**CREATING THE DATABASE AND TABLE:-**

create database assignment;

use assignment;

create table employee(

emp\_id int primary key,

emp\_name varchar(20),

job\_name varchar(20),

salary int

);

insert into employee values(1,"Yash","Web Developer",40000);

insert into employee values(2,"Ash","Accountant",30000);

insert into employee values(3,"Rahul","Accountant",30000);

insert into employee values(4,"Pragya","Web Developer",40000);

insert into employee values(5,"Abhigyan","Web Developer",40000);

insert into employee values(6,"Mona","Testor",20000);

insert into employee values(7,"Lola","Web Developer",40000);

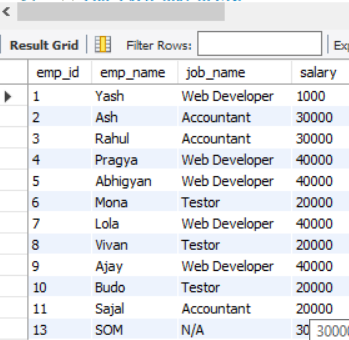
insert into employee values(8,"Vivan","Testor",20000);

insert into employee values(9,"Ajay","Web Developer",40000);

insert into employee values(10,"Budo","Testor",20000);

insert into employee values(11,"Sajal","Accountant",20000);

**TABLE:**



**CREATING STORED PROCEDURE AND USING IT TO PAGINATE:**

**STORED PROCEDURE PAGINATE:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `paginate`(in page\_no int,in rows\_per\_page int)

BEGIN

declare skip\_rows int default 0;

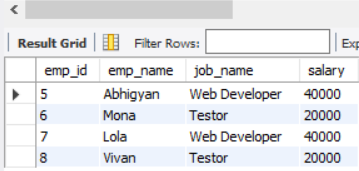
set skip\_rows=(page\_no-1)\*rows\_per\_page;

select \* from employee limit skip\_rows,rows\_per\_page;

END

**USING STORED PROCUDRE TO PAGINATE:**

call paginate(2,4);



**CREATING INSTEAD OF INSERT TRIGGER TO MAKE REPLACE NULL WITH N/A IN JOB NAME:**

DELIMITER $$

CREATE TRIGGER employeeInsertTrigger BEFORE INSERT ON employee

FOR EACH ROW BEGIN

IF (NEW.job\_name IS NULL) THEN

SET NEW.job\_name="N/A";

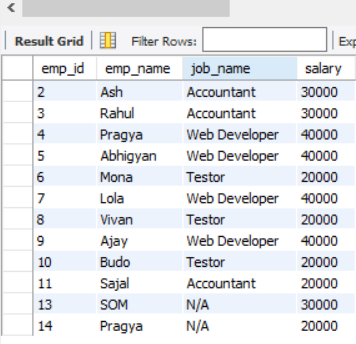
END IF;

END$$

DELIMITER ;

**USING INSTEAD OF INSERT TRIGGER:**

insert into employee values(14,”Pragya”,NULL,20000);



**CREATING INSTEAD OF DELETE TRIGGER TO CREATE AN AUDIT TABLE:**

**CREATING AUDIT TABLE:**

create table employee\_audit(

employee\_id int,

deleted\_date date,

deleted\_by varchar(50)

);

**CREATING INSTEAD OF DELETE TRIGGER:**

CREATE TRIGGER employeeDeleteTrigger BEFORE DELETE ON employee

FOR EACH ROW BEGIN

DECLARE vUser varchar(50);

SELECT USER() INTO vUser;

INSERT INTO employee\_audit

( employee\_id,

deleted\_date,

deleted\_by)

VALUES

( OLD.emp\_id,

SYSDATE(),

vUser );

END$$

DELIMITER ;

**USING INSTEAD OF DELETE TRIGGER:**

delete from employee where emp\_id=14;

select \* from employee\_audit;



**CREATING INSTEAD OF UPDATE TRIGGER TO CREATE AN UPDATE AUDIT TABLE:**

**CREATING UPDATE AUDIT TABLE:**

create table update\_log(

employee\_id int,

updated\_date date,

updated\_by varchar(50)

);

**CREATING INSTEAD OF UPDATE TRIGGER:**

DELIMITER $$

CREATE TRIGGER employeeUpdateTrigger BEFORE UPDATE ON employee

FOR EACH ROW BEGIN

DECLARE vUser varchar(50);

SELECT User() INTO vUser;

INSERT INTO update\_log

(employee\_id,

updated\_date,

updated\_by)

VALUES

(OLD.emp\_id,

SYSDATE(),

vUser);

END$$

DELIMITER ;

**USING INSTEAD OF UPDATE TRIGGER:**

update employee set emp\_name="Kshitij" where emp\_id=11;

select \* from update\_log;

