**Relational Database Management System Project Report**

**On**

**“Online Shopping Management System”**

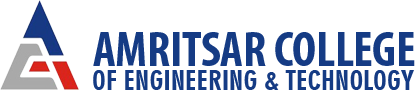
Submitted in the Partial fulfilment of the requirement for the Award of Degree of

**Bachelor of Technology**

**in**

**COMPUTER SCIENCE & ENGINEERING TECHNOLOGY**

**Batch**

**(2017-2021)**

|  |  |
| --- | --- |
| Submitted To:- **Er. Amarpreet Singh** | Submitted by:- **Aayush Khanna (1701214)**  **Anuj Sharma (1701234)**  **Arshpreet Kaur (1701237)**  **Badal Kashyap (1701242)** |

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**AMRITSAR GROUP OF COLLEGES, Amritsar**

**(Autonomous college under UGC Act – 1956[2(f) and 12(b)]**

**ACKNOWLEDGEMENT**

This is a humble effort to express our sincere gratitude towards those who have guided and helped us to complete this project

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. We are highly grateful and obliged to each and everyone making us help out of problems being faced by us.

It would not have been possible to see through the undertaken project without the guidance of Er. Amarpreet Singh. It was purely on the basis of their experience and knowledge that we able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department Er. Vinod sharma and all Members of Computer Science Deptt. who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them.

**DECLARATION**

We hereby declare that the project work entitled **“Online Shopping Management System”** is an authentic record of our work carried out as requirements of Institutional Training project for the award of degree of B.Tech(CSE), **Amritsar Group Of Colleges, Amritsar,** under the guidance of Er.Amarpreet singh

(Signature of student)

**Aayush Khanna (1701214)**

**Anuj Sharma (1701234)**

**Arshpreet Kaur (1701237)**

**Badal Kashyap (1701242)**

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

**Faculty Coordinator**

Er. Amarpreet Singh (Associate Professor – CSE Department)

INDEX PAGE

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Content** | Page No. |
| 1 | Introduction to the Subject | 5 |
| 2 | Introduction to PLSQL | 7 |
| 3 | Introduction to the Project | 9 |
| 4 | Step by Step implementation of the project | 10 |
| 5 | Snapshot at every step |  |
|  |  |  |

**INTRODUCTION TO SUBJECT**

**RDBMS:**

RDBMS stands for Relational Database Management System. RDBMS data is structured in database tables, fields and records. Each RDBMS table consists of database table rows. Each database table row consists of one or more database table fields. RDBMS store the data into collection of tables, which might be related by common fields (database table columns). RDBMS also provide relational operators to manipulate the data stored into the database tables. Most RDBMS use SQL as database querylanguage. The most popular RDBMS are MS SQL Server, DB2, Oracle and MySQL. The relational model is an example of record-based model. Record based models are so named because the database is structured in fixed format records of several types. Each table contains records of a particular type. Each record type defines a fixed number of fields, or attributes. The columns of the table correspond to the attributes of the record types.

The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.

**DIFFERENCE BETWEEN DBMS & RDBMS**

A DBMS has to be persistent, that is it should be accessible when the program created the data ceases to exist or even the application that created the data restarted. A DBMS also has to provide some uniform methods independent of a specific application for accessing the information that is stored. RDBMS is a Relational Data Base Management System Relational DBMS. This adds the additional condition that the system supports a tabular structure for the data, with enforced relationships between the tables. This excludes the databases that don't support a tabular structure or don't enforce relationships between tables. You can say DBMS does not impose any constraints or security with regard to data manipulation it is user or the programmer responsibility to ensure the ACID PROPERTY of the database whereas the RDBMS is more with this regard because RDBMS define the integrity constraint for the purpose of holding ACID PROPERTY.ACID property stands for

A:-Atomicity

C:-Consistency

I:-Isolation

D:-Durability

**SQL**

Structured Query Language (SQL) which uses a combination of Relational algebra and Relational calculus. It is a data sub language used to organize, manage and retrieve data from relational database, which is managed by Relational Database Management System (RDBMS). Vendors of DBMS like Oracle, IBM, DB2, Sybase, and Ingress use SQL as programming language for their database. Original version of SQL was SEQUEL which was an Application Program Interface (API) to the system R project. Oracle / 6 The predecessor of SEQUEL was named SQUARE. SQL-92 is the current standard and is the current version. The SQL language can be used in two ways : ↵ Interactively or ↵ Embedded inside another program. The SQL is used interactively to directly operate a database and produce the desired results. The user enters SQL command that is immediately executed. The second way to execute a SQL command is by embedding it in another language such as Cobol, Pascal, BASIC, C, Visual Basic,etc.

**SUBDIVISIONS OF SQL**

Regardless of whether SQL is embedded or used interactively, it can be divided into three groups of commands, depending on their purpose.

• Data Definition Language (DDL).

• Data Manipulation Language (DML).

• Data Control Language (DCL).

Data Definition Language : Data Definition Language is a part of SQL that is responsible for the creation, updation and deletion . The list of DDL commands is given below

Data Manipulation Language : Data manipulation commands manipulate (insert, delete, update and retrieve) data. These commands include:

SELECT , UPDATE ,DELETE , INSERT

Data Control Language : The commands that form data control language are related to the security of the database performing tasks of assigning privileges so users can access certain objects in the database. The DCL commands are :

GRANT , REVOKE , COMMIT , ROLLBACK

**INTRODUCTION TO PL/SQL**

PL/SQL is a block structured language that enables developers to combine the power of SQL with procedural statements. All the statements of the block are passed to oracle engine all at once which increases processing speed and decreases the traffic.PL/SQL is an extension of Structured Query Language (SQL) that is used in Oracle. Unlike SQL, PL/SQL allows the programmer to write code in a procedural format.

Structure of PLSQL block:

* This is the component which has the actual PL/SQL code.
* This consists of different sections to divide the code logically (declarative section for declaring purpose, execution section for processing statements, exception handling section for handling errors)
* It also contains the SQL instruction that used to interact with the database server.
* All the PL/SQL units are treated as PL/SQL blocks, and this is the starting stage of the architecture which serves as the primary input.
* Following are the different type of PL/SQL units.
  + Anonymous Block
  + Function
  + Library
  + Procedure
  + Package Body
  + Package Specification
  + Trigger
  + Type
  + Type Body

Features of PL/SQL:

1. PL/SQL is basically a procedural language, which provides the functionality of decision making, iteration and many more features of procedural programming languages.
2. PL/SQL can execute a number of queries in one block using single command.
3. One can create a PL/SQL unit such as procedures, functions, packages, triggers, and types, which are stored in the database for reuse by applications.
4. PL/SQL provides a feature to handle the exception which occurs in PL/SQL block known as exception handling block.
5. Applications written in PL/SQL are portable to computer hardware or operating system where Oracle is operational.
6. PL/SQL Offers extensive error checking.

Comparison between SQL and PL/SQL:

|  |  |
| --- | --- |
| SQL | PL/SQL |
| SQL is a single query that is used to perform DML and DDL operations. | PL/SQL is a block of codes that used to write the entire program blocks/ procedure/ function, etc. |
| It is declarative, that defines what needs to be done, rather than how things need to be done. | PL/SQL is procedural that defines how the things needs to be done. |
| Execute as a single statement. | Execute as a whole block. |
| Mainly used to manipulate data. | Mainly used to create an application. |
| Cannot contain PL/SQL code in it. | It is an extension of SQL, so it can contain SQL inside it. |

Advantage of Using PL/SQL:

1. Better performance, as SQL is executed in bulk rather than a single statement
2. High Productivity
3. Tight integration with SQL
4. Full Portability
5. Tight Security
6. Support Object Oriented Programming concepts.

Disadvantages of Oracle PLSQL:

If you're just running Oracle databases, there are no major disadvantages to using PL/SQL as long as you're aware of its limitations.

**INTRODUCTION TO PROJECT**

**Online Shopping Management System**

1-Firstly, We would like to share what we have used in this project

In this project we have used procedures for different tables.

2-Then initially we were given four tables:

* + Customer table
  + Products table
  + Purchase items table
  + Manage account table

3- In first table there are customer details i.e. customer id and customer name.

In second table there are product details i.e. product id , product name and product price.

In third table there are items details i.e. purchase id , product id and number of items

purchased.

In third table there are manage account details account name , account password and

account address.

4- We have used insertion, deletion and updation operations in all the three tables.

**CODING**

create or replace procedure insertcustomer(

pcusid customer.cusid%type,

pcusname in customer.cusname%type)

IS

begin

insert into customer (cusid,cusname)

values(pcusid,pcusname);

--commit;--

end;

declare

pn customer.cusid%type;

pg customer.cusname%type;

begin

pn:='&pcusid';

pg:='&pcusname';

insertcustomer(pn,pg);

end;

create or replace procedure insertproducts(

pproid products.proid%type,

pproname in products.proname%type,

pproprice in products.proprice%type)

IS

begin

insert into products (proid,proname,proprice)

values(pproid,pproname,pproprice);

--commit;--

end;

declare

pn products.proid%type;

pg products.proname%type;

pf products.proprice%type;

begin

pn:='&pproid';

pg:='&pproname';

pf:='&pproprice';

insertproducts(pn,pg,pf);

end;

create or replace procedure insertpurchase\_items(

ppurid purchase\_items.purid%type,

pproid in purchase\_items.proid%type,

pno\_of\_items in purchase\_items.no\_of\_items%type)

IS

begin

insert into purchase\_items (purid,proid,no\_of\_items)

values(ppurid,pproid,pno\_of\_items);

--commit;--

end;

declare

pn purchase\_items.purid%type;

pg purchase\_items.proid%type;

pf purchase\_items.no\_of\_items%type;

begin

pn:='&ppurid';

pg:='&pproid';

pf:='&pno\_of\_items';

insertpurchase\_items(pn,pg,pf);

end;

create or replace procedure insertmanageaccount(

paccname manageaccount.accname%type,

paccpass in manageaccount.accpass%type,

paccaddress in manageaccount.accaddress%type)

IS

begin

insert into manageaccount (accname,accpass,accaddress)

values(paccname,paccpass,paccaddress);

--commit;--

end;

declare

pn manageaccount.accname%type;

pg manageaccount.accpass%type;

pf manageaccount.accaddress%type;

begin

pn:='&paccname';

pg:='&paccpass';

pf:='&paccaddress';

insertmanageaccount(pn,pg,pf);

end;

create or replace procedure deletecustomer(

pcusid customer.cusid%type)

IS

begin

delete from customer where cusid=pcusid;

end;

declare

pn customer.cusid%type;

begin

pn:='&pcusid';

deletecustomer(pn);

end;

create or replace procedure deleteproducts(

pproid products.proid%type)

IS

begin

delete from products where proid=pproid;

end;

declare

pn products.proid%type;

begin

pn:='&pproid';

deleteproducts(pn);

end;

create or replace procedure deletepurchase\_items(

ppurid purchase\_items.purid%type

--pproid in purchase\_items.proid%type,--

)

IS

begin

delete from purchase\_items where purid=ppurid;

end;

declare

pn purchase\_items.purid%type;

begin

pn:='&ppurid';

deletepurchase\_items(pn);

end;

create or replace procedure deletemanageaccount(

paccname manageaccount.accname%type)

IS

begin

delete from manageaccount where accname=paccname;

--commit;--

end;

declare

pn manageaccount.accname%type;

begin

pn:='&paccname';

deletemanageaccount(pn);

end;

create or replace procedure updatecustomer(

pcusid customer.cusid%type,

pcusname in customer.cusname%type)

IS

begin

update customer set cusname=pcusname where cusid=pcusid;

--commit;--

end;

declare

pn customer.cusid%type;

pg customer.cusname%type;

begin

pn:='&pcusid';

pg:='&pcusname';

updatecustomer(pn,pg);

end;

create or replace procedure updateproducts(

pproid products.proid%type,

pproname in products.proname%type,

pproprice in products.proprice%type)

IS

begin

update products set proname=pproname,proprice=pproprice where proid=pproid;

--commit;--

end;

declare

pn products.proid%type;

pg products.proname%type;

pf products.proprice%type;

begin

pn:='&pproid';

pg:='&pproname';

pf:='&pproprice';

updateproducts(pn,pg,pf);

end;

create or replace procedure insertpurchase\_items(

ppurid purchase\_items.purid%type,

pproid in purchase\_items.proid%type,

pno\_of\_items in purchase\_items.no\_of\_items%type)

IS

begin

insert into purchase\_items (purid,proid,no\_of\_items)

values(ppurid,pproid,pno\_of\_items);

--commit;--

end;

declare

pn purchase\_items.purid%type;

pg purchase\_items.proid%type;

pf purchase\_items.no\_of\_items%type;

begin

pn:='&ppurid';

pg:='&pproid';

pf:='&pno\_of\_items';

updatepurchase\_items(pn,pg,pf);

end;

create or replace procedure updatemanageaccount(

paccname manageaccount.accname%type,

paccpass in manageaccount.accpass%type,

paccaddress in manageaccount.accaddress%type)

IS

begin

update manageaccount set accpass=paccpass,accaddress=paccaddress where accname=paccname;

--commit;--

end;

declare

pn purchase\_items.purid%type;

pg purchase\_items.proid%type;

pf purchase\_items.no\_of\_items%type;

begin

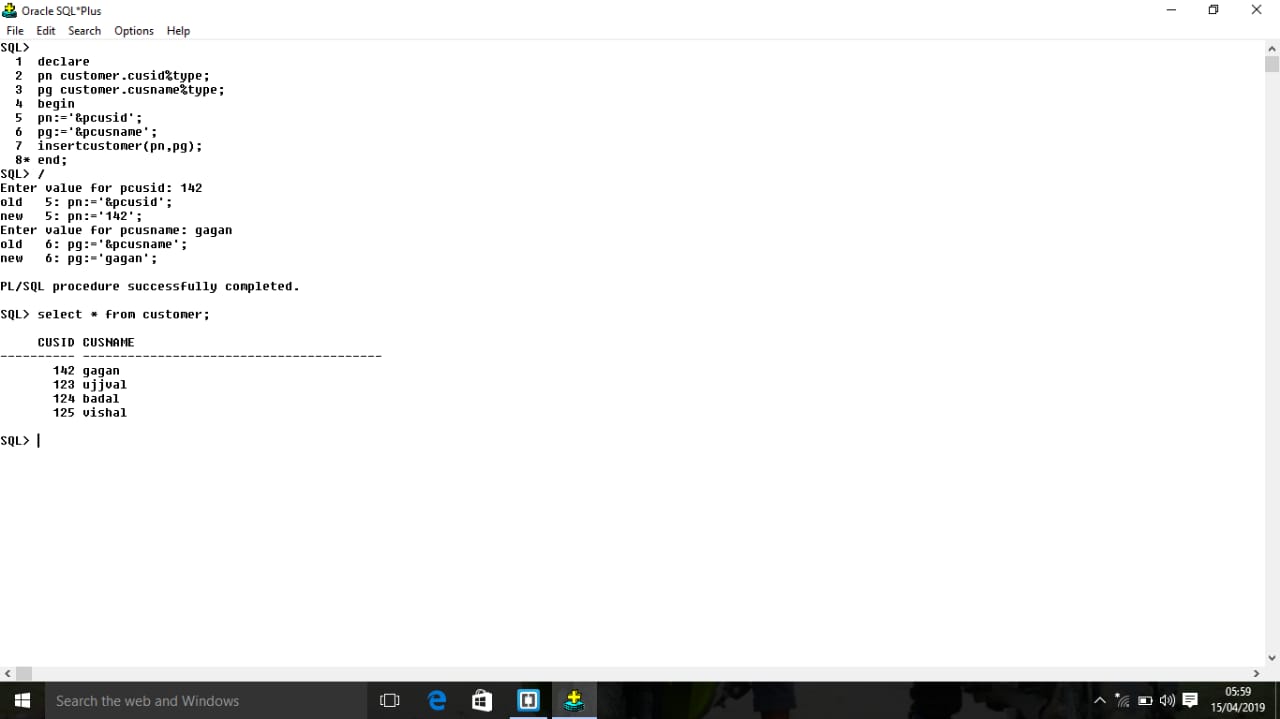
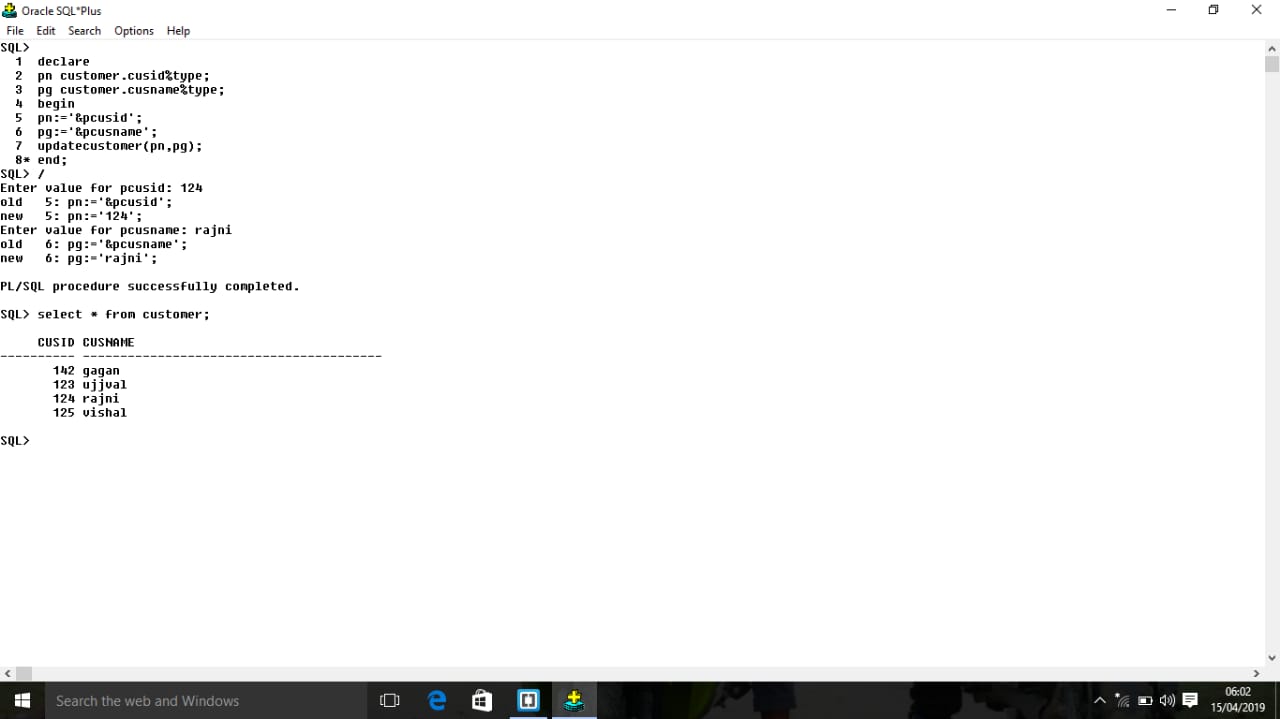
pn:='&ppurid';

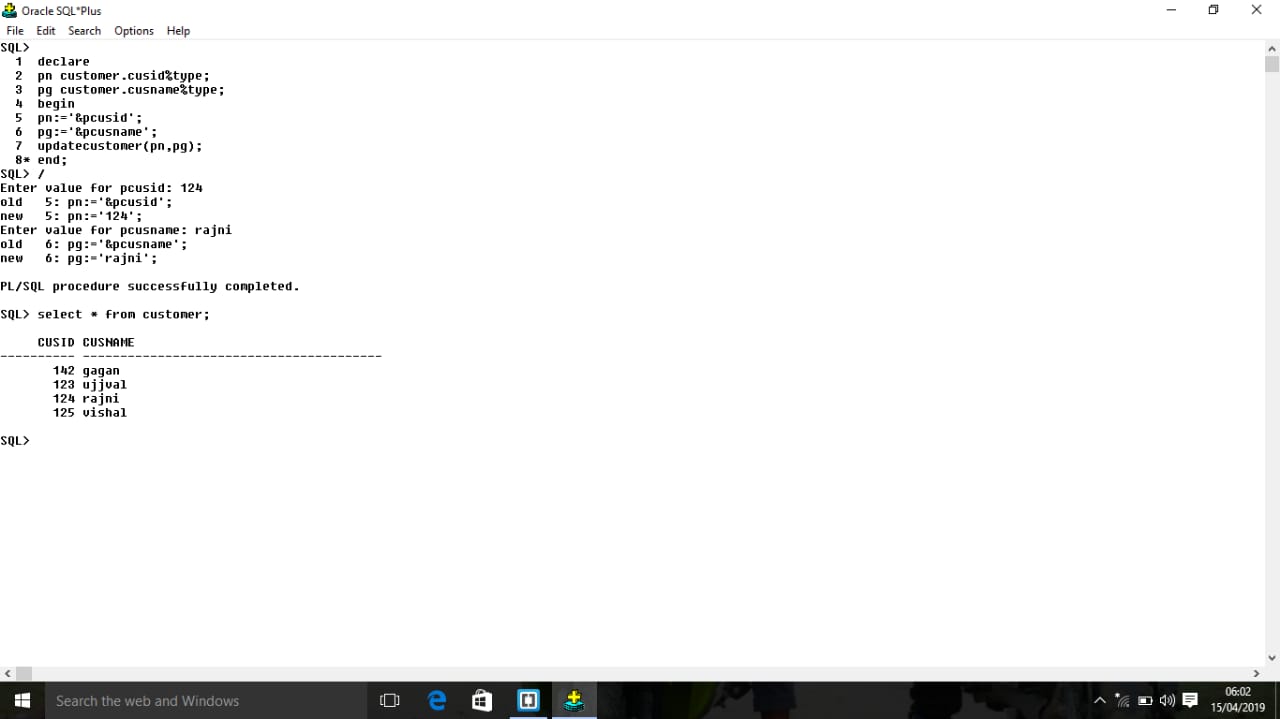
pg:='&pproid';

pf:='&pno\_of\_items';

updatepurchase\_items(pn,pg,pf);

end;

**SNAPSHOTS**

****