C.V. RAMAN GLOBAL UNIVERSITY

BHUBANESWAR, ODISHA-752054

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DBE CASE STUDY

REPORT

ON

ATM MANAGEMENT SYSTEM

NAME	SURAJ KUMAR SUBUDHI
REGD. NO.	2101020228
SEMESTER	4th
BRANCH	AI & DS

TABLE OF CONTENT

Declaration

I hereby declare that the case study work entitled *"ATM management System"* submitted to the Department of Computer Science Engineering, C.V. Raman Global University (CGU), Bhubaneswar, Odisha, Under the guidance of **Dr. Prashant Kumar Das** Sir and **Ms. Jayanti Mam**, Associated Professor, Department of Computer Science Engineering, CGU, Odisha.

I further declare that this work has not been submitted elsewhere for the award of any other degree.

Acknowledgement

We would like to thank everyone who contributed to the success of our initiative. First of all, we would like to thank Miss. Jayanti Rout and Dr. Prashant Kumar Dash from our computer science department for their guidance and support during the research. They helped us comprehend how to finish this job successfully, and without them, the project would not have been finished. This project has helped us learn and apply our academic understanding to the actual world. Therefore, we sincerely appreciate their assistance and direction with this project.

Abstract

The ATM Management System is a computer-based application that allows banking institutions to manage their ATM operations. This system provides a centralized platform to monitor, control, and track ATM-related activities, including transaction details, cash replenishment, maintenance schedules, and more. The system enables banks to manage their ATM network more efficiently, reducing downtime and minimizing the risk of fraud or security breaches. It also improves customer experience by providing reliable and secure access to banking services. The system is user-friendly and can be accessed from anywhere through a web-based interface, making it a valuable tool for banking institutions of all sizes. Overall, the ATM Management System is an essential tool that helps banks streamline their ATM operations and provide better service to their customers.

Introduction

Databases are a fundamental component of modern software systems. They are used to store, organize, and retrieve large amounts of data efficiently. In the context of a banking system, databases play a crucial role in managing customer account information and their transactions.

The ER diagram for bank management system in DBMS reveals the relationships of the bank management entities within its database. This describes the logical structure of the system's database or data storage. It is done by identifying the online banking process entities, their properties, and the interactions between them. The database design is sketched out using bank management system ER diagrams. This database sketch becomes the actual basis of the system's data storage that will serve as data destination and source.

A feasibility study and Requirement Analysis

- Accounts Management: The database should be able to manage the accounts of the bank, including account creation, deletion, and modification, as well as storing information about the balance, transaction history, and account holder details.
- Transaction Management: The database should be able to manage the transactions made by customers, including deposit, withdrawal, transfer, and other transactions. The transaction information should be associated with the corresponding accounts and customers.
- Customer Management: The database should be able to manage customer information, including personal details, contact information, and account details.
- Employee Management: The database should be able to manage employee information, including personal details, contact information, job title, and work schedule.
- Report Generation: The database should be able to generate reports on customer transactions, employee activities, loan status, and other key performance indicators.
- Security: The database should be able to ensure the security and privacy of customer and employee data by implementing appropriate authentication and authorization measures.

Table Description with all constraints

ENTITY USED:

- BANK
- BRANCH
- CUSTOMERS
- ACCOUNTS
- DEBIT
- REPORTS

1. BANK:

Field	Type	Null	Key	Default	Extra
bank_id	int(11)	NO	PRI	NULL	
bank_name	varchar(50)	NO		NULL	
bank_address	varchar(100)	YES		NULL	
bank_phone	varchar(20)	YES		NULL	
bank_email	varchar(50)	YES		NULL	

2. BRANCH:

Field	Type	Null	Key	Default	Extra
branch_id	int(11)	NO	PRI	NULL	
branch_name	varchar(50)	NO		NULL	
branch_address	varchar(100)	YES		NULL	
branch_phone	varchar(20)	YES		NULL	
branch_email	varchar(50)	YES		NULL	
bank_id	int(11)	NO	MUL	NULL	

FOREIGN KEY (bank_id) REFERENCES bank(bank_id)

3. CUSTOMERS:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(50)	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
middle_name	varchar(50)	YES		NULL	
last_name	varchar(50)	NO		NULL	
email	varchar(50)	NO		NULL	
contact_number	bigint(10)	NO		NULL	
address	varchar(100)	YES		NULL	

4. ACCOUNTS:

Field	Type	Null	Key	Default	Extra
account_number	varchar(50)	NO	PRI	NULL	
account_type	varchar(50)	YES		NULL	
balance	decimal(10,2)	YES		NULL	
customer_id	varchar(50)	YES	MUL	NULL	

FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

5. CUSTOMER ACCOUNTS:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(50)	NO	PRI		
account_number	varchar(50)	NO	PRI		
balance	int(10)	YES		NULL	

PRIMARY KEY (customer_id, account_number),

FOREIGN KEY (customer_id) REFERENCES customers (customer_id),

FOREIGN KEY (account_number) REFERENCES accounts (account_number)

6. DEBIT:

Field	Type	Null	Key	Default	Extra
debit_id	int(10)	NO		NULL	
Card_holder_name	varchar(255)	YES		NULL	
pin_number	varchar(255)	YES		NULL	
amount	varchar(255)	YES		NULL	
account_number	varchar(255)	YES		NULL	

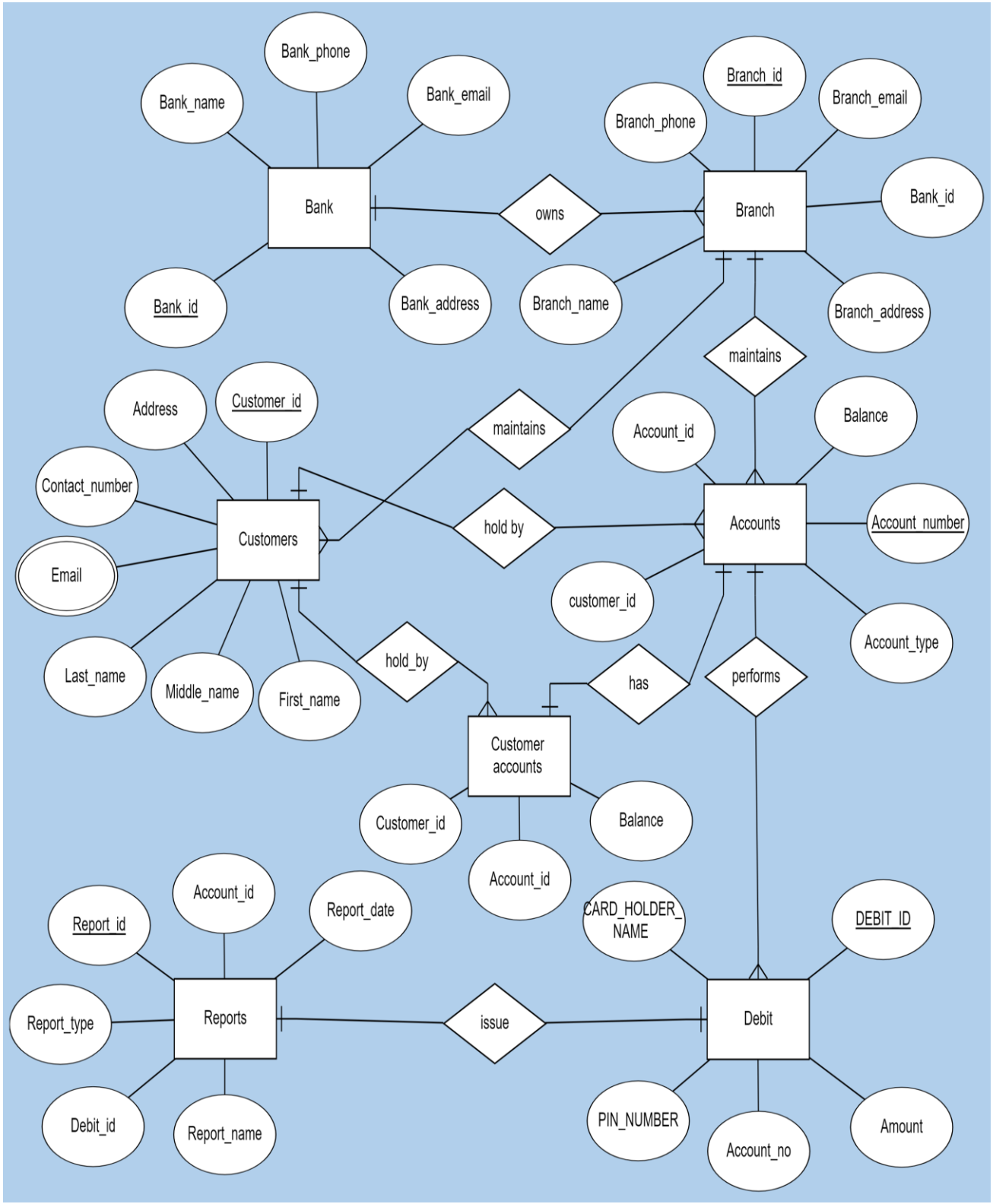
FOREIGN KEY (account_number) REFERENCES accounts(account_number)

7. REPORTS:

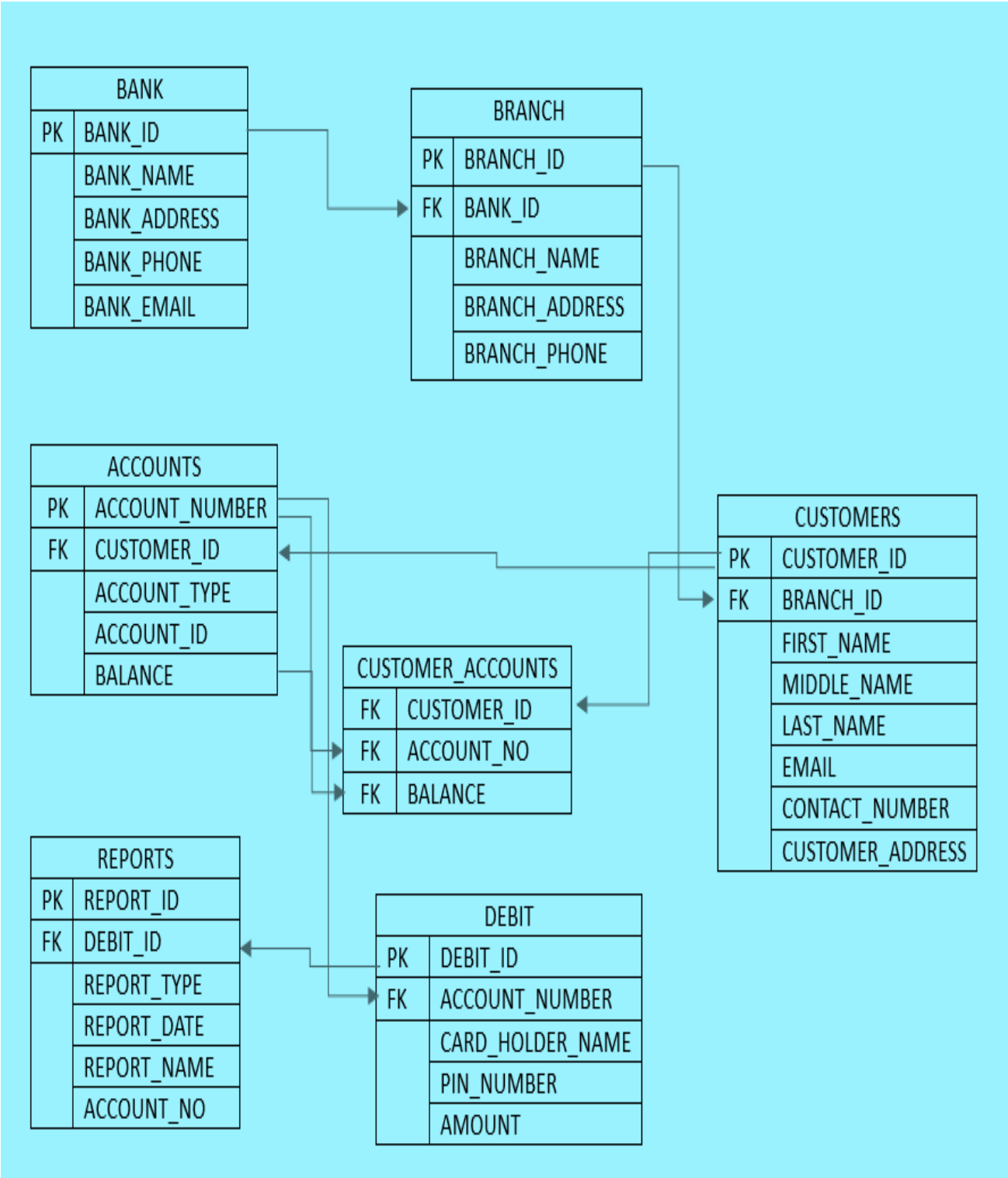
Field	Type	Null	Key	Default	Extra
report_id	int(11)	NO	PRI	NULL	
report_date	varchar(100)	YES		NULL	
report_type	varchar(50)	YES		NULL	
report_name	varchar(50)	NO		NULL	
transaction_id	int(11)	YES	MUL	NULL	

FOREIGN KEY (debit_id) REFERENCES debit(debit_id))

Design ER Diagram



Relational Database Schema



Unnormalized Table

Field	Type	Null	Key	Default	Extra
bank_id	int(10)	YES		NULL	
bank_name	varchar(50)	YES		NULL	
bank_address	varchar(100)	YES		NULL	
bank_phone	varchar(20)	YES		NULL	
bank_email	varchar(50)	YES		NULL	
branch_id	int(10)	YES		NULL	
branch_name	varchar(50)	YES		NULL	
branch_address	varchar(100)	YES		NULL	
branch_phone	varchar(20)	YES		NULL	
customer_id	varchar(50)	NO	PRI		
first_name	varchar(50)	YES		NULL	
middle_name	varchar(50)	YES		NULL	
last_name	varchar(50)	YES		NULL	
email	varchar(50)	YES		NULL	
contact_number	int(10)	YES		NULL	
customer_address	varchar(100)	YES		NULL	
account_number	varchar(50)	NO	PRI		
account_type	varchar(50)	YES		NULL	
account_id	varchar(20)	YES		NULL	
debit_id	int(10)	YES		NULL	
card_holder_name	varchar(255)	YES		NULL	
pin_number	varchar(255)	YES		NULL	
report_id	int(10)	YES		NULL	
report_date	varchar(100)	YES		NULL	
report_type	varchar(50)	YES		NULL	
report_name	varchar(50)	YES		NULL	

UNNORMALIZED TO 1NF

To convert the unnormalized relation for the entities Bank, Branch, Customers, Accounts, Report, and Debit to 1NF, we need to identify the repeating groups and separate them into their own tables. Based on the structure of the unnormalized relation, we can identify three repeating groups: Bank, Branch, and Customer.

We created the TABLE name as: BANK, BRANCH, ACCOUNTS, REPORTS, DEBIT.

1NF Normalized (Multivalued attribute->Email):

customer_id	name	email	contact_Number	address
1	Diwakar	diwakar12@gmail.com	6200896422	Ranchi, Jharkhand
1	Shakti	Shakti03@gmail.com	6200896422	Bhubaneswar, Odisha
2	Suraj	Suraj2505@gmail.com	9786123467	Bhubaneswar, Odisha
3	Amnita	Amnita07@gmail.com	9764679426	Dhanbad, Jharkhand

Here multivalued attribute is divided into another table:

customer_id	name	contact_Number	address
1	Diwakar	6200896422	Ranchi, Jharkhand
1	Shakti	6200896422	Ranchi, Odisha
2	Suraj	9786123467	Bhubaneswar, Odisha
3	Amnita	9764679426	Dhanbad, Jharkhand

customer_id	email
1	diwakar12@gmail.com
1	Shakti03@gmail.com
2	Suraj2505@gmail.com
3	Amnita07@gmail.com

2NF Normalization

In the 1NF version, we have separated the repeating groups of Bank, Branch, and Customer into their own tables, and established relationships between them using foreign keys. This ensures that each table contains only atomic values, and there is no data redundancy. The primary keys have been defined for each table to ensure unique identification of each row. To convert the 1NF tables for the entities Bank, Branch, Customers, Accounts, Report, and Debit to 2NF, we need to identify any partial dependencies and separate them into their own tables. Partial dependency exists when a non-key attribute is dependent on only a portion of a composite primary key. In our case, we can see that the Accounts table has a partial dependency on the Customers table, as the Account table's non-key attribute (balance) is dependent on only the customer_id portion of its composite primary key.

Field	Type	Null	Key	Default	Extra
customer_id	varchar(50)	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
middle_name	varchar(50)	YES		NULL	
last_name	varchar(50)	NO		NULL	
email	varchar(50)	NO		NULL	
contact_number	bigint(10)	NO		NULL	
address	varchar(100)	YES		NULL	

Partial dependent

Field	Type	Null	Key	Default	Extra
account_number	varchar(50)	NO	PRI	NULL	
account_type	varchar(50)	YES		NULL	
balance	decimal(10,2)	YES		NULL	
customer_id	varchar(50)	YES	MUL	NULL	

To address this, we can create a new table called Customer_Accounts that will join Customers and Accounts tables.

Here's how we can convert the 1NF tables to 2NF:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(50)	NO	PRI		
account_number	varchar(50)	NO	PRI		
balance	int(10)	YES		NULL	

Conclusion

In conclusion, an ATM management program is a useful tool for banks and financial institutions to manage their ATM network efficiently. The program allows them to monitor the ATMs in real-time, perform maintenance and repairs, manage cash and transactions, and generate reports for analysis. It also enhances customer experience by providing quick and easy access to their accounts and ensuring the availability of cash and other services. With the increasing use of technology in the banking industry, implementing an ATM management program can help institutions stay competitive and meet the changing needs of their customers.

Reference

- **https://www.tutorialspoint.com**
- **https://www.sanfoundry.com**
- **https://sourceforge.net**
- **https://worldline.com/en/home/main-navigation/solutions/financial- institutions/acquiring/atm-management.html**