A

Project Report

on

Inventory Management System (Electronic)

Developed by

YASH KANJIYA (IT-054) – Department of IT, DDU

KULDIP KARANGIYA (IT-055) - Department of IT, DDU

MIHIR KALARIYA (IT -051) - Department of IT, DDU

Guided By

:InternalGuide:

Prof.Archana N.Vyas

Department of Information Technology
Faculty of Technology

DD University

Department of Information Technology Faculty of
Technology, Dharmsinh Desai University
College Road, Nadiad-387001

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Inventory Management System**" is bonafied report of the work carried out by

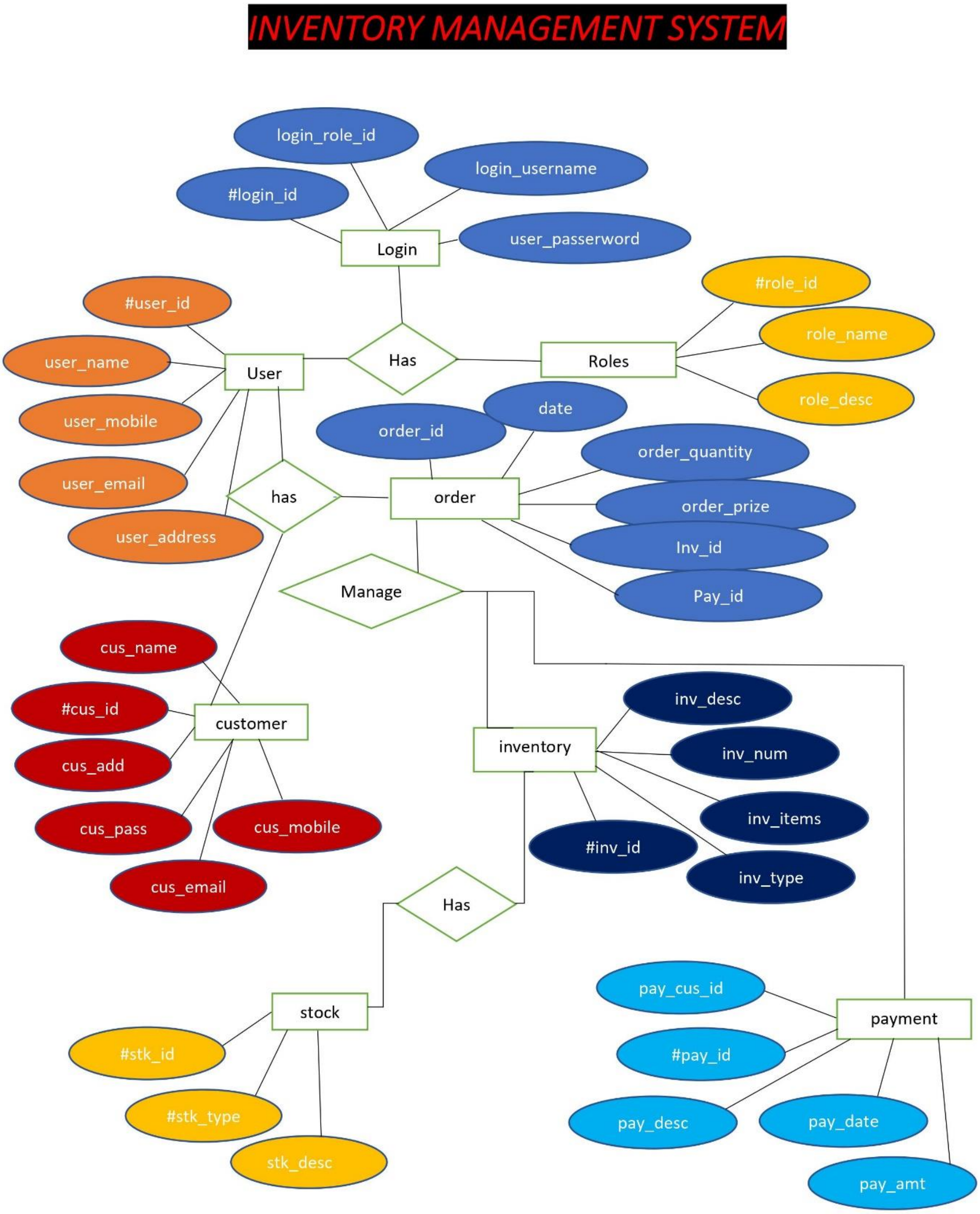
1) **Mr.YASH KANJIYA ,** Student ID No : **19ITUOS057**
2) **Mr. KULDIP KARANGIYA** , Student ID No : **19ITUBS068**
3) **Mr.MIHIR KALARIYA.** Student ID No : **19ITUOS124**

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2021-2022.

Prof. Archana N. Vyas
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad

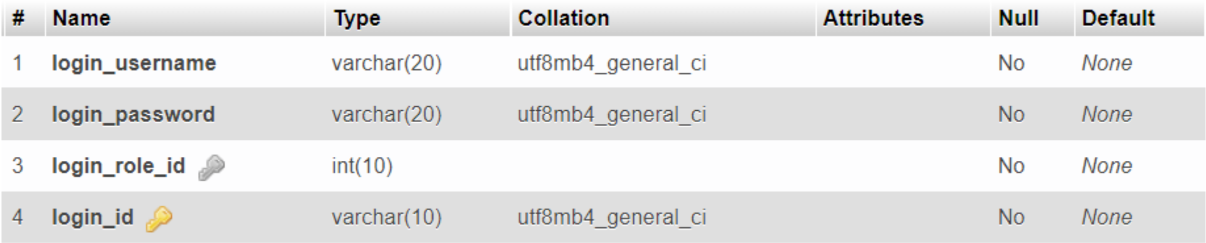
Prof. Vipul Dabhi
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad

ER DIAGRAM


INVENTORY MANAGEMENT SYSTEM
login_role_id
login_username
#login_id
user_passerword
Login
#role_id
role_name
role_desc
Roles
Has
#user_id
user_name
user_mobile
user_email
user_address
User
has
order
date
order_id
order_quantity
order_prize
Inv_id
Pay_id
Manage
cus_name
#cus_id
cus_add
cus_pass
cus_mobile
cus_email
customer
inventory
inv_desc
inv_num
inv_items
#inv_id
inv_type
Has
stock
#stk_id
#stk_type
stk_desc
payment
pay_cus_id
#pay_id
pay_desc
pay_date
pay_amt

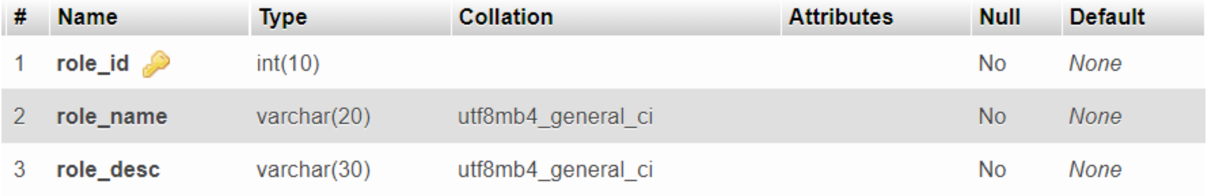
DATABASE TABLES

1) Login:-



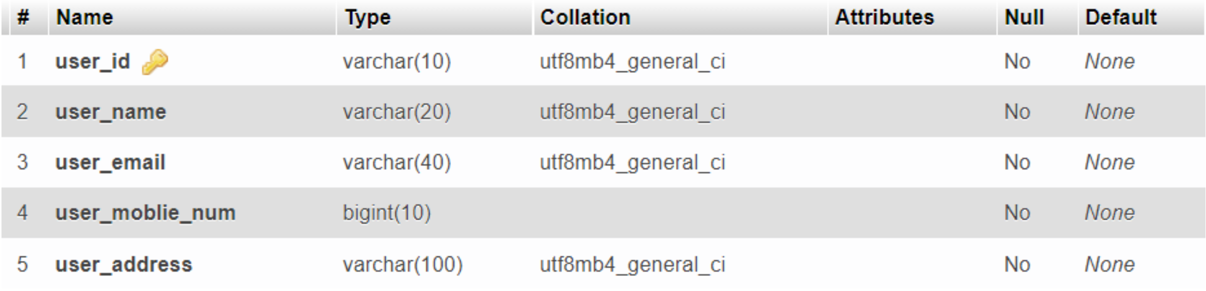
#	Name	Type	Collation	Attributes	Null	Default
1	**login_username**	varchar(20)	utf8mb4_general_ci		No	*None*
2	**login_password**	varchar(20)	utf8mb4_general_ci		No	*None*
3	**login_role_id** 🔑	int(10)			No	*None*
4	**login_id** 🔑	varchar(10)	utf8mb4_general_ci		No	*None*

2) Roles:-



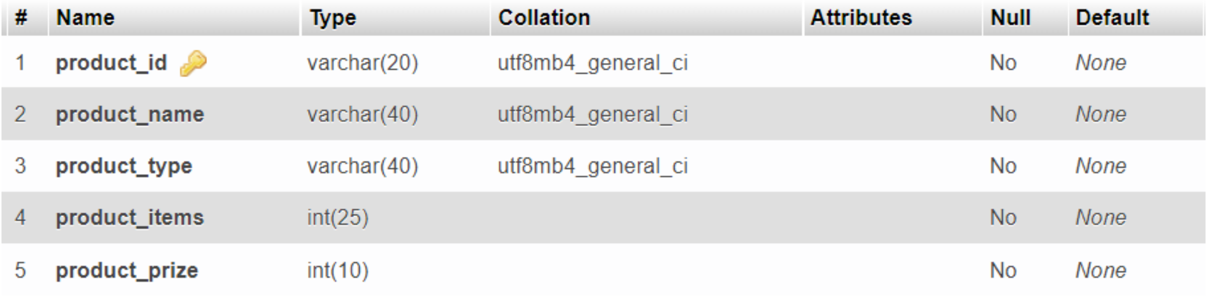
#	Name	Type	Collation	Attributes	Null	Default
1	**role_id** 🔑	int(10)			No	*None*
2	**role_name**	varchar(20)	utf8mb4_general_ci		No	*None*
3	**role_desc**	varchar(30)	utf8mb4_general_ci		No	*None*

3) Username:-



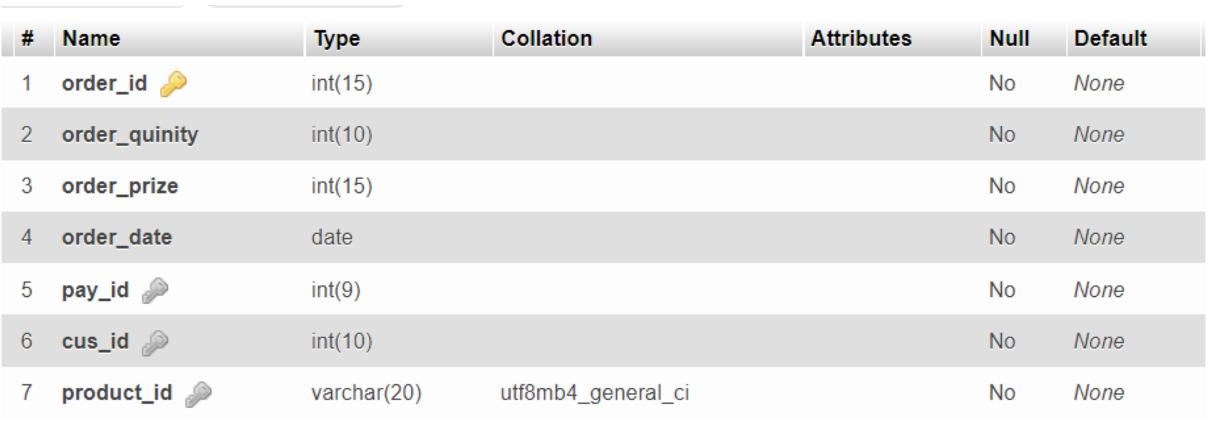
#	Name	Type	Collation	Attributes	Null	Default
1	**user_id** 🔑	varchar(10)	utf8mb4_general_ci		No	*None*
2	**user_name**	varchar(20)	utf8mb4_general_ci		No	*None*
3	**user_email**	varchar(40)	utf8mb4_general_ci		No	*None*
4	**user_moblie_num**	bigint(10)			No	*None*
5	**user_address**	varchar(100)	utf8mb4_general_ci		No	*None*

4) Product:-



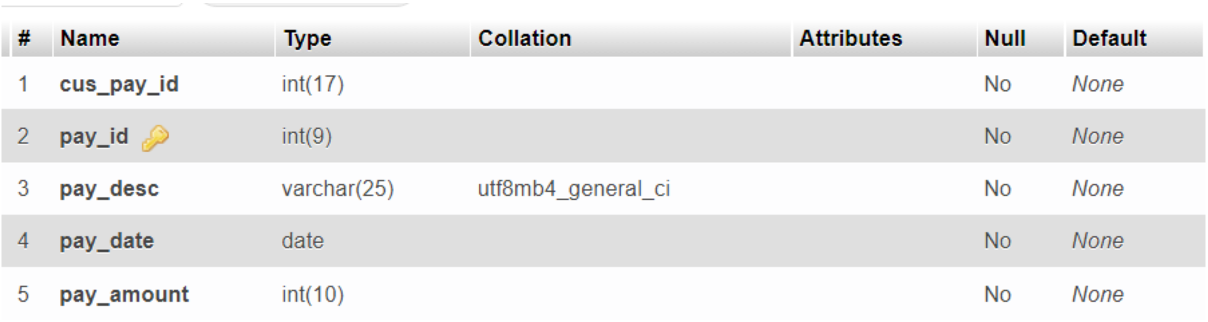
#	Name	Type	Collation	Attributes	Null	Default
1	**product_id** 🔑	varchar(20)	utf8mb4_general_ci		No	*None*
2	**product_name**	varchar(40)	utf8mb4_general_ci		No	*None*
3	**product_type**	varchar(40)	utf8mb4_general_ci		No	*None*
4	**product_items**	int(25)			No	*None*
5	**product_prize**	int(10)			No	*None*

5) Order_d:-



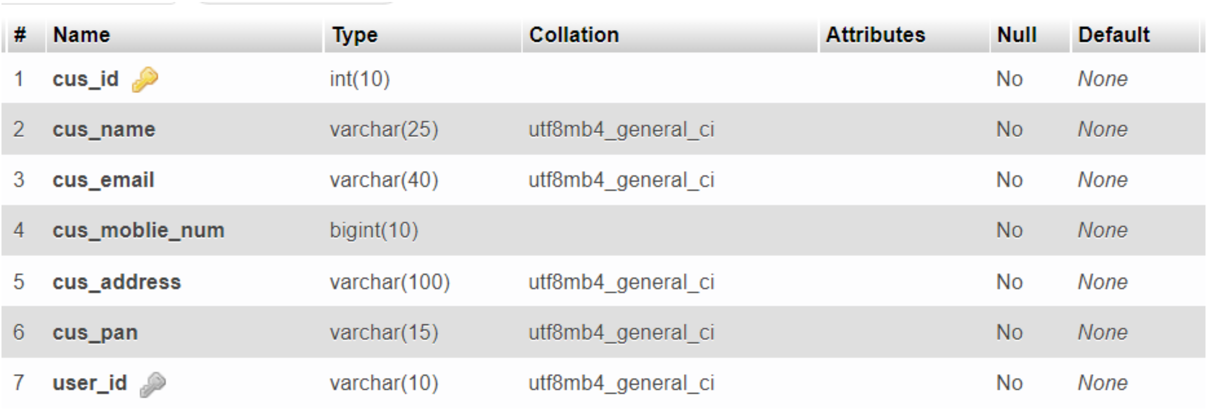
#	Name	Type	Collation	Attributes	Null	Default
1	**order_id** 🔑	int(15)			No	*None*
2	**order_quinity**	int(10)			No	*None*
3	**order_prize**	int(15)			No	*None*
4	**order_date**	date			No	*None*
5	**pay_id** 🔑	int(9)			No	*None*
6	**cus_id** 🔑	int(10)			No	*None*
7	**product_id** 🔑	varchar(20)	utf8mb4_general_ci		No	*None*

6) Pay:-



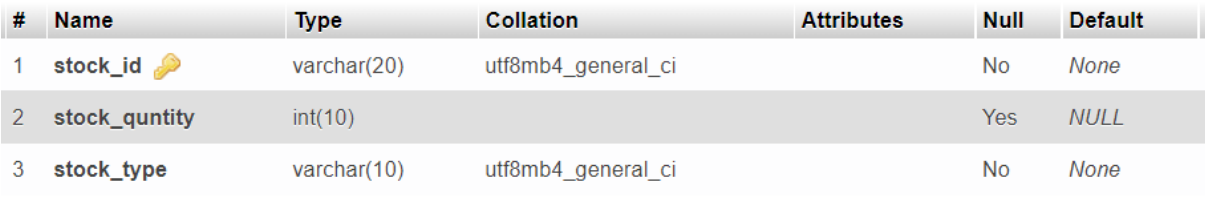
#	Name	Type	Collation	Attributes	Null	Default
1	**cus_pay_id**	int(17)			No	*None*
2	**pay_id** 🔑	int(9)			No	*None*
3	**pay_desc**	varchar(25)	utf8mb4_general_ci		No	*None*
4	**pay_date**	date			No	*None*
5	**pay_amount**	int(10)			No	*None*

7) Customer:-



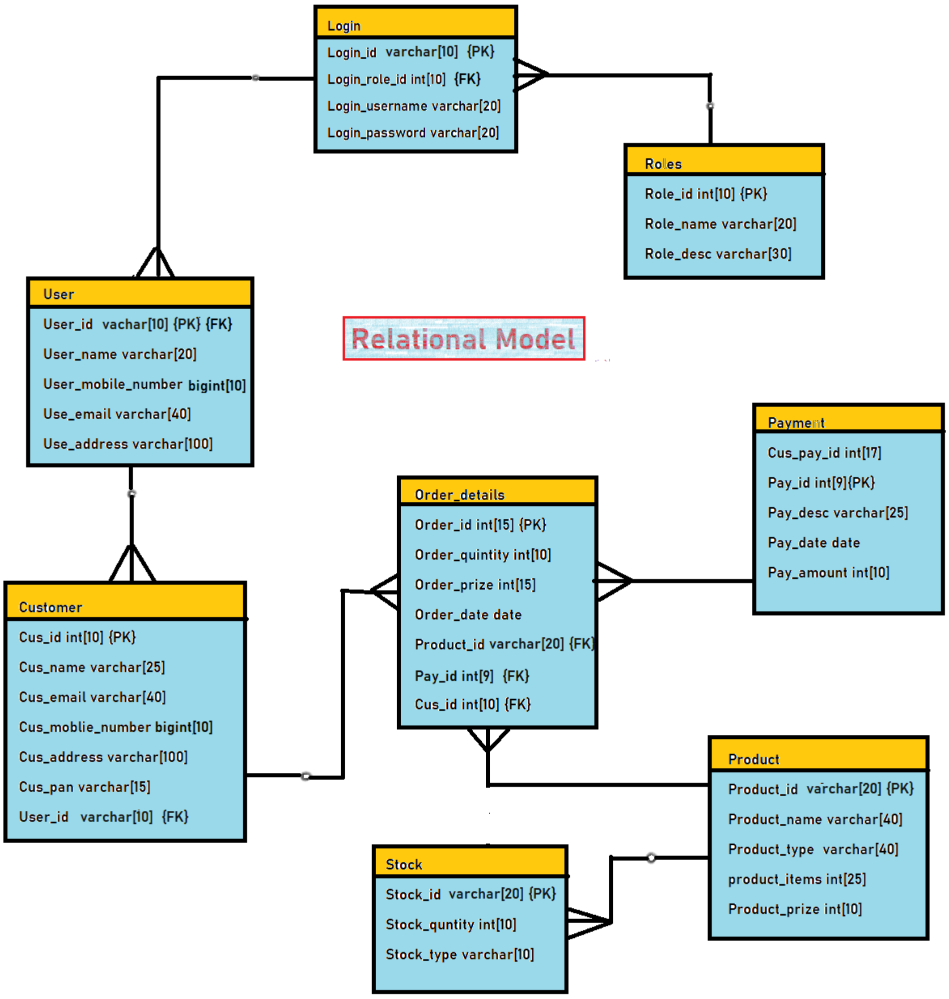
#	Name	Type	Collation	Attributes	Null	Default
1	**cus_id** 🔑	int(10)			No	*None*
2	**cus_name**	varchar(25)	utf8mb4_general_ci		No	*None*
3	**cus_email**	varchar(40)	utf8mb4_general_ci		No	*None*
4	**cus_moblie_num**	bigint(10)			No	*None*
5	**cus_address**	varchar(100)	utf8mb4_general_ci		No	*None*
6	**cus_pan**	varchar(15)	utf8mb4_general_ci		No	*None*
7	**user_id** 🔑	varchar(10)	utf8mb4_general_ci		No	*None*

8) Stock:-



#	Name	Type	Collation	Attributes	Null	Default
1	**stock_id** 🔑	varchar(20)	utf8mb4_general_ci		No	*None*
2	**stock_quntity**	int(10)			Yes	*NULL*
3	**stock_type**	varchar(10)	utf8mb4_general_ci		No	*None*

SCHEMA DIAGRAM



DATABASE IMPLEMENTATION

1) Login:-

```
CREATE TABLE public.login
(
    login_username character varying(20) COLLATE pg_catalog."default" NOT NULL,
    login_password character varying(20) COLLATE pg_catalog."default" NOT NULL,
    login_role_id integer[] NOT NULL,
    login_id character varying(10)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT login_pkey PRIMARY KEY (login_id),
    CONSTRAINT login_login_role_id_fkey FOREIGN KEY (login_role_id)
        REFERENCES public.roles (role_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

2) Roles:-

```
CREATE TABLE public.roles
(
    role_id integer[] NOT NULL,
    role_name character varying[] COLLATE pg_catalog."default" NOT NULL,
    role_desc character varying[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT roles_pkey PRIMARY KEY (role_id)
)
```

3) Username:-

```
CREATE TABLE public.username
(
    user_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
    user_email character varying(40)[] COLLATE pg_catalog."default" NOT NULL,
    user_moblie_num Bigint[] NOT NULL,
    user_address character varying(100)[] COLLATE pg_catalog."default" NOT NULL,
    user_id character varying(10)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT username_pkey PRIMARY KEY (user_id),
    CONSTRAINT username_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public.login (login_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

4) Product:-

```
CREATE TABLE public.product
(
    product_name character varying(40)[] COLLATE pg_catalog."default" NOT NULL,
    product_type character varying(40)[] COLLATE pg_catalog."default" NOT NULL,
    product_items integer[] NOT NULL,
    product_prize integer[] NOT NULL,
    product_id character varying(20)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT product_pkey PRIMARY KEY (product_id)
)
```

5) Order_d:-

```
CREATE TABLE public.order_d
(
    order_id integer[] NOT NULL,
    order_quinity integer[] NOT NULL,
    order_prize integer[] NOT NULL,
    order_date date NOT NULL,
    pay_id integer[] NOT NULL,
    cus_id integer[] NOT NULL,
    product_id character varying(20)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT order_d_pkey PRIMARY KEY (order_id),
    CONSTRAINT order_d_cus_id_fkey FOREIGN KEY (cus_id)
        REFERENCES public.customer (cus_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_d_payment_id_fkey FOREIGN KEY (pay_id)
        REFERENCES public.pay (pay_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_d_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (product_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

6) Pay:-

```
CREATE TABLE public.pay
(
    cus_pay_id integer[] NOT NULL,
    pay_id integer[] NOT NULL,
    pay_desc character varying[] COLLATE pg_catalog."default" NOT NULL,
    pay_date date NOT NULL,
    pay_amount integer[] NOT NULL,
    CONSTRAINT pay_pkey PRIMARY KEY (pay_id)
)
```

7) Customer:-

```
CREATE TABLE public.customer
(
    cus_id integer[] NOT NULL,
    cus_name character varying(25)[] COLLATE pg_catalog."default" NOT NULL,
    cus_email character varying(40)[] COLLATE pg_catalog."default" NOT NULL,
    cus_moblie_num Bigint[] NOT NULL,
    cus_address character varying(100)[] COLLATE pg_catalog."default" NOT NULL,
    cus_pan character varying(15)[] COLLATE pg_catalog."default" NOT NULL,
    user_id character varying(10)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT customer_pkey PRIMARY KEY (cus_id),
    CONSTRAINT customer_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public.username (user_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

8) Stock:-

```
CREATE TABLE public.stock
(
    stock_quntity integer[],
    stock_type character varying(10)[] COLLATE pg_catalog."default" NOT NULL,
    stock_id character varying(20)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT stock_pkey PRIMARY KEY (stock_id),
    CONSTRAINT stock_stock_id_fkey FOREIGN KEY (stock_id)
        REFERENCES public.product (product_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

INSERT DATA

The INSERT INTO command is used to insert new rows in a table.

Example:-

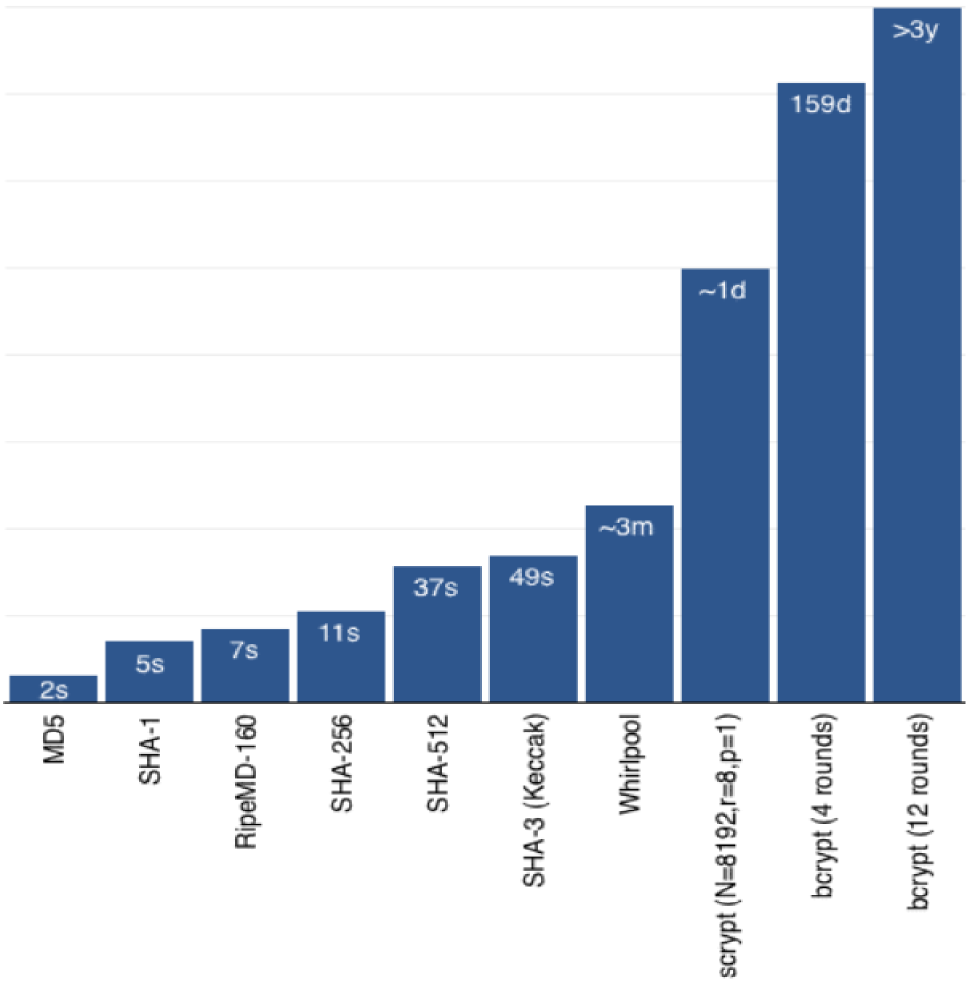
```
INSERT INTO Customers (CustomerName, ContactName, Address, City,
PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen
21', 'Stavanger', '4006', 'Norway');
```

1) Hashing:-

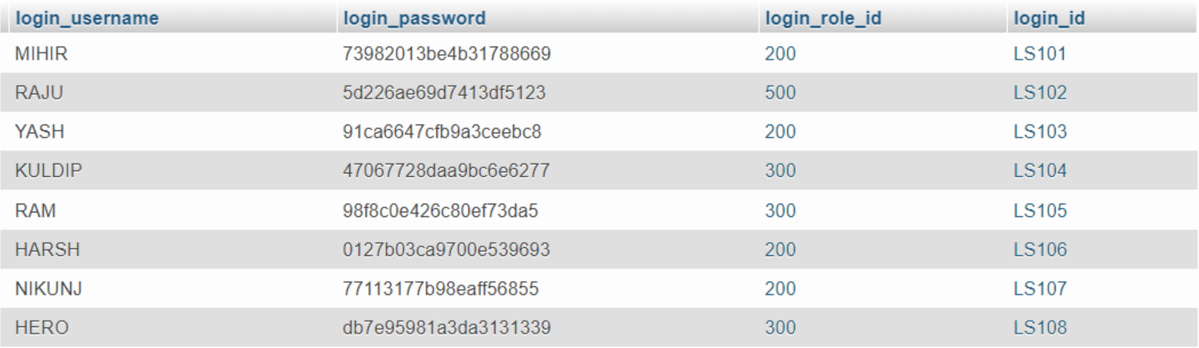
@A hash function is an underlying algorithm that computes the hash value of the supplied data.

@One of the interesting features of a hash function is that it is a one-way algorithm.

MD5(STR):(hashing function):-Convert the data 128bit checksum and returns as a string of 32 hex digit

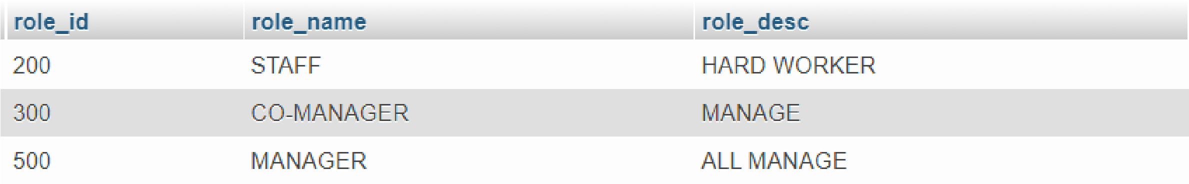


1)Login:-



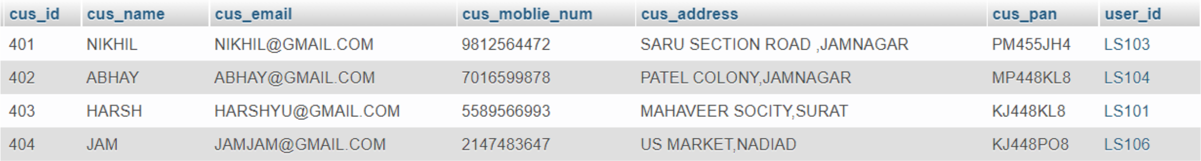
login_username	login_password	login_role_id	login_id
MIHIR	73982013be4b31788669	200	LS101
RAJU	5d226ae69d7413df5123	500	LS102
YASH	91ca6647cfb9a3ceebc8	200	LS103
KULDIP	47067728daa9bc6e6277	300	LS104
RAM	98f8c0e426c80ef73da5	300	LS105
HARSH	0127b03ca9700e539693	200	LS106
NIKUNJ	77113177b98eaff56855	200	LS107
HERO	db7e95981a3da3131339	300	LS108

2) Roles:-



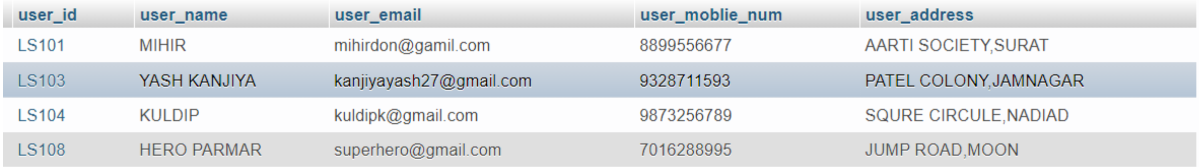
role_id	role_name	role_desc
200	STAFF	HARD WORKER
300	CO-MANAGER	MANAGE
500	MANAGER	ALL MANAGE

3) Customer:-



cus_id	cus_name	cus_email	cus_moblie_num	cus_address	cus_pan	user_id
401	NIKHIL	NIKHIL@GMAIL.COM	9812564472	SARU SECTION ROAD ,JAMNAGAR	PM455JH4	LS103
402	ABHAY	ABHAY@GMAIL.COM	7016599878	PATEL COLONY,JAMNAGAR	MP448KL8	LS104
403	HARSH	HARSHYU@GMAIL.COM	5589566993	MAHAVEER SOCITY,SURAT	KJ448KL8	LS101
404	JAM	JAMJAM@GMAIL.COM	2147483647	US MARKET,NADIAD	KJ448PO8	LS106

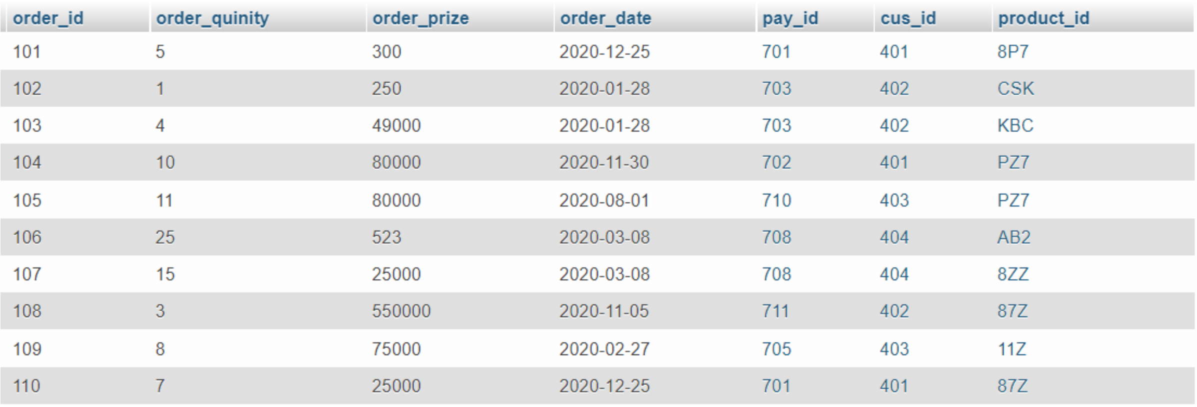
4) Username:-



user_id	user_name	user_email	user_moblie_num	user_address
LS101	MIHIR	mihirdon@gamil.com	8899556677	AARTI SOCIETY,SURAT
LS103	YASH KANJIYA	kanjiyayash27@gmail.com	9328711593	PATEL COLONY,JAMNAGAR
LS104	KULDIP	kuldipk@gmail.com	9873256789	SQURE CIRCULE,NADIAD
LS108	HERO PARMAR	superhero@gmail.com	7016288995	JUMP ROAD,MOON

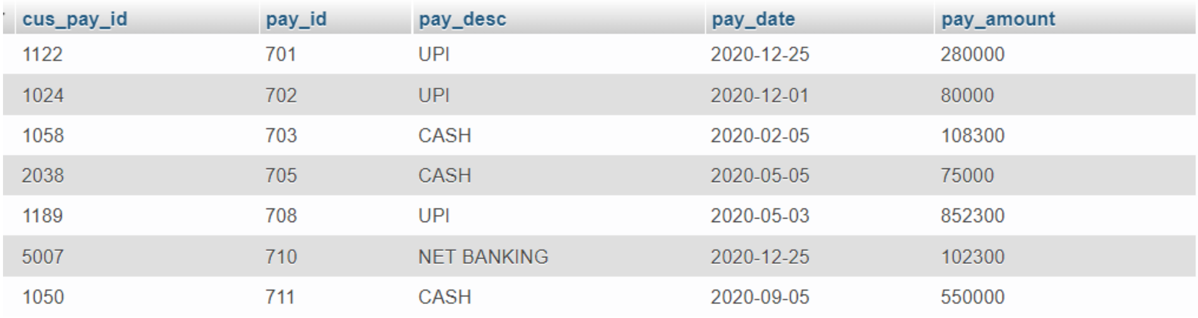
X

5) Order_d:-



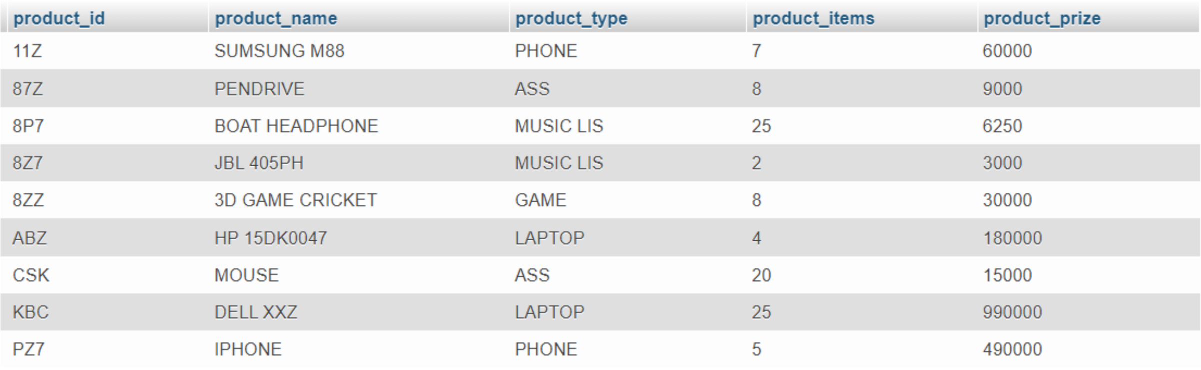
order_id	order_quinity	order_prize	order_date	pay_id	cus_id	product_id
101	5	300	2020-12-25	701	401	8P7
102	1	250	2020-01-28	703	402	CSK
103	4	49000	2020-01-28	703	402	KBC
104	10	80000	2020-11-30	702	401	PZ7
105	11	80000	2020-08-01	710	403	PZ7
106	25	523	2020-03-08	708	404	AB2
107	15	25000	2020-03-08	708	404	8ZZ
108	3	550000	2020-11-05	711	402	87Z
109	8	75000	2020-02-27	705	403	11Z
110	7	25000	2020-12-25	701	401	87Z

6) Pay:-



cus_pay_id	pay_id	pay_desc	pay_date	pay_amount
1122	701	UPI	2020-12-25	280000
1024	702	UPI	2020-12-01	80000
1058	703	CASH	2020-02-05	108300
2038	705	CASH	2020-05-05	75000
1189	708	UPI	2020-05-03	852300
5007	710	NET BANKING	2020-12-25	102300
1050	711	CASH	2020-09-05	550000

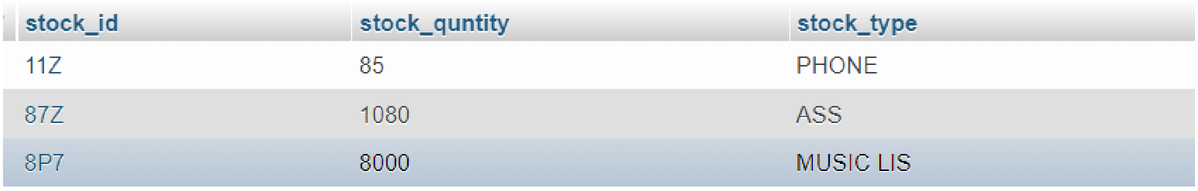
7) Product:-



product_id	product_name	product_type	product_items	product_prize
11Z	SUMSUNG M88	PHONE	7	60000
87Z	PENDRIVE	ASS	8	9000
8P7	BOAT HEADPHONE	MUSIC LIS	25	6250
8Z7	JBL 405PH	MUSIC LIS	2	3000
8ZZ	3D GAME CRICKET	GAME	8	30000
ABZ	HP 15DK0047	LAPTOP	4	180000
CSK	MOUSE	ASS	20	15000
KBC	DELL XXZ	LAPTOP	25	990000
PZ7	IPHONE	PHONE	5	490000

X

8) Stock:-

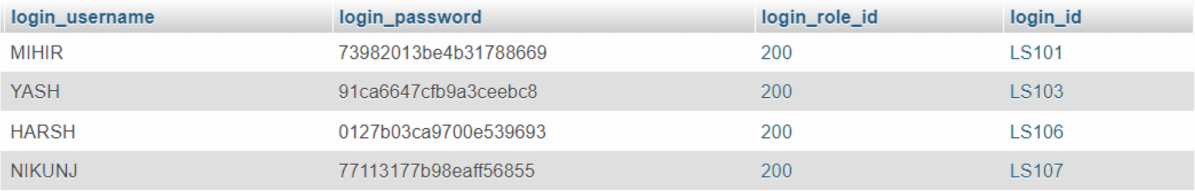


stock_id	stock_quntity	stock_type
11Z	85	PHONE
87Z	1080	ASS
8P7	8000	MUSIC LIS

QUERIES

1) WHERE:-

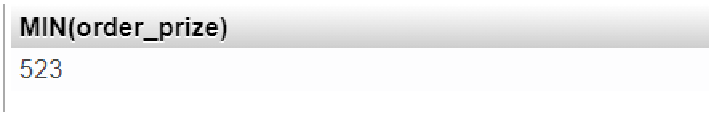
SELECT * FROM `login` WHERE `login_role_id`='200';



login_username	login_password	login_role_id	login_id
MIHIR	73982013be4b31788669	200	LS101
YASH	91ca6647cfb9a3ceebc8	200	LS103
HARSH	0127b03ca9700e539693	200	LS106
NIKUNJ	77113177b98eaff56855	200	LS107

2) Min:-

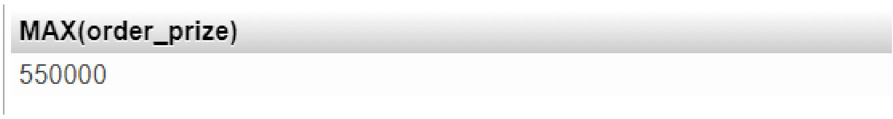
SELECT MIN(order_prize) FROM order_d WHERE `order_id`>105;



MIN(order_prize)
523

3) Max:-

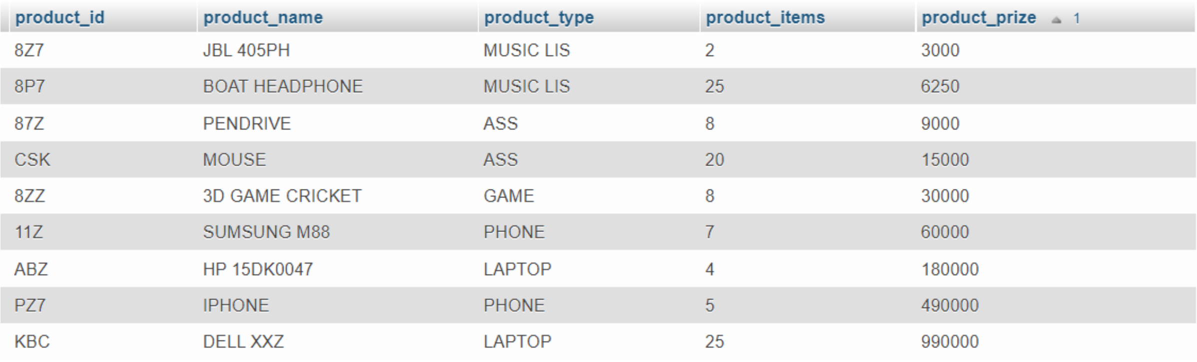
SELECT MAX(order_prize) FROM order_d WHERE `order_id`>105;



MAX(order_prize)
550000

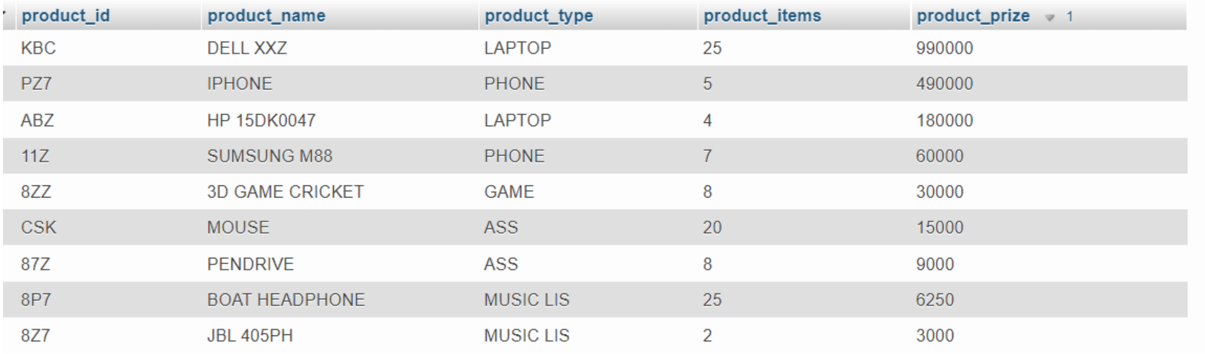
4) ORDER BY:-

SELECT * FROM product ORDER BY product_prize;



product_id	product_name	product_type	product_items	product_prize ▲ 1
8Z7	JBL 405PH	MUSIC LIS	2	3000
8P7	BOAT HEADPHONE	MUSIC LIS	25	6250
87Z	PENDRIVE	ASS	8	9000
CSK	MOUSE	ASS	20	15000
8ZZ	3D GAME CRICKET	GAME	8	30000
11Z	SUMSUNG M88	PHONE	7	60000
ABZ	HP 15DK0047	LAPTOP	4	180000
PZ7	IPHONE	PHONE	5	490000
KBC	DELL XXZ	LAPTOP	25	990000

SELECT * FROM product ORDER BY product_prize DESC;

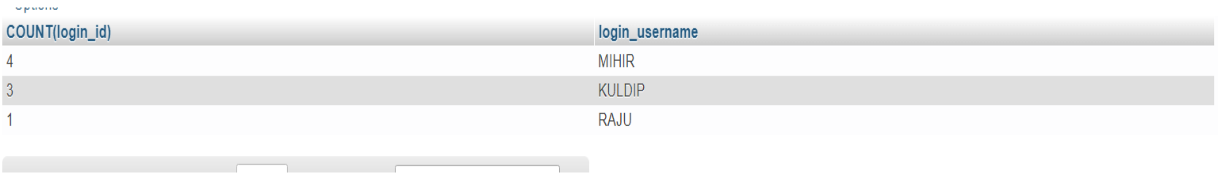


product_id	product_name	product_type	product_items	product_prize ▾ 1
KBC	DELL XXZ	LAPTOP	25	990000
PZ7	IPHONE	PHONE	5	490000
ABZ	HP 15DK0047	LAPTOP	4	180000
11Z	SUMSUNG M88	PHONE	7	60000
8ZZ	3D GAME CRICKET	GAME	8	30000
CSK	MOUSE	ASS	20	15000
87Z	PENDRIVE	ASS	8	9000
8P7	BOAT HEADPHONE	MUSIC LIS	25	6250
8Z7	JBL 405PH	MUSIC LIS	2	3000

5) HAVING:-

SELECT COUNT(login_id), login_username FROM login

GROUP BY login_role_id HAVING COUNT(login_id)<5;

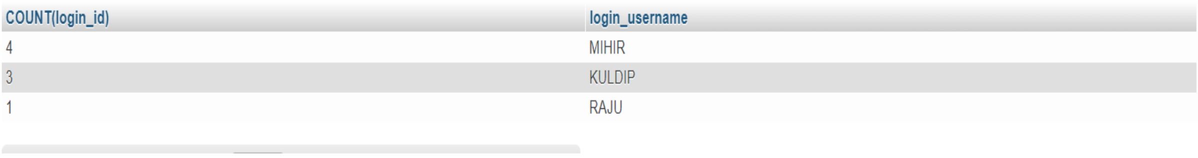


COUNT(login_id)	login_username
4	MIHIR
3	KULDIP
1	RAJU

6) GROUP BY:-

SELECT COUNT(login_id), login_username FROM login

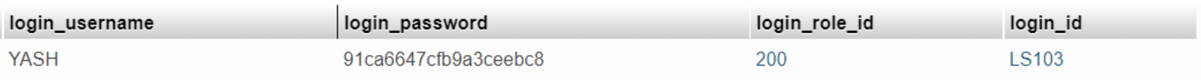
GROUP BY login_role_id



COUNT(login_id)	login_username
4	MIHIR
3	KULDIP
1	RAJU

7) LIKE:-

SELECT * FROM login WHERE login_username LIKE 'y%';



login_username	login_password	login_role_id	login_id
YASH	91ca6647cfb9a3ceebc8	200	LS103

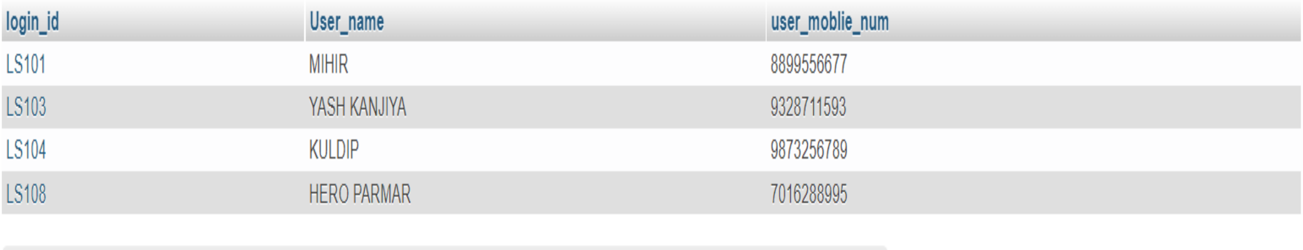
JOINTS AND SUBQUIERS

1) JOINTS:-

(i)INNER JOINTS:-

SELECT login.login_id,username.User_name,username.user_moblie_num

FROM login INNER JOIN username ON login.login_id=username.user_id;

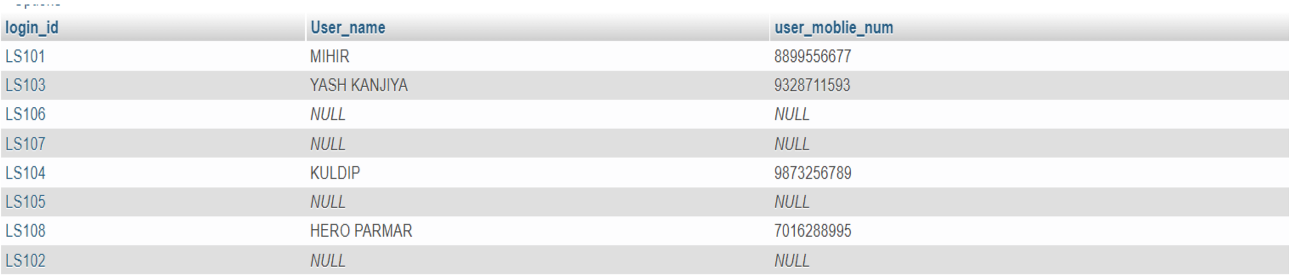


login_id	User_name	user_moblie_num
LS101	MIHIR	8899556677
LS103	YASH KANJIYA	9328711593
LS104	KULDIP	9873256789
LS108	HERO PARMAR	7016288995

(ii)LEFT JOINTS:-

SELECT login.login_id,username.User_name,username.user_moblie_num

FROM login LEFT JOIN username ON login.login_id=username.user_id;

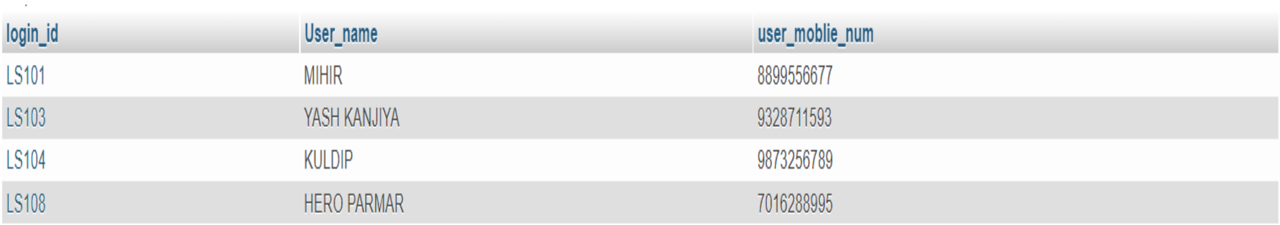


login_id	User_name	user_moblie_num
LS101	MIHIR	8899556677
LS103	YASH KANJIYA	9328711593
LS106	*NULL*	*NULL*
LS107	*NULL*	*NULL*
LS104	KULDIP	9873256789
LS105	*NULL*	*NULL*
LS108	HERO PARMAR	7016288995
LS102	*NULL*	*NULL*

(III)RIGHT JOINTS:-

SELECT login.login_id,username.User_name,username.user_moblie_num

FROM login RIGHT JOIN username ON login.login_id=username.user_id;

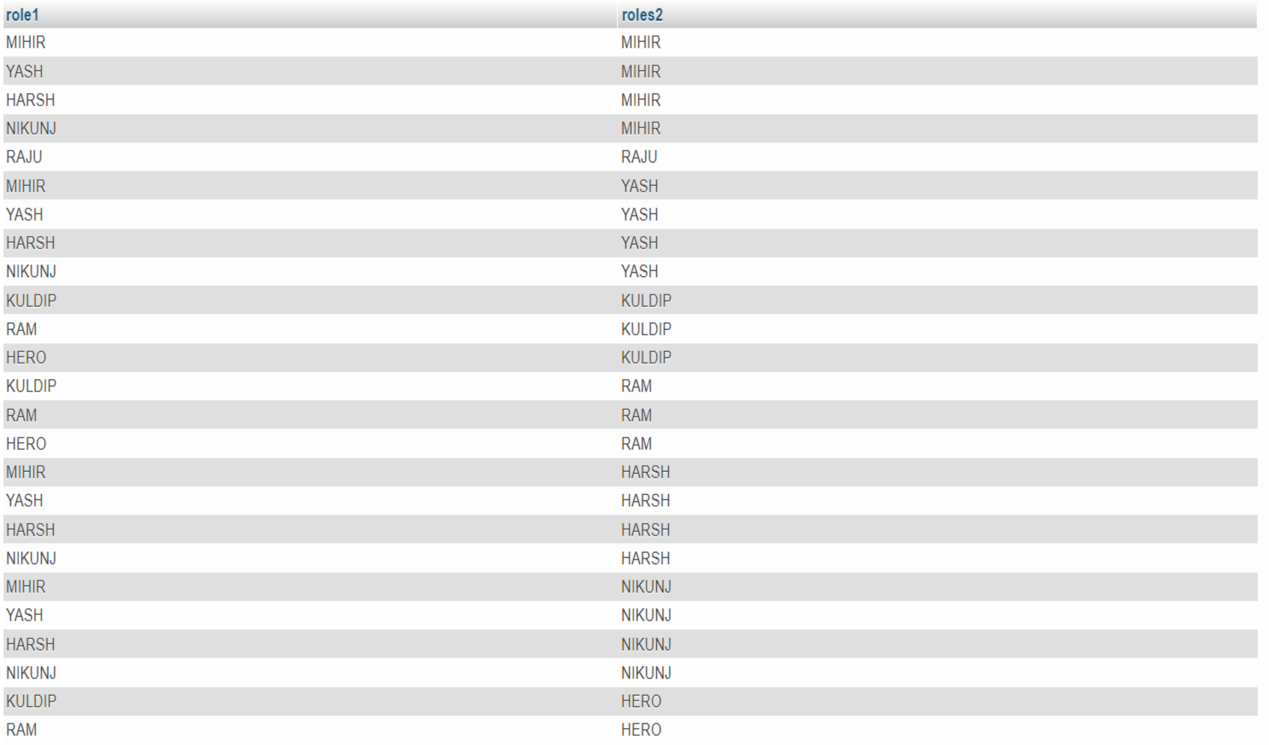


login_id	User_name	user_moblie_num
LS101	MIHIR	8899556677
LS103	YASH KANJIYA	9328711593
LS104	KULDIP	9873256789
LS108	HERO PARMAR	7016288995

(IV)Self JOINTS:-

SELECT A.login_username AS role1,B.login_username AS roles2

FROM login A,login B WHERE A.login_role_id=B.login_role_id



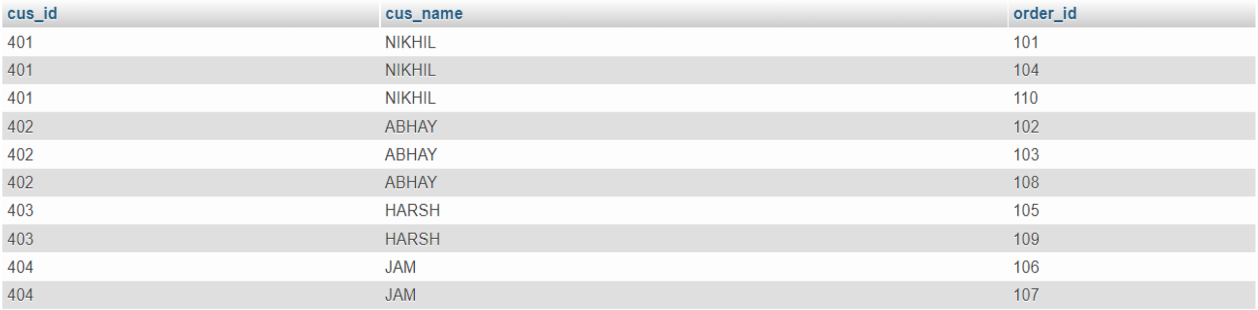
role1	roles2
MIHIR	MIHIR
YASH	MIHIR
HARSH	MIHIR
NIKUNJ	MIHIR
RAJU	RAJU
MIHIR	YASH
YASH	YASH
HARSH	YASH
NIKUNJ	YASH
KULDIP	KULDIP
RAM	KULDIP
HERO	KULDIP
KULDIP	RAM
RAM	RAM
HERO	RAM
MIHIR	HARSH
YASH	HARSH
HARSH	HARSH
NIKUNJ	HARSH
MIHIR	NIKUNJ
YASH	NIKUNJ
HARSH	NIKUNJ
NIKUNJ	NIKUNJ
KULDIP	HERO
RAM	HERO

(V)LEFT OUTER JOINT:-

SELECT customer.cus_id,customer.cus_name,order_d.order_id

FROM customer

LEFT OUTER JOIN order_d ON order_d.cus_id=customer.cus_id



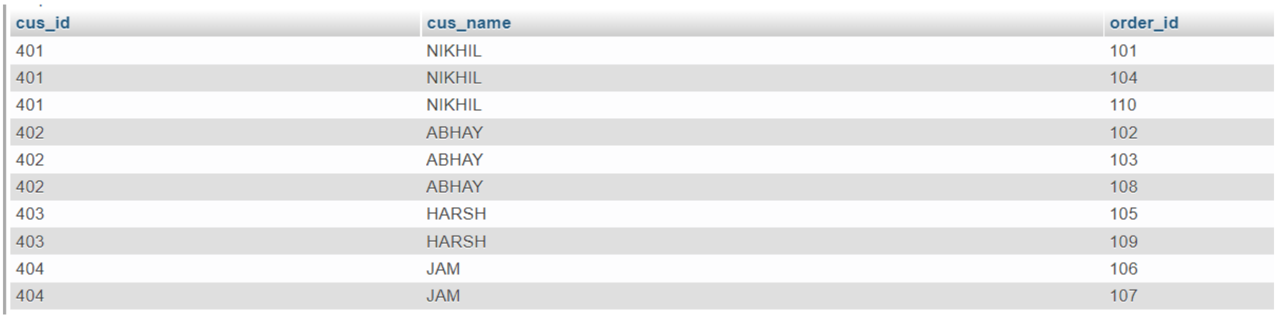
cus_id	cus_name	order_id
401	NIKHIL	101
401	NIKHIL	104
401	NIKHIL	110
402	ABHAY	102
402	ABHAY	103
402	ABHAY	108
403	HARSH	105
403	HARSH	109
404	JAM	106
404	JAM	107

(VI)RIGHT OUTER JOINT:-

SELECT customer.cus_id,customer.cus_name,order_d.order_id

FROM customer

RIGHT OUTER JOIN order_d ON order_d.cus_id=customer.cus_id

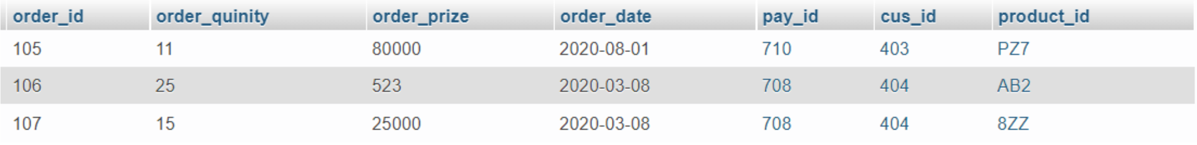


cus_id	cus_name	order_id
401	NIKHIL	101
401	NIKHIL	104
401	NIKHIL	110
402	ABHAY	102
402	ABHAY	103
402	ABHAY	108
403	HARSH	105
403	HARSH	109
404	JAM	106
404	JAM	107

2) SUBQUIERS:-

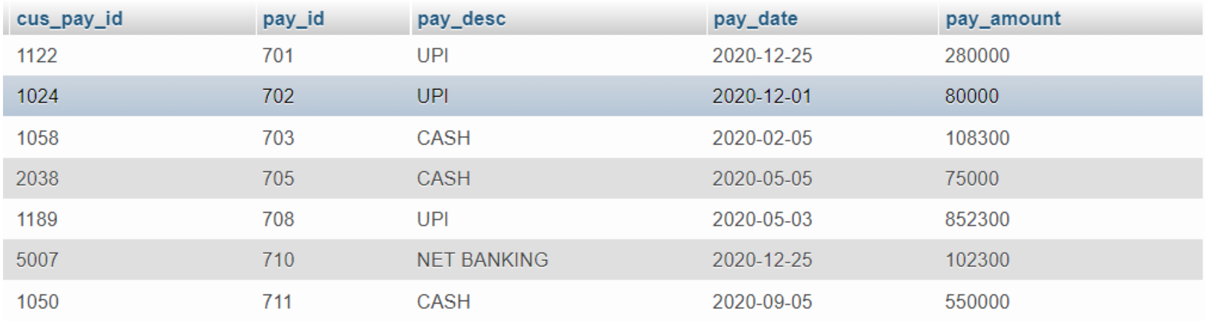
--

(i) SELECT * FROM order_d
WHERE order_d.order_id IN (SELECT order_d.order_id FROM order_d WHERE order_d.order_quinity>10)



order_id	order_quinity	order_prize	order_date	pay_id	cus_id	product_id
105	11	80000	2020-08-01	710	403	PZ7
106	25	523	2020-03-08	708	404	AB2
107	15	25000	2020-03-08	708	404	8ZZ

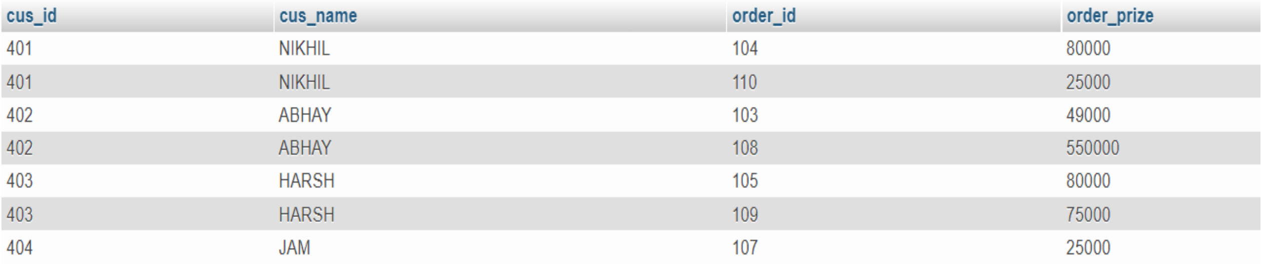
(ii) SELECT * FROM pay
WHERE pay.pay_id IN (SELECT pay.pay_id FROM pay WHERE pay.pay_amount>20000);



cus_pay_id	pay_id	pay_desc	pay_date	pay_amount
1122	701	UPI	2020-12-25	280000
1024	702	UPI	2020-12-01	80000
1058	703	CASH	2020-02-05	108300
2038	705	CASH	2020-05-05	75000
1189	708	UPI	2020-05-03	852300
5007	710	NET BANKING	2020-12-25	102300
1050	711	CASH	2020-09-05	550000

(iii)

```
SELECT customer.cus_id,customer.cus_name,order_d.order_id,order_d.order_prize
FROM  order_d JOIN customer ON customer.cus_id=order_d.cus_id
WHERE order_d.order_id IN (SELECT order_d.order_id FROM order_d WHERE
order_d.order_prize>20000);
```



cus_id	cus_name	order_id	order_prize
401	NIKHIL	104	80000
401	NIKHIL	110	25000
402	ABHAY	103	49000
402	ABHAY	108	550000
403	HARSH	105	80000
403	HARSH	109	75000
404	JAM	107	25000

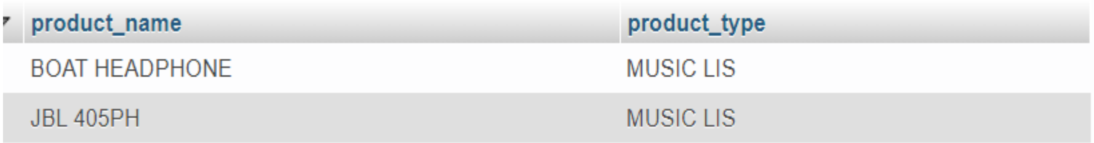
QUERIES

@PL/SQL:-

1) VIEW:-

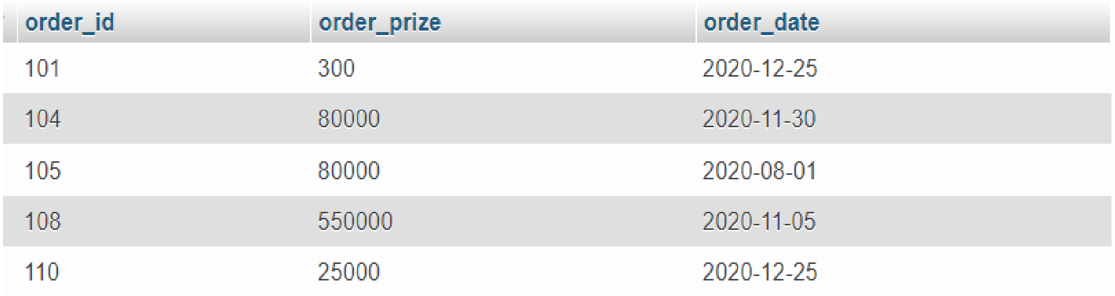
(i) CREATE:

```
CREATE VIEW product_specific AS
SELECT product_name,product_type FROM product
WHERE product_type='MUSIC LIS';
```



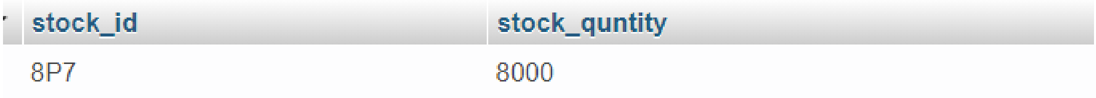
product_name	product_type
BOAT HEADPHONE	MUSIC LIS
JBL 405PH	MUSIC LIS

```
CREATE VIEW order_date_half_year AS
SELECT order_id,order_prize,order_date FROM order_d
WHERE order_date>='2020-06-01';
```



order_id	order_prize	order_date
101	300	2020-12-25
104	80000	2020-11-30
105	80000	2020-08-01
108	550000	2020-11-05
110	25000	2020-12-25

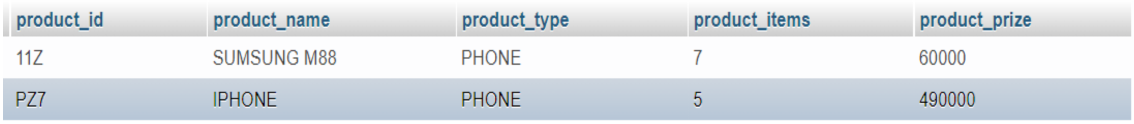
```
CREATE VIEW stock_man AS
SELECT stock_id,stock_quntity FROM stock
WHERE stock_quntity >=1000 and stock_type='MUSIC LIS';
```



stock_id	stock_quntity
8P7	8000

(ii) UPDATE:-

```
CREATE OR REPLACE VIEW product_specific AS
SELECT *FROM product
WHERE product_type='PHONE';
```



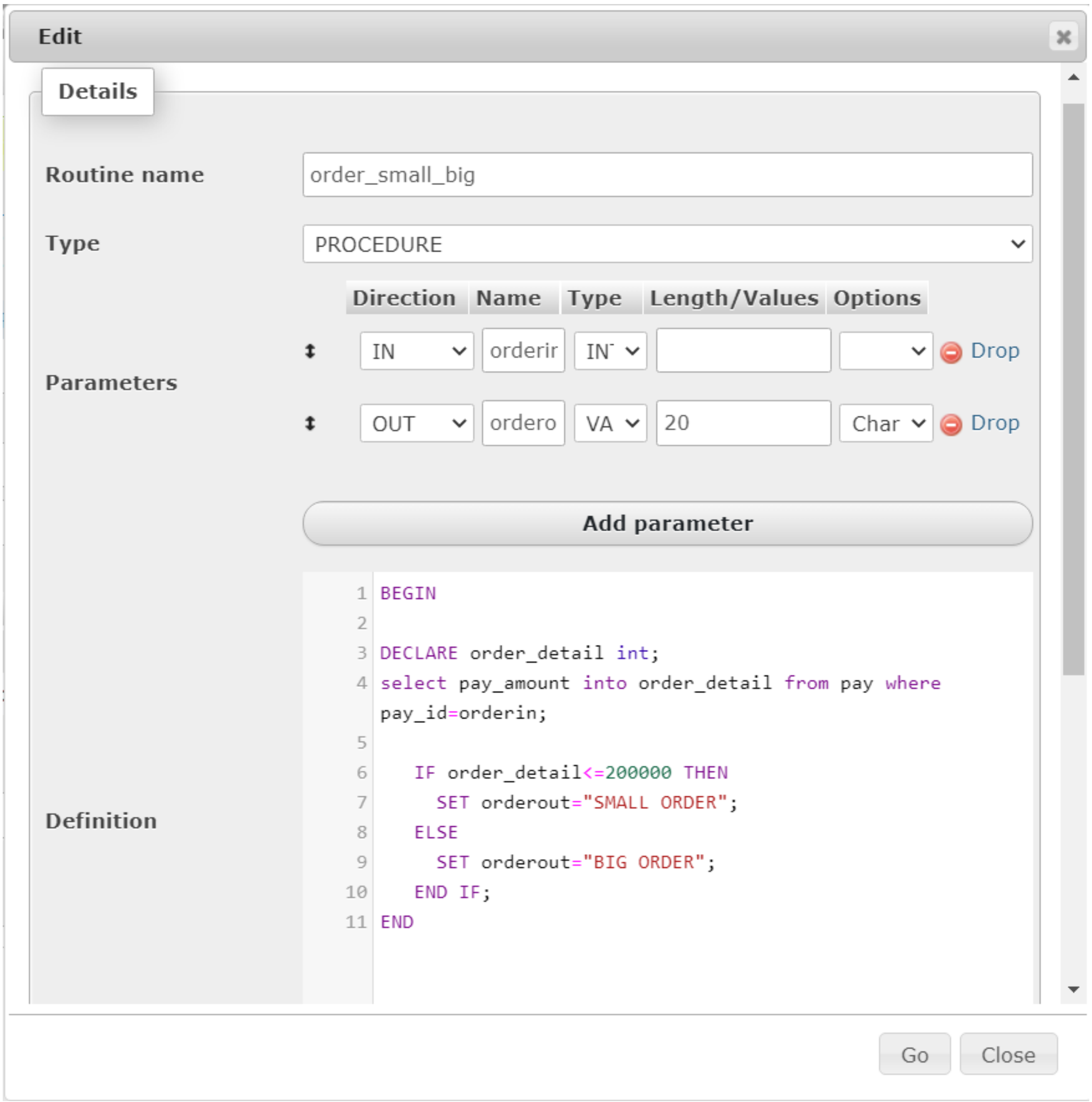
product_id	product_name	product_type	product_items	product_prize
11Z	SUMSUNG M88	PHONE	7	60000
PZ7	IPHONE	PHONE	5	490000

(iii) DROP:-

```
drop view product_specific;
```

2) IF AND ELSE:-

// SHOW THAT THE ORDER IS BIG ORDER OR SMALL ORDER USING PAY_ID.



Routine name: order_small_big

Type: PROCEDURE

Parameters

Direction	Name	Type	Length/Values	Options
IN	orderir	IN`		
OUT	ordero	VA	20	Char

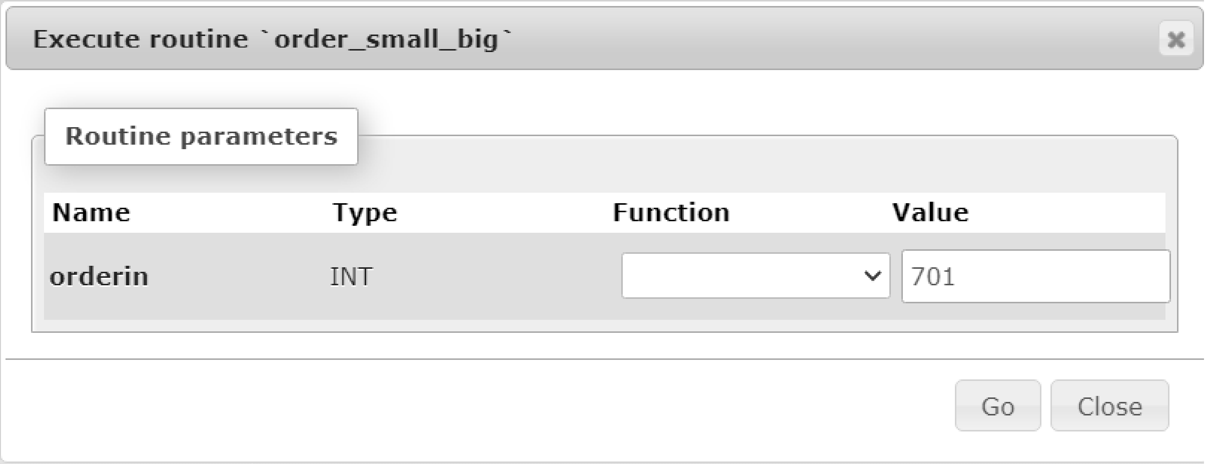
Definition

```
BEGIN

DECLARE order_detail int;
select pay_amount into order_detail from pay where
pay_id=orderin;

   IF order_detail<=200000 THEN
     SET orderout="SMALL ORDER";
   ELSE
     SET orderout="BIG ORDER";
   END IF;
END
```

X

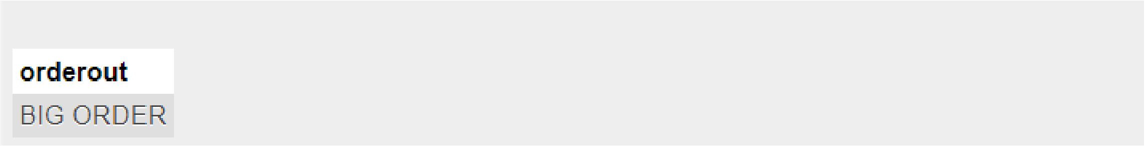
EXECUTE:-

(A):-


Execute routine `order_small_big`

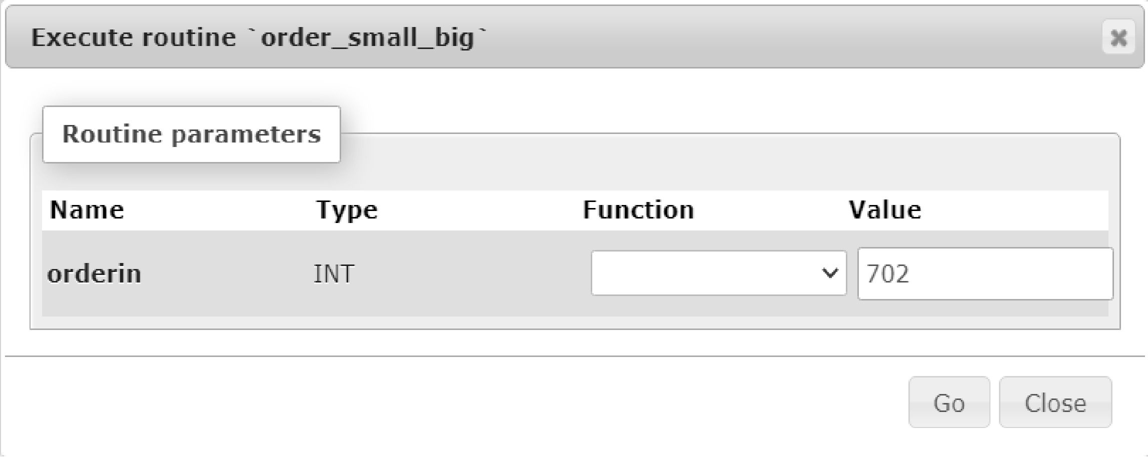
Routine parameters

Name	Type	Function	Value
orderin	INT		701



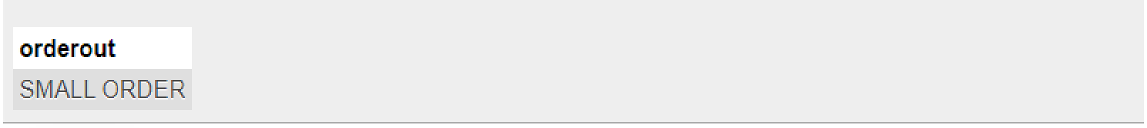
orderout
BIG ORDER

(B):-


Execute routine `order_small_big`

Routine parameters

Name	Type	Function	Value
orderin	INT		702

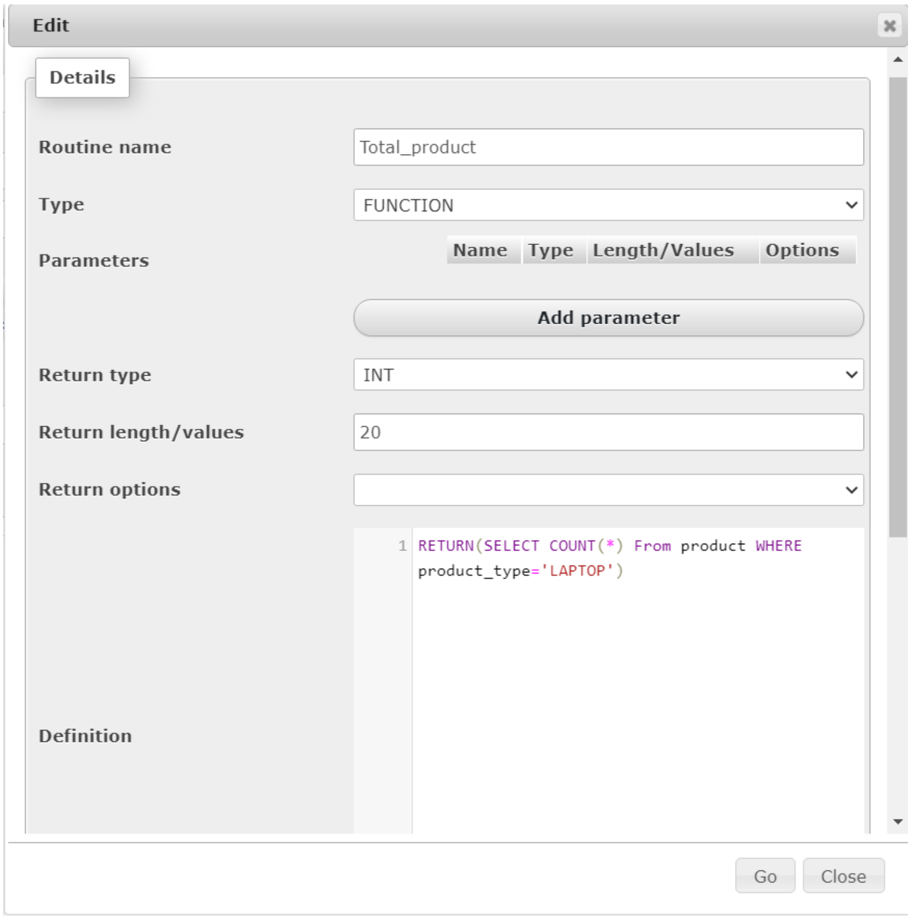


orderout
SMALL ORDER

X

3) FUNCTION:-

// SHOW THE COUNT WHERE THE PRODUCT TYPE IS EQUAL TO LAPTOP USING FUNCTION.



Routine name: Total_product

Type: FUNCTION

Parameters

Name	Type	Length/Values	Options

Return type: INT

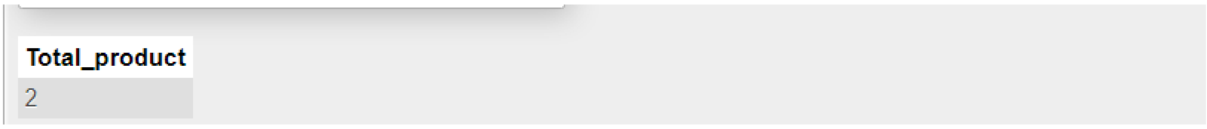
Return length/values: 20

Return options:

Definition

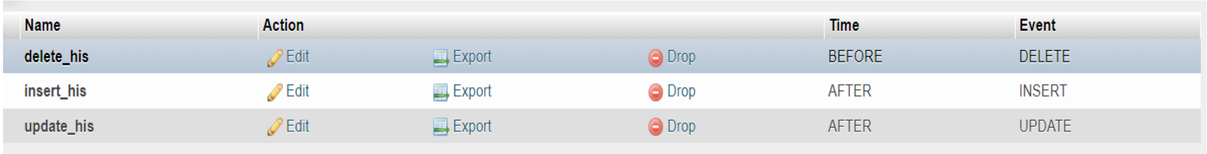
```
RETURN(SELECT COUNT(*) From product WHERE
product_type='LAPTOP')
```

EXECUTE:-

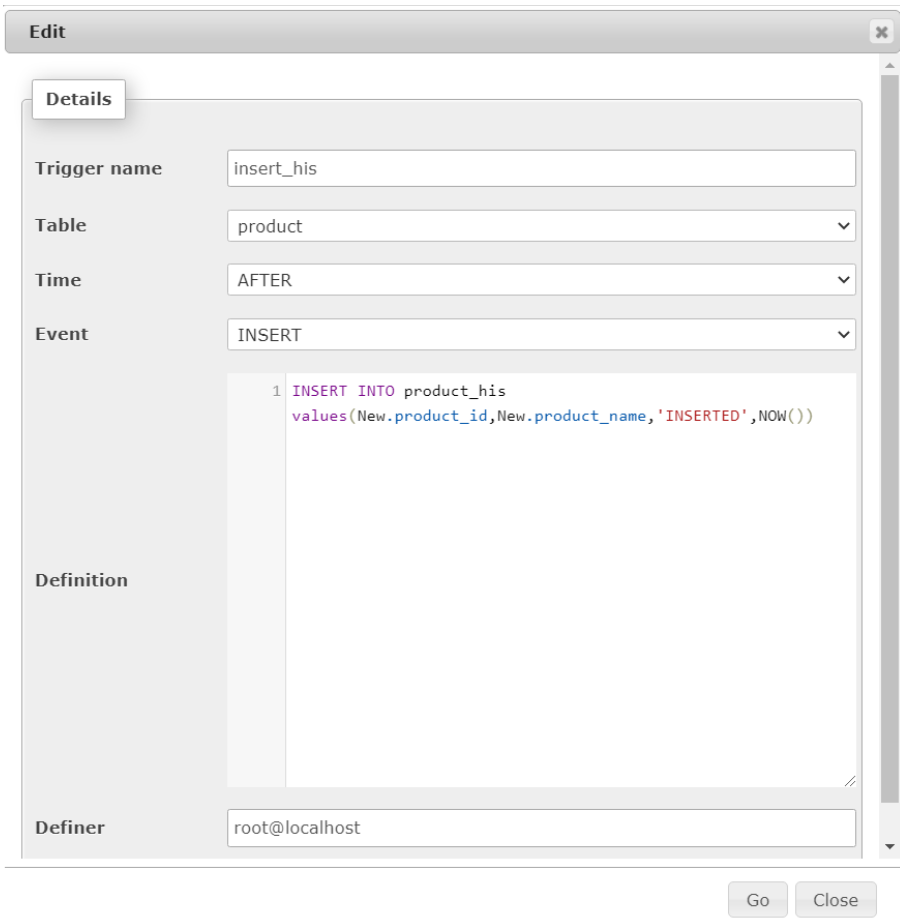


Total_product
2

4) Trigger:-



Name	Action			Time	Event
delete_his	Edit	Export	Drop	BEFORE	DELETE
insert_his	Edit	Export	Drop	AFTER	INSERT
update_his	Edit	Export	Drop	AFTER	UPDATE



Trigger name: insert_his

Table: product

Time: AFTER

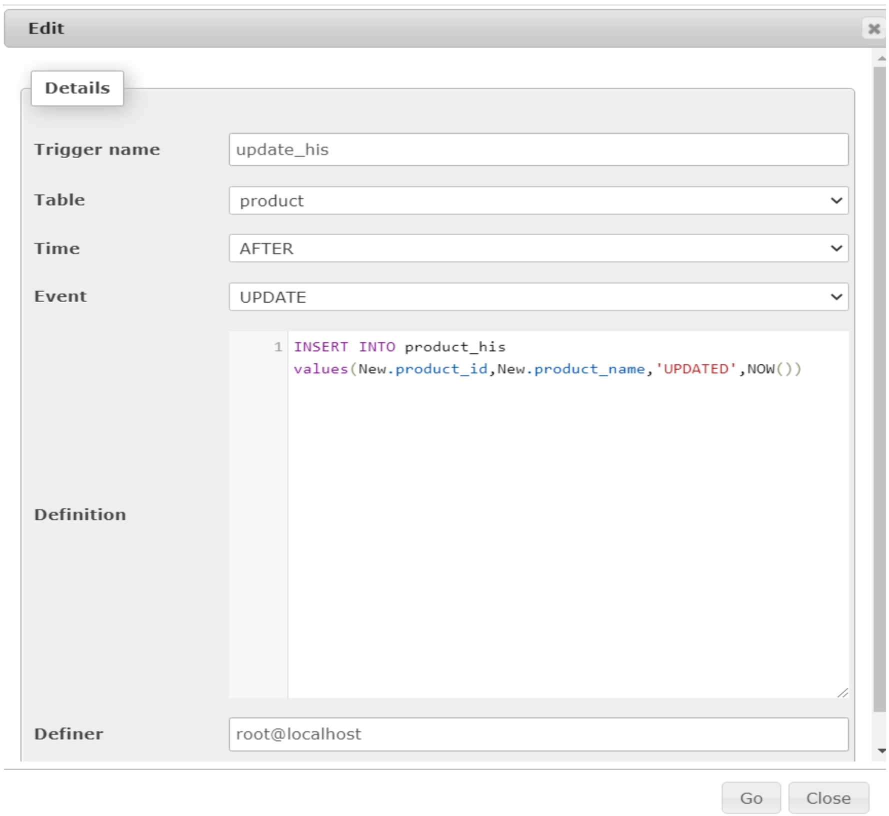
Event: INSERT

Definition:

```
INSERT INTO product_his
values(New.product_id,New.product_name,'INSERTED',NOW())
```

Definer: root@localhost

X

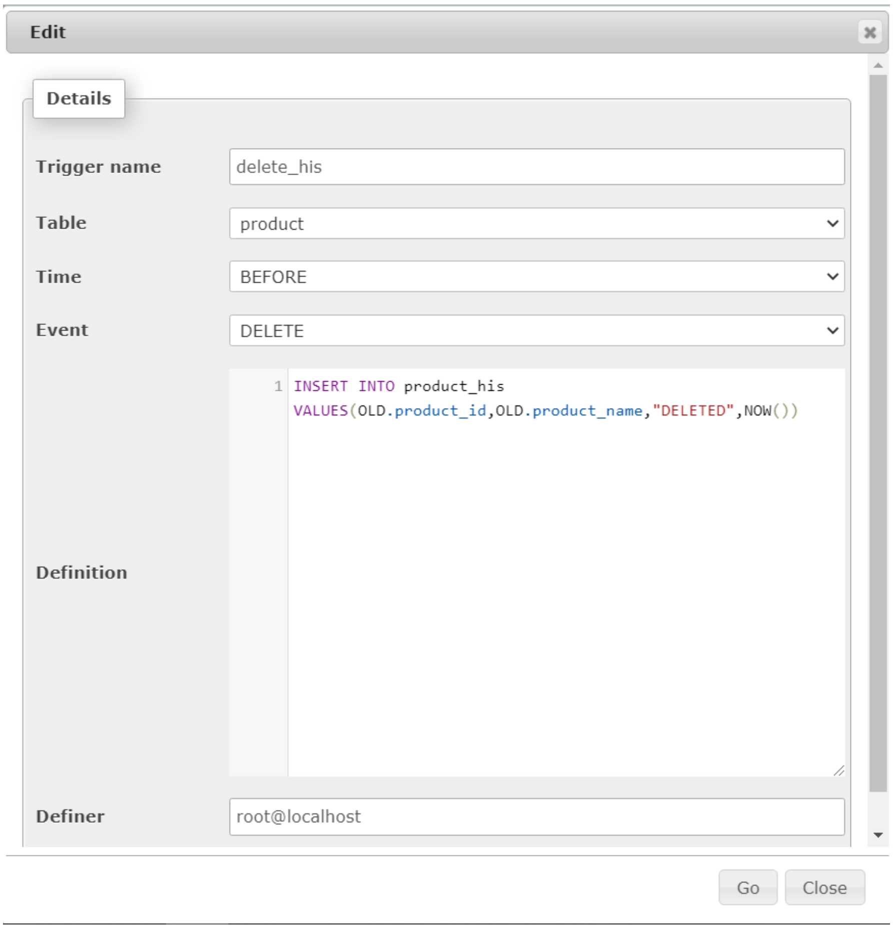


Trigger name	update_his
Table	product
Time	AFTER
Event	UPDATE

Definition

```
INSERT INTO product_his
values(New.product_id,New.product_name,'UPDATED',NOW())
```

Definer: root@localhost



Trigger name	delete_his
Table	product
Time	BEFORE
Event	DELETE

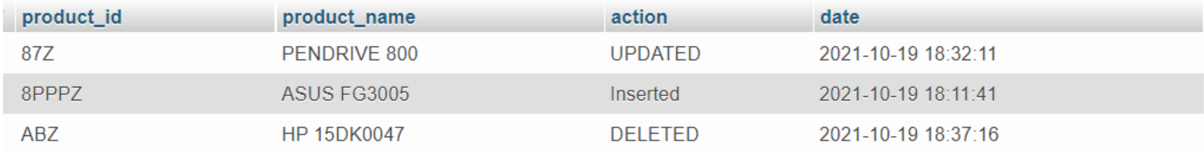
Definition

```
INSERT INTO product_his
VALUES(OLD.product_id,OLD.product_name,"DELETED",NOW())
```

Definer: root@localhost

X

EXECUTE:-



product_id	product_name	action	date
87Z	PENDRIVE 800	UPDATED	2021-10-19 18:32:11
8PPPZ	ASUS FG3005	Inserted	2021-10-19 18:11:41
ABZ	HP 15DK0047	DELETED	2021-10-19 18:37:16

DROP TRIGGER:-

DROP TRIGGER delete_his ON product;

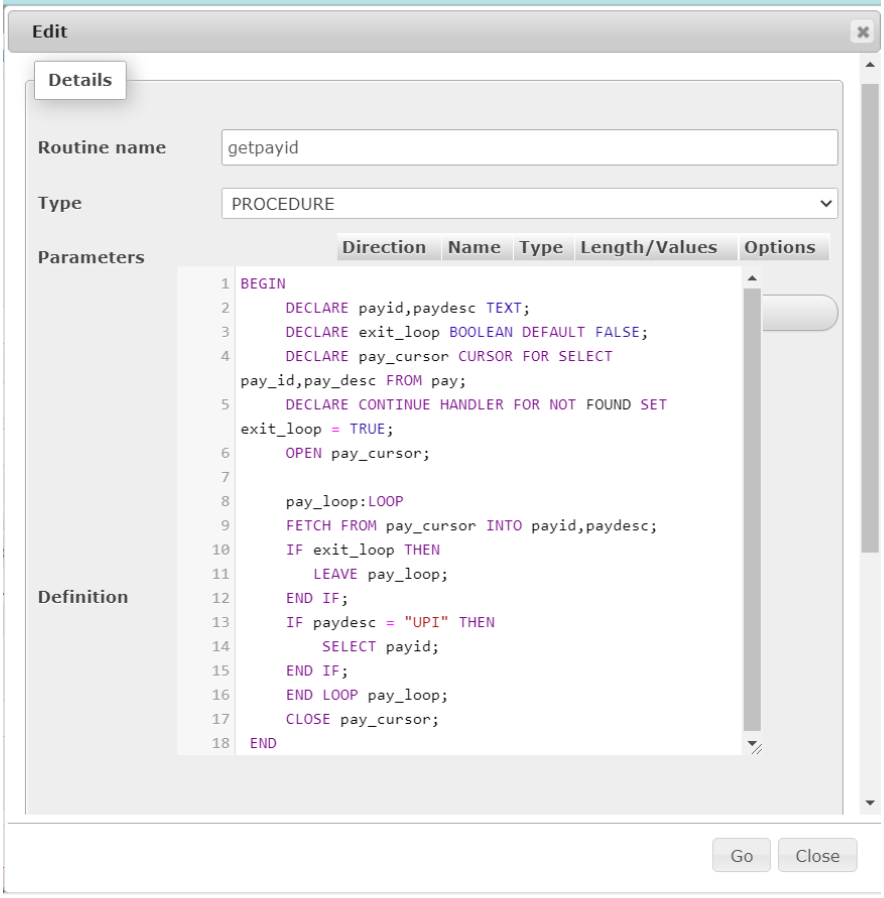
TRIGGER DISABLE AND ENABLE:-

ALTER TABLE product DISABLE TRIGGER delete_his/ALL

ALTER TABLE product ENABLE TRIGGER delete_his/ALL

5) Cursur:-

// SHOW ALL PAY ID WHICH PAID THROUGH UPI USING CURSOR.



Routine name: getpayid

Type: PROCEDURE

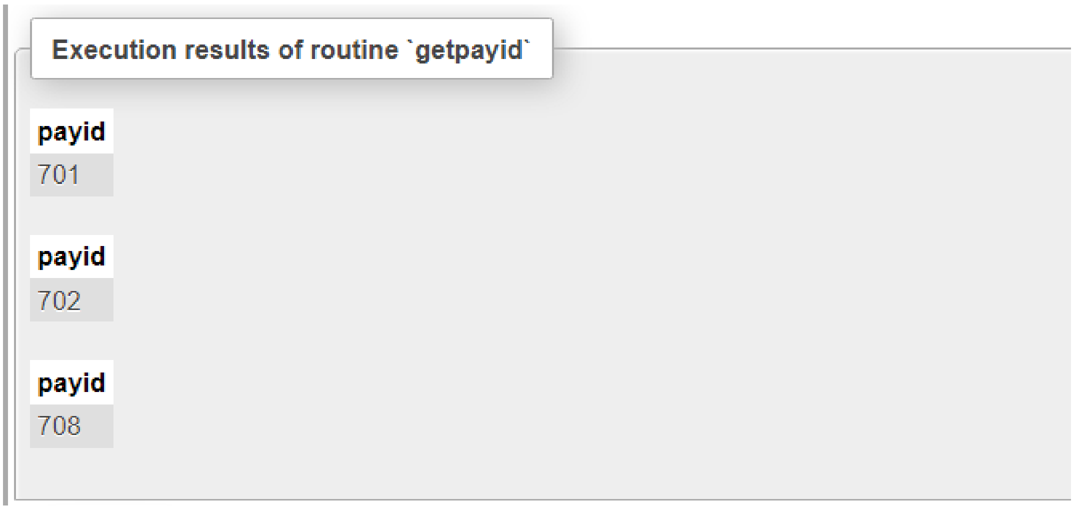
Parameters

Definition

```
BEGIN
     DECLARE payid,paydesc TEXT;
     DECLARE exit_loop BOOLEAN DEFAULT FALSE;
     DECLARE pay_cursor CURSOR FOR SELECT
pay_id,pay_desc FROM pay;
     DECLARE CONTINUE HANDLER FOR NOT FOUND SET
exit_loop = TRUE;
     OPEN pay_cursor;

     pay_loop:LOOP
     FETCH FROM pay_cursor INTO payid,paydesc;
     IF exit_loop THEN
        LEAVE pay_loop;
     END IF;
     IF paydesc = "UPI" THEN
         SELECT payid;
     END IF;
     END LOOP pay_loop;
     CLOSE pay_cursor;
 END
```

X

EXECUTE:-



Execution results of routine `getpayid`

```
payid
701

payid
702

payid
708
```
