Ahsanullah University of Science Technology

PROJECT REPORT

“Computer Store Management System”

Submitted by

Nafis Islam
ID: 14.02.04.014
Section: A1

Hafizul Islam Himel
ID: 14.02.04.024
Section: A1

Alauddin Al Azad
ID: 14.01.04.067
Section: A1

1 Introduction

A Computer store needs a management system to manage it's inventory. So, we are proposing a computer store management system using pl/sql.

2 Overview of the project

We've try to develop a ”Computer Store Management System”. Employees can perform some operations through this system. Employees are the administrator of the system.
Employees can-

- Log in & log out from the system.
- Add products.
- Sell products.
- Request products from other branches.
- Transfer products to other branches.
- Buy second hand products from customers.
- Search products based on different attributes.
- See old receipts.
- Add new employees & customers.
- View the current state of the cart.

3 Sites & Tables

3.1 Sites

In our project there are two sites or two branches of the computer store.

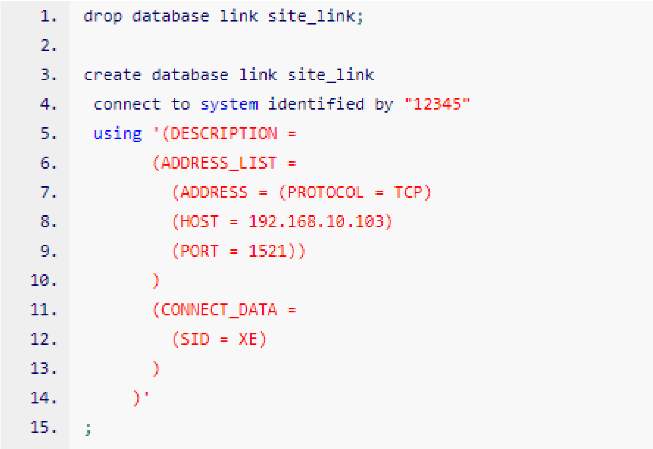
- Kalabagan (Main) Branch.
- Dhanmondi Branch.

3.2 Tables

- **Product**: product(pid, name, catagory, branch).
- **Supplier**: supplier(sid, name, numbers).
- **Customer**: customer(cid, name, numbers).
- **Employee**: employee(eid, name, numbers, passwords).
- **Log in History**: log_in_history(eid, log_time).
- **Product Information**: product_info(prduct_code, pid, sid, cid, buyprice, sellprice, branch, condition, status).
- **Purchase**: purchase(purchase_id, pid, product_code).
- **Purchase History**: purchase_history(purchase_id, eid, cid, buyprice, soldprice, capitalprice, purchase_type).
- **Cart**: temp(pid, product_code, buyprice, sellprice).
- **Request**: request(rid,product_code,eid_req,send_to_branch,at_branch,status,eid_done).

4 Database Links

In order to create the database links we've used the following code:

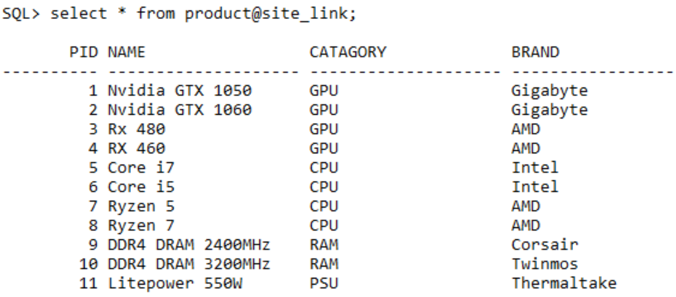


```
drop database link site_link;

create database link site_link
 connect to system identified by "12345"
 using '(DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)
         (HOST = 192.168.10.103)
         (PORT = 1521))
       )
       (CONNECT_DATA =
         (SID = XE)
       )
     )'
;
```

Figure 1: Database Links

Some examples are shown below:

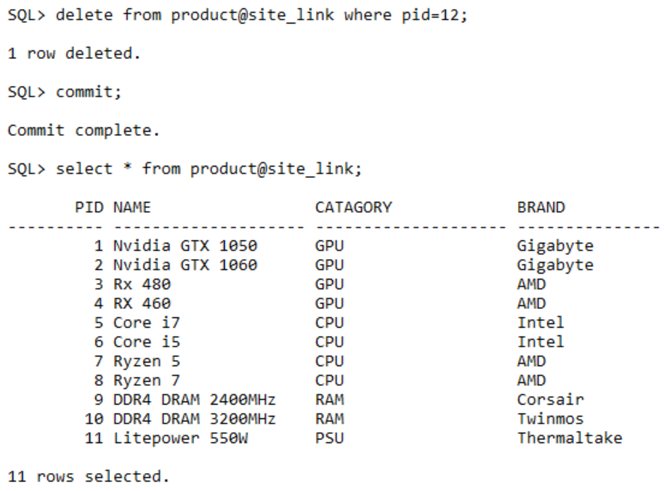
1. Select all data of product table from site_link:



```
SQL> select * from product@site_link;

       PID NAME                 CATAGORY             BRAND
---------- -------------------- -------------------- -----------------
         1 Nvidia GTX 1050      GPU                  Gigabyte
         2 Nvidia GTX 1060      GPU                  Gigabyte
         3 Rx 480               GPU                  AMD
         4 RX 460               GPU                  AMD
         5 Core i7              CPU                  Intel
         6 Core i5              CPU                  Intel
         7 Ryzen 5              CPU                  AMD
         8 Ryzen 7              CPU                  AMD
         9 DDR4 DRAM 2400MHz    RAM                  Corsair
        10 DDR4 DRAM 3200MHz    RAM                  Twinmos
        11 Litepower 550W       PSU                  Thermaltake
```

Figure 2: Select operation

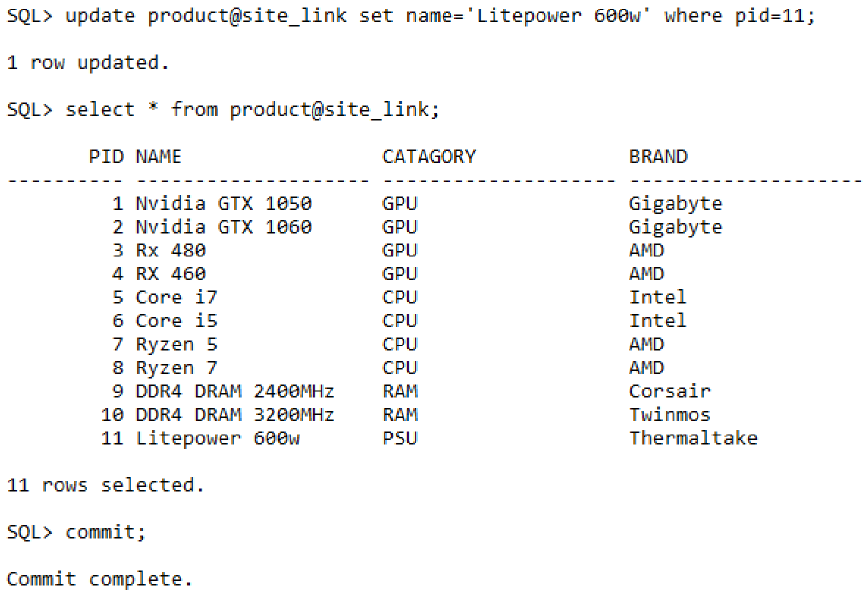
2. Delete pid=12 from site_link:



```
SQL> delete from product@site_link where pid=12;

1 row deleted.

SQL> commit;

Commit complete.

SQL> select * from product@site_link;

       PID NAME                 CATAGORY             BRAND
---------- -------------------- -------------------- ---------------
         1 Nvidia GTX 1050      GPU                  Gigabyte
         2 Nvidia GTX 1060      GPU                  Gigabyte
         3 Rx 480               GPU                  AMD
         4 RX 460               GPU                  AMD
         5 Core i7              CPU                  Intel
         6 Core i5              CPU                  Intel
         7 Ryzen 5              CPU                  AMD
         8 Ryzen 7              CPU                  AMD
         9 DDR4 DRAM 2400MHz    RAM                  Corsair
        10 DDR4 DRAM 3200MHz    RAM                  Twinmos
        11 Litepower 550W       PSU                  Thermaltake

11 rows selected.
```

Figure 3: Delete operation

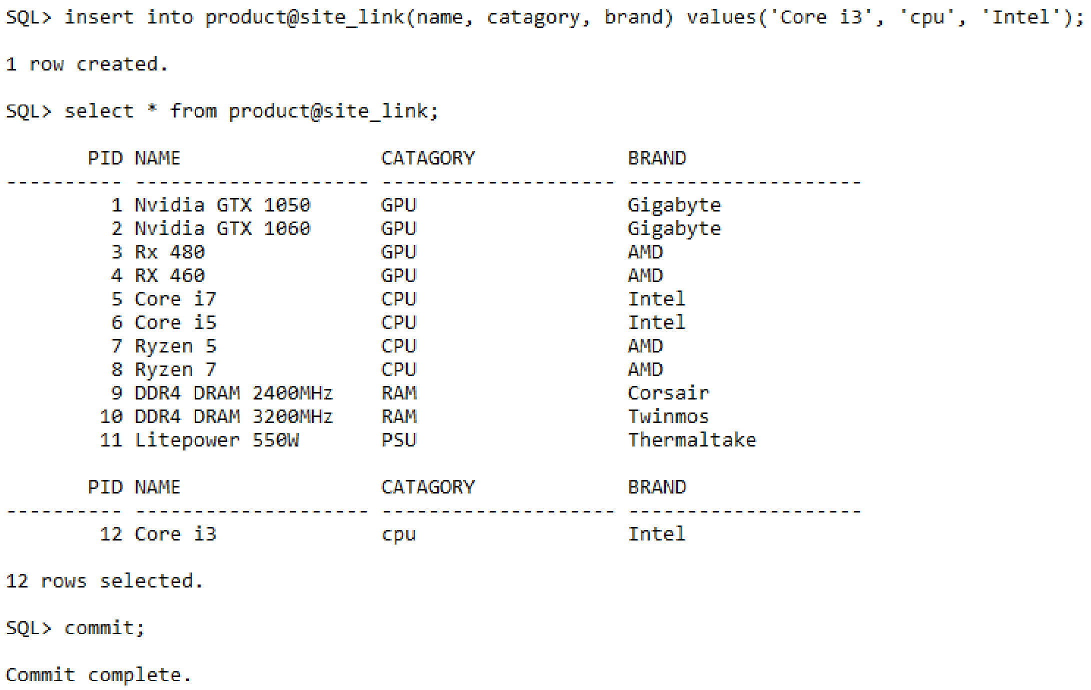
3. Update product name=Litepower where pid=11 from site_link:



```
SQL> update product@site_link set name='Litepower 600w' where pid=11;

1 row updated.

SQL> select * from product@site_link;

       PID NAME                 CATAGORY             BRAND
---------- -------------------- -------------------- --------------------
         1 Nvidia GTX 1050      GPU                  Gigabyte
         2 Nvidia GTX 1060      GPU                  Gigabyte
         3 Rx 480               GPU                  AMD
         4 RX 460               GPU                  AMD
         5 Core i7              CPU                  Intel
         6 Core i5              CPU                  Intel
         7 Ryzen 5              CPU                  AMD
         8 Ryzen 7              CPU                  AMD
         9 DDR4 DRAM 2400MHz    RAM                  Corsair
        10 DDR4 DRAM 3200MHz    RAM                  Twinmos
        11 Litepower 600w       PSU                  Thermaltake

11 rows selected.

SQL> commit;

Commit complete.
```

Figure 4: Update operation

4. Insert product into product table from site_link:

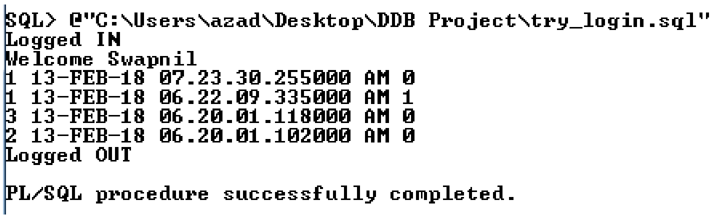


```
SQL> insert into product@site_link(name, catagory, brand) values('Core i3', 'cpu', 'Intel');

1 row created.

SQL> select * from product@site_link;

       PID NAME                 CATAGORY             BRAND
---------- -------------------- -------------------- --------------------
         1 Nvidia GTX 1050      GPU                  Gigabyte
         2 Nvidia GTX 1060      GPU                  Gigabyte
         3 Rx 480               GPU                  AMD
         4 RX 460               GPU                  AMD
         5 Core i7              CPU                  Intel
         6 Core i5              CPU                  Intel
         7 Ryzen 5              CPU                  AMD
         8 Ryzen 7              CPU                  AMD
         9 DDR4 DRAM 2400MHz    RAM                  Corsair
        10 DDR4 DRAM 3200MHz    RAM                  Twinmos
        11 Litepower 550W       PSU                  Thermaltake

       PID NAME                 CATAGORY             BRAND
---------- -------------------- -------------------- --------------------
        12 Core i3              cpu                  Intel

12 rows selected.

SQL> commit;

Commit complete.
```

Figure 5: Insert operation

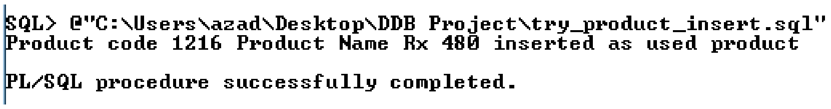
5 Packages, Functions & Procedures

- **Package log_in_package**
  - ∗ Procedure show_login_log
    - · Parameter: No parameter.
    - · Description: Shows the login log table.
  - ∗ Procedure login_log()
    - · Parameter: emp_id.
    - · Description: Inserts the employee into the log with log in time.
  - ∗ Function login()
    - · Parameter: user_name, password.
    - · Description: Logs in the user.
    - · Exceptions: If user id not found throws no data found exception.
    - · Return: Boolean flag.
  - ∗ Function logout()
    - · Parameter: emp_id.
    - · Description: Logs out the user.
    - · Exceptions: If user id not found throws no data found exception.
    - · Return: Boolean flag.



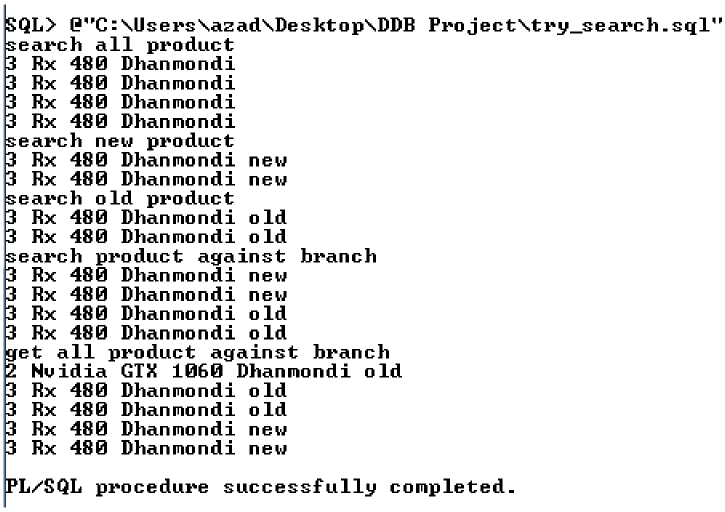
```
SQL> @"C:\Users\azad\Desktop\DDB Project\try_login.sql"
Logged IN
Welcome Swapnil
1 13-FEB-18 07.23.30.255000 AM 0
1 13-FEB-18 06.22.09.335000 AM 1
3 13-FEB-18 06.20.01.118000 AM 0
2 13-FEB-18 06.20.01.102000 AM 0
Logged OUT

PL/SQL procedure successfully completed.
```

- **Package product_package**
    - Function insert_product()
        - Parameter: name,brand,category.
        - Description: Insert product in to product tables.
        - Exceptions: Throws exception when not successfully inserted.
        - Return: 0 error,1 successfully inserted,2 already exist.
    - Function insert_product_new()
        - Parameter: name,brand,category,product_code,supplier_id,buy_price,sell_price,branch.
        - Description: Insert product in to product tables.
        - Exceptions: Throws exception when not successfully inserted.
        - Return: 0 error,1 successfully inserted,2 already exist.
    - Function insert_product_old()
        - Parameter: name,brand,category,product_code,customer_id,buy_price,sell_price,branch.
        - Description: Insert product in to product tables.
        - Exceptions: Throws exception when not successfully inserted.
        - Return: 0 error,1 successfully inserted,2 already exist.



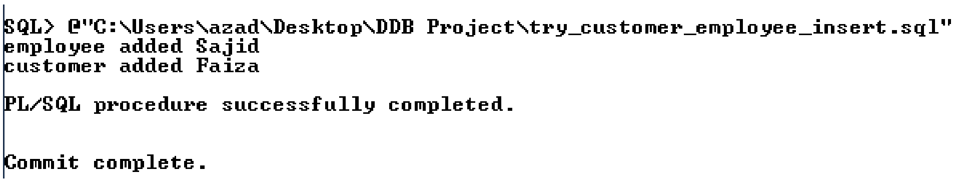
```
SQL> @"C:\Users\azad\Desktop\DDB Project\try_product_insert.sql"
Product code 1216 Product Name Rx 480 inserted as used product

PL/SQL procedure successfully completed.
```

-
    - Procedure search_all_pro()
        - Parameter: product_name.
        - Description: Search products and show.
    - Procedure search_con_pro()
        - Parameter: product_name,product_info.
        - Description: Search products and show.
    - Procedure search_bran_pro()
        - Parameter: product_name,product_brand.
        - Description: Search products and show.
    - Procedure get_all_pro_branch()
        - Parameter: product_info.
        - Description: Get all products.



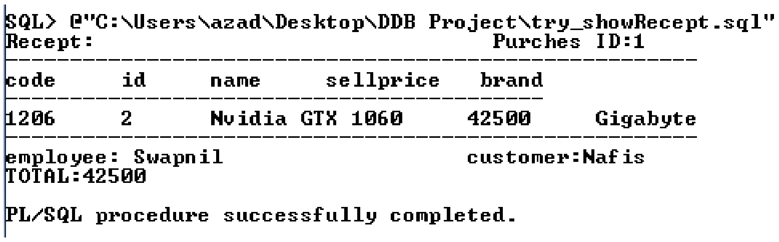
```
SQL> @"C:\Users\azad\Desktop\DDB Project\try_search.sql"
search all product
3 Rx 480 Dhanmondi
3 Rx 480 Dhanmondi
3 Rx 480 Dhanmondi
3 Rx 480 Dhanmondi
search new product
3 Rx 480 Dhanmondi new
3 Rx 480 Dhanmondi new
search old product
3 Rx 480 Dhanmondi old
3 Rx 480 Dhanmondi old
search product against branch
3 Rx 480 Dhanmondi new
3 Rx 480 Dhanmondi new
3 Rx 480 Dhanmondi old
3 Rx 480 Dhanmondi old
get all product against branch
2 Nvidia GTX 1060 Dhanmondi old
3 Rx 480 Dhanmondi old
3 Rx 480 Dhanmondi old
3 Rx 480 Dhanmondi new
3 Rx 480 Dhanmondi new

PL/SQL procedure successfully completed.
```

- **Package customer_employee_package**
    - Procedure add_employee
        - Parameter: name,number,password.
        - Description: Add employee.
    - Procedure add_customer
        - Parameter: name,number.
        - Description: Add customer.



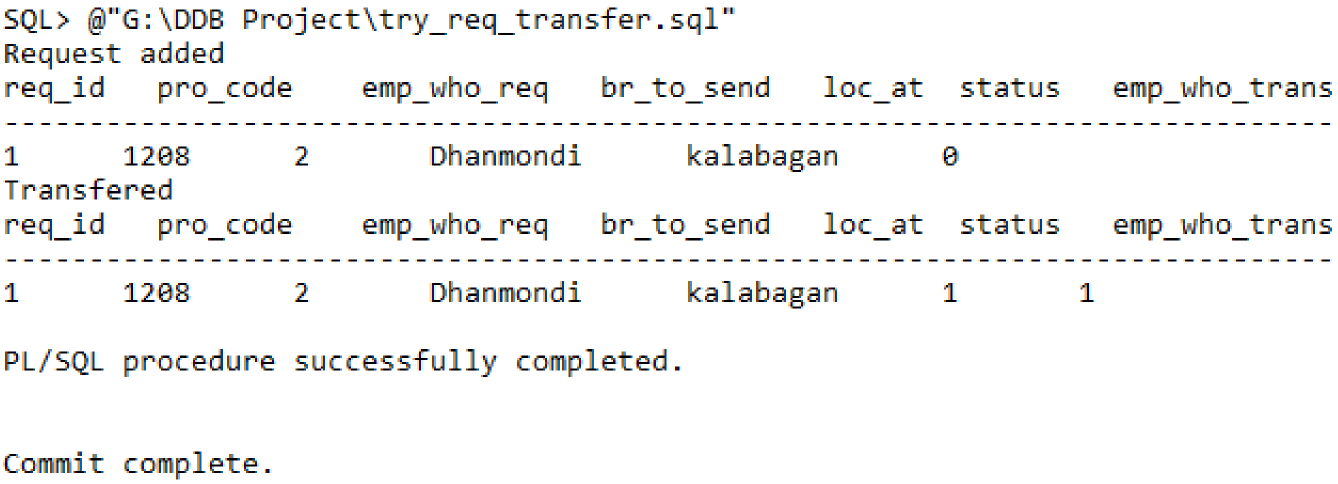
```
SQL> @"C:\Users\azad\Desktop\DDB Project\try_customer_employee_insert.sql"
employee added Sajid
customer added Faiza

PL/SQL procedure successfully completed.


Commit complete.
```

- **Package cart_sell_package**
    - Procedure show_cart
        - Parameter: No parameter.
        - Description: Shows the cart.
    - Procedure empty_cart
        - Parameter: No parameter.
        - Description: Removes the products from the cart.
    - Function add_to_cart()
        - Parameter: product_code.
        - Description: Add product to the cart.
    - Function delete_from_cart()
        - Parameter: product_code.
        - Description: Delete product from the cart.
    - Procedure show_receipt()
        - Parameter: purchase_id.
        - Description: Show the receipt.
    - Function sell()
        - Parameter: employee_id, customer_id.
        - Description: Sell the product to the customer.



```
SQL> @"C:\Users\azad\Desktop\DDB Project\try_showRecept.sql"
Recept:                               Purches ID:1
-----------------------------------------------------
code     id     name     sellprice   brand
-----------------------------------------
1206     2      Nvidia GTX 1060     42500     Gigabyte
-----------------------------------------------------
employee: Swapnil                   customer:Nafis
TOTAL:42500

PL/SQL procedure successfully completed.
```

- **Package transfer_package**
    - Procedure show_request
        - Parameter: no parameter.
        - Description: Shows the request table.
    - Function add_request()
        - Parameter: product_code, send_to_branch,employee_id.
        - Description: Request products from other branches.
    - Function transfer()
        - Parameter: request_id, employee_id.
        - Description: Transfer products from the requested branch.



```
SQL> @"G:\DDB Project\try_req_transfer.sql"
Request added
req_id   pro_code    emp_who_req   br_to_send   loc_at  status   emp_who_trans
-------------------------------------------------------------------------------
1      1208      2       Dhanmondi      kalabagan      0
Transfered
req_id   pro_code    emp_who_req   br_to_send   loc_at  status   emp_who_trans
-------------------------------------------------------------------------------
1      1208      2       Dhanmondi      kalabagan      1       1

PL/SQL procedure successfully completed.


Commit complete.
```

6 Triggers

- product_id
    - Description: Sets the product id of the product table when a new data is inserted.
- supplier_id
    - Description: Sets the supplier id of the product table when a new data is inserted.
- customer_id
    - Description: Sets the customer id of the product table when a new data is inserted.
- employee_id
    - Description: Sets the employee id of the product table when a new data is inserted.
- purchase_id
    - Description: Sets the purchase id of the product table when a new data is inserted.
- request_id
    - Description: Sets the request id of the product table when a new data is inserted.