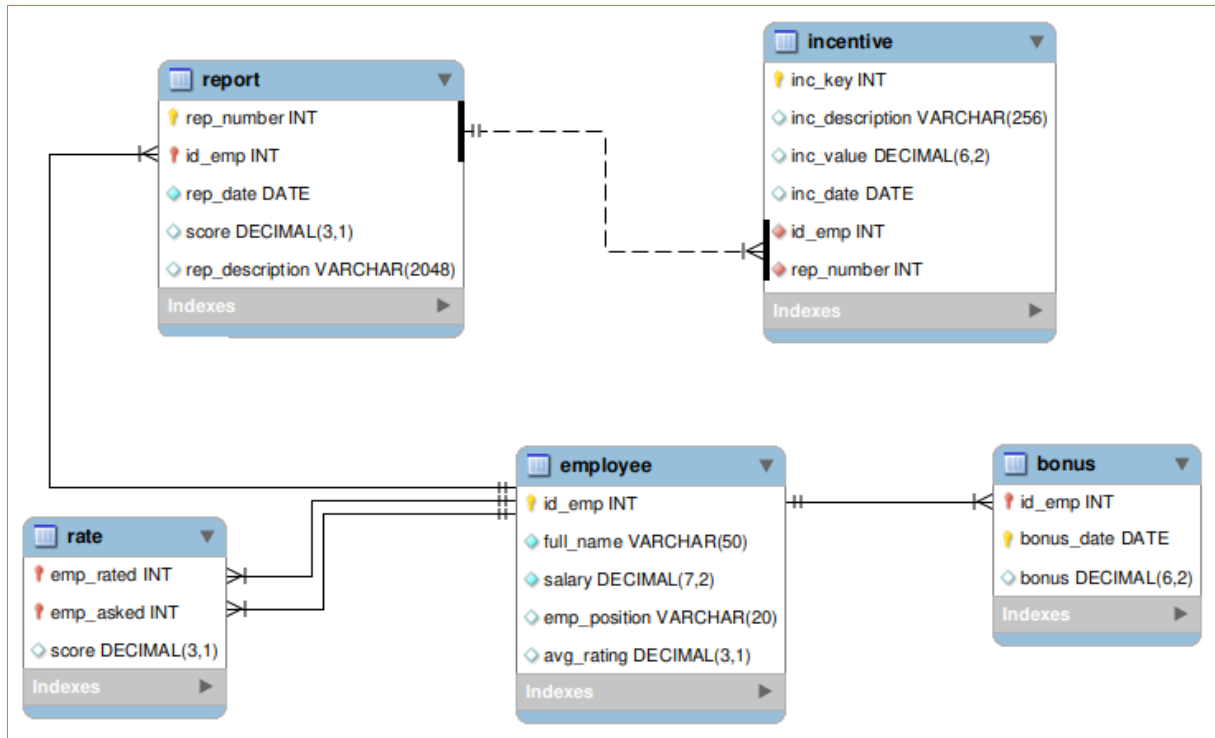
User management (0.75 points)

Data security (0.75 points)

## 2. PART 2: NoSQL databases (1 point)

## 1. PART 1: Relational databases (9 points)

This is the physical model of a database that stores information about employees. The database is composed of the following tables:



You can find the script to create the database in the Virtual Classroom.

- Use the **notation seen in the course**.
- Use indentation to make the queries more readable.
- Use alias when needed.

### 1.1. Queries and views (4 points)

|| Spend no more than 1 hour of the exam on this exercise.

#### Query 1 (1 point)

Show the *rep\_number*, *id\_emp* and *score* of the reports produced in 2023 that have caused an incentive of 120.00 *inc\_value* or are evaluating employees that have not 'Manager' *emp\_position*, sort by *score* from lowest to highest.

#### Query 2 (1 point)

Show the *full name* and *salary* of the employees that have not received any bonus and do not appear in any report. Order the list by *salary*, from highest to lowest, and *full name*.

### Query 3 (1 point)

Create a view that shows the *id\_emp* and *full\_name* of **all** the employees and the number of rates that they have received. Order the list by that number of rates, from highest to lowest.

### Query 4 (1 point)

Show the *emp\_position*, *full\_name* and the *salary* of the employee/s with the lowest *salary* of each *emp\_position*; show only the results when that *salary* is higher than the highest *bonus* received by the employees with 'Manager' *emp\_position*. Sort by *emp\_position* in reverse alphabetical order.

## 1.2. Implementation of scripts (3.5 points)

|| Spend no more than 45 minutes of the exam on this exercise.

### Script 1 (1.75 points)

Create a procedure that receives an *id\_emp* and shows a number of tables equivalent to the number of reports about that employee. Each table will contain the *rep\_number*, *inc\_data* and *inc\_value* of the incentives that the employee has received due to that report (it can show an empty set if there are no incentives related to that *rep\_number*). The procedure should "return" the number of existing reports about that employee. If the employee does not exist, the procedure will generate an error.

### Script 2 (1.75 points)

Create a function that receives an *id\_emp* and returns the average of all the rates that employee has received (not the *avg\_rating* field) if the employee is found on the database. If the employee has not received any rate the function can return NULL value. Then use the function to **UPDATE** the field *avg\_rating* (that has random data right now) with the correct number, for every employee on the database.

### 1.3. User management and data security (1.5 points)

|| Spend no more than 30 minutes of the exam on this exercise.

#### User management (0.75 points)

Create a new user with the following characteristics:

- Username: Your first surname
- Password: Your Name
- The user is allowed to connect from anywhere.
- Permissions: *SELECT* on the *employee* and *rate* tables.
- The user can give its permissions to others.

Create a role called *emp\_func* that has the permission to *EXECUTE* the function created in the *Script 2*.

Give the user the role *emp\_func* so it can be used without further configuration.

#### Data security (0.75 points)

Write 2 MySQL scripts, one to export all the data from the *incentive* table to a file called *incentive\_data.csv* and another to replace the data on the *incentive* table using that file. Explain the necessary configuration requirements to ensure the scripts can be executed without any problem and write the command to launch the scripts on the command line of the terminal.

## 2. PART 2: NoSQL databases (1 point)

|| Spend no more than 15 minutes of the exam on this exercise.

- Create a database in MongoDB called *EmpDB* with a collection called *employee* that includes the following fields with the following values:

<i>_id</i>	<i>full_name</i>	<i>score</i>	<i>reports</i>	<i>emp_position</i>
3	Alice Smith	6.5	ARRAY	Tester
9	Grace Red	4.0	ARRAY	

The *bonus* field is an array of documents with the following data:

<i>employee (_id)</i>	<i>number</i>	<i>incentive</i>
3		
	1	150.00
9		
	1	
	2	100.00

- Show the *full\_name*, and all the *incentive* for the employees who have at least 2 reports or have a known *emp\_position* and a *score* greater than 6.0. Order the answer by *score*, from highest to lowest, and *full\_name*.