JAVA SWINGS BASED- DAIRY MANAGEMENT SYSTEM

- SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

Joshua Peter<1602-19-737-075>

Under the guidance of Ms B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘**DAIRY MANAGEMENT SYSTEM**’ is a project work of Mr. Joshua Peter bearing roll no. 1602-19-737-075 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature   Signature

Internal Examiner   External Examiner

ABSTRACT:

The most popular thing now-a-days is Digital marketing. Which is also known as online or internet marketing. Dairy Product Management System is a console-based application which allows a customer to place online orders for the dairy products they need through an online store.

User can login and they can buy these products online. Customers can also maintain their account and their transactions. Here the user can search for the product they need. The application provides information on the availability of this product. The customer can order what he needs and simply get it (The customer can select the company as well as the Category they need). They can also check the status of your order. As soon as the customer decides to buy a certain product/item, they can place an order. It can also print the billing details of the transaction.

REQUIREMENT ANALYSIS:

List of Tables:

- Login
- Customer
- Orders_From
- Company
- Selects
- Category
- Orders
- Details
- Customer_login

List of attributes with their domain types:

Login:

- Login_id: number(5)
- Name: varchar2(20)
- Password: varchar2(20)

Customer:

- Customer_id: number(5)
- Customer_name: varchar2(20)
- phone_no: number(10)
- email: varchar2(20)
- address: varchar2(20)

orders_from:

- Customer_id: number(5)
- Company_id: number(5)
- Date: DATE

Company:

- Company_id: number(5)
- Company_name: varchar2(20)

Selects:

- company_id: number(5)
- category_id: number(5)

Category:

- Category_id: number(5)
- Category_name: varchar(20)
- Price: number(5)
- code: varchar(20)
- Availability: varchar2(20)

Orders:

- Customer_id: number(5)
- Date: DATE
- Order_id: number(5)

Details:

- Order_id: number(5)
- Total_amt: number(10)
- status: varchar2(20)

Customer_login:

- Login_id: number(5)
- Customer_id: number(5)
- day: DATE

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based dairy management system which takes the values like: user id, password for login and stores details of all users, admin and their details. It also stores the details of the company and products and their availability. These values are to be updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY

Software used: Java Eclipse, Oracle 11g Database, Java SE version 13, SQL*Plus.

Java SWINGS:

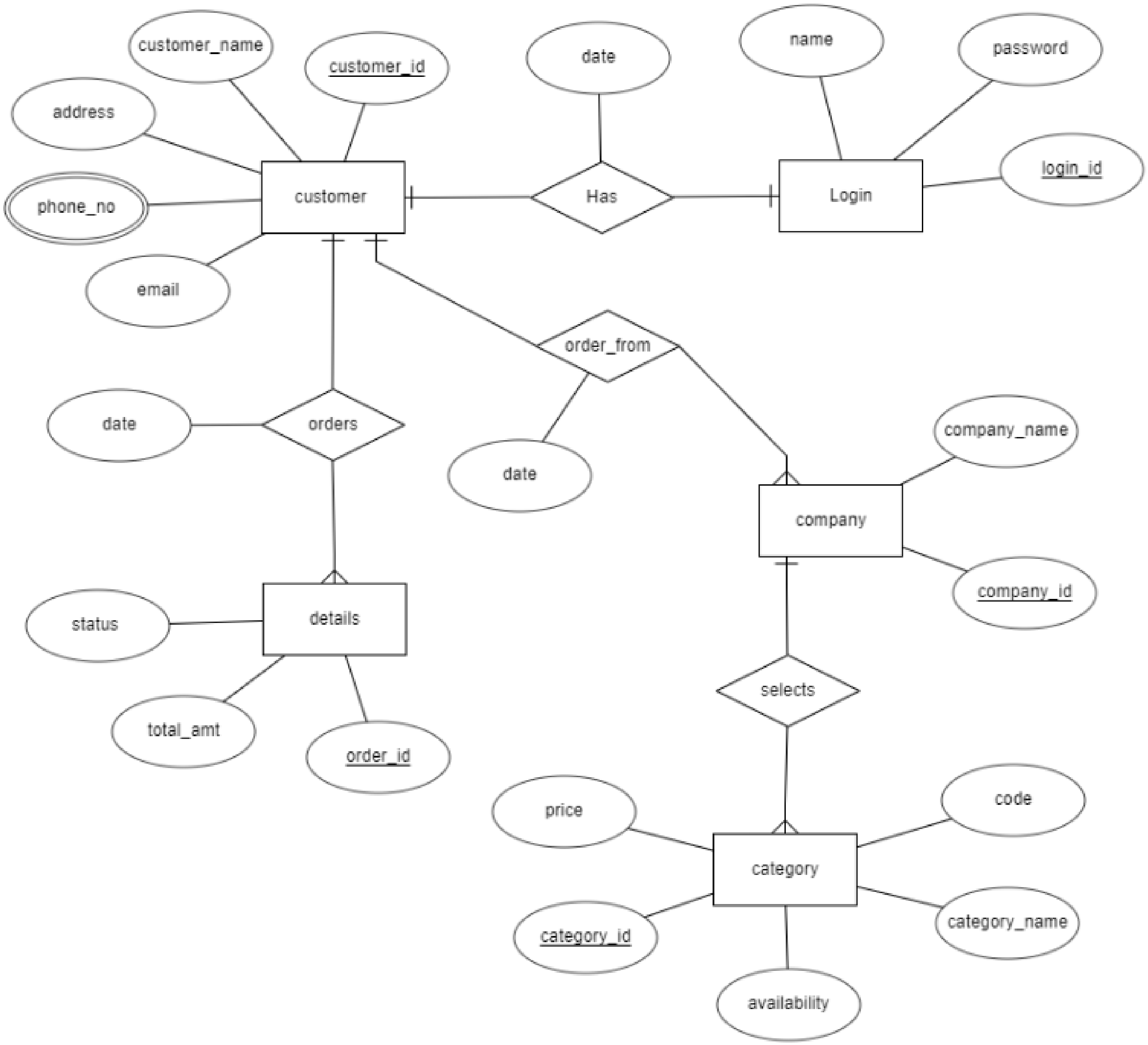
Java SWINGS is an API to develop GUI or window-based applications in java. Java SWING components are platform-independent. It is lightweight. The javax.swing package provides classes for SWING API such as JTextField, JLabel, JTextArea, JRadioButton, JCheckBox, JChoice, JList etc.

SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

DESIGN:

ER DIAGRAM:



Mapping Cardinality and participation constraints:

A customer can have a single login_id. Therefore, one to one mapping cardinality between login and customer account.

A customer can order from any number of companies and single company can be searched by any number of users. So, one to many mapping cardinality between customer and company.

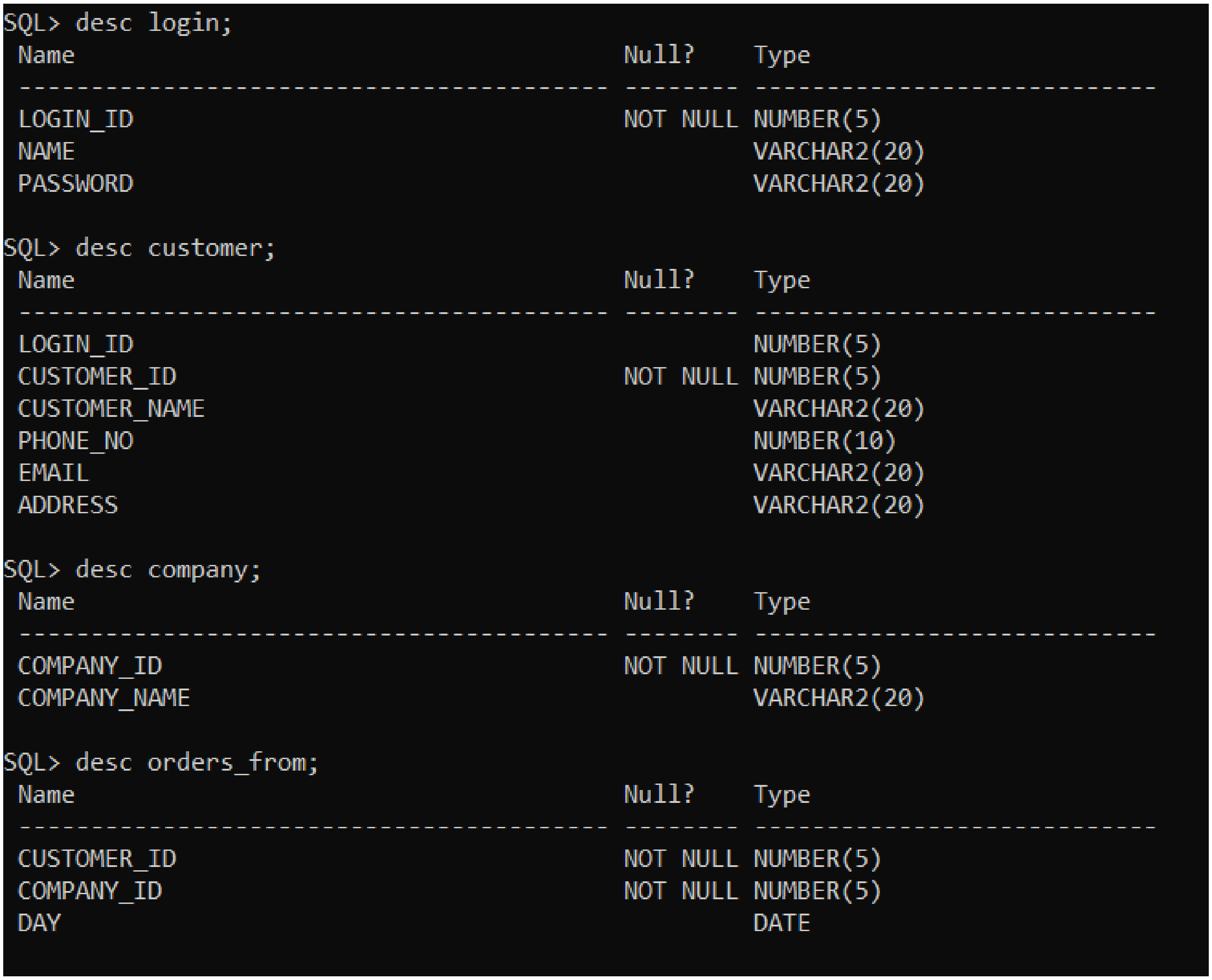
A single company can contain any number of Categories. Therefore, one to many mapping cardinality between company and Category.

A single customer can have any number of order details but all orders of one user are considered as one. So, one to many mapping cardinality between customer and details of the order.

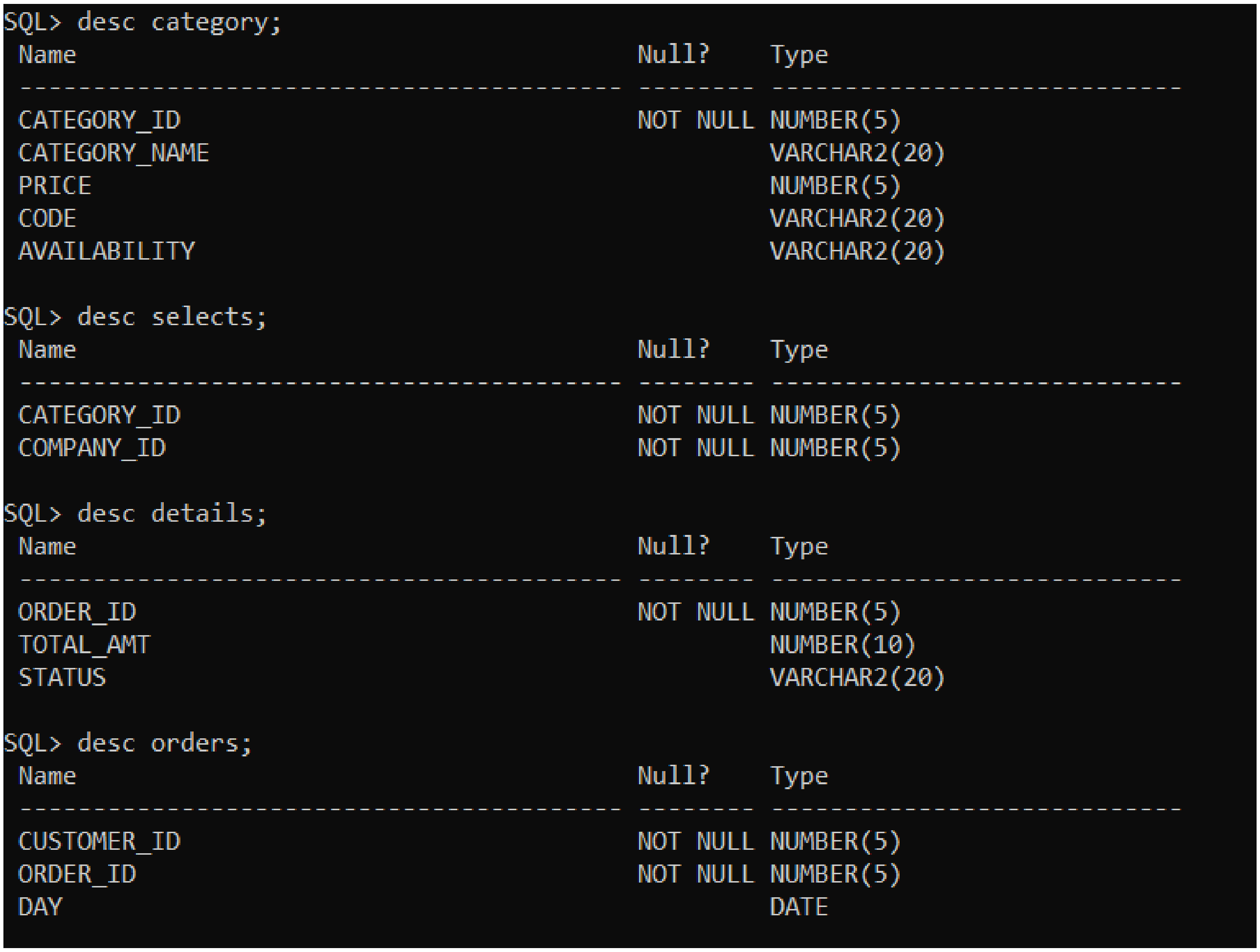
DDL COMMANDS:

```
create table login(login_id number(5), name varchar2(20),
   password varchar2(20), primary key (login_id));

create table customer_login(login_id number(5), Customer_id number(5),
  day date, primary key(login_id,customer_id),
 foreign key(login_id) references login(login_id),
foreign key(Customer_id) references customer(Customer_id));

create table customer (Customer_id number(5),
  customer_name varchar2(20), phone_no number(10),
  email varchar2(20), address varchar2(20), primary key (Customer_id));

create table Company( Company_id number(5), Company_name varchar2(20),
  primary key(Company_id));

create table orders_from( Customer_id number(5), Company_id number(5),
  day date, primary key(Customer_id,Company_id),
  foreign key(Customer_id) references customer(Customer_id),
  foreign key(Company_id) references company(Company_id));

create table Category(Category_id number(5), Category_name varchar2(20),
  Price number(5), Code varchar2(20), Availability varchar2(20),
  primary key(Category_id));
```

```
create table selects(Category_id number(5),Company_id number(5),
  primary key(Company_id,Category_id), day date,
  foreign key(Company_id) references company(Company_id),
  foreign key(Category_id) references Category(Category_id));



create table Details(order_id number(5), total_amt number(10),
 status varchar2(20), primary key(order_id));

create table Orders(Customer_id number(5), order_id number(5),
  day date, primary key(Customer_id,order_id),
  foreign key(Customer_id) references customer(Customer_id),
  foreign key(order_id) references Details(order_id));
```



```
SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOGIN_ID                                  NOT NULL NUMBER(5)
 NAME                                               VARCHAR2(20)
 PASSWORD                                           VARCHAR2(20)

SQL> desc customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOGIN_ID                                           NUMBER(5)
 CUSTOMER_ID                               NOT NULL NUMBER(5)
 CUSTOMER_NAME                                      VARCHAR2(20)
 PHONE_NO                                           NUMBER(10)
 EMAIL                                              VARCHAR2(20)
 ADDRESS                                            VARCHAR2(20)

SQL> desc company;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COMPANY_ID                                NOT NULL NUMBER(5)
 COMPANY_NAME                                       VARCHAR2(20)

SQL> desc orders_from;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(5)
 COMPANY_ID                                NOT NULL NUMBER(5)
 DAY                                                DATE
```



```
SQL> desc category;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATEGORY_ID                               NOT NULL NUMBER(5)
 CATEGORY_NAME                                      VARCHAR2(20)
 PRICE                                              NUMBER(5)
 CODE                                               VARCHAR2(20)
 AVAILABILITY                                       VARCHAR2(20)

SQL> desc selects;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATEGORY_ID                               NOT NULL NUMBER(5)
 COMPANY_ID                                NOT NULL NUMBER(5)

SQL> desc details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_ID                                  NOT NULL NUMBER(5)
 TOTAL_AMT                                          NUMBER(10)
 STATUS                                             VARCHAR2(20)

SQL> desc orders;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(5)
 ORDER_ID                                  NOT NULL NUMBER(5)
 DAY                                                DATE
```

DML COMMANDS

insert into login values(&login_id,’&name’,’&password’);

insert into customer_login values(&login_id,&customer_id,&day);

insert into customer values(&customer_id,’&customer_name’,&phone_no,’&email’,’&address’);

insert into company values(&company_id,’&company_name’);

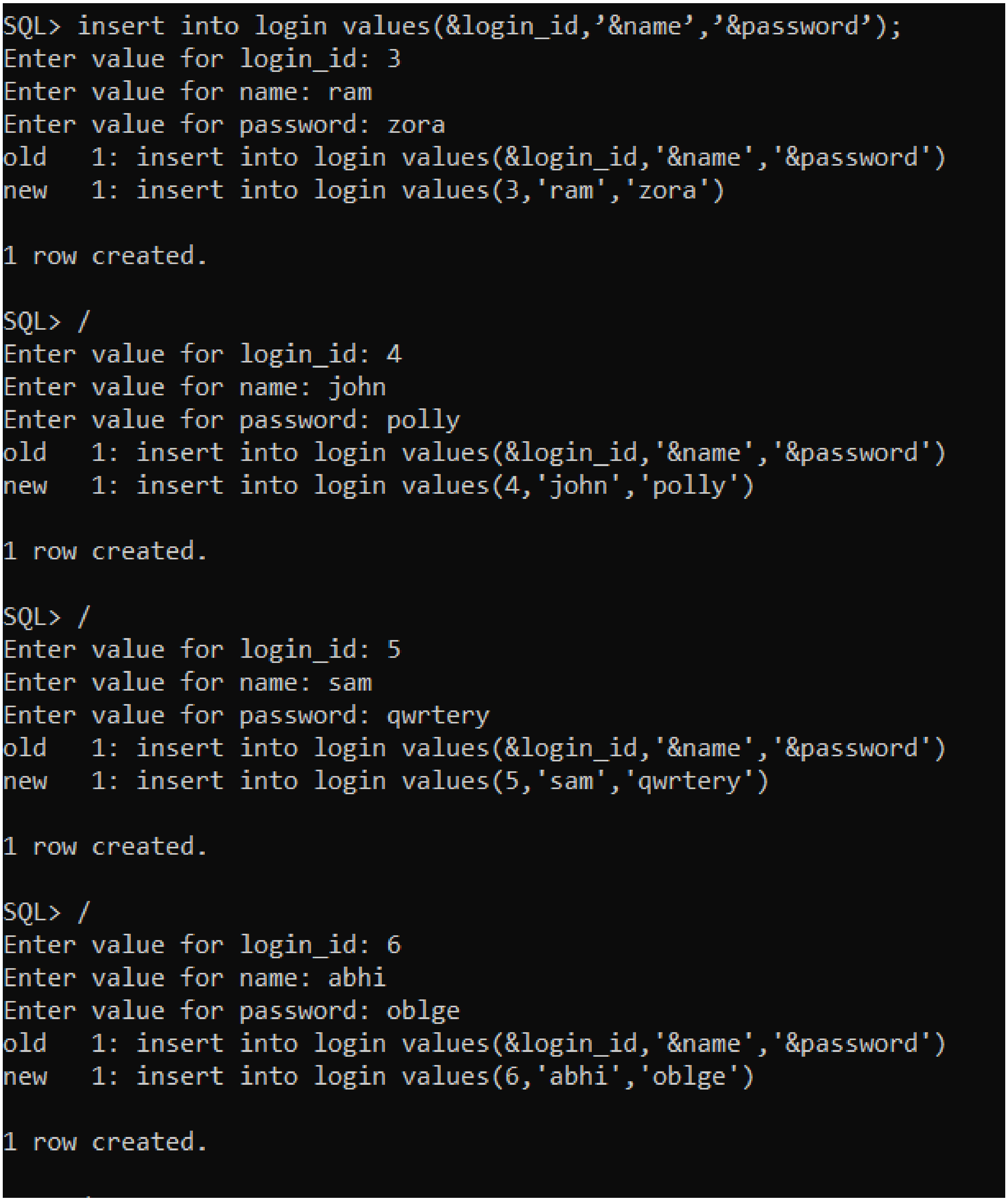
insert into category values(&category_id, ’&category_name’, &price, ’&code’,’&availability’);

insert into orders_from values(&customer_id,&company_id,&day);

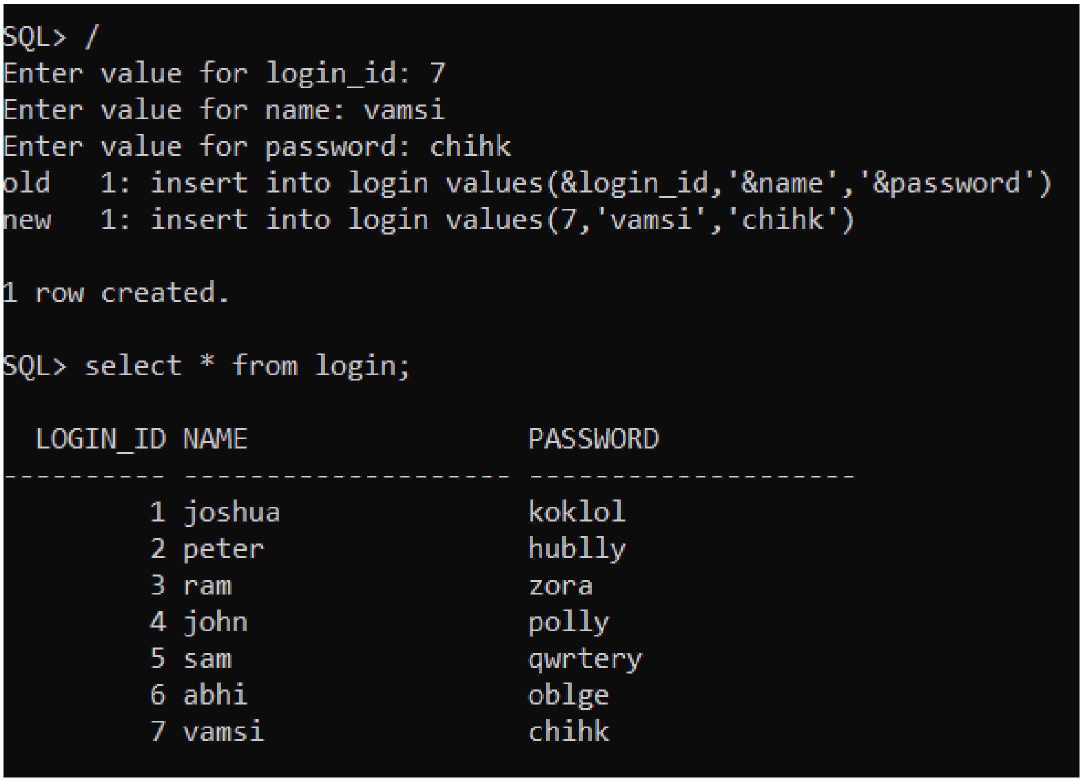
insert into details values(&order_id,&total_amt,’&status’);

insert into selects values(&category_id, &company_id);

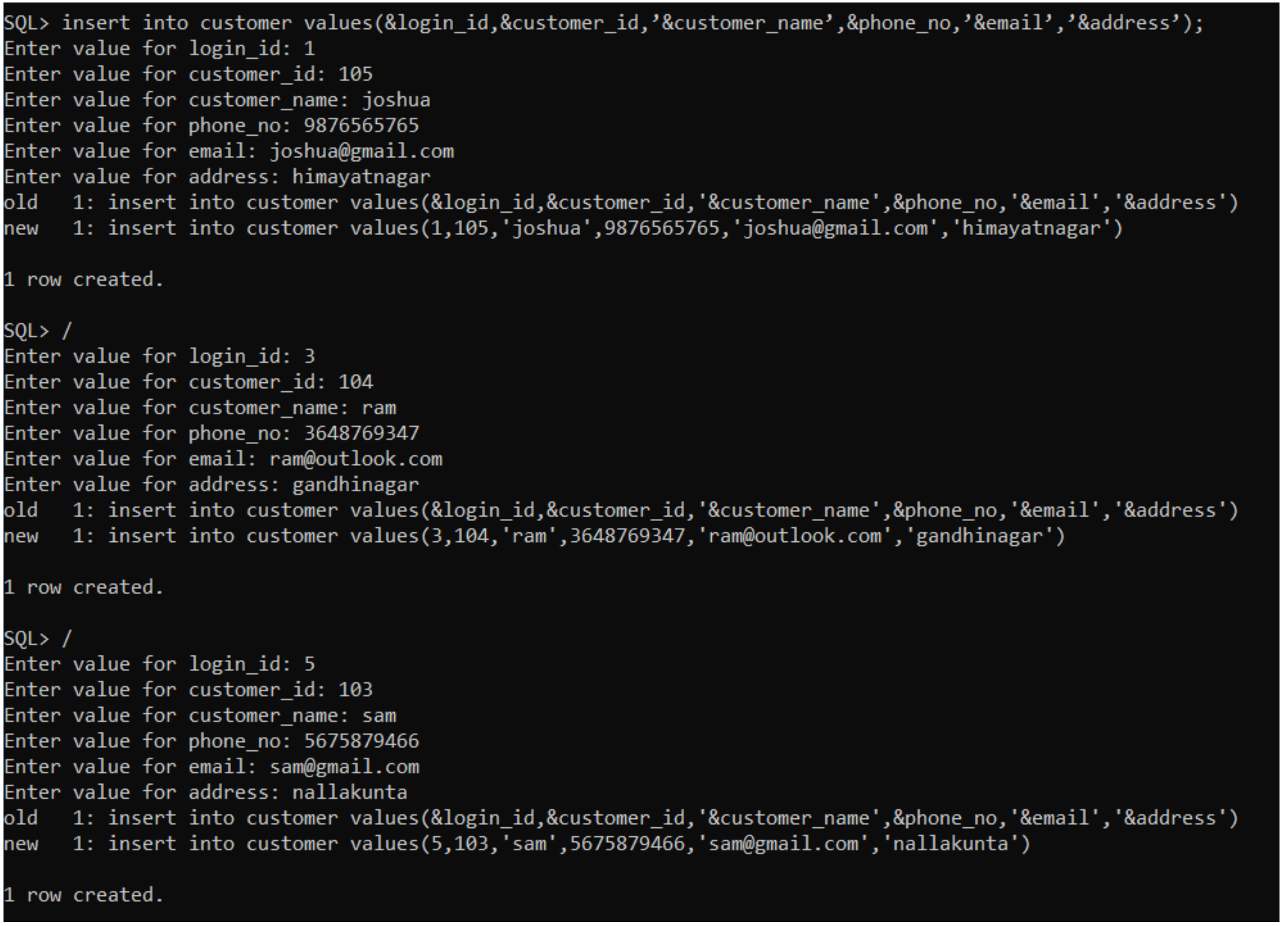
insert into orders values(&customer_id,&order_id,&day);



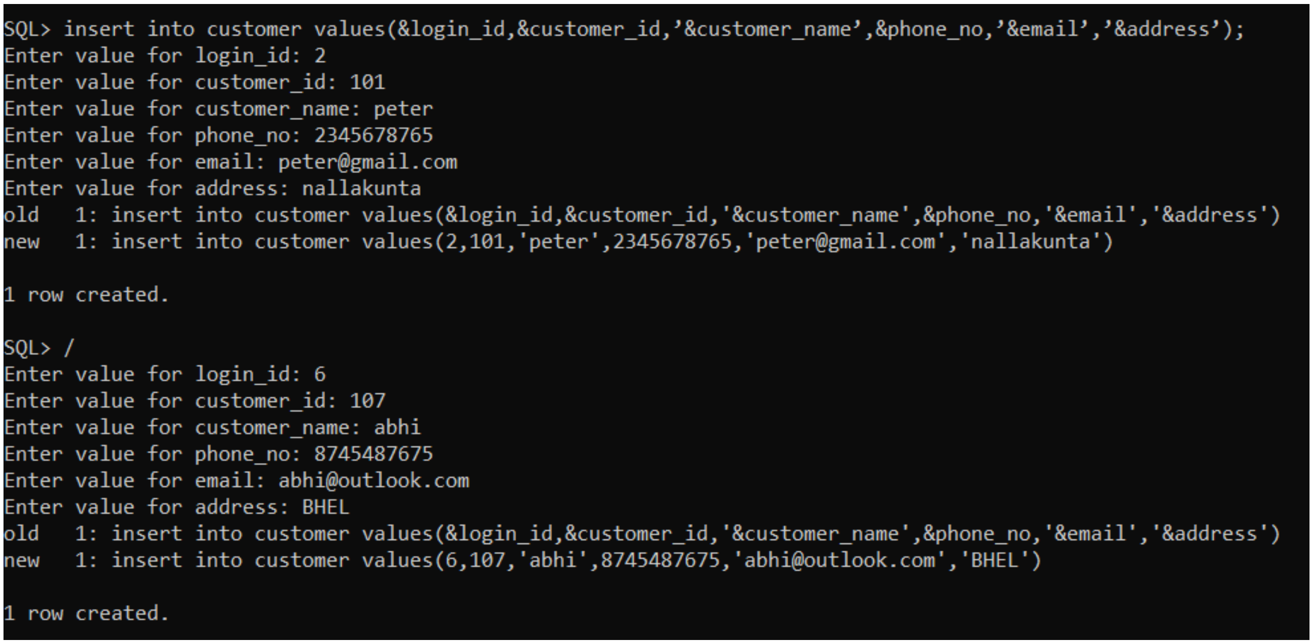
```
SQL> insert into login values(&login_id,’&name’,’&password’);
Enter value for login_id: 3
Enter value for name: ram
Enter value for password: zora
old   1: insert into login values(&login_id,'&name','&password')
new   1: insert into login values(3,'ram','zora')

1 row created.

SQL> /
Enter value for login_id: 4
Enter value for name: john
Enter value for password: polly
old   1: insert into login values(&login_id,'&name','&password')
new   1: insert into login values(4,'john','polly')

1 row created.

SQL> /
Enter value for login_id: 5
Enter value for name: sam
Enter value for password: qwrtery
old   1: insert into login values(&login_id,'&name','&password')
new   1: insert into login values(5,'sam','qwrtery')

1 row created.

SQL> /
Enter value for login_id: 6
Enter value for name: abhi
Enter value for password: oblge
old   1: insert into login values(&login_id,'&name','&password')
new   1: insert into login values(6,'abhi','oblge')

1 row created.
```



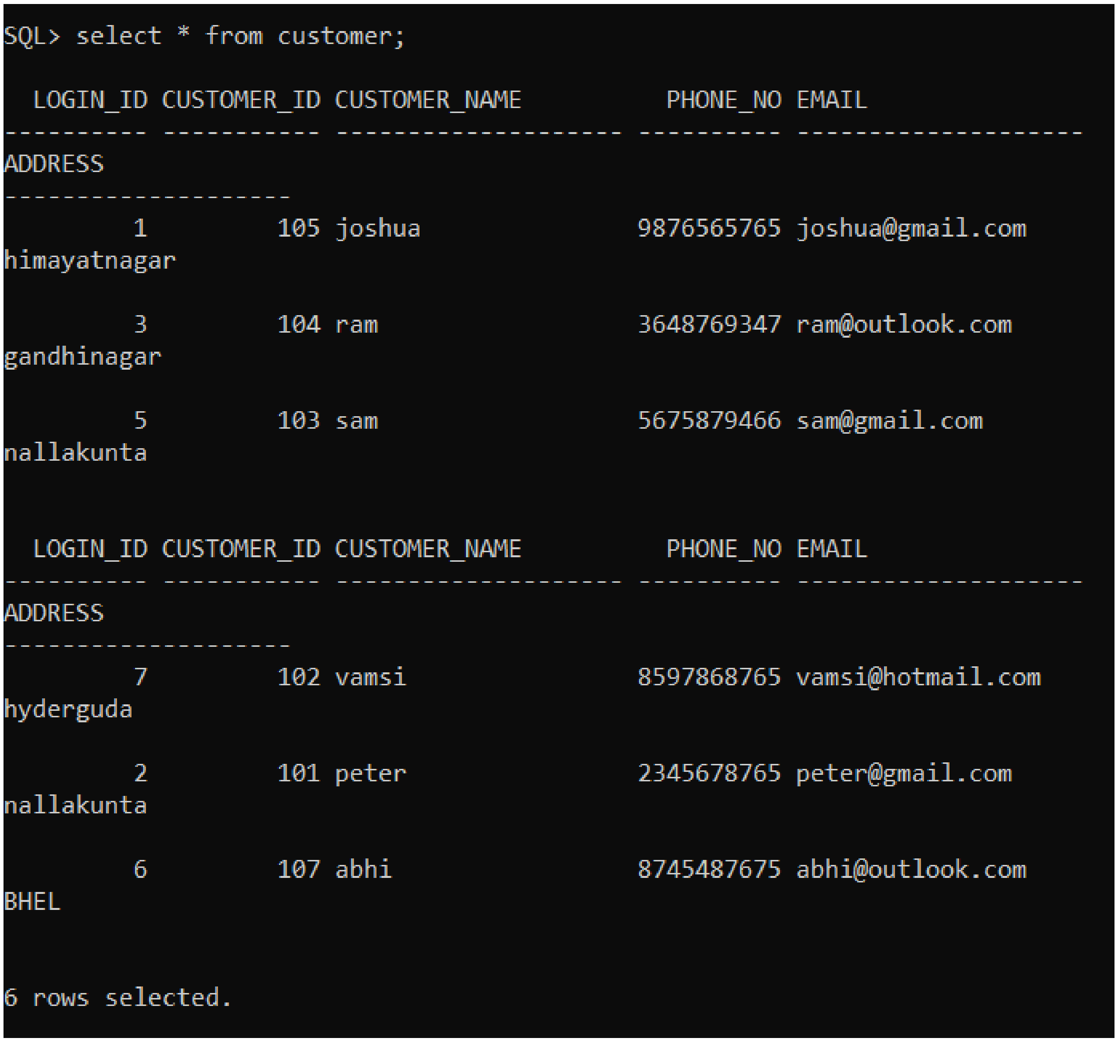
```
SQL> /
Enter value for login_id: 7
Enter value for name: vamsi
Enter value for password: chihk
old   1: insert into login values(&login_id,'&name','&password')
new   1: insert into login values(7,'vamsi','chihk')

1 row created.

SQL> select * from login;

  LOGIN_ID NAME                 PASSWORD
---------- -------------------- --------------------
         1 joshua               koklol
         2 peter                hublly
         3 ram                  zora
         4 john                 polly
         5 sam                  qwrtery
         6 abhi                 oblge
         7 vamsi                chihk
```



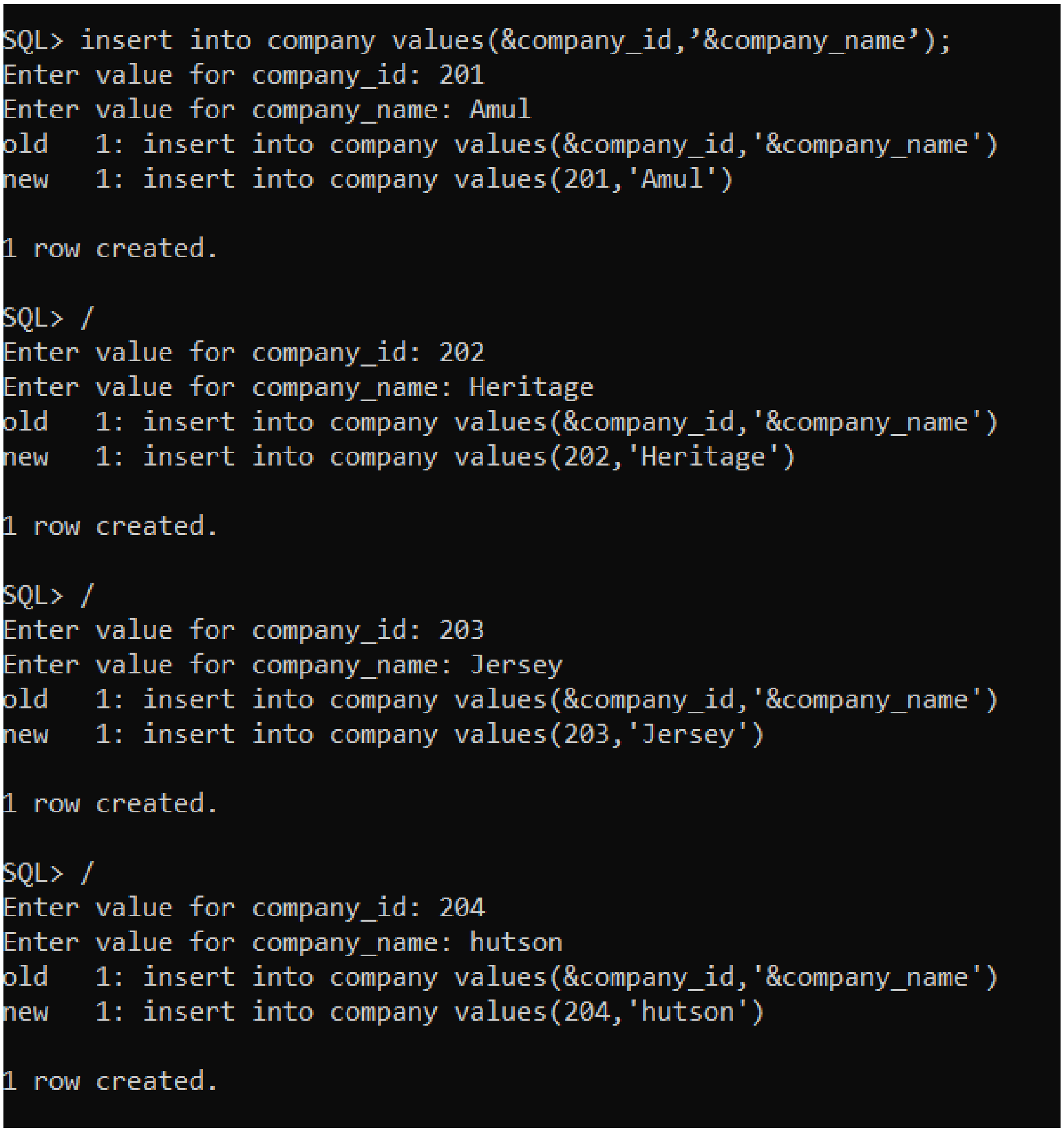
```
SQL> insert into customer values(&login_id,&customer_id,’&customer_name’,&phone_no,’&email’,’&address’);
Enter value for login_id: 1
Enter value for customer_id: 105
Enter value for customer_name: joshua
Enter value for phone_no: 9876565765
Enter value for email: joshua@gmail.com
Enter value for address: himayatnagar
old   1: insert into customer values(&login_id,&customer_id,'&customer_name',&phone_no,'&email','&address')
new   1: insert into customer values(1,105,'joshua',9876565765,'joshua@gmail.com','himayatnagar')

1 row created.

SQL> /
Enter value for login_id: 3
Enter value for customer_id: 104
Enter value for customer_name: ram
Enter value for phone_no: 3648769347
Enter value for email: ram@outlook.com
Enter value for address: gandhinagar
old   1: insert into customer values(&login_id,&customer_id,'&customer_name',&phone_no,'&email','&address')
new   1: insert into customer values(3,104,'ram',3648769347,'ram@outlook.com','gandhinagar')

1 row created.

SQL> /
Enter value for login_id: 5
Enter value for customer_id: 103
Enter value for customer_name: sam
Enter value for phone_no: 5675879466
Enter value for email: sam@gmail.com
Enter value for address: nallakunta
old   1: insert into customer values(&login_id,&customer_id,'&customer_name',&phone_no,'&email','&address')
new   1: insert into customer values(5,103,'sam',5675879466,'sam@gmail.com','nallakunta')

1 row created.
```



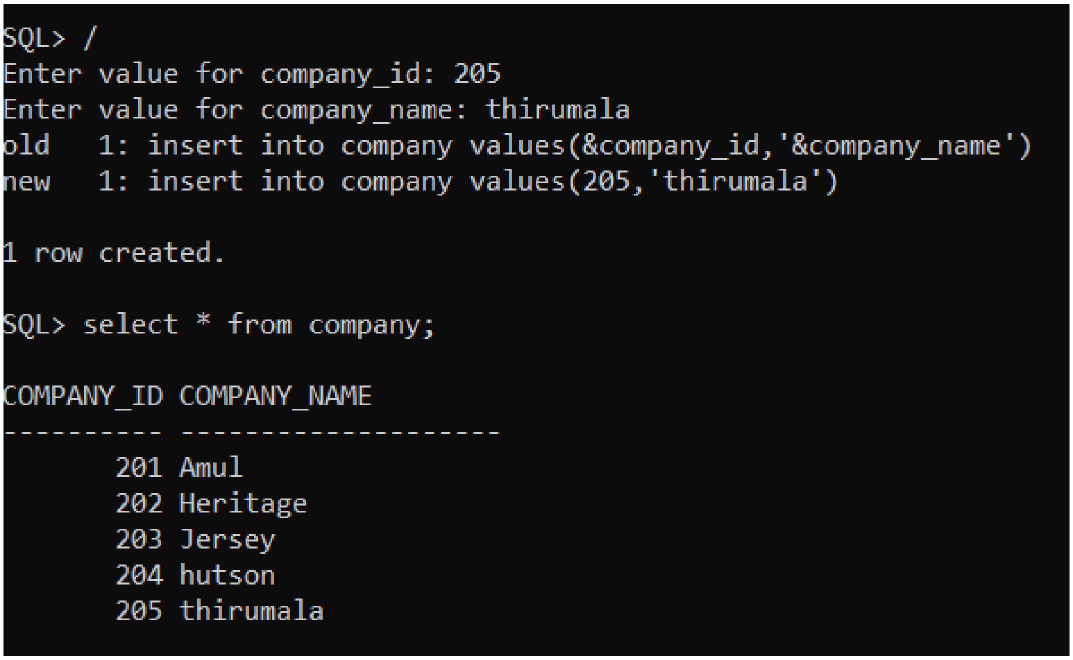
```
SQL> insert into customer values(&login_id,&customer_id,’&customer_name’,&phone_no,’&email’,’&address’);
Enter value for login_id: 2
Enter value for customer_id: 101
Enter value for customer_name: peter
Enter value for phone_no: 2345678765
Enter value for email: peter@gmail.com
Enter value for address: nallakunta
old   1: insert into customer values(&login_id,&customer_id,'&customer_name',&phone_no,'&email','&address')
new   1: insert into customer values(2,101,'peter',2345678765,'peter@gmail.com','nallakunta')

1 row created.

SQL> /
Enter value for login_id: 6
Enter value for customer_id: 107
Enter value for customer_name: abhi
Enter value for phone_no: 8745487675
Enter value for email: abhi@outlook.com
Enter value for address: BHEL
old   1: insert into customer values(&login_id,&customer_id,'&customer_name',&phone_no,'&email','&address')
new   1: insert into customer values(6,107,'abhi',8745487675,'abhi@outlook.com','BHEL')

1 row created.
```



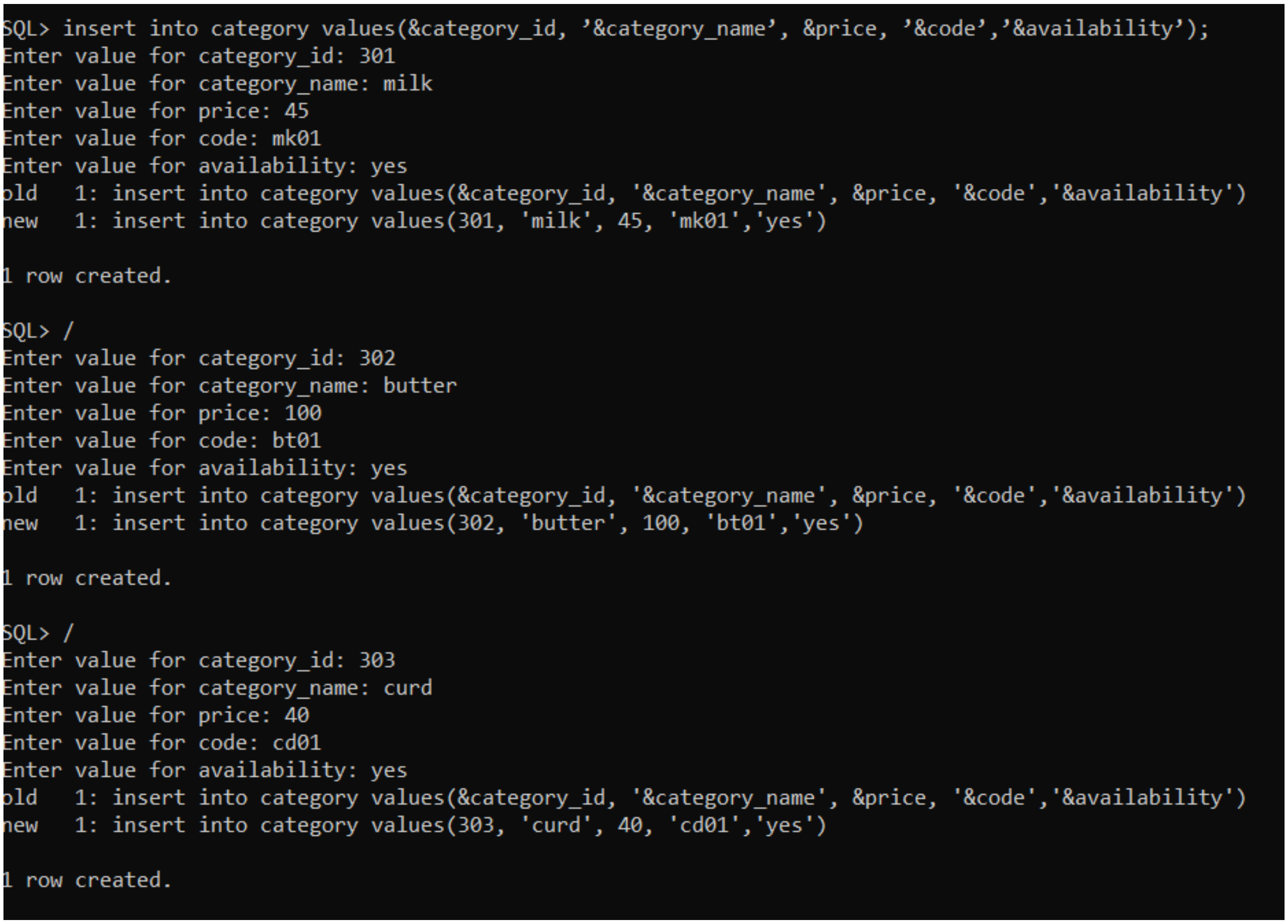
```
SQL> select * from customer;

  LOGIN_ID CUSTOMER_ID CUSTOMER_NAME          PHONE_NO EMAIL
---------- ----------- -------------------- ---------- --------------------
ADDRESS
--------------------
         1         105 joshua               9876565765 joshua@gmail.com
himayatnagar

         3         104 ram                  3648769347 ram@outlook.com
gandhinagar

         5         103 sam                  5675879466 sam@gmail.com
nallakunta


  LOGIN_ID CUSTOMER_ID CUSTOMER_NAME          PHONE_NO EMAIL
---------- ----------- -------------------- ---------- --------------------
ADDRESS
--------------------
         7         102 vamsi                8597868765 vamsi@hotmail.com
hyderguda

         2         101 peter                2345678765 peter@gmail.com
nallakunta

         6         107 abhi                 8745487675 abhi@outlook.com
BHEL


6 rows selected.
```



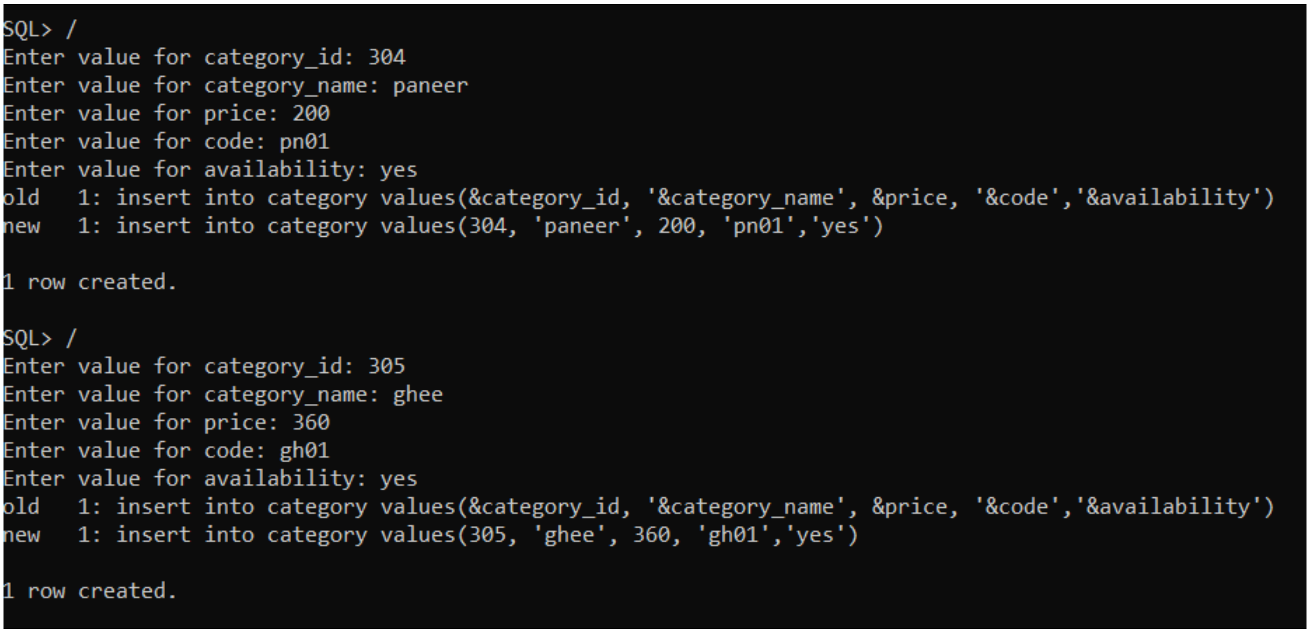
```
SQL> insert into company values(&company_id,’&company_name’);
Enter value for company_id: 201
Enter value for company_name: Amul
old   1: insert into company values(&company_id,'&company_name')
new   1: insert into company values(201,'Amul')

1 row created.

SQL> /
Enter value for company_id: 202
Enter value for company_name: Heritage
old   1: insert into company values(&company_id,'&company_name')
new   1: insert into company values(202,'Heritage')

1 row created.

SQL> /
Enter value for company_id: 203
Enter value for company_name: Jersey
old   1: insert into company values(&company_id,'&company_name')
new   1: insert into company values(203,'Jersey')

1 row created.

SQL> /
Enter value for company_id: 204
Enter value for company_name: hutson
old   1: insert into company values(&company_id,'&company_name')
new   1: insert into company values(204,'hutson')

1 row created.
```



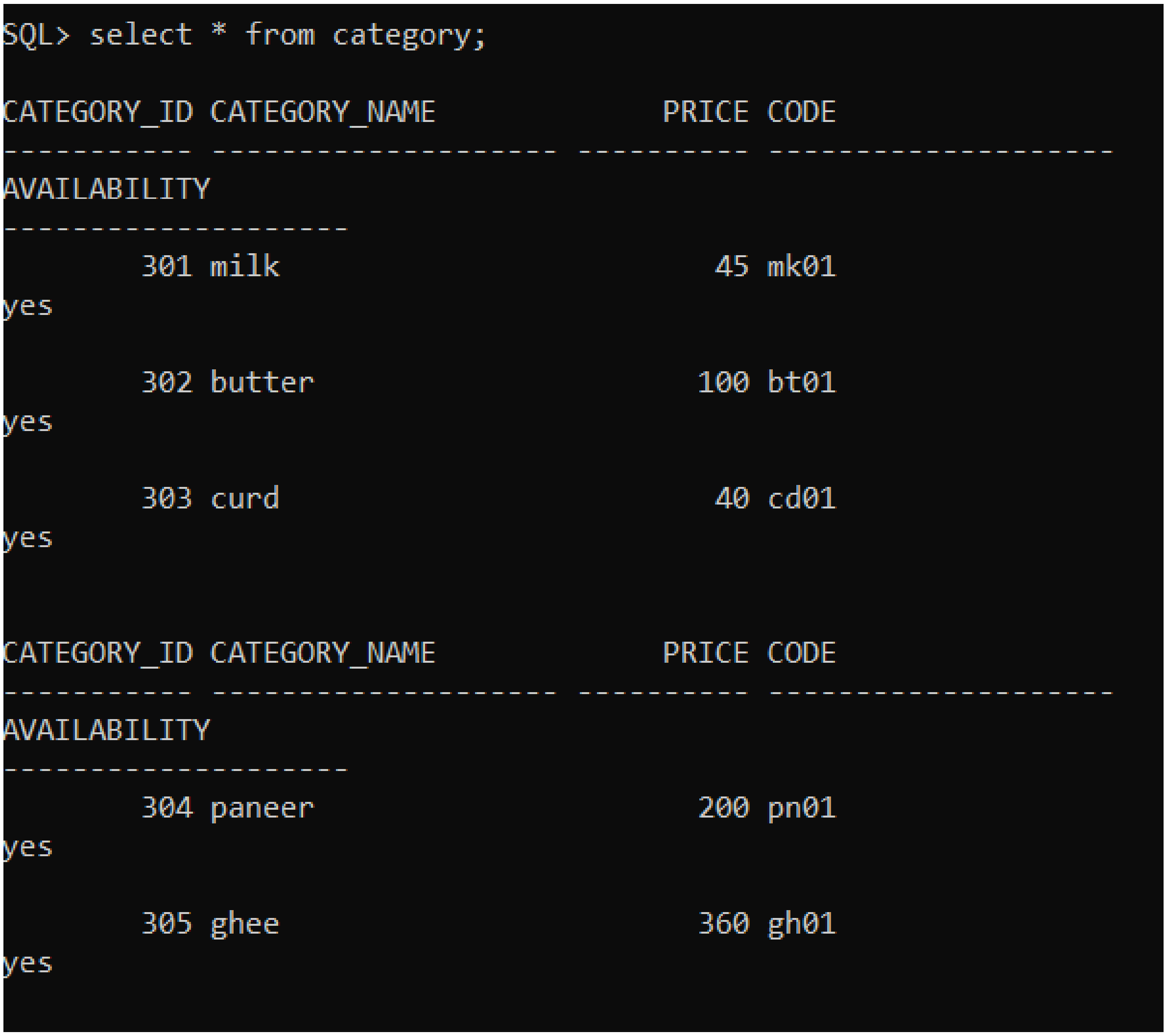
```
SQL> /
Enter value for company_id: 205
Enter value for company_name: thirumala
old   1: insert into company values(&company_id,'&company_name')
new   1: insert into company values(205,'thirumala')

1 row created.

SQL> select * from company;

COMPANY_ID COMPANY_NAME
---------- --------------------
       201 Amul
       202 Heritage
       203 Jersey
       204 hutson
       205 thirumala
```



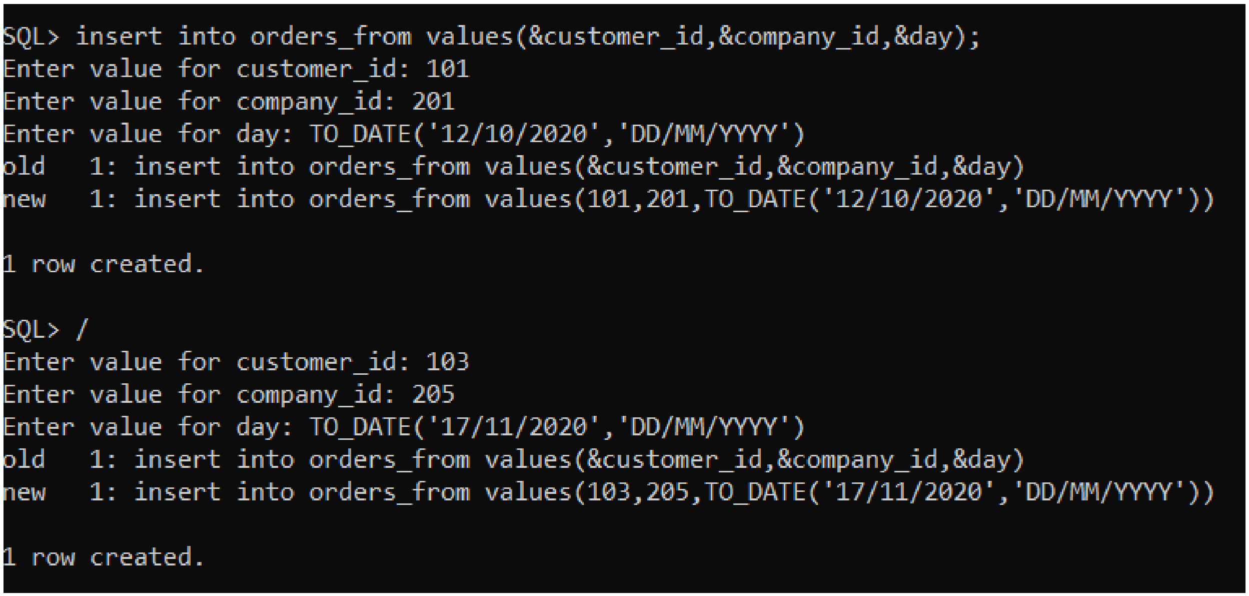
```
SQL> insert into category values(&category_id, ’&category_name’, &price, ’&code’,’&availability’);
Enter value for category_id: 301
Enter value for category_name: milk
Enter value for price: 45
Enter value for code: mk01
Enter value for availability: yes
old   1: insert into category values(&category_id, '&category_name', &price, '&code','&availability')
new   1: insert into category values(301, 'milk', 45, 'mk01','yes')

1 row created.

SQL> /
Enter value for category_id: 302
Enter value for category_name: butter
Enter value for price: 100
Enter value for code: bt01
Enter value for availability: yes
old   1: insert into category values(&category_id, '&category_name', &price, '&code','&availability')
new   1: insert into category values(302, 'butter', 100, 'bt01','yes')

1 row created.

SQL> /
Enter value for category_id: 303
Enter value for category_name: curd
Enter value for price: 40
Enter value for code: cd01
Enter value for availability: yes
old   1: insert into category values(&category_id, '&category_name', &price, '&code','&availability')
new   1: insert into category values(303, 'curd', 40, 'cd01','yes')

1 row created.
```



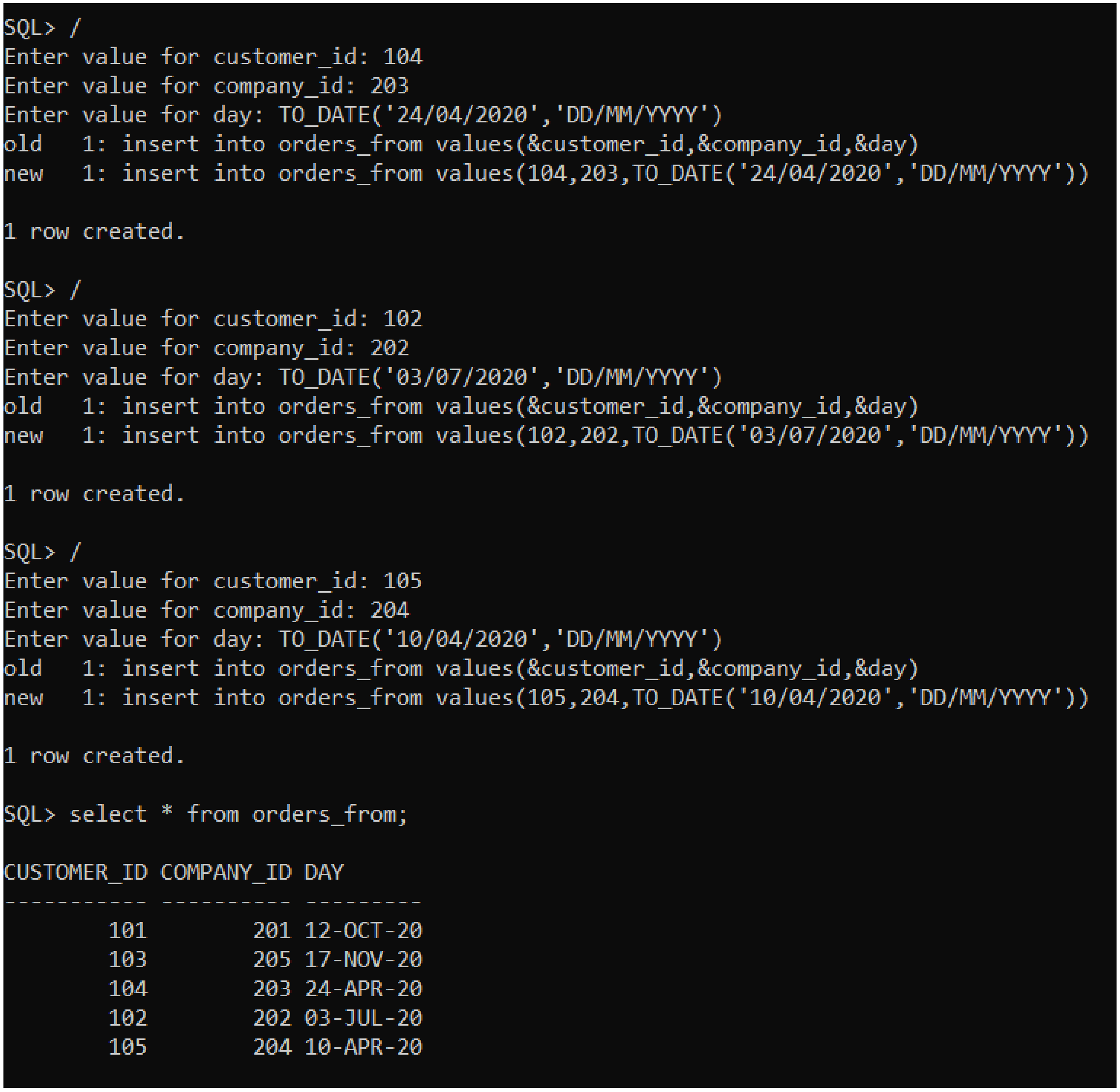
```
SQL> /
Enter value for category_id: 304
Enter value for category_name: paneer
Enter value for price: 200
Enter value for code: pn01
Enter value for availability: yes
old   1: insert into category values(&category_id, '&category_name', &price, '&code','&availability')
new   1: insert into category values(304, 'paneer', 200, 'pn01','yes')

1 row created.

SQL> /
Enter value for category_id: 305
Enter value for category_name: ghee
Enter value for price: 360
Enter value for code: gh01
Enter value for availability: yes
old   1: insert into category values(&category_id, '&category_name', &price, '&code','&availability')
new   1: insert into category values(305, 'ghee', 360, 'gh01','yes')

1 row created.
```



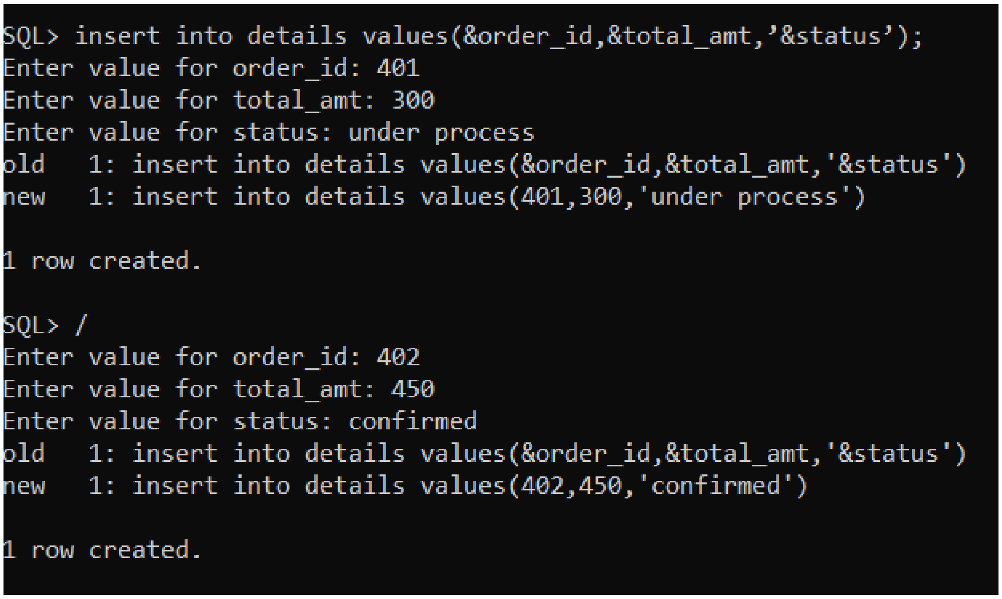
```
SQL> select * from category;

CATEGORY_ID CATEGORY_NAME             PRICE CODE
----------- -------------------- ---------- --------------------
AVAILABILITY
--------------------
        301 milk                         45 mk01
yes

        302 butter                      100 bt01
yes

        303 curd                         40 cd01
yes


CATEGORY_ID CATEGORY_NAME             PRICE CODE
----------- -------------------- ---------- --------------------
AVAILABILITY
--------------------
        304 paneer                      200 pn01
yes

        305 ghee                        360 gh01
yes
```



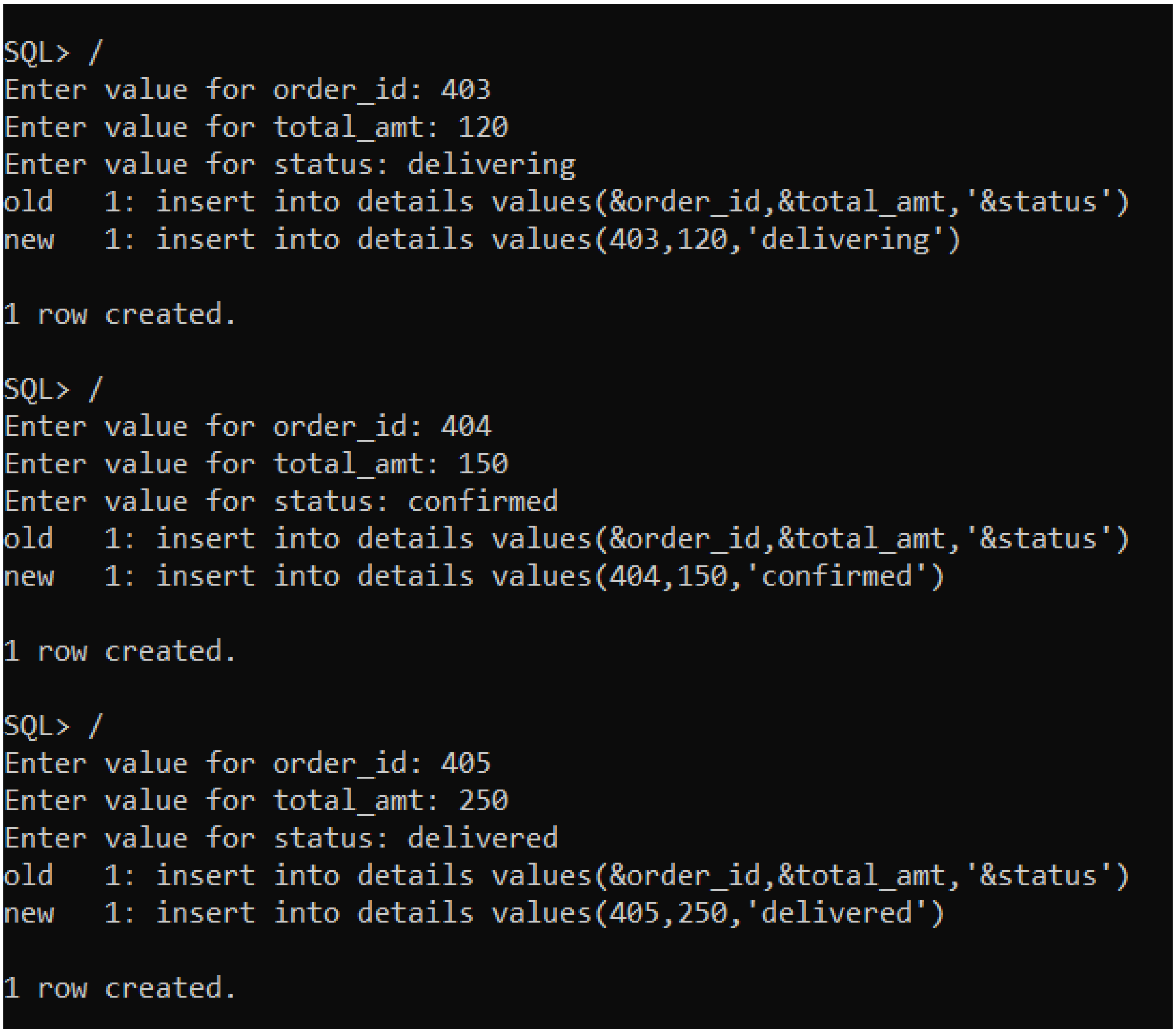
```
SQL> insert into orders_from values(&customer_id,&company_id,&day);
Enter value for customer_id: 101
Enter value for company_id: 201
Enter value for day: TO_DATE('12/10/2020','DD/MM/YYYY')
old   1: insert into orders_from values(&customer_id,&company_id,&day)
new   1: insert into orders_from values(101,201,TO_DATE('12/10/2020','DD/MM/YYYY'))

1 row created.

SQL> /
Enter value for customer_id: 103
Enter value for company_id: 205
Enter value for day: TO_DATE('17/11/2020','DD/MM/YYYY')
old   1: insert into orders_from values(&customer_id,&company_id,&day)
new   1: insert into orders_from values(103,205,TO_DATE('17/11/2020','DD/MM/YYYY'))

1 row created.
```



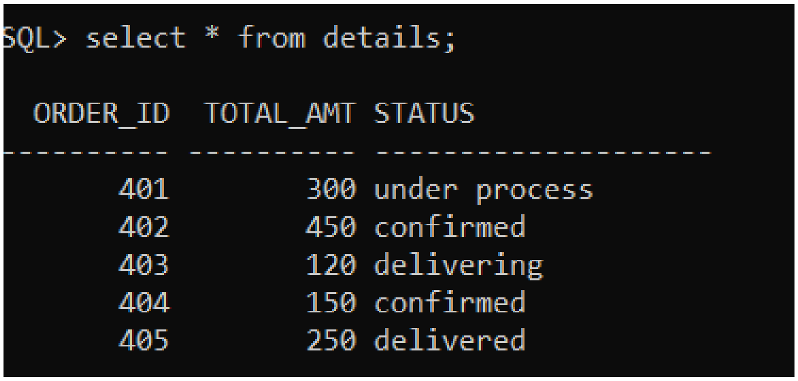
```
SQL> /
Enter value for customer_id: 104
Enter value for company_id: 203
Enter value for day: TO_DATE('24/04/2020','DD/MM/YYYY')
old   1: insert into orders_from values(&customer_id,&company_id,&day)
new   1: insert into orders_from values(104,203,TO_DATE('24/04/2020','DD/MM/YYYY'))

1 row created.

SQL> /
Enter value for customer_id: 102
Enter value for company_id: 202
Enter value for day: TO_DATE('03/07/2020','DD/MM/YYYY')
old   1: insert into orders_from values(&customer_id,&company_id,&day)
new   1: insert into orders_from values(102,202,TO_DATE('03/07/2020','DD/MM/YYYY'))

1 row created.

SQL> /
Enter value for customer_id: 105
Enter value for company_id: 204
Enter value for day: TO_DATE('10/04/2020','DD/MM/YYYY')
old   1: insert into orders_from values(&customer_id,&company_id,&day)
new   1: insert into orders_from values(105,204,TO_DATE('10/04/2020','DD/MM/YYYY'))

1 row created.

SQL> select * from orders_from;

CUSTOMER_ID COMPANY_ID DAY
----------- ---------- ---------
        101        201 12-OCT-20
        103        205 17-NOV-20
        104        203 24-APR-20
        102        202 03-JUL-20
        105        204 10-APR-20
```



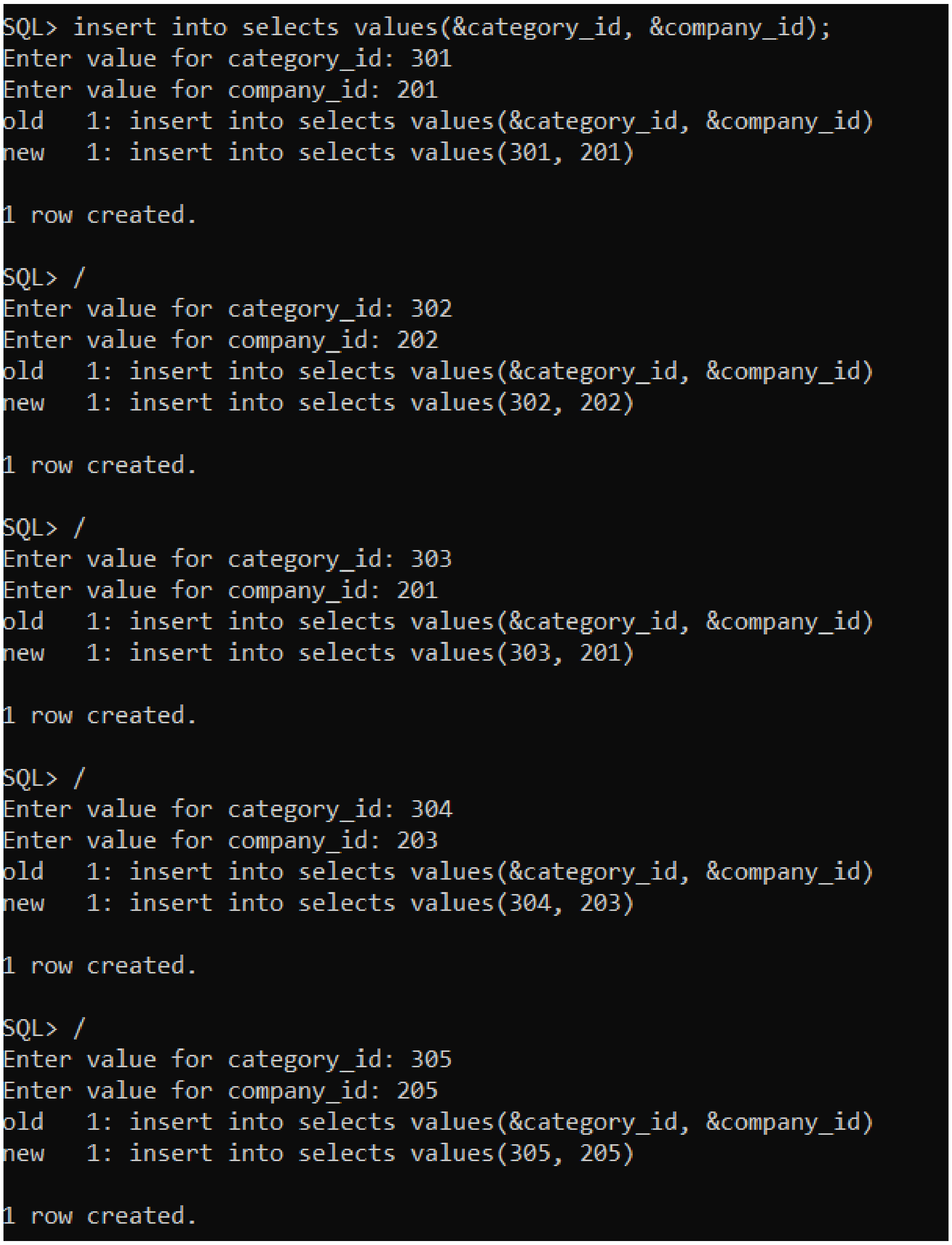
```
SQL> insert into details values(&order_id,&total_amt,’&status’);
Enter value for order_id: 401
Enter value for total_amt: 300
Enter value for status: under process
old   1: insert into details values(&order_id,&total_amt,'&status')
new   1: insert into details values(401,300,'under process')

1 row created.

SQL> /
Enter value for order_id: 402
Enter value for total_amt: 450
Enter value for status: confirmed
old   1: insert into details values(&order_id,&total_amt,'&status')
new   1: insert into details values(402,450,'confirmed')

1 row created.
```



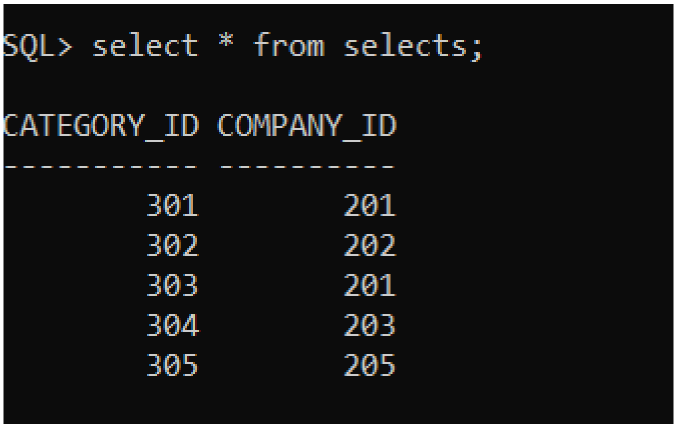
```
SQL> /
Enter value for order_id: 403
Enter value for total_amt: 120
Enter value for status: delivering
old   1: insert into details values(&order_id,&total_amt,'&status')
new   1: insert into details values(403,120,'delivering')

1 row created.

SQL> /
Enter value for order_id: 404
Enter value for total_amt: 150
Enter value for status: confirmed
old   1: insert into details values(&order_id,&total_amt,'&status')
new   1: insert into details values(404,150,'confirmed')

1 row created.

SQL> /
Enter value for order_id: 405
Enter value for total_amt: 250
Enter value for status: delivered
old   1: insert into details values(&order_id,&total_amt,'&status')
new   1: insert into details values(405,250,'delivered')

1 row created.
```



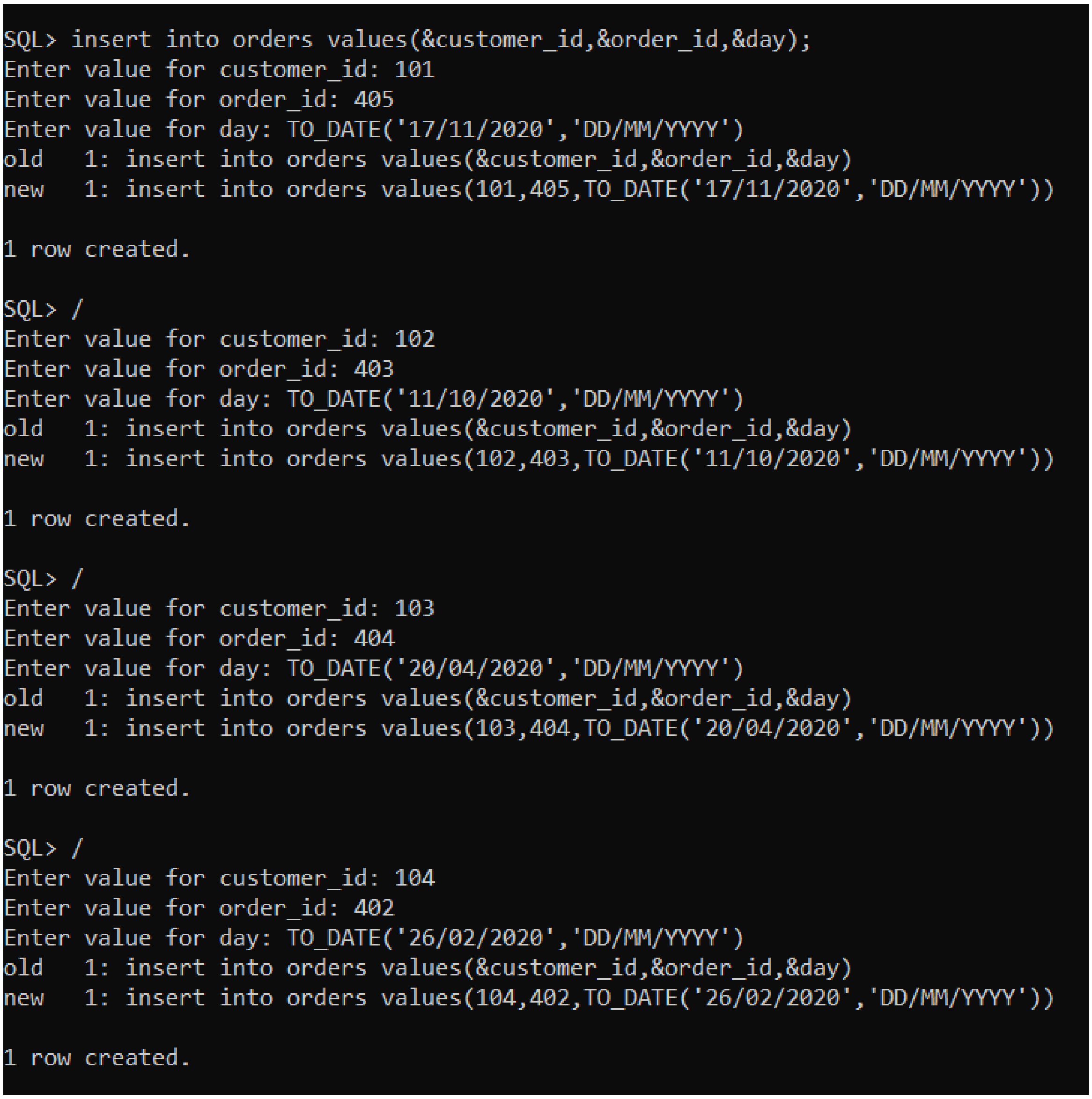
```
SQL> select * from details;

  ORDER_ID  TOTAL_AMT STATUS
---------- ---------- --------------------
       401        300 under process
       402        450 confirmed
       403        120 delivering
       404        150 confirmed
       405        250 delivered
```



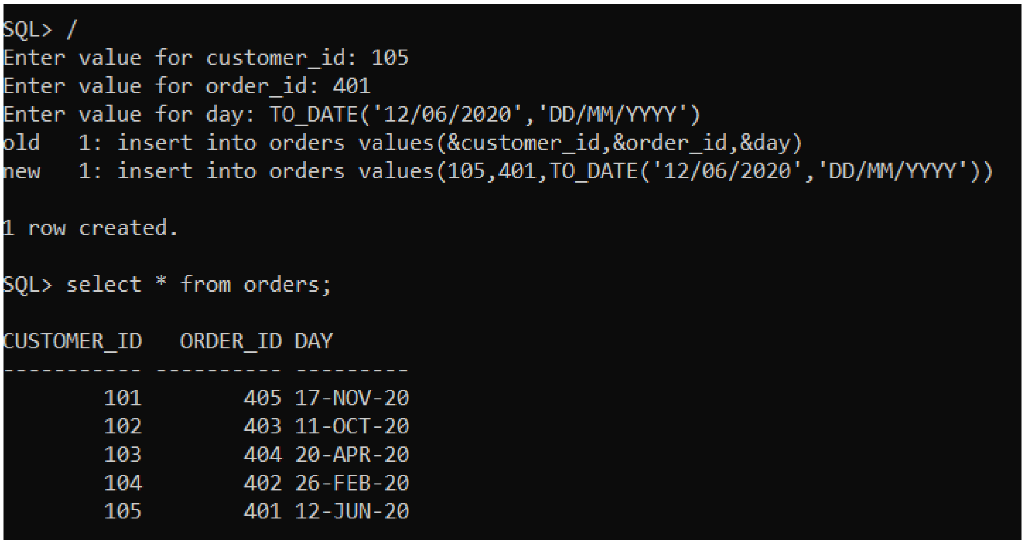
```
SQL> insert into selects values(&category_id, &company_id);
Enter value for category_id: 301
Enter value for company_id: 201
old   1: insert into selects values(&category_id, &company_id)
new   1: insert into selects values(301, 201)

1 row created.

SQL> /
Enter value for category_id: 302
Enter value for company_id: 202
old   1: insert into selects values(&category_id, &company_id)
new   1: insert into selects values(302, 202)

1 row created.

SQL> /
Enter value for category_id: 303
Enter value for company_id: 201
old   1: insert into selects values(&category_id, &company_id)
new   1: insert into selects values(303, 201)

1 row created.

SQL> /
Enter value for category_id: 304
Enter value for company_id: 203
old   1: insert into selects values(&category_id, &company_id)
new   1: insert into selects values(304, 203)

1 row created.

SQL> /
Enter value for category_id: 305
Enter value for company_id: 205
old   1: insert into selects values(&category_id, &company_id)
new   1: insert into selects values(305, 205)

1 row created.
```



```
SQL> select * from selects;

CATEGORY_ID COMPANY_ID
----------- ----------
        301        201
        302        202
        303        201
        304        203
        305        205
```



```
SQL> insert into orders values(&customer_id,&order_id,&day);
Enter value for customer_id: 101
Enter value for order_id: 405
Enter value for day: TO_DATE('17/11/2020','DD/MM/YYYY')
old   1: insert into orders values(&customer_id,&order_id,&day)
new   1: insert into orders values(101,405,TO_DATE('17/11/2020','DD/MM/YYYY'))

1 row created.

SQL> /
Enter value for customer_id: 102
Enter value for order_id: 403
Enter value for day: TO_DATE('11/10/2020','DD/MM/YYYY')
old   1: insert into orders values(&customer_id,&order_id,&day)
new   1: insert into orders values(102,403,TO_DATE('11/10/2020','DD/MM/YYYY'))

1 row created.

SQL> /
Enter value for customer_id: 103
Enter value for order_id: 404
Enter value for day: TO_DATE('20/04/2020','DD/MM/YYYY')
old   1: insert into orders values(&customer_id,&order_id,&day)
new   1: insert into orders values(103,404,TO_DATE('20/04/2020','DD/MM/YYYY'))

1 row created.

SQL> /
Enter value for customer_id: 104
Enter value for order_id: 402
Enter value for day: TO_DATE('26/02/2020','DD/MM/YYYY')
old   1: insert into orders values(&customer_id,&order_id,&day)
new   1: insert into orders values(104,402,TO_DATE('26/02/2020','DD/MM/YYYY'))

1 row created.
```



```
SQL> /
Enter value for customer_id: 105
Enter value for order_id: 401
Enter value for day: TO_DATE('12/06/2020','DD/MM/YYYY')
old   1: insert into orders values(&customer_id,&order_id,&day)
new   1: insert into orders values(105,401,TO_DATE('12/06/2020','DD/MM/YYYY'))

1 row created.

SQL> select * from orders;

CUSTOMER_ID   ORDER_ID DAY
----------- ---------- ---------
        101        405 17-NOV-20
        102        403 11-OCT-20
        103        404 20-APR-20
        104        402 26-FEB-20
        105        401 12-JUN-20
```

IMPLEMENTATION

Front end programs and its connectivity

Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()
    {
             try {
             Connection 
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
             statement=con.createStatement();
             statement.executeUpdate("commit");
             }
             catch (SQLException connectException)
             {
               System.out.println(connectException.getMessage());
               System.out.println(connectException.getSQLState());
               System.out.println(connectException.getErrorCode());
               System.exit(1);
             }
    }
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

MAIN METHOD:

```
package Edairy;
import java.awt.BorderLayout;
import java.awt.EventQueue;
import Edairy.*;
import java.sql.*;
import javax.sql.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.event.MenuEvent;
import javax.swing.event.MenuListener;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Main extends JFrame{
        private JFrame recommendedmedicines;
        private JTextField txtLoginId;
        private JPasswordField txtPassword;
        private JTextField txtname;
        private JPanel contentPane;
        Statement statement;
        Connection con;ResultSet rs;
        public static void main(String[] args) {
                EventQueue.invokeLater(new Runnable() {
                        public void run() {
                                try {
                                        Main window = new Main();
                                        window.recommendedmedicines.setVisible(true);
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                });
        }
        public void connectToDB()
    {
                try {
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
                statement=con.createStatement();
                statement.executeUpdate("commit");
                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
    }
        public Main() {
                initialize();
        }
        JButton btnLogin;
        JButton btnSignup;
        JLabel lblNewLabel_2;
        private void initialize() {
                recommendedmedicines = new JFrame();
                recommendedmedicines.setTitle("Dairy Store");
                recommendedmedicines.setBounds(100, 100, 450, 300);
                recommendedmedicines.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                recommendedmedicines.getContentPane().setLayout(null);
                JMenuBar menuBar = new JMenuBar();
                recommendedmedicines.setJMenuBar(menuBar);
                JMenu mnNewMenu = new JMenu("Log In");
                mnNewMenu.addMenuListener(new MenuListener() {
                        public void menuCanceled(MenuEvent e) {
                        }
                        public void menuDeselected(MenuEvent e) {
                        }
                        public void menuSelected(MenuEvent e) {
                                btnLogin.setVisible(true);
                                btnSignup.setVisible(false);
                                txtname.setVisible(false);
                                lblNewLabel_2.setVisible(false);

                        }
                });
                JMenu mnNewMenu1 = new JMenu("Signup");
                mnNewMenu1.addMenuListener(new MenuListener() {
                        public void menuCanceled(MenuEvent e) {
                        }
                        public void menuDeselected(MenuEvent e) {
                        }
                        public void menuSelected(MenuEvent e) {
                                btnLogin.setVisible(false);
                                btnSignup.setVisible(true);
                                txtname.setVisible(true);
                                lblNewLabel_2.setVisible(true);
                        }
                });
                menuBar.add(mnNewMenu);
                menuBar.add(mnNewMenu1);
                txtLoginId = new JTextField();
                txtLoginId.setBounds(198, 47, 96, 20);
                recommendedmedicines.getContentPane().add(txtLoginId);
                txtLoginId.setColumns(10);
                txtname = new JTextField();
                txtname.setBounds(198, 126, 96, 20);
                recommendedmedicines.getContentPane().add(txtname);
                txtname.setColumns(10);
                txtname.setVisible(false);
                JLabel lblNewLabel = new JLabel("Login Id");
                lblNewLabel.setBounds(76, 50, 84, 14);
                recommendedmedicines.getContentPane().add(lblNewLabel);
                JLabel lblNewLabel_1 = new JLabel("Password");
                lblNewLabel_1.setBounds(76, 88, 84, 14);
                recommendedmedicines.getContentPane().add(lblNewLabel_1);
                lblNewLabel_2 = new JLabel("User Name");
                lblNewLabel_2.setBounds(76, 126, 84, 14);
                recommendedmedicines.getContentPane().add(lblNewLabel_2);
                lblNewLabel_2.setVisible(false);
                btnLogin = new JButton("Login");
                btnLogin.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                             boolean isValidUser=false;
                             try {
                                 connectToDB();
                                 String txt = "select login_id,password from login where 
   login_id = '"+txtLoginId.getText()+"' and password ='"+new 
   String(txtPassword.getPassword())+"'";
                                 //System.out.println(txt);
                                 ResultSet rs=statement.executeQuery(txt);
                                 if(rs.next()) {
                                     isValidUser = true;
                                 }
                                 rs.close();
                             }
                             catch(Exception ex) {
                                 System.out.println(ex);
                             }
                             if(isValidUser) {
                                 EdairyUI ex = new EdairyUI();
                                 //login code here
                             }
                             else {
                                 JOptionPane.showMessageDialog(null,"Invalid User");
                             }
                    }
                });
                btnLogin.setBounds(150, 160, 89, 23);
                recommendedmedicines.getContentPane().add(btnLogin);
                btnSignup = new JButton("Signup");
                btnSignup.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                             try {
                                 connectToDB();
                                 String txt = "insert into login (login_id,name,password) 
   values('"+txtLoginId.getText()+"','"+txtname.getText()+"' ,'"+new 
   String(txtPassword.getPassword())+"')";
                                 //System.out.println(txt);
                                 ResultSet rs=statement.executeQuery(txt);
                                 if(rs.next()) {
                                     JOptionPane.showMessageDialog(null,"Sign in 
   successful");

                                     EdairyUI es = new EdairyUI();
                                 }
                                 rs.close();
                             }
                             catch(Exception ex) {
                                 System.out.println(ex);
```

```
                            }
                     }
              });
              btnSignup.setBounds(150, 160, 89, 23);
              recommendedmedicines.getContentPane().add(btnSignup);
              txtPassword = new JPasswordField();
              txtPassword.setBounds(198, 85, 96, 20);
              recommendedmedicines.getContentPane().add(txtPassword);
       }
}
```

EdairyUI:

```
package Edairy;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class EdairyUI extends JFrame{
       private static final long serialVersionUID = 1L;
       private JMenuBar mnu;
       private JMenu mnulogin;
       private JMenu mnuDetails;
       private JMenu mnuCustomer;
       private JMenu mnuCategory;
       private JMenu mnuCompany;
       private JMenu mnuOrders;
       private JMenu mnuCustomer_login;
       private JMenu mnuOrders_from;
       private JMenu mnuSelects;
       private JMenuItem insert1,update1,delete1,view1;
       private JMenuItem insert2,update2,delete2,view2;
       private JMenuItem insert3,update3,delete3,view3;
       private JMenuItem insert4,update4,delete4,view4;
       private JMenuItem insert5,update5,delete5,view5;
       private JMenuItem insert6,update6,delete6,view6;
       private JMenuItem insert7,update7,delete7,view7;
       private JMenuItem insert8,update8,delete8,view8;
       private JMenuItem insert9,update9,delete9,view9;
       private JLabel labelName;
       private static JPanel p0,p1;
```

```
void initialize() {
        mnu=new JMenuBar();
        mnulogin= new JMenu("Login");
        mnuCustomer= new JMenu("Customer");
        mnuCategory= new JMenu("Category");
        mnuCompany= new JMenu("Company");
        mnuDetails= new JMenu("Details");
        mnuOrders= new JMenu("Orders");
        mnuCustomer_login =  new JMenu("Customer_login");
        mnuOrders_from = new JMenu("Orders_from");
        mnuSelects = new JMenu("Selects");
        labelName=new JLabel("DAIRY MANAGEMENT SYSTEM");
         p1=new JPanel();
        p0=new JPanel();
        insert1=new JMenuItem("Insert");
        update1=new JMenuItem("Update");
        delete1=new JMenuItem("Delete");
        view1=new JMenuItem("View");
        insert2=new JMenuItem("Insert");
        update2=new JMenuItem("Update");
        delete2=new JMenuItem("Delete");
        view2=new JMenuItem("View");
        insert3=new JMenuItem("Insert");
        update3=new JMenuItem("Update");
        delete3=new JMenuItem("Delete");
        view3=new JMenuItem("View");
        insert4=new JMenuItem("Insert");
        update4=new JMenuItem("Update");
        delete4=new JMenuItem("Delete");
        view4=new JMenuItem("View");
        insert5=new JMenuItem("Insert");
        update5=new JMenuItem("Update");
        delete5=new JMenuItem("Delete");
        view5=new JMenuItem("View");
        insert6=new JMenuItem("Insert");
        update6=new JMenuItem("Update");
        delete6=new JMenuItem("Delete");
        view6=new JMenuItem("View");
        insert7=new JMenuItem("Insert");
        update7=new JMenuItem("Update");
        delete7=new JMenuItem("Delete");
        view7=new JMenuItem("View");
        insert8=new JMenuItem("Insert");
        update8=new JMenuItem("Update");
        delete8=new JMenuItem("Delete");
        view8=new JMenuItem("View");
        insert9=new JMenuItem("Insert");
        update9=new JMenuItem("Update");
        delete9=new JMenuItem("Delete");
        view9=new JMenuItem("View");
}
void addComponentsToFrame() {
         //mnulogin.add(insert1);
         mnulogin.add(delete1);
```

```
              mnulogin.add(update1);
              mnulogin.add(view1);
              mnuCustomer.add(insert2);
              mnuCustomer.add(delete2);
              mnuCustomer.add(update2);
              mnuCustomer.add(view2);
              mnuCategory.add(insert3);
              mnuCategory.add(delete3);
              mnuCategory.add(update3);
              mnuCategory.add(view3);
              mnuCompany.add(insert4);
              mnuCompany.add(delete4);
              mnuCompany.add(update4);
              mnuCompany.add(view4);
              mnuDetails.add(insert5);
              mnuDetails.add(delete5);
              mnuDetails.add(update5);
              mnuDetails.add(view5);
              mnuOrders.add(insert6);
              mnuOrders.add(delete6);
              mnuOrders.add(update6);
              mnuOrders.add(view6);
              mnuOrders_from.add(insert7);
              mnuOrders_from.add(delete7);
              mnuOrders_from.add(update7);
              mnuOrders_from.add(view7);
              mnuSelects.add(insert8);
              mnuSelects.add(delete8);
              mnuSelects.add(update8);
              mnuSelects.add(view8);
              mnuCustomer_login.add(insert9);
              mnuCustomer_login.add(delete9);
              mnuCustomer_login.add(update9);
              mnuCustomer_login.add(view9);
              mnu.add(mnulogin);
              mnu.add(mnuCustomer);
              mnu.add(mnuCategory);
              mnu.add(mnuCompany);
              mnu.add(mnuDetails);
              mnu.add(mnuOrders);
              mnu.add(mnuOrders_from);
              mnu.add(mnuSelects);
              mnu.add(mnuCustomer_login);
              setJMenuBar(mnu);
              p1.add(labelName);p1.setAlignmentY(CENTER_ALIGNMENT);
              p1.setBounds(500,500,800,100);
             p0.add(p1);
              add(p0);
       }
void closeWindow(){
             try {
                    int a=JOptionPane.showConfirmDialog(this,"Are you sure want to Quit Dairy 
Management System:");
                    if(a==JOptionPane.YES_OPTION){
```

```
                JOptionPane.showMessageDialog(this,
                        "Thank you!\nExiting Dairy Management System","Quit",
                        JOptionPane.WARNING_MESSAGE);
                System.exit(0);
            }
            else if (a== JOptionPane.NO_OPTION) {
                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
            }
            else if (a== JOptionPane.CANCEL_OPTION) {
                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
            }
        }
            catch(Exception e) {
                System.out.println(e);
                }
    }
    void register() {
        login log=new login(p0,EdairyUI.this,insert1,delete1,update1,view1);
        log.buildGUI();
        Customer Cust=new Customer(p0,EdairyUI.this,insert2,delete2,update2,view2);
        Cust.buildGUI();
        Details det=new Details(p0,EdairyUI.this,insert5,delete5,update5,view5);
        det.buildGUI();
        Category cat=new Category(p0,EdairyUI.this,insert3,delete3,update3,view3);
        cat.buildGUI();
        Company cmp=new Company(p0,EdairyUI.this,insert4,delete4,update4,view4);
        cmp.buildGUI();
        Orders ord=new Orders(p0,EdairyUI.this,insert6,delete6,update6,view6);
        ord.buildGUI();
        Orders_from of = new
Orders_from(p0,EdairyUI.this,insert7,delete7,update7,view7);
        of.buildGUI();
        Selects se = new Selects(p0,EdairyUI.this,insert8,delete8,update8,view8);
        se.buildGUI();
        Customer_login cl = new
Customer_login(p0,EdairyUI.this,insert9,delete9,update9,view9);
        cl.buildGUI();
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent we)
            {
                closeWindow();
            }
        });
}
public EdairyUI() {
        initialize();
        addComponentsToFrame();
        register();
        pack();
        setTitle("Dairy Management System");
        setSize(600,600);
        setVisible(true);
    }
}
```

Login:

```
package Edairy;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class login {
	private static final long serialVersionUID = 1L;
	private JButton loginButton,deleteButton,viewButton,updateButton;
	private JPanel p1,p2,p3,p;
	private JLabel lbllog_id,lblName,lblpwd;
	private JTextField txtlog_id,txtName,txtpwd;
	private List loginIDList;
	Connection con;ResultSet rs;
	Statement statement;
	private JFrame frame;
	private JMenuItem insert,delete,update,view;
	public login(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem 
update,JMenuItem view)
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
		this.frame=frame;
		this.insert=insert;
```

```
             this.delete=delete;
             this.update=update;
             this.view=view;
             lbllog_id=new JLabel("Login_ID");
             lblName=new JLabel("Name");
             lblpwd=new JLabel("Password");
             txtlog_id=new JTextField(15);
             txtName=new JTextField(15);
             txtpwd=new JPasswordField(8);
             this.p=p;
      }
      public void connectToDB()
    {
             try {
             Connection con=DriverManager.getConnection(
             "jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
             statement=con.createStatement();
             statement.executeUpdate("commit");
             }
             catch (SQLException connectException)
             {
               System.out.println(connectException.getMessage());
               System.out.println(connectException.getSQLState());
               System.out.println(connectException.getErrorCode());
               System.exit(1);
             }
    }
      private void displaySQLErrors(SQLException e)
      {
             JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() +
"\n"+"SQLState:     " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
      }
      public void loadLogins() {
             try {
                    loginIDList.removeAll();
                    rs=statement.executeQuery("select login_id from login");
                    while(rs.next()) {
                           loginIDList.add(rs.getString("login_ID"));
                    }
                    }
             catch(SQLException e) {
                    displaySQLErrors(e);
             }
      }
      public void buildGUI() {
             loginButton=new JButton("Submit");
             insert.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent arg0) {
                           // TODO Auto-generated method stub
                           txtlog_id.setText(null);
                           txtName.setText(null);
                           txtpwd.setText(null);
                           p.removeAll();
```

```
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();
                          p1=new JPanel();
                           p1.setLayout(new GridLayout(3,2));
                           p1.add(lbllog_id);
                           p1.add(txtlog_id);
                           p1.add(lblName);
                           p1.add(txtName);
                           p1.add(lblpwd);
                           p1.add(txtpwd);
                           p3=new JPanel(new FlowLayout());
                           p3.add(loginButton);
                          p3.setBounds(200,250,75,35);
                           p1.setBounds(115,80,300,150);
                           p2 = new JPanel(new FlowLayout());
                                  loginIDList=new List(10);
                                  loadLogins();
                                  p2.add(loginIDList);
                                  p2.setBounds(100,320,300,180);
                           p. add(p1);
                           p.add(p3);
                           p. add(p2);
                           p.setLayout(new BorderLayout());
                                 frame.add(p);
                                 frame.setSize(600,600);
                                 frame.validate();
                           loginButton.addActionListener(new ActionListener() {
                                  @Override
                                        public void actionPerformed(ActionEvent e) {
                                              // TODO Auto-generated method stub
                          try {
                                 String query="INSERT INTO login
VALUES("+txtlog_id.getText()+",'"+txtName.getText()+"','"+txtpwd.getText()+"')";

                                 int i=statement.executeUpdate(query);
                                 JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows
succesfully");

                                 loadLogins();
                                 System.out.println("done");
                          }
                          catch(SQLException insertException){
                                 displaySQLErrors(insertException);
                          }
                           }
                                 });
                    }
                    });
             delete.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent arg0) {
                          // TODO Auto-generated method stub
                          deleteButton=new JButton("Delete");
                          txtlog_id.setText(null);
                          txtName.setText(null);
                          txtpwd.setText(null);
                          p.removeAll();
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();
                          p1=new JPanel();
                          p1.setLayout(new GridLayout(3,2));
                           p1.add(lbllog_id);
                           p1.add(txtlog_id);
                           p1.add(lblName);
                           p1.add(txtName);
                           p1.add(lblpwd);
                           p1.add(txtpwd);
                           p3=new JPanel(new FlowLayout());
                           p3.add(deleteButton);
                          p3.setBounds(200,250,75,35);
                           p1.setBounds(115,80,300,150);
                           p2 = new JPanel(new FlowLayout());
                                  loginIDList=new List(10);
                                  loadLogins();
                                  p2.add(loginIDList);
                                  p2.setBounds(100,320,300,180);
                           p. add(p1);
                           p.add(p3);
                           p. add(p2);
                           p.setLayout(new BorderLayout());
                                 frame.add(p);
                                 frame.setSize(600,600);
                                 frame.validate();
                           loginIDList.addItemListener(new ItemListener()
                                 {
                                        public void itemStateChanged(ItemEvent e)
                                        {
                                              try
                                              {
                                                     rs=statement.executeQuery("select * from
login");

                                                     while (rs.next())
                                                     {
                                                            if
(rs.getString("login_id").equals(loginIDList.getSelectedItem()))
                                                            break;
                                                     }
                                                     if (!rs.isAfterLast())
                                                     {

      txtlog_id.setText(rs.getString("login_id"));

      txtName.setText(rs.getString("Name"));

      txtpwd.setText(rs.getString("password"));
                                                     }
                                              }
                                              catch (SQLException selectException)
                                              {
                                                    displaySQLErrors(selectException);
                                              }
                                       }
                                 });
                           deleteButton.addActionListener(new ActionListener() {
                                  @Override
                                       public void actionPerformed(ActionEvent e) {
                                              // TODO Auto-generated method stub
                          try {
                                 String password=JOptionPane.showInputDialog(p,"Enter the 
password");
                                 txtpwd.setText(password);
                                 int a=JOptionPane.showConfirmDialog(p,"Are you sure want to 
delete:");
                                 if(a==JOptionPane.YES_OPTION){
                                 String query="DELETE FROM LOGIN WHERE 
LOGIN_ID="+txtlog_id.getText()+"AND PASSWORD='"+txtpwd.getText()+"'";
                                 int i=statement.executeUpdate(query);
                                 JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows 
succesfully");
                                 loadLogins();
                                 }
                          }
                          catch(SQLException deleteException){
                                 displaySQLErrors(deleteException);
                          }
                           }
                                 });
                    }
                    });
             update.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent arg0) {
                          // TODO Auto-generated method stub
                                 updateButton = new JButton("Modify");
                          txtlog_id.setText(null);
                          txtName.setText(null);
                          txtpwd.setText(null);
                          p.removeAll();
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();
                          p1=new JPanel();
                          p1.setLayout(new GridLayout(3,2));
                           p1.add(lbllog_id);
                           p1.add(txtlog_id);
                           p1.add(lblName);
                           p1.add(txtName);
                           p1.add(lblpwd);
                           p1.add(txtpwd);
                           p3=new JPanel(new FlowLayout());
                           p3.add(updateButton);
                          p3.setBounds(200,250,75,35);
                           p1.setBounds(115,80,300,150);
                           p2 = new JPanel(new FlowLayout());
                                  loginIDList=new List(10);
                                  loadLogins();
                                  p2.add(loginIDList);
                                  p2.setBounds(100,320,300,180);
                           p. add(p1);
                           p.add(p3);
                           p. add(p2);
                           p.setLayout(new BorderLayout());
                                 frame.add(p);
                                 frame.setSize(600,600);
                                 frame.validate();
                           loginIDList.addItemListener(new ItemListener()
                                 {
                                        public void itemStateChanged(ItemEvent e)
                                        {
                                              try
                                              {
                                                     rs=statement.executeQuery("select * from
login");
                                                     while (rs.next())
                                                     {
                                                            if
(rs.getString("login_id").equals(loginIDList.getSelectedItem()))
                                                            break;
                                                     }
                                                     if (!rs.isAfterLast())
                                                     {

      txtlog_id.setText(rs.getString("login_id"));

      txtName.setText(rs.getString("Name"));

      txtpwd.setText(rs.getString("password"));
                                                     }
                                              }
                                              catch (SQLException selectException)
                                              {
                                                     displaySQLErrors(selectException);
                                              }
                                        }
                                 });
                           updateButton.addActionListener(new ActionListener() {
                                  @Override
                                        public void actionPerformed(ActionEvent e) {
                                              // TODO Auto-generated method stub
                          try {
                                 loadLogins();
                                 String password=JOptionPane.showInputDialog(p,"Enter the new
password");
                                 txtpwd.setText(password);
                                 String query=" update login set password='"+password+"' 
where login_id="+txtlog_id.getText();
                                 int i=statement.executeUpdate(query);
                                JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows 
succesfully");
                                loadLogins();
                          }
                          catch(SQLException updateException){
                                displaySQLErrors(updateException);
                          }
                           }
                                });
                    }
                    });
             view.addActionListener(new ActionListener(){
                    @Override
                    public void actionPerformed(ActionEvent arg0) {
                          // TODO Auto-generated method stub
                          p.removeAll();
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();
                Label view1=new Label("View logins");
                Font myFont = new Font("Serif",Font.BOLD,50);
                view1.setFont((myFont));
                viewButton=new JButton("View");
                p1=new JPanel();
                p1.add(view1);
                p2=new JPanel(new FlowLayout());
                p2.add(viewButton);
                p2.setBounds(220,220,75,35);
                p1.setBounds(130,20,300,100);
                p.add(p1);
                p.add(p2);
                p.setLayout(new BorderLayout());
                frame.add(p);
                          frame.setSize(600,600);
                          frame.validate();
                           viewButton.addActionListener(new ActionListener() {
                                 @Override
                                       public void actionPerformed(ActionEvent e) {
                                             // TODO Auto-generated method stub
                                        JFrame f;
                                           JTable j;
                                               f = new JFrame();
                                               f.setTitle("login details");
                                               DefaultTableModel model = new 
DefaultTableModel();
                                               j = new JTable(model);
                                              model.addColumn("login_id");
                                              model.addColumn("Name");
                                              model.addColumn("password");
                                              try {
```

```
                                                           rs=statement.executeQuery("select 
* from login");
                                                           while(rs.next()) {
                                                                  model.addRow(new 
Object[]{rs.getString("login_id"), rs.getString("Name"),rs.getString("password")});
                                                           }
                                                           }
                                                    catch(SQLException viewException) {
                                                           displaySQLErrors(viewException);
                                                    }
                                                    j.setEnabled(false);
                                               j.setBounds(30, 40, 150, 150);
                                               JScrollPane sp = new JScrollPane(j);
                                               f.add(sp);
                                               f.setSize(400, 400);

                                               f.setVisible(true);

                                           }
                                 });
                    }
             });
       }
}
```

Customer:

package Edairy;

import java.awt.BorderLayout;

import java.awt.Choice;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.*;

```
import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;


public class Customer {

        private static final long serialVersionUID = 1L;

        private JButton insertButton,deleteButton,updateButton,viewButton;

        private JPanel p1,p2,p3,p;

        private JLabel lblcustomer_id,lblcustomer_name,lblpno,lblemail,lbladd;

        private JTextField txtcustomer_id,txtcustomer_name,txtpno,txtemail;

        private JTextArea txtadd;

        private List customerIDList;

        Connection con;ResultSet rs;

        Statement statement;

        private JFrame frame;

        private JMenuItem insert,delete,update,view;

        public Customer(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)

        {
```

```
try
{
        Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
        System.err.println("Unable to find and load driver");
        System.exit(1);
}
connectToDB();
this.frame=frame;
this.insert=insert;
this.delete=delete;
this.update=update;
this.view=view;
//lbllog_id=new JLabel("Login_id");
lblcustomer_id=new JLabel("customer_id");
lblcustomer_name=new JLabel("customer_name");
lblpno=new JLabel("phone_no");
lblemail=new JLabel("email");
lbladd=new JLabel("address");
//txtlog_id = new JTextField(15);
txtcustomer_id=new JTextField(15);
txtcustomer_name=new JTextField(15);
txtpno=new JTextField(15);
txtemail=new JTextField(30);
txtadd=new JTextArea(15,15);
```

```
        this.p=p;
    }
    public void connectToDB()
  {
        try {
        Connection con=DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
        statement=con.createStatement();
        statement.executeUpdate("commit");
        }
        catch (SQLException connectException)
        {
          System.out.println(connectException.getMessage());
          System.out.println(connectException.getSQLState());
          System.out.println(connectException.getErrorCode());
          System.exit(1);
        }
  }
    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:     " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadcustomerIDs() {
        try {
            customerIDList.removeAll();
            rs=statement.executeQuery("select customer_id from Customer");
            while(rs.next()) {
                customerIDList.add(rs.getString("customer_id"));
            }
            }
        catch(SQLException e) {
            displaySQLErrors(e);
        }
    }
    public void buildGUI() {
        insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                insertButton=new JButton("Submit");
                txtcustomer_id.setText(null);
                txtcustomer_name.setText(null);
                txtpno.setText(null);
                txtemail.setText(null);
                txtadd.setText(null);
                //txtlog_id.setText(null);
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                //loadlogin();
                p1=new JPanel();
                 p1.setLayout(new GridLayout(5,2));
                 p1.add(lblcustomer_id);
                 p1.add(txtcustomer_id);
```

```
 p1.add(lblcustomer_name);
 p1.add(txtcustomer_name);
 p1.add(lblpno);
 p1.add(txtpno);
 p1.add(lblemail);
 p1.add(txtemail);
 p1.add(lbladd);
 p1.add(txtadd);
 //p1.setBackground(Color.pink) ;
 p3=new JPanel(new FlowLayout());
 p3.add(insertButton);
 //p1.add(txtf1);
 //p3.setBackground(Color.yellow);
p3.setBounds(160,450,75,35);
 p1.setBounds(50,50,350,320);
 p2 = new JPanel(new FlowLayout());
 customerIDList=new List(10);
 loadcustomerIDs();
 p2.add(customerIDList);//p2.setBackground(Color.cyan) ;
 p2.setBounds(330,150,350,180);
 p. add(p1);
 p.add(p3);
 p. add(p2);
 p.setLayout(new BorderLayout());
         frame.add(p);
         frame.setSize(600,600);
         frame.validate();
```

```
                insertButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
                    String query="INSERT INTO Customer VALUES('"+txtcustomer_id.getText()+"','"+txtcustomer_name.getText()+"','"+txtpno.getText()+"','"+txtemail.getText()+"','"+txtadd.getText()+"')";
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully")
                    loadcustomerIDs();
                    //loadlogin();
                }
                catch(SQLException insertException){
                    displaySQLErrors(insertException);
                }
                 }
                    });
            }
            });
        delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                deleteButton=new JButton("Delete");
                txtcustomer_id.setText(null);
                txtcustomer_name.setText(null);
                txtpno.setText(null);
                txtemail.setText(null);
```

```
txtadd.setText(null);
//txtlog_id.setText(null);
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
 p1.setLayout(new GridLayout(5,2));
 p1.add(lblcustomer_id);
 p1.add(txtcustomer_id);
 p1.add(lblcustomer_name);
 p1.add(txtcustomer_name);
 p1.add(lblpno);
 p1.add(txtpno);
 p1.add(lblemail);
 p1.add(txtemail);
 p1.add(lbladd);
 p1.add(txtadd);
 //p1.setBackground(Color.pink) ;
 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
 //p1.add(txtf1);
 //p3.setBackground(Color.yellow);
p3.setBounds(160,450,75,35);
 p1.setBounds(50,50,350,320);
 p2 = new JPanel(new FlowLayout());
 customerIDList=new List(10);
 loadcustomerIDs();
```

```
                                        p2.add(customerIDList);//p2.setBackground(Color.cyan) ;
                                        p2.setBounds(330,150,350,180);
                                        p. add(p1);
                                        p.add(p3);
                                        p. add(p2);
                                        p.setLayout(new BorderLayout());
                                                frame.add(p);
                                                frame.setSize(600,600);
                                                frame.validate();
                                        customerIDList.addItemListener(new ItemListener()
                                                {
                                                        public void itemStateChanged(ItemEvent e)
                                                        {
                                                                try
                                                                {
                                                                        rs=statement.executeQuery("select * from
Customer");
                                                                        while (rs.next())
                                                                        {
                                                                                if
(rs.getString("customer_id").equals(customerIDList.getSelectedItem()))
                                                                                break;
                                                                        }
                                                                        if (!rs.isAfterLast())
                                                                        {

        //txtlog_id.setText(rs.getString("Login_id"));

        txtcustomer_id.setText(rs.getString("customer_id"));
```

```
    txtcustomer_name.setText(rs.getString("customer_name"));
                                    txtpno.setText(rs.getString("phone_no"));
                                    txtemail.setText(rs.getString("email"));
                                    txtadd.setText(rs.getString("address"));
                                }
                            }
                            catch (SQLException selectException)
                            {
                                displaySQLErrors(selectException);
                            }
                        }
                    });
                deleteButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
                    int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                    if(a==JOptionPane.YES_OPTION){
                        String query="DELETE FROM customer WHERE
customer_id="+customerIDList.getSelectedItem();
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");loadcustomerIDs();
                    }
                }
                catch(SQLException deleteException){
                    displaySQLErrors(deleteException);
                }
```

```
                        }
                                });
                }
                });
        update.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub
                        updateButton = new JButton("Modify");
                        txtcustomer_id.setText(null);
                        txtcustomer_name.setText(null);
                        txtpno.setText(null);
                        txtemail.setText(null);
                        txtadd.setText(null);
                        p.removeAll();
                        frame.invalidate();
                        frame.validate();
                        frame.repaint();
                        p1=new JPanel();
                        p1.setLayout(new GridLayout(5,2));
                         p1.add(lblcustomer_id);
                         p1.add(txtcustomer_id);
                         p1.add(lblcustomer_name);
                         p1.add(txtcustomer_name);
                         p1.add(lblpno);
                         p1.add(txtpno);
                         p1.add(lblemail);
```

```
 p1.add(txtemail);
 p1.add(lbladd);
 p1.add(txtadd);
 //p1.setBackground(Color.pink) ;
 p3=new JPanel(new FlowLayout());
 p3.add(updateButton);
 //p1.add(txtf1);
 //p3.setBackground(Color.yellow);
p3.setBounds(160,450,75,35);
 p1.setBounds(50,50,350,320);
 p2 = new JPanel(new FlowLayout());
 customerIDList=new List(10);
 loadcustomerIDs();
 p2.add(customerIDList);//p2.setBackground(Color.cyan) ;
 p2.setBounds(330,150,350,180);
 p. add(p1);
 p.add(p3);
 p. add(p2);
 p.setLayout(new BorderLayout());
         frame.add(p);
         frame.setSize(600,600);
         frame.validate();
 customerIDList.addItemListener(new ItemListener()
         {
                 public void itemStateChanged(ItemEvent e)
                 {
                         try
                         {
                                rs=statement.executeQuery("select * from
Customer");
                                while (rs.next())
                                {
                                    if
(rs.getString("customer_id").equals(customerIDList.getSelectedItem()))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {

    //txtlog_id.setText(rs.getString("Login_id"));

    txtcustomer_id.setText(rs.getString("customer_id"));

    txtcustomer_name.setText(rs.getString("customer_name"));
                                    txtpno.setText(rs.getString("phone_no"));
                                    txtemail.setText(rs.getString("email"));
                                    txtadd.setText(rs.getString("address"));
                                }
                            }
                            catch (SQLException selectException)
                            {
                                displaySQLErrors(selectException);
                            }
                        }
                    });
                updateButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                        try {
                                int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");
                                if(a==JOptionPane.YES_OPTION){
                                String query="update Customer set
customer_name='"+txtcustomer_name.getText()+"',phone_no="+txtpno.getText()+",email='"+txtemail.getText()+"
',address='"+txtadd.getText()+"' WHERE customer_id="+customerIDList.getSelectedItem();
                                int i=statement.executeUpdate(query);
                                JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loadcustomerIDs();
                                }
                            }
                            catch(SQLException deleteException){
                                displaySQLErrors(deleteException);
                            }
                }
                    });
            }
            });
        view.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                Label view1=new Label("CustomerAccount view");
                //view1.setAlignment(Label.CENTER);
```

```
                        Font myFont = new Font("Serif",Font.BOLD,28);
                        view1.setFont((myFont));
                        viewButton=new JButton("View");
                        p1=new JPanel();
                        p1.add(view1);
                        p2=new JPanel(new FlowLayout());
p2.add(viewButton);
p2.setBounds(220,220,75,35);
p1.setBounds(130,20,300,100);
p.add(p1);
p.add(p2);
p.setLayout(new BorderLayout());
frame.add(p);
                        frame.setSize(600,600);
                        frame.validate();
                         viewButton.addActionListener(new ActionListener() {
                                 @Override
                                        public void actionPerformed(ActionEvent e) {
                                                 // TODO Auto-generated method stub
                                         JFrame f;
                                           JTable j;
                                             f = new JFrame();
                                             f.setTitle("login_User details");
                                             DefaultTableModel model = new DefaultTableModel();
                                             j = new JTable(model);
                                            model.addColumn("customer_id");
                                            model.addColumn("customer_name");
                                            model.addColumn("phone_no");
                                            model.addColumn("E-Mail");
                                            model.addColumn("Address");
                                            try {
                                                        rs=statement.executeQuery("select * from Customer");
                                                        while(rs.next()) {
                                                            model.addRow(new Object[]{rs.getString("customer_id"), rs.getString("customer_name"),rs.getString("phone_no"),rs.getString("email"),rs.getString("address"),});
                                                        }
                                                        }
                                                    catch(SQLException viewException) {
                                                        displaySQLErrors(viewException);
                                                    }
                                                    j.setEnabled(false);
                                             j.setBounds(30, 40, 300, 300);
                                             JScrollPane sp = new JScrollPane(j);
                                             f.add(sp);
                                             f.setSize(800, 400);
                                             f.setVisible(true);
                                           }
                                       });
                               }
                           });
                       }
                   }
```

Category:

```
package Edairy;
```

```
import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.*;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;


public class Category{

        private static final long serialVersionUID = 1L;

        private JButton insertButton,deleteButton,updateButton,viewButton;

        private JPanel p1,p2,p3,p;

        private JLabel lblcategory_id,lblcategory_name,lblprice,lblavailability,lblcode;
```

```
	private JTextField txtcategory_id,txtcategory_name,txtprice,txtavailability,txtcode;


	private List category_idList;
	Connection con;ResultSet rs;
	Statement statement;
	private JFrame frame;
	private JMenuItem insert,delete,update,view;
	public Category(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem view)
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
		this.frame=frame;
		this.insert=insert;
		this.delete=delete;
		this.update=update;
		this.view=view;
		lblcategory_id=new JLabel("Category ID");
		lblcategory_name=new JLabel("Category Name");
		lblprice=new JLabel("Price");
		lblavailability=new JLabel("Availability");
```

```
                lblcode = new JLabel("Code");
                txtcategory_id=new JTextField(15);
                txtcategory_name=new JTextField(15);
                txtprice=new JTextField(8);
                txtavailability=new JTextField(15);
                txtcode = new JTextField(15);
                this.p=p;
        }
        public void connectToDB()
  {
                try {
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
                statement=con.createStatement();
                statement.executeUpdate("commit");
                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
  }
        private void displaySQLErrors(SQLException e)
        {
                JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:     " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
```

```
}
public void loadcategoryIDs() {
        try {
                category_idList.removeAll();
                rs=statement.executeQuery("select category_id from Category");
                while(rs.next()) {
                        category_idList.add(rs.getString("category_id"));
                }
                }
        catch(SQLException e) {
                displaySQLErrors(e);
        }
}
public void buildGUI() {
        insert.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub
                        insertButton=new JButton("Submit");
                        txtcategory_id.setText(null);
                        txtcategory_name.setText(null);
                        txtprice.setText(null);
                        txtavailability.setText(null);
                        txtcode.setText(null);
                        p.removeAll();
                        frame.invalidate();
                        frame.validate();
frame.repaint();
p1=new JPanel();
 p1.setLayout(new GridLayout(5,2));
 p1.add(lblcategory_id);
 p1.add(txtcategory_id);
 p1.add(lblcategory_name);
 p1.add(txtcategory_name);
 p1.add(lblprice);
 p1.add(txtprice);
 p1.add(lblavailability);
 p1.add(txtavailability);
 p1.add(lblcode);
 p1.add(txtcode);
 p3=new JPanel(new FlowLayout());
 p3.add(insertButton);
 //p1.add(txtf1);
 //p3.setBackground(Color.yellow);
 p1.setBounds(100,80,300,250);p3.setBounds(200,400,75,35);
 //p1.setBackground(Color.pink) ;
 p2 = new JPanel(new FlowLayout());
            category_idList=new List(10);
            loadcategoryIDs();
            p2.add(category_idList);//p2.setBackground(Color.cyan) ;
            p2.setBounds(320,150,350,180);
 p. add(p1);p.add(p3);
 p. add(p2);
 p.setLayout(new BorderLayout());
           frame.add(p);
```

```
                    frame.setSize(600,600);
                    frame.validate();
                insertButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
                    String query="INSERT INTO Category VALUES('"+txtcategory_id.getText()+"','"+txtcategory_name.getText()+"','"+txtprice.getText()+"','"+txtcode.getText()+"','"+txtavailability.getText()+"')";
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");loadcategoryIDs();
                }
                catch(SQLException insertException){
                    displaySQLErrors(insertException);
                }
                }
                    });
            }
            });
        delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                deleteButton=new JButton("Delete");
                txtcategory_id.setText(null);
                txtcategory_name.setText(null);
                txtprice.setText(null);
                txtavailability.setText(null);
```

```
txtcode.setText(null);
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
p1.setLayout(new GridLayout(5,2));
 p1.add(lblcategory_id);
 p1.add(txtcategory_id);
 p1.add(lblcategory_name);
 p1.add(txtcategory_name);
 p1.add(lblprice);
 p1.add(txtprice);
 p1.add(lblavailability);
 p1.add(txtavailability);
 p1.add(lblcode);
 p1.add(txtcode);
 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
 //p1.add(txtf1);
 //p3.setBackground(Color.yellow);
 p1.setBounds(100,80,300,250);p3.setBounds(200,400,75,35);
 //p1.setBackground(Color.pink) ;
 p2 = new JPanel(new FlowLayout());
            category_idList=new List(10);
            loadcategoryIDs();
            p2.add(category_idList);//p2.setBackground(Color.cyan) ;
            p2.setBounds(320,150,350,180);
```

```
                p. add(p1);p.add(p3);
                p. add(p2);
                p.setLayout(new BorderLayout());
                    frame.add(p);
                    frame.setSize(600,600);
                    frame.validate();
                 category_idList.addItemListener(new ItemListener()
                    {
                        public void itemStateChanged(ItemEvent e)
                        {
                            try
                            {
                                rs=statement.executeQuery("select * from
Category");
                                while (rs.next())
                                {
                                    if
(rs.getString("category_id").equals(category_idList.getSelectedItem()))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {

    txtcategory_id.setText(rs.getString("category_id"));

    txtcategory_name.setText(rs.getString("category_name"));
                                    txtprice.setText(rs.getString("price"));
                                    txtcode.setText(rs.getString("code"));

    txtavailability.setText(rs.getString("Availability"));
```

```
                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
     deleteButton.addActionListener(new ActionListener() {
         @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
    try {
        int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
        if(a==JOptionPane.YES_OPTION){
        String query="DELETE FROM Category WHERE category_id="+category_idList.getSelectedItem();
        int i=statement.executeUpdate(query);
        JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loadcategoryIDs();
        }
    }
    catch(SQLException deleteException){
        displaySQLErrors(deleteException);
    }}
});
}
});
```

```
update.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        updateButton = new JButton("Modify");
        txtcategory_id.setText(null);
        txtcategory_name.setText(null);
        txtprice.setText(null);
        txtavailability.setText(null);
        txtcode.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        p1=new JPanel();
        p1.setLayout(new GridLayout(5,2));
         p1.add(lblcategory_id);
         p1.add(txtcategory_id);
         p1.add(lblcategory_name);
         p1.add(txtcategory_name);
         p1.add(lblprice);
         p1.add(txtprice);
         p1.add(lblavailability);
         p1.add(txtavailability);
         p1.add(lblcode);
         p1.add(txtcode);
         p3=new JPanel(new FlowLayout());
         p3.add(updateButton);
```

```
                                //p1.add(txtf1);
                                //p3.setBackground(Color.yellow);
                                p1.setBounds(100,80,300,250);p3.setBounds(200,400,75,35);
                                //p1.setBackground(Color.pink) ;
                                p2 = new JPanel(new FlowLayout());
                                        category_idList=new List(10);
                                        loadcategoryIDs();
                                        p2.add(category_idList);//p2.setBackground(Color.cyan) ;
                                        p2.setBounds(320,150,350,180);
                                p. add(p1);p.add(p3);
                                p. add(p2);
                                p.setLayout(new BorderLayout());


                                        frame.add(p);
                                        frame.setSize(600,600);
                                        frame.validate();
                                category_idList.addItemListener(new ItemListener()
                                        {
                                                public void itemStateChanged(ItemEvent e)
                                                {
                                                        try
                                                        {
                                                                rs=statement.executeQuery("select * from
Category");
                                                                while (rs.next())
                                                                {
                                                                        if
(rs.getString("category_id").equals(category_idList.getSelectedItem()))
                                                                        break;
```

```
                                                                  }
                                                                  if (!rs.isAfterLast())
                                                                  {

        txtcategory_id.setText(rs.getString("category_id"));

        txtcategory_name.setText(rs.getString("category_name"));
                                                                          txtprice.setText(rs.getString("price"));
                                                                          txtcode.setText(rs.getString("code"));

        txtavailability.setText(rs.getString("Availability"));
                                                                  }
                                                          }
                                                          catch (SQLException selectException)
                                                          {
                                                                  displaySQLErrors(selectException);
                                                          }
                                                  }
                                          });
                                  updateButton.addActionListener(new ActionListener() {
                                          @Override
                                                  public void actionPerformed(ActionEvent e) {
                                                          // TODO Auto-generated method stub
                                                  try {
                                                                  int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");
                                                                  if(a==JOptionPane.YES_OPTION){
                                                                  String query="update Category set
category_name='"+txtcategory_name.getText()+"',price='"+txtprice.getText()+"',code='"+txtcode.getText()+"',Avai
lability='"+txtavailability.getText()+"' WHERE category_id="+category_idList.getSelectedItem();
```

```
                                                                int i=statement.executeUpdate(query);
                                                                JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loadcategoryIDs();
                                                                }
                                                        }
                                                        catch(SQLException deleteException){
                                                                displaySQLErrors(deleteException);
                                                        }
                                  }
                                         });
                        }
                        });
                view.addActionListener(new ActionListener(){
                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub
                                p.removeAll();
                                frame.invalidate();
                                frame.validate();
                                frame.repaint();
                                Label view1=new Label("Category view");
                                //view1.setAlignment(Label.CENTER);
                                Font myFont = new Font("Serif",Font.BOLD,40);
                                view1.setFont((myFont));
                                viewButton=new JButton("View");
                                p1=new JPanel();
                                p1.add(view1);
```

```
                                    p2=new JPanel(new FlowLayout());
          p2.add(viewButton);
          p2.setBounds(220,220,75,35);
          p1.setBounds(130,20,300,100);
          p.add(p1);
          p.add(p2);
          p.setLayout(new BorderLayout());
          frame.add(p);
                                    frame.setSize(600,600);
                                    frame.validate();
                                     viewButton.addActionListener(new ActionListener() {
                                              @Override
                                                      public void actionPerformed(ActionEvent e) {
                                                               // TODO Auto-generated method stub
                                                       JFrame f;
                                                         JTable j;
                                                           f = new JFrame();
                                                           f.setTitle("Category details");
                                                           DefaultTableModel model = new DefaultTableModel();
                                                           j = new JTable(model);
                                                          model.addColumn("Category id");
                                                          model.addColumn("Category Name");
                                                          model.addColumn("Price");
                                                          model.addColumn("code");
                                                          model.addColumn("availability");
                                                         try {
                                                                                 rs=statement.executeQuery("select *
from Category");
```

```
                                    while(rs.next()) {
                                        model.addRow(new Object[]{
rs.getString("category_id"),
rs.getString("category_name"),rs.getString("price"),rs.getString("code"),rs.getString("Availability")});
                                    }
                                    }
                                catch(SQLException viewException) {
                                    displaySQLErrors(viewException);
                                }
                                j.setEnabled(false);
                          j.setBounds(30, 40, 300, 300);
                          JScrollPane sp = new JScrollPane(j);
                          f.add(sp);
                          f.setSize(800, 400);
                          f.setVisible(true);
                         }
                    });
            }
        });
    }
}
```

Company:

```
package Edairy;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Label;
import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.*;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;


public class Company{

        private static final long serialVersionUID = 1L;

        private JButton insertButton,deleteButton,updateButton,viewButton;

        private JPanel p1,p2,p3,p;

        private JLabel lblcompany_id,lblcompany_name;

        private JTextField txtcompany_id,txtcompany_name;

        private List CompanyIDList;

        Connection con;ResultSet rs;

        Statement statement;

        private JFrame frame;

        private JMenuItem insert,delete,update,view;
```

```
    public Company(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
    {


        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        this.frame=frame;
        this.insert=insert;
        this.delete=delete;
        this.update=update;
        this.view=view;
        lblcompany_id=new JLabel("Company ID");
        lblcompany_name=new JLabel("company_name");
        txtcompany_id=new JTextField(15);
        txtcompany_name=new JTextField(15);
        this.p=p;
    }
    public void connectToDB()
  {
        try {
        Connection con=DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
        statement=con.createStatement();
        statement.executeUpdate("commit");
        }
        catch (SQLException connectException)
        {
          System.out.println(connectException.getMessage());
          System.out.println(connectException.getSQLState());
          System.out.println(connectException.getErrorCode());
          System.exit(1);
        }
  }
    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:     " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadCompanyIDs() {
        try {
            CompanyIDList.removeAll();
            rs=statement.executeQuery("select company_id from company");
            while(rs.next()) {
                      CompanyIDList.add(rs.getString("company_id"));
            }
            }
        catch(SQLException e) {
            displaySQLErrors(e);
        }
    }
```

```
    public void buildGUI() {
        insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                insertButton=new JButton("Submit");
                txtcompany_id.setText(null);
                txtcompany_name.setText(null);
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                p1=new JPanel();
                 p1.setLayout(new GridLayout(2,2));
                 p1.add(lblcompany_id);
                 p1.add(txtcompany_id);
                 p1.add(lblcompany_name);
                 p1.add(txtcompany_name);
                 p3=new JPanel(new FlowLayout());
                 p3.add(insertButton);
                p3.setBounds(200,220,75,35); p1.setBounds(100,80,300,100);
                 p2 = new JPanel(new FlowLayout());
                 CompanyIDList=new List(10);
                 loadCompanyIDs();
                 p2.add(CompanyIDList);//p2.setBackground(Color.cyan) ;

                     p2.setBounds(70,320,350,180);
                 p. add(p1);p.add(p3);
                 p. add(p2);
                 p.setLayout(new BorderLayout());
                    frame.add(p);
                    frame.setSize(600,600);
                    frame.validate();
                 insertButton.addActionListener(new ActionListener() {
                     @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
                    String query="INSERT INTO company
VALUES("+txtcompany_id.getText()+",'"+txtcompany_name.getText()+"')";
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");
                    loadCompanyIDs();
                }
                catch(SQLException insertException){
                    displaySQLErrors(insertException);
                }
                 }
                    });
            }
            });
        delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                deleteButton=new JButton("Delete");
                txtcompany_id.setText(null);
                txtcompany_name.setText(null);
```

```
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
p1.setLayout(new GridLayout(2,2));
 p1.add(lblcompany_id);
 p1.add(txtcompany_id);
 p1.add(lblcompany_name);
 p1.add(txtcompany_name);
 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
p3.setBounds(200,220,75,35); p1.setBounds(100,80,300,100);
 p2 = new JPanel(new FlowLayout());
 CompanyIDList=new List(10);
 loadCompanyIDs();
 p2.add(CompanyIDList);
             p2.setBounds(70,320,350,180);
 p. add(p1);p.add(p3);
 p. add(p2);
 p.setLayout(new BorderLayout());
            frame.add(p);
            frame.setSize(600,600);
            frame.validate();
 CompanyIDList.addItemListener(new ItemListener()
            {
                        public void itemStateChanged(ItemEvent e)
                        {
                            try
                            {
                                rs=statement.executeQuery("select * from
company");
                                while (rs.next())
                                {
                                    if
(rs.getString("company_id").equals(CompanyIDList.getSelectedItem()))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {

    txtcompany_id.setText(rs.getString("company_id"));

    txtcompany_name.setText(rs.getString("company_name"));
                                }
                            }
                            catch (SQLException selectException)
                            {
                                displaySQLErrors(selectException);
                            }
                        }
                    });
                deleteButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
                    int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                    if(a==JOptionPane.YES_OPTION){
                    String query="DELETE FROM company WHERE
company_id="+CompanyIDList.getSelectedItem();
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");loadCompanyIDs();
                    }
                }
                catch(SQLException deleteException){
                    displaySQLErrors(deleteException);
                }
                 }
                    });
            }
            });
        update.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                updateButton = new JButton("Modify");
                txtcompany_id.setText(null);
                txtcompany_name.setText(null);
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                p1=new JPanel();
                p1.setLayout(new GridLayout(2,2));
                 p1.add(lblcompany_id);
                 p1.add(txtcompany_id);
                 p1.add(lblcompany_name);
                 p1.add(txtcompany_name);
                 p3=new JPanel(new FlowLayout());
                 p3.add(updateButton);
                p3.setBounds(200,220,75,35); p1.setBounds(100,80,300,100);
                 p2 = new JPanel(new FlowLayout());
                 CompanyIDList=new List(10);
                 loadCompanyIDs();
                 p2.add(CompanyIDList);
                     p2.setBounds(70,320,350,180);
                 p. add(p1);p.add(p3);
                 p. add(p2);
                 p.setLayout(new BorderLayout());
                    frame.add(p);
                    frame.setSize(600,600);
                    frame.validate();
                 CompanyIDList.addItemListener(new ItemListener()
                    {
                        public void itemStateChanged(ItemEvent e)
                        {
                            try
                            {
                                rs=statement.executeQuery("select * from
company");
                                while (rs.next())
                                {
                                    if
(rs.getString("company_id").equals(CompanyIDList.getSelectedItem()))
```

```
                                                        break;
                                                    }
                                                    if (!rs.isAfterLast())
                                                    {

        txtcompany_id.setText(rs.getString("company_id"));

        txtcompany_name.setText(rs.getString("company_name"));
                                                    }
                                                }
                                                catch (SQLException selectException)
                                                {
                                                    displaySQLErrors(selectException);
                                                }
                                        }
                                });
                            updateButton.addActionListener(new ActionListener() {
                                @Override
                                        public void actionPerformed(ActionEvent e) {
                                                // TODO Auto-generated method stub
                                         try {
                                                    int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");
                                                    if(a==JOptionPane.YES_OPTION){
                                                    String query="update company set
company_name='"+txtcompany_name.getText()+"' WHERE company_id="+CompanyIDList.getSelectedItem();

                                                    int i=statement.executeUpdate(query);
                                                    JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loadCompanyIDs();
                                                    }
```

```
                        }
                        catch(SQLException deleteException){
                            displaySQLErrors(deleteException);
                        }
            }
                }});
        }
        });
    view.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();
            Label view1=new Label("Company view");
            Font myFont = new Font("Serif",Font.BOLD,40);
            view1.setFont((myFont));
            viewButton=new JButton("View");
            p1=new JPanel();
            p1.add(view1);
            p2=new JPanel(new FlowLayout());
p2.add(viewButton);
p2.setBounds(220,220,75,35);
p1.setBounds(130,20,300,100);
p.add(p1);
p.add(p2);
```

```
          p.setLayout(new BorderLayout());
          frame.add(p);
                                    frame.setSize(600,600);
                                    frame.validate();
                                     viewButton.addActionListener(new ActionListener() {
                                              @Override
                                                      public void actionPerformed(ActionEvent e) {
                                                               // TODO Auto-generated method stub
                                                       JFrame f;
                                                         JTable j;
                                                           f = new JFrame();
                                                           f.setTitle("Company Details");
                                                           DefaultTableModel model = new DefaultTableModel();
                                                           j = new JTable(model);
                                                          model.addColumn("Company ID");
                                                          model.addColumn("company_name");

                                                          try {
                                                                                 rs=statement.executeQuery("select *
from company");
                                                                                 while(rs.next()) {
                                                                                           model.addRow(new
Object[]{rs.getString("company_id"),rs.getString("company_name")});
                                                                                 }
                                                                                 }
                                                                        catch(SQLException viewException) {
                                                                                 displaySQLErrors(viewException);
                                                                        }
                                                                        j.setEnabled(false);
                                                           j.setBounds(30, 40, 150, 150);
```

```
                          JScrollPane sp = new JScrollPane(j);
                          f.add(sp);
                          f.setSize(400, 400);
                          f.setVisible(true);
                         }
                    });
            }
        });
    }
}
```

Details:

```
package Edairy;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;


public class Details{

        private static final long serialVersionUID = 1L;

        private JButton insertButton,deleteButton,updateButton,viewButton;

        private JPanel p1,p2,p3,p;

        private JLabel lblorder_id,lbltot,lblstatus;

        private JTextField txtorder_id,txttot,txtstatus;

        private List OrderIDList;

        Connection con;ResultSet rs;

        Statement statement;

        private JFrame frame;

        private JMenuItem insert,delete,update,view;

        public Details(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem
view)

        {

                try

                {

                        Class.forName("oracle.jdbc.driver.OracleDriver");

                }

                catch (Exception e)

                {

                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connectToDB();
                this.frame=frame;
                this.insert=insert;
                this.delete=delete;
                this.update=update;
                this.view=view;
                lblorder_id=new JLabel("Order ID");
                lbltot=new JLabel("Amount");
                lblstatus=new JLabel("Status");
                txtorder_id=new JTextField(15);
                txttot=new JTextField(15);
                txtstatus=new JTextField(8);
                this.p=p;
        }
        public void connectToDB()
 {
                try {
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","joshua","vasavi");
                statement=con.createStatement();
                statement.executeUpdate("commit");
                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
```

```
          System.out.println(connectException.getErrorCode());
          System.exit(1);
        }
  }
    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:     " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadOrderIDs() {
        try {
            OrderIDList.removeAll();
            rs=statement.executeQuery("select order_id from details");
            while(rs.next()) {
                OrderIDList.add(rs.getString("order_id"));
            }
            }
        catch(SQLException e) {
            displaySQLErrors(e);
        }
    }
    public void buildGUI() {
        insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                insertButton=new JButton("Submit");
                txtorder_id.setText(null);
                txttot.setText(null);
```

```
txtstatus.setText(null);
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
 p1.setLayout(new GridLayout(3,2));
 p1.add(lblorder_id);
 p1.add(txtorder_id);
 p1.add(lbltot);
 p1.add(txttot);
 p1.add(lblstatus);
 p1.add(txtstatus);
 p3=new JPanel(new FlowLayout());
 p3.add(insertButton);
p3.setBounds(200,320,75,35); p1.setBounds(100,75,300,200);
 p2 = new JPanel(new FlowLayout());
 OrderIDList=new List(10);
 loadOrderIDs();
 p2.add(OrderIDList);//p2.setBackground(Color.cyan) ;
              p2.setBounds(300,100,350,180);
 p. add(p1);p.add(p3);
 p. add(p2);
 p.setLayout(new BorderLayout());
             frame.add(p);
             frame.setSize(600,600);
             frame.validate();
 insertButton.addActionListener(new ActionListener() {
```

```
                     @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
                    String query="INSERT INTO details 
VALUES("+txtorder_id.getText()+","+txttot.getText()+",'"+txtstatus.getText()+"')";

                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");
                    loadOrderIDs();
                }
                catch(SQLException insertException){
                    displaySQLErrors(insertException);
                }
                 }
                    });
            }
            });
        delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                deleteButton=new JButton("Delete");
                txtorder_id.setText(null);
                txttot.setText(null);
                txtstatus.setText(null);
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                p1=new JPanel();
                p1.setLayout(new GridLayout(3,2));
                 p1.add(lblorder_id);
                 p1.add(txtorder_id);
                 p1.add(lbltot);
                 p1.add(txttot);
                 p1.add(lblstatus);
                 p1.add(txtstatus);
                 p3=new JPanel(new FlowLayout());
                 p3.add(deleteButton);
                p3.setBounds(200,320,75,35); p1.setBounds(100,75,300,200);
                 p2 = new JPanel(new FlowLayout());
                 OrderIDList=new List(10);
                 loadOrderIDs();
                 p2.add(OrderIDList);//p2.setBackground(Color.cyan) ;
                     p2.setBounds(300,100,350,180);
                 p. add(p1);p.add(p3);
                 p. add(p2);
                 p.setLayout(new BorderLayout());
                    frame.add(p);
                    frame.setSize(600,600);
                    frame.validate();
                 OrderIDList.addItemListener(new ItemListener()
                    {
                        public void itemStateChanged(ItemEvent e)
                        {
                            try
```

```
                            {
                                rs=statement.executeQuery("select * from
details");
                                while (rs.next())
                                {
                                    if
(rs.getString("order_id").equals(OrderIDList.getSelectedItem()))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {

    txtorder_id.setText(rs.getString("order_id"));
                                    txttot.setText(rs.getString("total_amt"));
                                    txtstatus.setText(rs.getString("status"));
                                }
                            }
                            catch (SQLException selectException)
                            {
                                displaySQLErrors(selectException);
                            }
                        }
                    });

                deleteButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                try {
```

```
                                                int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                                                if(a==JOptionPane.YES_OPTION){
                                                String query="DELETE FROM details WHERE Order_ID="+OrderIDList.getSelectedItem();

                                                int i=statement.executeUpdate(query);
                                                JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loadOrderIDs();
                                                }
                                        }
                                        catch(SQLException deleteException){
                                                displaySQLErrors(deleteException);
                                        }
                                         }
                                                });
                                }
                                });
                        update.addActionListener(new ActionListener() {
                                @Override
                                public void actionPerformed(ActionEvent arg0) {
                                        // TODO Auto-generated method stub
                                        updateButton = new JButton("Modify");
                                        txtorder_id.setText(null);
                                        txttot.setText(null);
                                        txtstatus.setText(null);
                                        p.removeAll();
                                        frame.invalidate();
                                        frame.validate();
                                        frame.repaint();
```

```
p1=new JPanel();
p1.setLayout(new GridLayout(3,2));
 p1.add(lblorder_id);
 p1.add(txtorder_id);
 p1.add(lbltot);
 p1.add(txttot);
 p1.add(lblstatus);
 p1.add(txtstatus);
 p3=new JPanel(new FlowLayout());
 p3.add(updateButton);
p3.setBounds(200,320,75,35); p1.setBounds(100,75,300,200);
 p2 = new JPanel(new FlowLayout());
 OrderIDList=new List(10);
 loadOrderIDs();
 p2.add(OrderIDList);//p2.setBackground(Color.cyan) ;
              p2.setBounds(300,100,350,180);
 p. add(p1);p.add(p3);
 p. add(p2);
 p.setLayout(new BorderLayout());
             frame.add(p);
             frame.setSize(600,600);
             frame.validate();
 OrderIDList.addItemListener(new ItemListener()
             {
                         public void itemStateChanged(ItemEvent e)
                         {
                                     try
                                     {
```

```
                                rs=statement.executeQuery("select * from
details");
                                while (rs.next())
                                {
                                    if
(rs.getString("order_id").equals(OrderIDList.getSelectedItem()))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {

    txtorder_id.setText(rs.getString("order_id"));
                                    txttot.setText(rs.getString("total_amt"));
                                    txtstatus.setText(rs.getString("status"));
                                }
                            }
                            catch (SQLException selectException)
                            {
                                displaySQLErrors(selectException);
                            }
                        }
                    });
                updateButton.addActionListener(new ActionListener() {
                    @Override
                        public void actionPerformed(ActionEvent e) {
                            // TODO Auto-generated method stub
                         try {
                                int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");
                                if(a==JOptionPane.YES_OPTION){
```

```
                                String query="update details set total_amt="+txttot.getText()+",status='"+txtstatus.getText()+"' WHERE order_ID="+OrderIDList.getSelectedItem();


                                int i=statement.executeUpdate(query);
                                JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadOrderIDs();
                                }

                            }
                            catch(SQLException deleteException){
                                displaySQLErrors(deleteException);
                            }
                }
                    });
            }
            });
        view.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                Label view1=new Label("Details view");
                Font myFont = new Font("Serif",Font.BOLD,50);
                view1.setFont((myFont));
                viewButton=new JButton("View");
                p1=new JPanel();
                p1.add(view1);
```

```
                            p2=new JPanel(new FlowLayout());
        p2.add(viewButton);
        p2.setBounds(220,220,75,35);
        p1.setBounds(130,20,300,100);
        p.add(p1);
        p.add(p2);
        p.setLayout(new BorderLayout());
        frame.add(p);
                            frame.setSize(600,600);
                            frame.validate();
                             viewButton.addActionListener(new ActionListener() {
                                    @Override
                                           public void actionPerformed(ActionEvent e) {
                                                  // TODO Auto-generated method stub
                                           JFrame f;
                                             JTable j;
                                               f = new JFrame();
                                               f.setTitle("Details ");
                                               DefaultTableModel model = new DefaultTableModel();
                                               j = new JTable(model);
                                              model.addColumn("Order ID");
                                              model.addColumn("Amount");
                                              model.addColumn("Status");
                                              try {

                                                                rs=statement.executeQuery("select * from details");
                                                                while(rs.next()) {
```

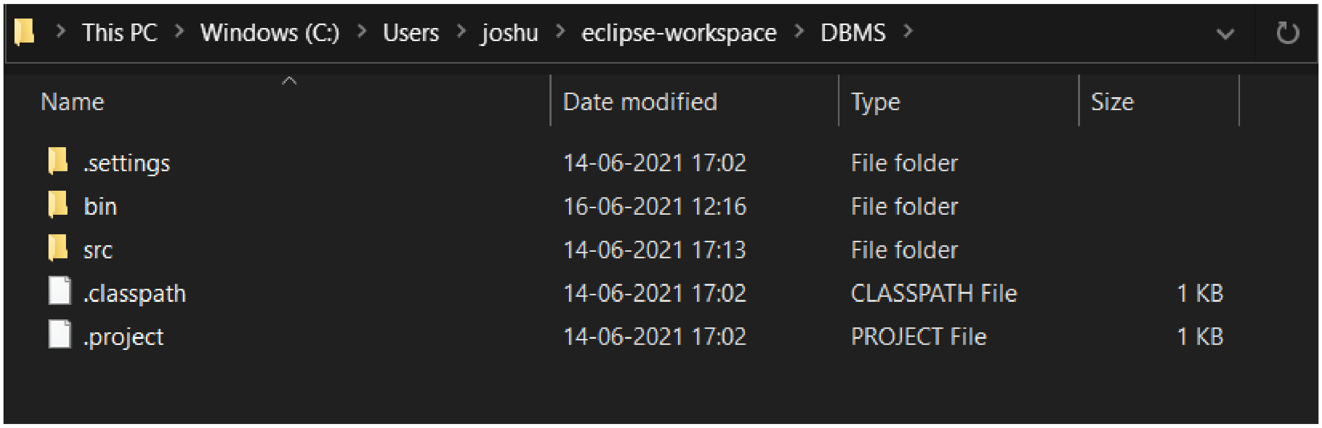
```
                                        model.addRow(new
Object[]{rs.getString("order_id"), rs.getString("total_amt"),rs.getString("status")});
                                    }
                                    }
                                catch(SQLException viewException) {
                                    displaySQLErrors(viewException);
                                }
                                j.setEnabled(false);
                            j.setBounds(30, 40, 150, 150);
                            JScrollPane sp = new JScrollPane(j);
                            f.add(sp);
                            f.setSize(400, 400);
                            f.setVisible(true);
                        }
                    });
            }
        });
    }
}
```

Github links and folder structure:

Link:

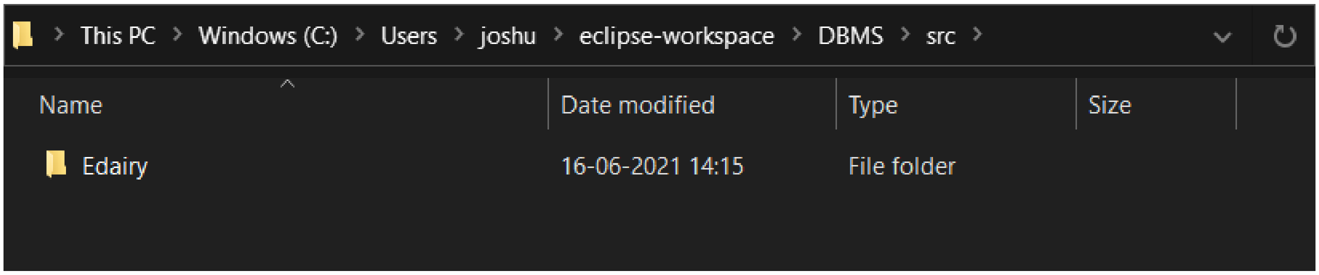
https://github.com/joshua-peter/DBMS---Project

Folder structure:



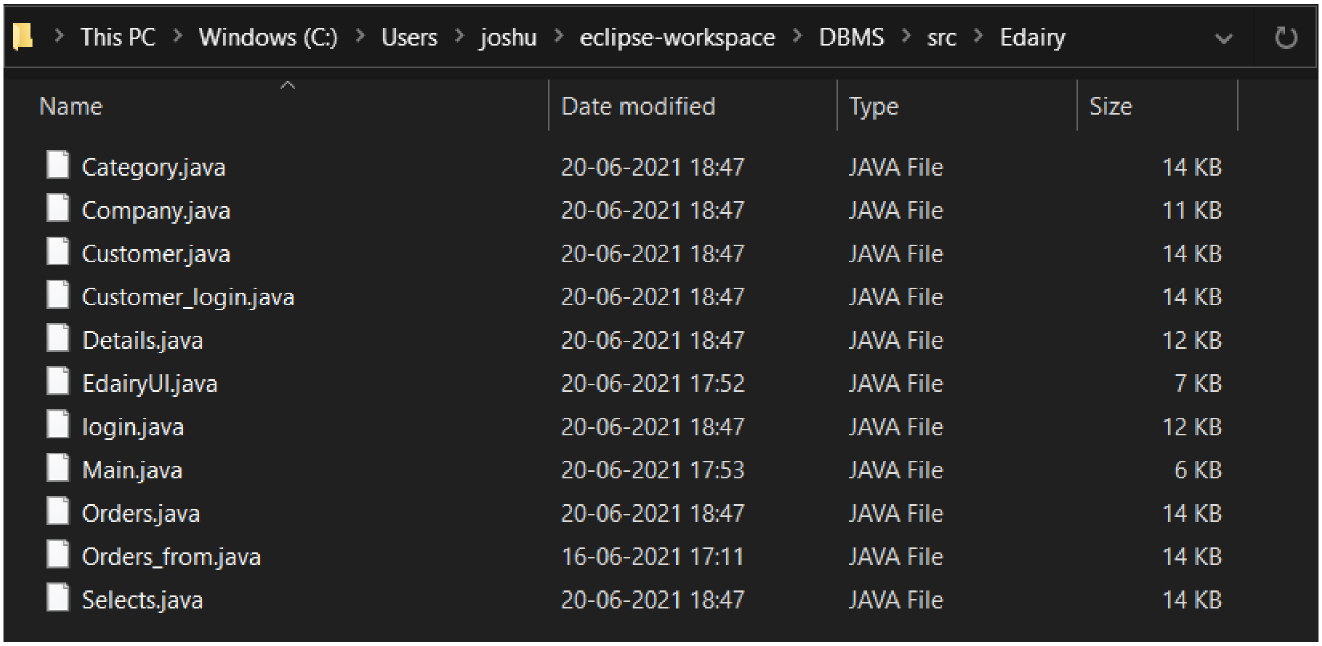
This PC > Windows (C:) > Users > joshu > eclipse-workspace > DBMS >

Name	Date modified	Type	Size
.settings	14-06-2021 17:02	File folder	
bin	16-06-2021 12:16	File folder	
src	14-06-2021 17:13	File folder	
.classpath	14-06-2021 17:02	CLASSPATH File	1 KB
.project	14-06-2021 17:02	PROJECT File	1 KB



This PC > Windows (C:) > Users > joshu > eclipse-workspace > DBMS > src >

Name	Date modified	Type	Size
Edairy	16-06-2021 14:15	File folder	

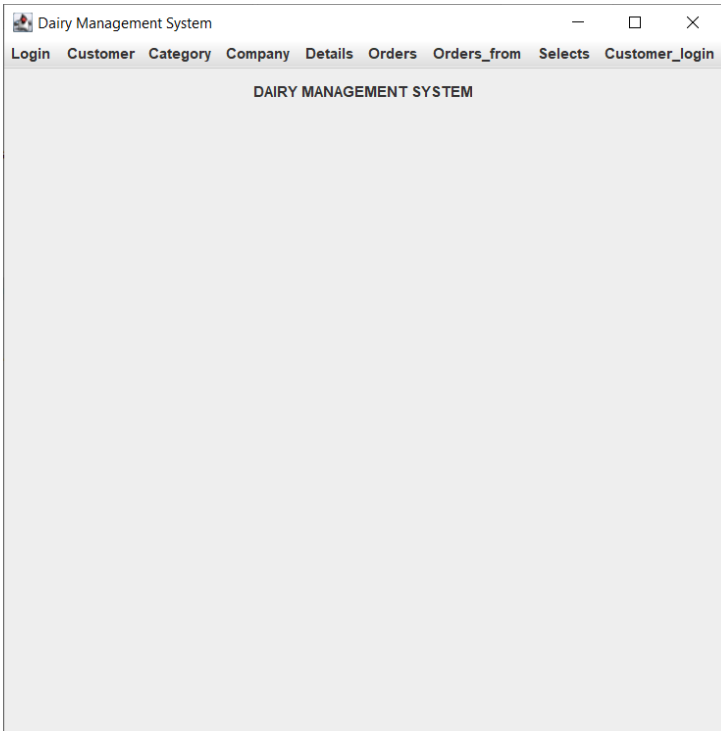


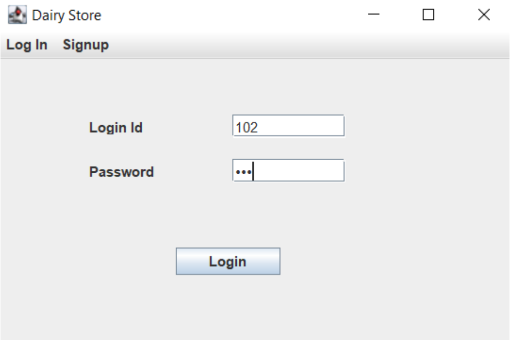
This PC > Windows (C:) > Users > joshu > eclipse-workspace > DBMS > src > Edairy

Name	Date modified	Type	Size
Category.java	20-06-2021 18:47	JAVA File	14 KB
Company.java	20-06-2021 18:47	JAVA File	11 KB
Customer.java	20-06-2021 18:47	JAVA File	14 KB
Customer_login.java	20-06-2021 18:47	JAVA File	14 KB
Details.java	20-06-2021 18:47	JAVA File	12 KB
EdairyUI.java	20-06-2021 17:52	JAVA File	7 KB
login.java	20-06-2021 18:47	JAVA File	12 KB
Main.java	20-06-2021 17:53	JAVA File	6 KB
Orders.java	20-06-2021 18:47	JAVA File	14 KB
Orders_from.java	16-06-2021 17:11	JAVA File	14 KB
Selects.java	20-06-2021 18:47	JAVA File	14 KB

TESTING

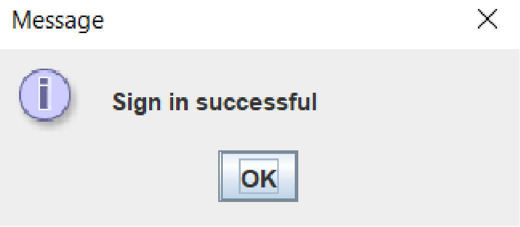
OUTPUT SCREENSHOTS:



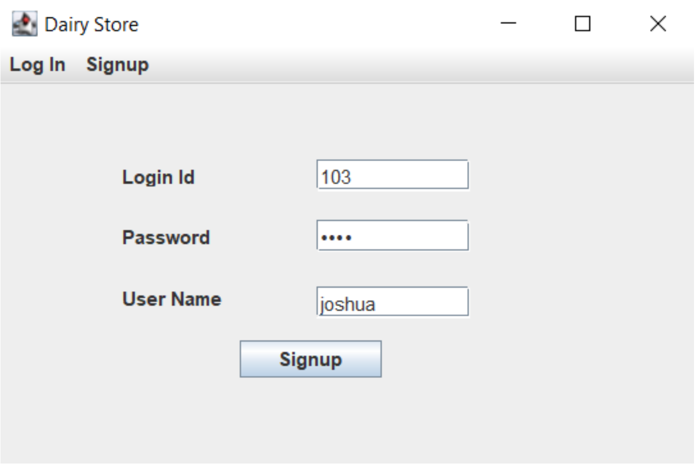


Login Id 102

Password •••



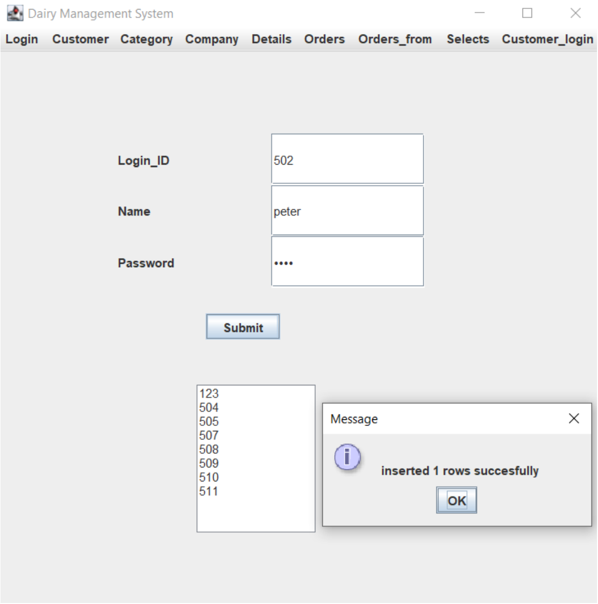
Sign in successful



Login Id 103

Password ••••

User Name joshua



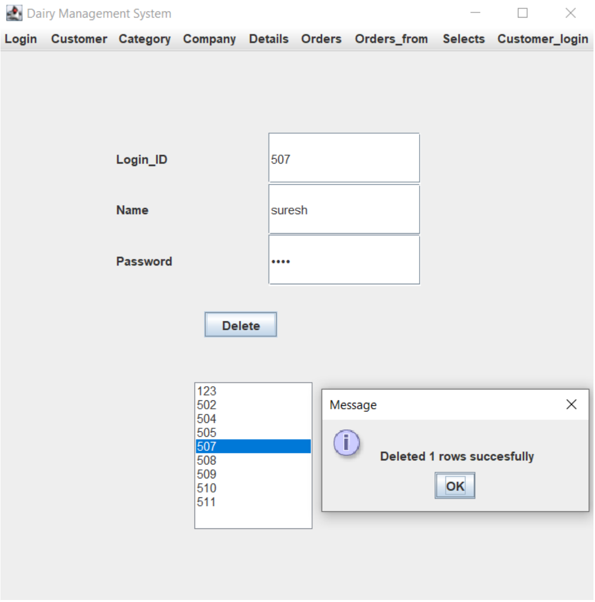
Login_ID 502

Name peter

Password ••••

123
504
505
507
508
509
510
511

inserted 1 rows succesfully



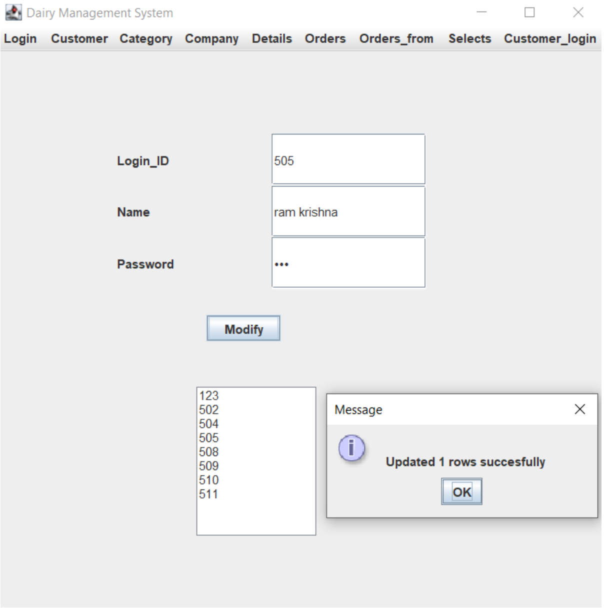
Login_ID 507

Name suresh

Password ••••

123
502
504
505
507
508
509
510
511

Deleted 1 rows succesfully

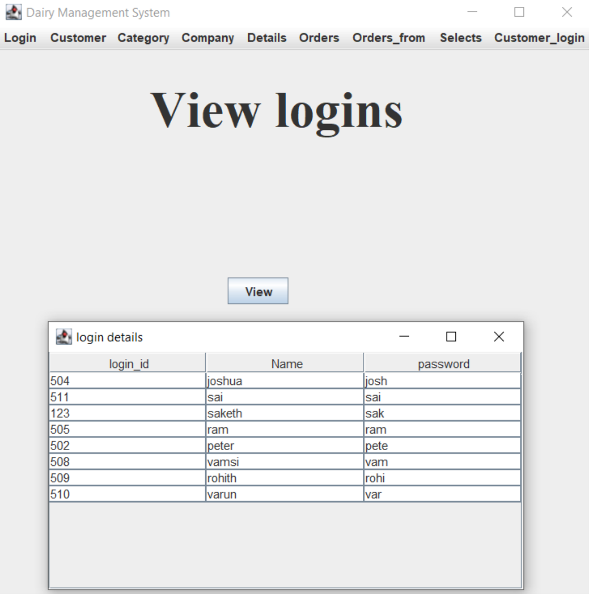

Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
123
502
504
505
508
509
510
511
Message
OK

Login_ID: 505

Name: ram krishna

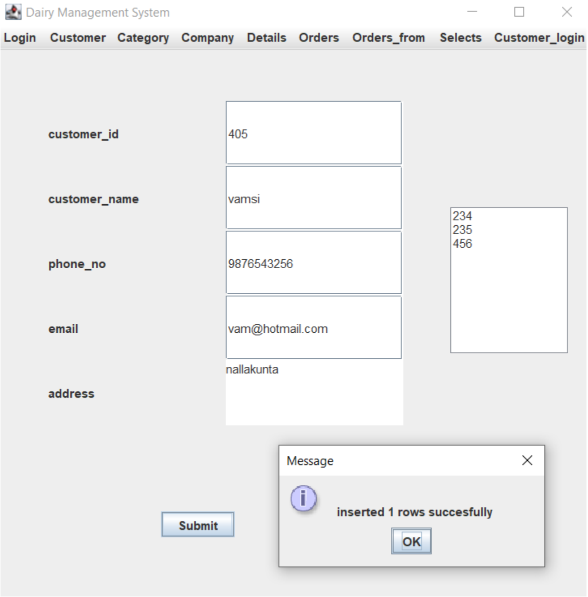
Password: •••

Updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View
login details

View logins

login_id	Name	password
504	joshua	josh
511	sai	sai
123	saketh	sak
505	ram	ram
502	peter	pete
508	vamsi	vam
509	rohith	rohi
510	varun	var


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
234
235
456
Submit
Message
OK

customer_id: 405

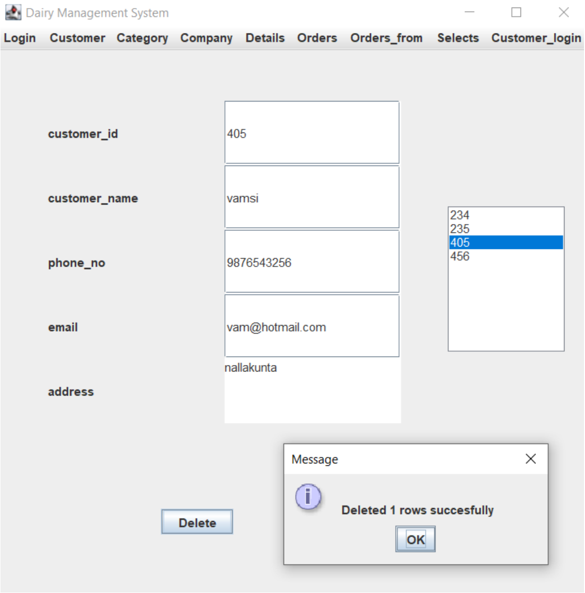
customer_name: vamsi

phone_no: 9876543256

email: vam@hotmail.com

address: nallakunta

inserted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
234
235
405
456
Delete
Message
OK

customer_id: 405

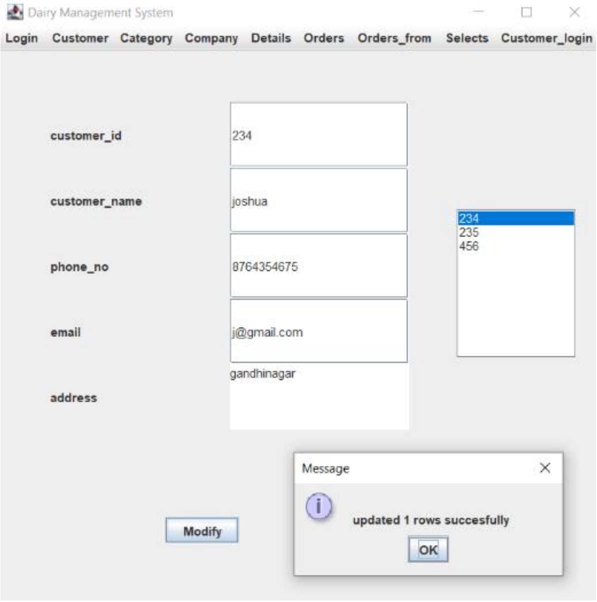
customer_name: vamsi

phone_no: 9876543256

email: vam@hotmail.com

address: nallakunta

Deleted 1 rows succesfully



customer_id 234

customer_name joshua

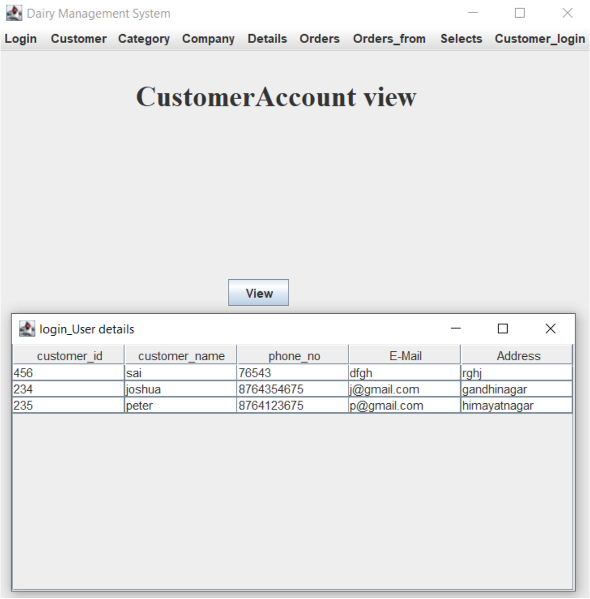
phone_no 8764354675

email j@gmail.com

address gandhinagar

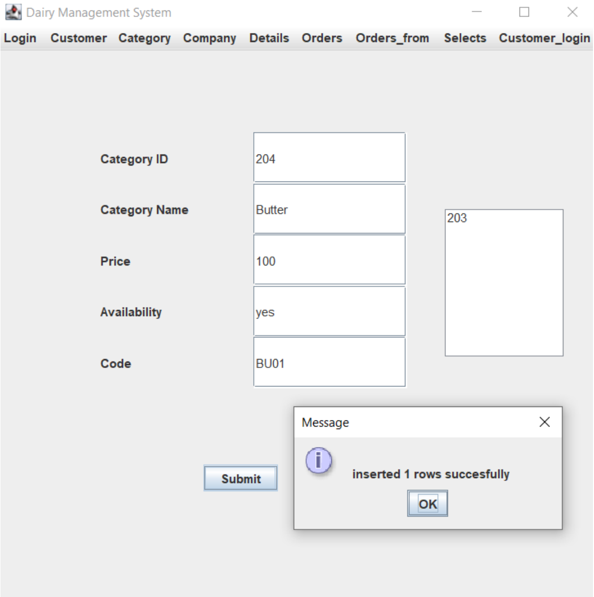
234
235
456

updated 1 rows succesfully



CustomerAccount view

customer_id	customer_name	phone_no	E-Mail	Address
456	sai	76543	dfgh	rghj
234	joshua	8764354675	j@gmail.com	gandhinagar
235	peter	8764123675	p@gmail.com	himayatnagar



Category ID 204

Category Name Butter

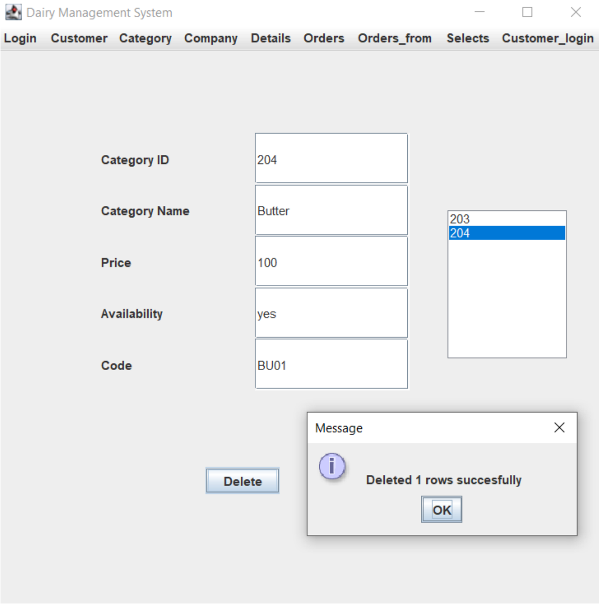
Price 100

Availability yes

Code BU01

203

inserted 1 rows succesfully



Category ID 204

Category Name Butter

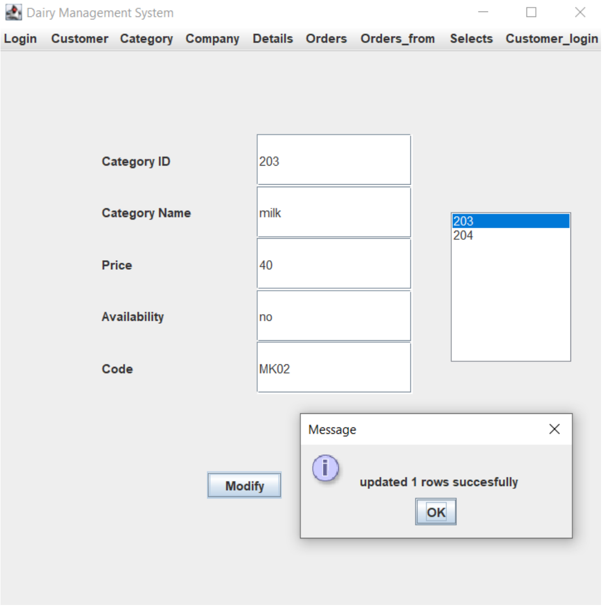
Price 100

Availability yes

Code BU01

203
204

Deleted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
OK

Category ID 203

Category Name milk

Price 40

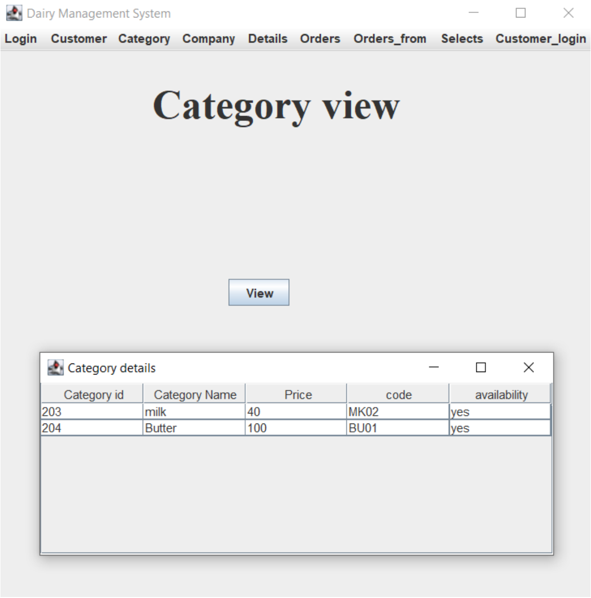
Availability no

Code MK02

203
204

Message

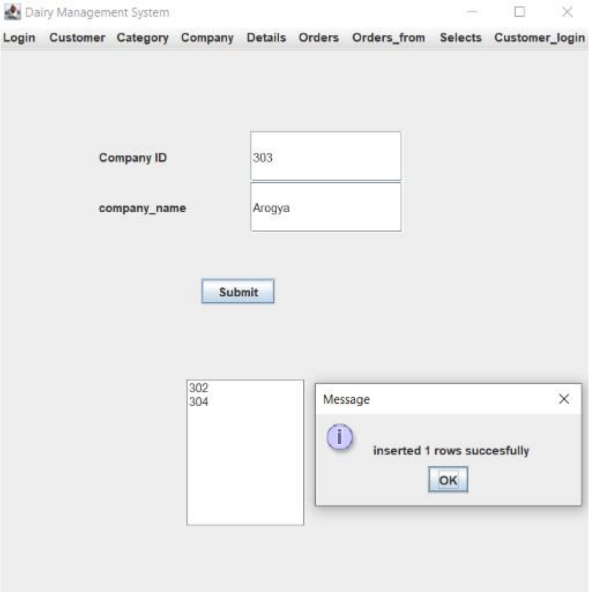
updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View

Category view

Category details

Category id	Category Name	Price	code	availability
203	milk	40	MK02	yes
204	Butter	100	BU01	yes


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Submit
OK

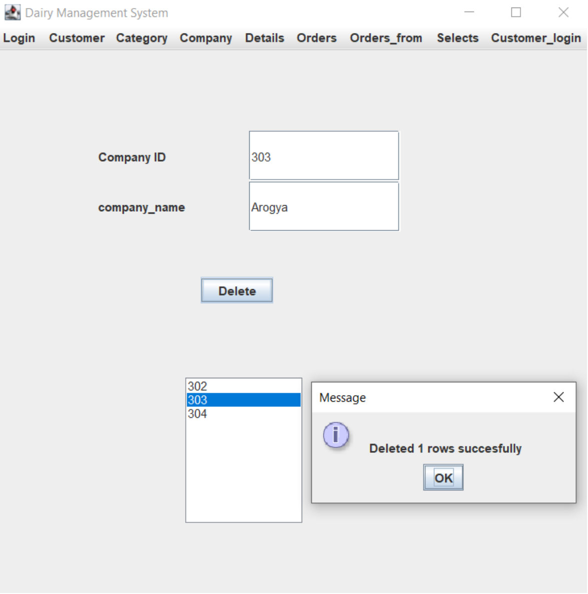
Company ID 303

company_name Arogya

302
304

Message

inserted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Delete
OK

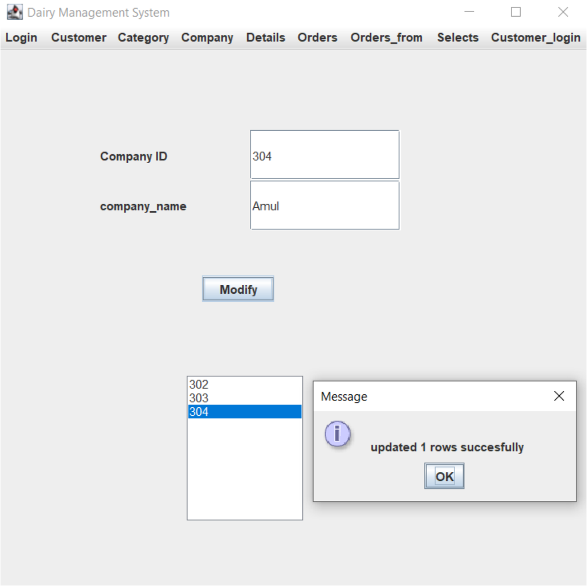
Company ID 303

company_name Arogya

302
303
304

Message

Deleted 1 rows succesfully



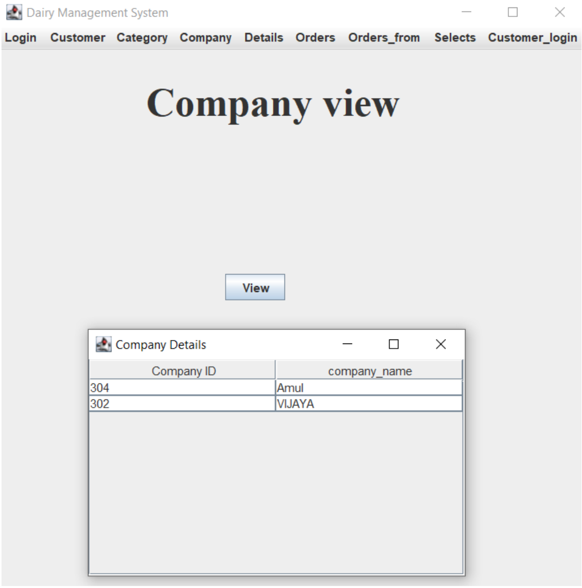
Company ID 304

company_name Amul

302
303
304

Message

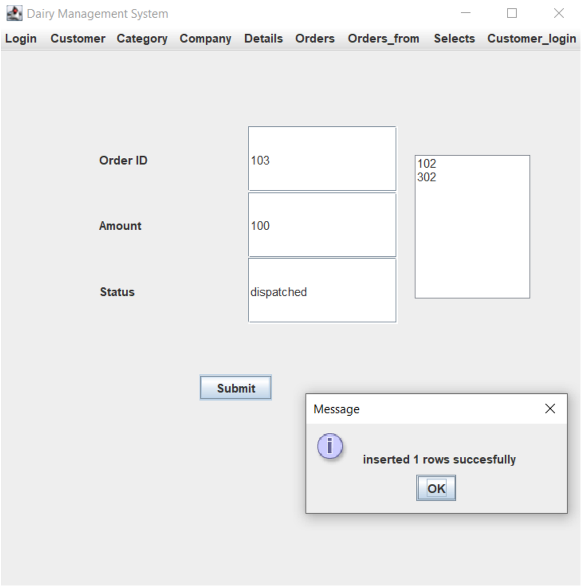
updated 1 rows succesfully



Company view

Company Details

Company ID	company_name
304	Amul
302	VIJAYA



Order ID 103

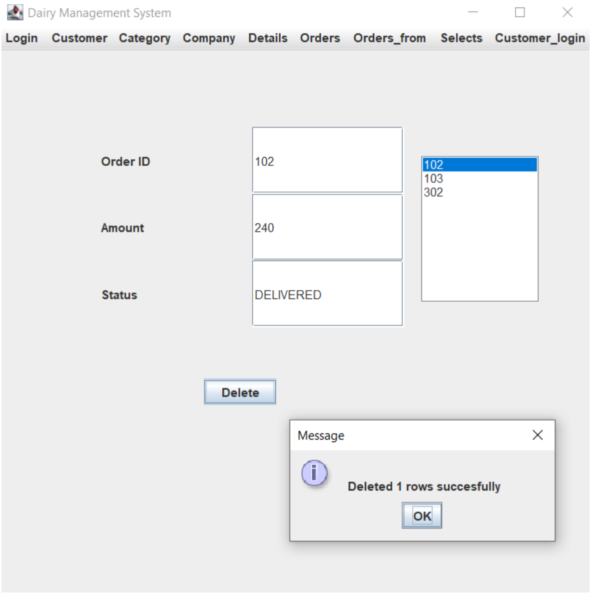
Amount 100

Status dispatched

102
302

Message

inserted 1 rows succesfully



Order ID 102

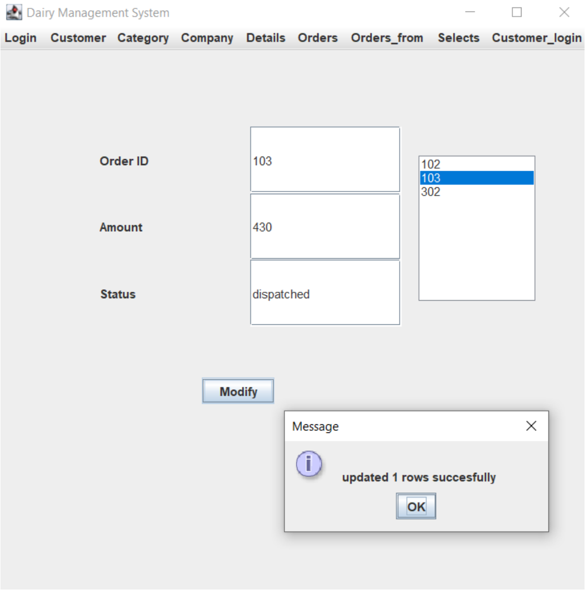
Amount 240

Status DELIVERED

102
103
302

Message

Deleted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
OK

Order ID 103

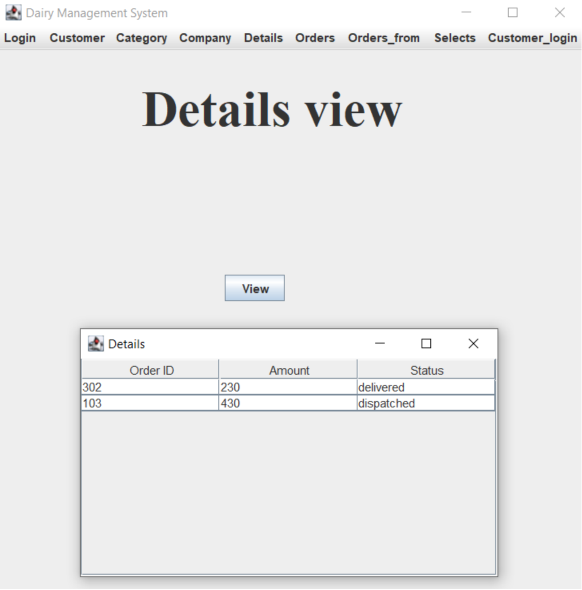
Amount 430

Status dispatched

102
103
302

Message

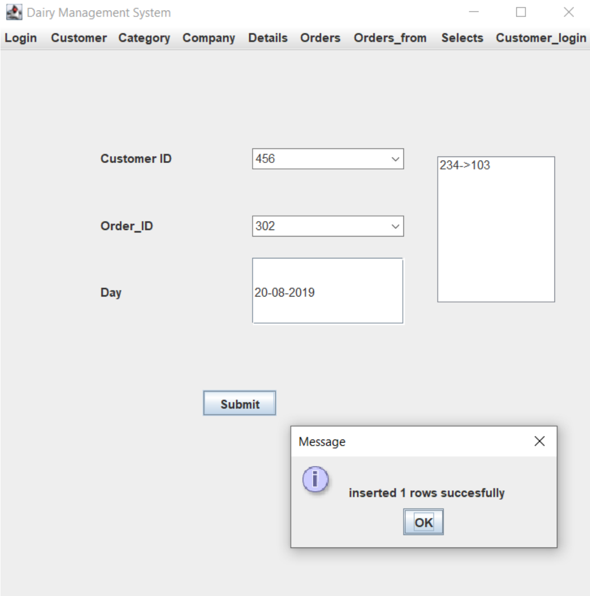
updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View

Details view

Details

Order ID	Amount	Status
302	230	delivered
103	430	dispatched


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Submit
OK

Customer ID 456

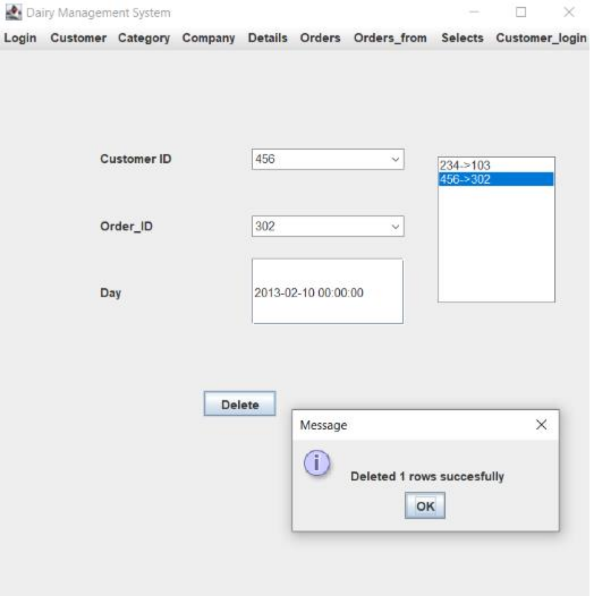
Order_ID 302

Day 20-08-2019

234->103

Message

inserted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Delete
OK

Customer ID 456

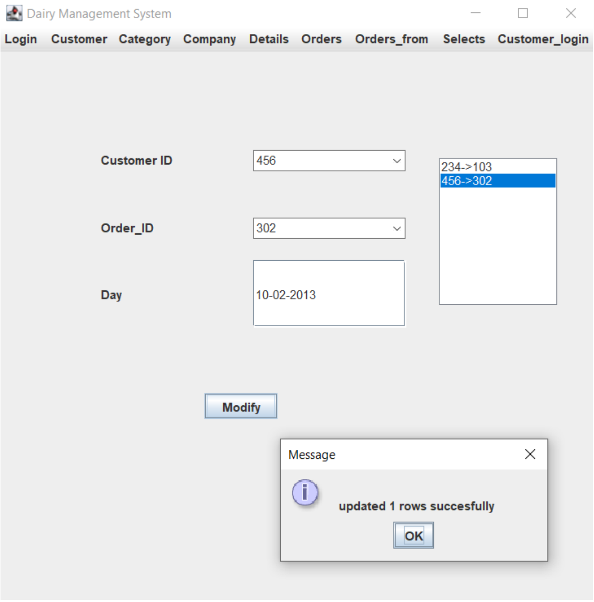
Order_ID 302

Day 2013-02-10 00:00:00

234->103
456->302

Message

Deleted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
OK

Customer ID: 456

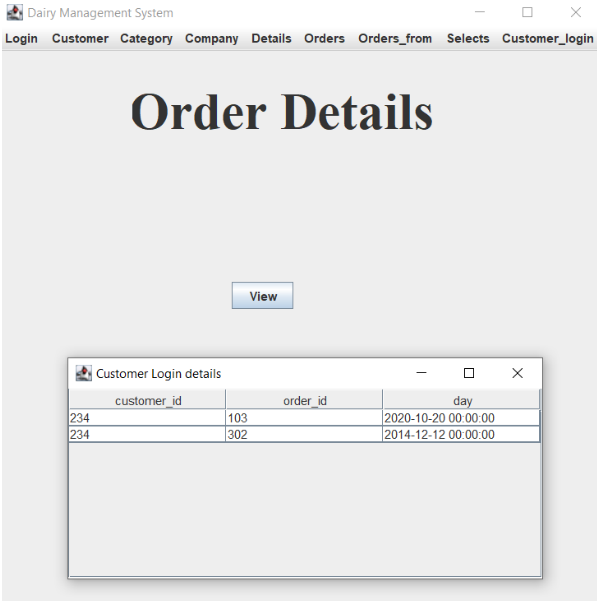
Order_ID: 302

Day: 10-02-2013

234->103
456->302

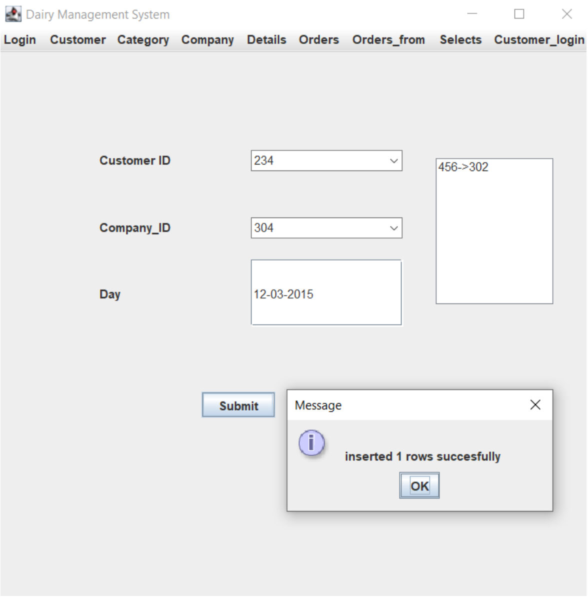
Message

updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View
Customer Login details

Order Details

customer_id	order_id	day
234	103	2020-10-20 00:00:00
234	302	2014-12-12 00:00:00


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Submit
OK

Customer ID: 234

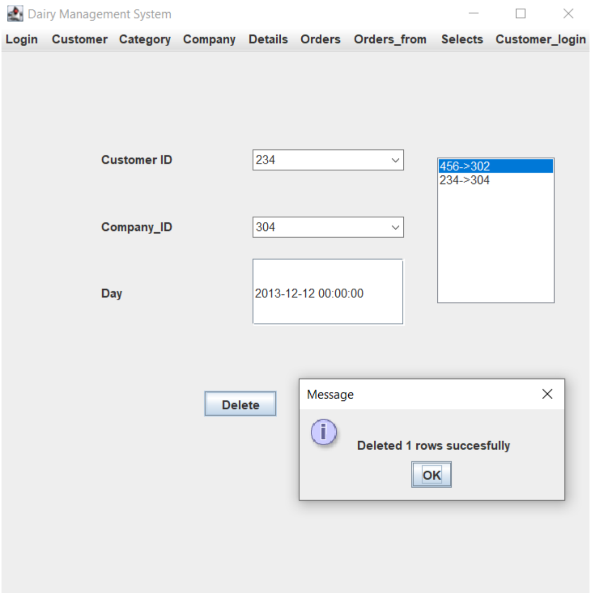
Company_ID: 304

Day: 12-03-2015

456->302

Message

inserted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Delete
OK

Customer ID: 234

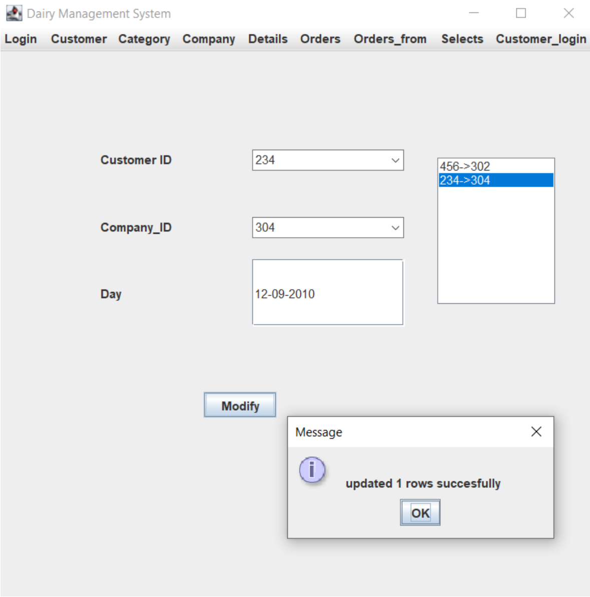
Company_ID: 304

Day: 2013-12-12 00:00:00

456->302
234->304

Message

Deleted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
OK

Customer ID: 234

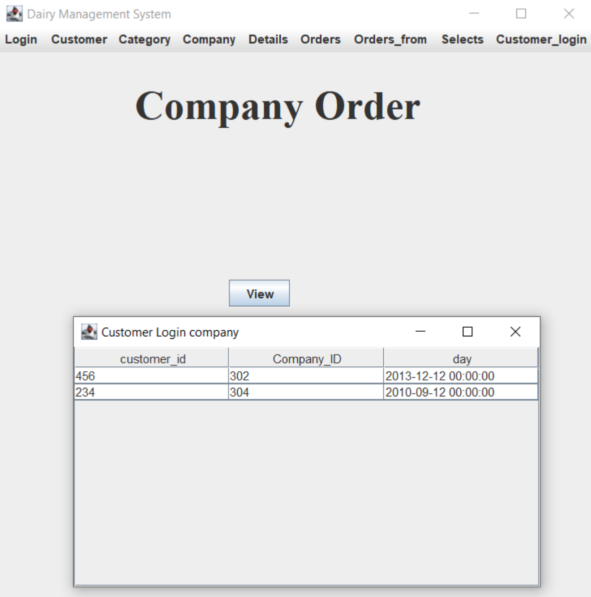
Company_ID: 304

Day: 12-09-2010

456->302
234->304

Message

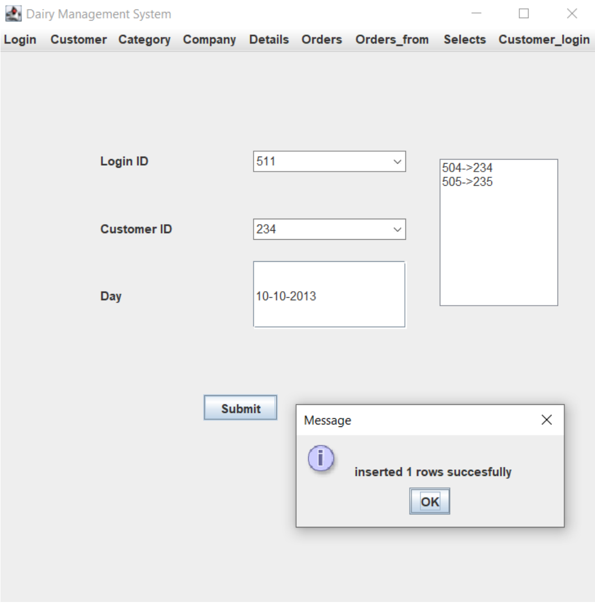
updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View

Company Order

Customer Login company

customer_id	Company_ID	day
456	302	2013-12-12 00:00:00
234	304	2010-09-12 00:00:00


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Submit
OK

Login ID: 511

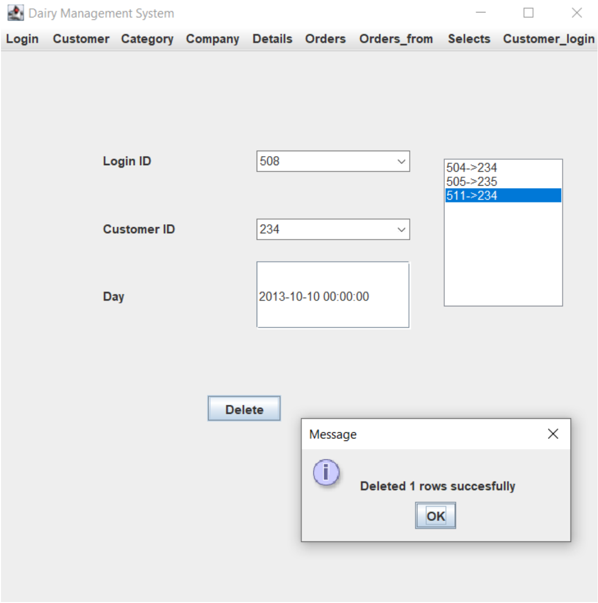
Customer ID: 234

Day: 10-10-2013

504->234
505->235

Message

inserted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Delete
OK

Login ID: 508

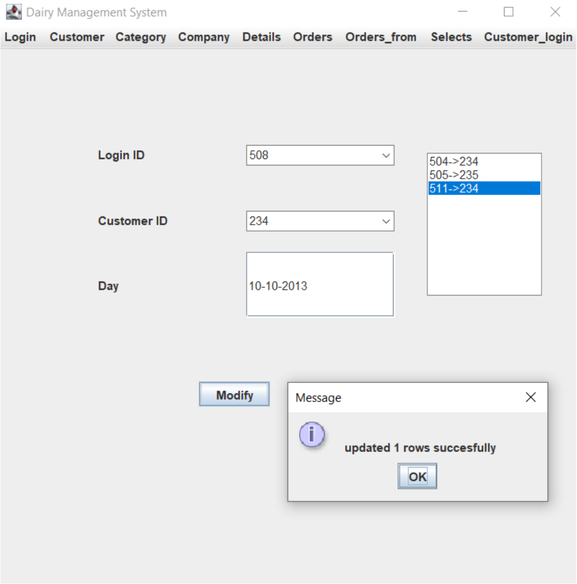
Customer ID: 234

Day: 2013-10-10 00:00:00

504->234
505->235
511->234

Message

Deleted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
OK

Login ID: 508

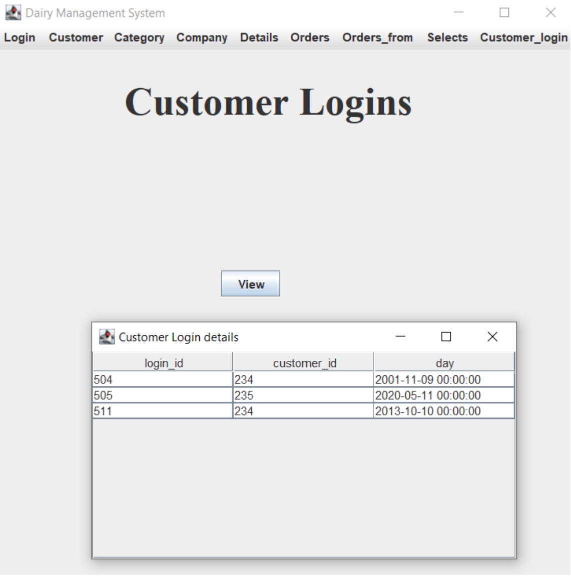
Customer ID: 234

Day: 10-10-2013

504->234
505->235
511->234

Message

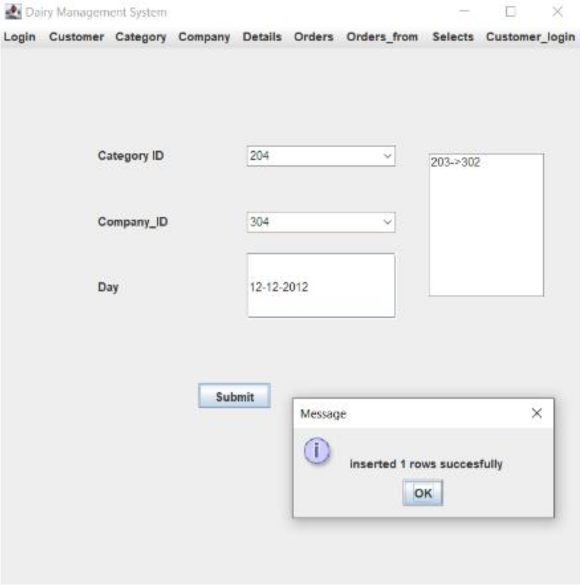
updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View

Customer Logins

Customer Login details

login_id	customer_id	day
504	234	2001-11-09 00:00:00
505	235	2020-05-11 00:00:00
511	234	2013-10-10 00:00:00


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Submit
OK

Category ID: 204

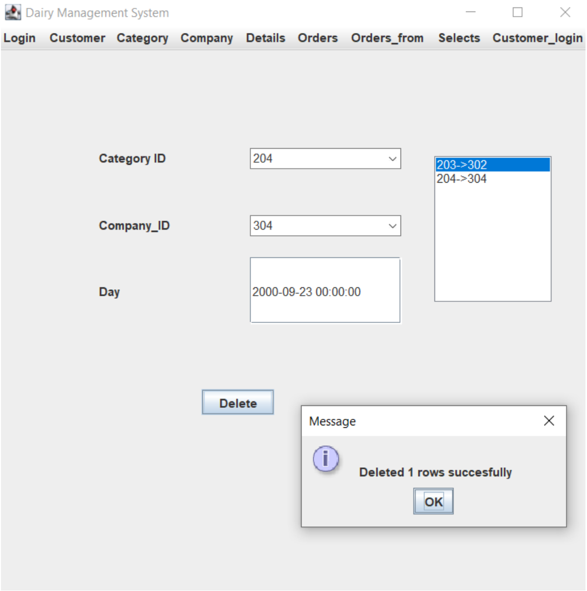
Company_ID: 304

Day: 12-12-2012

203->302

Message

inserted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Delete
OK

Category ID: 204

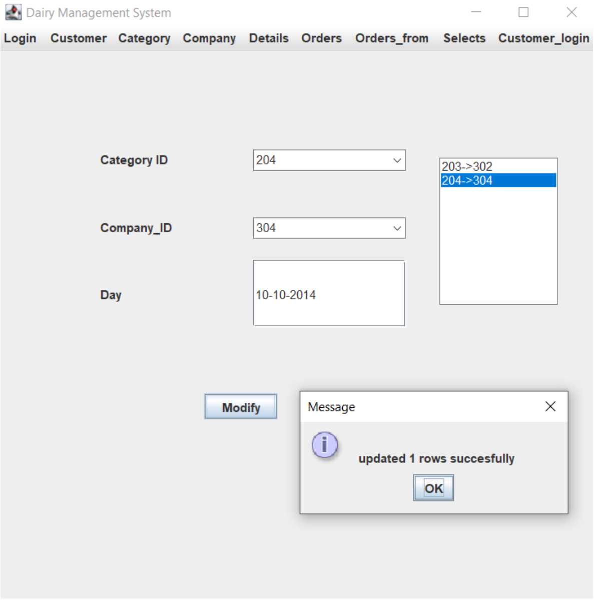
Company_ID: 304

Day: 2000-09-23 00:00:00

203->302
204->304

Message

Deleted 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
Modify
OK

Category ID: 204

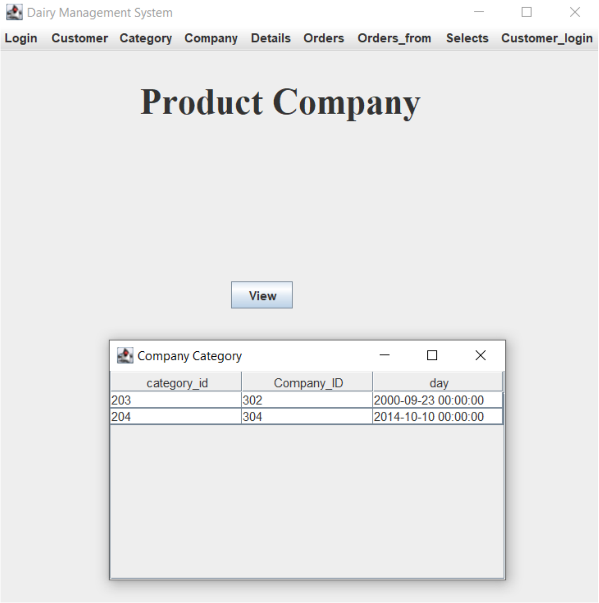
Company_ID: 304

Day: 10-10-2014

203->302
204->304

Message

updated 1 rows succesfully


Dairy Management System
Login Customer Category Company Details Orders Orders_from Selects Customer_login
View

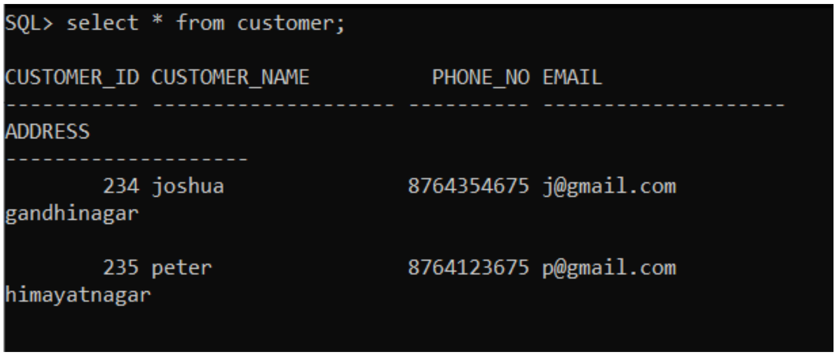
Product Company

Company Category

category_id	Company_ID	day
203	302	2000-09-23 00:00:00
204	304	2014-10-10 00:00:00

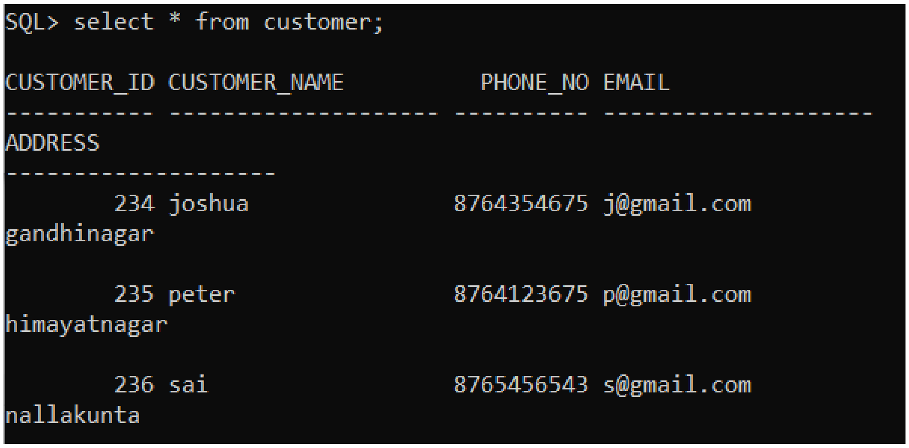
DATABASE OUTPUTS:

BEFORE



```
SQL> select * from customer;

CUSTOMER_ID CUSTOMER_NAME           PHONE_NO EMAIL
----------- -------------------- ---------- --------------------
ADDRESS
--------------------
        234 joshua                8764354675 j@gmail.com
gandhinagar

        235 peter                 8764123675 p@gmail.com
himayatnagar
```

AFTER



```
SQL> select * from customer;

CUSTOMER_ID CUSTOMER_NAME           PHONE_NO EMAIL
----------- -------------------- ---------- --------------------
ADDRESS
--------------------
        234 joshua                8764354675 j@gmail.com
gandhinagar

        235 peter                 8764123675 p@gmail.com
himayatnagar

        236 sai                   8765456543 s@gmail.com
nallakunta
```

DISCUSSION AND FUTURE WORK

So far, this project has helped us to sell products to the customers if they are available. It helped us to view the details of the products, customers and the company.

In future we try to include an option which allows the customer to pre book the products and see that if they are not available. We ensure to contact the company and get the products as soon as possible and deliver them to the customers.

REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill International Edition, 2010.
2. https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001

THANK YOU