ALEJANDRO ALMAGRO TORREGROSA **21013980F**

EXAMEN 2 EVALUACION BASES DE DATOS

ALEJANDRO ALMAGRO TORREGROSA **21013980F**

1.1. Queries and views

Query 1

```
SELECT r.rep_number, r.id_emp, r.score FROM report r
JOIN employee e ON e.id_emp=r.id_emp
LEFT JOIN incentive i ON i.id_emp=r.id_emp AND i.rep_number=r.rep_number
WHERE  rep_date = 2023
AND (i.inc_value = 120.00 OR e.emp_position <> 'Manager')
ORDER BY 3 ASC;
```

```
SELECT rep_number, id_emp, rep_date, score FROM report
WHERE rep_date = 2023;
```

He intentado comprobar si habia alguna rep_date de 2023 pero me seguia saliendo el reporte a nulo.

QUERY 2

```
SELECT e.full_name, e.salary FROM employee e , bonus b, report r
WHERE b.id_emp=e.id_emp AND r.id_emp=e.id_emp AND r.rep_number = 0
AND b.bonus = 0
ORDER BY 2, 1 DESC;
```

Esta primera no me retorna nada pero la segunda en cambio si me retorna campos.

```
SELECT e.full_name, e.salary FROM employee e
WHERE e.id_emp NOT IN (SELECT 1 FROM employee e1, bonus b, report r
                                    WHERE b.id_emp=e1.id_emp AND r.id_emp=e.id_emp)

ORDER BY 2, 1 DESC;
```

ALEJANDRO ALMAGRO TORREGROSA 21013980F

Query 3

```
CREATE OR REPLACE VIEW v_employee AS
SELECT e.id_emp AS employee, e.full_name,COUNT(emp_rated) AS cuenta FROM
employee e, rate r
WHERE r.emp_rated= e.id_emp AND r.emp_asked=id_emp
GROUP BY e.id_emp
ORDER BY 3 DESC;


SELECT employee FROM v_employee;
```

Me da row vacia.

Query 4

```
SELECT e.emp_position,e.full_name, e.salary FROM employee e
JOIN bonus b ON b.id_emp=e.id_emp
WHERE (SELECT e1.salary, e1.emp_position, MAX(b1.bonus) FROM employee e1,
bonus b1
                                        WHERE b1.id_emp=e1.id_emp
            AND MAX(b1.bonus) >= e1.salary
            AND e1.emp_position = 'Manager')
GROUP BY e.salary
ORDER by 2, 3 DESC;
```

He intentado hacerlo con una subconsutla pero me da error Invalid use of group function

ALEJANDRO ALMAGRO TORREGROSA **21013980F**

SCRIPT 1

```
DELIMITER $$
DROP PROCEDURE IF EXISTS emp_tablas $$
CREATE PROCEDURE emp_tablas(tb_idemp INT)
BEGIN

IF NOT EXISTS (SELECT 1 FROM employee WHERE id_emp = tb_idemp)
THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '\nUnexpected parameter\
n*****\
n====> Este Empleado no existe\n*****\n';
END IF;

SELECT i.rep_number, i.inc_data, i.inc_value FROM incentive i,  report r
WHERE r.id_emp=i.id_emp AND r.rep_number=i.rep_number
ORDER BY 1;

END$$

DELIMITER ;

CALL emp_tablas(1);
```

Me acepta la funcion pero me devuelve 0 rows affected.

ALEJANDRO ALMAGRO TORREGROSA **21013980F**

SCRIPT 2

```
DELIMITER $$
DROP FUNCTION IF EXISTS e_rate$$
CREATE FUNCTION e_rate ( e_emp INT)
RETURNS INT
NOT DETERMINISTIC
READS SQL DATA
BEGIN
RETURN (SELECT AVG(score) AS media FROM rate
        WHERE e_emp=emp_rated);

END$$

DELIMITER ;

SELECT e_rate(2);
```

1.3. User management and data security

User management

Create a new user with the following characteristics:

```
CREATE USER ‘ALMAGRO’@’%’ IDENTIFIED BY ‘alejandro’;

GRANT SELECT ON employeedb. employee  rate TO

‘ALMAGRO’@’%’ ;

GRANT privilege ON employeedb.* TO ‘ALMAGRO’@’%’  [WITH GRANT OPTION];

FLUSH PRIVILEGES;
```

Create a role called emp_func that has the permission to EXECUTE the function created in the Script 2.

```
CREATE ROLE emp_func

GRANT EXECUTE ON FUNCTION e_rate TO emp_func;

GRANT emp_func TO ‘ALMAGRO’@’%’;

FLUSH PRIVILEGES;
```

ALEJANDRO ALMAGRO TORREGROSA 21013980F

Give the user the role *emp_func* so it can be used without further configuration:

SET DEFAULT ROLE emp_func TO ‘ALMAGRO’@’%’;

Data security

exportar

util.exportTable('employeedb.incentive', '/incentive_data.csv', { dialect: 'csv-unix'})

Sera exportada en la carpeta que estemos situados en ese momento.

Importar

Set foreign_key_checks = 0;

REPLACE option deletes and replaces the id_emp and rep_number, violating the foreign key constraint on report table.

LOAD DATA INFILE ‘(Si estamos en el mismo directorio no haria falta poner nada sino /export/ o /Descargas/ como ejemplos)/incentive_data.csv’ REPLACE

INTO TABLE incentive

FIELDS TERMINATED BY ‘,’ ENCLOSED BY ‘””;

SET foreign_key_checks = 1;

ALEJANDRO ALMAGRO TORREGROSA 21013980F

2. PART 2: NoSQL databases

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

use EmpDB

db.createCollection(“employee”)

var p1 = {_id:3, full_name:”Alice Smith”, score: 6.5, reports: [{employee(_id): 3, number: ””, incentive: “”},{employee(_id): “”, number: 1, incentive: 150.00}], emp_postion:”Tester”}

var p2 = {_id:9, full_name:”Grace Red”, score: 4.0 , reports: [{employee(_id): 9, number: ””, incentive: “”},{employee(_id): “”, number: 1, incentive: “”},{employee(_id): “”, number: 2, incentive: 100.00}],emp_postion:””}

db.employee.insertMany([p1, p2])

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.

db.employee.find({“full_name”, “reports”: 1,”emp_postion”: “Tester”, “score”: {&gt:6.0 } }).sort({“score”: 1, “full_name: 1})