

Overview

- similar to [[Procedures and Functions]] but execute on event
 - on delete/insert/update
- allow complex check constraints and active behaviour
 - replication
 - auto deletion
 - auditing
- very powerful but complex
 - need to be careful
 - lots of implicate dependencies
 - trigger may trigger other triggers
 - * infinite loop

Trigger Template

```
CREATE TRIGGER <triggername>
  BEFORE | AFTER | INSTEAD OF
  INSERT | DELETE | (UPDATE OF <column_list>)
  ON <tablename>
  [REFERENCING <old_new_alias_list>]
  [FOR EACH {ROW | STATEMENT}]
  [WHEN (<search condition>)]
  <SQL procedure statement> |
{ BEGIN ATOMIC
  {<SQL Procedure statement>;}...
END }
```

The diagram illustrates the structure of a **CREATE TRIGGER** statement. It uses blue brackets on the right side to group parts of the statement into three categories: **Event**, **Condition**, and **Action**. The **Event** group includes the trigger type (**BEFORE**, **AFTER**, or **INSTEAD OF**), the event type (**INSERT**, **DELETE**, or **UPDATE OF <column_list>**), and the table name (**ON <tablename>**). The **Condition** group includes the **REFERENCING** clause, the **FOR EACH** clause, and the **WHEN** clause. The **Action** group includes the SQL procedure statement, which can be a single statement or a block enclosed in **BEGIN ATOMIC** and **END** keywords. A red dot is placed above the **Event** label.

[[SQL]]