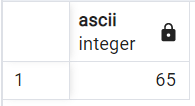
**BUIT IN FUNCTIONS IN SQL:**

* String Functions
* Math/ Numeric Functions

**String BUILT-IN FUNCTIONS:**

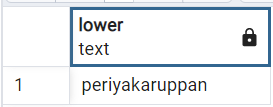
SELECT ASCII(‘A’);



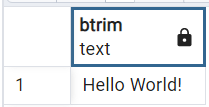
SELECT CONCAT('Hello','World!');



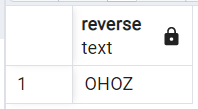
SELECT LOWER('PERIYAKARUPPAN');



SELECT TRIM(' Hello World! ');

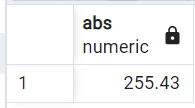


SELECT REVERSE('ZOHO');

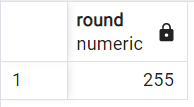


**MATH/NUMERIC BUILT-IN FUNCTIONS:**

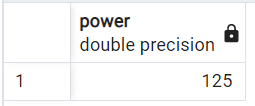
SELECT ABS(-255.43);



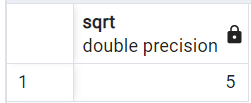
SELECT ROUND(255.43);



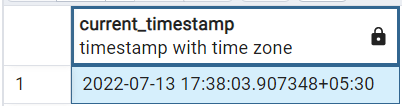
SELECT POWER(5,3);



SELECT SQRT(25);



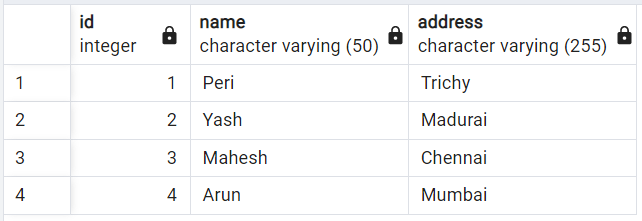
SELECT CURRENT\_TIMESTAMP;



**TRIGGERS:**

I have created a sample table **records** consisting of id, name and address.

I have used a **trigger** function that gets triggered once I update the address in records table.



Trigger: Trigger is a function that executes automatically if it satisfies a particular function. Then it executes the user defined function.

**Code of Creating a Procedure/Function:**

create or replace function **add\_func()**

returns trigger as

$BODY$

begin

if NEW.address <> OLD.address then

insert into address\_log values (old.id,old.name,old.address,new.address);

end if;

return new;

end;

$BODY$

language plpgsql;

**Code for Creating a trigger and mapping it to execute the required function:**

create trigger trigger\_func

before update

on records

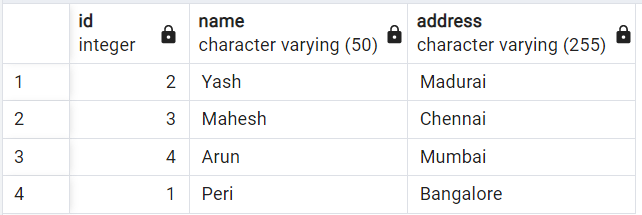
for each row

execute procedure add\_func();

So if I update address of a person in row, old and new address gets stored into new row of the address\_log table.

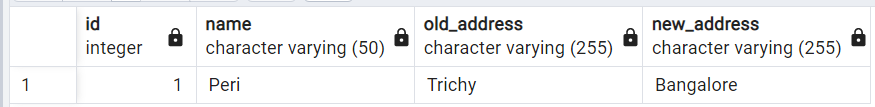
update records set address='Bangalore' where id=1;

select \* from records;





Select \* from address\_log;



**TRANSACTIONS:**

Transactions in SQL follow ACID properties.

A – Atomicity

C – Consistency

I – Isolation

D – Durability

The most important commands in a transaction are

* COMMIT
* ROLLBACK
* SAVEPOINT
* SET TRANSACTION

**COMMIT:**

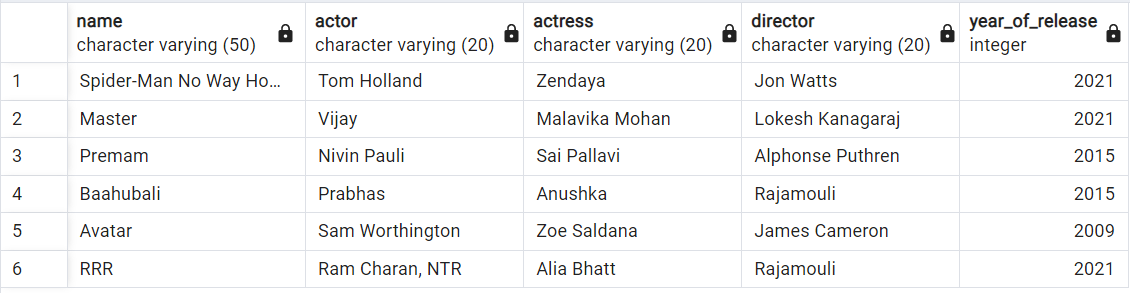
I will start the transaction by

BEGIN;

BEGIN Query returned successfully in 54 msec.

I have created a movie DB with sample data.

Select \* from movies;



I’m Performing a delete operation and saving it with the COMMIT;

delete from movies where name = 'RRR';

COMMIT;

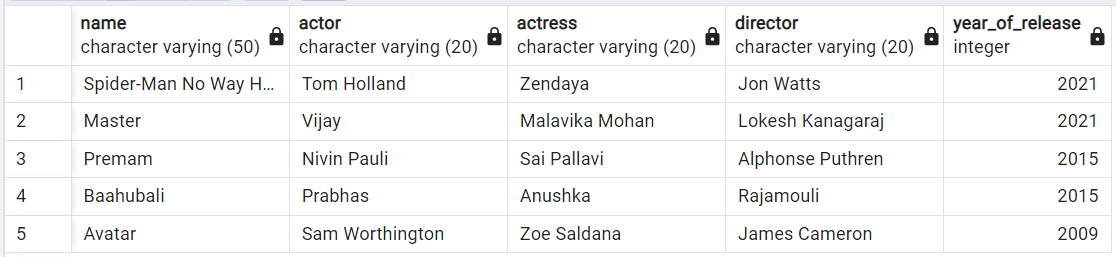
Output

COMMIT

Query returned successfully in 698 msec.

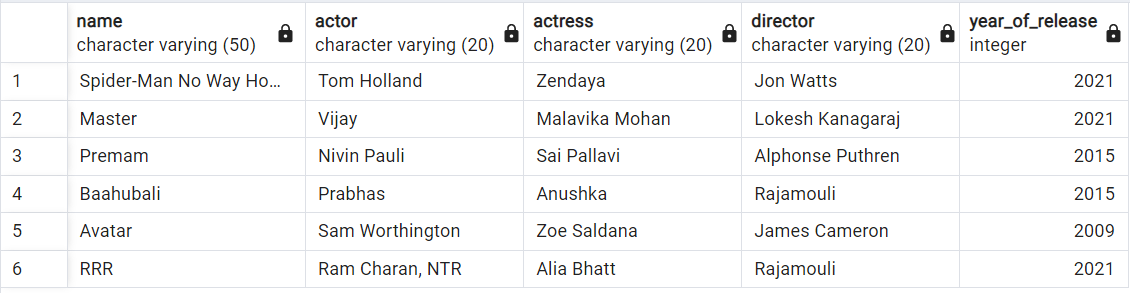
The resultant deleted table after commit is:

Select \* from movies;



Now I want to rollback to previous savepoint that is the begin point so the delete operation will be un done.

**ROLLBACK;**



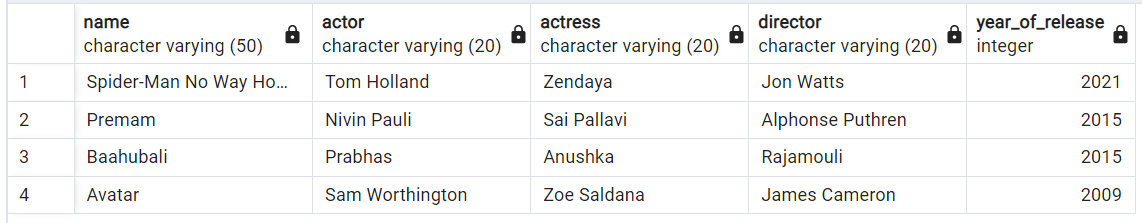
**SAVEPOINT:**

BEGIN SAVEPOINT S1;

SAVEPOINT Query returned successfully in 52 msec.

DELETE FROM movies where name = 'Master';

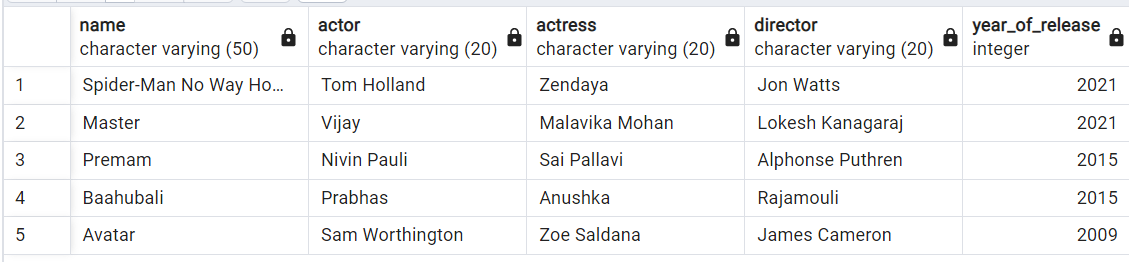
select \* from movies;



ROLLBACK TO S1;

ROLLBACK Query returned successfully in 66 msec.

SELECT \* From movies;



MASTER Movie deletion is UNDONE my ROLLBACK;

**SET TRANSACTION:**

It will set read / read write mode for a transaction

SET TRANSACTION READ ONLY;

DELETE FROM movies where name = 'Master';

ERROR: cannot execute DELETE in a read-only transaction SQL state: 25006