

A

**Project Activity Report  
Submitted for Database Management System  
(UCS-310)**

---

**TOPIC : BANK DATABASE MANAGEMENT  
SYSTEM**

---

**Submitted by:**

**Arpit Sagar (102003130)  
Saransh Upadhyay (102003131)  
Rupinderpal Singh (102003167)  
Madhvan Jindal (102003184)**

**Submitted to**

**Sugandhi Ma'am**



**THAPAR INSTITUTE  
OF ENGINEERING & TECHNOLOGY  
(Deemed to be University)**

**DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING  
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,  
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB  
INDIA  
Jan-June 2022**

## **ABSTRACT:**

The Satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance has been a source of inspiration throughout the course of this project. We take this opportunity to express our gratitude to all those who have helped us in this project.

Our Sincere thanks to our guide **Sugandhi ma'am** for giving us their support and constant encouragement for completion of the project successfully.

## **INTRODUCTION:**

Usually, all persons want money for personal and commercial purposes. Banks are the oldest lending institutions in Indian scenario. They are providing all facilities to all citizens for their own purposes by their terms. To survive in this modern market every bank implements so many new innovative ideas, strategies, and advanced technologies. Banks can play a dynamic role in delivery and purchase of consumer durables.

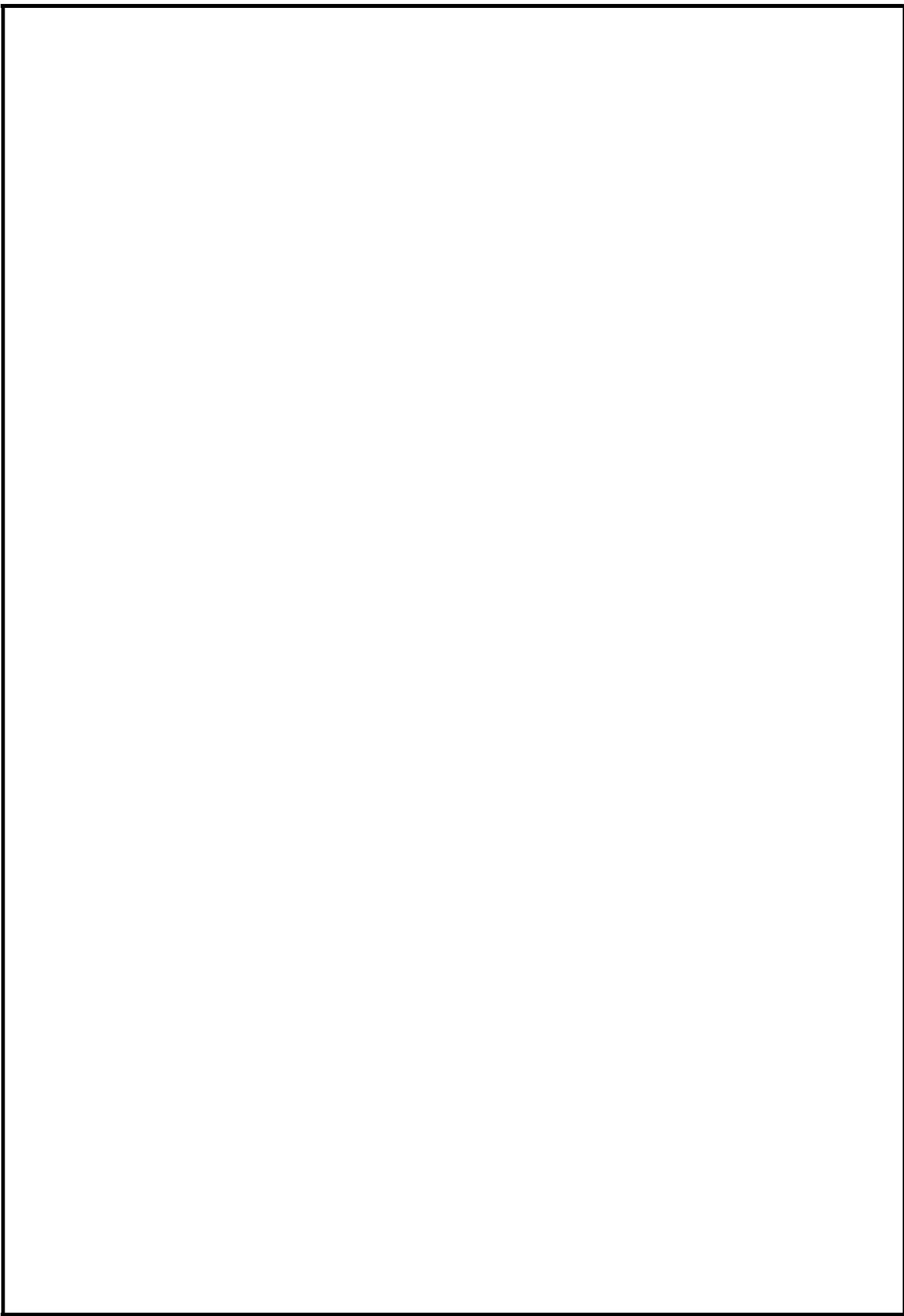
## **OVERVIEW OF PROJECT :**

The purpose of this project is to create a database management system for a bank. The DDL and DML statements have been written using Oracle PL/SQL developer. This project intends to provide a simplistic approach towards designing a database for a banking system. In any banking database system the most important relationship is that of the customer to the account and simultaneously loan database of customers.

The bank database schema has a combination of multiple tables, where we will creating database schema tables like customer, branch, account, loan.

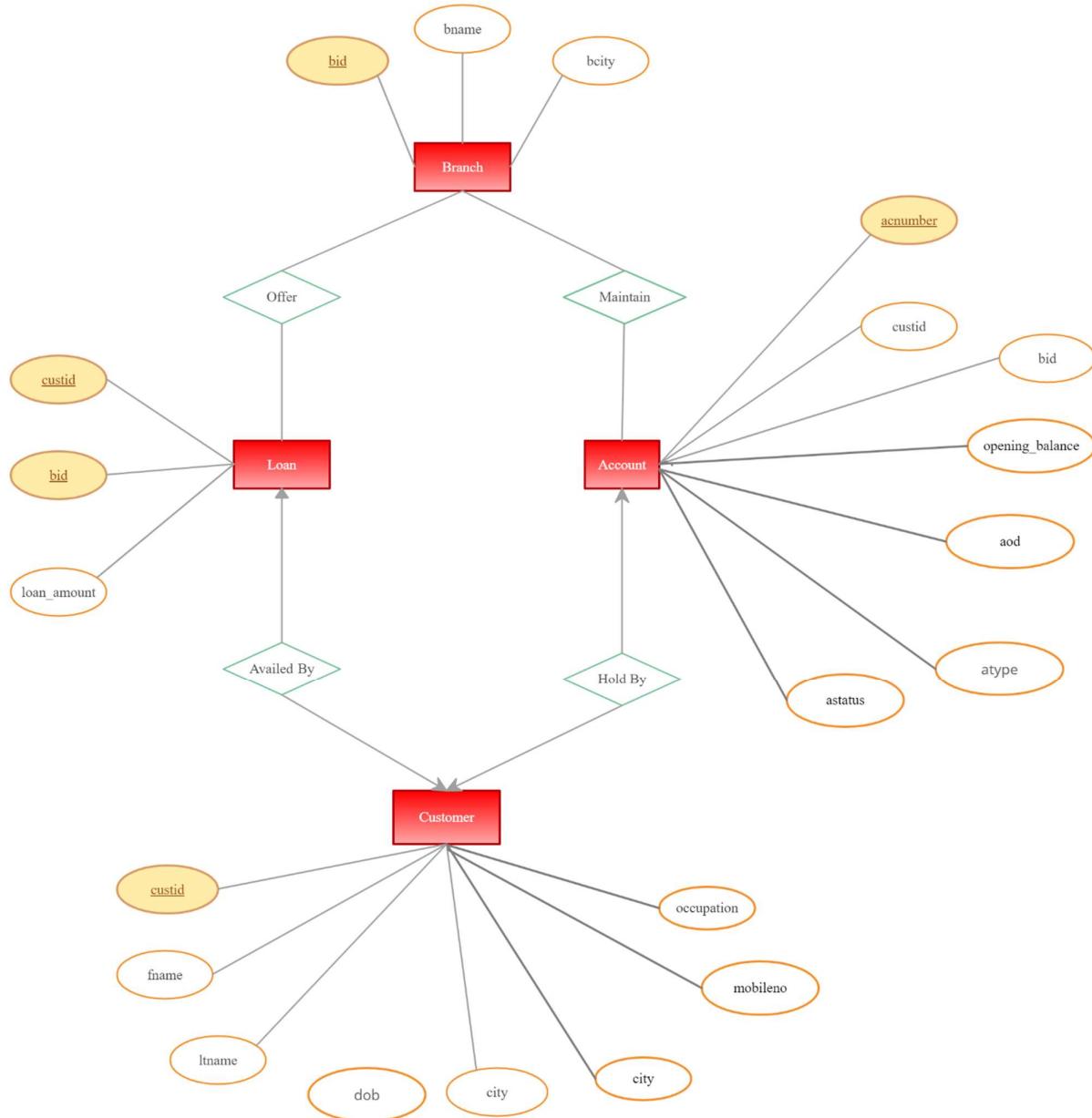
As a part of this project the essential details of the customer are to be captured like name, address, date of birth, phone number, etc. A customer id is assigned to every customer who wants to be a part of the banking system. This customer id acts as a key to the other details related to the customer such as account information, loan information .The accounts table carries important information about the account such as the balance in the account and the type of the account. Each of the entries of the account carry a customer id associated with the account to form a correct relation. Going with the industry standard we have chosen to include the details of any loan that has been issued to the customer as a part of the banking relationship.

As with every customer most of the accounts are opened at a branch so it is logical to have a relationship between a customer and a branch. Every branch can have one or many customers associated with it. This makes it easier for the administrator or a bank employee to look up and track the branch where an account was first opened. This project encompasses a complete view of how a banking system database would look like. All the tables mentioned above intend to capture and retain important information related to the account and the customer. SQL queries can then be used to query the details of customers, customer phone numbers, accounts or loans.

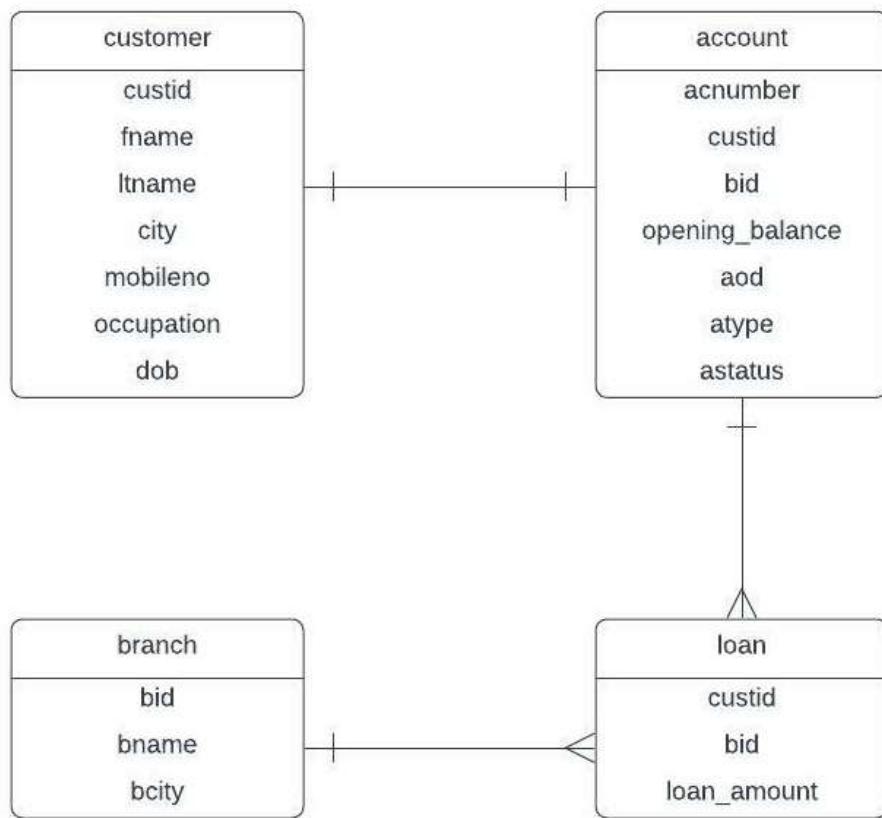


# Banking Database Management System

## ER Diagram



## ER to table



# BANKING DATABASE MANAGEMENT SYSTEM

The screenshot shows the Oracle Live SQL interface. In the SQL Worksheet, the following SQL code is run:

```
CREATE TABLE customer
(
    custid VARCHAR(6),
    fname VARCHAR(30),
    lname VARCHAR(30),
```

The output shows the table creation status:

Table created.

Below the table creation message, the table structure is displayed:

| Column     | Null?    | Type         |
|------------|----------|--------------|
| CUSTID     | NOT NULL | VARCHAR2(6)  |
| FNAME      | -        | VARCHAR2(30) |
| LNAME      | -        | VARCHAR2(30) |
| CITY       | -        | VARCHAR2(15) |
| MOBILENO   | -        | VARCHAR2(10) |
| OCCUPATION | -        | VARCHAR2(20) |
| DOB        | -        | DATE         |

At the bottom of the interface, the following footer information is visible:

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

The screenshot shows the Oracle Live SQL interface. In the SQL Worksheet, Statement 1 runs the same SQL code as the previous screenshot to create the 'customer' table:

```
CREATE TABLE customer
(
    custid VARCHAR(6),
    fname VARCHAR(30),
    lname VARCHAR(30),
    city VARCHAR(15),
    mobileno VARCHAR(10),
    occupation VARCHAR(20),
    dob DATE,
    CONSTRAINT customer_custid_pk PRIMARY KEY(custid)
)
```

The output shows the table creation status:

Table created.

In Statement 2, the following SQL code is run to describe the 'customer' table:

```
desc customer
```

The output shows the table structure:

| Column     | Null?    | Type         |
|------------|----------|--------------|
| CUSTID     | NOT NULL | VARCHAR2(6)  |
| FNAME      | -        | VARCHAR2(30) |
| LNAME      | -        | VARCHAR2(30) |
| CITY       | -        | VARCHAR2(15) |
| MOBILENO   | -        | VARCHAR2(10) |
| OCCUPATION | -        | VARCHAR2(20) |
| DOB        | -        | DATE         |

Oracle Live SQL - My Session | Creating Bank database t... | Bank database queries w... | What are SQL queries for... | GitHub - sonamsinha/db... | +

livesql.oracle.com/apex/f?p=590:6:1280827353261::LEVEL1::

## Live SQL

### My Session

Table created.

Statement 2 desc customer

| Column     | Null?    | Type         |
|------------|----------|--------------|
| CUSTID     | NOT NULL | VARCHAR2(6)  |
| FNAME      | -        | VARCHAR2(30) |
| LNAME      | -        | VARCHAR2(30) |
| CITY       | -        | VARCHAR2(15) |
| MOBILENO   | -        | VARCHAR2(10) |
| OCCUPATION | -        | VARCHAR2(20) |
| DOB        | -        | DATE         |

Download CSV  
7 rows selected.

Statement 3 CREATE TABLE branch

7 rows selected.

Statement 4 desc branch

| Column | Null?    | Type         |
|--------|----------|--------------|
| BID    | NOT NULL | VARCHAR2(6)  |
| BNAME  | -        | VARCHAR2(30) |
| BCITY  | -        | VARCHAR2(30) |

Download CSV  
3 rows selected.

Type here to search

Feedback Help asagar\_be20@thapar.edu Save

Actions Reset Session

38°C ENG 13:55 11-05-2022

Oracle Live SQL - My Session | Creating Bank database t... | Bank database queries w... | What are SQL queries for... | GitHub - sonamsinha/db... | +

livesql.oracle.com/apex/f?p=590:6:1280827353261::LEVEL1::

## Live SQL

### My Session

7 rows selected.

Statement 3 CREATE TABLE branch  
(  
bid VARCHAR(6),  
bname VARCHAR(30),  
bcity VARCHAR(30),  
CONSTRAINT branch\_bid\_pk PRIMARY KEY(bid)  
)

Table created.

Statement 4 desc branch

| Column | Null?    | Type         |
|--------|----------|--------------|
| BID    | NOT NULL | VARCHAR2(6)  |
| BNAME  | -        | VARCHAR2(30) |
| BCITY  | -        | VARCHAR2(30) |

Download CSV  
3 rows selected.

Type here to search

Feedback Help asagar\_be20@thapar.edu Save

Actions Reset Session

38°C ENG 13:55 11-05-2022

Previous Session

Live SQL

Home

SQL Worksheet

My Session

Previous Sessions

Utilization

NLS

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Type here to search

Statement 5

```
CREATE TABLE account
(
    acnumber VARCHAR(6),
    custid VARCHAR(6),
    bid VARCHAR(6),
    opening_balance NUMBER(7),
    aod DATE,
    atype VARCHAR(10),
    astatus VARCHAR(10),
    CONSTRAINT account_acnumber_pk PRIMARY KEY(acnumber),
    CONSTRAINT account_custid_fk FOREIGN KEY(custid) REFERENCES customer(custid),
    CONSTRAINT account_bid_fk FOREIGN KEY(bid) REFERENCES branch(bid)
)
```

Table created.

Statement 6

```
desc account
```

TABLE ACCOUNT

| Column          | Null?    | Type         |
|-----------------|----------|--------------|
| ACNUMBER        | NOT NULL | VARCHAR2(6)  |
| CUSTID          | -        | VARCHAR2(6)  |
| BID             | -        | VARCHAR2(6)  |
| OPENING_BALANCE | -        | NUMBER(7,0)  |
| AOD             | -        | DATE         |
| ATYPE           | -        | VARCHAR2(10) |
| ASTATUS         | -        | VARCHAR2(10) |

39°C 14:38 11-05-2022

Feedback Help Save this Session

This screenshot shows the Oracle Live SQL interface. On the left, there's a sidebar with navigation links like Home, SQL Worksheet, My Session, Previous Sessions, Utilization, NLS, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. Below the sidebar is a search bar. The main area has a breadcrumb path: My Session \ Previous Sessions \ Previous Session. It displays two statements. Statement 5 creates the 'account' table with columns: acnumber, custid, bid, opening\_balance, aod, atype, and astatus, along with primary key and foreign key constraints. Statement 6 describes the 'account' table, showing its columns and data types. A table titled 'TABLE ACCOUNT' is displayed with the following schema:

| Column          | Null?    | Type         |
|-----------------|----------|--------------|
| ACNUMBER        | NOT NULL | VARCHAR2(6)  |
| CUSTID          | -        | VARCHAR2(6)  |
| BID             | -        | VARCHAR2(6)  |
| OPENING_BALANCE | -        | NUMBER(7,0)  |
| AOD             | -        | DATE         |
| ATYPE           | -        | VARCHAR2(10) |
| ASTATUS         | -        | VARCHAR2(10) |

The status bar at the bottom shows the date and time: 11-05-2022, 14:38, and system information: 39°C, ENG.

Previous Session

Live SQL

Home

SQL Worksheet

My Session

Previous Sessions

Utilization

NLS

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Type here to search

Statement 6

```
desc account
```

TABLE ACCOUNT

| Column          | Null?    | Type         |
|-----------------|----------|--------------|
| ACNUMBER        | NOT NULL | VARCHAR2(6)  |
| CUSTID          | -        | VARCHAR2(6)  |
| BID             | -        | VARCHAR2(6)  |
| OPENING_BALANCE | -        | NUMBER(7,0)  |
| AOD             | -        | DATE         |
| ATYPE           | -        | VARCHAR2(10) |
| ASTATUS         | -        | VARCHAR2(10) |

7 rows selected.

Statement 7

```
CREATE TABLE loan
(
    custid VARCHAR(6),
    bid VARCHAR(6)
)
```

39°C 14:41 11-05-2022

Feedback Help Save this Session

This screenshot shows the Oracle Live SQL interface, continuing from the previous one. The main area displays Statement 6, which describes the 'account' table, showing its columns and data types. A table titled 'TABLE ACCOUNT' is displayed with the following schema:

| Column          | Null?    | Type         |
|-----------------|----------|--------------|
| ACNUMBER        | NOT NULL | VARCHAR2(6)  |
| CUSTID          | -        | VARCHAR2(6)  |
| BID             | -        | VARCHAR2(6)  |
| OPENING_BALANCE | -        | NUMBER(7,0)  |
| AOD             | -        | DATE         |
| ATYPE           | -        | VARCHAR2(10) |
| ASTATUS         | -        | VARCHAR2(10) |

The status bar at the bottom shows the date and time: 11-05-2022, 14:41, and system information: 39°C.

Previous Session

Live SQL

Home

SQL Worksheet

My Session

Previous Sessions

Utilization

NLS

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Type here to search

CREATE TABLE loan  
(  
    custid VARCHAR(6),  
    bid VARCHAR(6),  
    loan\_amount INT,  
    CONSTRAINT loan\_customer\_custid\_bid\_pk PRIMARY KEY(custid,bid),  
    CONSTRAINT loan\_custid\_fk FOREIGN KEY(custid) REFERENCES customer(custid),  
    CONSTRAINT loan\_bid\_fk FOREIGN KEY(bid) REFERENCES branch(bid)  
)

Table created.

Statement 8

desc loan

TABLE LOAN

| Column      | Null?    | Type        |
|-------------|----------|-------------|
| CUSTID      | NOT NULL | VARCHAR2(6) |
| BID         | NOT NULL | VARCHAR2(6) |
| LOAN_AMOUNT | -        | NUMBER      |

3 rows selected.

Statement 9

INSERT INTO customer VALUES('C1', 'Ramesh', 'Sharma', 'Delhi', '9543198345', 'Service', DATE '1976-12-06')

1 row(s) inserted.

Statement 10

INSERT INTO customer VALUES('C2', 'Avinash', 'Minha', 'Delhi', '9876532109', 'Service', DATE '1974-10-16')

Feedback Help Save this Session

Delete Re-Run

asagar\_be20@thapar.edu

39°C 1442 11-05-2022

Previous Session

Live SQL

Home

SQL Worksheet

My Session

Previous Sessions

Utilization

NLS

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Type here to search

desc loan

TABLE LOAN

| Column      | Null?    | Type        |
|-------------|----------|-------------|
| CUSTID      | NOT NULL | VARCHAR2(6) |
| BID         | NOT NULL | VARCHAR2(6) |
| LOAN_AMOUNT | -        | NUMBER      |

3 rows selected.

Statement 9

INSERT INTO customer VALUES('C1', 'Ramesh', 'Sharma', 'Delhi', '9543198345', 'Service', DATE '1976-12-06')

1 row(s) inserted.

Statement 10

INSERT INTO customer VALUES('C2', 'Avinash', 'Minha', 'Delhi', '9876532109', 'Service', DATE '1974-10-16')

Feedback Help Save this Session

Delete Re-Run

asagar\_be20@thapar.edu

39°C 1442 11-05-2022

Previous Session

Live SQL

Home SQL Worksheet My Session Previous Sessions Utilization NLS Schema Quick SQL My Scripts My Tutorials Code Library Type here to search

My Session \ Previous Sessions \ Previous Session

Feedback Help Save this Session Delete Re-Run

Statement 12: `INSERT INTO customer VALUES('C3','Rahul','Rastogi','Delhi','9765178901','Student',DATE'1981-09-26')`  
1 row(s) inserted.

Statement 13: `INSERT INTO customer VALUES('C4','Parul','Gandhi','Delhi','9876532109','Housewife',DATE'1976-11-03')`  
1 row(s) inserted.

Statement 14: `INSERT INTO customer VALUES('C5','Naveen','Aedekar','Mumbai','8976523190','Service',DATE'1976-09-19')`  
1 row(s) inserted.

Statement 15: `INSERT INTO customer VALUES('C6','Chitresh','Barwe','Mumbai','7651298321','Student',DATE'1992-11-06')`

39°C 1442 11-05-2022

Previous Session

Live SQL

Home SQL Worksheet My Session Previous Sessions Utilization NLS Schema Quick SQL My Scripts My Tutorials Code Library Type here to search

My Session \ Previous Sessions \ Previous Session

Feedback Help Save this Session Delete Re-Run

Statement 19: `select * from customer`

| CUSTID | FNAME    | LTNAME  | CITY    | MOBILENO   | OCCUPATION | DOB       |
|--------|----------|---------|---------|------------|------------|-----------|
| C1     | Ramesh   | Sharma  | Delhi   | 9543198345 | Service    | 06-DEC-76 |
| C2     | Avinash  | Minha   | Delhi   | 9876532109 | Service    | 16-OCT-74 |
| C3     | Rahul    | Rastogi | Delhi   | 9765178901 | Student    | 26-SEP-81 |
| C4     | Parul    | Gandhi  | Delhi   | 9876532109 | Housewife  | 03-NOV-76 |
| C5     | Naveen   | Aedekar | Mumbai  | 8976523190 | Service    | 19-SEP-76 |
| C6     | Chitresh | Barwe   | Mumbai  | 7651298321 | Student    | 06-NOV-92 |
| C7     | Amit     | Borkar  | Mumbai  | 9875189761 | Student    | 06-SEP-81 |
| C8     | Nisha    | Damle   | Mumbai  | 7954198761 | Service    | 03-DEC-75 |
| C9     | Abhishek | Dutta   | Kolkata | 9856198761 | Service    | 22-MAY-73 |
| C10    | Shankar  | Nair    | Chennai | 8765489076 | Service    | 12-JUL-76 |

10 rows selected.

Statement 20: `INSERT INTO branch VALUES('B1','Asaf ali road','Delhi')`

39°C 1442 11-05-2022

Worksheet | Query Builder

```

CREATE TABLE loan
(
    custid VARCHAR(6),
    bid VARCHAR(6),
    loan_amount INT,
    CONSTRAINT loan_customer_custid_pk PRIMARY KEY(custid,bid),
    CONSTRAINT loan_custid_fk FOREIGN KEY(custid) REFERENCES customer(custid),
    CONSTRAINT loan_bid_fk FOREIGN KEY(bid) REFERENCES branch(bid)
);

INSERT INTO loan VALUES('C1','B1',1289000)
;

INSERT INTO loan VALUES('C2','B3',156890)
;
INSERT INTO loan VALUES('C3','B1',255700)
;
INSERT INTO loan VALUES('C4','B1',2269000)
;
INSERT INTO loan VALUES('C5','B5',1188000)
;
INSERT INTO loan VALUES('C6','B6',829600)
;
INSERT INTO loan VALUES('C7','B7',659000)
;
select * from loan;
desc loan;

```

Script Output | Query Result 1 | All Rows Fetched: 5 in 0.01 seconds

| CUSTID | BID | LOAN_AMOUNT |
|--------|-----|-------------|
| 1 C1   | B1  | 1289000     |
| 2 C2   | B3  | 156890      |
| 3 C5   | B5  | 1188000     |
| 4 C6   | B6  | 829600      |
| 5 C7   | B7  | 659000      |

Previous Session | + | Live SQL

Feedback | Help | asagar\_be20@thapar.edu

Home | SQL Worksheet | My Session | Previous Sessions | Previous Session | 1 rows inserted.

Statement 22 | `INSERT INTO branch VALUES('B3','Delhi cantt','Delhi')`  
1 row(s) inserted.

Statement 23 | `INSERT INTO branch VALUES('B4','Jasola','Delhi')`  
1 row(s) inserted.

Statement 24 | `INSERT INTO branch VALUES('B5','Mahim','Mumbai')`  
1 row(s) inserted.

Statement 25 |

Type here to search | 39°C | 14:43 | 11-05-2022

Previous Session | + | Live SQL

Feedback | Help | asagar\_be20@thapar.edu

Home | SQL Worksheet | My Session | Previous Sessions | Previous Session

Statement 29 | `select * from branch`

| BID | BNAME                 | BCITY   |
|-----|-----------------------|---------|
| B1  | Asaf ali road         | Delhi   |
| B2  | New delhi main branch | Delhi   |
| B3  | Delhi cantt           | Delhi   |
| B4  | Jasola                | Delhi   |
| B5  | Mahim                 | Mumbai  |
| B6  | Vile parle            | Mumbai  |
| B7  | Mandvi                | Mumbai  |
| B8  | Jadavpur              | Kolkata |
| B9  | Kodambakkam           | Chennai |

9 rows selected.

Statement 30 | `INSERT INTO account VALUES('A1','C1','B1',1000,DATE'2012-12-15','Saving','Active')`

Type here to search | 39°C | 14:44 | 11-05-2022

The screenshot shows the Oracle Live SQL interface. On the left, there's a sidebar with options like Home, SQL Worksheet, My Session (Previous Sessions, Utilization, NLS), Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "My Session \ Previous Sessions \ Previous Session". It displays three rows of data from statements 37, 38, and 39. Each row shows an INSERT INTO account statement and a message indicating "1 row(s) inserted." The statements insert data into columns A1 through B7.

This screenshot shows the same Oracle Live SQL interface. In the main area, a SELECT \* FROM account statement is run (Statement 40). The results are displayed in a grid with 10 rows. The columns are ACNUMBER, CUSTID, BID, OPENING\_BALANCE, AOD, ATYPE, and ASTATUS. The data includes various account numbers, customer IDs, branch IDs, opening balances, dates, types, and statuses.

## NORMALISATION :

### #1) 1NF (First Normal Form)

Following is how our Customer table in first normal form (1NF):

| Custid | lastName | firstName | Occupation | CustCity |
|--------|----------|-----------|------------|----------|
| 1001   | Andrews  | Jack      | Accounts   | New York |
| 1002   | Schwartz | Mike      | Technology | New York |
| 1009   | Beker    | Harry     | HR         | Berlin   |
| 1007   | Harvey   | Parker    | Admin      | London   |
| 1007   | Harvey   | Parker    | HR         | London   |

## #2) 2NF (Second Normal Form)

By definition, an entity that is 1NF and one of its attributes is defined as the primary key and the remaining attributes are dependent on the primary key.

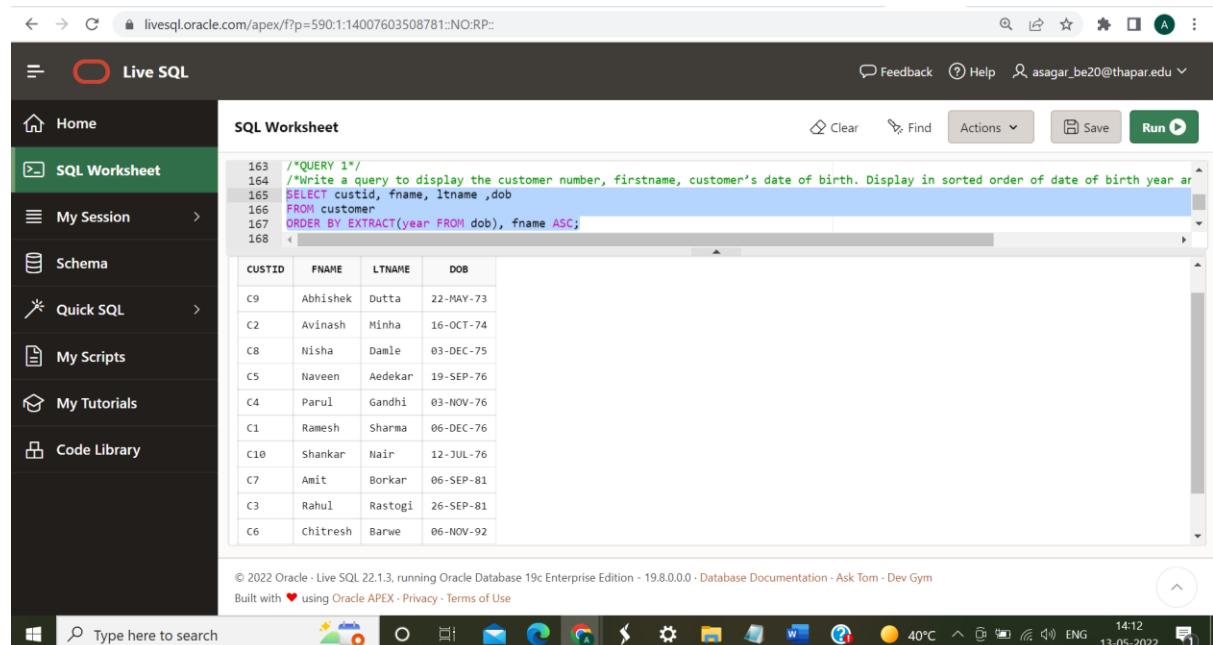
**Customer Table:**

| Custid | lastName | firstName |
|--------|----------|-----------|
| 1001   | Andrews  | Jack      |
| 1002   | Schwartz | Mike      |
| 1009   | Beker    | Harry     |
| 1007   | Harvey   | Parker    |
| 1007   | Harvey   | Parker    |

**Branch Table:**

| bid | bname      | bcity    |
|-----|------------|----------|
| 1   | Delhi      | New York |
| 2   | Chandigarh | New York |
| 3   | Jaipur     | Surart   |
| 4   | Admin      | London   |

## Queries for retrieving information from tables:



The screenshot shows the Oracle Live SQL interface. The left sidebar has a navigation menu with options like Home, SQL Worksheet (which is selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet". It contains a code editor with the following SQL query:

```
163 /*QUERY 1*/
164 /*Write a query to display the customer number, firstname, customer's date of birth. Display in sorted order of date of birth year ar
165 SELECT custid, fname, lname ,dob
166 FROM customer
167 ORDER BY EXTRACT(year FROM dob), fname ASC;
168 */
```

Below the code editor is a results grid showing customer data:

| CUSTID | FNAME    | LNAME   | DOB       |
|--------|----------|---------|-----------|
| C9     | Abhishek | Dutta   | 22-MAY-73 |
| C2     | Avinash  | Minha   | 16-OCT-74 |
| C8     | Nisha    | Damle   | 03-DEC-75 |
| C5     | Naveen   | Adekar  | 19-SEP-76 |
| C4     | Parul    | Gandhi  | 03-NOV-76 |
| C1     | Ramesh   | Sharma  | 06-DEC-76 |
| C10    | Shankar  | Nair    | 12-JUL-76 |
| C7     | Amit     | Borkar  | 06-SEP-81 |
| C3     | Rahul    | Rastogi | 26-SEP-81 |
| C6     | Chitresh | Barwe   | 06-NOV-92 |

At the bottom of the interface, there is a status bar with system information and a search bar.

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO::P::

### Live SQL

**SQL Worksheet**

```
170 /*Write a query to display account number, customer's number, customer's firstname,lastname,account opening date.*/
171 SELECT account.acnnumber, customer.custid, customer.fname, customer.lname, account.aod
172 FROM account
173 INNER JOIN customer
174 ON account.custid = customer.custid;
175
```

| ACNUMBER | CUSTID | FNAME    | LNAME   | AOD       |
|----------|--------|----------|---------|-----------|
| A1       | C1     | Ramesh   | Sharma  | 15-DEC-12 |
| A8       | C1     | Ramesh   | Sharma  | 09-NOV-09 |
| A2       | C2     | Avinash  | Minha   | 12-JUN-12 |
| A4       | C2     | Avinash  | Minha   | 27-JAN-13 |
| A3       | C3     | Rahul    | Rastogi | 17-MAY-12 |
| A9       | C3     | Rahul    | Rastogi | 30-NOV-08 |
| A10      | C4     | Parul    | Gandhi  | 01-MAR-13 |
| A5       | C6     | Chitresh | Barwe   | 17-DEC-12 |
| A6       | C7     | Amit     | Borkar  | 12-AUG-10 |
| A7       | C7     | Amit     | Borkar  | 02-OCT-12 |

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Type here to search

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO::P::

### Live SQL

**SQL Worksheet**

```
170 /*Write a query to display account number, customer's number, customer's firstname,lastname,account opening date.*/
171 SELECT account.acnnumber, customer.custid, customer.fname, customer.lname, account.aod
172 FROM account
173 INNER JOIN customer
174 ON account.custid = customer.custid;
175
176 /*QUERY 3*/
177 /*Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.*/
178
179 SELECT count(custid) Cust_Count
180 FROM customer
181 WHERE city='Delhi'
182
183
```

| CUST_COUNT |
|------------|
| 4          |

Download CSV

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO::P::

### Live SQL

**SQL Worksheet**

```
199 /*QUERY 5*/
200 /*Write a query to display the female customers firstname, city and account number who are not into business, service or studies.*/
201 SELECT DISTINCT customer.fname, customer.city, account.acnnumber
202 FROM account, customer
203 WHERE account.custid = customer.custid
204 AND NOT(occupation='business' or occupation='service' or occupation='student');
205
```

| Parul    | Delhi  | A10 |
|----------|--------|-----|
| Amit     | Mumbai | A6  |
| Ramesh   | Delhi  | A8  |
| Avinash  | Delhi  | A2  |
| Avinash  | Delhi  | A4  |
| Amit     | Mumbai | A7  |
| Rahul    | Delhi  | A3  |
| Chitresh | Mumbai | A5  |
| Ramesh   | Delhi  | A1  |
| Rahul    | Delhi  | A9  |

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO:RP::

Live SQL

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

```
202 FROM account, customer
203 WHERE account.custid = customer.custid
204 AND NOT(occupation='business' or occupation='service' or occupation='student');
205
206 /*QUERY 6*/
207 /*Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.*/
208 SELECT bcity, count(*) AS Count_Branch
209 FROM branch
210 GROUP BY bcity;
211
212
213
```

| BCITY   | COUNT_BRANCH |
|---------|--------------|
| Chennai | 1            |
| Kolkata | 1            |
| Delhi   | 4            |
| Mumbai  | 3            |

Download CSV  
4 rows selected.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Type here to search

Clear Find Actions Save Run

Feedback Help asagar\_be20@thapar.edu

41°C ENG 14:39 13-05-2022

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO:RP::

Live SQL

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

```
213 /*QUERY 7*/
214 /*Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active.*/
215 SELECT account.acnumber, customer.fname, customer.lname
216 FROM account, customer
217 WHERE account.custid = customer.custid
218 AND astatus = 'Active';
219
220
```

| ACNUMBER | FNAME    | LNAME   |
|----------|----------|---------|
| A1       | Ramesh   | Sharma  |
| A2       | Avinash  | Minha   |
| A4       | Avinash  | Minha   |
| A3       | Rahul    | Rastogi |
| A10      | Parul    | Gandhi  |
| A5       | Chitresh | Barwe   |
| A7       | Amit     | Borkar  |

Download CSV

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Type here to search

Clear Find Actions Save Run

Feedback Help asagar\_be20@thapar.edu

41°C ENG 14:40 13-05-2022

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO::P::

Live SQL

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

```
226 /*QUERY 9*/
227 /*Write a query to display customer number, customer name, account number where the account status is terminated.*/
228 SELECT customer.custid, customer.fname, account.acnumber
229 FROM account, customer
230 WHERE account.custid = customer.custid
231 AND astatus = 'Terminated';
232
```

| CUSTID | FNAME  | ACNUMBER |
|--------|--------|----------|
| C1     | Ramesh | A8       |
| C3     | Rahul  | A9       |

Download CSV  
2 rows selected.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Type here to search

Feedback Help asagar\_be20@thapar.edu

Actions Save Run

Clear Find

41°C ENG 14:43 13-05-2022

## PL/SQL transactions:

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO::P::

Live SQL

Home SQL Worksheet My Session Schema Quick SQL My Scripts My Tutorials Code Library

SQL Worksheet

```
265 );
266 );
267 create table acct_master(acct_no number(5) primary key,
268 acct_name varchar2(10),
269 balance number(10));
270 /*INSERTING data in acct_mstr*/
271 insert into acct_master values(1, 'aaa', 1000);
272 insert into acct_master values(2, 'bbb', 100);
273 insert into acct_master values(3, 'ccc', 1100);
274 insert into acct_master values(4, 'ddd', 700);
275 insert into acct_master values(5, 'eee', 1700);
276 -- DECLARING VARIABLES
277 declare
```

1 row(s) inserted.  
1 row(s) inserted.  
1 row(s) inserted.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Banking Databas...docx Banking Databas...docx

Type here to search

Show all

Feedback Help asagar\_be20@thapar.edu

Actions Save Run

Clear Find

41°C ENG 14:59 13-05-2022

livesql.oracle.com/apex/f?p=590:1:14007603508781::NO::

```

277 /*PL/SQL QUERIES */
278 -- DECLARING VARIABLES
279 DECLARE
280   xacct_no number(5);
281
282   -- here, minimum balance is set to 1000;
283   xmin_bal number(5):=1000;
284   xbalance number(5);
285
286   --
287
288   1   aaa    1000
289   2   bbb    100
290   3   ccc    1100
291   4   ddd    700
292   5   eee    1700
293
294 Download CSV
295 5 rows selected.

```

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Banking Databas...docx Banking Databas...docx Show all

livesql.oracle.com/apex/f?p=590:1:121217656254::NO::

```

37   -- remaining amount
38   xbalance:=xbalance-100;
39   dbms_output.put_line('Rs 100 is deducted and current balance is'||xbalance);
40
41   -- if condition is false
42   ELSE
43   dbms_output.put_line('Current balance is'||xbalance);
44
45   --ENDING IF
46   END IF;
47
48   -- ENDING OF BEGIN
49   END;
50

```

Statement processed.  
Rs 100 is deducted and current balance is 0

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

Type here to search

**Join query :**

livesql.oracle.com/apex/f?p=590:1:9522305254401::NO:::

```

194 time:=2;
195 amount:=ci(principle,rate,time);
196 dbms_output.put_line('the interest on loan amount is :');
197 dbms_output.put_line(amount-principle);
198
199
200 /* Query 10
201 Write a query that returns customer name, customer_id,loan_amount from the customers and accounts table.
202 */
203 SELECT fname||' '|| lname AS customer_name , customer.custid, loan.loan_amount
204 FROM customer
205 JOIN loan ON
206 customer.custid= loan.custid
207 WHERE loan_amount > 829000;
208

```

| CUSTOMER_NAME  | CUSTID | LOAN_AMOUNT |
|----------------|--------|-------------|
| Ramesh Sharma  | C1     | 1289000     |
| Naveen Aedekar | C5     | 1188000     |

[Download CSV](#)  
2 rows selected.

## PL/SQL Queries for exceptions :

### Query 1:

livesql.oracle.com/apex/f?p=590:1:15281018721155::NO:::

```

160 select * from account
161 ;
162 Declare
163 temp varchar(20);
164 begin
165 select custid into temp from customer where fname='ABC';
166 exception
167 WHEN no_data_found THEN
168 DBMS_OUTPUT.PUT_LINE('EXCEPTION FOUND ....ERROR-->NO DATA FOUND');
169 END;
170 Declare
171 temp varchar(20);
172 begin
173 select custid into temp from customer where fname='Parul';
174 dbms_output.put_line(temp);
175 exception
176 WHEN too_many_rows THEN
177 DBMS_OUTPUT.PUT_LINE('EXCEPTION FOUND ....ERROR-->Too many rows');
178 end;

```

Statement processed.  
EXCEPTION FOUND ....ERROR-->NO DATA FOUND

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation · Ask Tom · Dev Gym  
Built with ❤️ using Oracle APEX · Privacy · Terms of Use

### Query 2:

```

164 ;
163 Declare
164 temp varchar(20);
165 begin
166 select custid into temp from customer where fname='ABC';
167 exception
168 WHEN no_data_found THEN
169 DBMS_OUTPUT.PUT_LINE('EXCEPTION FOUND ....ERROR-->NO DATA FOUND');
170 END;
171 Declare
172 temp varchar(20);
173 begin
174 select custid into temp from customer;
175 dbms_output.put_line(temp);
176 exception
177 WHEN too_many_rows THEN
178 DBMS_OUTPUT.PUT_LINE('EXCEPTION FOUND ....ERROR-->Too many rows');
179 END;
180

```

Statement processed.  
EXCEPTION FOUND ....ERROR-->Too many rows

## Triggers :

### Trigger 1:

/\*TRIGGER 1 :To Create a trigger for the customer table, which makes the entry in custNAME column in uppercase.\*/

```

CREATE OR REPLACE TRIGGER upper_trigger
BEFORE INSERT OR UPDATE OF fname ON customer
FOR EACH ROW
BEGIN
:new.fname := UPPER (:new.fname);
END;

```

Script Output | Task completed in 0.123 seconds

10 rows selected.

Trigger UPPER\_TRIGGER compiled

LINE/COL ERROR

2/17 FLS-00103: Encountered the symbol " " when expecting one of the following: := . { ; % ; indicator Errors: check compiler log

Trigger UPPER\_TRIGGER compiled

### Trigger 2:

/\*To Create a trigger on the customer table, which shows the old values and new value of fname after any updation on fname of customer table.\*/

```

--Q2
CREATE OR REPLACE TRIGGER CUST_UPDATE
AFTER UPDATE OF FNAME ON customer FOR EACH ROW
BEGIN
DBMS_OUTPUT.PUT_LINE('OLD NAME:' ||:OLD.FNAME);
DBMS_OUTPUT.PUT_LINE('NEW NAME:' ||:NEW.FNAME);
END;

```

Script Output | Task completed in 0.097 seconds

Trigger CUST\_UPDATE compiled

Compiler - Log  
Messages | Statements | Compiler | Logging Page

### **Trigger 3:**

/\*To Create a trigger on customer table, which store the custid and fname in table auditor for each operation i.e. Insert, Update and Delete.\*/

```
--q3
--To Create a trigger on customer table, which store the custid and fname in table auditor for each operation i.e. Insert, Update and Delete.

--Solution:
create table AUDITOR(custnewid varchar(35));
CREATE OR REPLACE TRIGGER cust_audit
AFTER INSERT OR UPDATE OR DELETE ON customer
FOR EACH ROW
BEGIN
IF INSERTING THEN
    INSERT INTO AUDITOR VALUES(:NEW.custid,'INSERT');
ELSIF UPDATING THEN
    INSERT INTO AUDITOR VALUES(:NEW.custid,'UPDATE');
ELSIF DELETING THEN
    INSERT INTO AUDITOR VALUES(:OLD.custid,'DELETE');
END IF;
END;

select * from AUDITOR;
```

## Cursors :

```
/* Q1: Write %type and %rowtype to display details from branch table */
```

livesql.oracle.com/apex/f?p=5901:17651586593566:NO::P::

## Live SQL

Feedback Help asagar\_be20@thapar.edu

### SQL Worksheet

Clear Find Actions Save Run

```
193 OPEN c_emp;
194 LOOP
195   FETCH c_emp into c_id,c_name,c_city;
196   EXIT WHEN c_emp%notfound;
197   dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_city);
198 END LOOP;
199 CLOSE c_emp;
200 END;
```

Statement processed.

B1 Asaf ali road Delhi  
B2 New delhi main branch Delhi  
B3 Delhi cantt Delhi  
B4 Jasola Delhi  
B5 Mahim Mumbai  
B6 Vile parle Mumbai  
B7 Mandvi Mumbai  
B8 Jadavpur Kolkata  
B9 Kadambakkam chennai

**/\*Q2:PL/SQL code to calculate the total loan amount of first n records of loan table. The value of n is passed to cursor as parameter\*/**

The screenshot shows the Oracle Live SQL interface. The left sidebar includes links for Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following PL/SQL code:

```
207 WHERE loan_amount > 829000;
208 /*PL/SQL code to calculate the total loan amount of first n records of loan table. The value of n is passed to cursor as parameter*/
209 select * from loan order by loan_amount desc;
210 Declare
211 e number(30);
212 t number(30):=0;
213 n number :=3;
214 cursor c1(n number) is select loan_amount from loan;
215 begin
216 open c1(3);
217 loop
218 fetch c1 into e;
219 exit when c1%notfound;
220 t:=t+e;
221 end loop;
222 dbms_output.put_line('Total loan amount of first n customers is -->');
223 dbms_output.put_line(t);
224 close c1;
225 end;
```

The output window below shows the results:

```
Statement processed.
Total loan amount of first n customers is -->
4121890
```

At the bottom, the footer reads: © 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym. It also mentions "Built with ❤ using Oracle APEX - Privacy - Terms of Use".

## FUNCTIONS :

### /\*Function which returns total number of accounts in a branch\*/

The screenshot shows the Oracle Live SQL interface. In the SQL Worksheet, the following PL/SQL code is executed:

```
199   CLOSE c_emp;
200
201 /*Function which returns total number of accounts in a branch*/
202 create or replace function totalAccounts(bid in varchar)
203 return number
204 is
205   total number(2):=0;
206 begin
207   select count(*) into total
208   from account
209   where bid=bid;
210   return total;
211   dbms_output.put_line('total no. of accounts in the branch are :');
212   dbms_output.put_line(total);
213 end;
214
215 declare
216   c number(2);
217 begin
218   c:=totalAccounts('B1');
219   dbms_output.put_line('total no. of accounts in the branch are : '|| c);
220 end;
221 dbms_output.put_line('');
222 dbms_output.put_line(total);
223
```

The output window shows the message "Function created.".

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy - Terms of Use

The screenshot shows the Oracle Live SQL interface. In the SQL Worksheet, the same PL/SQL code is run again. The output window shows the message "Statement processed." followed by the output "total no. of accounts in the branch are : 10".

## PROCEDURE :

### /\*Q1 Procedure which returns the opening balance in account with the given acnumber

### Procedure with exception handling\*/

Oracle Live SQL - SQL Worksheet

Live SQL

SQL Worksheet

```
--  
221 dbms_output.put_line(':');  
222 dbms_output.put_line(total);  
223  
--PROCEDURE  
224 /*Procedure which returns the opening_balance in account with the given acnnumber  
225 Procedure with exception handling*/  
226  
227 create or replace procedure acc_details(a_id in varchar)  
228 is  
229 acc number(2);  
230 begin  
231 select opening_balance into acc from account where acnnumber=a_id;  
232 exception  
233 when no_data_found then  
234 dbms_output.put_line('Sorry no such account exist !!');  
235 end;  
236  
237
```

Procedure created.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy · Terms of Use

Type here to search

## /\*Q2 PROCEDURE TO CALCULATE INTEREST ON LOAN AMOUNT\*/

livesql.oracle.com/apex/f?p=590:1:9522305254401::NO::

Live SQL

SQL Worksheet

```
4/9 /*PROCEDURE TO CALCULATE INTEREST ON LOAN AMOUNT*/  
180 DECLARE  
181 principle number;  
182 rate number;  
183 time number;  
184 amount number;  
185 interest number;  
186 function ci (p in number, r in number, t in number) return number is  
187 BEGIN  
188 interest:=p * power(1+(r/100),t);  
189 return(interest);  
190 END;  
191 BEGIN  
192 principle := 9678000;  
193 rate:=5;  
194 time:=2;  
195 amount:=ci(principle,rate,time);  
196 dbms_output.put_line('the interest on loan amount is :');  
197 dbms_output.put_line(amount-principle);  
198 END;
```

Statement processed.  
the interest on loan amount is :  
991995

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤ using Oracle APEX - Privacy · Terms of Use

```
CREATE TABLE customer
(
    custid VARCHAR(6),
    fname VARCHAR(30),
    lname VARCHAR(30),
    city VARCHAR(15),
    mobileno VARCHAR(10),
    occupation VARCHAR(20),
    dob DATE,
    CONSTRAINT customer_custid_pk PRIMARY KEY(custid)
)
;

desc customer
```

```
CREATE TABLE branch
(
    bid VARCHAR(6),
    bname VARCHAR(30),
    bcity VARCHAR(30),
    CONSTRAINT branch_bid_pk PRIMARY KEY(bid)
)
;
```

```
desc branch
```

```
CREATE TABLE account
(
    acnumber VARCHAR(6),
    custid VARCHAR(6),
    bid VARCHAR(6),
    opening_balance NUMBER(7),
    aod DATE,
    atype VARCHAR(10),
    astatus VARCHAR(10),
    CONSTRAINT account_acnumber_pk PRIMARY KEY(acnumber),
    CONSTRAINT account_custid_fk FOREIGN KEY(custid) REFERENCES customer(custid),
    CONSTRAINT account_bid_fk FOREIGN KEY(bid) REFERENCES branch(bid)
)
;

desc account
```

```
CREATE TABLE loan
(
    custid VARCHAR(6),
    bid VARCHAR(6),
    loan_amount INT,
    CONSTRAINT loan_customer_custid_bid_pk PRIMARY KEY(custid,bid),
    CONSTRAINT loan_custid_fk FOREIGN KEY(custid) REFERENCES customer(custid),
    CONSTRAINT loan_bid_fk FOREIGN KEY(bid) REFERENCES branch(bid)
```

```

        )
;

desc loan
INSERT INTO loan VALUES('C1','B1',1289000)
;

INSERT INTO loan VALUES('C2','B2',156890)
;
INSERT INTO loan VALUES('C3','B3',255700)

INSERT INTO loan VALUES('C4','B4',2289000)
;
INSERT INTO loan VALUES('C5','B5',1188000)
;
INSERT INTO loan VALUES('C6','B6',829000)
;
INSERT INTO loan VALUES('C7','B7',659000)
;
select * from loan;

INSERT INTO customer
VALUES('C1','Ramesh','Sharma','Delhi','9543198345','Service',DATE'1976-12-06')
;

INSERT INTO customer
VALUES('C2','Avinash','Minha','Delhi','9876532109','Service',DATE'1974-10-16')
;

INSERT INTO customer
VALUES('C3','Rahul','Rastogi','Delhi','9765178901','Student',DATE'1981-09-26')
;

INSERT INTO customer
VALUES('C4','Parul','Gandhi','Delhi','9876532109','Housewife',DATE'1976-11-03')
;

INSERT INTO customer
VALUES('C5','Naveen','Aedekar','Mumbai','8976523190','Service',DATE'1976-09-19')
;

INSERT INTO customer
VALUES('C6','Chitresh','Barwe','Mumbai','7651298321','Student',DATE'1992-11-06')
;

INSERT INTO customer
VALUES('C7','Amit','Borkar','Mumbai','9875189761','Student',DATE'1981-09-06')
;

INSERT INTO customer
VALUES('C8','Nisha','Damle','Mumbai','7954198761','Service',DATE'1975-12-03')
;

INSERT INTO customer
VALUES('C9','Abhishek','Dutta','Kolkata','9856198761','Service',DATE'1973-05-22')
;

```

```
INSERT INTO customer
VALUES('C10','Shankar', 'Nair', 'Chennai', '8765489076', 'Service', DATE'1976-07-12')
;

select * from customer
;

INSERT INTO branch VALUES('B1', 'Asaf ali road', 'Delhi')
;

INSERT INTO branch VALUES('B2', 'New delhi main branch', 'Delhi')
;

INSERT INTO branch VALUES('B3', 'Delhi cantt', 'Delhi')
;

INSERT INTO branch VALUES('B4', 'Jasola', 'Delhi')
;

INSERT INTO branch VALUES('B5', 'Mahim', 'Mumbai')
;

INSERT INTO branch VALUES('B6', 'Vile parle', 'Mumbai')
;

INSERT INTO branch VALUES('B7', 'Mandvi', 'Mumbai')
;

INSERT INTO branch VALUES('B8', 'Jadavpur', 'Kolkata')
;

INSERT INTO branch VALUES('B9', 'Kodambakkam', 'Chennai')
;

select * from branch
;

INSERT INTO account VALUES('A1', 'C1', 'B1', 1000, DATE'2012-12-15', 'Saving', 'Active')
;

INSERT INTO account VALUES('A2', 'C2', 'B1', 1000, DATE'2012-06-12', 'Saving', 'Active')
;

INSERT INTO account VALUES('A3', 'C3', 'B2', 1000, DATE'2012-05-17', 'Saving', 'Active')
;

INSERT INTO account VALUES('A4', 'C2', 'B5', 1000, DATE'2013-01-27', 'Saving', 'Active')
;

INSERT INTO account VALUES('A5', 'C6', 'B6', 1000, DATE'2012-12-17', 'Saving', 'Active')
;

INSERT INTO account VALUES('A6', 'C7', 'B7', 1000, DATE'2010-08-12', 'Saving', 'Suspended')
;

INSERT INTO account VALUES('A7', 'C7', 'B1', 1000, DATE'2012-10-02', 'Saving', 'Active')
```

```

;

INSERT INTO account VALUES('A8', 'C1', 'B3', 1000, DATE'2009-11-09', 'Saving', 'Terminated')
;

INSERT INTO account VALUES('A9', 'C3', 'B7', 1000, DATE'2008-11-30', 'Saving', 'Terminated')
;

INSERT INTO account VALUES('A10', 'C4', 'B2', 1000, DATE'2013-03-01', 'Saving', 'Active')
;

select * from account
;
/*QUERY 1*/
/*Write a query to display the customer number, firstname, customer's date of birth. Display in sorted order of date of birth year and within that sort by firstname.*/
SELECT custid, fname, lname, dob
FROM customer
ORDER BY EXTRACT(year FROM dob), fname ASC;

/*QUERY 2*/
/*Write a query to display account number, customer's number, customer's firstname, lastname, account opening date.*/
SELECT account.acnumber, customer.custid, customer.fname, customer.lname,
account.aod
FROM account
INNER JOIN customer
ON account.custid = customer.custid;

/*QUERY 3*/
/*Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.*/
SELECT
    (SELECT COUNT(city)
     FROM customer
     WHERE city='Delhi')
AS Cust_Count;
/*QUERY 4*/
/*Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month.*/
SELECT account.custid, customer.fname, account.acnumber
FROM account, customer
WHERE account.custid = customer.custid
AND day(aod) > 15;

/*QUERY 5*/
/*Write a query to display the female customers firstname, city and account number who are not into business, service or studies.*/
SELECT DISTINCT customer.fname, customer.city, account.acnumber
FROM account, customer
WHERE account.custid = customer.custid
AND NOT(occupation='business' or occupation='service' or occupation='student');

/*QUERY 6*/
/*Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.*/

```

```

SELECT bcity, count(*)
AS Count_Branch
FROM branch
Group By bcity;

/*QUERY 7*/
/*Write a query to display account id, customer's firstname, customer's lastname
for the customer's whose account is Active.*/
SELECT account.acnumber, customer.fname, customer.lname
FROM account, customer
WHERE account.custid = customer.custid
AND astatus = 'Active';

/*QUERY 8*/
/*Write a query to display the customer's number, customer's firstname, branch id
and loan amount for people who have taken loans.*/
SELECT customer.custid, customer.fname, branch.bid, loan.loan_amount
FROM ((loan
INNER JOIN customer ON loan.custid=customer.custid)
INNER JOIN branch ON loan.bid=branch.bid);

/*QUERY 9*/
/*Write a query to display customer number, customer name, account number where the
account status is terminated.*/
SELECT customer.custid, customer.fname, account.acnumber
FROM account, customer
WHERE account.custid = customer.custid
AND astatus = 'Terminated';

/* Query 10
Write a query that returns customer name, customer_id, loan_amount from the
customers and accounts table.
*/
SELECT fname||' '|| lname AS customer_name , customer.custid, loan.loan_amount
FROM customer
JOIN loan  ON
customer.custid= loan.custid
WHERE loan_amount > 829000;

/*PL/SQL TRANSACTIONS*/
acct_master (acct_no number(5) primary key,
            acct_name varchar2(10),
            balance number(10));
/* CREATING table acct_master */
create table acct_master(acct_no number(5) primary key,
                        acct_name varchar2(10),
                        balance number(10));
/*INSERTING data in acct_mstr*/
insert into acct_master values(1, 'aaa', 1000);
insert into acct_master values(2, 'bbb', 100);
insert into acct_master values(3, 'ccc', 1100);
insert into acct_master values(4, 'ddd', 700);
insert into acct_master values(5, 'eee', 1700);
select * from acct_master;
-- DECLARING VARIABLES
DECLARE
xacct_no number(5);

```

```

-- here, minimum balance is set to 1000;
xmin_bal number(5):=1000;
xbalance number(5);

BEGIN

-- taking input from user
xacct_no:= 2;

-- selecting balance of that user INTO "xbalance";
select balance into xbalance
from acct_master
where acct_no=xacct_no;

-- if condition true, updating balance
-- with balance = balance - 100
IF(xbalance < xmin_bal) THEN --condition check
update acct_master
set balance=balance-100
where acct_no=xacct_no;

-- remaining amount

xbalance:=xbalance-100;
dbms_output.put_line('Rs 100 is deducted and current balance is'||xbalance);

-- if condition is false
ELSE
dbms_output.put_line('Current balance is'||xbalance);

--ENDING IF
END IF;

-- ENDING OF BEGIN
END;

#####
/*CURSORS*/
/*IMPLICIT CURSORS */
/*The following program will update the table and increase the salary of each
customer by 500
and use the SQL%ROWCOUNT attribute to determine the number of rows affected */

DECLARE
    total_rows number(2);
BEGIN
    UPDATE customers
    SET salary = salary + 500;
    IF sql%notfound THEN
        dbms_output.put_line('no customers selected');
    ELSIF sql%found THEN
        total_rows := sql%rowcount;
        dbms_output.put_line( total_rows || ' customers selected ');
    END IF;
END;
Select * from customers;
/*EXPLICIT CURSORS*/

```

```

DECLARE
    c_id customers.id%type;
    c_name customers.name%type;
    c_addr customers.address%type;
    CURSOR c_customers is
        SELECT id, name, address FROM customers;
BEGIN
    OPEN c_customers;
    LOOP
        FETCH c_customers into c_id, c_name, c_addr;
        EXIT WHEN c_customers%notfound;
        dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_addr);
    END LOOP;
    CLOSE c_customers;
END;

```

```

#####
/*PL/SQL exception handling queries*/
Declare
temp varchar(20);
begin
select custid into temp from customer where fname='ABC';
exception
WHEN no_data_found THEN
DBMS_OUTPUT.PUT_LINE('EXCEPTION FOUND ....ERROR-->NO DATA FOUND');
END;
Declare
temp varchar(20);
begin
select custid into temp from customer where fname='Parul';
dbms_output.put_line(temp);
exception
WHEN too_many_rows THEN
DBMS_OUTPUT.PUT_LINE('EXCEPTION FOUND ....ERROR-->Too many rows');
END;

#####
/*TRIGGERS*/
/*TRIGGER 1 :To Create a trigger for the customer table, which makes the entry in
custNAME column in uppercase.*/

```

Solution:

```

CREATE OR REPLACE TRIGGER upper_trigger
BEFORE INSERT OR UPDATE OF fname ON customer
FOR EACH ROW
BEGIN
    :new.fname := UPER (:new.fname);
END;

```

```

###
/*Trigger2*/
To Create a trigger on the customer table, which shows the old values and new value
of fname after any updation on fname of customer table.*/
--Q2
CREATE OR REPLACE TRIGGER CUST_UPDATE

```

```

AFTER UPDATE OF FNAME ON customer FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('OLD NAME:' || :OLD.FNAME);
    DBMS_OUTPUT.PUT_LINE('NEW NAME:' || :NEW.FNAME);
END;

to update the total amount of customer everytime he makes a new account*/
create or replace function total_cust(custid in varchar)
return number
is
total number(2) :=0;
begin
select sum(cost) into total from account where
product.product_id=cart_item.product_id and cart_id=cId;
return total;
end;

create or replace trigger before_pay_up
before insert
on
payment
for each row
declare
total number(3);
begin
total :=total_cost(:new.cart_id);
insert into payment
values(:new.payment_id,:new.payment_date,:new.payment_type,:new.customer_id,:new.ca
rt_id,total);
end;

/*Trigger2 on the customer table, which store the mobileno and occupation in table
auditor for each operation i.e. Insert, Update and Delete.

```

Solution:

```

CREATE OR REPLACE TRIGGER CUST_AUDIT
AFTER INSERT OR UPDATE OR DELETE ON customer
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO AUDITOR VALUES(:NEW.mobileno, 'INSERT');
    ELSIF UPDATING THEN
        INSERT INTO AUDITOR VALUES(:NEW.mobileno, 'UPDATE');
    ELSIF DELETING THEN
        INSERT INTO AUDITOR VALUES(:OLD.mobileno, 'DELETE');
    END IF;
END;

```

/\*Trigger 3 --To Create a trigger on customer table, which store the custid and fname in table auditor for each operation i.e. Insert, Update and Delete.

```

--Solution:
create table AUDITOR(custnewid varchar(35));
CREATE OR REPLACE TRIGGER cust_audit
AFTER INSERT OR UPDATE OR DELETE ON customer
FOR EACH ROW
BEGIN
    IF INSERTING THEN

```

```

    INSERT INTO AUDITORR VALUES(:NEW.custid,'INSERT');
ELSIF UPDATING THEN
    INSERT INTO AUDITORR VALUES(:NEW.custid,'UPDATE');
ELSIF DELETING THEN
    INSERT INTO AUDITORR VALUES(:OLD.custid,'DELETE');
END IF;
END;

```

#### ###CURSORS

```

/*Q1 Write %type and %rowtype to display details from branch table */
DECLARE
c_id branch.bid%type;
c_name branch.bname%type;
c_city branch.bcity%type;
CURSOR c_emp is
    SELECT bid,bname,bcity from branch;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp into c_id,c_name,c_city;
        EXIT WHEN c_emp%notfound;
        dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_city);
    END LOOP;
    CLOSE c_emp;
END;

```

```

/*Q2 PL/SQL code to calculate the total loan amount of first n records of loan
table. The value of n is passed to cursor as parameter*/
select * from loan order by loan_amount desc;
Declare
e number(30);
t number(30):=0;
n number :=3;
cursor c1(n number) is select loan_amount from loan;
begin
open c1(3);
loop
fetch c1 into e;
exit when c1%notfound;
t:=t+e;
end loop;
dbms_output.put_line('Total loan amount of first n customers is -->');
dbms_output.put_line(t);
close c1;
end;

```

#### ###FUNCTIONS

```

/*Function which returns total number of accounts in a branch*/
create or replace function totalAccounts(bid in varchar)
return number
is
total number(2):=0;
begin
select count(*) into total
from account

```

```

where bid=bid;
return total;
dbms_output.put_line('total no. of accounts in the branch are :');
dbms_output.put_line(total);
end;

declare
c number(2);
begin
c:=totalAccounts('B1');
dbms_output.put_line('total no. of accounts in the branch are : '|| c);
end;
dbms_output.put_line(':');
dbms_output.put_line(total);

##PROCEDURE
/*Q1 Procedure which returns the opening_balance in account with the given acnumber
Procedure with exception handling*/

create or replace procedure acc_details(a_id in varchar)
is
acc number(2);
begin
select opening_balance into acc from account where acnumber=a_id;
exception
when no_data_found then
dbms_output.put_line('Sorry no such account exist !!!');
end;

/*Q2 PROCEDURE TO CALCULATE INTEREST ON LOAN AMOUNT*/
DECLARE
principle number;
rate number;
time number;
amount number;
interest number;
function ci (p in number, r in number, t in number) return number is
BEGIN
interest:=p * power(1+(r/100),t);
return(interest);
END;
BEGIN
principle := 9678000;
rate:=5;
time:=2;
amount:=ci(principle,rate,time);
dbms_output.put_line('the interest on loan amount is :');
dbms_output.put_line(amount-principle);
END;

```