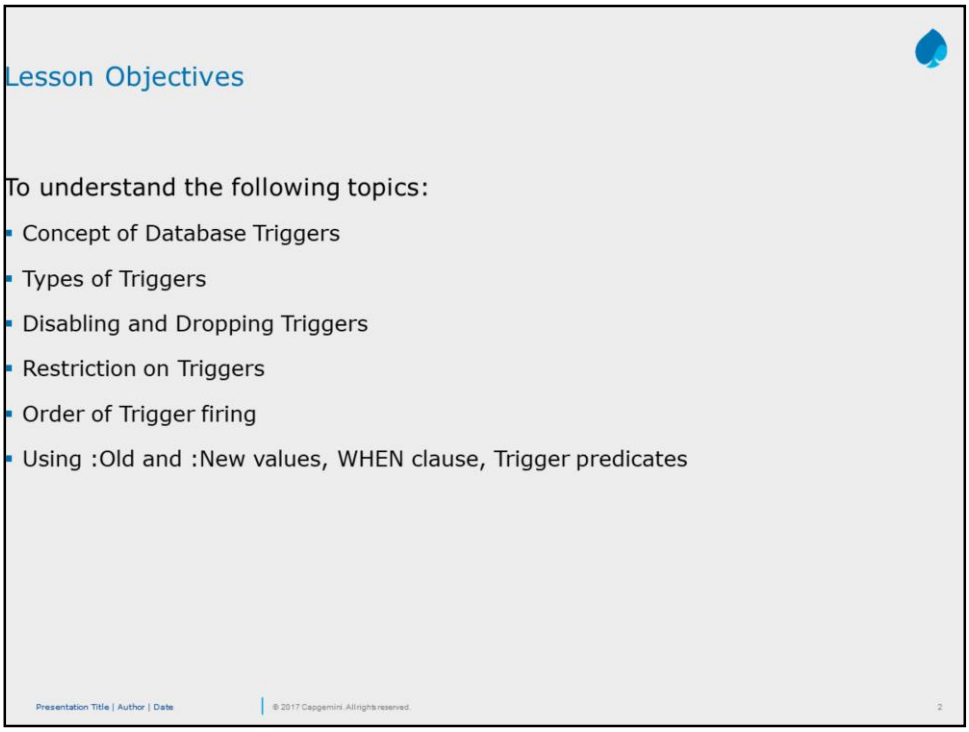
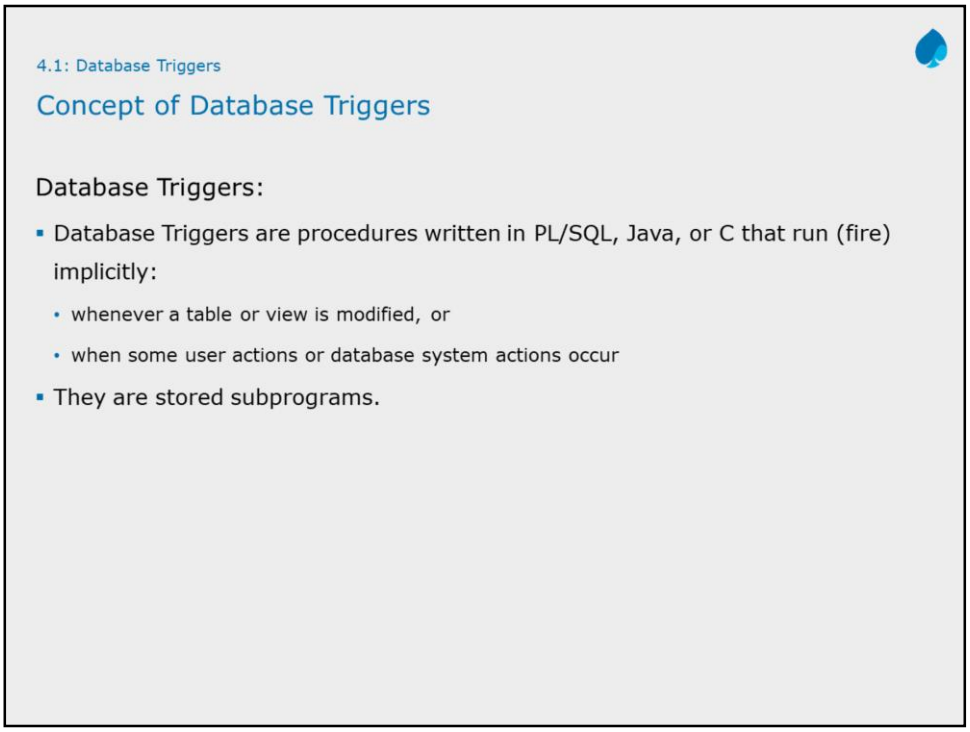
Instructor Notes:

Instructor Notes:



Instructor Notes:



Database Triggers:

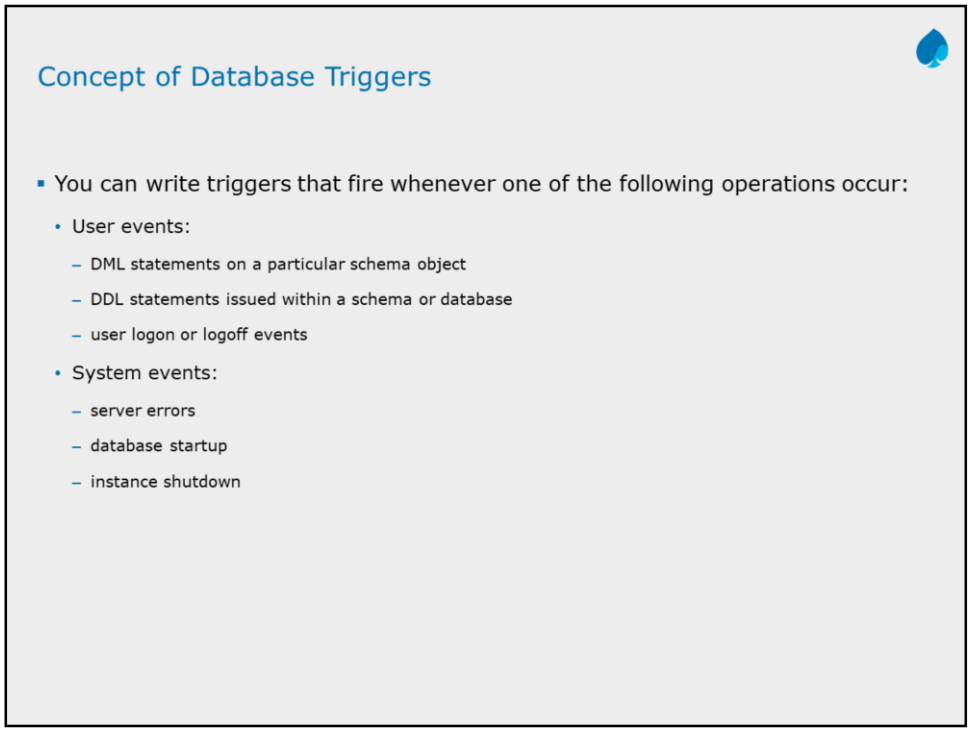
A Trigger defines an action the database should take when some database related event occurs.

Anonymous blocks as well as stored subprograms need to be explicitly invoked by the user.

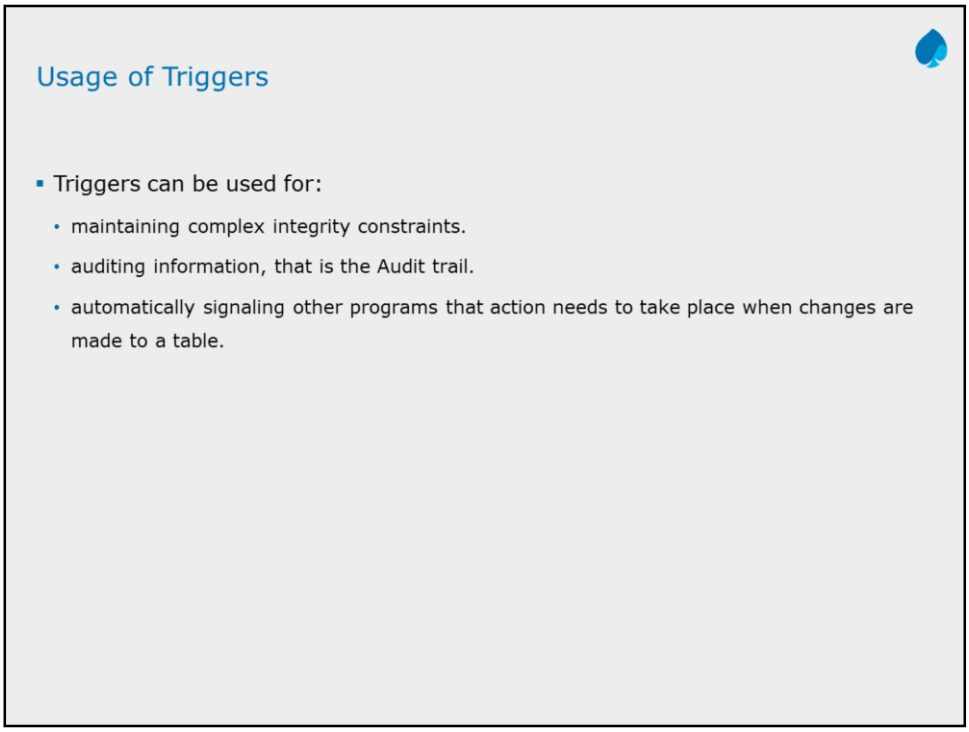
Database Triggers are PL/SQL blocks, which are implicitly fired by ORACLE RDBMS whenever an event such as INSERT, UPDATE, or DELETE takes place on a table.

Database triggers are also stored subprograms.

Instructor Notes:



Instructor Notes:



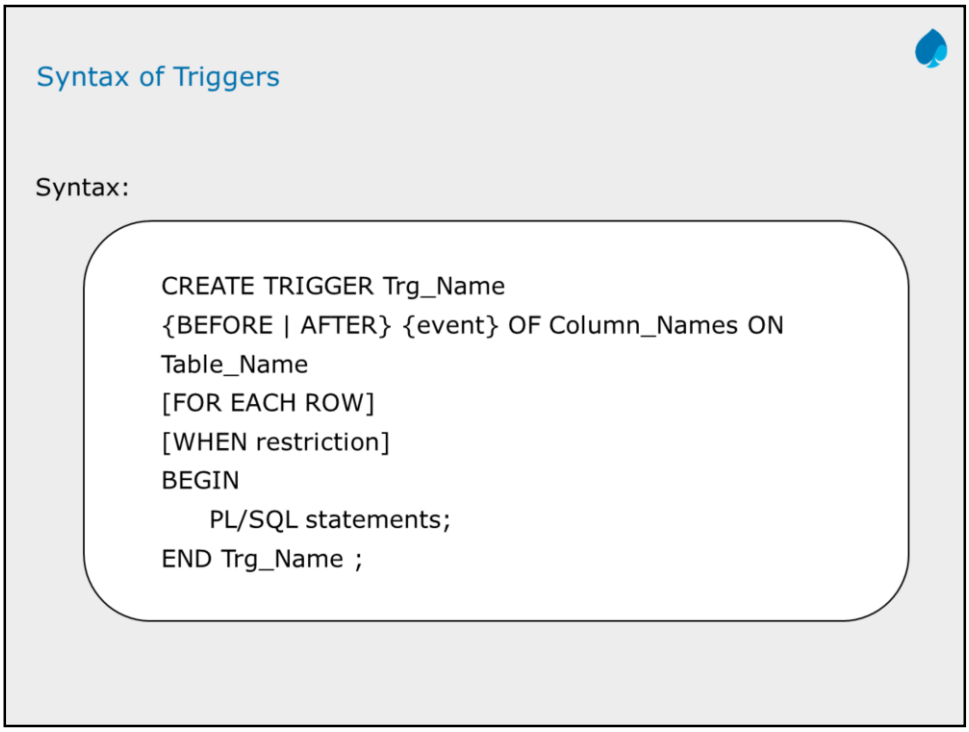
Database Triggers (contd.):

Triggers can be used for:

- Maintaining complex integrity constraints. This is not possible through declarative constraints that are enabled at table creation.
- Auditing information in a table by recording the changes and the identify of the person who made them. This is called as an audit trail.
- Automatically signaling other programs that action needs to take place when changes are made to a table.

contd.

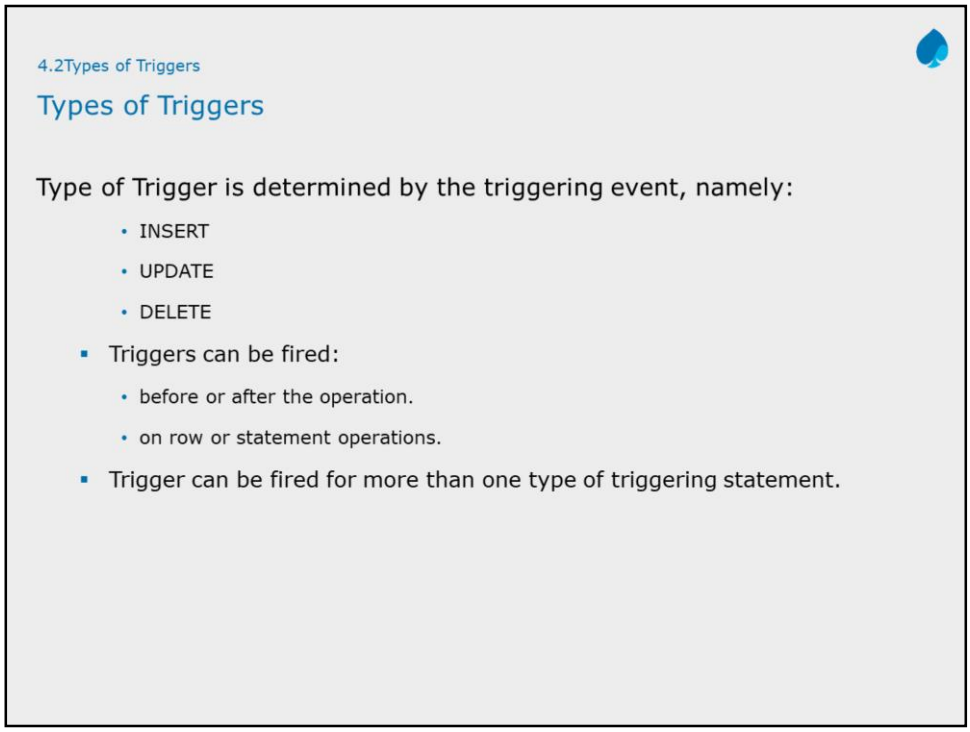
Instructor Notes:



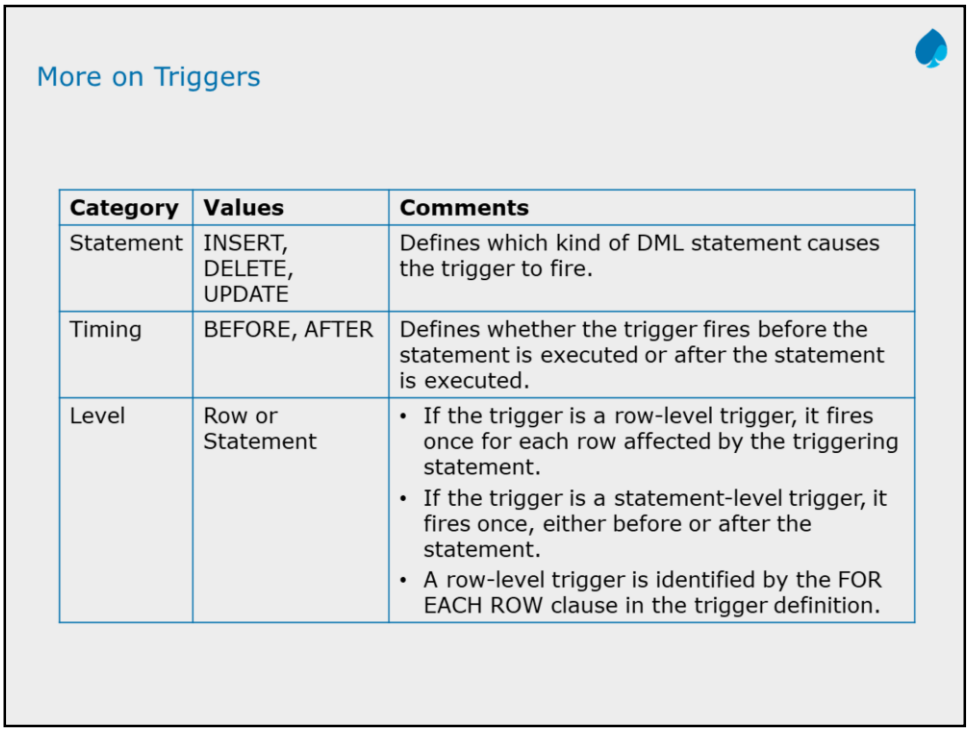
Database Triggers (contd.):
To create a trigger on a table you must be able to alter that table. The slide shows the syntax for creating a table.

contd.

Instructor Notes:



Instructor Notes:



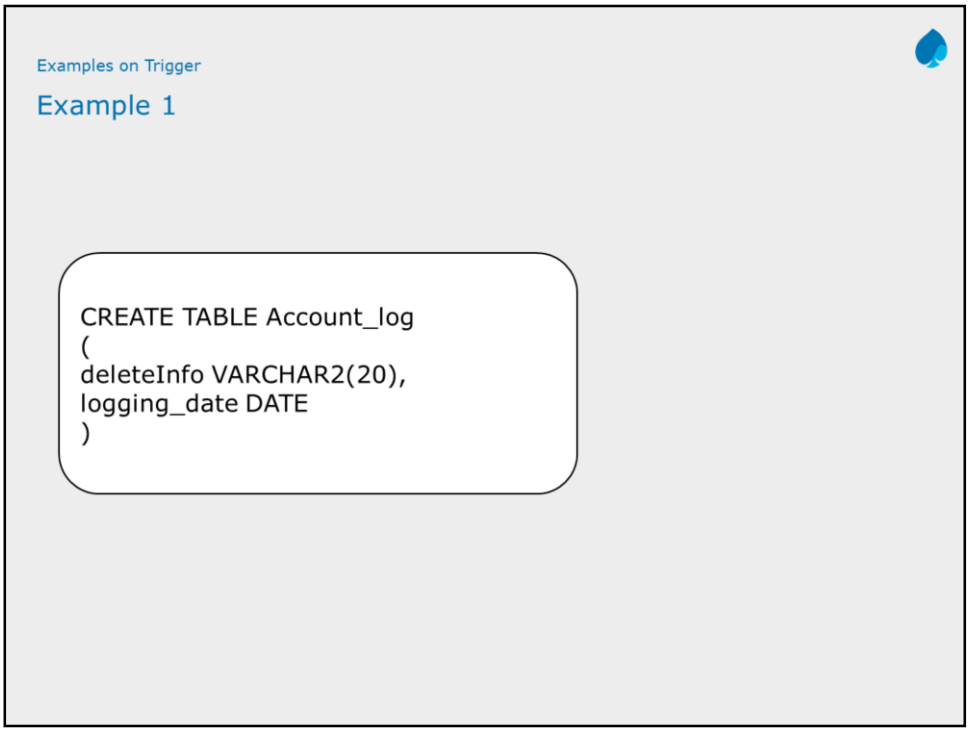
More on Triggers

Category	Values	Comments
Statement	INSERT, DELETE, UPDATE	Defines which kind of DML statement causes the trigger to fire.
Timing	BEFORE, AFTER	Defines whether the trigger fires before the statement is executed or after the statement is executed.
Level	Row or Statement	• If the trigger is a row-level trigger, it fires once for each row affected by the triggering statement. • If the trigger is a statement-level trigger, it fires once, either before or after the statement. • A row-level trigger is identified by the FOR EACH ROW clause in the trigger definition.

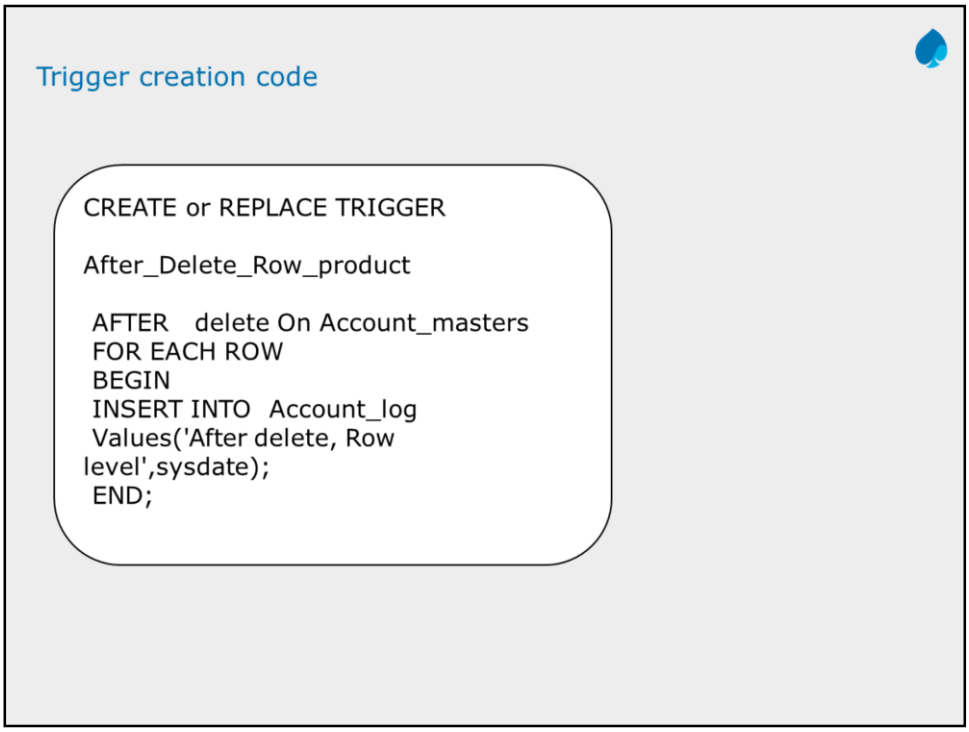
Note:

- A trigger can be fired for more than one type of triggering statement. In case of multiple events, OR separates the events.
- From ORACLE 7.1 there is no limit on number of triggers you can write for a table.
    - Earlier you could write maximum of 12 triggers per table, one of each type.

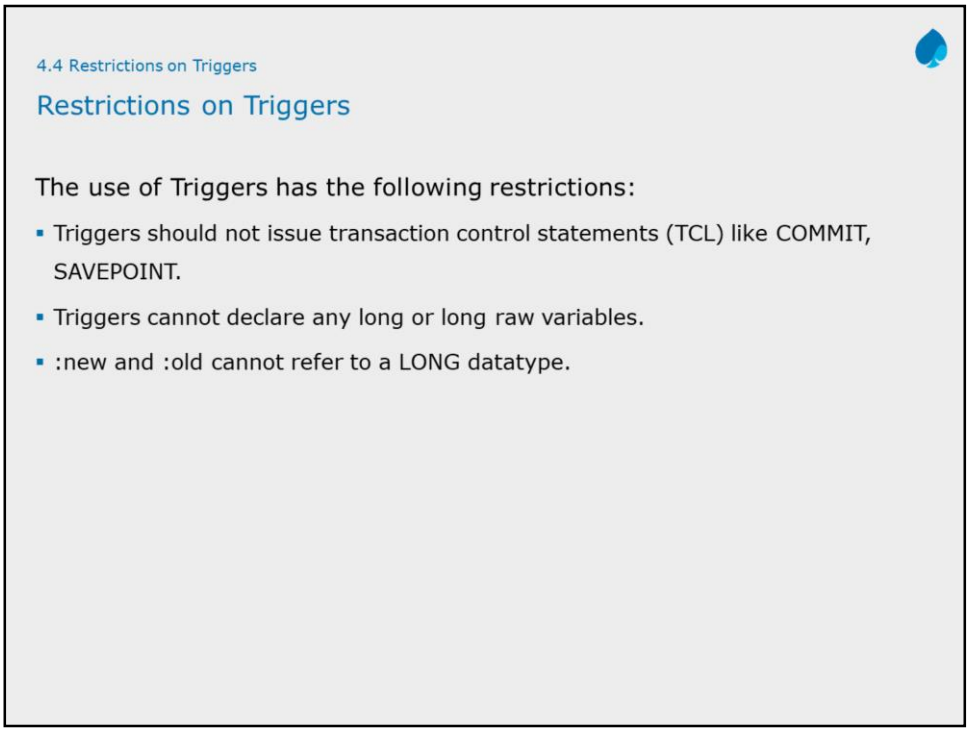
Instructor Notes:



Instructor Notes:



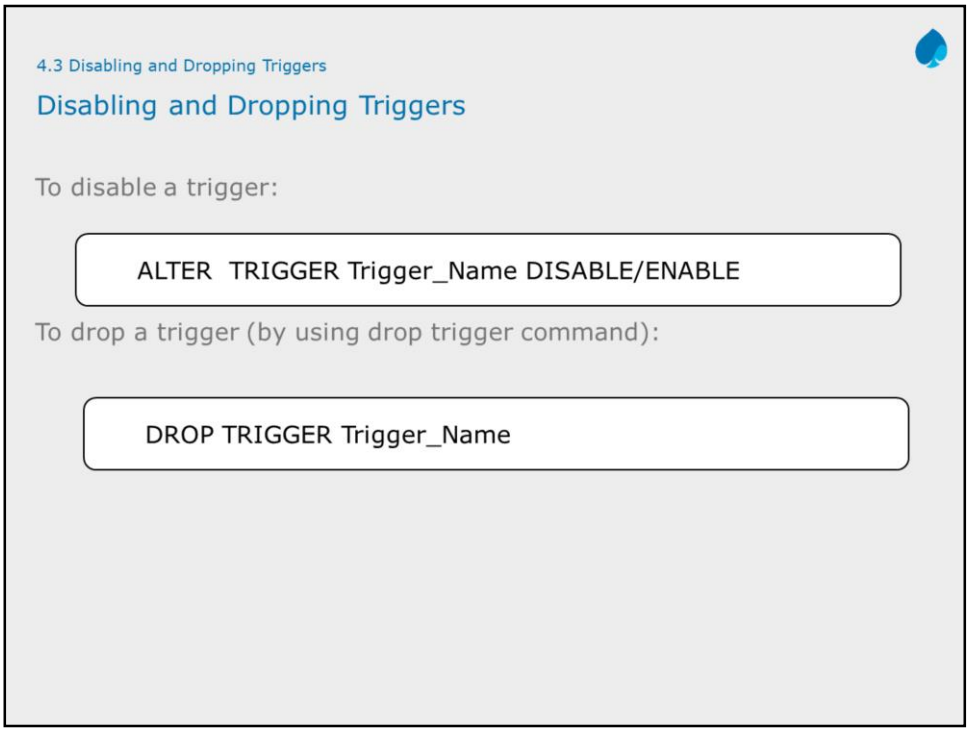
Instructor Notes:



Restrictions on Triggers:

- A Trigger may not issue any transaction control statements (TCL) like COMMIT, SAVEPOINT.
    - Since the triggering statement and the trigger are part of the same transaction, whenever triggering statement is committed or rolled back the work done in the trigger gets committed or rolled back, as well.
- Any procedures or functions that are called by the Trigger cannot have any transaction control statements.
- Trigger body cannot declare any long or long raw variables. Also :new and :old cannot refer to a LONG datatype.
- There are restrictions on the tables that a trigger body can access depending on type of “triggers” and “constraints” on the table.

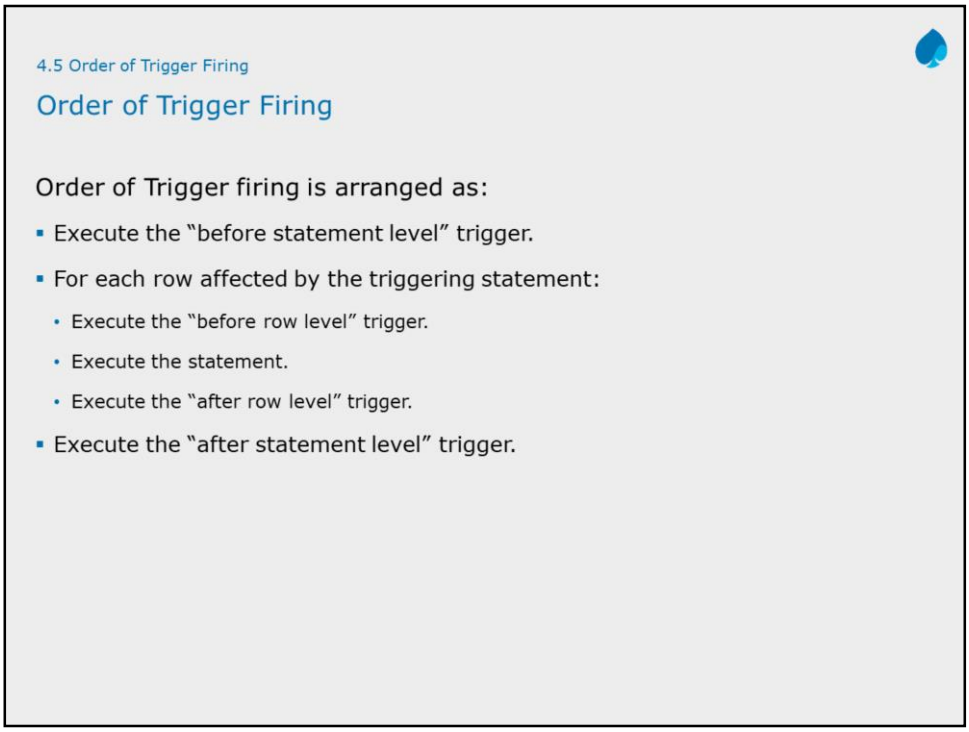
Instructor Notes:



Note:

An UPDATE or DELETE statement affects multiple rows of a table.
If the trigger is fired once for each affected row, then FOR EACH ROW clause is required. Such a trigger is called a ROW trigger.
If the FOR EACH ROW clause is absent, then the trigger is fired only once for the entire statement. Such triggers are called STATEMENT triggers

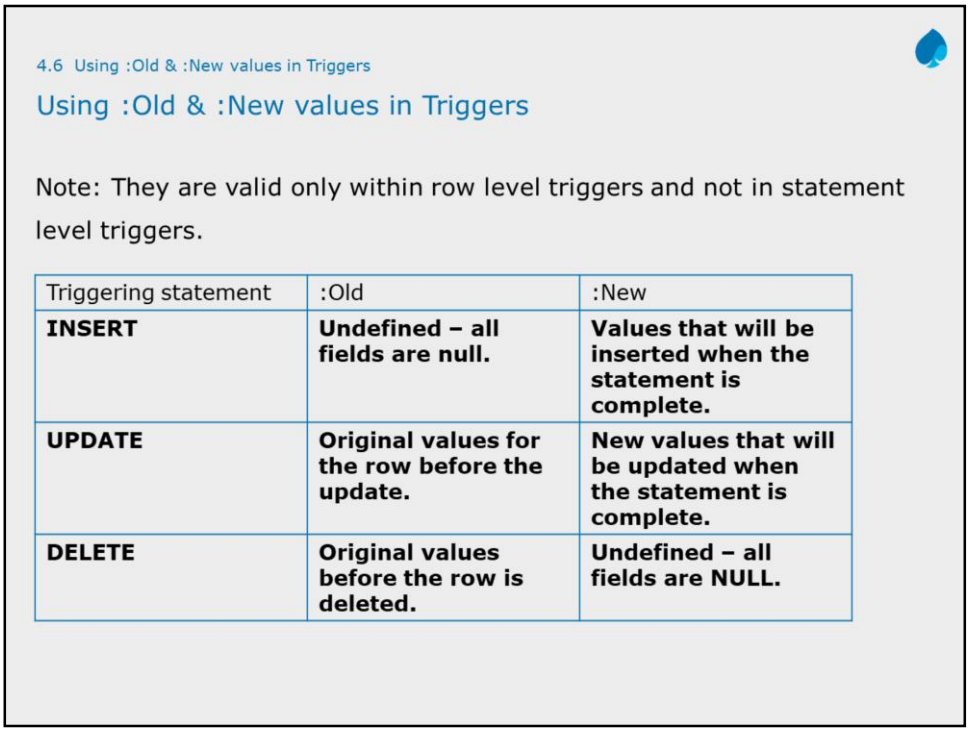
Instructor Notes:



Note:

- As each trigger is fired, it will see the changes made by the earlier triggers, as well as the database changes made by the statement.
- The order in which triggers of same type are fired is not defined.
- If the order is important, combine all the operations into one trigger.

Instructor Notes:



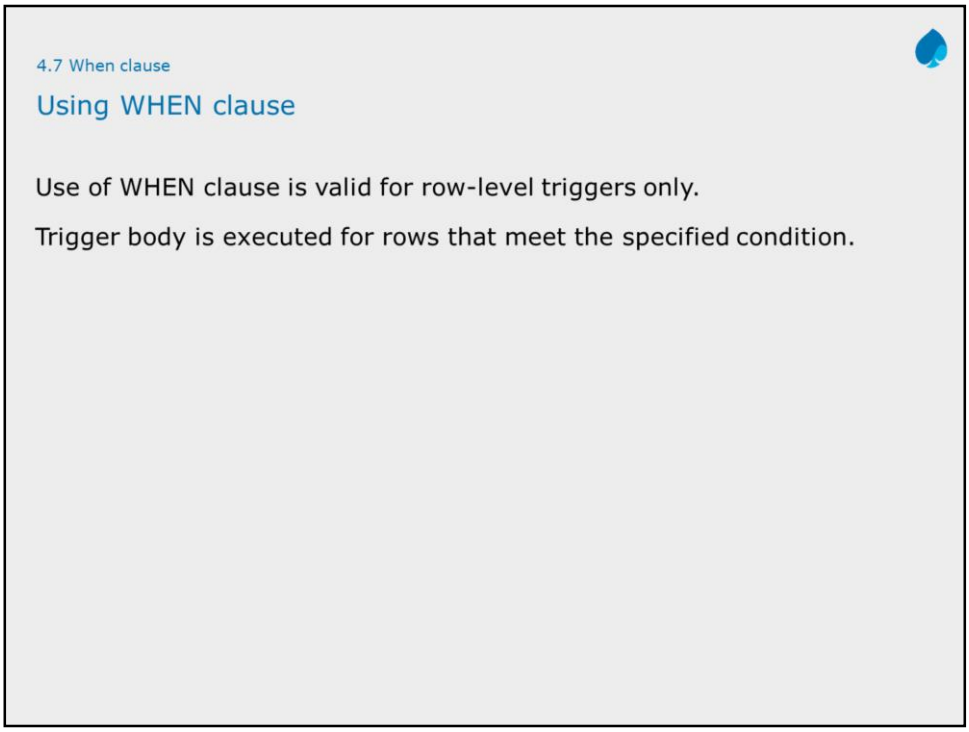
Using :old and :new values in Row Level Triggers:

- Row level trigger fires once per row that is being processed by the triggering statement.
- Inside the trigger, you can access the row that is currently being processed. This is done through keywords :new and :old (they are called as pseudo records). The meaning of the terms is as shown in the slide.

Note:

- The pseudo records are valid only within row level triggers and not in statement level triggers.
  - :old values are not available if the triggering statement is INSERT.
  - :new values are not available if the triggering statement is DELETE.
- Each column is referenced by using the notation :old.Column_Name or :new.Column_Name.
- If a column is not updated by the triggering update statement, then :old and :new values remain the same.

Instructor Notes:

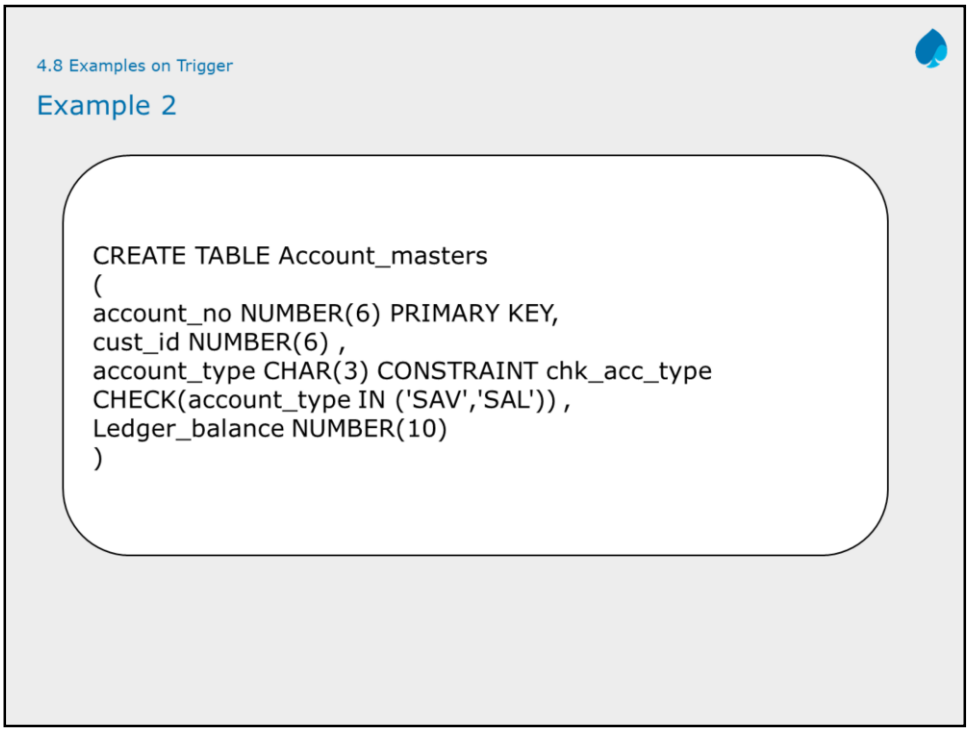


Using WHEN Clause:

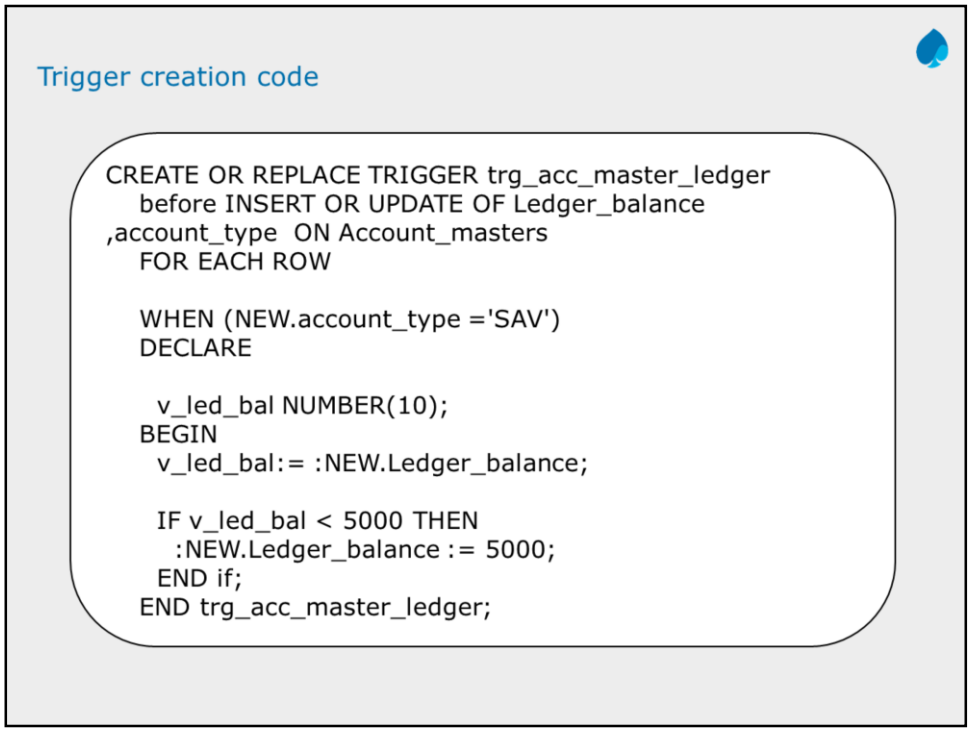
- The WHEN clause is valid for row-level triggers only. If present, the trigger body will be executed only for those rows that meet the condition specified by the WHEN clause.
- The :new and :old records can be used here without colon.

contd.

Instructor Notes:



Instructor Notes:



Instructor Notes:

Summary

Database Triggers are procedures written in PL/SQL, Java, or C that run (fire) implicitly: Database Triggers are procedures written in PL/SQL, Java, or C that run (fire) implicitly:

- whenever a table or view is modified, or
- when some user actions or database system actions occur

There are three types of triggers:

- Statement based triggers
- Timing based triggers
- Level based triggers

Instructor Notes:

Summary

Disabling and Dropping triggers can be done instead of actually removing the triggers

Order of trigger firing is decided depending on the type of triggers used in the sequence

Instructor Notes:

Answers to Review Questions
Question 1: True
Question 2:False

Review – Questions

Question 1: Triggers should not issue Transaction Control Statements (TCL).

- True / False

Question 2: The :new and :old records must be used in WHEN clause with a colon.

Instructor Notes:

Answers to Review Questions
Question 3: Mutating
Question 4: Statement level

Review – Questions

Question 3: A ___ is a table that is currently being modified by a DML statement.

Question 4: A ___ is fired once on behalf of the triggering statement, regardless of the number of rows in the table that the triggering statement affects.