NEW, OLD en Disparadores

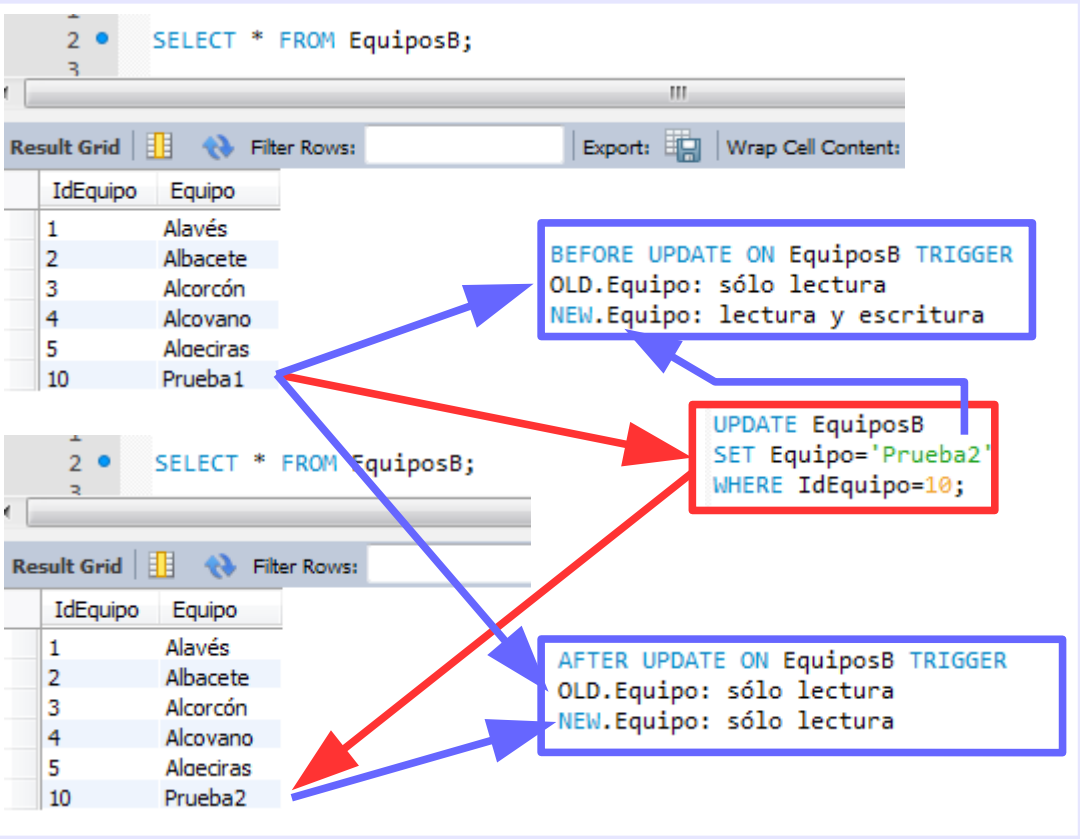
Índice de contenido

Introducción

```
-- Codigo ejecutado para hacer las capturas de imagen:

DROP TABLE IF EXISTS EquiposB;
CREATE TABLE EquiposB (
  IdEquipo INT,
  Equipo   CHAR(30));
INSERT INTO EquiposB VALUES (1,'Alavés');
INSERT INTO EquiposB VALUES (2,'Albacete');
INSERT INTO EquiposB VALUES (3,'Alcorcón');
INSERT INTO EquiposB VALUES (4,'Alcoyano');
INSERT INTO EquiposB VALUES (5,'Algeciras');

SELECT * FROM EquiposB;

INSERT INTO EquiposB VALUES (10,'Prueba1');
UPDATE EquiposB SET Equipo='Prueba2' WHERE IdEquipo=10;
DELETE FROM EquiposB WHERE IdEquipo=10;
```

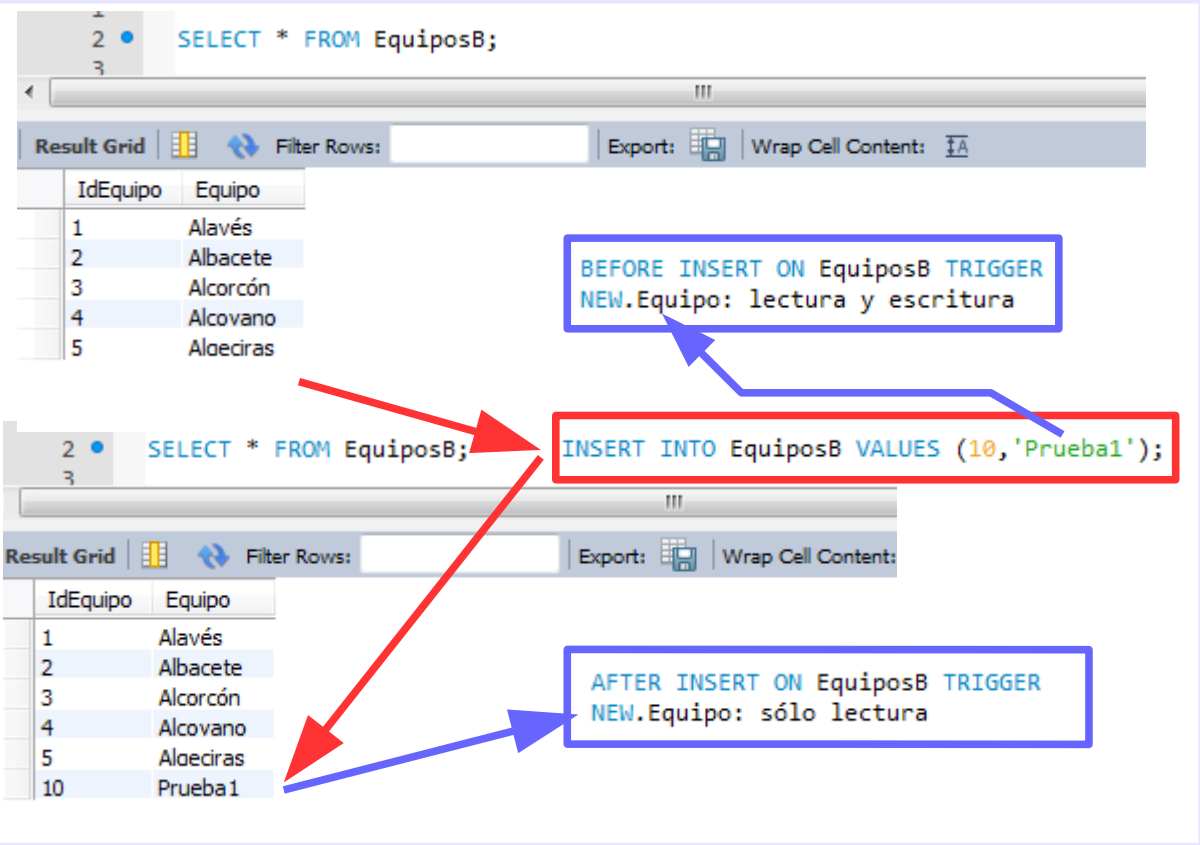
Dentro de un disparador podemos usar la palabras clave OLD y NEW.

En disparadores UPDATE



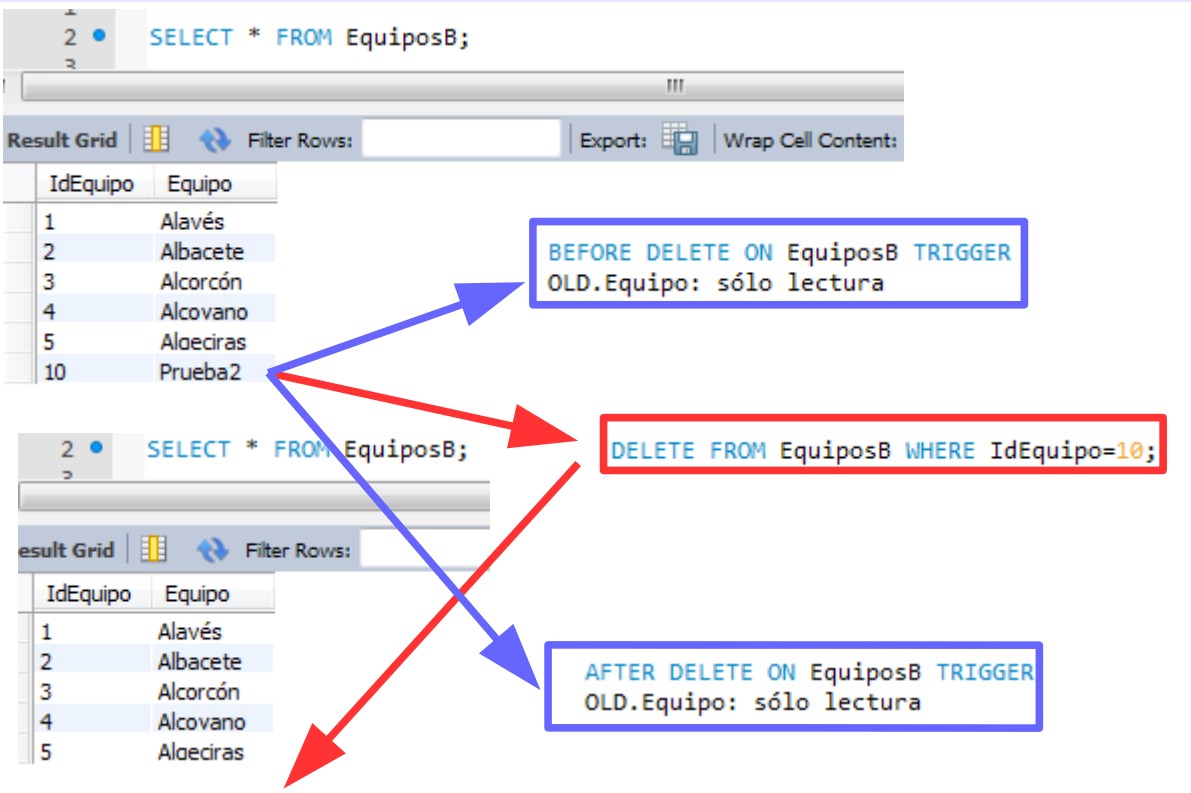
En los disparadores UPDATE podemos hacer referencia tanto a OLD.Campo como a NEW.Campo.
OLD.Campo hace referencia al valor que tenía el campo antes de ser actualizado y es de sólo lectura porque no tiene sentido modificarlo ya que va a ser cambiado en breve.
NEW.Campo hace referencia al nuevo valor que se va a introducir y tiene sentido cambiarlo sólo antes de hacer el UPDATE. Por lo tanto NEW.Campo es de lectura y escritura en un disparador BEFORE UPDATE y sólo de lectura en uno AFTER UPDATE.

En disparadores INSERT



En los disparadores INSERT sólo podemos hacer referencia a NEW.Campo. No tiene sentido acceder a OLD.Campo porque el registro no existe.
NEW.Campo hace referencia al nuevo valor que se va a introducir y tiene sentido cambiarlo sólo antes de hacer el INSERT. Por lo tanto NEW.Campo es de lectura y escritura en un disparador BEFORE INSERT y de sólo lectura en uno AFTER INSERT.

En disparadores DELETE



En los disparadores DELETE sólo podemos hacer referencia a OLD.Campo. No tiene sentido acceder a NEW.Campo porque el registro no existe.
OLD.Campo hace referencia al valor que tenía el campo antes de ser borrado y es de sólo lectura porque no tiene sentido modificarlo ya que va a ser borrado en breve.

Resumen

- En un disparador INSERT sólo podemos hacer referencia a NEW.Campo
- En un disparador DELETE sólo podemos hacer referencia a OLD.Campo
- En un disparador UPDATE podemos hacer referencia tanto a NEW.Campo como a OLD.Campo
- En disparadores BEFORE INSERT y BEFORE UPDATE, NEW.Campo es de lectura y escritura en el resto son de sólo lectura
- OLD.Campo es siempre de sólo lectura

Código de prueba

```
-- Codigo para probarlo todo:

DROP TABLE IF EXISTS EquiposB;
CREATE TABLE EquiposB (
```

```
  IdEquipo INT,
  Equipo   CHAR(30));
INSERT INTO EquiposB VALUES (1,'Alavés');
INSERT INTO EquiposB VALUES (2,'Albacete');
INSERT INTO EquiposB VALUES (3,'Alcorcón');
INSERT INTO EquiposB VALUES (4,'Alcoyano');
INSERT INTO EquiposB VALUES (5,'Algeciras');
SELECT * FROM EquiposB;

DROP TABLE IF EXISTS NotasDisparadores;
CREATE TABLE NotasDisparadores (
  Disparador   CHAR(30),
  ValorOLD     CHAR(30),
  ValorNEW     CHAR(30));
SELECT * FROM NotasDisparadores;

DROP TRIGGER IF EXISTS BeforeEquiposBInsert;
DELIMITER //
CREATE TRIGGER BeforeEquiposBInsert
BEFORE INSERT ON EquiposB
FOR EACH ROW
BEGIN
INSERT INTO NotasDisparadores VALUES ('BeforeEquiposBInsert','OLD.Equipo da error',NEW.Equipo);
SET NEW.Equipo='Modificado1';
INSERT INTO NotasDisparadores VALUES ('BeforeEquiposBInsert NEW modificado','OLD.Equipo da
error',NEW.Equipo);
END//
DELIMITER ;

DROP TRIGGER IF EXISTS AfterEquiposBInsert;
DELIMITER //
CREATE TRIGGER AfterEquiposBInsert
AFTER INSERT ON EquiposB
FOR EACH ROW
BEGIN
INSERT INTO NotasDisparadores VALUES ('AfterEquiposBInsert','OLD.Equipo da error',NEW.Equipo);
-- SET NEW.Equipo='Modificado1'; Esta orden da error
END//
DELIMITER ;

DROP TRIGGER IF EXISTS BeforeEquiposBUpdate;
DELIMITER //
CREATE TRIGGER BeforeEquiposBUpdate
BEFORE UPDATE ON EquiposB
FOR EACH ROW
BEGIN
INSERT INTO NotasDisparadores VALUES ('BeforeEquiposBUpdate',OLD.Equipo,NEW.Equipo);
SET NEW.Equipo='Modificado2';
INSERT INTO NotasDisparadores VALUES ('BeforeEquiposBUpdate NEW modificado',OLD.Equipo,NEW.Equipo);
END//
DELIMITER ;

DROP TRIGGER IF EXISTS AfterEquiposBUpdate;
DELIMITER //
CREATE TRIGGER AfterEquiposBUpdate
AFTER UPDATE ON EquiposB
FOR EACH ROW
BEGIN
INSERT INTO NotasDisparadores VALUES ('AfterEquiposBUpdate',OLD.Equipo,NEW.Equipo);
-- SET NEW.Equipo='Modificado1'; Esta orden da error
END//
DELIMITER ;

DROP TRIGGER IF EXISTS BeforeEquiposBDelete;
DELIMITER //
CREATE TRIGGER BeforeEquiposBDelete
BEFORE DELETE ON EquiposB
FOR EACH ROW
BEGIN
INSERT INTO NotasDisparadores VALUES ('BeforeEquiposBDelete',OLD.Equipo,'NEW.Equipo da error');
-- SET NEW.Equipo='Modificado1'; Esta orden da error
END//
```

```
DELIMITER ;

DROP TRIGGER IF EXISTS AfterEquiposBDelete;
DELIMITER //
CREATE TRIGGER AfterEquiposBDelete
AFTER DELETE ON EquiposB
FOR EACH ROW
BEGIN
INSERT INTO NotasDisparadores VALUES ('AfterEquiposBDelete',OLD.Equipo,'NEW.Equipo da error');
-- SET NEW.Equipo='Modificado1'; Esta orden da error
END//
DELIMITER ;

TRUNCATE NotasDisparadores;
INSERT INTO EquiposB VALUES (10,'Prueba1');
SELECT * FROM NotasDisparadores;

TRUNCATE NotasDisparadores;
UPDATE EquiposB SET Equipo='Prueba2' WHERE IdEquipo=10;
SELECT * FROM NotasDisparadores;

TRUNCATE NotasDisparadores;
DELETE FROM EquiposB WHERE IdEquipo=10;
SELECT * FROM NotasDisparadores;
```

Licencia