

## A2. Sentencias Insert, Delete y Update.

### Índice.

1.	Sentencia Insert.....	2
1.1.	Agregar registros individuales sin especificar columnas.....	5
1.2.	Agregar registros individuales especificando columnas.....	6
1.3.	Agregar varios registros sin especificar columnas.....	7
1.4.	Agregar varios registros específico de columnas.....	8
1.5.	Agregar registros con el comando SET.....	9
1.6.	Agregar registros a través de una selección.....	10
2.	Sentencia Delete.....	11
2.1.	Borrar todos los registros.....	14
2.2.	Borrar registros según un filtro.....	15
2.3.	Borrar registros según su ordenación.....	16
2.4.	Borrar un número específico de registros.....	17
2.5.	Borrar registros de varias tablas.....	18
3.	Sentencia Update.....	19
3.1.	Actualizar todos los registros.....	21
3.2.	Actualizar registros según un filtro.....	22
3.3.	Actualizar registros según su ordenación.....	23
3.4.	Actualizar un número específico de registros.....	24
3.5.	Actualizar registros de varias tablas.....	25

## A2. Sentencias Insert, Delete y Update.

### 1. Sentencia Insert.

---

La sentencia INSERT añade filas en una tabla de la base de datos.

```
INSERT [INTO] nombre_tabla [ (nombre_columna, ...) ]  
VALUES ( {lista_valores | DEFAULT}, ... )
```



## A2. Sentencias Insert, Delete y Update.

### 1. Sentencia Insert.

La sentencia INSERT en MySQL tiene el siguiente formato:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      [(col_name [, col_name] ...)]
      { {VALUES | VALUE} (value_list) [, (value_list)] ...
        |
        VALUES row_constructor_list
      }
```

value:  
    {expr | DEFAULT}

value\_list:  
    value [, value] ...

row\_constructor\_list:  
    ROW(value\_list)[, ROW(value\_list)][, ...]

assignment:  
    col\_name = [row\_alias.]value

assignment\_list:  
    assignment [, assignment] ...

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      [AS row_alias[(col_alias [, col_alias] ...)]]
      SET assignment_list
      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      [(col_name [, col_name] ...)]
      [AS row_alias[(col_alias [, col_alias] ...)]]
      {SELECT ... | TABLE table_name}
      [ON DUPLICATE KEY UPDATE assignment_list]
```

## A2. Sentencias Insert, Delete y Update.

### 1. Sentencia Insert.

---

El significado de algunos de los elementos anteriores es:

LOW_PRIORITY	retrasa la inserción hasta que ningún Cliente esté leyendo de la tabla
DELAYED	retrasa la inserción hasta que la tabla no esté en uso. NO EN USO
HIGH_PRIORITY	anula el efecto de baja prioridad y NO emplea inserciones concurrentes
IGNORE	ignora los errores que se puedan obviar, generando advertencias
INTO	especifica la tabla en la que se insertarán los datos
PARTITION	toma una lista de nombres separados por comas de una o más particiones de la tabla
col_name	nombre de la columna en la que se va a insertar un dato
VALUES VALUE	valores que se van a agregar
SET	asigna valores a campos (nombre = 'Sara')
SELECT	valores que se agregan procedentes de otra tabla
ON DUPLICATE KEY UPDATE	se modifica un valor si se duplica un valor ÚNICO o CLAVE

## A2. Sentencias Insert, Delete y Update.

### 1.1. Agregar registros individuales sin especificar columnas.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name)]
      { {VALUES | VALUE} (value_list)
        |
        VALUES row_constructor_list
      }

      [AS row_alias[(col_alias)]]

      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
insert into Provincias VALUE ( 'A Coruña', 7950 );
insert into Provincias VALUE ( 'Lugo', 9856 );
insert into Provincias VALUE ( 'Ourense', 7273 );
insert into Provincias VALUE ( 'Pontevedra', 4495 );
```

## A2. Sentencias Insert, Delete y Update.

### 1.2. Agregar registros individuales especificando columnas.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name)]
      [(col_name)]

      { {VALUES | VALUE} (value_list)
        |
        VALUES row_constructor_list
      }

      [AS row_alias[(col_alias)]]
      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
insert into Provincias ( nombre, superficie )  VALUE ( 'A Coruña', 7950 );
insert into Provincias ( nombre, superficie )  VALUE ( 'Lugo', 9856 );
insert into Provincias ( nombre, superficie )  VALUE ( 'Ourense', 7273 );
insert into Provincias ( nombre, superficie )  VALUE ( 'Pontevedra', 4495 );
```

## A2. Sentencias Insert, Delete y Update.

### 1.3. Agregar varios registros sin especificar columnas.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      { {VALUES | VALUE} (value_list) [, (value_list)] ...
        |
        VALUES row_constructor_list
      }

      [AS row_alias[(col_alias [, col_alias] ...)]]
      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
insert into Provincias
      VALUE ( 'A Coruña', 7950 ), ( 'Lugo', 9856 ), ( 'Ourense', 7273 ), ( 'Pontevedra', 4495 );
```



## A2. Sentencias Insert, Delete y Update.

### 1.4. Agregar varios registros especificando columnas.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      [(col_name [, col_name] ...)]

      { {VALUES | VALUE} (value_list) [, (value_list)] ...
        |
        VALUES row_constructor_list
      }

      [AS row_alias[(col_alias [, col_alias] ...)]]
      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
insert into Provincias ( nombre, superficie )
      VALUE ( 'A Coruña', 7950 ), ( 'Lugo', 9856 ), ( 'Ourense', 7273 ), ( 'Pontevedra', 4495 );
```



## A2. Sentencias Insert, Delete y Update.

### 1.5. Agregar registros con el comando SET.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      [AS row_alias[(col_alias [, col_alias] ...)]]
      SET assignment_list
      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
insert into Provincias set nombre = 'A Coruña',    superficie = 7950;
insert into Provincias set nombre = 'Lugo',        superficie = 9856;
insert into Provincias set nombre = 'Ourense',     superficie = 7273;
insert into Provincias set nombre = 'Pontevedra',  superficie = 4495;
```

## A2. Sentencias Insert, Delete y Update.

### 1.6. Agregar registros a través de una selección.

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      [PARTITION (partition_name [, partition_name] ...)]
      [(col_name [, col_name] ...)]
      [AS row_alias[(col_alias [, col_alias] ...)]]
      SELECT ... | TABLE table_name
      [ON DUPLICATE KEY UPDATE assignment_list]
```

```
insert into Provincias_Copia SELECT * FROM Provincias;
```

## A2. Sentencias Insert, Delete y Update.

### 2. Sentencia Delete.

---

La sentencia DELETE borra registros de una tabla.

Si se utiliza el filtro WHERE, su codificación es idéntica a la sentencia SELECT.

```
DELETE FROM nombre_tabla  
[WHERE filtro]
```



## A2. Sentencias Insert, Delete y Update.

### 2. Sentencia Delete.

La sentencia DELETE en MySQL tiene el siguiente formato:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name [[AS] tbl_alias]  
  [PARTITION (partition_name [, partition_name] ...)]  
  [WHERE where_condition]  
  [ORDER BY ...]  
  [LIMIT row_count]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]  
  tbl_name [...] [, tbl_name [...]] ...  
FROM table_references  
[WHERE where_condition]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]  
FROM tbl_name [...] [, tbl_name [...]] ...  
USING table_references  
[WHERE where_condition]
```

## A2. Sentencias Insert, Delete y Update.

### 2. Sentencia Delete.

---

El significado de algunos de los elementos anteriores es:

LOW_PRIORITY	retrasa el borrado de los elementos has que ningún Cliente esté accediendo
QUICK	da máxima prioridad al borrado
IGNORE	ignora los errores que se puedan obviar, generando advertencias
FROM	tabla en la que se borrarán las tuplas indicadas
AS	pseudónimo de la tabla
PARTITION	toma una lista de nombres separados por comas de una o más particiones de la tabla
WHERE	identifica qué filas se borrarán
ORDER BY	ordena las filas que se borrarán
LIMIT	número máximo de filas que se borran

## A2. Sentencias Insert, Delete y Update.

### 2.1. Borrar todos los registros.

---

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      FROM tbl_name [[AS] tbl_alias]
      [PARTITION (partition_name [, partition_name] ...)]
```

```
DELETE FROM Provincias;
```

## A2. Sentencias Insert, Delete y Update.

### 2.2. Borrar registros según un filtro.

---

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      FROM tbl_name [[AS] tbl_alias]
      [PARTITION (partition_name [, partition_name] ...)]
      [WHERE where_condition]
```

```
DELETE FROM Provincias      WHERE superficie > 7500;
```



## A2. Sentencias Insert, Delete y Update.

### 2.3. Borrar registros según su ordenación.

---

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      FROM tbl_name [[AS] tbl_alias]
      [PARTITION (partition_name [, partition_name] ...)]
      [ORDER BY ...]
```

```
DELETE FROM Provincias      ORDER BY superficie;
```

## A2. Sentencias Insert, Delete y Update.

### 2.4. Borrar un número específico de registros .

---

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      FROM tbl_name [[AS] tbl_alias]
      [PARTITION (partition_name [, partition_name] ...)]
      [LIMIT row_count]
```

```
DELETE FROM Provincias    LIMIT 2;
```

## A2. Sentencias Insert, Delete y Update.

### 2.5. Borrar registros de varias tablas.

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      tbl_name[.*] [, tbl_name[.*]] ...
FROM table_references
[WHERE where_condition]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      FROM tbl_name[.*] [, tbl_name[.*]] ...
      USING table_references
      [WHERE where_condition]
```

```
DELETE Provincias, Provincias_Copia FROM Provincias JOIN Provincias_Copia;
```

## A2. Sentencias Insert, Delete y Update.

### 3. Sentencia Update.

La sentencia UPDATE modifica el contenido de cualquier columna y fila de una tabla.

Si se utiliza el filtro WHERE, su codificación es idéntica a la de la sentencia SELECT.

```
UPDATE nombre_tabla  
    SET nombre_columna1 = valor1 [, nombre_columna2 = valor2 ] ...  
    [WHERE filtro]
```



## A2. Sentencias Insert, Delete y Update.

### 3. Sentencia Update.

---

La sentencia UPDATE en MySQL tiene el siguiente formato:

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
  SET assignment_list
  [WHERE where_condition]
  [ORDER BY ...]
  [LIMIT row_count]

value:
    {expr | DEFAULT}

assignment:
    col_name = value

assignment_list:
    assignment [, assignment] ...

UPDATE [LOW_PRIORITY] [IGNORE] table_references
  SET assignment_list
  [WHERE where_condition]
```

## A2. Sentencias Insert, Delete y Update.

### 3.1. Actualizar todos los registros.

---

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference  
    SET assignment_list
```

```
UPDATE provincias_galicia  
    SET nombre = 'A Coruña';
```

```
UPDATE provincias_galicia  
    SET nombre = 'A Coruña', superficie = 4000;
```

## A2. Sentencias Insert, Delete y Update.

### 3.2. Actualizar registros según un filtro.

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference  
    SET assignment_list  
    [WHERE where_condition]
```

```
UPDATE provincias_galicia  
    SET nombre = 'A Coruña'  
    WHERE nombre = 'Lugo';
```

```
UPDATE provincias_galicia  
    SET nombre = 'A Coruña', superficie = 4000  
    WHERE nombre = 'Lugo';
```



## A2. Sentencias Insert, Delete y Update.

### 3.3. Actualizar registros según su ordenación.

---

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference  
  SET assignment_list  
  [ORDER BY ...]
```

```
UPDATE provincias_galicia  
  SET nombre = 'A Coruña'  
  ORDER BY Superficie;
```

## A2. Sentencias Insert, Delete y Update.

### 3.4. Actualizar un número específico de registros.

---

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference  
  SET assignment_list  
  [LIMIT row_count]
```

```
UPDATE provincias_galicia  
  SET nombre = 'A Coruña'  
  LIMIT 2;
```

## A2. Sentencias Insert, Delete y Update.

### 3.5. Actualizar registros de varias tablas.

```
UPDATE [LOW_PRIORITY] [IGNORE] table_references  
  SET assignment_list  
  [WHERE where_condition]
```

```
UPDATE provincias_galicia  
      SET nombre = 'A Coruña', superficie = 4000;
```

```
UPDATE provincias_galicia  
      SET nombre = 'A Coruña', superficie = 4000  
      WHERE nombre = 'Lugo';
```