<u>**PROJECT**</u>

On

"COMPUTER SERVICE CENTRE USING SQL"

<u>**Student Details :**</u>

Name of the Student : ADDETLA SAI CHETHAN

Registration Number : 12109834

Name of the programe : P132::B.Tech(Computer Science & Engineering)

--INDEX--

1. Abstract

This **Computer Service System** has a public site where the shop's clients or possible clients can explore the services they offers. Site visitors can also list all the products with some details such as the price of the item that is available in the shop. On the Management Panel, the system requires the users to log in with their valid user credentials in order to gain access to the said side. The system consists of 3 types of user roles which are **Administrator**, **Staff**, and **Technician**. The **Administrator users** have the privilege to access and manage all the features and functionalities of the system. They are also the ones who can manage the list of users and update system information. The **Staff users** have only limited permissions while the **Technician** is only allowed to manage the transaction assigned to him/her. The system also generates a printable transaction detail.

2.Schema and Normalisation

2.1 DESIGN

2.1.1 E-R Diagram:

In the last page

2.1.2 Entities :

In total we have 4 entities and information of each entity is mentioned below

1.CUSTOMER DETAILS :

(Cust_ID numeric, Fname varchar, Town varchar, Telephone varchar, Email varchar)

This table will store the information of the customer/clients. In this Table Staff can see all required detailes of the client. It includes all the information of the client which includes telephone number and Email id also…

```
CID numeric(9),
cname varchar(100) NOT NULL,
city varchar(30) NOT NULL,
mobileno varchar(15) NOT NULL,
Email varchar(500) NOT NULL,
```

2. Staff Details:

(SID int, sname varchar(20),city varchar(20), Telephone char, Email varchar)

This table will store the information of the Staff who are working. In this Table Customer can see all required detailes of the Staff. It includes all the information of the Staff which includes Place and Email id also…

SID int,

sname varchar(100) NOT NULL,

city varchar(30) NOT NULL,

mobileno int NOT NULL,

Email varchar(500) NOT NULL,

3. Repairing Details

(RID int,R_date(30),R_problem varchar(30), C_ID numeric, S_ID numeric)

This table gives information about the types of repairing services to laptops and the brand name also mentioned.

It give complete information to the client..

RID numeric (9),

R_date varchar(20),

Cust_ID numeric(9) NOT NULL,

Staff_ID numeric(9) NOT NULL,

R_problem varchar(1000) NOT NULL,

4. Totalprice Details :

(Rep_ID numeric, Staff_ID numeric, GST decimal, Discount decimal, Total decimal,price int)

This table give the billing details to the customer. By this customer can check the Prices and Discounts on products which they are selected…

Rep_ID numeric(9) NOT NULL,

Staff_ID numeric(9) NOT NULL,

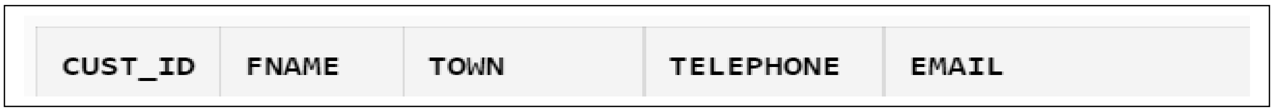
GST decimal(4,2),

Discount decimal(4,2),

Price int

Normalization

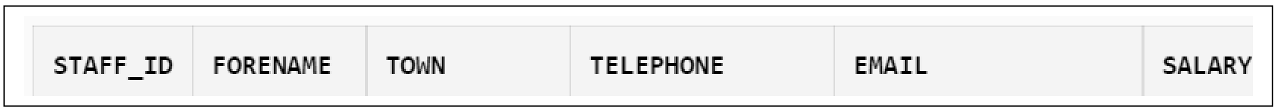
1. CUSTOMER



CUST_ID	FNAME	TOWN	TELEPHONE	EMAIL

- Relation Customer has no multi-valued attribute.
- All non-key attributes are fully functional dependent on the primary key.
- There is no transitive dependency for non-prime attributes.
- The relation is in 3NF.

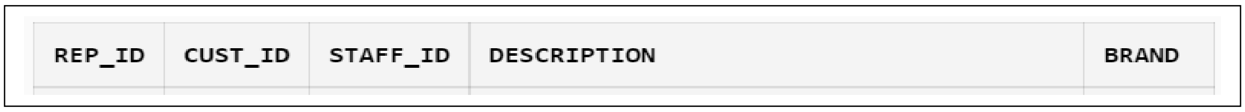
2. STAFF



STAFF_ID	FORENAME	TOWN	TELEPHONE	EMAIL	SALARY

- Relation Customer has no multi-valued attribute.
- All non-key attributes are fully functional dependent on the primary key.
- There is no transitive dependency for non-prime attributes.
- The relation is in 3NF.

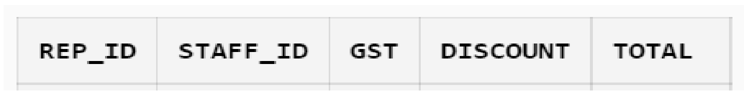
3.REPAIR



REP_ID	CUST_ID	STAFF_ID	DESCRIPTION	BRAND

- Relation Customer has no multi-valued attribute.
- All non-key attributes are fully functional dependent on the primary key.
- There is no transitive dependency for non-prime attributes.
- The relation is in 3NF.

4.ESTIMATION



REP_ID	STAFF_ID	GST	DISCOUNT	TOTAL

- Relation Customer has no multi-valued attribute.
- All non-key attributes are fully functional dependent on the primary key.
- There is no transitive dependency for non-prime attributes.
- The relation is in 3NF. `

RELATIONSHIP

1.Customer and Repair :

A Customer can give many Items for repair. And a customer can give many items on one Customer ID

Type=”many to many”

2.Customer and Estimation:

A customer can pay many bills through this Customer ID

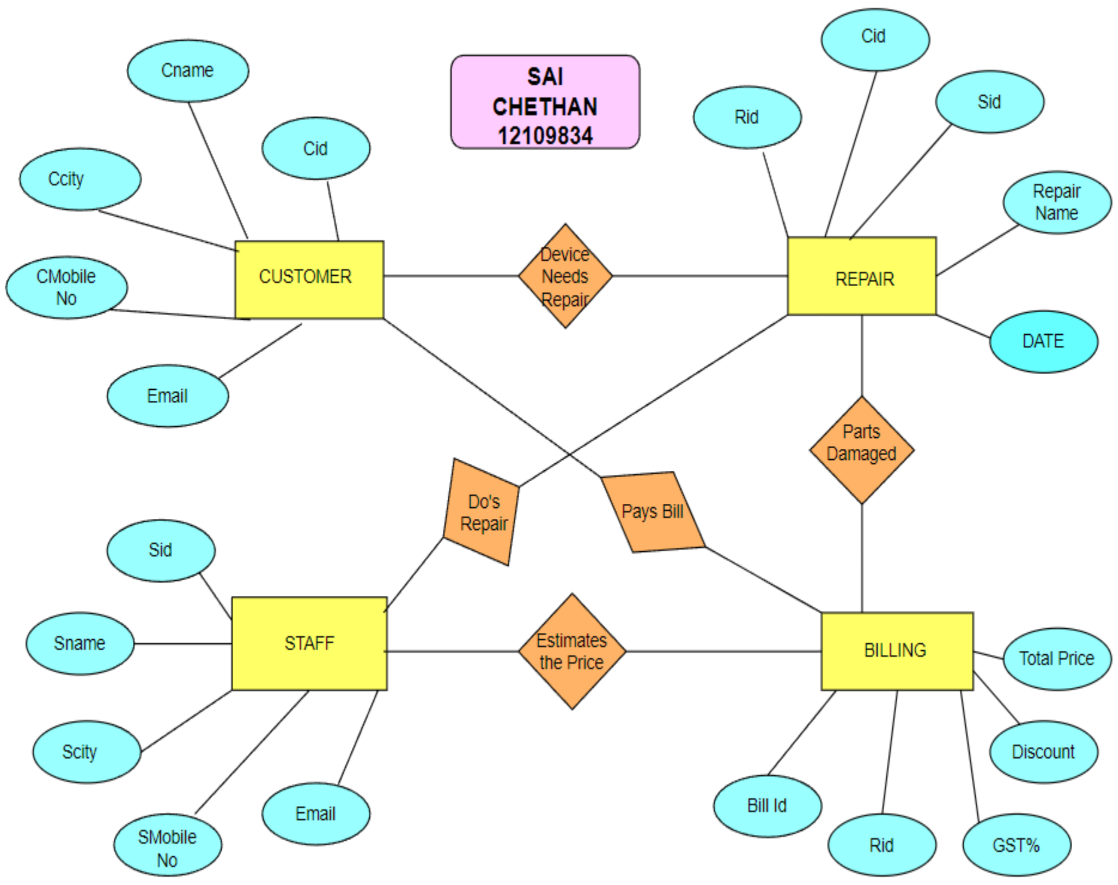
Type=”many to many”

3.Staff and Estimation:

Staff Estimates the total cost of the customer. One staff member can estimate the total cost once only.

Type=”one to one”

ER DIAGRAM


SAI
CHETHAN
12109834
CUSTOMER
Cname
Cid
Ccity
CMobile No
Email
Device Needs Repair
REPAIR
Rid
Cid
Sid
Repair Name
DATE
Parts Damaged
Do's Repair
Pays Bill
STAFF
Sid
Sname
Scity
SMobile No
Email
Estimates the Price
BILLING
Total Price
Discount
Bill Id
Rid
GST%

3. SQL&PLSQL

3.1 SQL Table Creation

The Implementation of SQL server :

```
CREATE TABLE cust(cid int primary key,
cname varchar(50) not null,
ccity varchar(50) not null,
cmobile_no int ,
email varchar(50))

CREATE TABLE Staff (
sid int primary key,
sname varchar(50) not null,
scity varchar(50) not null,
smobile_no int ,
email varchar(50))


  CONSTRAINT EMAIL UNIQUE (Email),
  CONSTRAINT smobile UNIQUE (smobile),
CONSTRAINT pk_Staff PRIMARY KEY (sid))
```

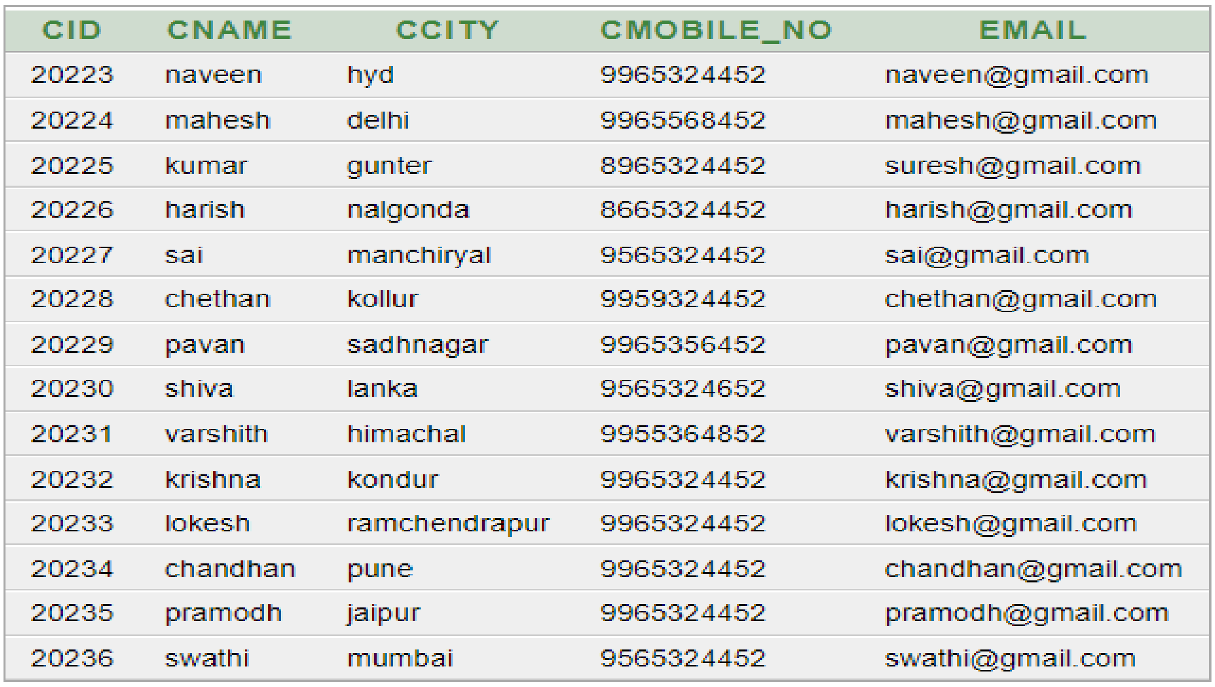
```
CREATE TABLE Repairs (
        Rep_ID numeric(9),
        Cust_ID numeric(9) NOT NULL,
        Staff_ID numeric(9) NOT NULL,
        Description varchar(1000) NOT NULL,
        Brand varchar(50) NOT NULL,
        CONSTRAINT pk_Repairs PRIMARY KEY (Rep_ID),
        CONSTRAINT fk_Repairs_Cust FOREIGN KEY (Cust_ID)
REFERENCES Customers,
CONSTRAINT fk_Repairs_Staff FOREIGN KEY (Staff_ID)
REFERENCES Staff);


CREATE TABLE Totalbill (
        Rep_ID numeric(9) NOT NULL,
        Staff_ID numeric(9) NOT NULL,
        GST decimal(4,2),
        Discount decimal(4,2),
        Price decimal(9,2) NOT NULL,
        CONSTRAINT pk_EstiSHOmates PRIMARY KEY (Rep_ID),
  CONSTRAINT fk_Estimates_Staff FOREIGN KEY (Staff_ID)
REFERENCES Staff,
CONSTRAINTS fk_Estimates_Repairs FOREIGN KEY (Rep_ID)
REFERENCES Repairs);
```

3.2 Data Entry

Customer table

```
insert into cust values (20223,'naveen','hyd',9965324452,'naveen@gmail.com')

insert into cust values
(20224,'mahesh','delhi',9965568452,'mahesh@gmail.com')

insert into cust values
(20225,'kumar','gunter',8965324452,'suresh@gmail.com')

insert into cust values
(20226,'harish','nalgonda',8665324452,'harish@gmail.com')

insert into cust values (20227,'sai','manchiryal',9565324452,'sai@gmail.com')

insert into cust values
(20228,'chethan','kollur',9959324452,'chethan@gmail.com')

insert into cust values
(20229,'pavan','sadhnagar',9965356452,'pavan@gmail.com')

insert into cust values (20230,'shiva','lanka',9565324652,'shiva@gmail.com')

insert into cust values
(20231,'varshith','himachal',9955364852,'varshith@gmail.com')

insert into cust values
(20232,'krishna','kondur',9965324452,'krishna@gmail.com')

insert into cust values
(20233,'lokesh','ramchendrapur',9965324452,'lokesh@gmail.com')

insert into cust values
(20234,'chandhan','pune',9965324452,'chandhan@gmail.com')

insert into cust values
(20235,'pramodh','jaipur',9965324452,'pramodh@gmail.com')

insert into cust values
(20236,'swathi','mumbai',9565324452,'swathi@gmail.com')
```

```
select *from Customer
```



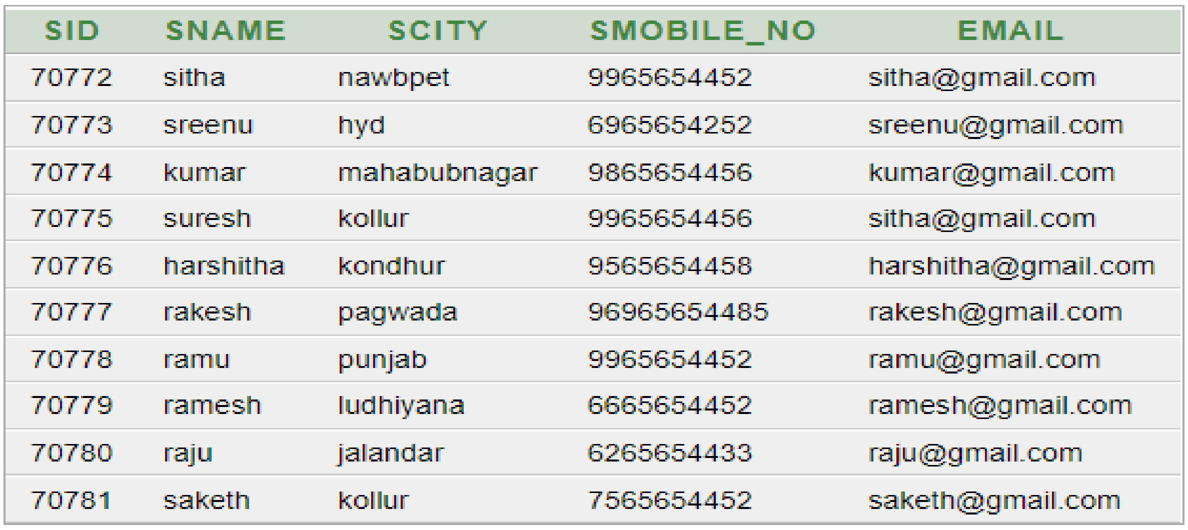
CID	CNAME	CCITY	CMOBILE_NO	EMAIL
20223	naveen	hyd	9965324452	naveen@gmail.com
20224	mahesh	delhi	9965568452	mahesh@gmail.com
20225	kumar	gunter	8965324452	suresh@gmail.com
20226	harish	nalgonda	8665324452	harish@gmail.com
20227	sai	manchiryal	9565324452	sai@gmail.com
20228	chethan	kollur	9959324452	chethan@gmail.com
20229	pavan	sadhnagar	9965356452	pavan@gmail.com
20230	shiva	lanka	9565324652	shiva@gmail.com
20231	varshith	himachal	9955364852	varshith@gmail.com
20232	krishna	kondur	9965324452	krishna@gmail.com
20233	lokesh	ramchendrapur	9965324452	lokesh@gmail.com
20234	chandhan	pune	9965324452	chandhan@gmail.com
20235	pramodh	jaipur	9965324452	pramodh@gmail.com
20236	swathi	mumbai	9565324452	swathi@gmail.com

STAFF TABLE

```
create table staf(sid int primary key,sname varchar(50) not null,scity
varchar(50) not null,smobile_no int ,email varchar(50))

insert into staf values
(70772,'sitha','nawbpet',9965654452,'sitha@gmail.com')

insert into staf values
(70773,'sreenu','hyd',6965654252,'sreenu@gmail.com')

insert into staf values
(70774,'kumar','mahabubnagar',9865654456,'kumar@gmail.com')

insert into staf values
(70775,'suresh','kollur',9965654456,'sitha@gmail.com')

insert into staf values
(70776,'harshitha','kondhur',9565654458,'harshitha@gmail.com')

insert into staf values
(70777,'rakesh','pagwada',96965654485,'rakesh@gmail.com')
```

```
insert into staf values
(70778,'ramu','punjab',9965654452,'ramu@gmail.com')

insert into staf values
(70779,'ramesh','ludhiyana',6665654452,'ramesh@gmail.com')

insert into staf values
(70780,'raju','jalandar',6265654433,'raju@gmail.com')

insert into staf values
(70781,'saketh','kollur',7565654452,'saketh@gmail.com')
```

select *from Staff

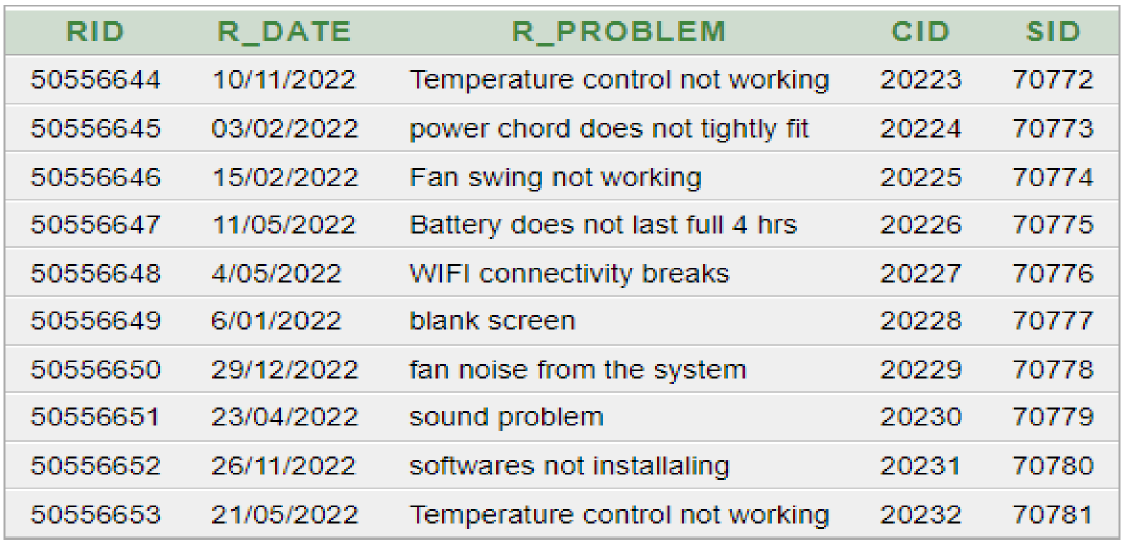


SID	SNAME	SCITY	SMOBILE_NO	EMAIL
70772	sitha	nawbpet	9965654452	sitha@gmail.com
70773	sreenu	hyd	6965654252	sreenu@gmail.com
70774	kumar	mahabubnagar	9865654456	kumar@gmail.com
70775	suresh	kollur	9965654456	sitha@gmail.com
70776	harshitha	kondhur	9565654458	harshitha@gmail.com
70777	rakesh	pagwada	96965654485	rakesh@gmail.com
70778	ramu	punjab	9965654452	ramu@gmail.com
70779	ramesh	ludhiyana	6665654452	ramesh@gmail.com
70780	raju	jalandar	6265654433	raju@gmail.com
70781	saketh	kollur	7565654452	saketh@gmail.com

REPAIR TABLLE

```
create table repair(rid int primary key,r_date varchar(20) ,r_problem varchar(50) not null,cid int ,CONSTRAINT FK_PersonOrder FOREIGN KEY (cid) REFERENCES cust(cid),sid int ,CONSTRAINT FK_Person FOREIGN KEY (sid) REFERENCES staf(sid))

insert into repair values (50556644,'10/11/2022','Temperature control not working',20223,70772)

insert into repair values (50556645,'03/02/2022','power chord does not tightly fit',20224,70773)

insert into repair values (50556646,'15/02/2022','Fan swing not working',20225,70774)

insert into repair values (50556647,'11/05/2022','Battery does not last full 4 hrs',20226,70775)

insert into repair values (50556648,'4/05/2022','WIFI connectivity breaks ',20227,70776)

insert into repair values (50556649,'6/01/2022','blank screen',20228,70777)

insert into repair values (50556650,'29/12/2022','fan noise from the system ',20229,70778)

insert into repair values (50556651,'23/04/2022','sound problem',20230,70779)

insert into repair values (50556652,'26/11/2022','softwares not installaling',20231,70780)

insert into repair values (50556653,'21/05/2022','Temperature control not working',20232,70781)
```

```
select *from Repairs
```

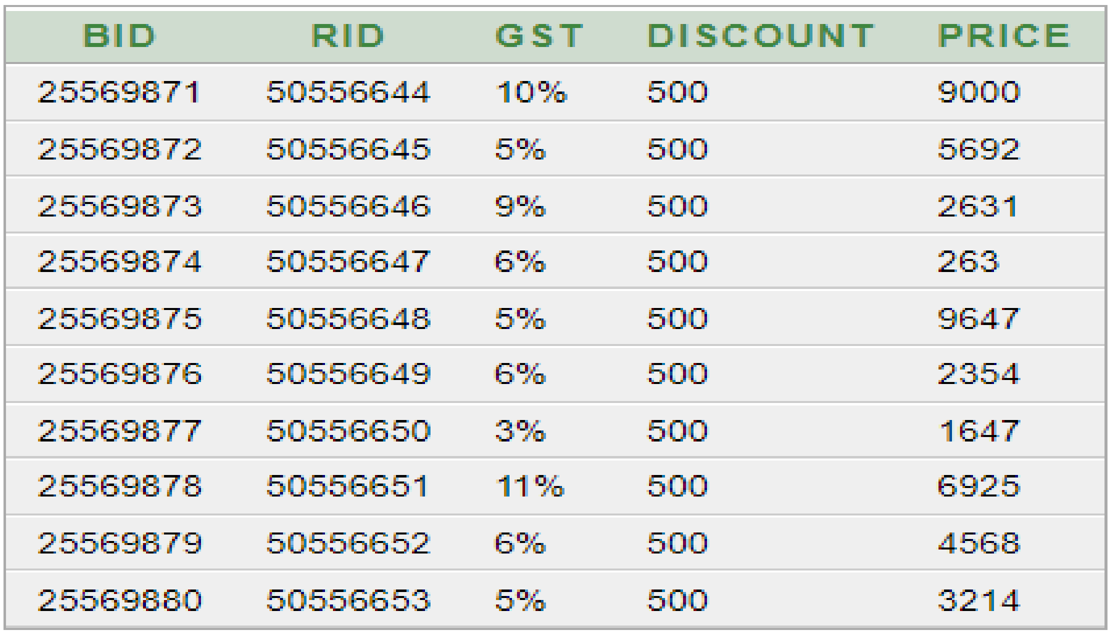


RID	R_DATE	R_PROBLEM	CID	SID
50556644	10/11/2022	Temperature control not working	20223	70772
50556645	03/02/2022	power chord does not tightly fit	20224	70773
50556646	15/02/2022	Fan swing not working	20225	70774
50556647	11/05/2022	Battery does not last full 4 hrs	20226	70775
50556648	4/05/2022	WIFI connectivity breaks	20227	70776
50556649	6/01/2022	blank screen	20228	70777
50556650	29/12/2022	fan noise from the system	20229	70778
50556651	23/04/2022	sound problem	20230	70779
50556652	26/11/2022	softwares not installaling	20231	70780
50556653	21/05/2022	Temperature control not working	20232	70781

TOTALBILL TABLE

```
create table totalbill(bid int primary key,rid int ,constraint FK_hello
FOREIGN KEY(rid) REFERENCES repair(rid),GST varchar(20),discount
int )

alter table totalbill add price int;

insert into totalbill values (25569871,50556644,'10%',500,9000)

insert into totalbill values (25569872,50556645,'5%',500,5692)

insert into totalbill values (25569873,50556646,'9%',500,2631)

insert into totalbill values (25569874,50556647,'6%',500,263)

insert into totalbill values (25569875,50556648,'5%',500,9647)

insert into totalbill values (25569876,50556649,'6%',500,2354)

insert into totalbill values (25569877,50556650,'3%',500,1647)

insert into totalbill values (25569878,50556651,'11%',500,6925)

insert into totalbill values (25569879,50556652,'6%',500,4568)

insert into totalbill values (25569880,50556653,'5%',500,3214)
```

select *from totalbill



BID	RID	GST	DISCOUNT	PRICE
25569871	50556644	10%	500	9000
25569872	50556645	5%	500	5692
25569873	50556646	9%	500	2631
25569874	50556647	6%	500	263
25569875	50556648	5%	500	9647
25569876	50556649	6%	500	2354
25569877	50556650	3%	500	1647
25569878	50556651	11%	500	6925
25569879	50556652	6%	500	4568
25569880	50556653	5%	500	3214

BACK UP TABLES

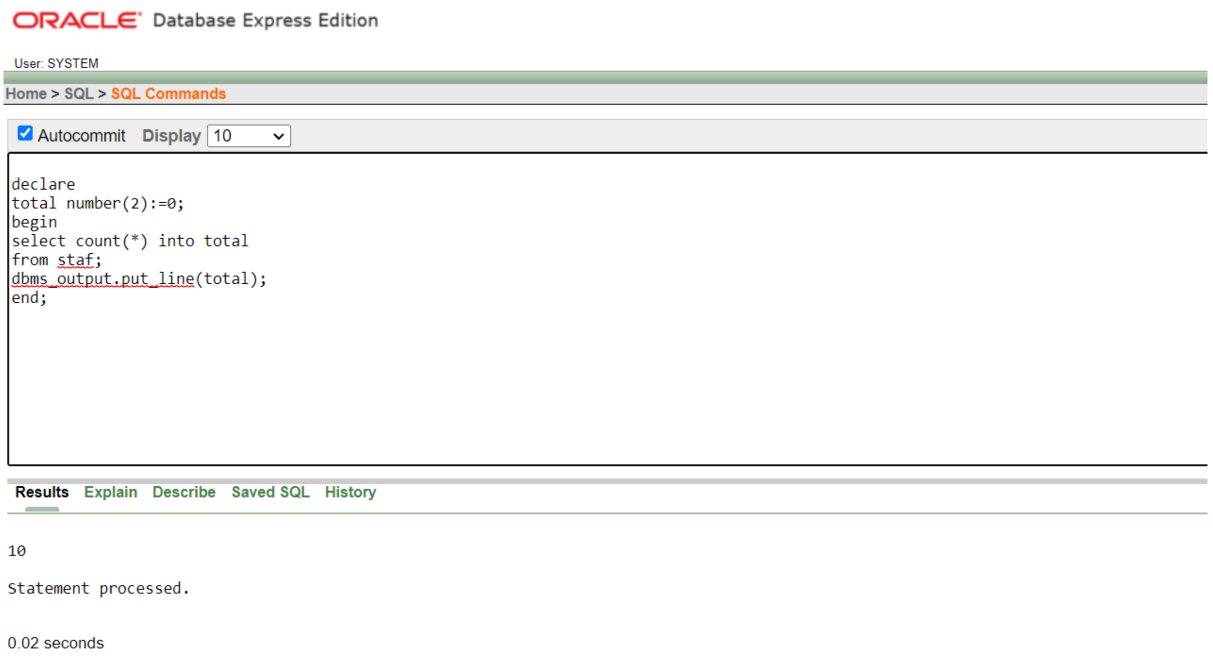
```
create table BACKUP1(cid int primary key,
cname varchar(50) not null,
ccity varchar(50) not null,
cmobile_no int ,
email varchar(50))
```

```
create table BACKUP2(cid int primary key,
cname varchar(50) not null,
ccity varchar(50) not null,
cmobile_no int ,
email varchar(50))
```

3.3 PL/ SQL Examples

<u>Example </u>-1

```
declare
total number(2):=0;
begin
select count(*) into total
from staff;
dbms_output.put_line(total);
end;
```



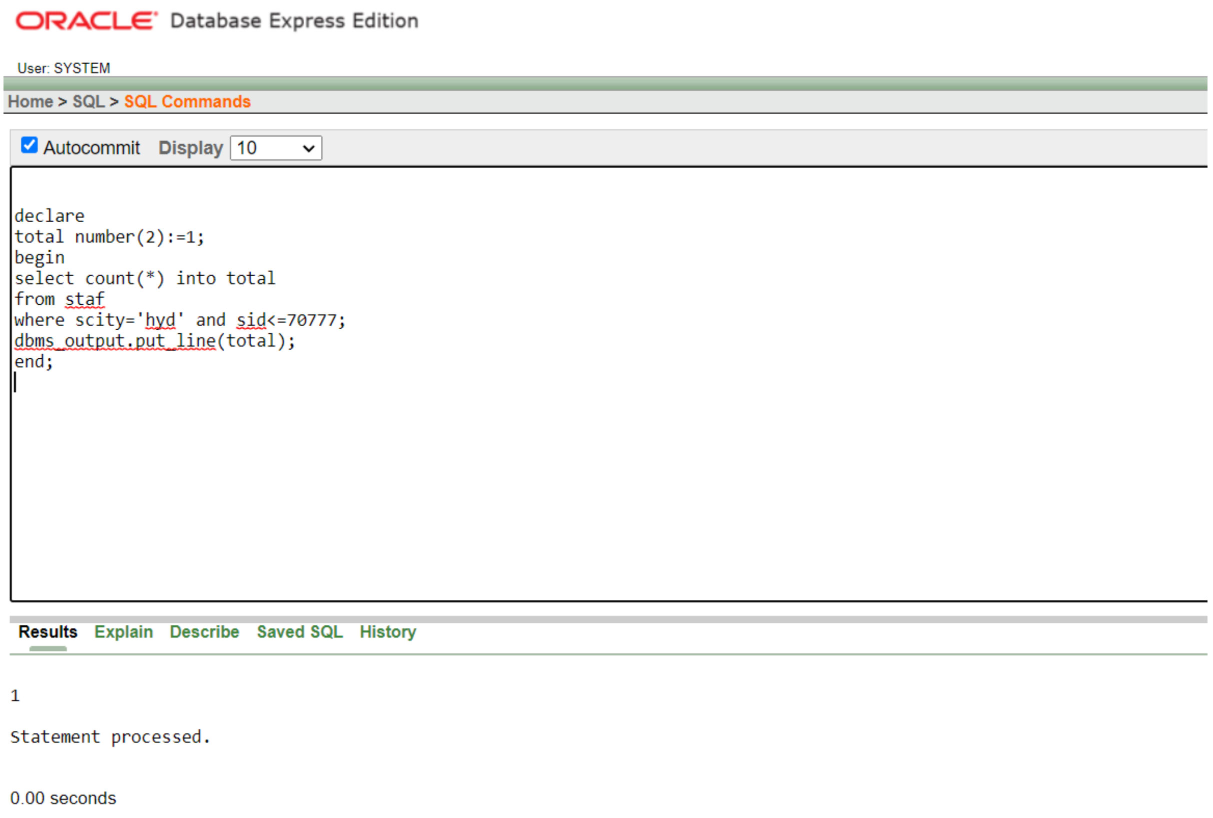
```
declare
total number(2):=0;
begin
select count(*) into total
from staf;
dbms_output.put_line(total);
end;
```

```
10

Statement processed.
```

0.02 seconds

Example-2

```
declare
total number(2):=0;
begin
select count(*) into total
from staff
where town='KKG' and salary>=50000;
dbms_output.put_line(total);
end;
```

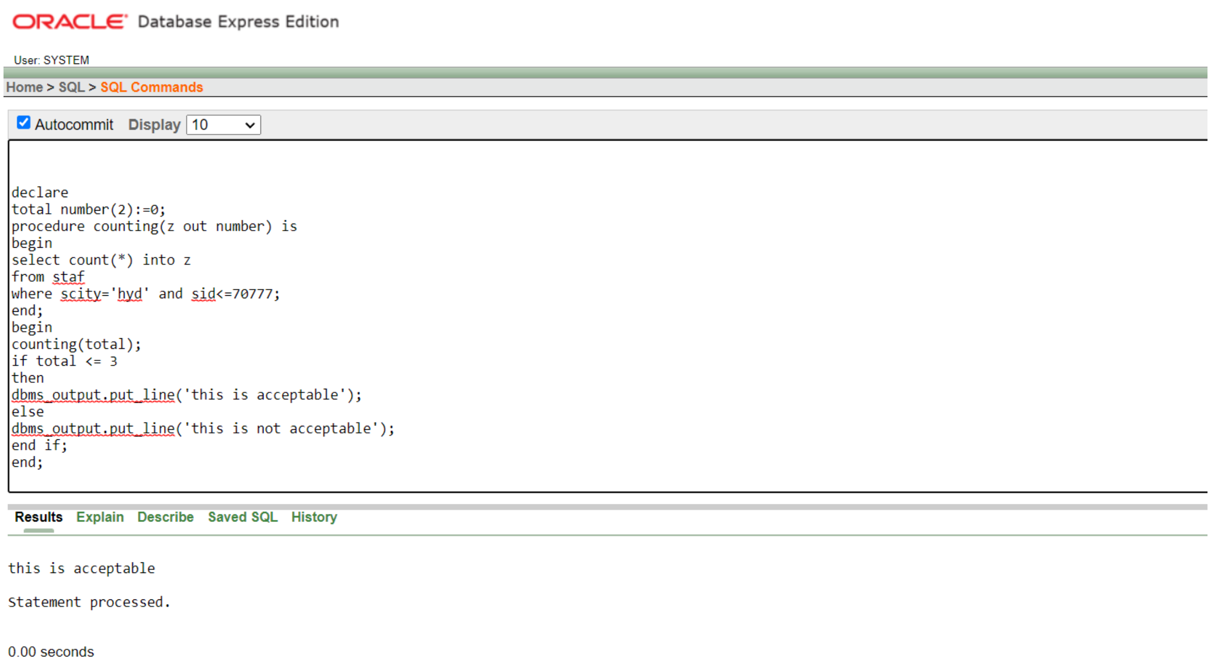

ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
declare
total number(2):=1;
begin
select count(*) into total
from staf
where scity='hyd' and sid<=70777;
dbms_output.put_line(total);
end;
```

```
1

Statement processed.

0.00 seconds
```

Example -3

```
declare
total number(2):=0;
procedure counting(z out number) is
begin
select count(*) into z
from staff
where town='KKG' and salary>=50000;
end;
begin
counting(total);
if total >= 3
then
dbms_output.put_line('this is acceptable');
else
dbms_output.put_line('this is not acceptable');
end if;
end;
```

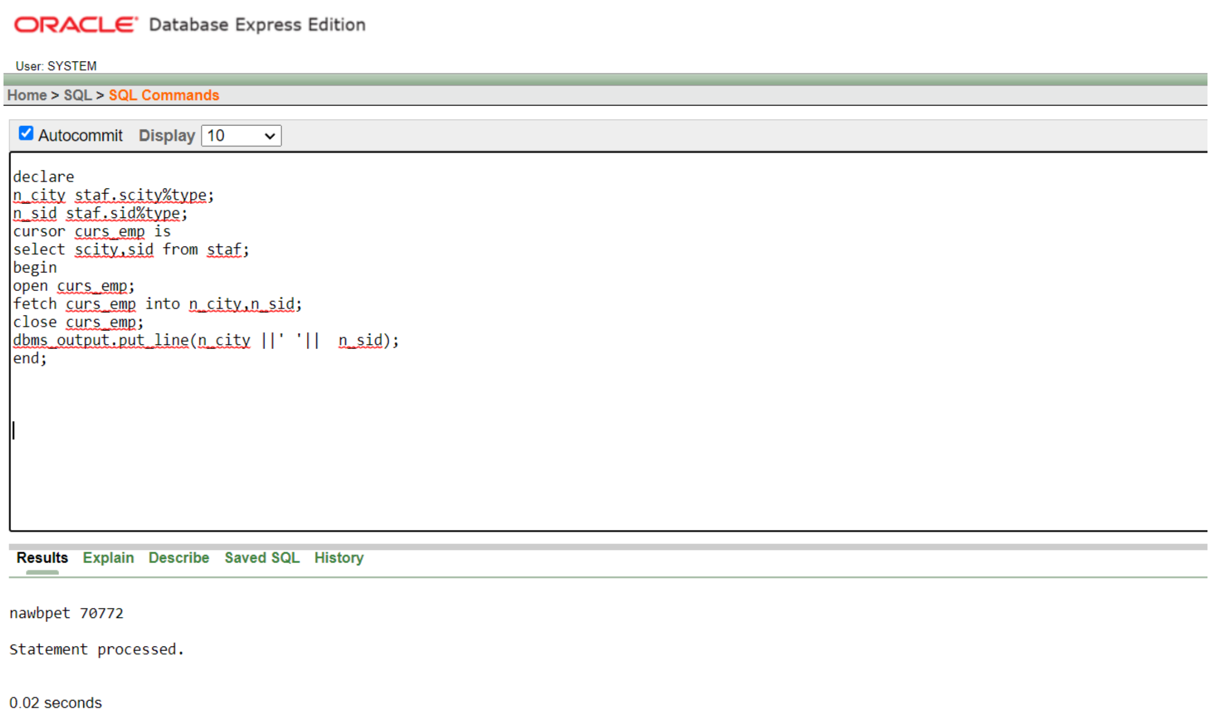

ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
declare
total number(2):=0;
procedure counting(z out number) is
begin
select count(*) into z
from staf
where scity='hyd' and sid<=70777;
end;
begin
counting(total);
if total <= 3
then
dbms_output.put_line('this is acceptable');
else
dbms_output.put_line('this is not acceptable');
end if;
end;
```

```
this is acceptable

Statement processed.

0.00 seconds
```

Example -4

```
declare
n_town staff.town%type;
n_salary staff.salary%type;
cursor curs_emp is
select town,salary from staff;
begin
open curs_emp;
fetch curs_emp into n_town,n_salary;
close curs_emp;
dbms_output.put_line(n_town ||  n_salary);
end;
```


ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
declare
n_city staf.scity%type;
n_sid staf.sid%type;
cursor curs_emp is
select scity,sid from staf;
begin
open curs_emp;
fetch curs_emp into n_city,n_sid;
close curs_emp;
dbms_output.put_line(n_city ||' '||  n_sid);
end;
```

```
nawbpet 70772

Statement processed.

0.02 seconds
```

<u>Example</u> -5

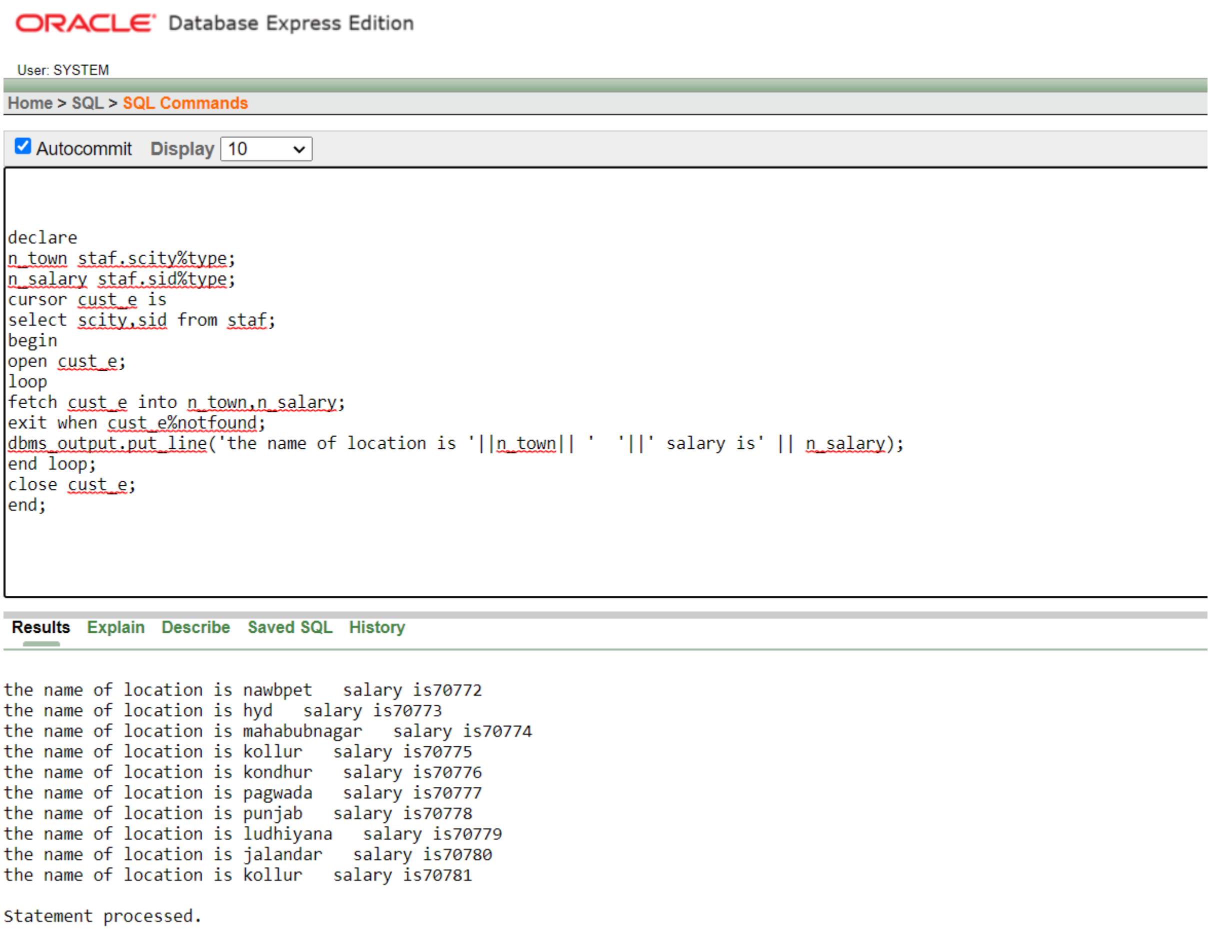
```
declare
n_town staff.town%type;
n_salary staff.salary%type;
cursor cust_e is
select town,salary from staff;
begin
open cust_e;
loop
fetch cust_e into n_town,n_salary;
exit when cust_e%notfound;
```

dbms_output.put_line('the name of location is '||n_town|| ' salary is' || n_salary);

end loop;

close cust_e;

end;


ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

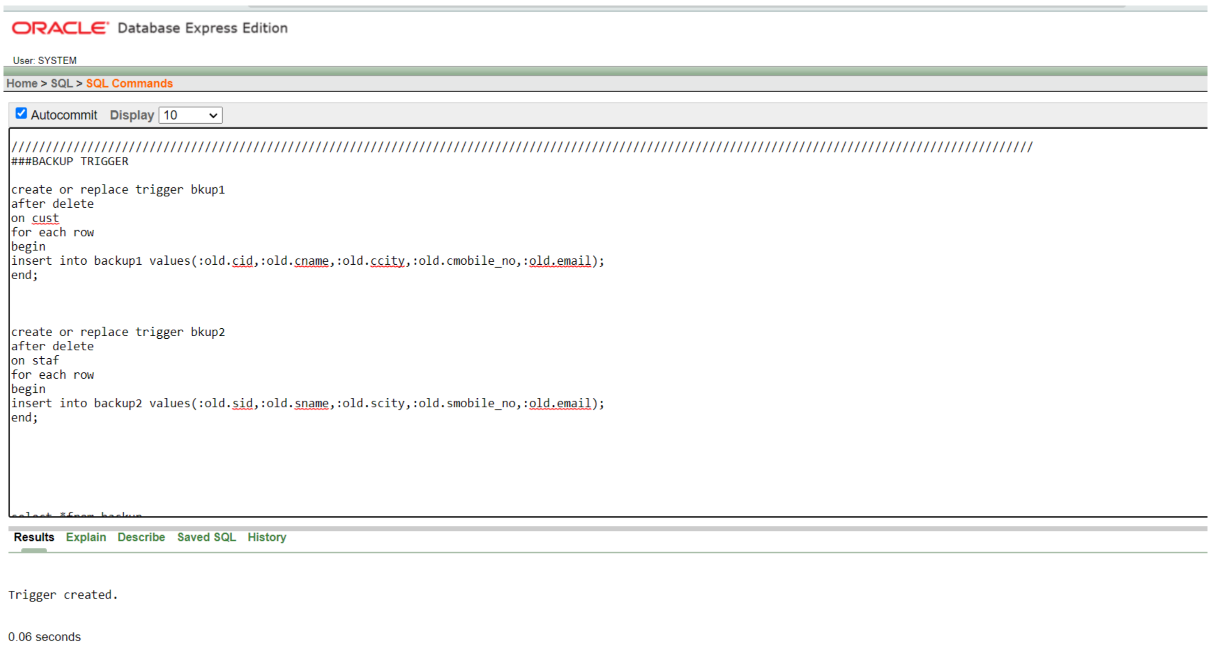
```
declare
n_town staf.scity%type;
n_salary staf.sid%type;
cursor cust_e is
select scity,sid from staf;
begin
open cust_e;
loop
fetch cust_e into n_town,n_salary;
exit when cust_e%notfound;
dbms_output.put_line('the name of location is '||n_town|| '  '||' salary is' || n_salary);
end loop;
close cust_e;
end;
```

```
the name of location is nawbpet   salary is70772
the name of location is hyd   salary is70773
the name of location is mahabubnagar   salary is70774
the name of location is kollur   salary is70775
the name of location is kondhur   salary is70776
the name of location is pagwada   salary is70777
the name of location is punjab   salary is70778
the name of location is ludhiyana   salary is70779
the name of location is jalandar   salary is70780
the name of location is kollur   salary is70781

Statement processed.
```

Example :6

>>> CREATING TRIGGER FOR BACKUP

```
create or replace trigger bkup1
after delete
on cust
for each row
begin
insert into backup1
values(:old.cid,:old.cname,:old.ccity,:old.cmobile_no,:old.em
ail);
end;
```


ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
###BACKUP TRIGGER

create or replace trigger bkup1
after delete
on cust
for each row
begin
insert into backup1 values(:old.cid,:old.cname,:old.ccity,:old.cmobile_no,:old.email);
end;


create or replace trigger bkup2
after delete
on staf
for each row
begin
insert into backup2 values(:old.sid,:old.sname,:old.scity,:old.smobile_no,:old.email);
end;




select *from backup
```

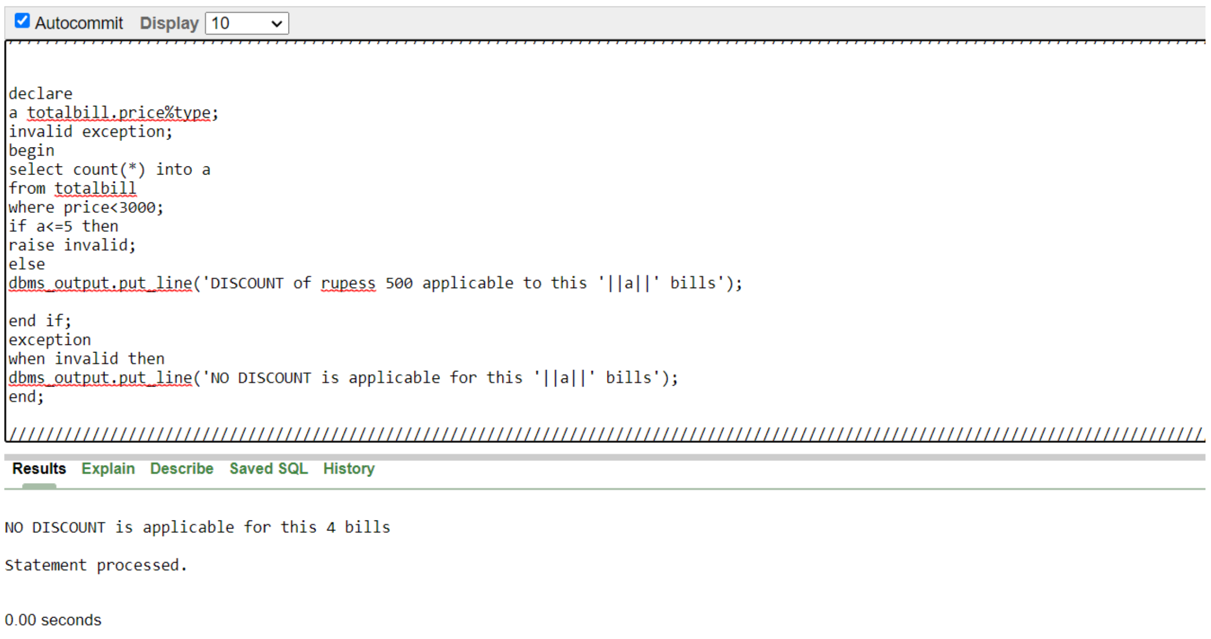
```
Trigger created.
```

0.06 seconds

EXAMPLE:-7

>>>FOR FINDING NO. OF BILLS APPLICABLE FOR DISCOUNT

```
declare
a totalbill.price%type;
invalid exception;
begin
select count(*) into a
from totalbill
where price<3000;
if a<=5 then
raise invalid;
else
dbms_output.put_line('DISCOUNT of rupess 500 applicable
to this '||a||' bills');

end if;
exception
when invalid then
dbms_output.put_line('NO DISCOUNT is applicable for this
'||a||' bills');
end;
```



```
declare
a totalbill.price%type;
invalid exception;
begin
select count(*) into a
from totalbill
where price<3000;
if a<=5 then
raise invalid;
else
dbms_output.put_line('DISCOUNT of rupess 500 applicable to this '||a||' bills');

end if;
exception
when invalid then
dbms_output.put_line('NO DISCOUNT is applicable for this '||a||' bills');
end;
```

```
NO DISCOUNT is applicable for this 4 bills

Statement processed.
```

0.00 seconds

CONCLUSION

The project as a whole describes the scope and viability of the Trading industry and mainly of the financial, technical and its market potential. The project guarantee sufficient fund to repay the loan and also give a good return on capital investment.When analyzing the social- economic impact, this project is able to generate an employment of 5 and above. It will cater the demand of Trading and thus helps the other business entities to increase the production and service which provide service and support to this industry. Thus more cyclic employment and livelihood generation. So in all ways, we can conclude the project is technically and socially viable and commercially sound too…

--- DONE ---