SQL Task for 31st JULY

--Create a loop for a table to insert a record into a tale for two columns in first coulmn you have to inset a data ranging from 1 to 100 and in second column you have to inset a square of the first column --

```
create table square_table1(var int, var1 int)
delimiter $$
create procedure insert_data_square()
begin
set @var= 1;
generate_data: loop
if @var <=100 then
set @var= @var;
set @var1 = @var * @var;
insert into square_table1 values (@var,@var1);
set @var = @var +1;
if @var > 100 then
leave generate_data;
end if;
end if;
end loop generate_data;
end $$
drop procedure insert_data_square;
call insert_data_square()
select * from square_table1;
```

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

---create a user defined function to find out a date differences in number of days// no in built fun:--

```
select * from sales1;
delimiter $$
CREATE FUNCTION weekdays (DATE1 DATETIME, DATE2 DATETIME)
   RETURNS DECIMAL(20,0)
   BEGIN
      RETURN (DATE1 - DATE2)/1000000;
      END $$
```

```
drop function weekdays
SELECT CURRENT_TIMESTAMP;
select weekdays('2022-08-16 14:30:18','2022-08-12 03:50:18');
SELECT weekdays('2010-01-16', '2010-01-01');

--create a UDF to find out a log base 10 of any given number :

SELECT LOG10(4.5);
drop function log_of_number;
delimiter &&
create function log_of_number( a int)
returns decimal(20,6)
deterministic
begin
return log10(a);
end &&

select log_of_number(4.5)

---create a UDF which will be able to check a total number of records avaible
in your table--
SELECT count( * ) as total_record FROM sales1;
select * from sales1;
delimiter $$
create function final_count_sales ()
returns int
deterministic
begin
return(select count(*) from sales1);
end $$

drop function final_count_sales;
select final_count_sales();

--create a procedure to find out  5th highest profit in your sales table you
dont have to use rank and windowing function --
select * from sales1;

select profit from sales1 order by profit desc limit 5;
select profit from (select profit from sales1 order by profit desc limit 5) AS P
```

```
order by profit asc limit 1;

delimiter $$
create procedure testing()
begin
select profit from (select profit from sales1 order by profit desc limit 5) AS P
order by profit asc limit 1;
end $$

call testing();
```