

UD4 – Examen (Modelo B)

LEE LAS INSTRUCCIONES ATENTAMENTE. DEBERÁS ENTREGAR ESTE MISMO DOCUMENTO EN FORMATO PDF (CUALQUIER OTRO FORMATO DISTINTO DE PDF NO SERÁ CORREGIDO). SE ESPERA QUE PUEDAS RESOLVER ESTAS ACTIVIDADES EMPLEANDO EXCLUSIVAMENTE TUS CONOCIMIENTOS Y LOS MATERIALES QUE HAYAS DESCARGADO PREVIAMENTE. EN CASO DE CUALQUIER EVIDENCIA DE COPIA, EL EXAMEN SE EVALUARÁ COMO SUSPENSO (0).

PARA ESTE EXAMEN, REUTILIZAREMOS EL SCRIPT QUE USAMOS PARA LA ACTIVIDAD 3.5 QUE PUEDES ENCONTRAR EN MOODLE. POR LO TANTO, NO OLVIDES CORREGIR EL ERROR (SUGERENCIA: ORDEN INCORRECTO PARA LOS INSERTS).

Pregunta 1. Para empezar, vamos a crear algunos usuarios y unas vistas (3 puntos):

1. Crea las siguientes cuentas de usuario. (0,5 puntos)
 - a. Usuario: alan | Contraseña: turing
 - b. Usuario: steve | Contraseña: wozniak

```
CREATE USER 'alan' IDENTIFIED BY 'turing';
CREATE USER 'steve' IDENTIFIED BY 'wozniak';
```

✓ 1 10:37:30 CREATE USER 'steve' IDENTIFIED BY 'wozniak'

✓ 2 10:37:39 CREATE USER 'alan' IDENTIFIED BY 'turing'

alan	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
steve	%

2. Crea las siguientes vistas. (2 puntos)
 - a. Crea una vista llamada pobreza que muestra el nombre, apellidos y salario del empleado mejor pagado en ese momento.

```
CREATE OR REPLACE VIEW POBREZA AS
SELECT E.FIRST_NAME, E.LAST_NAME, S.SALARY
FROM EMPLOYEES E
JOIN SALARIES S ON E.EMP_NO = S.EMP_NO
WHERE E.TO_DATE="9999-01-01" AND S.TO_DATE="9999-01-01"
ORDER BY S.SALARY DESC LIMIT 1;
```

```
select * from POBREZA;
```

	FIRST_NAME	LAST_NAME	SALARY
▶	Charlene	Brattka	113229

- b. Crea una vista llamada RRHH que muestre el nombre y apellido de todos los empleados que actualmente trabajen en dicho departamento.

```
CREATE OR REPLACE VIEW RRHH AS
SELECT E.FIRST_NAME, E.LAST_NAME
FROM EMPLOYEES E
JOIN DEPT_EMP DE ON E.EMP_NO = DE.EMP_NO
WHERE DE.DEPT_NO="d003" AND DE.TO_DATE="9999-01-01";
```

```
select * from RRHH;
```

	FIRST_NAME	LAST_NAME
►	Kyoichi	Maliniak
	Eberhardt	Terkki
	Adamantios	Portugali
	Alejandro	Brender
	Mayumi	Schueller
	Hisao	Lipner
	Mona	Azuma
	Premal	Baek
	Somnath	Foote
	Hironobu	Haraldson

3. Otorga a los usuarios los siguientes privilegios. (0,5 puntos)

- a. alan – todos los privilegios para ambas vistas.

```
GRANT SELECT ON EMPLOYEES.POBREZA TO ALAN;
GRANT SELECT ON EMPLOYEES.RRHH TO ALAN;
```

- b. steve – sólo se le permite consultar sobre la tabla de empleados, sin embargo, debe poder modificar la columna last_name.

```
GRANT SELECT ON EMPLOYEES.EMPLOYEES TO STEVE;
GRANT UPDATE ON EMPLOYEES.EMPLOYEES.LAST_NAME TO STEVE;
```

No olvides mostrar tu código y capturas de pantalla de sus resultados, incluido el código necesario para lograr la funcionalidad.

Pregunta 2. Sigamos. A continuación, encontrarás 5 enunciados de scripts diferentes para implementar. **Elige sólo 3 de ellos** e implementa esos 3. (4,5 puntos)

1. **Enchufados.** Crea una función que busque el primer salario de un empleado dado su emp_no y retorne un mensaje de error si dicha cuantía es inferior a \$100.000. En caso contrario, debe devolver el primer salario del empleado. (1,5 puntos)
2. **Reducción de jornada.** Crea un procedimiento que baje el salario un 40% a un trabajador dado su emp_no. El procedimiento debe asegurarse de cerrar correctamente el registro del salario anterior y abrir un nuevo registro para el nuevo salario. (1,5 puntos)

```
DELIMITER $$
CREATE PROCEDURE reducciondejornada(IN p_emp_no int, IN p_salario decimal(8,2))
BEGIN
    DECLARE BAJAR_SALARIO decimal(8,2);
    SELECT S.SALARY INTO BAJAR_SALARIO FROM SALARIES S
    JOIN EMPLOYEES E ON E.EMP_NO=S.EMP_NO
    WHERE E.EMP_NO=P_EMP_NO;

    SET P_SALARIO = SALARY / 1.4;
END $$

CALL reducciondejornada(10001, 800.2);
```

✓ 29 11:10:07 CREATE PROCEDURE reducciondejornada(IN p_emp_no int, IN p_salar...

3. **La extra.** Crea un trigger que evite que un empleado tenga dos salarios superpuestos en salarios. Si un nuevo salario se solapa en fechas con otro salario existente para el mismo empleado, muestra un mensaje de error. (1,5 puntos)

```
DELIMITER $$
CREATE TRIGGER LA_EXTRA
BEFORE INSERT ON SALARIES
FOR EACH ROW
BEGIN
    UPDATE SALARIES SET TO_DATE=CURDATE() WHERE EMP_NO=NEW.EMP_NO;
END$$
```

4. **Migajas.** Crea un trigger que evite que un empleado pueda recibir un nuevo salario que sea un 5% más alto que su último salario, mostrando un mensaje de error en tal caso. (1,5 puntos)

5. **Las mujeres facturan.** Escribe un procedimiento almacenado que devuelva una lista de las empleadas (mujeres) que actualmente cobran más una determinada cuantía, la cual se pasa como parámetro. (1,5 puntos)

```
DELIMITER $$  
CREATE PROCEDURE LASMUJERESFACTURAN(IN salario decimal(8,2))  
BEGIN  
DECLARE SALARIO DECIMAL(8,2);  
SELECT * INTO SALARIO FROM SALARIES S JOIN EMPLOYEES E ON E.EMP_NO=S.EMP_NO  
WHERE E.GENDER="M" AND S.TO_DATE="9999-01-01" AND S.SALARY >= SALARIO;  
END $$
```

```
CALL LASMUJERESFACTURAN(60470.2);
```

```
✓ 22 11:02:04 CREATE PROCEDURE LASMUJERESFACTURAN(IN salario decimal(8,...
```

No olvides mostrar tu código y capturas de pantalla de sus resultados, incluido el código necesario para mostrar la funcionalidad.

Pregunta 3. Nuestro superintendente quiere que cada vez que un empleado reciba un nuevo título, se registren el emp_no, la fecha del nombramiento, el nuevo título y, especialmente, el emp_no del manager del departamento en una tabla llamada "titulines". Considera las siguientes preguntas. (2,5 puntos)

1. Para lograr la funcionalidad requerida, ¿qué herramienta sería la más apropiada en este caso? ¿Trigger, función o procedimiento? Justifica tu respuesta. (1 punto)

Se podría usar un procedimiento ya que se quiere a partir de un dato (parámetro de entrada, que invalida directamente al trigger), insertar otro dato en otra tabla.

2. Desarrolla la funcionalidad solicitada y demuestra su funcionamiento. (1,5 puntos)

```
CREATE TABLE TITULINES(  
  EMP_NO int NOT NULL,  
  EMP_NO_MANAGER INT ,  
  FROM_DATE DATE,  
  TITLE VARCHAR(50)  
);  
  
DELIMITER $$  
CREATE PROCEDURE reducciondejornada(IN p_title VARCHAR(50))  
BEGIN  
  SELECT E.EMP_NO, DM.EMP_NO FROM EMPLOYEES E JOIN DEPT_MANAGER ON E.EMP_NO=DM.EMP_NO;  
  INSERT INTO TITULINES VALUES(EMP_NO, EMP_NO_MANAGER, P_TITLE, CURDATE());  
END $$
```

NO OLVIDES AÑADIR CAPTURAS DE PANTALLA DE TU CÓDIGO, ASÍ COMO DE TODOS TUS RESULTADOS