Trigger

SQL Triggers

- To monitor a database and take a corrective action when a condition occurs

- A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, or DELETE statement) is performed on a specified table.

  - Examples:
    - Charge $10 overdraft fee if the balance of an account after a withdrawal transaction is less than $500
    - Limit the salary increase of an employee to no more than 5% raise

```
CREATE TRIGGER trigger-name
 trigger-time (Before | After)
 trigger-event (Insert | Update | Delete)
 ON table-name
 FOR EACH ROW
 trigger-action;
```

Uses for Triggers

- Enforce business rules
- Validate input data
- Generate a unique value for a newly-inserted row in a different file.
- Write to other files for audit trail purposes
- Query from other files for cross-referencing purposes
- Access system functions
- Replicate data to different files to achieve data consistency

Firing of Trigger: Before Insert

- create table student (sname varchar(20), marks int);
- insert into student values ("A", 70),("B", 50),("C", 80),("D", 90);
- select * from student;

	sname	marks
▶	A	70
	B	50
	C	80
	D	90

- A trigger will fire when a new student get admission and his/her marks will be increased by 2 just before inclusion in the table.

```
create TRIGGER calculate
before INSERT on student
FOR EACH ROW
set new.marks=new.marks+2;
```

```
insert into student values ("F", 40);
select * from student;
```

	sname	marks
▶	A	70
	B	50
	C	80
	D	90
	F	42

Marks of F is 40, but inserted as 42.

Firing of Trigger: After Insert

- create table Final_mark(total_marks int);

- A trigger will fire after inclusion of a new student and marks will be inserted in a new table.

- create TRIGGER cal
after INSERT on student
FOR EACH ROW
insert into Final_mark values(new.marks);

- insert into student values ("G", 45);

select * from student;

sname	marks
A	70
B	50
C	80
D	90
F	42
G	47

select *from Final_mark;

total_marks
45

Firing of Trigger: After Delete

- Create a replica table of student. Now a trigger will fire when a student will be deleted and all the marks in replica table will be increased by 2.

- create table student_replica
(sname varchar(20), marks int);
insert into student_replica values
("A", 70),("B", 50),("C", 80),("D", 90),("F", 42),("G", 47);

- CREATE TRIGGER student_delete
AFTER DELETE ON student
FOR EACH ROW UPDATE student_replica
SET student_replica.marks=student_replica.marks+2;
- delete from student where marks=70;

- A same type of trigger can be written for before delete.

select * from student;

	sname	marks
▶	B	50
	C	80
	D	90
	F	42
	G	47

select * from student_replica;

	sname	marks
▶	A	72
	B	52
	C	82
	D	92
	F	44
	G	49

Firing of Trigger: Before Update

- A trigger will fire before the value of any student getting updated and all the marks in replica table will be decreased by 10.

- CREATE TRIGGER student_update

BEFORE UPDATE ON student

FOR EACH ROW UPDATE student_replica SET student_replica.marks=student_replica.marks-10;

- UPDATE student set student.sname="NEW"

where marks=90;

- A same type of trigger can be written

after update.

select * from student;

	sname	marks
▶	B	50
	C	80
	NEW	90
	F	42
	G	47

select * from student_replica;

	sname	marks
▶	A	62
	B	42
	C	72
	D	82
	F	34
	G	39

Thankyou