

TAREA OPCIONAL BASE DE DATOS



1. Insertar una nueva fila en una tabla especificando algunos campos (no todos) de dicha tabla.
`INSERT INTO empleado (nif, nombre, apellido1, id_departamento) VALUES ('12345678A', 'Luis', 'Martínez', 1);`
2. Insertar una nueva fila en una tabla que incluya todos los campos de dicha tabla, pero en orden diferente.
`INSERT INTO empleado (apellido1, apellido2, nombre, id_departamento, nif) VALUES ('García', 'Hernández', 'Ana', 3, '98765432B');`
3. Insertar una nueva fila en una tabla que incluya todos los campos de dicha tabla, en el mismo orden.
`INSERT INTO empleado (nif, nombre, apellido1, apellido2, id_departamento) VALUES ('11223344C', 'Carlos', 'Pérez', 'López', 4);`
4. Inserta una nueva fila en una tabla, dejando que, al menos un campo, tome el valor por defecto y, al menos, otro campo tome el valor NULL (sin especificarlo en la sentencia).
`INSERT INTO empleado (nif, nombre, apellido1) VALUES ('55667788D', 'Elena', 'Morales');`
5. Inserta en una tabla el resultado de ejecutar una consulta sobre otra tabla.
`INSERT INTO departamento (nombre, presupuesto, gastos) SELECT 'Nuevo Departamento', presupuesto, gastos FROM departamento WHERE id = 1;`
6. Insertar tres filas en una tabla que incluyan todos los campos de dicha tabla. `INSERT INTO empleado (nif, nombre, apellido1, apellido2, id_departamento) VALUES ('66554433E', 'Fernando', 'Ramos', 'Gómez', 1), ('77889900F', 'Isabel', 'Vega', 'Fernández', 2), ('88990011G', 'Rafael', 'González', 'Martínez', 3);`
7. Crear una tabla vacía copia de otra existente. Insertar un conjunto de filas (no toda la tabla) de la tabla base original a la nueva tabla.
`CREATE TABLE empleado_nuevo LIKE empleado; INSERT INTO empleado_nuevo (nif, nombre, apellido1, apellido2, id_departamento) SELECT nif, nombre, apellido1, apellido2, id_departamento FROM empleado WHERE id_departamento = 2;`
8. Crear una VISTA asociada a una SELECT multitabla.
`CREATE VIEW empleados_departamentos AS SELECT e.nombre AS empleado_nombre, e.apellido1, e.apellido2, d.nombre AS departamento_nombre FROM empleado e JOIN departamento d ON e.id_departamento = d.id;`
9. Modificar el valor de un campo asignando un literal para una única fila de una tabla.
`UPDATE empleado SET apellido1 = 'López' WHERE nif = '32481596F';`
10. Modificar el valor de un campo para un grupo de filas de una tabla asignado un nuevo valor en relación al valor actual de dicho campo.
`UPDATE departamento SET gastos = gastos * 1.1 WHERE presupuesto < 150000;`
11. Modifica el valor de un campo de una tabla utilizando una subconsulta.
`UPDATE empleado SET id_departamento = (SELECT id FROM departamento WHERE nombre = 'Sistemas') WHERE nif = '38382980M';`
12. Eliminar una fila de una tabla (que no esté referenciada).
`DELETE FROM departamento WHERE id = 6;`
13. Eliminar un grupo de filas de una tabla (que no estén referenciadas).
`DELETE FROM departamento WHERE presupuesto = 0;`
14. Elimina filas de una tabla mediante una subconsulta.
`DELETE FROM empleado WHERE id_departamento IN (SELECT id FROM departamento WHERE presupuesto < 100000);`
15. Eliminar el MAX/MIN de una tabla, asociando una subconsulta SELECT sobre la misma tabla, para que genere el error de no poder eliminarlo. Crear una tabla auxiliar de copia de dicha tabla y ejecutar la eliminación usando dicha tabla copia. Por último eliminar la tabla auxiliar de copia.
`DELETE FROM empleado WHERE nif = (SELECT nif FROM empleado ORDER BY nif DESC LIMIT 1); CREATE TABLE empleado_aux AS SELECT * FROM empleado; DELETE FROM empleado_aux WHERE nif = (SELECT nif FROM empleado_aux ORDER BY nif DESC LIMIT 1); DROP TABLE empleado_aux;`

16. Eliminar una/s fila/s de una tabla que se encuentren referenciadas por otra/s tabla/s sin que tengan la definición de ON DELETE CASCADE por lo que se produce un error que impide la eliminación. Modificar la CONSTRAINT de FOREIGN KEY tal y como se ha hecho en clase para que se produzca la eliminación de dicha fila dejando como NULL los registros de otras tablas que lo referencian (no eliminar estas filas).

```
ALTER TABLE empleado DROP FOREIGN KEY empleado_ibfk_1;  
ALTER TABLE empleado ADD CONSTRAINT empleado_ibfk_1 FOREIGN KEY  
(id_departamento) REFERENCES departamento(id) ON DELETE SET NULL; DELETE FROM  
departamento WHERE id = 5;
```

17. Modificar la clave principal de una/s fila/s de una tabla que se encuentre referenciada por otra/s tabla/s sin que tengan la definición de ON UPDATE CASCADE por lo que se produce un error que impide la actualización. Modificar la CONSTRAINT de FOREIGN KEY tal y como se ha hecho en clase para que se puede llevar a cabo la modificación en CASCADA de la clave principal y de todos los registros que la referencian.

```
ALTER TABLE empleado DROP FOREIGN KEY empleado_ibfk_1;  
ALTER TABLE empleado ADD CONSTRAINT empleado_ibfk_1 FOREIGN KEY  
(id_departamento)  
REFERENCES departamento(id) ON UPDATE CASCADE; UPDATE departamento SET id = 8  
WHERE id = 1;
```

18. Insertar dos filas en una tabla, en dos sentencias diferentes dentro de una transacción. Acabar la transacción retrocediendo los cambios.

```
START TRANSACTION;  
INSERT INTO empleado (nif, nombre, apellido1, apellido2, id_departamento)  
VALUES ('99887766H', 'Teresa', 'García', 'Hernández', 3); INSERT INTO empleado (nif, nombre,  
apellido1, apellido2, id_departamento) VALUES ('11221122I', 'Roberto', 'Martínez', 'López', 4);  
ROLLBACK;
```

19. Repetir el ejercicio anterior, confirmando los cambios. START TRANSACTION; INSERT INTO empleado (nif, nombre, apellido1, apellido2, id_departamento) VALUES ('99887766H', 'Teresa', 'García', 'Hernández', 3);

```
INSERT INTO empleado (nif, nombre, apellido1, apellido2, id_departamento) VALUES  
('11221122I', 'Roberto', 'Martínez', 'López', 4); COMMIT;
```