

A Project Report on  
**COMPUTER SERVICE CENTRE**

**SUBMITTED BY**

**NAME: PRATHAM THAKUR**

**REGISTRATION NO: 12114778**

**ROLL NO: B62**

**SETION: K21MR**

Under the guidance of

**BHANU TALWAR**

**Dept. of CSE**



**L** OVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

---

*Transforming Education Transforming India*

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,  
LOVELY PROFESIONAL UNIVERSITY  
PHAGWARA, PUNJAB**

# CERTIFICATE

This is to certify that **PRATHAM THAKUR** bearing Registration no. **12114778** have completed INT306 project titled, **COMPUTER SERVICE CENTER DATABASE PROJECT** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original development, effort, and study.

BHANU TALWAR

Department of Computer Science and Engineering  
School of Computer Science and Engineering  
Lovely Professional University

Phagwara , Punjab

Date: 20-11-2022

# **DECLARATION**

I, **PRATHAM THAKUR** student of B.TECH CSE under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

PRATHAM THAKUR

12114778

# TABLE OF CONTENT

- Abstract
- Schema and Normalisation
- ER Diagram
- About SQL and PL/SQL used in the project
- Implementation
- Conclusion

## **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.

# Schema and Normalisation

## 2.1 DESIGN

### 2.1.1 E-R Diagram

### **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 details of the client. It includes all the information of the client which includes telephone number and Email id also...

Cust\_ID numeric(9),

Fname varchar(100) NOT NULL,

Town varchar(30) NOT NULL,

Telephone varchar(15) NOT NULL,

Email varchar(500) NOT NULL,

## **2. Staff Details:**

(Staff\_ID numeric, Forename varchar, Town varchar, Telephone char, Email varchar)

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

Staff\_ID numeric(9),  
Forename varchar(100) NOT NULL,  
Town varchar(30) NOT NULL,  
Telephone char(15) NOT NULL,  
Email varchar(500) NOT NULL,

## **3. Repairing Details**

( Rep\_ID numeric, Cust\_ID numeric, Staff\_ID numeric, Description varchar, Brand varchar)

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..

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,

## **4. Totalprice Details :**

( Rep\_ID numeric, Staff\_ID numeric, GST decimal, Discount decimal, Total decimal)

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),

## 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:**

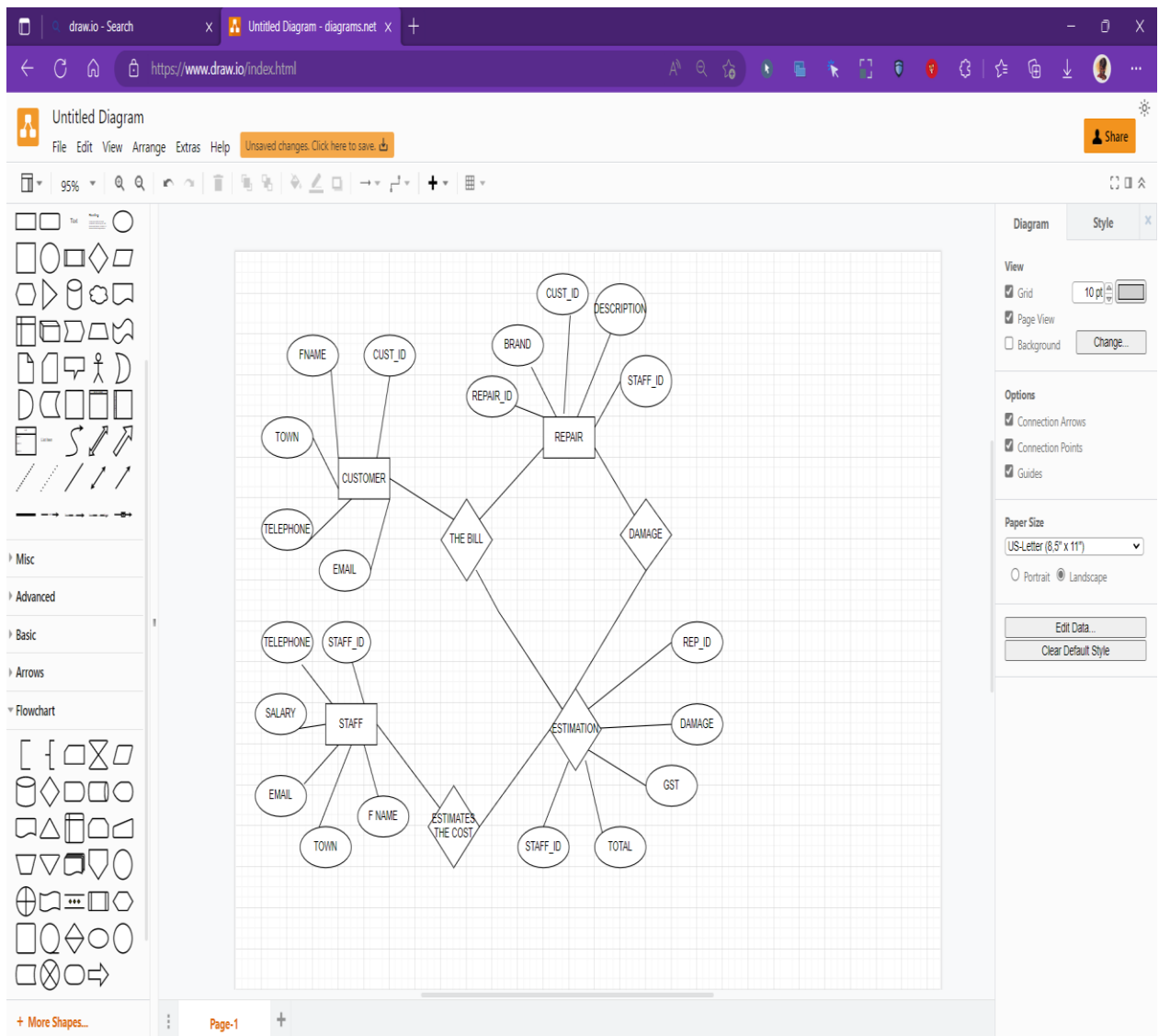
A customer can pay many bills through this Customer ID

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"



### 3. SQL&PLSQL

#### 3.1 SQL Table Creation

The Implementation of SQL server :

```
CREATE TABLE Customers (  
    Cust_ID numeric(9),  
    Fname varchar(100) NOT NULL,  
    Town varchar(30) NOT NULL,  
    Telephone varchar(15) NOT NULL,  
    Email varchar(500) NOT NULL,  
    CONSTRAINT uq_Cust_email UNIQUE (Email),  
    CONSTRAINT uq_Cust_tel UNIQUE (Telephone),  
    CONSTRAINT pk_Customers PRIMARY KEY (Cust_ID));
```

```
CREATE TABLE Staff (  
    Staff_ID numeric(9),  
    Forename varchar(100) NOT NULL,  
    Town varchar(30) NOT NULL,  
    Telephone char(15) NOT NULL,  
    Email varchar(500) NOT NULL,  
    Salary NUMBER NOT NULL,  
    CONSTRAINT uq_Staff_email UNIQUE (Email),  
    CONSTRAINT uq_Staff_tel UNIQUE (Telephone),  
    CONSTRAINT pk_Staff PRIMARY KEY (Staff_ID))
```

```
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 Estimates (  
    Rep_ID numeric(9) NOT NULL,  
    Staff_ID numeric(9) NOT NULL,  
    GST decimal(4,2),  
    Discount decimal(4,2),  
    Total 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

INSERT INTO Customers VALUES

(1,'Bhargav','Tiruvur','9876543210','bhargav@gmail.com');

INSERT INTO Customers VALUES

(2,'Venku','Armoor','9876543211','venku@gmail.com');

INSERT INTO Customers VALUES

(3,'Naveen','Aluru','9876543212','naveen@gmail.com');

INSERT INTO Customers VALUES

(4,'Pramod','Peddapalli','9876543213','pramod@gmail.com');

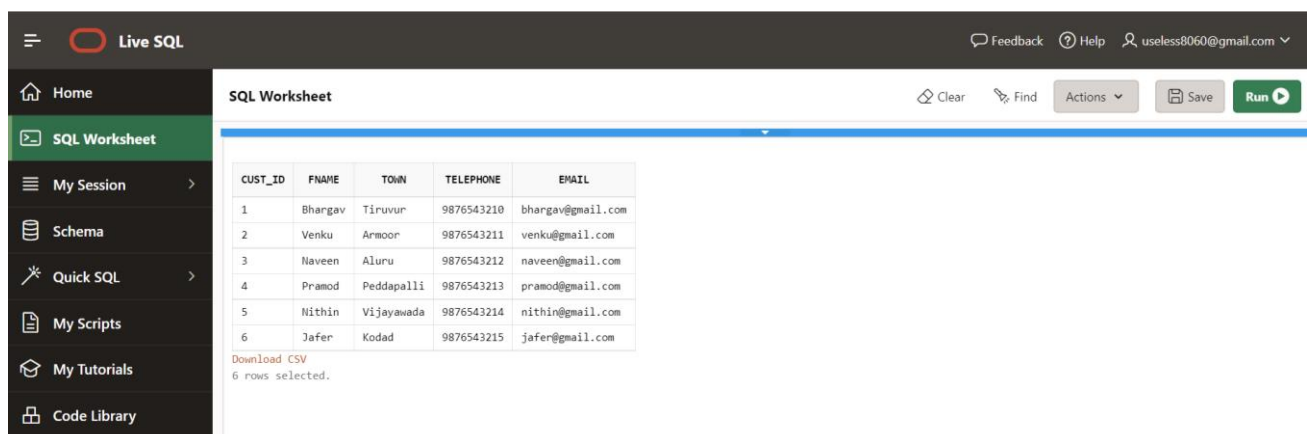
INSERT INTO Customers VALUES

(5,'Nithin','Vijayawada','9876543214','nithin@gmail.com');

INSERT INTO Customers VALUES

(6,'Jafer','Kodad','9876543215','jafer@gmail.com');

select \*from Customer



Live SQL

Feedback Help useless8060@gmail.com

Home SQL Worksheet Clear Find Actions Save Run

CUST_ID	FNAME	TOWN	TELEPHONE	EMAIL
1	Bhargav	Tiruvur	9876543210	bhargav@mail.com
2	Venku	Armoor	9876543211	venku@gmail.com
3	Naveen	Aluru	9876543212	naveen@gmail.com
4	Pramod	Peddapalli	9876543213	pramod@gmail.com
5	Nithin	Vijayawada	9876543214	nithin@mail.com
6	Jafer	Kodad	9876543215	jafer@gmail.com

Download CSV  
6 rows selected.

```

INSERT INTO Staff VALUES (1,'Kovid','KKG','8976543210','kovid@gmail.com',100000);

INSERT INTO Staff VALUES (2,'Joel','Kerala','8976543215','joel@gmail.com',10000);

INSERT INTO Staff VALUES (3,'Tharun','Tirupati','8976543214','tharun@gmail.com',20000);

INSERT INTO Staff VALUES (4,'Koti','KP','8976543213','koti@gmail.com',5000);

INSERT INTO Staff VALUES (5,'Sudarshan','Kadapa','8976543212','sudha@gmail.com',50000);

INSERT INTO Staff VALUES (6,'Arun','ARG','8976543211','arun@gmail.com',18000);

INSERT INTO Staff VALUES (7,'Kamala','Sobbla','8976543221','kamala@gmail.com',80000);

INSERT INTO Staff VALUES (8,'Ramya','Khammam','8976543231','ramya@gmail.com',88000);

INSERT INTO Staff VALUES (9,'Jaya','Wyra','8976543241','jaya@gmail.com',48000);

INSERT INTO Staff VALUES (10,'Kusuma','Guntur','8976543251','kusuma@gmail.com',58000);

INSERT INTO Staff VALUES (11,'Deepthi','Suryapet','8976543261','deepthi@gmail.com',98000);

INSERT INTO Staff VALUES (12,'Sahithi','Thammara','8976543271','sahithi@gmail.com',108000);

INSERT INTO Staff VALUES (13,'Satish','Chennai','8976543281','satish@gmail.com',38000);

INSERT INTO Staff VALUES (14,'Pavani','Eluru','8976543291','pavani@gmail.com',28000);

INSERT INTO Staff VALUES (15,'Jaswanth','Nedincharla','8976543311','jaswanth@gmail.com',78000);

```

select \*from Staff

Home	SQL Worksheet					Clear	Find	Actions	Save	Run
SQL Worksheet	STAFF_ID	FORENAME	TOWN	TELEPHONE	EMAIL	SALARY				
My Session	1	Kovid	KKG	8976543210	kovid@gmail.com	100000				
Schema	2	Joel	Kerala	8976543215	joel@gmail.com	10000				
Quick SQL	3	Tharun	Tirupati	8976543214	tharun@gmail.com	20000				
My Scripts	4	Koti	KP	8976543213	koti@gmail.com	5000				
My Tutorials	5	Sudarshan	Kadapa	8976543212	sudha@gmail.com	50000				
Code Library	6	Arun	ARG	8976543211	arun@gmail.com	18000				
	7	Kamala	Sobbla	8976543221	kamala@gmail.com	80000				
	8	Ramya	Khammam	8976543231	ramya@gmail.com	88000				
	9	Jaya	Wyra	8976543241	jaya@gmail.com	48000				
	10	Kusuma	Guntur	8976543251	kusuma@gmail.com	58000				
	11	Deepthi	Suryapet	8976543261	deepthi@gmail.com	98000				
	12	Sahithi	Thammara	8976543271	sahithi@gmail.com	108000				
	13	Satish	Chennai	8976543281	satish@gmail.com	38000				
	14	Pavani	Eluru	8976543291	pavani@gmail.com	28000				
	15	Jaswanth	Nedincharla	8976543311	jaswanth@gmail.com	78000				



INSERT INTO Repairs VALUES (1,1,1,'Virus problem','Sony');

INSERT INTO Repairs VALUES (2,2,2,'Internet is not working','HP');

INSERT INTO Repairs VALUES (3,3,3,'Virus problem','Dell');

INSERT INTO Repairs VALUES (4,4,4,'Pop-up ads are appearing on my desktop','Lenovo');

INSERT INTO Repairs VALUES (5,5,5,'Downloads are taking forever','Acer');

SQL Worksheet

Clear Find Actions Save Run

```
1 INSERT INTO Repairs VALUES (1,1,1,'Virus problem','Sony');
2 INSERT INTO Repairs VALUES (2,2,2,'Internet is not working','HP');
3 INSERT INTO Repairs VALUES (3,3,3,'Virus problem','Dell');
4 INSERT INTO Repairs VALUES (4,4,4,'Pop-up ads are appearing on my desktop','Lenovo');
5 INSERT INTO Repairs VALUES (5,5,5,'Downloads are taking forever','Acer');
```

select \*from Repairs

SQL Worksheet

Clear Find Actions Save Run

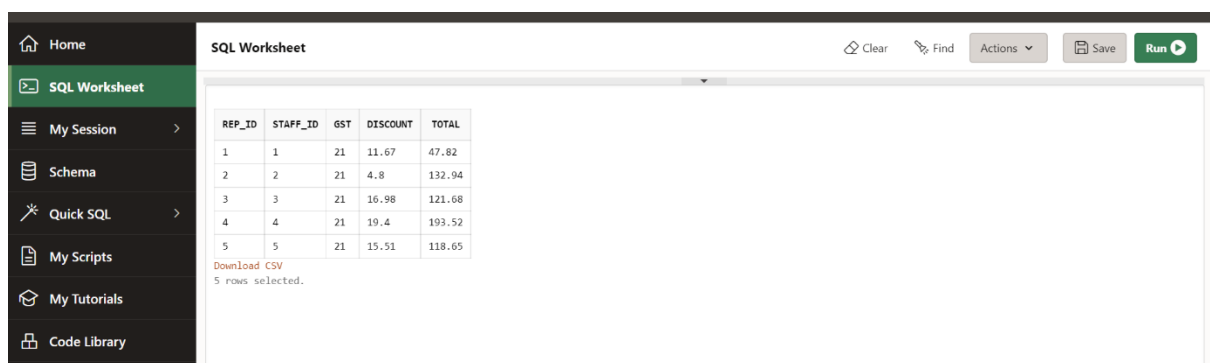
REP_ID	CUST_ID	STAFF_ID	DESCRIPTION	BRAND
1	1	1	Virus problem	Sony
2	2	2	Internet is not working	HP
3	3	3	Virus problem	Dell
4	4	4	Pop-up ads are appearing on my desktop	Lenovo
5	5	5	Downloads are taking forever	Acer

Download CSV  
5 rows selected.

```
INSERT INTO Estimates VALUES (1,1,21,11.67,47.82);
INSERT INTO Estimates VALUES (2,2,21,4.80,132.94);
INSERT INTO Estimates VALUES (3,3,21,16.98,121.68);
INSERT INTO Estimates VALUES (4,4,21,19.40,193.52);
INSERT INTO Estimates VALUES (5,5,21,15.51,118.65);
```



select \*from Estimates



## 3.3 PL/ SQL Examples

### Example -1

declare

total number(2):=0;

begin

select count(\*) into total

from staff;

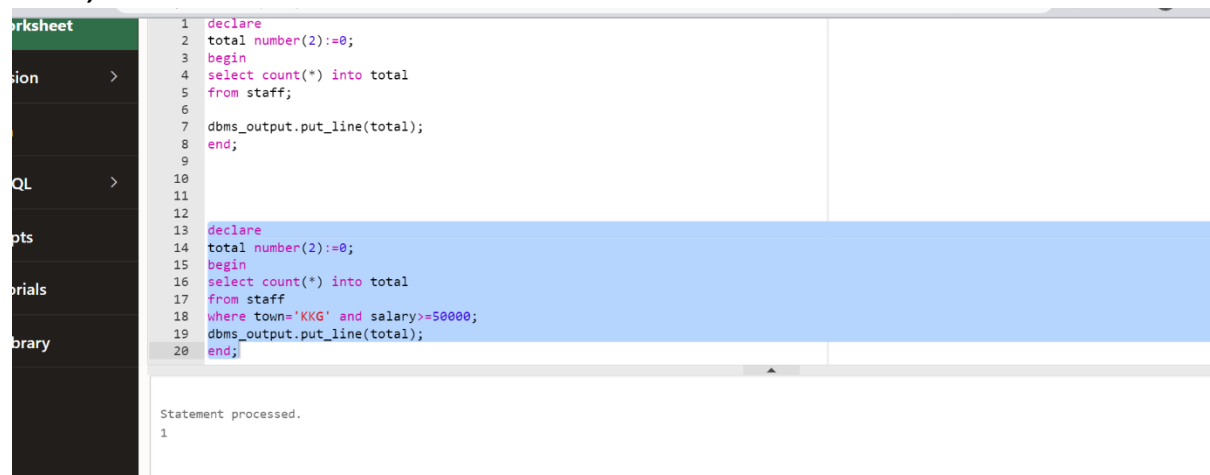
dbms\_output.put\_line(total);

end;



## 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;
```



```
1 declare
2 total number(2):=0;
3 begin
4 select count(*) into total
5 from staff;
6
7 dbms_output.put_line(total);
8 end;
9
10
11
12
13 declare
14 total number(2):=0;
15 begin
16 select count(*) into total
17 from staff
18 where town='KKG' and salary>=50000;
19 dbms_output.put_line(total);
20 end;
```

Statement processed.

1

## 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;

```

The screenshot shows an SQL Worksheet interface. On the left is a dark sidebar with navigation links: 'Worksheet', 'Session', 'a', 'SQL', 'Scripts', 'Tutorials', and 'Library'. The main area is titled 'SQL Worksheet' and contains a PL/SQL procedure named 'counting'. The procedure declares a variable 'total' of type 'number(2)' and initializes it to 0. It then defines a procedure 'counting(z out number)' which selects the count of staff members from the 'staff' table where the town is 'KKG' and the salary is greater than or equal to 50000. The main procedure calls 'counting(total)', and based on the value of 'total', it prints a message to the output. In this case, the output shows 'this is not acceptable'. The interface includes a toolbar with 'Clear', 'Find', 'Actions', and 'Save' buttons. A status bar at the bottom indicates 'Statement processed.' and 'this is not acceptable'.

```

24
25
26 declare
27 total number(2):=0;
28 procedure counting(z out number) is
29 begin
30 select count(*) into z
31 from staff
32 where town='KKG' and salary>=50000;
33 end;
34 begin
35 counting(total);
36 if total >= 3
37 then
38 dbms_output.put_line('this is acceptable');
39 else
40 dbms_output.put_line('this is not acceptable');
41 end if;
42 end;
43

```

Statement processed.  
this is not acceptable

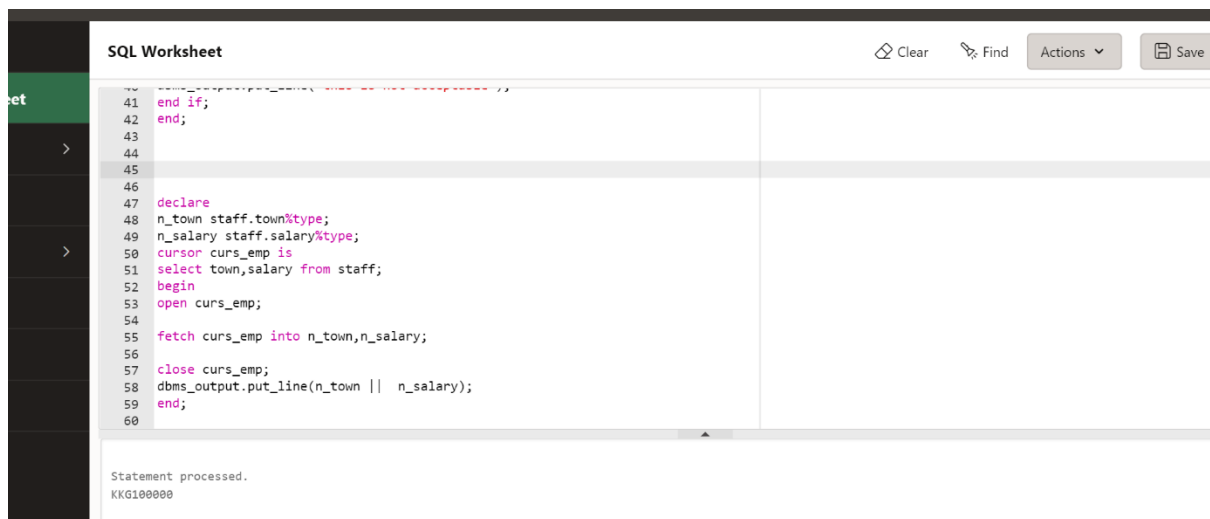
## Example -4

```

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;
```



The screenshot shows an SQL Worksheet interface. The main area contains a PL/SQL script with line numbers 41 to 60. The script defines variables `n_town` and `n_salary` of type `staff.town%type` and `staff.salary%type` respectively, declares a cursor `curs_emp` for the query `select town,salary from staff;`, and then opens, fetches, and closes the cursor. Finally, it prints the concatenated values of `n_town` and `n_salary` using `dbms_output.put_line`. The interface includes a 'Clear' button, a 'Find' button, an 'Actions' dropdown, and a 'Save' button. Below the script, a status message indicates 'Statement processed.' followed by the output 'KKG100000'.

```
41 end if;
42 end;
43
44
45
46
47 declare
48 n_town staff.town%type;
49 n_salary staff.salary%type;
50 cursor curs_emp is
51 select town,salary from staff;
52 begin
53 open curs_emp;
54
55 fetch curs_emp into n_town,n_salary;
56
57 close curs_emp;
58 dbms_output.put_line(n_town || n_salary);
59 end;
60
```

Statement processed.  
KKG100000

## Example -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;
```

The screenshot displays the Oracle APEX SQL Worksheet interface. On the left is a navigation sidebar with options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a PL/SQL script. The script declares two variables, opens a cursor, enters a loop to fetch data from the 'staff' table, and prints the location name and salary for each row. The output at the bottom shows the results of the script execution for six different locations.

```
62
63 declare
64   n_town staff.town%type;
65   n_salary staff.salary%type;
66
67 cursor cust_e is
68   select town,salary from staff;
69 begin
70   open cust_e;
71   loop
72     fetch cust_e into n_town,n_salary;
73     exit when cust_e%notfound;
74     dbms_output.put_line('the name of location is '||n_town|| ' salary is' || n_salary);
75   end loop;
76   close cust_e;
77 end;
```

the name of location is KKG salary is100000  
the name of location is Kerala salary is100000  
the name of location is Tirupati salary is200000  
the name of location is KP salary is5000  
the name of location is Kadapa salary is500000  
the name of location is ARG salary is180000  
the name of location is Sobbla salary is800000

© 2022 Oracle - Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ♥ using Oracle APEX - Privacy - Terms of Use

## **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...

**--- THANK YOU---**