



# Physical Design

## *Week 5*

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

**Improving Database Performance**  
**Triggers**

Assignment Project Exam Help

TRIGGERS

<https://tutorcs.com>

WeChat: cstutorcs

---

# What is a trigger?

- Database Object
- A Stored Block of Code
  - With **[Declare], Begin, End.**
  - Attached to an event.
  - When that event occurs the trigger code is fired.
- Associated with a database table or action

# What is a trigger?

- Called an **event-condition-action (ECA) rule**
  - When event occurs
    - test condition;
    - if condition is satisfied, execute action

<https://tutorcs.com>

WeChat: estutores

# What is a trigger?

- Fires **automatically** when certain DML action is carried out on the table:
  - *Before* or *after* an event
  - Change may be INSERT, DELETE, UPDATE
  - Can be performed on a *single* or *multiple* rows

# Trigger Uses

1. Auditing **Assignment Project Exam Help**
  - Write information about (sensitive) data modifications to an audit table **<https://tutorcs.com>**
  - May include old and new values, user, timestamp
    - E.g. new and old salary **WeChat: cstutorcs**
2. Data Integrity
  - Implement checks on data against business rules
  - Can compare with live database values
  - NEW and OLD values can be compared
    - E.g. prices cannot be reduced for some items

# Trigger Uses

## 3. Referential integrity

- Allows implementation of a "cascade update"
  - E.g. if authID (a/D) is changed, appropriately change authID in foreign key

## 4. Derived data

- Update any stored derived data when base data changes
  - E.g. if total number of employees is stored, add 1 if new employee added

# Trigger Uses

5. Security **Assignment Project Exam Help**
  - Logging of database access
  - E.g. date and time each user logs on
  - E.g. deny access at weekend
6. Maintaining synchronous replication **https://tutorcs.com**
  - In a distributed database
7. Generating statistics on table access **WeChat: cstutorcs**



# Trigger Uses - Examples

- Automatically generate derived column values
  - Prevent invalid transactions
  - Enforce complex security authorizations
  - Enforce referential integrity across nodes in a distributed database
  - Enforce complex business rules
- Provide transparent event logging
- Provide auditing
- Maintain synchronous table replicates
- Gather statistics on table access
- Modify table data when DML statements are issued against views
- Publish information about database events, user events, and SQL statements to subscribing applications

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Trigger General Types

- DML statements (INSERT, UPDATE, DELETE)
  - on a particular table or view, issued by any user
- DDL statements (CREATE or ALTER primarily)
  - issued either by a particular schema/user or by any schema/user in the database
- Database events, such as logon/logoff, errors, or startup/shutdown
  - issued either by a particular schema/user or by any schema/user in the database

# Trigger General Types

- Triggers are similar to stored procedures.
- A trigger can include SQL and PL/SQL or Java statements to run as a unit and can invoke stored procedures.
- Key difference – how they are invoked:
  - A stored procedure is explicitly run by a user, application, or trigger.
  - Triggers are implicitly fired by Oracle when a triggering event occurs
    - It does matter which user is connected or which application is being used.

# Trigger General Types

- A trigger has three basic parts:
  - A triggering event or statement
  - A trigger restriction
  - A trigger action

# Trigger Options

- Possible events include:
  - INSERT ON table
  - DELETE ON table
  - UPDATE [OF column] ON table
- Granularity—trigger can be activated:
  - FOR EACH ROW modified
  - FOR EACH STATEMENT that performs modification
- Timing—action can be executed:
  - AFTER or BEFORE the triggering event
  - INSTEAD OF the triggering event on views (more later)

# Trigger Naming

- Must be unique within a schema
- Can have same name as table on which it is defined but this may be confusing to developers

WeChat: cstutorcs

# Trigger Example

## REORDER Trigger

AFTER UPDATE OF parts_on_hand ON inventory	Triggering Statement
WHEN (new.parts_on_hand < new.reorder_point)	Trigger Restriction
<div>Triggered Action</div> <pre>FOR EACH ROW DECLARE     NUMBER X; BEGIN     SELECT COUNT(*) INTO X     FROM pending_orders     WHERE part_no=:new.part_no;      IF x = 0     THEN         INSERT INTO pending_orders         VALUES (new.part_no, new.reorder_quantity, sysdate);     END IF; END;</pre> <div>/* a dummy variable for counting */</div> <div>/* query to find out if part has already been */</div> <div>/* reordered-if yes, x=1, if no, x=0 */</div> <div>/* part has not been reordered yet, so reorder */</div> <div>/* part has already been reordered */</div>	

# Triggering Statement

- A triggering event or statement is the SQL statement, database event, or user event that causes a trigger to fire.
- A triggering event can be one or more of the following:
  - An INSERT, UPDATE, or DELETE statement on a specific table (or view, in some cases)
  - A CREATE, ALTER, or DROP statement on any schema object
  - A database startup or instance shutdown
  - A specific error message or any error message
  - A user logon or logoff

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# Triggering Statement

- Our Example: from slide 15

- Triggering statement:

<https://tutorcs.com>

... UPDATE OF parts\_on\_hand ON inventory ...

- When the parts\_on\_hand column of a row in the inventory table is updated, fire the trigger.
  - The trigger will create an order for the part if the level in stock has fallen below the required level

# Trigger Restriction

- Specifies a Boolean expression that must be true for the trigger to fire.
- The trigger action is not run if the trigger restriction evaluates to false or unknown.
- Our Example: from slide 15:

```
new.parts_on_hand < new.reorder_point
```

- This trigger will not fire unless the number of available parts is less than a present reorder amount.

# Trigger Action

- The procedure (PL/SQL block, Java program, or C callout) that contains the SQL statements and code to be run when the following events occur:
  - A triggering statement is issued.
  - The trigger restriction evaluates to true.

# Statement v Row Trigger

- When you define a trigger, you can specify the number of times the trigger action is to be run:  
<https://tutorcs.com>
- Once for every row affected by the triggering statement, such as a trigger fired by an UPDATE statement that updates many rows
- Once for the triggering statement, no matter how many rows it affects

# Statement Level Trigger

- Fired whenever a trigger event occurs on a table regardless of how many rows are affected.
  - A statement-level trigger executes once for each transaction.
  - E.g. if you update 1000 rows in a table, then a statement-level trigger on that table would only be executed once..

# Statement Level Trigger - Syntax

CREATE [OR REPLACE] TRIGGER trigger\_name

{BEFORE | AFTER } triggering\_event

ON table\_name

[FOLLOWS | PRECEDES another\_trigger]

[ENABLE / DISABLE ]

[WHEN condition]

DECLARE

declaration statements

BEGIN

executable statements

EXCEPTION

exception\_handling statements

END;

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Statement Level Trigger - Example

CUSTOMERS
* CUSTOMER_ID
NAME
ADDRESS
WEBSITE
CREDIT_LIMIT

Suppose, you want to restrict users so that the credit limit of customers can only be done between 28th to 31st of every month to make it easier to get an accurate monthly picture of customer business at the end of the month.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Statement Level Trigger - Example

```
CREATE OR REPLACE TRIGGER customers_credit_trg
  BEFORE UPDATE OF credit_limit
  ON customers
DECLARE
  l_day_of_month NUMBER;
BEGIN
  -- determine the day of month of the current date
  l_day_of_month := EXTRACT(DAY FROM sysdate);

  IF l_day_of_month BETWEEN 28 AND 31 THEN
    raise_application_error(-20100,'Cannot update customer credit from
28th to 31st');
  END IF;
END;
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# Statement Level Trigger - Example

CREATE OR REPLACE TRIGGER **customers\_credit\_trg**

**Assignment Project Exam Help**

First, create a new trigger `customers_credit_trg`.

The OR REPLACE modifies the trigger if it already exists.

**WeChat: cstutorcs**

<https://tutorcs.com>

# Statement Level Trigger - Example

BEFORE UPDATE OF credit\_limit

ON customers

Assignment Project Exam Help

Then tell Oracle when to trigger the event – here it says only before an update event for the credit\_limit column of the customers table.

<https://tutorcs.com>  
WeChat: cstutorcs

If you update values in other columns rather than the credit\_limit column, the trigger will not execute.

# Statement Level Trigger - Example

```
DECLARE
```

```
l_day_of_month NUMBER;
```

Assignment Project Exam Help

Create a variable to hold the current day of the month which we will retrieve from sysdate.

<https://tutorcs.com>

WeChat: cstutorcs

```
l_day_of_month := EXTRACT(DAY FROM sysdate);
```

# Statement Level Trigger - Example

```
IF l_day_of_month BETWEEN 28 AND 31 THEN  
    raise_application_error(-20100,'Cannot update customer credit from  
    28th to 31st');  
END IF;
```

<https://tutorcs.com>

**WeChat: cstutorcs**

Now, check whether the day of the month is between 28 and 31. If it is raise an application error.

This is defined by the application developer – using the numbering defined for that application (it is not controlled by Oracle)

# Statement Level Trigger – Example Testing

UPDATE

customers

SET

credit\_limit = credit\_limit \* 110;

Assignment Project Exam Help

<https://tutorcs.com>

Current day of the month < 20

WeChat: cstutorcs

Trigger should succeed

If we are outside the limits we will see:

```
UPDATE
```

```
customers
```

```
SET
```

```
credit_limit = credit_limit * 110
```

```
Error report -
```

```
ORA-20100: Cannot update customer credit from 28th to 31st
```

```
ORA-06512: at "DL_DEV_USER.CUSTOMERS_CREDIT_TRG", line 8
```

```
ORA-04088: error during execution of trigger 'DL_DEV_USER.CUSTOMERS_CREDIT_TRG'
```

# Statement Level Trigger – Example Testing

Note:

Oracle automatically rollbacks the update because we call the `raise_application_error` procedure inside the trigger.

<https://tutorcs.com>

WeChat: cstutorcs

# Row Level Trigger

- Fires once for each row affected by the triggering event such as INSERT, UPDATE, or DELETE.
- Useful for data-related activities such as data auditing and data validation.
- To create a new row-level trigger, you use the CREATE TRIGGER statement with the FOR EACH ROW clause.

# Row Level Trigger - Syntax

```
CREATE OR REPLACE TRIGGER trigger_name
```

```
  BEFORE | AFTER
```

```
  INSERT OR DELETE OR UPDATE OF column1, column2,...
```

```
  ON table_name
```

```
  FOR EACH ROW
```

```
  REFERENCING OLD AS old_name
```

```
  NEW AS new_name
```

```
  WHEN (condition)
```

```
DECLARE
```

```
  ...
```

```
BEGIN
```

```
  ...
```

```
EXCEPTION
```

```
  ...
```

```
END;
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# Row Level Trigger - Syntax

The features which are applicable to the statement-level triggers are also available the row-level trigger:

Assignment Project Exam Help

- Use `raise_application_error()` to raise a user-defined exception
- Check if the current event is INSERT, UPDATE, and DELETE using the INSERTING or DELETING or UPDATING.

<https://tutorcs.com>

WeChat: cstutorcs

# Row Level Trigger – Transition Variables

- Because we are working at row level in a row level trigger when a DML statement changes a column the old and new values are visible to the executing code  
<https://tutorcs.com>
- This is done by prefixing the table column with :old or :new  
WeChat: cstutorcs
- **:new** is useful for INSERT and UPDATE
- **:old** is useful for DELETE and UPDATE

# Row Level Trigger - Example

```
CREATE OR REPLACE TRIGGER customers_update_credit_trg
BEFORE UPDATE OF credit_limit
ON customers
FOR EACH ROW
WHEN (NEW.credit_limit > 0)
BEGIN
    -- check the credit limit
    IF :NEW.credit_limit >= 2 * :OLD.credit_limit THEN
        raise_application_error(-20101,'The new credit ' || :NEW.credit_limit ||
        ' cannot increase to more than double, the current credit ' ||
        :OLD.credit_limit);
    END IF;
END;
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Row Level Trigger - Example

```
CREATE OR REPLACE TRIGGER customers_update_credit_trg
  BEFORE UPDATE OF credit_limit
  ON customers
  FOR EACH ROW
  WHEN (NEW.credit_limit > 0)
BEGIN
  -- check the credit limit
  IF :NEW.credit_limit >= 2 * :OLD.credit_limit THEN
    raise_application_error(-20101,'The new credit ' || :NEW.credit_limit ||
      ' cannot increase to more than double, the current credit ' ||
      :OLD.credit_limit);
  END IF;
END;
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Row Level Trigger – Example Testing

We will pick a customer:

Assignment Project Exam Help

Suppose customer 10 has a credit limit of 2200.

If we execute the following update:

WeChat: cstutorcs

```
UPDATE customers
SET credit_limit = 5000
WHERE customer_id = 10
Error report -
```

```
ORA-20101: The new credit 5000 cannot increase to more than double, the current credit 2200
ORA-06512: at "DL_DEV_USER.CUSTOMERS_UPDATE_CREDIT_TRG", line 4
ORA-04088: error during execution of trigger 'DL_DEV_USER.CUSTOMERS_UPDATE_CREDIT_TRG'
```

# Raise\_application\_error

- Used inside trigger
- Purpose:
  - output an error message and
  - immediately **stop** the event that fired the trigger
  - For example, data insertion
- Can include variables/trigger values, see previous slide
- E.g.

```
RAISE_APPLICATION_ERROR(-20000, 'trigger violated')
```

label

Message text

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Compilation  
errors

When you create a trigger, Oracle  
responds

"Trigger created" or

"Warning: Trigger  
created with  
compilation errors."

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Type in the command

**SHOW ERRORS**

on its own to make the error messages  
visible

What to  
think  
about

---

BEFORE or AFTER?

---

INSERT, DELETE, UPDATE?

Assignment Project Exam Help

---

<https://tutorcs.com>  
FOR EACH ROW or once  
only?

WeChat: cstutorcs

---

Any error  
conditions/messages?

---

How can I test that the  
trigger works?



What  
else can  
we do  
with  
triggers

DROP

DROP TRIGGER trigger\_name;  
• No further firing will occur when dropped

ALTER

ALTER TRIGGER trigger\_name DISABLE;

ALTER

ALTER TRIGGER trigger\_name ENABLE;

ALTER

ALTER TABLE table\_name DISABLE ALL TRIGGERS;

ALTER

ALTER TABLE table\_name ENABLE ALL TRIGGERS;

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Trigger  
problems

---

Easy to confuse `:=` with `=` with  
`=` :old

---

Compilation errors:

Assignment Project Exam Help  
use `SHOW ERRORS` to see them

<https://tutorcs.com>

---

Errors may need to be solved in  
a different place from where  
they occur

---

"table is mutating"