Alliance University

Alliance College of Engineering and Design

Assignment on Case Study

"Bank Database"

Database Management Systems: - Code No: CS504

V-Semester Section A – 2016 Batch

Under guidance of:

Prof.Bhoomika.A.P

Submitted by:

Abhishek .M
Reg.No: 16030141CSE006

Jay Dev Rai .M
Reg.No: 16030141CSE037

Kishan kumar Reddy
Reg.No: 16030141CSE043

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the HOSPITAL DATABASE REPORT title STUDENT REGISTRATION was presented by **Abhishek.M** (Reg. No: 16030141CSE006) ,**Jay dev rai.M** (Reg. No: 16030141CSE037) &**Kishan Kumar Reddy** (Reg. No: 16030141CSE043) in partial fulfillment of the requirement for the Award of Degree of Bachelor of Technology in Computer Science and Engineering during the academic year 2018-2019 at Alliance University, Bangalore.

Faculty Signature:

_____

TABLE OF CONTENTS:

INTRODUCTION

Database Management System (DBMS)

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

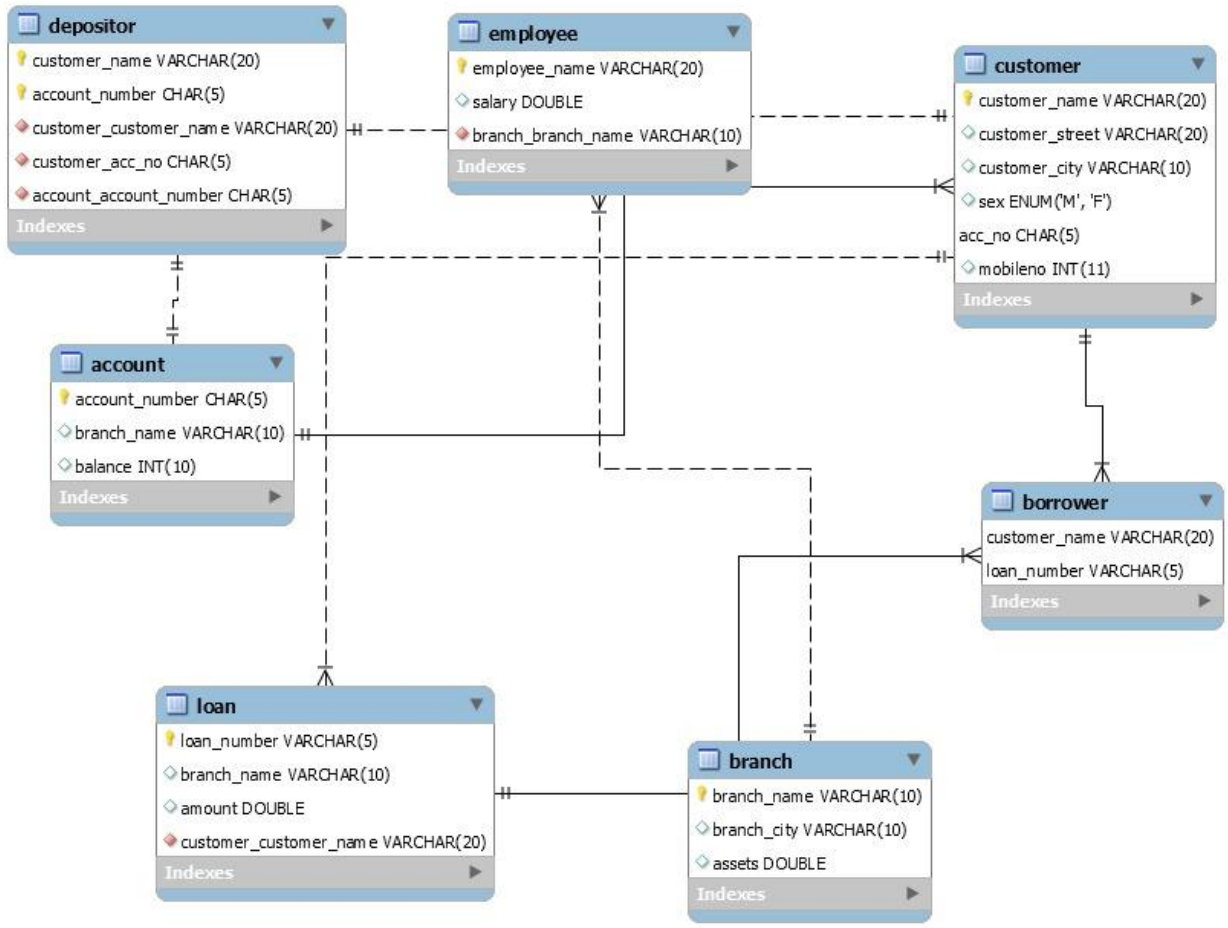
The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware). As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because changes have been made to the database.

□ Database Applications:

- Data abstraction and independence
- Data security
- A locking mechanism for concurrent access
- An efficient handler to balance the needs of multiple applications using the same data
- The ability to swiftly recover from crashes and errors, including restartability and recoverability
- Robust data integrity capabilities
- Logging and auditing of activity
- Simple access using a standard application programming interface (API)
- Uniform administration procedures for data

DESIGN

ER diagram



```
CREATE SCHEMA IF NOT EXISTS `bank` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
;
USE `bank` ;


-- -----------------------------------------------------
-- Table `bank`.`account`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`account` (
  `account_number` CHAR(5) NOT NULL,
  `branch_name` VARCHAR(10) NULL DEFAULT NULL,
  `balance` DOUBLE NULL DEFAULT NULL,
  PRIMARY KEY (account_number));


-- -----------------------------------------------------
-- Table `bank`.`customer`
-- -----------------------------------------------------
CREATE TABLE accountIF NOT EXISTS `bank`.`customer` (
  `customer_name` VARCHAR(20) NOT NULL,
  `customer_street` VARCHAR(20) NULL DEFAULT NULL,
  `customer_city` VARCHAR(10) NULL DEFAULT NULL,
  `sex` ENUM('M', 'F') NULL,
  `acc_no` CHAR(5) NOT NULL,
  PRIMARY KEY (`customer_name`, `acc_no`),
  INDEX `acc_no_idx` (`acc_no` ASC) VISIBLE,
```

```
  CONSTRAINT `acc_no`
    FOREIGN KEY (`acc_no`)
    REFERENCES `bank`.`account` (`account_number`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `bank`.`loan`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`loan` (
  `loan_number` VARCHAR(5) NOT NULL,
  `branch_name` VARCHAR(10) NULL DEFAULT NULL,
  `amount` DOUBLE NULL DEFAULT NULL,
  `customer_customer_name` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`loan_number`),
  INDEX `fk_loan_customer1_idx` (`customer_customer_name` ASC) VISIBLE,
  CONSTRAINT `fk_loan_customer1`
    FOREIGN KEY (`customer_customer_name`)
    REFERENCES `bank`.`customer` (`customer_name`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----------------------------------------------------
-- Table `bank`.`borrower`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`borrower` (
  `customer_name` VARCHAR(20) NOT NULL,
  `loan_number` VARCHAR(5) NOT NULL,
  PRIMARY KEY (`customer_name`, `loan_number`),
  INDEX `b_idx` (`loan_number` ASC) VISIBLE,
  CONSTRAINT `a`
    FOREIGN KEY (`customer_name`)
    REFERENCES `bank`.`customer` (`customer_name`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `b`
    FOREIGN KEY (`loan_number`)
    REFERENCES `bank`.`loan` (`loan_number`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `bank`.`branch`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`branch` (
  `branch_name` VARCHAR(10) NOT NULL,
  `branch_city` VARCHAR(10) NULL DEFAULT NULL,
```

```
  `assets` DOUBLE NULL DEFAULT NULL,
  PRIMARY KEY (`branch_name`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `bank`.`depositor`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`depositor` (
  `customer_name` VARCHAR(20) NOT NULL,
  `account_number` CHAR(5) NOT NULL,
  PRIMARY KEY (`customer_name`, `account_number`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `bank`.`employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`employee` (
  `employee_name` VARCHAR(20) NOT NULL,
  `branch_name` VARCHAR(10) NOT NULL,
  `salary` DOUBLE NULL DEFAULT NULL,
  `branch_branch_name` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`employee_name`, `branch_name`),
  INDEX `fk_employee_branch_idx` (`branch_branch_name` ASC) VISIBLE,
```

```
  CONSTRAINT `fk_employee_branch`
    FOREIGN KEY (`branch_branch_name`)
    REFERENCES `bank`.`branch` (`branch_name`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

Code Implementation:

```
package p1;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.Scanner;


public class a1 {

	static int i=18;


	public static void main(String[] args) throws Exception {

		int deposit1,deposit2,deposit4,with1,with2,with4;

		String deposit3,with3,get1,get2;

		String name,street,gender,accno2;

		int mob;

		Scanner sc=new Scanner(System.in);

		while(true) {

			System.out.println("Enter your choice\n1.create
account\n2.deposit\n3.withdraw\n4.customer details\n5.loan details\n6.exit\n");

			int choice;

			choice=sc.nextInt();


			switch(choice) {

				case 1:{

					System.out.println("enter name : \n");

					name=sc.next();

					System.out.println("enter street : \n");
```

```
                street=sc.next();
                System.out.println("enter gender(M/F) : \n");
                gender=sc.next();
                System.out.println("enter mobile number : \n");
                mob=sc.nextInt();
                System.out.println("enter account number : \n");
                accno2=sc.next();
                createaccount(name,street,gender,mob,accno2);

        }break;
        case 2:{
                System.out.println("enter account number : ");
                deposit3=sc.next();
                System.out.println("enter amount to deposit : ");
                deposit1=sc.nextInt();
                deposit2=getbalance(deposit3).get(0);
                deposit4=deposit2+deposit1;
                putbalance(deposit3,deposit4);
                System.out.println("deposit Completed.\n");
        }break;
        case 3:{
                System.out.println("enter account number : ");
                with3=sc.next();
                System.out.println("enter amount to withdraw : ");
                with1=sc.nextInt();
                with2=getbalance(with3).get(0);
                if(with2>=with1) {
                with4=with2-with1;
                putbalance(with3,with4);
```

```
                    System.out.println("withdraw Completed.\n");}
                    else {System.out.println("\nInsufficient balance\n");}

                }break;
                case 4:{
                    System.out.println("enter account number : ");
                    get1=sc.next();
                    getdetails(get1);
                }break;
                case 5:{
                    System.out.println("enter loan number : ");
                    get2=sc.next();
                    loandetails(get2);
                }break;
                default:System.exit(0);sc.close();break;
            }

        }

    }
    public static void putbalance(String accno,int newbal) throws Exception{

        try{
            Connection con = getconnection();
            PreparedStatement updatebalance = con.prepareStatement("update account set balance
="+newbal+" where account_number = '"+accno+"'; ");

            updatebalance.executeUpdate();
        } catch(Exception e){System.out.println(e);}
```

```
            finally {
               //System.out.println("deposit Completed.\n");
            }
          }
        public static int accountnumber() {
                i++;
               return i;

        }
        public static void createaccount(String name,String street,String gender,int mob,String accno2)
throws Exception{

            try{
               Connection con = getconnection();
               PreparedStatement insertaccount = con.prepareStatement("insert into
account(account_number,branch_name,balance) values('"+accno2+"','anekal','100');");
               PreparedStatement insertcustomer = con.prepareStatement("insert into
customer(customer_name,customer_street,customer_city,sex,acc_no,mobileno)
values('"+name+"','"+street+"','bangalore','"+gender+"','"+accno2+"','"+mob+"');");

               insertaccount.executeUpdate();
               insertcustomer.executeUpdate();
            } catch(Exception e){System.out.println(e);}
            finally {
               System.out.println(" account created.");
            }
          }
        public static ArrayList<Integer> getbalance(String accno1) {
               try{
               Connection con = getconnection();
```

```
                PreparedStatement statement = con.prepareStatement("SELECT balance FROM account
where account_number = '"+accno1+"'");


                ResultSet result = statement.executeQuery();
                ArrayList<Integer> array = new ArrayList<Integer>();
                while(result.next()){
                   array.add(result.getInt("balance"));
                }
                return array;


            }catch(Exception e){System.out.println(e);}
            return null;
        }
        public static void getdetails(String accno1) {
                  try{
                Connection con = getconnection();
                PreparedStatement statement = con.prepareStatement("SELECT * FROM customer where
acc_no = '"+accno1+"'");


                ResultSet result = statement.executeQuery();
                while(result.next()){
                   System.out.println("name : "+result.getString("customer_name"));
                   System.out.println("street : "+result.getString("customer_street"));
                   System.out.println("city : "+result.getString("customer_city"));
                   System.out.println("gender : "+result.getString("sex"));
                   System.out.println("account number : "+result.getString("acc_no"));
                   System.out.println("mobile number : "+result.getString("mobileno"));
                }
            }catch(Exception e){System.out.println(e);}
```

```
        }


        public static void loandetails(String accno1) {
            try{
            Connection con = getconnection();
            PreparedStatement statement = con.prepareStatement("SELECT * FROM loan where
loan_number = '"+accno1+"';");

            ResultSet result = statement.executeQuery();
            while(result.next()){
              System.out.println("loan number : "+result.getString("loan_number"));
              System.out.println("branch name : "+result.getString("branch_name"));
              System.out.println("amount : "+result.getString("amount"));
              System.out.println("customer name : "+result.getString("customer_customer_name"));

            }
          }catch(Exception e){System.out.println(e);}
        }

        public static Connection getconnection() throws Exception{
            try {
                String driver="com.mysql.jdbc.Driver";
                String url="jdbc:mysql://localhost/bank";
                String user="root";
                String pass="jaydev@1";
                Class.forName(driver);

                Connection conn= DriverManager.getConnection(url,user,pass);
                //System.out.println("connected");
```

```
                return conn;
            }catch(Exception e) {System.out.println(e);}
            return null;
        }

}
```

IMPLEMENTATION

Connection

```
                System.out.println("account number :  "+result.getString("acc_no"));
                System.out.println("mobile number : "+result.getString("mobileno"));
            }
        }catch(Exception e){System.out.println(e);}
    }

    public static void loandetails(String accno1) {
        try{
                Connection con = getconnection();
                PreparedStatement statement = con.prepareStatement("SELECT * FROM loan where loan_number = '"+accno1+"';");

                ResultSet result = statement.executeQuery();
                while(result.next()){
                    System.out.println("loan number : "+result.getString("loan_number"));
                    System.out.println("branch name : "+result.getString("branch_name"));
                    System.out.println("amount : "+result.getString("amount"));
                    System.out.println("customer name : "+result.getString("customer_customer_name"));

                }
        }catch(Exception e){System.out.println(e);}
    }

    public static Connection getconnection() throws Exception{
        try {
            String driver="com.mysql.jdbc.Driver";
            String url="jdbc:mysql://localhost/bank";
            String user="root";
            String pass="jaydev@1";
            Class.forName(driver);

            Connection conn= DriverManager.getConnection(url,user,pass);
            //System.out.println("connected");
            return conn;
        }catch(Exception e) {System.out.println(e);}
        return null;
    }

}
```

Inserting to the Database

```
                    deposit1=sc.nextInt();
                    deposit2=getbalance(deposit3).get(0);
                    deposit4=deposit2+deposit1;
                    putbalance(deposit3,deposit4);
                    System.out.println("deposit Completed.\n");
                }break;
                case 3:{
                    System.out.println("enter account number : ");
                    with3=sc.next();
                    System.out.println("enter amount to withdraw : ");
                    with1=sc.nextInt();
                    with2=getbalance(with3).get(0);
                    if(with2>=with1) {
                    with4=with2-with1;
                    putbalance(with3,with4);
                    System.out.println("withdraw Completed.\n");}
                    else {System.out.println("\nInsufficient balance\n");}

                }break;
                case 4:{
                    System.out.println("enter account number : ");
                    get1=sc.next();
                    getdetails(get1);
                }break;
                case 5:{
                    System.out.println("enter loan number : ");
                    get2=sc.next();
                    loandetails(get2);
                }break;
                default:System.exit(0);sc.close();break;
            }

        }

    }
    public static void putbalance(String accno,int newbal) throws Exception{

            try{
```

Updating database

```
78
79              try{
80                  Connection con = getconnection();
81                  PreparedStatement updatebalance = con.prepareStatement("update account set balance ="+newbal+" where account_number = '"+accno+"'; ");
82
83                  updatebalance.executeUpdate();
84              } catch(Exception e){System.out.println(e);}
85              finally {
86                  //System.out.println("deposit Completed.\n");
87              }
88          }
89      public static int accountnumber() {
90          i++;
91         return i;
92
93      }
94      public static void createaccount(String name,String street,String gender,int mob,String accno2) throws Exception{
95
96              try{
97                  Connection con = getconnection();
98                  PreparedStatement insertaccount = con.prepareStatement("insert into account(account_number,branch_name,balance) values('"+accno2+"','anekal','100');");
99                  PreparedStatement insertcustomer = con.prepareStatement("insert into customer(customer_name,customer_street,customer_city,sex,acc_no,mobileno) values('"+name+
100
101                 insertaccount.executeUpdate();
102                 insertcustomer.executeUpdate();
103             } catch(Exception e){System.out.println(e);}
104             finally {
105                 System.out.println(" account created.");
106             }
107         }
108     public static ArrayList<Integer> getbalance(String accno1) {
109         try{
110                 Connection con = getconnection();
111                 PreparedStatement statement = con.prepareStatement("SELECT balance FROM account where account_number = '"+accno1+"'");
112
113                 ResultSet result = statement.executeQuery();
114                 ArrayList<Integer> array = new ArrayList<Integer>();
115                 while(result.next()){
116                     array.add(result.getInt("balance"));
```

Deleting from database

```
110                 Connection con = getconnection();
111                 PreparedStatement statement = con.prepareStatement("SELECT balance FROM account where account_number = '"+accno1+"'");
112
113                 ResultSet result = statement.executeQuery();
114                 ArrayList<Integer> array = new ArrayList<Integer>();
115                 while(result.next()){
116                     array.add(result.getInt("balance"));
117                 }
118                 return array;
119
120             }catch(Exception e){System.out.println(e);}
121             return null;
122     }
123     public static void getdetails(String accno1) {
124         try{
125                 Connection con = getconnection();
126                 PreparedStatement statement = con.prepareStatement("SELECT * FROM customer where acc_no = '"+accno1+"'");
127
128                 ResultSet result = statement.executeQuery();
129                 while(result.next()){
130                     System.out.println("name : "+result.getString("customer_name"));
131                     System.out.println("street : "+result.getString("customer_street"));
132                     System.out.println("city : "+result.getString("customer_city"));
133                     System.out.println("gender : "+result.getString("sex"));
134                     System.out.println("account number : "+result.getString("acc_no"));
135                     System.out.println("mobile number : "+result.getString("mobileno"));
136                 }
137             }catch(Exception e){System.out.println(e);}
138     }
139
140     public static void loandetails(String accno1) {
141         try{
142                 Connection con = getconnection();
143                 PreparedStatement statement = con.prepareStatement("SELECT * FROM loan where loan_number = '"+accno1+"';");
144
145                 ResultSet result = statement.executeQuery();
146                 while(result.next()){
147                     System.out.println("loan number : "+result.getString("loan_number"));
```

RESULT AND SNAPSHOTS

Creating and Inserting to the Database

```
a1 [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\java
Enter your choice
1.create account
2.deposit
3.withdraw
4.customer details
5.loan details
6.exit

1
enter name :

abc
enter street :

5thcross
enter gender(M/F) :

M
enter mobile number :

99866547
enter account number :

ac22
Loading class `com.mysql.jdbc.Driver'. This is d
Wed Oct 24 22:26:14 IST 2018 WARN: Establishing
 account created.
Enter your choice
1.create account
2.deposit
3.withdraw
4.customer details
5.loan details
6.exit
```

Deposit And Withdraw

```
2
enter account number :
ac22
enter amount to deposit :
200
Wed Oct 24 22:26:38 IST 2018 WARN: Establishing SSL conn
Wed Oct 24 22:26:38 IST 2018 WARN: Establishing SSL conn
deposit Completed.

Enter your choice
1.create account
2.deposit
3.withdraw
4.customer details
5.loan details
6.exit

3
enter account number :
ac22
enter amount to withdraw :
100
Wed Oct 24 22:26:53 IST 2018 WARN: Establishing SSL conn
Wed Oct 24 22:26:53 IST 2018 WARN: Establishing SSL conn
withdraw Completed.

Enter your choice
1.create account
2.deposit
3.withdraw
4.customer details
5.loan details
6.exit
```

Customer details and Loan Details

```
4
enter account number :
ac22
Wed Oct 24 22:27:02 IST 2018 WARN: Establishing
name : abc
street : 5thcross
city : bangalore
gender : M
account number : ac22
mobile number : 99866547
Enter your choice
1.create account
2.deposit
3.withdraw
4.customer details
5.loan details
6.exit

5
enter loan number :
l17
Wed Oct 24 22:27:21 IST 2018 WARN: Establishing
loan number : l17
branch name : mejestic
amount : 250.0
customer name : apoorva
Enter your choice
1.create account
2.deposit
3.withdraw
4.customer details
5.loan details
6.exit
```