

# Modificación de tablas

---

Agregar una columna a una tabla

## NOTA

No hay forma de insertar una nueva columna en una posición concreta de la tabla. La columna, que se añade mediante la cláusula ADD, se inserta siempre al final de la tabla.

**USE** sample

```
ALTER TABLE employee  
ADD telephone_no CHAR(12) NULL
```

---

Eliminar columna de una tabla

```
ALTER TABLE employee  
DROP COLUMN telephone_no
```

---

Modificación de las propiedades de una columna

```
ALTER TABLE department  
ALTER COLUMN ubication CHAR(25) NOT NULL;
```

---

Cambiar nombre de una columna

```
sp_RENAME 'employee.emp.fname', 'employee_name' , 'COLUMN'
```

Nombre antiguo

Nombre nuevo

```
SELECT *  
FROM employee
```

---

Cambiar nombre de una tabla

**sp\_RENAME** 'employee', 'empleados'

Nombre antiguo

Nombre nuevo

```
SELECT *  
FROM empleados
```

---

Adición o Eliminación de restricciones de integridad

```
CREATE TABLE sales  
(order_no INTEGER NOT NULL,  
order_date DATE NOT NULL,  
ship_date DATE NOT NULL);
```

#### Agregar la restricción CHECK

```
ALTER TABLE sales  
ADD CONSTRAINT order_check CHECK(order_date <= ship_date)
```

#### Agregar la restricción PRIMARY KEY

```
ALTER TABLE sales  
ADD CONSTRAINT primaryk_sales PRIMARY KEY (order_no)
```

#### Agregar la restricción PRIMARY KEY [SQL SERVER y POSTGRESQL]

```
ALTER TABLE sales  
ADD PRIMARY KEY (order_no)
```

#### Eliminar la restricción CHECK

```
ALTER TABLE sales  
DROP CONSTRAINT order_check
```

#### Agregar la restricción NOT NULL

```
ALTER TABLE department  
ALTER COLUMN ubication CHAR(30) NOT NULL;
```

#### Eliminar la restricción NOT NULL

```
ALTER TABLE department  
ALTER COLUMN ubication CHAR(30) NULL;
```

Eliminación de objetos de la base de datos

### Creación de base de datos

```
CREATE DATABASE prueba1  
CREATE DATABASE prueba2
```

```
USE prueba1;
```

### Creación de tablas en la base de datos

```
CREATE TABLE tabla1 (  
columna1 VARCHAR(10) NOT NULL );
```

```
CREATE TABLE tabla2 (  
columna1 VARCHAR(10) NOT NULL);
```

### Eliminación de tablas de la base de datos

```
DROP TABLE tabla1, tabla2
```

**Debemos estar en otra base de datos para borrar las bases de datos de la sentencia.**

```
USE sample
```

### Eliminación de bases de datos

```
DROP DATABASE prueba1, prueba2
```

**Además de DATABASE y TABLE, los objetos de la sentencia DROP pueden ser, entre otros, los siguientes:**

- DROP PROCEDURE
- DROP INDEX
- DROP VIEW
- DROP SCHEMA
- DROP TRIGGER

Insertar filas en una tabla

### Primera forma

```
USE sample
INSERT INTO employee VALUES (25348, 'Matthew', 'Smith','d3');
INSERT INTO employee VALUES (10102, 'Ann', 'Jones','d3');
INSERT INTO employee VALUES (18316, 'John', 'Barrimore', 'd1');
INSERT INTO employee VALUES (29346, 'James', 'James', 'd2');
INSERT INTO employee VALUES (9031, 'Elsa', 'Bertoni', 'd2');
INSERT INTO employee VALUES (2581, 'Elke', 'Hansel', 'd2');
INSERT INTO employee VALUES (28559, 'Sybill', 'Moser', 'd1');
```

### Segunda forma

```
USE sample;
INSERT INTO department VALUES ('d4', 'Human Resources', 'Chicago'),
                                ('d5', 'Distribution', 'New Orleans'),
                                ('d6', 'Sales', 'Chicago');
```

La inserción de valores en algunas (pero no todas) de las columnas de una tabla suele requerir la especificación explícita de las columnas correspondientes. Las columnas omitidas deben ser anulables o tener un valor DEFAULT.

El orden de los nombres de las columnas en la cláusula VALUE de la sentencia INSERT puede ser diferente del orden original de esas columnas, que se determina en la sentencia CREATE TABLE.

---

UPDATE: Actualizar filas

### NOTA

Una sentencia UPDATE sólo puede modificar los datos de una única tabla

**Establezca que el job del empleado número 18316, que trabaja en el proyecto p2, sea "Manager"**

```
USE sample;
UPDATE works_on
  SET job = 'Manager'
  WHERE emp_no = 18316
  AND project_no = 'p2';
```

Modificar los presupuestos de todos los proyectos para que se representen en libras esterlinas. El tipo de cambio actual es de 0,51 libras por 1 dólar.

```
USE sample
UPDATE project
  SET budget = budget*0.51
```

Debido a su enfermedad, establezca todos los jobs de todos los proyectos de la Sra. Jones como NULL.

```
USE sample
UPDATE works_on
  SET job = NULL
  WHERE emp_no =
    (SELECT emp_no FROM employee WHERE emp_lname = 'Jones')
```

Igual al anterior

```
USE sample;
UPDATE works_on
  SET job = NULL
  FROM works_on, employee
  WHERE emp_lname = 'Jones'
  AND works_on.emp_no = employee.emp_no;
```

El presupuesto de cada proyecto debe incrementarse en un porcentaje (20, 10 o 5) dependiendo de su cantidad de dinero anterior. Los proyectos con un presupuesto más bajo se incrementarán en los porcentajes más altos.

```
USE sample;
UPDATE project
  SET budget =
    CASE
      WHEN budget > 0 and budget < 100000 THEN budget*1.2
      WHEN budget >= 100000 and budget < 200000 THEN budget*1.1
      ELSE budget*1.05
    END
```

Establecer nombre = fideos y categoria = abarrotes siempre y cuando el nombre contenga "pan" o la categoria sea golosinas.

```
UPDATE Producto
  SET nombre = 'fideos', categoria = 'abarrotes'
  WHERE nombre LIKE '%pan%' OR categoria = 'golosinas'
```

DELETE: Eliminar filas

Se borrarán todas las filas que cumplan la condición de la cláusula WHERE.

### Eliminar todos los managers de la tabla works\_on

```
USE sample;  
DELETE FROM works_on  
WHERE job = 'Manager'
```

### La Sra. Moser está de baja. Eliminar todas las filas de la base de datos que la conciernen

```
USE sample  
DELETE FROM works_on  
WHERE emp_no IN  
(SELECT emp_no  
FROM employee  
WHERE emp_lname = 'Moser')
```

```
DELETE FROM employee  
WHERE emp_lname = 'Moser'
```

### Igual al anterior

```
USE sample  
DELETE works_on  
FROM works_on, employee  
WHERE works_on.emp_no = employee.emp_no  
AND emp_lname = 'Moser'
```

```
DELETE FROM employee  
WHERE emp_lname = 'Moser'
```

### Si se omite la cláusula WHERE, se borrarán todas las filas de una tabla.

```
USE sample;  
DELETE FROM works_on;
```

## TRUNCATE: Eliminar TODAS las filas de una tabla

Nos permite eliminar por completo los registros de una tabla. Los elimina por completo y resetea los valores que la tabla pueda tener campos identity, es decir, si yo tuviera registros agregados y hago un TRUNCATE, cuando siga agregando registros lo que voy a obtener es el primer campo identity con el valor inicial, si fuese 1 va a estar en 1, y si fuese el que fuese va a arrancar de ese valor

TRUNCATE es similar a DELETE en cuanto a eliminar todas las filas de una tabla, sin embargo, DELETE no reinicia el identity y TRUNCATE si lo hace.