Triggers in Oracle SQL are special stored procedures that automatically execute (or "fire") in response to specific events on a particular table or view. They are commonly used for maintaining data integrity, enforcing business rules, and auditing changes.

Types of Triggers
Row-Level Triggers: Execute for each row affected by the triggering event.
Statement-Level Triggers: Execute once for the triggering event, regardless of the number of rows affected.
Trigger Timing
BEFORE: Fires before the triggering event (INSERT, UPDATE, DELETE).
AFTER: Fires after the triggering event.
INSTEAD OF: Used mainly for views, it replaces the triggering event with the trigger action.

Syntax:
Here’s a basic syntax for creating a trigger:

```
CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF}
{INSERT | UPDATE | DELETE}
ON table_name
[FOR EACH ROW]
DECLARE
   -- variable declarations
BEGIN
   -- trigger logic
END;
```

Example 1:

~~~
``````````

CREATE OR REPLACE TRIGGER check_salary
BEFORE UPDATE OF salary ON employees
FOR EACH ROW
BEGIN
   IF :NEW.salary < 0 THEN
      RAISE_APPLICATION_ERROR(-20001, 'Salary cannot be negative.');
   END IF;
END;
/

ALTER TRIGGER trigger_name ENABLE;
ALTER TRIGGER trigger_name DISABLE;
~~~