

ENCHARGED JOB

Joins – Triggers -History procedure

COURSE:

Diseño de base de datos

TEACHER:

Ubel

STUDENTS:

Juan Miguel Campos Marchan

Cristhian Elvis Velasquez Ventura

CYCLE:

Four

**Tingo María**

**2019**

INDEX

[OBJETIVES 1](#_Toc28230388)

[CHAPTER I : Joins 2](#_Toc28230389)

[1.1. Natural Joins 2](#_Toc28230390)

[1.2. Outer Joins 3](#_Toc28230391)

[1.3. Cartesian Products 3](#_Toc28230392)

[1.4. Equijoins 3](#_Toc28230393)

[1.5. Nonequijoins 4](#_Toc28230394)

[1.6. Self Joins 4](#_Toc28230395)

[1.7. Inner Joins 4](#_Toc28230396)

[1.8. Antijoins 4](#_Toc28230397)

[1.9. Semijoins 4](#_Toc28230398)

[CHAPTER II : Trigger 5](#_Toc28230399)

[2.1. Prerequisites 5](#_Toc28230400)

[2.2. Create or replace 6](#_Toc28230401)

[CHAPTER III : History procedure 8](#_Toc28230402)

[BIBLIOGRAPHY 9](#_Toc28230403)

# OBJETIVES

Understand the Joins theme, know the concepts of Triggers and History process

1. Know the concept and structure that a JOIN has
2. Learn to use all types of JOIN
3. Understand the operation of TRIGGERS

# CHAPTER I : Joins

(Oracle, 2013) tell us that

“A Join is a query that combines rows from two or more tables, views, or materialized views. Oracle Database performs a Join whenever multiple tables appear in the FROM clause of the query. The select list of the query can select any columns from any of these tables. If any two of these tables have a column name in common, then you must qualify all references to these columns throughout the query with table names to avoid ambiguity.”

Each Oracle function has conditions for its use, and Join is not indifferent, that is why (Oracle, 2013) points out:

“To execute a join of three or more tables, Oracle first joins two of the tables based on the join conditions comparing their columns and then joins the result to another table based on Join conditions containing columns of the joined tables and the new table. Oracle continues this process until all tables are joined into the result. The optimizer determines the order in which Oracle joins tables based on the join conditions, indexes on the tables, and, any available statistics for the tables.

WHERE clause that contains a Join condition can also contain other conditions that refer to columns of only one table. These conditions can further restrict the rows returned by the join query.”

According to (Puja & Brian, 2009) , the Joins that are compliant with the SQL 1999 standard are the following:

* Natural Joins:
* OUTER Joins:
  + Left Outer Join
  + Right Outer Join
* Cartesian Products
* Equijoins
* Nonequijoins
* Self Joins

## Natural Joins

According to (Puja & Brian, 2009) the operation of Join is as follows:

* The NATURAL JOIN clause is based on all columns in the two tables that have the same name.
* It selects rows from the two tables that have equal values in all matched columns.
* If the columns having the same names have different data types, an error is returned.

## Outer Joins

(Oracle, 2013) says that, an Outer Join extends the result of a simple join. An Outer Join returns all rows that satisfy the Join condition and also returns some or all of those rows from one table for which no rows from the other satisfy the Join condition.

There are 3 types of external combinations:

* Left Outer Join

This query retrieves all the rows in the left table, even if there is no match in the right table.

* Right Outer Join

Retrieves all the rows in the right table, even if there is no match in the left table.

* Full Outer Join

This query retrieves all rows in the left table, even if there is no match in the right table. It also retrieves all rows in the right table, even if there is no match in the left table.

## Cartesian Products

For this type of Join, (Oracle, 2013) tells us that if two tables in a Join query do not have a join condition, Database returns its Cartesian product. If a query Joins three or more tables and you do not specify a Join condition for a specific pair, then the optimizer may choose a Join order that avoids producing an intermediate Cartesian product.

(Puja & Brian, 2009) says that the following conditions must be met for a Cartesian product to form:

* A join condition is omitted
* A join condition is invalid
* All rows in the first table are joined to all rows in the second table

They also point out that Cross Joins are called Cartesian Products.

## Equijoins

For this type of Join (Oracle, 2013) tells us the following:

“An equijoin is a join with a join condition containing an equality operator. An equijoin combines rows that have equivalent values for the specified columns. Depending on the internal algorithm the optimizer chooses to execute the join, the total size of the columns in the equijoin condition in a single table may be limited to the size of a data block minus some overhead. The size of a data block is specified by the initialization parameter DB\_BLOCK\_SIZE.”

## Nonequijoins

## Self Joins

## Inner Joins

(Oracle Tutorial, 2017) refers to

“The query returns a result set by combining column values of both tables T1 and T2 based on the join predicate. It compares each row of table T1 with rows of table T2 to find all pairs of rows that satisfy the join predicate. Whenever the join predicate is satisfied by matching non-NULL values, column values for each matching pair of rows of T1 and T2 tables are combined into a row in the result set.”

For (Oracle, 2013) An inner join (sometimes called a simple join) is a join of two or more tables that returns only those rows that satisfy the join condition.

## Antijoins

According to (Oracle, 2013) un antijoin returns rows from the left side of the predicate for which there are no corresponding rows on the right side of the predicate. It returns rows that fail to match (NOT IN) the subquery on the right side.

## Semijoins

(Oracle, 2013) says that, a semijoin returns rows that match an EXISTS subquery without duplicating rows from the left side of the predicate when multiple rows on the right side satisfy the criteria of the subquery. Semijoin and antijoin transformation cannot be done if the subquery is on an OR branch of the WHERE clause.

# CHAPTER II : Trigger

(Moisset, 2017) tell us that a Trigger is a block of code that runs automatically when an event occurs. It’s created to record the changes that are made on the tables and the identity of the person who made them.

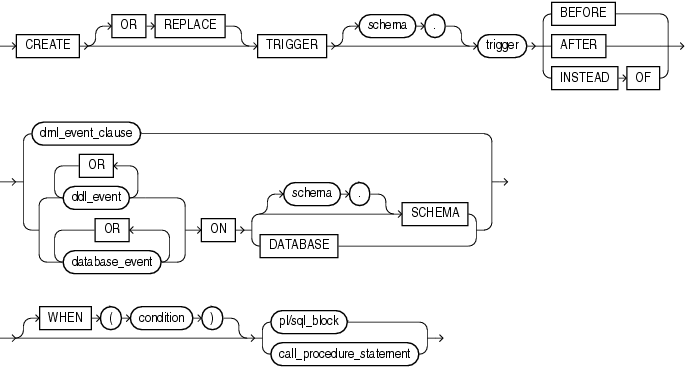
He also refers that:

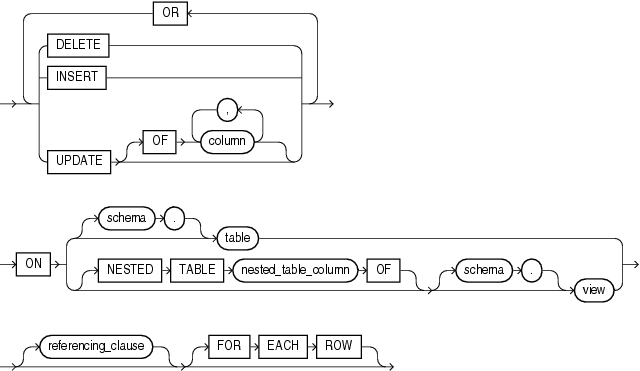
* They cannot be invoked directly; when trying to modify the data of a table associated with a trigger, the trigger is automatically executed.
* Doesn´t receive and return parameters.
* Is appropriate to maintain data integrity, not to obtain query results.

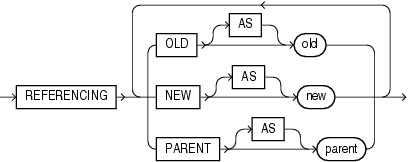
## 2.1. Prerequisites

According to Oracle:

1. Before a trigger can be created, the user SYS must run a SQL script commonly called DBMSSTDX.SQL. The exact name and location of this script depend on your operating system.
2. To create a trigger in your own schema on a table in your own schema or on your own schema (SCHEMA), you must have the CREATE TRIGGER system privilege.
3. To create a trigger in any schema on a table in any schema, or on another user's schema (schema.SCHEMA), you must have the CREATE ANY TRIGGER system privilege.
4. In addition to the preceding privileges, to create a trigger on DATABASE, you must have the ADMINISTER DATABASE TRIGGER system privilege.
5. If the trigger issues SQL statements or calls procedures or functions, then the owner of the trigger must have the privileges necessary to perform these operations. These privileges must be granted directly to the owner rather than acquired through roles.







## 2.2. Create or replace

(Oracle Tutorial, 2017) tell us the CREATE keyword specifies that you are creating a new trigger. The OR REPLACE keywords are optional. They are used to modify an existing trigger.

There is something interesting with the Trigger, and if the next day we decide to modify the Trigger and the OR REPLACE is not set, it will throw an error. To avoid that it is better to add the OR REPLACE for each modification.

(Oracle Tutorial, 2017) give us extra instructions:

1. Trigger name

Specify the name of the trigger that you want to create after the CREATE OR REPLACE keywords.

1. BEFORE | AFTER

The BEFORE or AFTER option specifies when the trigger fires, either before or after a triggering event e.g., INSERT, UPDATE, or DELETE

1. ON table\_name

The table\_name is the name of the table associated with the trigger.

1. FOR EACH ROW

The clause FOR EACH ROW specifies that the trigger is a row-level trigger. A row-level trigger fires once for each row inserted, updated, or deleted.

Besides the row-level triggers, we have statement-level triggers. A statement-trigger fire once regardless of the number of rows affected by the triggering event. If you omit the FOR EACH ROW clause, the CREATE TRIGGER statement will create a statement-level trigger.

1. ENABLE / DISABLE

The ENABLE / DISABLE option specifies whether the trigger is created in the enabled or disabled state. Note that if a trigger is disabled, it is not fired when the triggering event occurs.

By default, if you don’t specify the clause ENABLE / DISABLE , the trigger is created with the enabled state.

1. FOLLOWS | PRECEDES another\_trigger

For each triggering event e.g., INSERT, UPDATE, or DELETE, you can define multiple triggers to fire. In this case, you need to specify the firing sequence using the FOLLOWS or PRECEDES option.

Let’s create a trigger to see understand how it works.

# CHAPTER III : History procedure

# BIBLIOGRAPHY

Moisset, D. (2017, Julio 4). *Tutoriales programacion ya*. Retrieved from Descripción : Disparador (trigger) (Oracle): https://www.tutorialesprogramacionya.com/oracleya/temarios/descripcion.php?inicio=100&cod=261&punto=103

Oracle. (2013, Setiembre 6). *Oracle*. Retrieved from Database SQL Language Reference: https://docs.oracle.com/cd/B28359\_01/server.111/b28286/queries006.htm?fbclid=IwAR0PMgdH1Q6nD-XEnvP3mEzkWggz9pyP\_KnfMKxzjOoGsEgF0KFuzMQhpTE#SQLRF52331

Oracle Tutorial. (2017, Agosto). *Oracle Tutorial*. Retrieved from Oracle INNER JOIN Demonstrated with Practical Examples: https://www.oracletutorial.com/oracle-basics/oracle-inner-join/

Puja, S., & Brian, P. (2009). *Oracle Database 11g: SQL.* California.