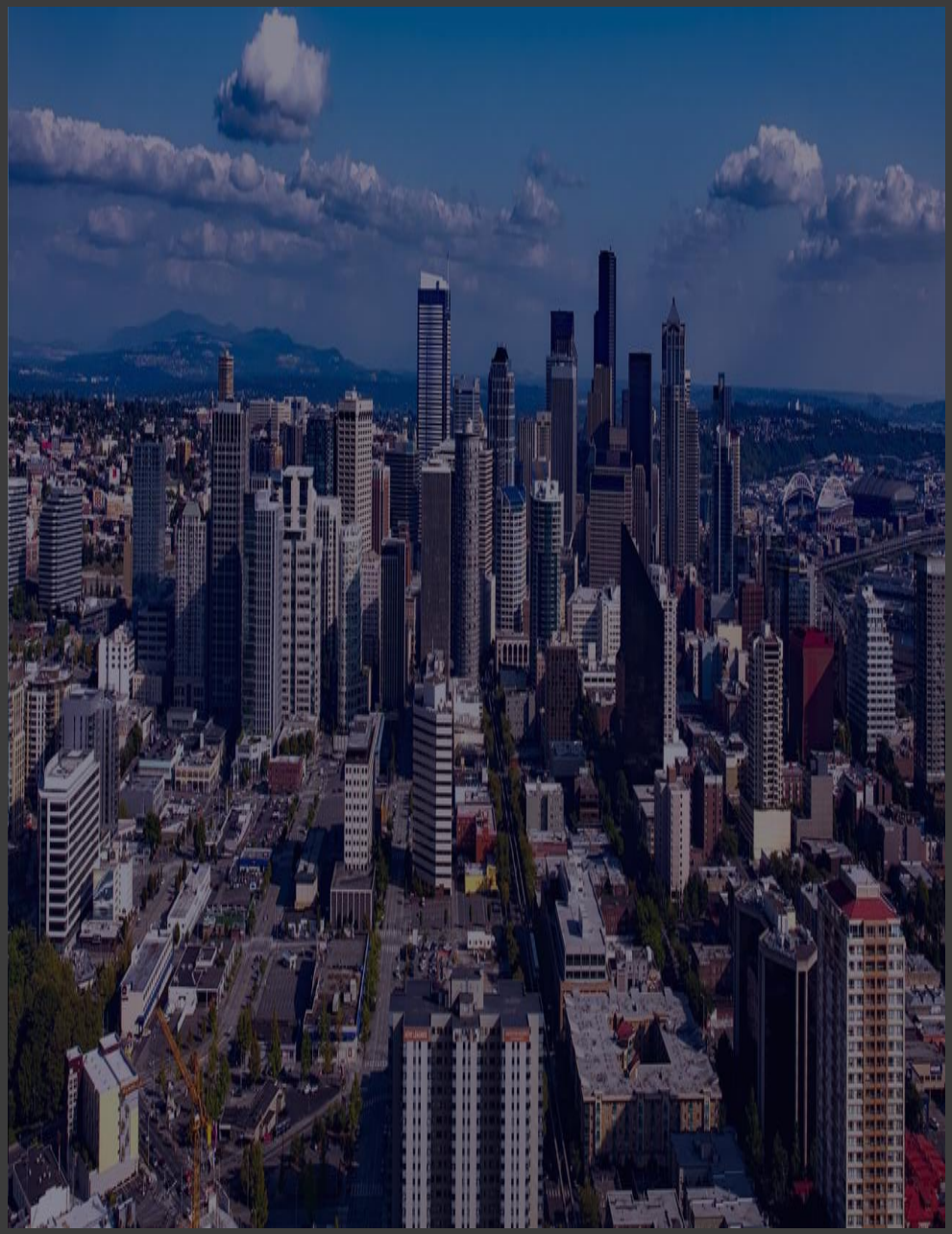


TRUST BANK -POSTGRESQL

PostgreSQL

PostgreSQL, also known as Postgres, is a powerful open-source relational database management system.

It offers a rich set of features like tablespaces, replication, nested transactions, backups, and a refined query planner/optimizer.

Additionally, PostgreSQL supports international character sets, various data types, storage of large binary objects, foreign keys, joins, views, triggers, and stored procedures

About PostgreSQL

Open-source DBMS

Fosters community development with accessible source code.

Feature-rich, supporting advanced functions like complex queries, foreign keys, and diverse data types.

Highly scalable, capable of handling large volumes of data and high-traffic applications

Boasts robust security with features like role-based access control and encryption.

Why PostgreSQL?

Offers cost-effective solutions for banks, eliminating licensing fees.

Robust security features

Multifactor authentication, encryption, and access control, which are essential for safeguarding sensitive financial data

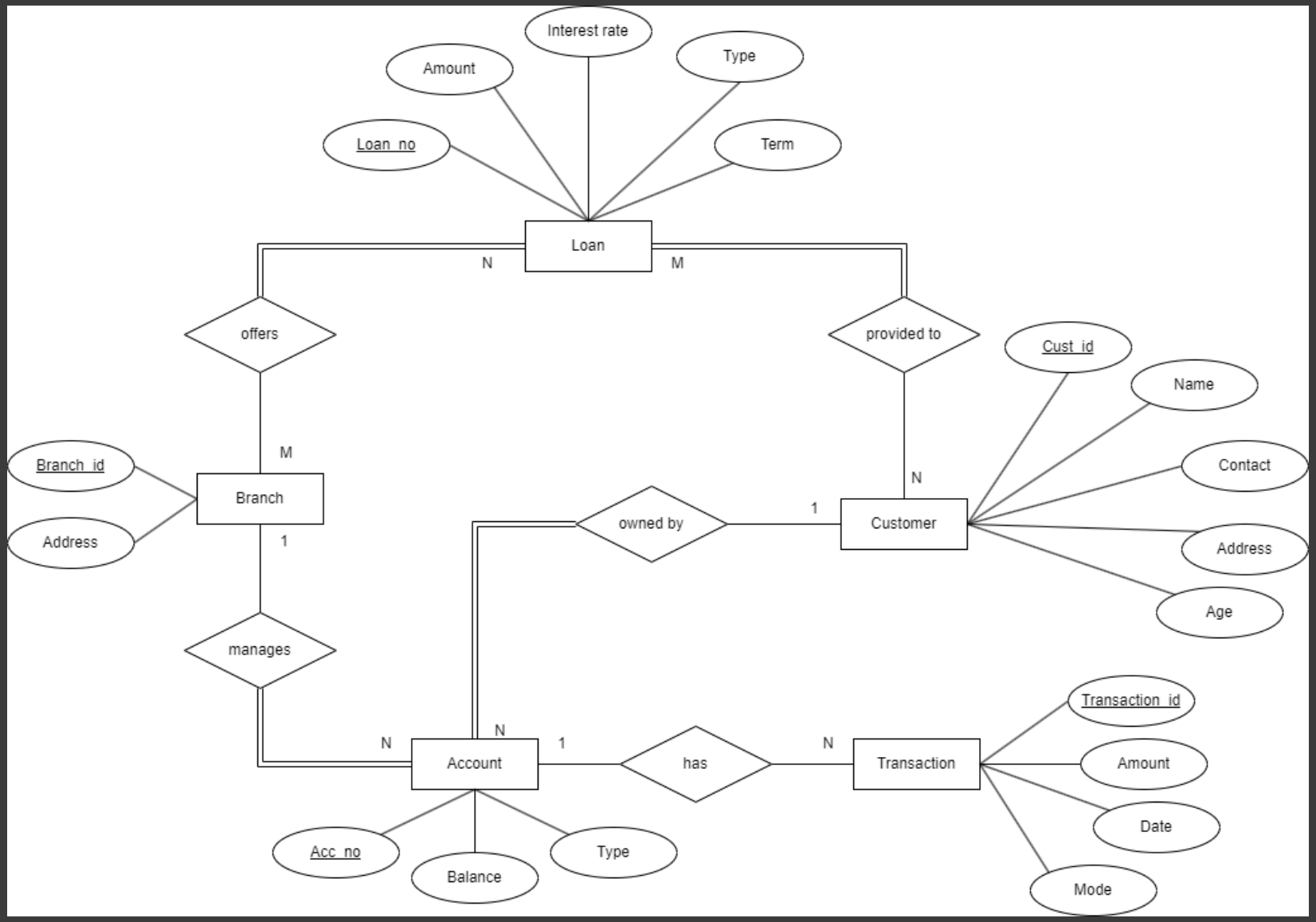
Performance tools like indexing, query optimization, and parallel processing, ideal for banks' demanding performance needs

With ACID transactions and robust data integrity features, PostgreSQL ensures compliance with banking regulations and data integrity.

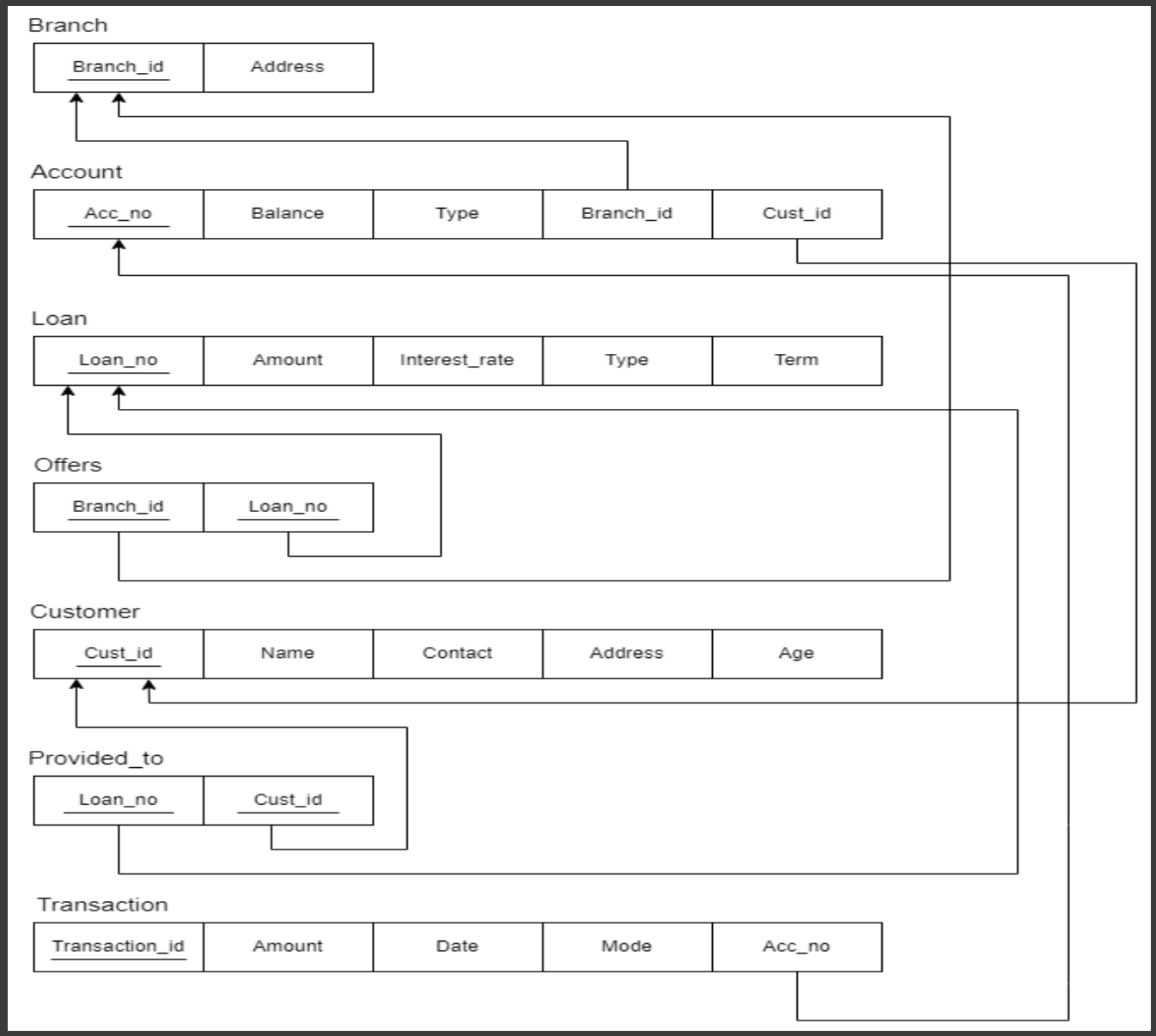
SCENARIO

- The bank operates multiple branches in different locations, each with several customer accounts.
- The database tracks the branch ID and address for each branch.
- Customers can have multiple accounts, and each account can have multiple transactions.
- Various types of loans are offered by the bank, each with a loan number, amount, interest rate, type, and term.
- Customers can obtain multiple loans simultaneously and can also avail loans jointly with other customers.
- This system is designed for a single bank.

ER DIAGRAM


Interest rate
Amount
Type
Loan_no
Term
Loan
N
M
offers
provided to
Cust_id
Name
Branch_id
M
Contact
Branch
N
Address
1
owned by
1
Customer
Address
Age
manages
Transaction_id
N
N
1
N
Account
has
Transaction
Amount
Date
Acc_no
Type
Balance
Mode

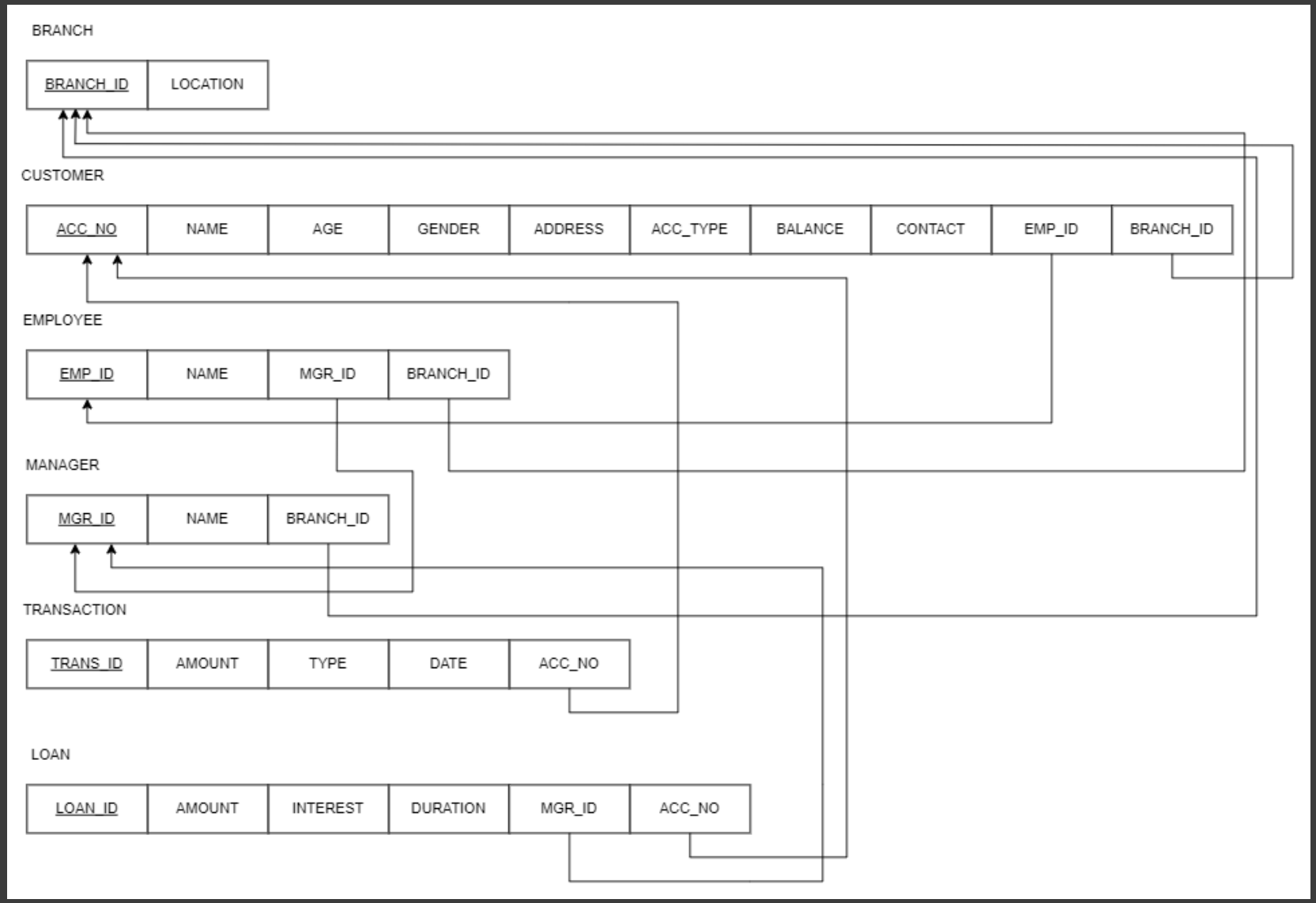
RELATIONAL SCHEMA



REFINEMENT PROCESS

- In the refinement process, we clarify the roles of employee and manager in the bank.
- Employees handle the customer interactions and account management.
- Customer interactions involve creating account for the customer and deleting the account of the customer and the account transactions.
- Managers oversee the operations of employee and customer and can provide loan to the customer.

REFINED SCHEMA



NORMALIZATION

The following normal forms are satisfied in our relational model.

- 1NF
- 2NF
- 3NF

Reason:

1NF:

The relations hold atomic values so it follows 1NF.

2NF:

The relations follow 1NF and there is no partial dependency among tables so it follows 2NF.

3NF:

The relations follow 2NF and there is no transitivity dependency so it follows 3NF.

CREATION OF TABLES

Query | Query History

```sql
CREATE TABLE branch(
    branch_id NUMERIC(3) PRIMARY KEY,
    location varchar(50)
);
SELECT * from branch;
```

Data Output | Messages | Notifications

branch_id [PK] numeric (3)	location character varying (50)

Query | Query History

```sql
CREATE TABLE manager(
    mgr_id NUMERIC(10) PRIMARY KEY,
    name VARCHAR(30),
    branch_id NUMERIC(3) REFERENCES branch(branch_id) ON DELETE CASCADE
);
SELECT * from manager;
```

Data Output | Messages | Notifications

mgr_id [PK] numeric (10)	name character varying (30)	branch_id numeric (3)

Query | Query History

```sql
CREATE TABLE employee(
    emp_id NUMERIC(10) PRIMARY KEY,
    name VARCHAR(30),
    mgr_id NUMERIC(10) REFERENCES manager(mgr_id) ON DELETE CASCADE,
    branch_id NUMERIC(3) REFERENCES branch(branch_id) ON DELETE CASCADE
);
SELECT * from employee;
```

Data Output | Messages | Notifications

emp_id [PK] numeric (10)	name character varying (30)	mgr_id numeric (10)	branch_id numeric (3)

CREATION OF TABLES

Query | Query History

```sql
CREATE TABLE customer(
    acc_no NUMERIC(10) PRIMARY KEY,
    cus_name VARCHAR(30),
    age NUMERIC(3),
    gender VARCHAR(2),
    address VARCHAR(50),
    acc_type VARCHAR(20),
    balance NUMERIC(10,2),
    contact NUMERIC(10),
    emp_id NUMERIC(10) REFERENCES employee(emp_id) ON DELETE CASCADE,
    branch_id NUMERIC(3) REFERENCES branch(branch_id) ON DELETE CASCADE
);
SELECT * from customer;
```

Data Output | Messages | Notifications

acc_no [PK] numeric (10)	cus_name character varying (30)	age numeric (3)	gender character varying (2)	address character varying (50)	acc_type character varying (20)	balance numeric (10,2)	contact numeric (10)	emp_id numeric (10)	branch_id numeric (3)

Query | Query History

```sql
CREATE TABLE transaction(
    trans_id SERIAL PRIMARY KEY,
    amount NUMERIC(12,2),
    trans_mode VARCHAR(20),
    trans_date date,
    acc_no NUMERIC(10) REFERENCES customer(acc_no) ON DELETE CASCADE);
SELECT * from transaction;
```

Data Output | Messages | Notifications

trans_id [PK] integer	amount numeric (12,2)	trans_mode character varying (20)	trans_date date	acc_no numeric (10)

Query | Query History

```sql
CREATE TABLE loan(
    loan_id SERIAL PRIMARY KEY,
    amount NUMERIC(10,2),
    interest NUMERIC(3),
    duration NUMERIC(3),
    mgr_id NUMERIC(10) REFERENCES manager(mgr_id) ON DELETE CASCADE,
    acc_no NUMERIC(10) REFERENCES customer(acc_no) ON DELETE CASCADE
);
SELECT * from loan;
```

Data Output | Messages | Notifications

loan_id [PK] integer	amount numeric (10,2)	interest numeric (3)	duration numeric (3)	mgr_id numeric (10)	acc_no numeric (10)

CONSTRAINT

Query Query History

```
1  alter table customer add constraint check_age check(age>18);
```

Data Output Messages Notifications

```
ALTER TABLE

Query returned successfully in 111 msec.
```

INSERTION OF RECORDS

Query | Query History

```sql
INSERT INTO branch VALUES
    (1,'Guindy'),
    (2,'Chrompet'),
    (3,'Pallavaram'),
    (4,'Tambaram');
SELECT * from branch;
```

Data Output | Messages | Notifications

	branch_id [PK] numeric (3)	location character varying (50)
1	1	Guindy
2	2	Chrompet
3	3	Pallavaram
4	4	Tambaram

Query | Query History

```sql
INSERT INTO manager VALUES
    (101,'lakshmi',1),
    (102,'priya',2),
    (103,'micheal',3),
    (104,'daniel',4);
SELECT * from manager;
```

Data Output | Messages | Notifications

	mgr_id [PK] numeric (10)	name character varying (30)	branch_id numeric (3)
1	101	lakshmi	1
2	102	priya	2
3	103	micheal	3
4	104	daniel	4

Query | Query History

```sql
INSERT INTO employee VALUES
    (201,'arun',101,1),
    (202,'anu',102,2),
    (203,'ram',103,3),
    (204,'arjun',103,3),
    (205,'kumar',104,4);
SELECT * from employee;
```

Data Output | Messages | Notifications

	emp_id [PK] numeric (10)	name character varying (30)	mgr_id numeric (10)	branch_id numeric (3)
1	201	arun	101	1
2	202	anu	102	2
3	203	ram	103	3
4	204	arjun	103	3
5	205	kumar	104	4

LOGIN DATABASE

Another database ‘Login’ was created to manage the account of employees and manager.

Query | Query History

```sql
CREATE TABLE login(
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(50) NOT NULL,
    designation VARCHAR(20) NOT NULL
);
SELECT * from login;
```

Data Output | Messages | Notifications

	id [PK] integer	username character varying (50)	password character varying (50)	designation character varying (20)

Query | Query History

```sql
INSERT INTO login(username,password,designation) VALUES
    ('arun','arun123*','employee'),
    ('anu','anu123*','employee'),
    ('ram','ram123*','employee'),
    ('lakshmi','lakshmi123*','manager'),
    ('priya','priya123*','manager'),
    ('michael','michael123*','manager'),
    ('daniel','daniel123*','manager');
SELECT * from login;
```

	id [PK] integer	username character varying (50)	password character varying (50)	designation character varying (20)
1	1	arun	arun123*	employee
2	2	anu	anu123*	employee
3	3	ram	ram123*	employee
4	4	arjun	arjun123*	employee
5	5	kumar	kumar123*	employee
6	6	lakshmi	lakshmi123*	manager
7	7	priya	priya123*	manager
8	8	michael	michael123*	manager
9	9	daniel	daniel123*	manager

THANK YOU

S. ABINAYA - 2022503510
A. TASNEEM - 2022503562