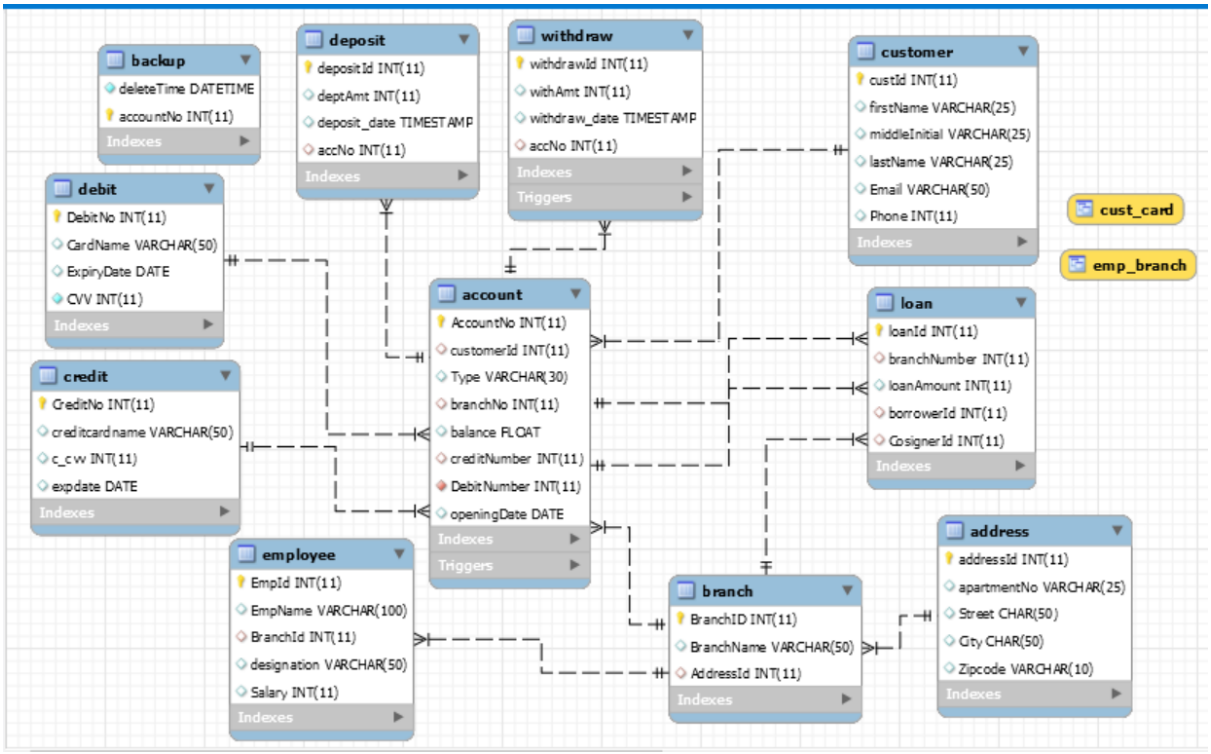
SHRUTI WALAWALKAR

BANKING SYSTEM DATABASE

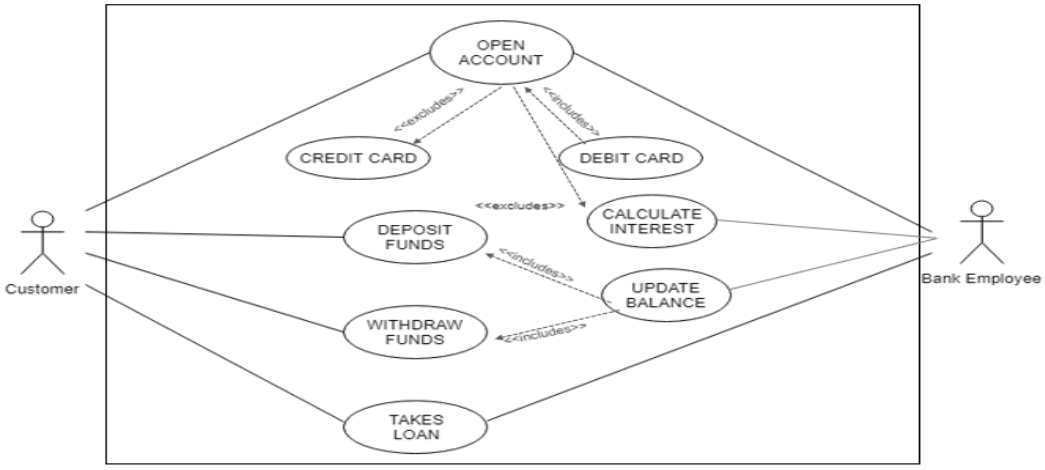
The project aims to support internal business process of a bank. The objective is to upgrade the banking business from paper to electronic thus enabling us to carry out banking processes easily and quickly. The project will be deployed with a front end and would run from workstations to assist the bank employees.

The database was designed considering requirements and work flow of daily transactions of the banking industry.

ER DIAGRAM:



USE CASE:



TRIGGERS:

SHRUTI WALAWALKAR

1. Trigger to check if an account already exists.

```
mysql> create trigger checkvalidity
    -> before insert on account
    -> for each row begin
    -> if new.accountNo in (
    -> select account.accountNo from account where new.accountNo = account.accountNo)
    -> then signal sqlstate '45000'
    ->
    -> set message_text= 'account already exists';
    -> end if;
    -> end//
Query OK, 0 rows affected (0.22 sec)
```

```
mysql> insert into account values(265,1233,'saving',3,599,70,8654,'2020-10-28');
    -> //
ERROR 1644 (45000): account already exists
```

2. Trigger to take backup of account details on deletion.

```
mysql> create table backup( deleteTime datetime not null default '0000-00-00 00:00:00', accountNo int , primary key (accountNo) );
    -> //
Query OK, 0 rows affected (0.56 sec)

mysql> create trigger backup
    -> after delete on account
    -> for each row
    -> begin
    -> insert into backup values (now(),old.accountNo);
    -> end//
Query OK, 0 rows affected (0.17 sec)

mysql> DELETE FROM ACCOUNT WHERE ACCOUNTNO=50;
    -> //
Query OK, 1 row affected (0.10 sec)

mysql> SELECT * FROM BACKUP;
    -> //
+---------------------+-----------+
| deleteTime          | accountNo |
+---------------------+-----------+
| 2018-12-08 00:27:02 |        50 |
+---------------------+-----------+
1 row in set (0.00 sec)
```

3. Trigger to update balance on withdraw

```
mysql> select balance from account where accountNo=6344//
+---------+
| balance |
+---------+
|     570 |
+---------+
1 row in set (0.00 sec)
```

```
mysql> create trigger updatebalance
    -> after insert on withdraw
    -> for each row
    -> begin
    -> select withdraw.withAmt into @amt from withdraw inner join account on
withdraw.accNo=account.accountNo;
    -> update account
    -> set balance=balance - @amt
    -> where accountNo in (select accNo from withdraw);
    -> end;
    -> //
Query OK, 0 rows affected (0.35 sec)

mysql> insert into withdraw(withdrawid, withAmt, accNo) values (1,20,6344)//
Query OK, 1 row affected (0.33 sec)

mysql> select balance from account where accountno=6344//
+---------+
| balance |
+---------+
|     550 |
+---------+
1 row in set (0.00 sec)

mysql>
```

PROCEDURES:

1. Procedure to calculate interest of given account number

```
mysql> delimiter //
mysql> create procedure calculate_interest(in a_account_id int, out b_balance float)
    -> begin
    -> declare o_openingdate date;
    -> declare diff1 int;
    -> select balance,openingdate into b_balance,o_openingdate from account where accountno=a_account_id;
    -> if o_openingdate+interval 12 month<=curdate() then
    -> SELECT DATEDIFF(curdate(), o_openingdate) into diff1;
    -> set diff1 = (diff1/365);
    -> repeat
    -> update account
    -> set balance = b_balance*1.03
    -> where accountno=a_account_id;
    -> set diff1=diff1-1;
    -> until diff1>=0
    -> end repeat;
    -> end if;
    -> end//
Query OK, 0 rows affected (0.19 sec)
```

```
mysql> select * from account//
+-----------+------------+----------+----------+---------+--------------+-------------+-------------+
| AccountNo | customerId | Type     | branchNo | balance | creditNumber | DebitNumber | openingDate |
+-----------+------------+----------+----------+---------+--------------+-------------+-------------+
|       265 |       1233 | checking |        1 | 562.754 |         NULL |        3066 | 2016-01-01  |
|       874 |       2753 | checking |        2 |     550 |          335 |        4807 | 2016-01-01  |
|      4994 |       1233 | saving   |        1 |   50000 |         NULL |        3066 | 2016-01-01  |
|      5221 |       2467 | saving   |        2 |   55000 |         NULL |        2739 | 2016-01-01  |
|      5600 |       1479 | saving   |        2 |   40000 |         NULL |        4180 | 2016-01-01  |
|      6344 |       1479 | checking |        2 |     554 |         NULL |        4180 | 2016-01-01  |
|      6444 |       2099 | saving   |        1 |   75000 |           42 |        1914 | 2016-01-01  |
|      9307 |       2467 | checking |        2 |     550 |         NULL |        2739 | 2016-01-01  |
|      9601 |       2099 | checking |        1 |     650 |           42 |        1914 | 2016-01-01  |
+-----------+------------+----------+----------+---------+--------------+-------------+-------------+
9 rows in set (0.00 sec)
```

```
mysql> call calculate_interest(9307,@bb);
    -> //
Query OK, 1 row affected (0.26 sec)
```

```
mysql> select * from account//
+-----------+------------+----------+----------+---------+--------------+-------------+-------------+
| AccountNo | customerId | Type     | branchNo | balance | creditNumber | DebitNumber | openingDate |
+-----------+------------+----------+----------+---------+--------------+-------------+-------------+
|       265 |       1233 | checking |        1 | 562.754 |         NULL |        3066 | 2016-01-01  |
|       874 |       2753 | checking |        2 | 583.495 |          335 |        4807 | 2016-01-01  |
|      4994 |       1233 | saving   |        1 | 54636.4 |         NULL |        3066 | 2016-01-01  |
|      5221 |       2467 | saving   |        2 |   55000 |         NULL |        2739 | 2016-01-01  |
|      5600 |       1479 | saving   |        2 |   40000 |         NULL |        4180 | 2016-01-01  |
|      6344 |       1479 | checking |        2 |     554 |         NULL |        4180 | 2016-01-01  |
|      6444 |       2099 | saving   |        1 |   75000 |           42 |        1914 | 2016-01-01  |
|      9307 |       2467 | checking |        2 | 583.495 |         NULL |        2739 | 2016-01-01  |
|      9601 |       2099 | checking |        1 |     650 |           42 |        1914 | 2016-01-01  |
+-----------+------------+----------+----------+---------+--------------+-------------+-------------+
9 rows in set (0.00 sec)
```

2. Procedure to increment salary of employee.

```
mysql> create procedure incr_sal_emp(in percent int, in id int)
    -> begin
    -> update employee
    -> set salary=salary + (salary * percent/100)
    -> where empid=id;
    -> end//
Query OK, 0 rows affected (0.68 sec)

mysql> select salary from employee where empid=2;
    -> //
+--------+
| salary |
+--------+
|  36750 |
+--------+
1 row in set (0.00 sec)

mysql> call incr_sal_emp(10,2)//
Query OK, 1 row affected (0.12 sec)

mysql> select salary from employee where empid=2//
+--------+
| salary |
+--------+
|  40425 |
+--------+
1 row in set (0.00 sec)

mysql>
```

3. Procedure to update balance on deposit

```
mysql> create procedure updatebal_depo(in accNumbr int)
    -> begin
    -> select deposit.deptAmt into @amt from deposit inner join account on deposit.accNo=account.accountNo;
    -> update account
    -> set balance = balance+ @amt
    -> where accountNo=accNumbr;
    -> end//
Query OK, 0 rows affected (0.10 sec)

mysql> select balance from account where accountNo=6344//
+---------+
| balance |
+---------+
|     560 |
+---------+
1 row in set (0.00 sec)

mysql> select * from deposit//
+-----------+---------+---------------------+-------+
| depositId | deptAmt | deposit_date        | accNo |
+-----------+---------+---------------------+-------+
|        12 |      10 | 2018-12-12 16:19:04 |  6344 |
+-----------+---------+---------------------+-------+
1 row in set (0.00 sec)

mysql> call updatebal_depo(6344)//
Query OK, 1 row affected (0.15 sec)

mysql> select balance from account where accountNo=6344//
+---------+
| balance |
+---------+
|     570 |
+---------+
1 row in set (0.00 sec)
```

4. Procedure to transfer money between both accounts

```
mysql> create procedure transfer_money(in AccountNum int,in c_customerid int,in Amount int )
    -> begin
    -> declare account_1 int;
    -> declare account_2 int;
    -> declare customer_id int;
    -> declare b_balance int;
    -> declare b_balance_2 int;
    ->
    ->
    ->
    -> select count(*) into account_1 from account where customerid=c_customerid and accountno!=accountnum;
    -> select count(*) into customer_id from customer where custid=c_customerid;
    -> if account_1 != 1 or customer_id!=1 then
    -> select 'Invalid Account number or Customer Id';
    -> end if;
    ->
    -> select balance into b_balance from account where accountno=accountnum;
    ->
    -> if amount>b_balance then
    -> select 'Insufficient balance';
    ->
    -> else
    -> select accountno into account_2 from account where customerid=c_customerid and accountno!=accountnum;
    -> select balance into b_balance_2 from account where accountno=account_2;
    -> Select concat('Old Account balance of ',accountnum, 'is ',b_balance);
    -> Select concat('Old Account balance of ',account_2, 'is ',b_balance_2);
    ->
    -> update account
    -> set balance=balance-amount
    -> where accountno=accountnum;
    ->
    -> update account
    -> set balance=balance+amount
    -> where customerid=c_customerid and accountno!=accountnum;
    -> end if;
    ->
    -> select balance into b_balance from account where accountno=accountnum;
    -> select balance into b_balance_2 from account where accountno=account_2;
   -> Select concat('New Account balance of ',accountnum, 'is ',b_balance);
   -> Select concat('New Account balance of ',account_2, 'is ',b_balance_2);
   ->
   -> end //
Query OK, 0 rows affected (0.21 sec)
```

```
mysql> call transfer_money(3523,2753,50);
    -> //
+-------------------------------------------------------------+
| concat('Old Account balance of ',accountnum, 'is ',b_balance) |
+-------------------------------------------------------------+
| Old Account balance of 3523is 500                           |
+-------------------------------------------------------------+
1 row in set (0.00 sec)

+-------------------------------------------------------------+
| concat('Old Account balance of ',account_2, 'is ',b_balance_2) |
+-------------------------------------------------------------+
| Old Account balance of 9665is 75050                         |
+-------------------------------------------------------------+
1 row in set (0.02 sec)

+-------------------------------------------------------------+
| concat('New Account balance of ',accountnum, 'is ',b_balance) |
+-------------------------------------------------------------+
| New Account balance of 3523is 450                           |
+-------------------------------------------------------------+
1 row in set (0.33 sec)

+-------------------------------------------------------------+
| concat('New Account balance of ',account_2, 'is ',b_balance_2) |
+-------------------------------------------------------------+
| New Account balance of 9665is 75100                         |
+-------------------------------------------------------------+
1 row in set (0.34 sec)

Query OK, 0 rows affected (0.35 sec)
```

VIEWS

1. View of Employee-Branch to see which employee belongs to which branch.

```
mysql> create view emp_branch as
    -> select employee.empname, branch.branchName from employee inner join branch
 where employee.branchid=branch.branchid;
    -> //
Query OK, 0 rows affected (1.32 sec)

mysql> select *  from emp_branch//
+----------------+-------------------+
| empname        | branchName        |
+----------------+-------------------+
| Ram Gupta      | Harvest st branch |
| Sham Sood      | Harvest st branch |
| Pranali Shetty | Harvest st branch |
| Shilpa Shetty  | Airport Branch    |
| Sayli Bhutkar  | Airport Branch    |
| Ashish Prabhu  | Airport Branch    |
| Gopal Krishna  | Mall Road Branch  |
| Kevin Thomas   | Mall Road Branch  |
+----------------+-------------------+
8 rows in set (0.15 sec)
```

SHRUTI WALAWALKAR

2. View to see card details of all employees

```
mysql> create view cust_card as
    -> select accountNo, customerid, concat_ws(',',creditno, creditcardname,expdate,c_cvv) as creditDetails,
 concat_ws(',',debitno,cardname,expirydate,cvv) as debitDetails from account a, credit c, debit d
    -> where a.creditnumber=c.creditno and
    -> a.debitnumber=d.debitno;
    -> //
Query OK, 0 rows affected (0.26 sec)

mysql> select * from cust_card//
+-----------+------------+-----------------------------------+-----------------------------------+
| accountNo | customerid | creditDetails                     | debitDetails                      |
+-----------+------------+-----------------------------------+-----------------------------------+
|      6444 |       2099 | 42,Amey Pai,2025-10-10,382        | 1914,Amey Pai,2025-05-13,687      |
|      9601 |       2099 | 42,Amey Pai,2025-10-10,382        | 1914,Amey Pai,2025-05-13,687      |
|       874 |       2753 | 335,Mayuri Khanolkar,2029-12-10,556 | 4807,Mayuri Khanolkar,2020-09-29,725 |
+-----------+------------+-----------------------------------+-----------------------------------+
3 rows in set (0.08 sec)
```

SCRIPT

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,E
RROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
-- -----------------------------------------------------

-- Schema bank

-- -----------------------------------------------------

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

USE `bank` ;

-- -----------------------------------------------------

-- Table `bank`.`customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`customer` (
  `custId` INT(11) NOT NULL,
  `firstName` VARCHAR(25) NULL DEFAULT NULL,
  `middleInitial` VARCHAR(25) NULL DEFAULT NULL,
  `lastName` VARCHAR(25) NULL DEFAULT NULL,
  `Email` VARCHAR(50) NULL DEFAULT NULL,
  `Phone` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`custId`))
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`address`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`address` (
`addressId` INT(11) NOT NULL AUTO_INCREMENT,
`apartmentNo` VARCHAR(25) NULL DEFAULT NULL,
`Street` CHAR(50) NULL DEFAULT NULL,
`City` CHAR(50) NULL DEFAULT NULL,
`Zipcode` VARCHAR(10) NULL DEFAULT NULL,
PRIMARY KEY (`addressId`))
ENGINE = InnoDB
AUTO_INCREMENT = 10
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`branch`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`branch` (
`BranchID` INT(11) NOT NULL,
`BranchName` VARCHAR(50) NULL DEFAULT NULL,
`AddressId` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`BranchID`),
INDEX `AddressId` (`AddressId` ASC) VISIBLE,
CONSTRAINT `branch_ibfk_1`
FOREIGN KEY (`AddressId`)
REFERENCES `bank`.`address` (`addressId`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`debit`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`debit` (
`DebitNo` INT(11) NOT NULL,
`CardName` VARCHAR(50) NULL DEFAULT NULL,
`ExpiryDate` DATE NULL DEFAULT NULL,
`CVV` INT(11) NOT NULL,
PRIMARY KEY (`DebitNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

SHRUTI WALAWALKAR

```
-- -----------------------------------------------------

-- Table `bank`.`credit`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`credit` (
`CreditNo` INT(11) NOT NULL,
`creditcardname` VARCHAR(50) NULL DEFAULT NULL,
`c_cvv` INT(11) NULL DEFAULT NULL,
`expdate` DATE NULL DEFAULT NULL,
PRIMARY KEY (`CreditNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`account`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`account` (
  `AccountNo` INT(11) NOT NULL,
  `customerId` INT(11) NULL DEFAULT NULL,
  `Type` VARCHAR(30) NULL DEFAULT NULL,
  `branchNo` INT(11) NULL DEFAULT NULL,
  `balance` FLOAT NULL DEFAULT NULL,
  `creditNumber` INT(11) NULL DEFAULT NULL,
  `DebitNumber` INT(11) NOT NULL,
  `openingDate` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`AccountNo`),
  INDEX `customerId` (`customerId` ASC) VISIBLE,
  INDEX `branchNo` (`branchNo` ASC) VISIBLE,
  INDEX `DebitNumber` (`DebitNumber` ASC) VISIBLE,
  INDEX `creditNumber` (`creditNumber` ASC) VISIBLE,
  CONSTRAINT `account_ibfk_1`
    FOREIGN KEY (`customerId`)
    REFERENCES `bank`.`customer` (`custId`),
  CONSTRAINT `account_ibfk_2`
    FOREIGN KEY (`branchNo`)
    REFERENCES `bank`.`branch` (`BranchID`),
  CONSTRAINT `account_ibfk_3`
    FOREIGN KEY (`DebitNumber`)
    REFERENCES `bank`.`debit` (`DebitNo`),
  CONSTRAINT `account_ibfk_4`
    FOREIGN KEY (`creditNumber`)
    REFERENCES `bank`.`credit` (`CreditNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

SHRUTI WALAWALKAR

```
-- -----------------------------------------------------

-- Table `bank`.`backup`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`backup` (
`deleteTime` DATETIME NOT NULL DEFAULT '0000-00-00 00:00:00',
`accountNo` INT(11) NOT NULL,
PRIMARY KEY (`accountNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`deposit`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`deposit` (
`depositId` INT(11) NOT NULL,
`deptAmt` INT(11) NULL DEFAULT NULL,
`deposit_date` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
`accNo` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`depositId`),
INDEX `accNo` (`accNo` ASC) VISIBLE,
CONSTRAINT `deposit_ibfk_1`
FOREIGN KEY (`accNo`)
REFERENCES `bank`.`account` (`AccountNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`employee`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`employee` (
`EmpId` INT(11) NOT NULL AUTO_INCREMENT,
`EmpName` VARCHAR(100) NULL DEFAULT NULL,
`BranchId` INT(11) NULL DEFAULT NULL,
`designation` VARCHAR(50) NULL DEFAULT NULL,
`Salary` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`EmpId`),
INDEX `BranchId` (`BranchId` ASC) VISIBLE,
CONSTRAINT `employee_ibfk_1`
FOREIGN KEY (`BranchId`)
REFERENCES `bank`.`branch` (`BranchID`))
ENGINE = InnoDB
AUTO_INCREMENT = 9
DEFAULT CHARACTER SET = utf8mb4
```

```
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`loan`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`loan` (
`loanId` INT(11) NOT NULL,
`branchNumber` INT(11) NULL DEFAULT NULL,
`loanAmount` INT(11) NULL DEFAULT NULL,
`borrowerId` INT(11) NULL DEFAULT NULL,
`CosignerId` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`loanId`),
INDEX `branchNumber` (`branchNumber` ASC) VISIBLE,
INDEX `borrowerId` (`borrowerId` ASC) VISIBLE,
INDEX `CosignerId` (`CosignerId` ASC) VISIBLE,
CONSTRAINT `loan_ibfk_1`
FOREIGN KEY (`branchNumber`)
REFERENCES `bank`.`branch` (`BranchID`),
CONSTRAINT `loan_ibfk_2`
FOREIGN KEY (`borrowerId`)
REFERENCES `bank`.`account` (`AccountNo`),
CONSTRAINT `loan_ibfk_3`
FOREIGN KEY (`CosignerId`)
REFERENCES `bank`.`account` (`AccountNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-- -----------------------------------------------------

-- Table `bank`.`withdraw`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`withdraw` (
`withdrawId` INT(11) NOT NULL,
`withAmt` INT(11) NULL DEFAULT NULL,
`withdraw_date` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
`accNo` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`withdrawId`),
INDEX `accNo` (`accNo` ASC) VISIBLE,
CONSTRAINT `withdraw_ibfk_1`
FOREIGN KEY (`accNo`)
REFERENCES `bank`.`account` (`AccountNo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

USE `bank` ;
```

SHRUTI WALAWALKAR

```
-- --------------------------------------------------------

-- Placeholder table for view `bank`.`cust_card`

-- --------------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`cust_card` (`accountNo` INT, `customerid` INT, `creditDetails`
INT, `debitDetails` INT);

-- --------------------------------------------------------

-- Placeholder table for view `bank`.`emp_branch`

-- --------------------------------------------------------

CREATE TABLE IF NOT EXISTS `bank`.`emp_branch` (`empname` INT, `branchName` INT);

-- --------------------------------------------------------

-- procedure calculate_interest

-- --------------------------------------------------------

DELIMITER $$
USE `bank`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `calculate_interest`(in a_account_id int, out
b_balance float)
begin
declare o_openingdate date;
declare diff1 int;
select balance,openingdate into b_balance,o_openingdate from account where
accountno=a_account_id;
if o_openingdate+interval 12 month<=curdate() then
SELECT DATEDIFF(curdate(), o_openingdate) into diff1;
set diff1 = (diff1/365);
repeat
update account
set balance = b_balance*1.03
where accountno=a_account_id;
set diff1=diff1-1;
until diff1>=0
end repeat;
end if;
end$$

DELIMITER ;

-- --------------------------------------------------------

-- procedure find_employeebranch

-- --------------------------------------------------------
```

SHRUTI WALAWALKAR

```
DELIMITER $$
USE `bank`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `find_employeebranch`(in id int)
begin
select employee.empid, branch.branchName from employee inner join branch where
employee.branchid=branch.branchid;
end$$

DELIMITER ;

-- -----------------------------------------------------

-- procedure get_accountNo

-- -----------------------------------------------------


DELIMITER $$

USE `bank`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `get_accountNo`(in customer_id int, out
account_no int)

begin

select accountNo into account_no

from account

where customerid=customer_id;

end$$


DELIMITER ;

-- -----------------------------------------------------

-- procedure incr_sal_emp

-- -----------------------------------------------------

DELIMITER $$
USE `bank`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `incr_sal_emp`(in percent int, in id int)
begin
update employee
set salary=salary + (salary * percent/100)
where empid=id;
end$$

DELIMITER ;
```

SHRUTI WALAWALKAR

```
-- -----------------------------------------------------

-- procedure transfer_money

-- -----------------------------------------------------

DELIMITER $$
USE `bank`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `transfer_money`(in AccountNum int,in
c_customerid int,in Amount int )
begin
declare account_1 int;
declare account_2 int;
declare customer_id int;
declare b_balance int;
declare b_balance_2 int;



select count(*) into account_1 from account where customerid=c_customerid and
accountno!=accountnum;
select count(*) into customer_id from customer where custid=c_customerid;
if account_1 != 1 or customer_id!=1 then
select 'Invalid Account number or Customer Id';
end if;

select balance into b_balance from account where accountno=accountnum;

if amount>b_balance then
select 'Insufficient balance';

else
select accountno into account_2 from account where customerid=c_customerid and
accountno!=accountnum;
select balance into b_balance_2 from account where accountno=account_2;
Select concat('Old Account balance of ',accountnum, 'is ',b_balance);
Select concat('Old Account balance of ',account_2, 'is ',b_balance_2);

update account

set balance=balance-amount

where accountno=accountnum;



update account
set balance=balance+amount
where customerid=c_customerid and accountno!=accountnum;
```

```
end if;

select balance into b_balance from account where accountno=accountnum;
select balance into b_balance_2 from account where accountno=account_2;


Select concat('New Account balance of ',accountnum, 'is ',b_balance);
Select concat('New Account balance of ',account_2, 'is ',b_balance_2);

end$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure update_onDeposit
-- ----------------------------------------------------

DELIMITER $$
USE `bank`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `update_onDeposit`(in accNumbr int, in dAmt int,
out accbal int)
begin
select balance  into accbal  from account where accountNo=accNumbr;
update account
set balance= accbal + dAmt
where accountNo=accNumbr;
end$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure updatebal_depo
-- ----------------------------------------------------

DELIMITER $$
USE `bank`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `updatebal_depo`(in accNumbr int)
begin
select deposit.deptAmt into @amt from deposit inner join account on
deposit.accNo=account.accountNo;
update account
set balance = balance+ @amt
where accountNo=accNumbr;
end$$

DELIMITER ;

-- ----------------------------------------------------
```

SHRUTI WALAWALKAR

```
-- View `bank`.`cust_card`
-- ------------------------------------------------------
DROP TABLE IF EXISTS `bank`.`cust_card`;
USE `bank`;
CREATE  OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER
VIEW `bank`.`cust_card` AS select `a`.`AccountNo` AS `accountNo`,`a`.`customerId` AS
`customerid`,concat_ws(',',`c`.`CreditNo`,`c`.`creditcardname`,`c`.`expdate`,`c`.`c_cvv`) AS
`creditDetails`,concat_ws(',',`d`.`DebitNo`,`d`.`CardName`,`d`.`ExpiryDate`,`d`.`CVV`) AS
`debitDetails` from ((`bank`.`account` `a` join `bank`.`credit` `c`) join `bank`.`debit` `d`) where
((`a`.`creditNumber` = `c`.`CreditNo`) and (`a`.`DebitNumber` = `d`.`DebitNo`));

-- ------------------------------------------------------
-- View `bank`.`emp_branch`
-- ------------------------------------------------------
DROP TABLE IF EXISTS `bank`.`emp_branch`;
USE `bank`;
CREATE  OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER
VIEW `bank`.`emp_branch` AS select `bank`.`employee`.`EmpName` AS
`empname`,`bank`.`branch`.`BranchName` AS `branchName` from (`bank`.`employee` join
`bank`.`branch`) where (`bank`.`employee`.`BranchId` = `bank`.`branch`.`BranchID`);
USE `bank`;

DELIMITER $$
USE `bank`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `bank`.`backup`
AFTER DELETE ON `bank`.`account`
FOR EACH ROW
begin
insert into backup values (now(),old.accountNo);
end$$

USE `bank`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `bank`.`checkvalidity`
BEFORE INSERT ON `bank`.`account`
FOR EACH ROW
begin
if new.accountNo in (
select account.accountNo from account where new.accountNo = account.accountNo)
then signal sqlstate '45000'

set message_text= 'account already exists';
end if;
end$$
```

SHRUTI WALAWALKAR

```
USE `bank`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `bank`.`updatebalance`
AFTER INSERT ON `bank`.`withdraw`
FOR EACH ROW
begin
select withdraw.withAmt into @amt from withdraw inner join account on
withdraw.accNo=account.accountNo;
update account
set balance=balance - @amt
where accountNo in (select accNo from withdraw);
end$$


DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```