A trigger is an event within the DBMS that can cause some code to execute automatically.

There are four types of database triggers:

1. Table-level triggers can initiate activity before or after an INSERT, UPDATE, or DELETE event.
2. View-level triggers defines what can be done to the view.
3. Database-level triggers can be activated at startup and shutdown of a database.
4. Session-level triggers can be used to store specific information.

```
 create table company(
  2     product_id        number(4)    not null,
  3     company_id          NUMBER(8)    not null,
  4     company_short_name  varchar2(30) not null,
  5     company_long_name   varchar2(60)
  6  );

Table created.

 insert into company values(1,1001,'A Inc.','Long Name A Inc.');

1 row created.

insert into company values(1,1002,'B Inc.','Long Name B Inc.');

1 row created.

 insert into company values(1,1003,'C Inc.','Long Name C Inc.');

1 row created.

 insert into company values(2,1004,'D Inc.','Long Name D Inc.');

1 row created.
insert into company values(2,1005,'E Inc.','Long Name E Inc.');

1 row created.

 insert into company values(2,1006,'F Inc.','Long Name F Inc.');

1 row created.


 create table product_audit(
  2     product_id number(4) not null,
  3     num_rows number(8) not null
  4  );
```

```
Table created.


 CREATE OR REPLACE TRIGGER myTrigger
     AFTER INSERT ON company
     FOR EACH ROW
     BEGIN
       UPDATE product_audit
       SET num_rows =num_rows+1
       WHERE product_id =:NEW.product_id;
       IF (SQL%NOTFOUND) THEN
         INSERT INTO product_audit VALUES (:NEW.product_id,1);
       END IF;
     END;
     /

Trigger created.

 drop table product_audit;

Table dropped.

 drop table company;

Table dropped.
```