

TRANSACTION MANAGEMENT SYSTEM

SQL CASE STUDY



3516 – Sowmen Parui
3517 – Sanchita Patil
3518 - Vibha Rajput
3519 - Samriddhi Sharma
3520 - Sanket Veerkar

ABSTRACT:

This report includes a development presentation of an information system for managing the transaction data within a small bank or financial institute. The system as such as it has been developed is called Transaction Management System. It consists of database. The choice of the programming tools is individual.

KEYWORDS:

Information system, Database system, DBMS, parent table, child table, table fields, primary key, foreign key, relationship, SQL queries, objects, classes, controls.

CONTENTS –

A. INTRODUCTION

1. Background
2. Problem Statement
3. Problem Discussion
4. Report Overview

B. PROBLEM'S SOLUTION

1. Method
2. Programming Environments
3. Database analyzing, design and implementation

C. CONCLUSION

A. INTRODUCTION

This case study gives a brief theoretical preview upon the database information systems and goes through the essence of the problem that should be resolved.

1. BACKGROUND

Most of the contemporary Information systems are based on the Database technology as a collection of logically related data, and DBMS as a software system allowing the users to define, create, maintain, and control access to the database.

The process of constructing such kind of systems is not so simple. It involves a mutual development of database. Thus, the well-developed database is very important for the reliability, flexibility, and functionality of the system.

Information system suggests a computer technology to be used to provide information to users in an organization (for instance), as for the purposes of data transformation into useful information; computer hardware and software are designed and used. A particular case is the transaction management system.

This kind of systems are responsible for storing data of the transaction data and generating reports upon request. Dbase Queries Application Program DBMS Database Retrieve Insert Update Delete users. Such kind of system could be integrated with other Information systems or modules:

Transaction management “Organizations depend on Information Systems to stay competitive. Productivity, which is crucial to staying competitive, can be increased through better Information Systems

2. PROBLEM STATEMENT

This report's documentation goes through the whole process of database development. It also comprises the development tools have been utilized for these purposes.

3. PROBLEM DISCUSSION

This system should consist of an application program, on one hand, and a database (repository of data) on the other. The program should perform the basic operations upon the database as retrieving, inserting, updating, and deleting data.

Any additional functionality is a goal of a further module development. It is a kind of strategy to start the development from designing and constructing the database, as this structure will determine the further structure of the application program.

The logical database model (tables, their content, and the relationships between them) should respond to the given task and cover the basic requirements.

4. REPORT OVERVIEW

The next part and its subsections will turn the attention to the method for resolving the problem, the programming environments used for developing the system and the implementation of the operations performed upon the database.

B. PROBLEM'S SOLUTION

This chapter involves some subsections that concern the basic scheme of resolving the given task and comprise both the methods and tools of its development as well.

1. METHOD

At the very commencement, I proceeded to a decision to carry out the development of my task into the following steps:

1. Exploring the available development environments and techniques.
2. Database Analyzing.
3. Database design and Implementation.
4. Program's Structure Analyzing.

Each one of these steps could be explained in some brief details as follows:

1. Exploring the available development environments and techniques There is a lot of programming environments available to be used for such kind of elaborations. The point is to choose such an environment that we will be able to operate with in a convenient and easy way. This is optional and individual process, that depends on the developer's experience as well.

2. Database Analyzing It concerns all the demands, put upon the database content and its functionality. The database should be designed and implemented in a way that the user would expect it to be.

3. Database design and Implementation This step is tightly related with the previous one as it is completely determined by the requirements, analyzed, and discussed in step 2

2.PROGRAMMING ENVIRONMENTS

The given task concerns a small company (organization). For instance, for the needs of a small company, we could use one set of tools, but for the needs of a larger one, it would be much better if we apply our approach by using some different, that could be more appropriate and would fit much better the requirements we have to satisfy. I decided to use the Oracle Database Environment as a Database Management System

Database Environment:.. The evolution of the Database systems could be divided into three phases: the Manual-filing System, the File-based systems, and the Database and the Database Management systems (DBMS)

The DBMS is a powerful set of tools, enabling users to operate easily with data into a database as: Inserting, Updating, Deleting and Retrieving data. It prevents unauthorized access to the database and maintains the consistency of the stored data. The whole information is stored in separate tables in one file. Such an information arrangement is convenient as we can view, add, and delete information using online forms, and search or retrieve by using queries. The term “Database” comprises collection of objects (tables, forms, pages, queries, macros) for manipulating, retrieving, and storing data. It is a very good advantage that the information (data) can be populated in different tables (related to each other) and it is not necessary to store all data into one table. That leads to a less redundancy of data and reduces the required disk storage space and speeds up processing as well. There are three relationships between the tables in a database: (one-to-many), (one-to-one) and (many-to-many).

One of the mostly used relationships is the (one-to-many) type. If we have a main table (so called Parent table) and some other obedient tables (so called Child tables), then let us assume that the relationship between the parent table and the child tables is of type (one-to-many). It means that every single record from the parent table could have several records into each one of its related child tables.

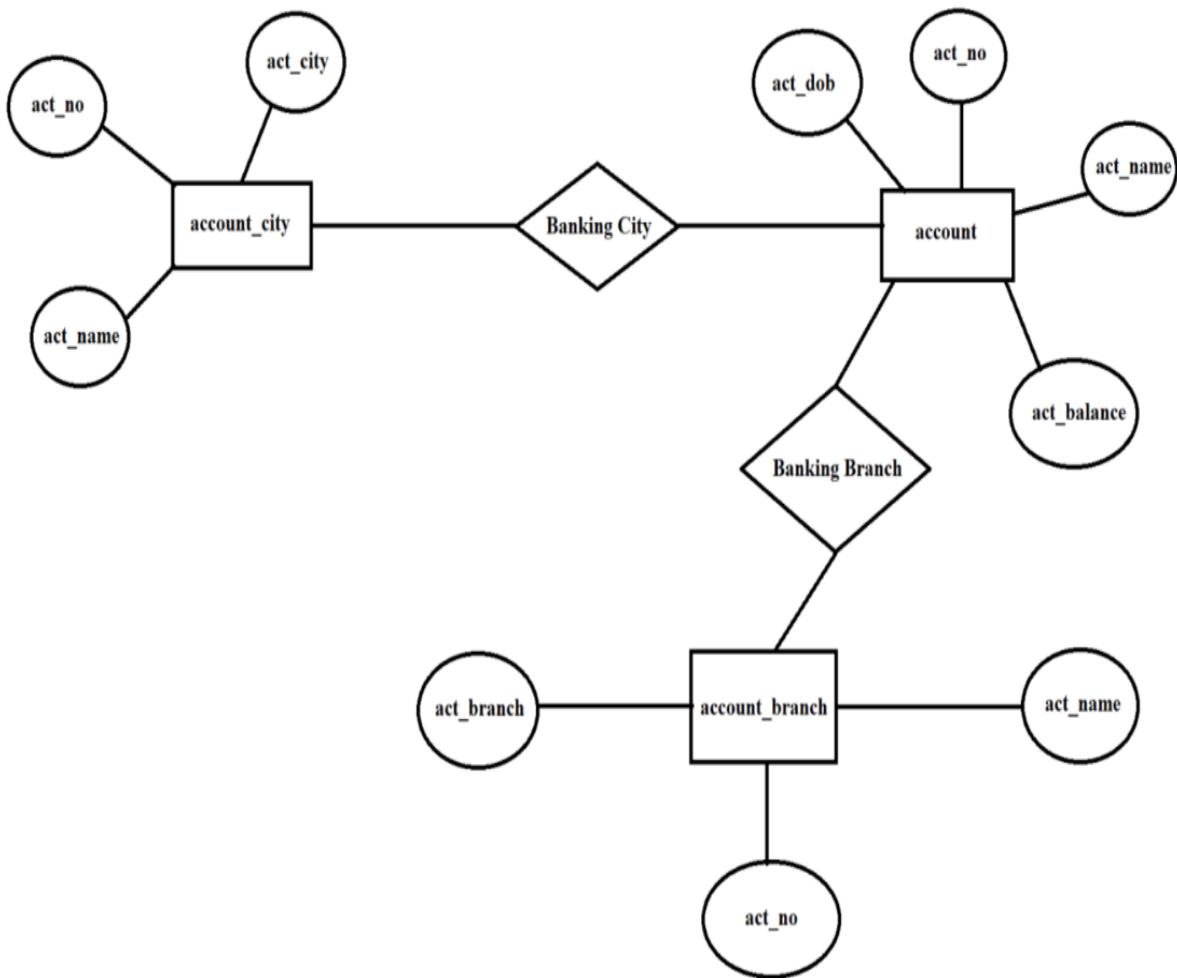
The tables are related by their key-values. The key into the parent table is called primary key, and the keys into its child tables are called foreign keys. The primary and the foreign key have the same unique value.

3.DATABASE ANALYZING, DESIGN AND IMPLEMENTATION

The database for the system should include information of account holders, branches, and branch city of a bank, respectively of its employees. The data is subdivided into the following groups:

Account	Account_Branch	Account_city
act_no varchar2 (17), act_name varchar2 (17), act_balance number (17), act_dob date	act_no varchar2 (17), act_name varchar2 (17), act_branch varchar2 (17)	act_no varchar2 (17), act_name varchar2 (17), act_city varchar2(17)

E – R DIAGRAM

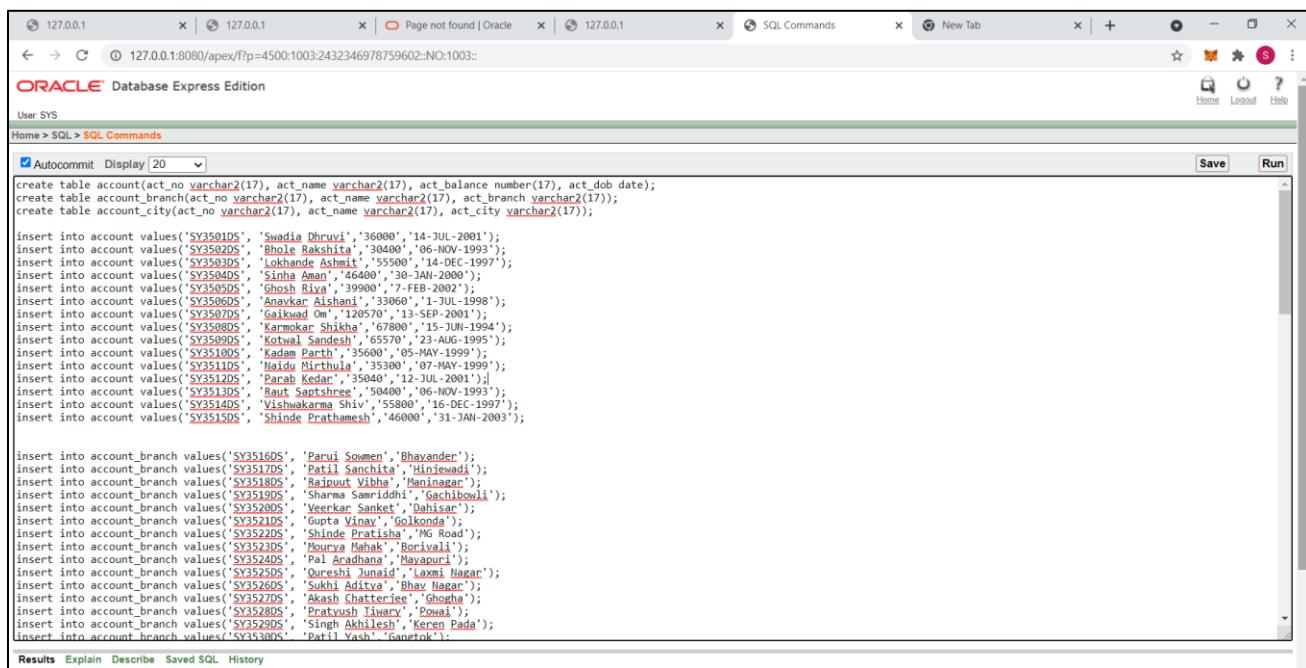


Here, we have **banking city** as a relationship set between the entity sets: `account` and `account_city` and **banking branch** as a relationship set between `account` and `account_branch`. Moving further `account` is subdivided into attributes that are : `act_no`, `act_name`, `act_balance` ,`act_dob`. `Account_city` table is further divided into :`act_city`,`act_name` and `act_no` . `Account_branch` table is further divided into : `act_name` , `act_branch` and `act_no`. This is an example of aggregation model.

CREATING TABLES AND INSERTING DATA -

A. First, we will create 3 tables which are account, account branch and branch city

1. In Account table we will store the information about account holder.
2. In Account_Branch we will put info about branch name of account holder
3. In Account_city we will store data about city of account



```
create table account(act_no varchar2(17), act_name varchar2(17), act_balance number(17), act_dob date);
create table account_branch(act_no varchar2(17), act_name varchar2(17), act_branch varchar2(17));
create table account_city(act_no varchar2(17), act_name varchar2(17), act_city varchar2(17));

insert into account values('SY3501DS', 'Swadia Dhrushi', '36000', '14-JUL-2001');
insert into account values('SY3502DS', 'Bhole Rakshita', '304000', '06-NOV-1993');
insert into account values('SY3503DS', 'Lokhande Ashmit', '55500', '14-DEC-1997');
insert into account values('SY3504DS', 'Sinha Aman', '46400', '30-JAN-2000');
insert into account values('SY3505DS', 'Ghosh Riya', '39900', '7-FEB-2002');
insert into account values('SY3506DS', 'Anavkar Aishani', '33060', '1-JUL-1998');
insert into account values('SY3507DS', 'Gaikwad Um', '120570', '13-SEP-2001');
insert into account values('SY3508DS', 'Karmkarkar Shikha', '67800', '15-JUN-1994');
insert into account values('SY3509DS', 'Korade Samarth', '45000', '22-AUG-1995');
insert into account values('SY3510DS', 'Mujde Bhat', '35600', '05-MAY-1999');
insert into account values('SY3511DS', 'Naide Mirthula', '35300', '07-MAY-1999');
insert into account values('SY3512DS', 'Parab Kedar', '35040', '12-JUL-2001');
insert into account values('SY3513DS', 'Raut Santoshree', '50400', '06-NOV-1993');
insert into account values('SY3514DS', 'Vishwas Karma Shiv', '55800', '16-DEC-1997');
insert into account values('SY3515DS', 'Shinde Prathamesh', '46000', '31-JAN-2003');

insert into account_branch values('SY3516DS', 'Parui Sowmen', 'Bhavander');
insert into account_branch values('SY3517DS', 'Patil Sanchita', 'Minjewadi');
insert into account_branch values('SY3518DS', 'Rajput Vibha', 'Maninagar');
insert into account_branch values('SY3519DS', 'Vaidya Sandip', 'Gachibowli');
insert into account_branch values('SY3520DS', 'Veenkar Sanket', 'Danisar');
insert into account_branch values('SY3521DS', 'Gupta Vinay', 'Golkonda');
insert into account_branch values('SY3522DS', 'Shinde Pratisha', 'MG Road');
insert into account_branch values('SY3523DS', 'Mourya Mahak', 'Ronivali');
insert into account_branch values('SY3524DS', 'Pal Aradhana', 'Mayapur');
insert into account_branch values('SY3525DS', 'Qureshi Junaid', 'Laxmi Nagar');
insert into account_branch values('SY3526DS', 'Sukhi Aditya', 'Bhav Nagar');
insert into account_branch values('SY3527DS', 'Akash Chatterjee', 'Ghophra');
insert into account_branch values('SY3528DS', 'Pratyush Tiwary', 'Pawal');
insert into account_branch values('SY3529DS', 'Singh Akhilesh', 'Keren Pada');
insert into account_branch values('SY3530DS', 'Patil Yash', 'Gantotri');
```

```

ORAQUEL Database Express Edition
User SYS
Home > SQL > SQL Commands
 Autocommit Display 20  Save  Run
insert into account_branch values('SY3516DS', 'Parui Sommen', 'Bhayander');
insert into account_branch values('SY3517DS', 'Patil Sanchita', 'Hinewadi');
insert into account_branch values('SY3518DS', 'Raipuri Vibha', 'Maninagar');
insert into account_branch values('SY3519DS', 'Sharma Samiriddhi', 'Gachibowli');
insert into account_branch values('SY3520DS', 'Veerkar Sanket', 'Dahisar');
insert into account_branch values('SY3521DS', 'Gupta Vinay', 'Golkonda');
insert into account_branch values('SY3522DS', 'Shinde Pratisha', 'MG Road');
insert into account_branch values('SY3523DS', 'Mehra Bhavna', 'Berivali');
insert into account_branch values('SY3524DS', 'Pal Aradhana', 'Mankhurd');
insert into account_branch values('SY3525DS', 'Qureshi Junaid', 'Laxmi Nagar');
insert into account_branch values('SY3526DS', 'Sukhi Aditua', 'Bhay Nagar');
insert into account_branch values('SY3527DS', 'Akash Chatterjee', 'Ghophla');
insert into account_branch values('SY3528DS', 'Pratyush Tiswary', 'Pawai');
insert into account_branch values('SY3529DS', 'Singh Akhilesh', 'Keren Pada');
insert into account_branch values('SY3530DS', 'Patil Yash', 'Gangtok');

insert into account_city values('SY3531DS', 'Karpe Yash', 'Vadodara');
insert into account_city values('SY3532DS', 'Rathore Jagdish', 'Bangalore');
insert into account_city values('SY3533DS', 'Vishwakarma Vishal', 'Delhi');
insert into account_city values('SY3534DS', 'Gupta Akanksha', 'Mumbai');
insert into account_city values('SY3535DS', 'Tiwari Shubham', 'Bhavnagar');
insert into account_city values('SY3536DS', 'Yadav Roma', 'Mumbai');
insert into account_city values('SY3537DS', 'Lavanya Karan', 'Kolkata');
insert into account_city values('SY3538DS', 'Bansle Dinesh', 'Pune');
insert into account_city values('SY3539DS', 'Shukla Ruchi', 'Ahmedabad');
insert into account_city values('SY3540DS', 'Morva Kundan Singh', 'Hyderabad');
insert into account_city values('SY3541DS', 'Aditee Patil', 'Vaddoda');
insert into account_city values('SY3542DS', 'Aditi Singh', 'Bangalore');
insert into account_city values('SY3543DS', 'Ajay Mahi', 'Delhi');
insert into account_city values('SY3544DS', 'Karan Lavate', 'Mumbai');
insert into account_city values('SY3545DS', 'Trishna Shreshtha', 'Bhavnagar');

```

select * from account;
select * from account_branch;

```

ORAQUEL Database Express Edition
User SYS
Home > SQL > SQL Commands
 Autocommit Display 20  Save  Run
select * from account;
select * from account_branch;
select * from account_city;

select avg(act_balance)
from account;

select count(*)
from account_city;

select act_name,act_dob
from account
where act_name like 'S%';

select act_no, act_balance
from account
where act_balance in (select act_balance
                      from account
                      where act_balance >= 55000);

DECLARE
xact_no account.act_no%type;
xact_balance account.act_balance%type;
opt number(1);
deposit number(17);
withdrawal number(17);

BEGIN
xact_no:= :Enter_Account_Number;
opt:= :Deposit_Withdrawal_Balance;
select act_balance into xact_balance
from account
where act_no=xact_no ;
if opt = 1 then

```

Result [Execute](#) [Describe](#) [Saved SQL](#) [History](#)

```

DECLARE
    xact_no account.act_no%type;
    act_balance account.act_balance%type;
    opt number(1);
    deposit number(17);
    withdrawal number(17);

BEGIN
    xact_no:= :Enter_Account_Number;
    opt:= :Deposit_Withdrawal_Balance;
    select act_balance into xact_balance
    from account
    where act_no=xact_no ;
    |
    IF( opt = 1 ) THEN
        deposit:= :Enter_Deposit_Amount;
        update account
        set act_balance=act_balance+deposit
        where act_no=xact_no ;
        xact_balance :=xact_balance +deposit;
        dbms_output.put_line('Rs.'||deposit||' has been credited in your account');
        dbms_output.put_line('Your final balance is Rs.'||xact_balance );
    ELSEIF( opt = 2 ) THEN
        withdrawal:= :Enter_Withdrawal_Amount;
        If ( withdrawal>xact_balance ) THEN
            update account
            set act_balance=act_balance-withdrawal
            where act_no=xact_no ;
            xact_balance:=xact_balance -withdrawal;
            dbms_output.put_line('Rs.'||withdrawal||' has been debited in your account');
            dbms_output.put_line('Your final balance is Rs.'||xact_balance );
        ELSE
            dbms_output.put_line('Your withdrawal amount is greater than balance');
            dbms_output.put_line('Your transaction failed');
            dbms_output.put_line('Your final balance is Rs.'||xact_balance );
        END IF;
    ELSE
        dbms_output.put_line('Your balance is Rs.'||xact_balance );
    END IF;
END;
/

```

```

from account
where act_no=xact_no ;
|
IF( opt = 1 ) THEN
    deposit:= :Enter_Deposit_Amount;
    update account
    set act_balance=act_balance+deposit
    where act_no=xact_no ;

    xact_balance :=xact_balance +deposit;
    dbms_output.put_line('Rs.'||deposit||' has been credited in your account');
    dbms_output.put_line('Your final balance is Rs.'||xact_balance );

ELSEIF( opt = 2 ) THEN
    withdrawal:= :Enter_Withdrawal_Amount;
    If ( withdrawal>xact_balance ) THEN
        update account
        set act_balance=act_balance-withdrawal
        where act_no=xact_no ;
        xact_balance:=xact_balance -withdrawal;
        dbms_output.put_line('Rs.'||withdrawal||' has been debited in your account');
        dbms_output.put_line('Your final balance is Rs.'||xact_balance );
    ELSE
        dbms_output.put_line('Your withdrawal amount is greater than balance');
        dbms_output.put_line('Your transaction failed');
        dbms_output.put_line('Your final balance is Rs.'||xact_balance );
    END IF;
ELSE
    dbms_output.put_line('Your balance is Rs.'||xact_balance );
END IF;

END;
/

```

B. Select the statement and click on run.

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 20

```
insert into account_city values('SY351005', 'Parul Suman', 'Mumbai');
insert into account_city values('SY351005', 'Patil Sanjita', 'Kolkata');
insert into account_city values('SY351005', 'Rajput Vibha', 'Pune');
insert into account_city values('SY351005', 'Sharma Samridhi', 'Ahmedabad');
insert into account_city values('SY352005', 'Veerkar Sanket', 'Hyderabad');

select * from account;
select * from account_branch;
select * from account_city;
select avg(act_balance)
from account;
select count(*)
```

Results Explain Describe Saved SQL History

ACT_NO	ACT_NAME	ACT_BALANCE	ACT_DOB
SY350105	Swadee Devvi	36000	14-JUL-91
SY350205	Bhole Raisinha	30400	06-NOV-93
SY350305	Lohande Ashmit	55500	14-DEC-97
SY350405	Sinha Aman	40400	30-JAN-00
SY350505	Ghosh Riya	36000	07-FEB-02
SY350605	Anavkar Aishani	33000	01-JUL-99
SY350705	Gakkad Om	120579	12-SEP-91
SY350805	Karmolar Shilpa	67800	15-JUN-94
SY350905	Kothari Sandesh	65579	23-AUG-95
SY351005	Kadam Parth	35600	05-MAY-99
SY351105	Naidu Mirthula	35300	07-MAY-99
SY351205	Pawar Kedar	36040	12-JUL-91
SY351305	Raut Sapthlee	50400	06-NOV-93
SY351405	Vishwakarma Shiv	55800	16-DEC-97
SY351505	Mayekar Prathmesh	46000	31-JAN-03
SY351605	Patel Sumeen	376600	15-MAY-02
SY351705	Pell Sanjita	360560	03-JUL-90
SY351805	Rajput Vibha	1244070	27-SEP-91
SY351905	Sharma Samridhi	87800	11-JUN-94
SY352005	Veerkar Sanket	65600	22-AUG-95

20 rows returned in 0.35 seconds

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 20

```
insert into account_city values('SY351005', 'Parul Suman', 'Mumbai');
insert into account_city values('SY351005', 'Patil Sanjita', 'Kolkata');
insert into account_city values('SY351005', 'Rajput Vibha', 'Pune');
insert into account_city values('SY351005', 'Sharma Samridhi', 'Ahmedabad');
insert into account_city values('SY352005', 'Veerkar Sanket', 'Hyderabad');

select * from account;
select * from account_branch;
select * from account_city;
select avg(act_balance)
from account;
select count(*)
```

Results Explain Describe Saved SQL History

ACT_NO	ACT_NAME	ACT_BRANCH
SY350105	Swadee Devvi	Manjarpur
SY350205	Bhole Raisinha	MG Road
SY350305	Lohande Ashmit	Bonvall
SY350405	Sinha Aman	Mayapuri
SY350505	Ghosh Riya	Ghophna
SY350605	Anavkar Aishani	Powai
SY350705	Gakkad Om	Hinjewadi
SY350805	Karmolar Shilpa	Mannagar
SY350905	Kothari Sandesh	Gachibowli
SY351005	Kadam Parth	Mira Road
SY351105	Naidu Mirthula	Dahsar
SY351205	Pawar Kedar	Golkonda
SY351305	Raut Sapthlee	Shiv road
SY351405	Vishwakarma Shiv	Meena Nagar
SY351505	Mayekar Prathmesh	Dhanu
SY351605	Patel Sumeen	Hayander
SY351705	Pell Sanjita	Hinjewadi
SY351805	Rajput Vibha	Mannagar
SY351905	Sharma Samridhi	Gachibowli
SY352005	Veerkar Sanket	Dahzar

20 rows returned in 6.04 seconds

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
insert into account_city values('SY351005', 'Paril Suman', 'Mumbai');
insert into account_city values('SY351105', 'Paril Sanchita', 'Kolkata');
insert into account_city values('SY351205', 'Rajiv+ Vibha', 'Pune');
insert into account_city values('SY351305', 'Sharma Sanidhi', 'Ahmedabad');
insert into account_city values('SY352005', 'Veerkar Sanjat', 'Hyderabad');

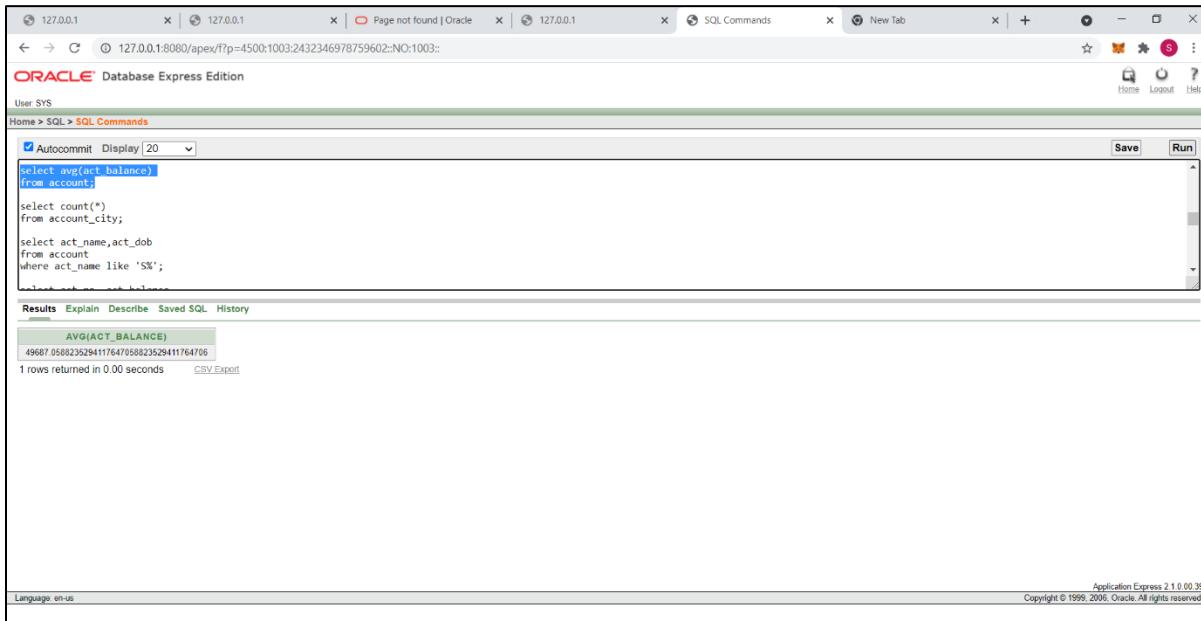
select * from account;
select * from account_branch;
select * from account_city;
select avg(act_balance)
from account;
select count(*)
```

ACT_NO	ACT_NAME	ACT_CITY
SY350105	Savda Divvi	Vadodara
SY350205	Bhole Rastrak	Bangalore
SY350305	Lomande Ashmit	Delhi
SY350405	Sinha Aman	Mumbai
SY350505	Groh Riya	Ernakulam
SY350605	Anavkar Jishani	Mumbai
SY350705	Oakwood Om	Kolkata
SY350805	Karmoor Shikha	Pune
SY350905	Kokil Sondech	Ahmedabad
SY351005	Kaden Path	Hyderabad
SY351105	Naidu Mithila	Vadodara
SY351205	Parol Kellar	Bangalore
SY351305	Raut Sapnaee	Delhi
SY351405	Vithusakarma Shy	Mumbai
SY351505	Mayekar Prathmesh	Ernakulam
SY351605	Parus Soumee	Mumbai
SY351705	Pall Sanchita	Kolkata
SY351805	Rajput Vibha	Pune
SY351905	Sharma Samridhi	Ahmedabad
SY352005	Veekur Sanjat	Hyderabad

20 rows returned in 0.06 seconds

c. Creating Queries according to our database.

QUERY 1 : TO FIND THE AVERAGE OF ACCOUNT BALANCE FROM ACCOUNT TABLE.



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
select avg(act_balance)
from account;
select count(*)
from account_city;
select act_name,act_dob
from account
where act_name like 'SK';
```

The results section displays the output of the first query:

AVG(ACCOUNT_BALANCE)
49887.0589235294117647058823529411764706

1 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Here, with the help of aggregate function: avg (), we have found the average amount of balance present in the account table.

QUERY 2: TO FIND AVERAGE OF BALANCE PRESENT IN ACCOUNT

```
select avg(act_balance)
from account;
select count(*)
from account_city;
select act_name,act_dob
from account
where act_name like 'S%';
```

Results Explain Describe Saved SQL History

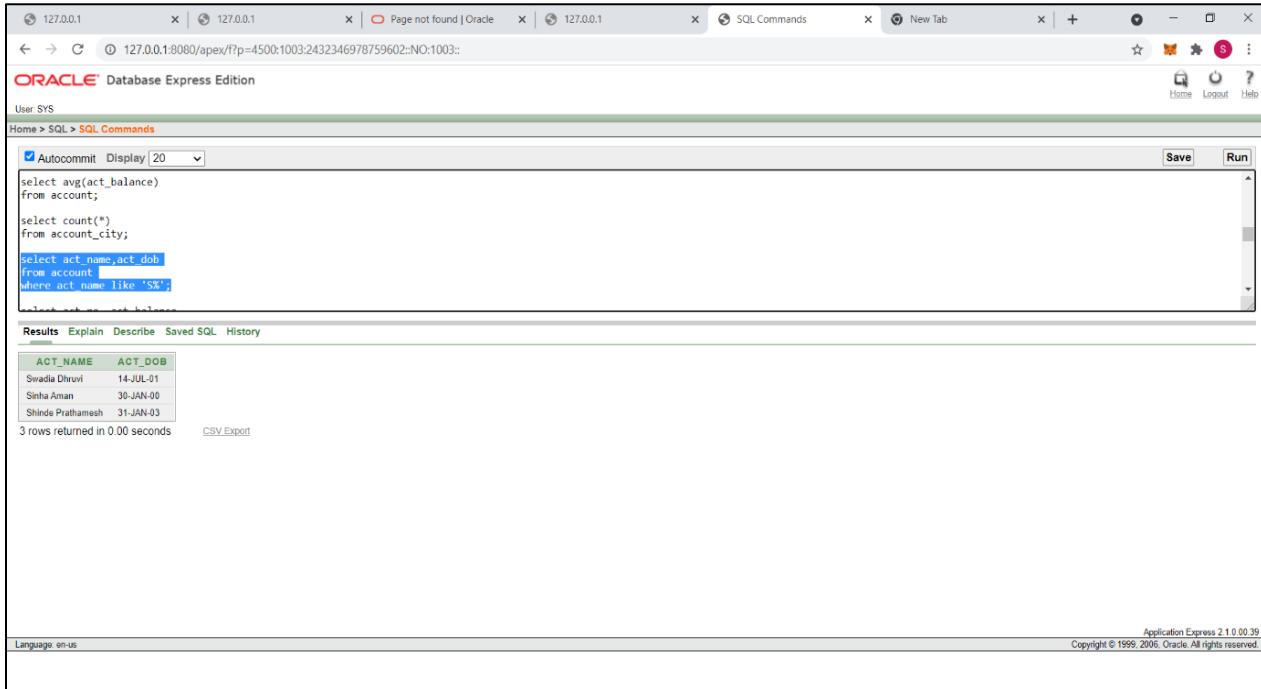
COUNT(*)
18

1 rows returned in 0.00 seconds CSV Export

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2005, Oracle. All rights reserved.

In this query, with the help of count function we have found no.of rows present in account_city table.

QUERY 3: TO FIND THE NAME AND DOB OF THE ACCOUNTANTS WITH THEIR LAST NAME STARTING WITH “S”.



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
select avg(act_balance)
from account;
select count(*)
from account_city;
select act_name,act_dob
from account
where act_name like 'S%';
```

The results section displays a table with three rows:

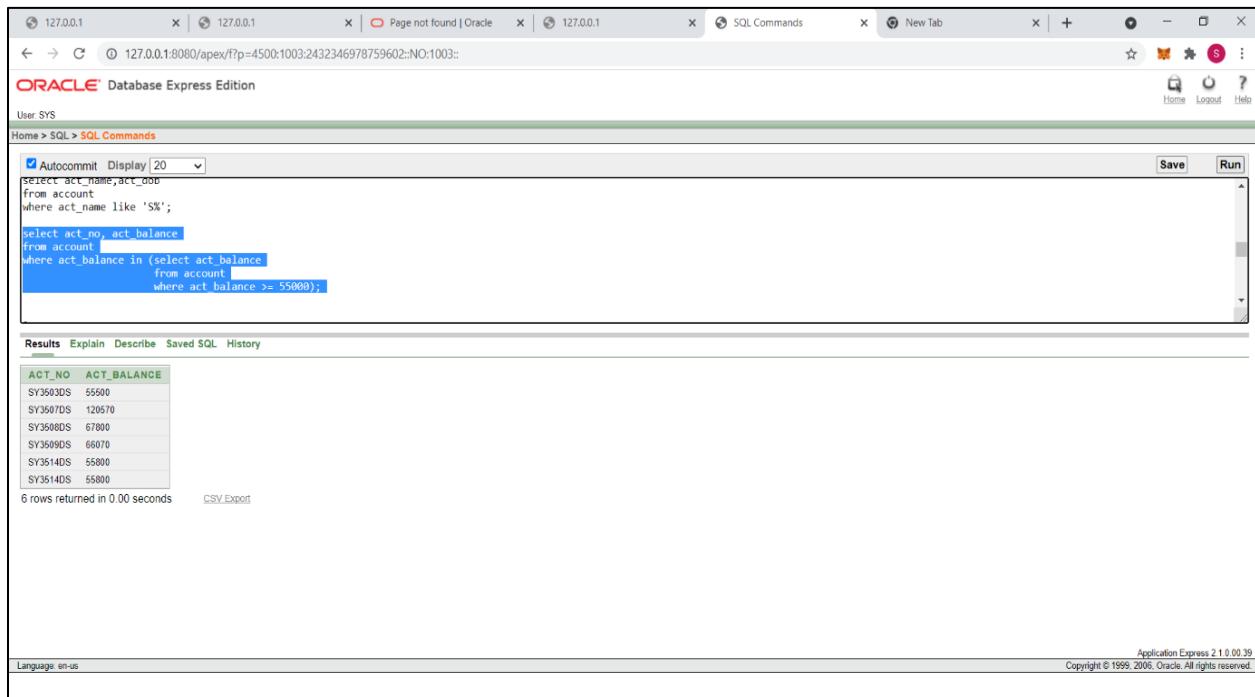
ACT_NAME	ACT_DOB
Sravita Dhrovi	14-JUL-01
Sinha Aman	30-JAN-00
Shinde Prathamesh	31-JAN-03

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39
Copyright © 1995, 2006, Oracle. All rights reserved.

In this query , we have selected act_name and act_dob from account table to find the name of the accountants whose last name's first letter start with S.

QUERY 4: LIST THE ACCOUNT NUMBER AND BALANCE OF ALL THE ACCOUNTANTS WHO HAVE MORE THAN 55000 IN THEIR ACCOUNT. (USING SUB QUERY)



The screenshot shows a browser window with multiple tabs open. The active tab is titled "SQL Commands". The URL is 127.0.0.1:8080/apex/f?p=4500:1003:2432346978759602::NO::1003:. The page title is "ORACLE Database Express Edition". The SQL command entered is:

```
select act_name,act_no
from account
where act_name like '%';

select act_no, act_balance
from account
where act_balance in (select act_balance
                      from account
                      where act_balance >= 55000);
```

The results section displays a table with two columns: ACT_NO and ACT_BALANCE. The data is:

ACT_NO	ACT_BALANCE
SY3593DS	55500
SY3507DS	120570
SY3508DS	67800
SY3509DS	66070
SY3514DS	55800
SY3514DS	55800

6 rows returned in 0.00 seconds. There is a "CSV Export" link.

At the bottom, it says "Language: en-us" and "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

Here, we have selected act_no and act_balance from account table and further continuing we have used sub query with where clause to find the act_balance of all the accountants who have 55000 balance in their accounts.

TO FIND:

1. DEPOSIT

- Depositing Cash

As we can see in below picture , the following SQL query is to deposit money in account first you have to enter account number then you have to enter 1 as we have to deposit. Then you have to enter deposit amount. And then click on submit.

The screenshot shows the Oracle Database Express Edition interface. In the main window, there is a SQL command window with the following code:

```
DECLARE
xact_no account.act_no%type;
xact_balance account.act_balance%type;
opt number(1);
deposit number(17);
withdrawal number(17);

BEGIN
:ENTER_ACCOUNT_NUMBER:=:SY3513DS;
opt:=:DEPOSIT;
withdrawal:=:BALANCE;
select act_balance into xact_balance
from account
where act_no=xact_no;

IF( opt = 1 ) THEN
deposit:=:ENTER_DEPOSIT_AMOUNT;
update account
set act_balance=act_balance+deposit
where act_no=xact_no;

xact_balance :=act_balance+deposit;
dbms_output.put_line('Rs.'||deposit||' has been credited in your account');
dbms_output.put_line('Your final balance is Rs.'||xact_balance);

ELSIF( opt = 2 ) THEN
withdrawal:=:WITHDRAWAL_AMOUNT;
IF( withdrawal>xact_balance ) THEN
update account
set act_balance=act_balance-withdrawal
where act_no=xact_no;

xact_balance:=xact_balance-withdrawal;
dbms_output.put_line('Rs.'||withdrawal||' has been debited in your account');
dbms_output.put_line('Your final balance is Rs.'||xact_balance);
END IF;
END;
```

A separate browser window titled "Enter Bind Variables - Google Chrome" is overlaid on the main window, showing the following input fields:

:ENTER_ACCOUNT_NUMBER	SY3513DS
:DEPOSIT	1
:ENTER_DEPOSIT_AMOUNT	1500
:ENTER_WITHDRAWAL_AMOUNT	

After clicking you can see message that amount is credited and final balance after transaction.

The screenshot shows the Oracle Database Express Edition interface with the results of the SQL command execution. The results pane displays the following output:

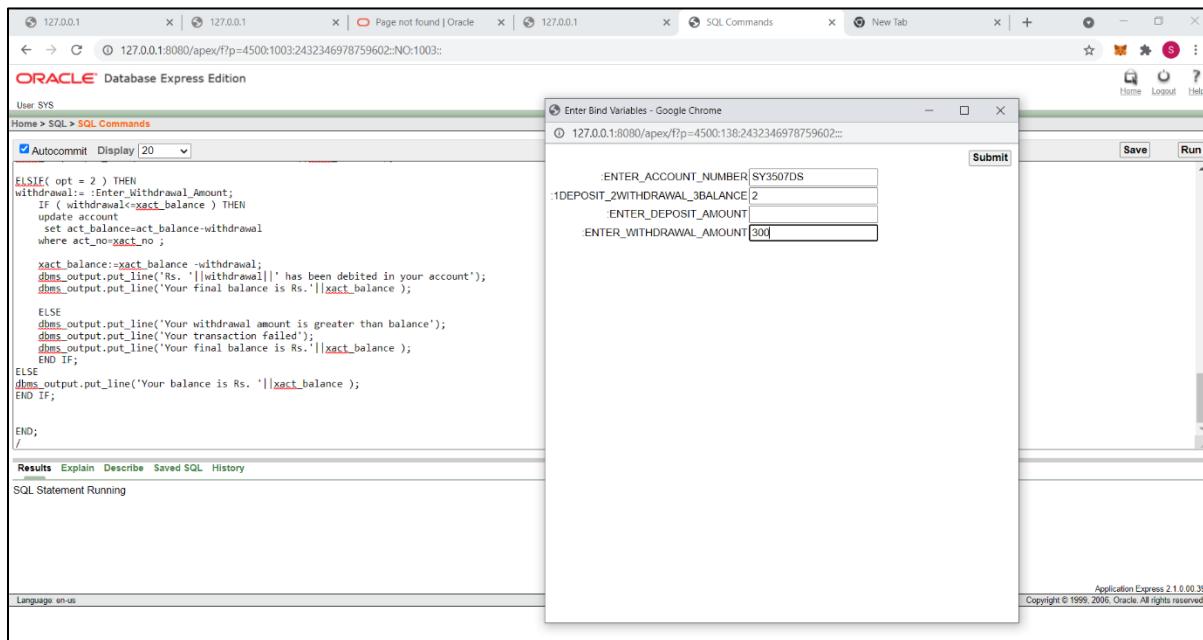
```
1 row(s) updated.

0.02 seconds
```

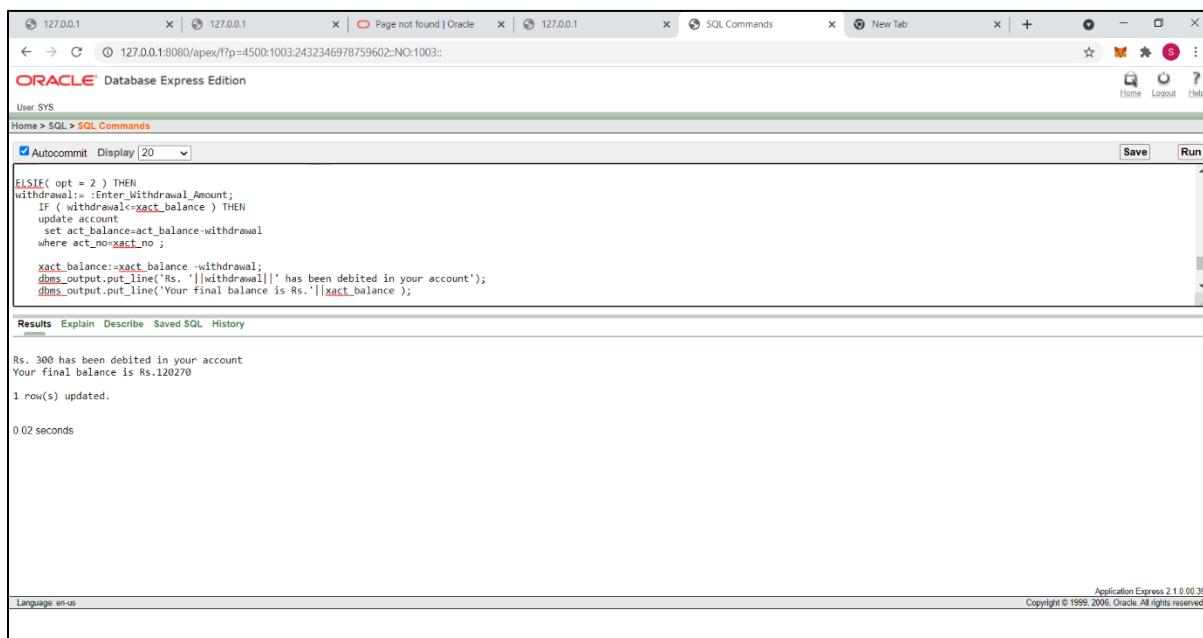
At the bottom of the results pane, it says "Language: en-us".

2. WTHDRAWAL

- For withdrawning amount, first you have to enter account number. Then you have to enter 2 and enter withdrawal amount then click on submit.



After clicking submit you see a message of amount debited and then final balance after the transaction.



3. BALANCE

For checking account balance first you have to enter account number and then enter in following tab and click on submit.

The screenshot shows the Oracle Database Express Edition interface. On the left, a SQL Commands window displays a PL/SQL block. The code handles deposit and withdrawal operations, including updates to account balance and error handling for insufficient funds. On the right, a separate browser window titled "Enter Bind Variables - Google Chrome" shows input fields for "ENTER_ACCOUNT_NUMBER" (set to SY3510DS), "ENTER_DEPOSIT_AMOUNT" (set to 3), and "ENTER_WITHDRAWAL_AMOUNT". A "Submit" button is visible in the top right of the browser window.

After clicking on submit you can see your balance.

The screenshot shows the Oracle Database Express Edition interface after the "Submit" button was clicked. The SQL Commands window now displays the result of the executed PL/SQL block. It shows the account balance has been updated to 35600. The message "Your balance is Rs. 35600" is displayed at the bottom of the results area. The status bar at the bottom right indicates "Application Express 2.1.0.0.39 Copyright © 1999, 2009, Oracle. All rights reserved."

TRANSACTION FAILURES:

a. If withdrawal amount > Balance

If your balance amount is less than your withdrawal amount the transaction will fail and will show your balance.

The screenshot shows a PL/SQL block being run in the SQL Commands window of Oracle Database Express Edition. The code attempts to update an account's balance by subtracting the withdrawal amount from the current balance. If the withdrawal amount is greater than the balance, it prints an error message and exits. Otherwise, it prints a success message and updates the balance. The results show that the withdrawal failed because the amount was greater than the balance, and the final balance remains at Rs. 30400.

```
IF ( withdrawal<xact_balance ) THEN
    update account
    set act_balance=act_balance-withdrawal
    where act_no=xact_no ;
    xact_balance:=xact_balance-withdrawal;
    dbms_output.put_line('Rs.'||withdrawal||' has been debited in your account');
    dbms_output.put_line('Your final balance is Rs.'||xact_balance );
ELSE
    dbms_output.put_line('Your withdrawal amount is greater than balance');
    dbms_output.put_line('Your transaction failed');
    dbms_output.put_line('Your final balance is Rs.'||xact_balance );
END IF;
```

Results Explain Describe Saved SQL History

Your withdrawal amount is greater than balance
Your transaction failed
Your final balance is Rs.30400
1 row(s) updated.
0.00 seconds

Language: en-us Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

b. If you enter number more than 3 in transaction choice box

You have 3 choices for transaction 1. Deposit ,2. Withdrawal 3. Balance check ; but if you enter other than these 3 number the transaction will fail

The screenshot shows a PL/SQL block being run in the SQL Commands window of Oracle Database Express Edition. The code checks the value of 'opt'. If 'opt' is 1, it prints a success message and updates the balance. If 'opt' is 2, it prints a success message and updates the balance. If 'opt' is 3, it prints a success message and updates the balance. If 'opt' is greater than 3, it prints an error message stating 'Sorry, there is no option of greater than 3'. The results show that the transaction failed because the entered option (4) was greater than 3, and the final balance remains at Rs. 30400.

```
xact_balance:=xact_balance-withdrawal;
dbms_output.put_line('Rs.'||withdrawal||' has been debited in your account');
dbms_output.put_line('Your final balance is Rs.'||xact_balance );

ELSIF ( opt = 2 )
    dbms_output.put_line('Your withdrawal amount is greater than balance');
    dbms_output.put_line('Your transaction failed');
    dbms_output.put_line('Your final balance is Rs.'||xact_balance );
END IF;
ELSIF ( opt = 3 )
    dbms_output.put_line('Your balance is Rs.'||xact_balance );
END IF;

ELSE
    dbms_output.put_line('Sorry, there is no option of greater than 3');
END IF;
```

Results Explain Describe Saved SQL History

Sorry, there is no option of greater than 3
1 row(s) updated.
0.02 seconds

CONCLUSION:

- In this report, an information system's development has been presented. It was emphasized on the basic steps, consequently taken during the project's development course as a particular attention was turned to the basic operative functions performed upon the data into the database.
- The report's content comprises the whole task solution, starting from the programming environments have been selected, going through the database, and writing queries. As a future work, some additional stuff could be implemented and integrated into the application code making it much more reliable and flexible; especially what concerns a pay-roll module, for instance. Apparently, the role of such systems is basic and essential within each company that wants to keep a good control and record concerning its transactions, functionality, and performance on all levels in its structure. Every Bank, in nowadays, has the necessity of managing its transactions on a really good level as the staff has the greatest merit of building up a company as such as it is.
- The well managed transactions means giving the appropriate financial award-ness and all kind of benefits as such as they have been deserved. That's why the development of such systems is not just a programming business – a lot of people are ordinarily involved in such projects and one of the basic requirements is the reliability of the system, especially what concerns the storage of data and all the operations that will be performed upon it.

THANK YOU
