DATABASE MANAGEMENT SYSTEMS CSC620

Online Banking Database Design

Group Name: Database 101

Chao Jiang
Ran Xu
Xuecen Zhang
Zihao Liu

Table of Contents

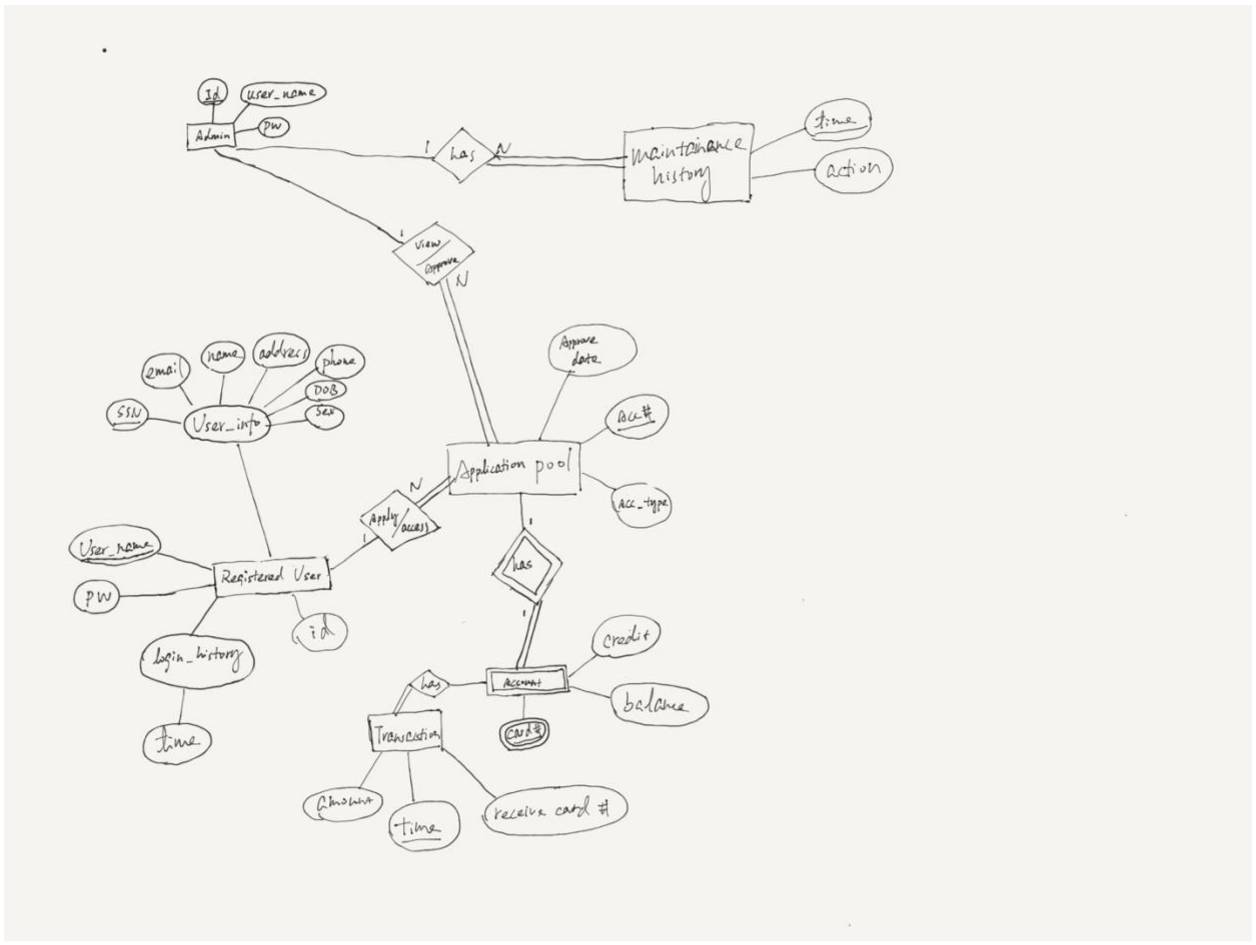
1. Overview

In this project, we are going to build up an online banking system. To achieve our goal, we divide our process into 3 steps:

1. Generate ER/EER diagram based on our database design;
2. Create MySQL database based on the design from first step;
3. Using Flask, Flask-SQLAlchemy framework to build up the server for user to access our database.

The purpose of this project is to propose a design and explore the feasibility of the design through prototyping.

2. Online Banking Database ER Diagram



3. Diagram Brief Description

Admin
The admin entity type is a type that stores administrator's login information such as user name and password.

Maintenance History
This is to record administrator's actions on registered user's password or account information. Each entity will be identified by admin's ID and edit time.

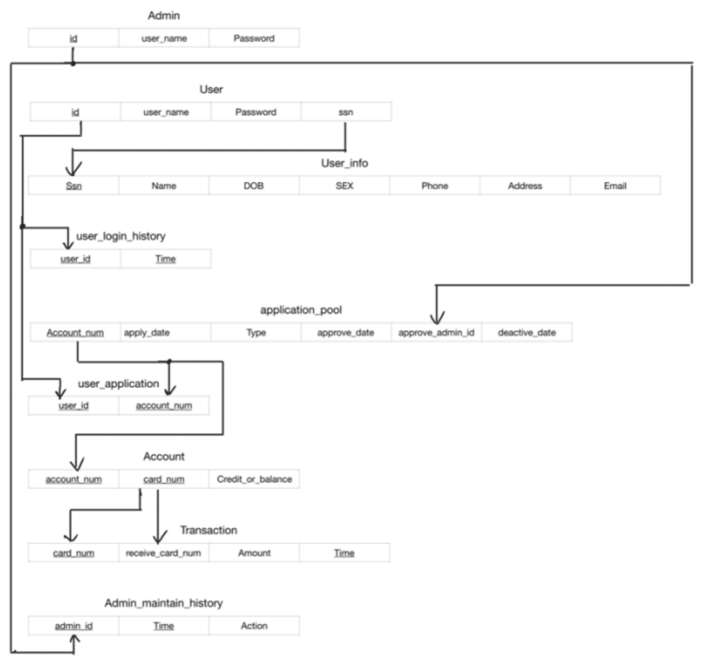
Registered User
Registered user is the entity to store customer's information such as his/her SSN, which will be the part of the primary key, address, date of birth, user name, password and so on.

Application Pool
This is the KEY entity type of this database.
Once the registered user applies for a debit or visa account, the application will be recorded in application pool with an approved-date set as null and an auto-generated account number. When one administrator accesses to this pool, he can view all applications. If the application hasn't been approved, he can choose to approve it by stamping a current date on approved-date attribute. Once the approved-date has a time stamp, the registered user can access his/her account. After that, every time the user login to his/her online banking account and tries to access his/her account, the system will check if there is a row that has his/her SSN and a time stamp in application pool. If so, he/she can access.
Note: when the administrator tries to delete the account, he/she will set the approved-date to 1900-01-01. In that case, the user won't be able to access the account any more. However, the administrator will not delete any data of this account in case for further reference.

Account
Each account will store the data such as the transaction and balance information. Once a user has a saving account setup, he/she can place an order or pay for the credit account if he/she has the credit account as well.

4. Relational Model

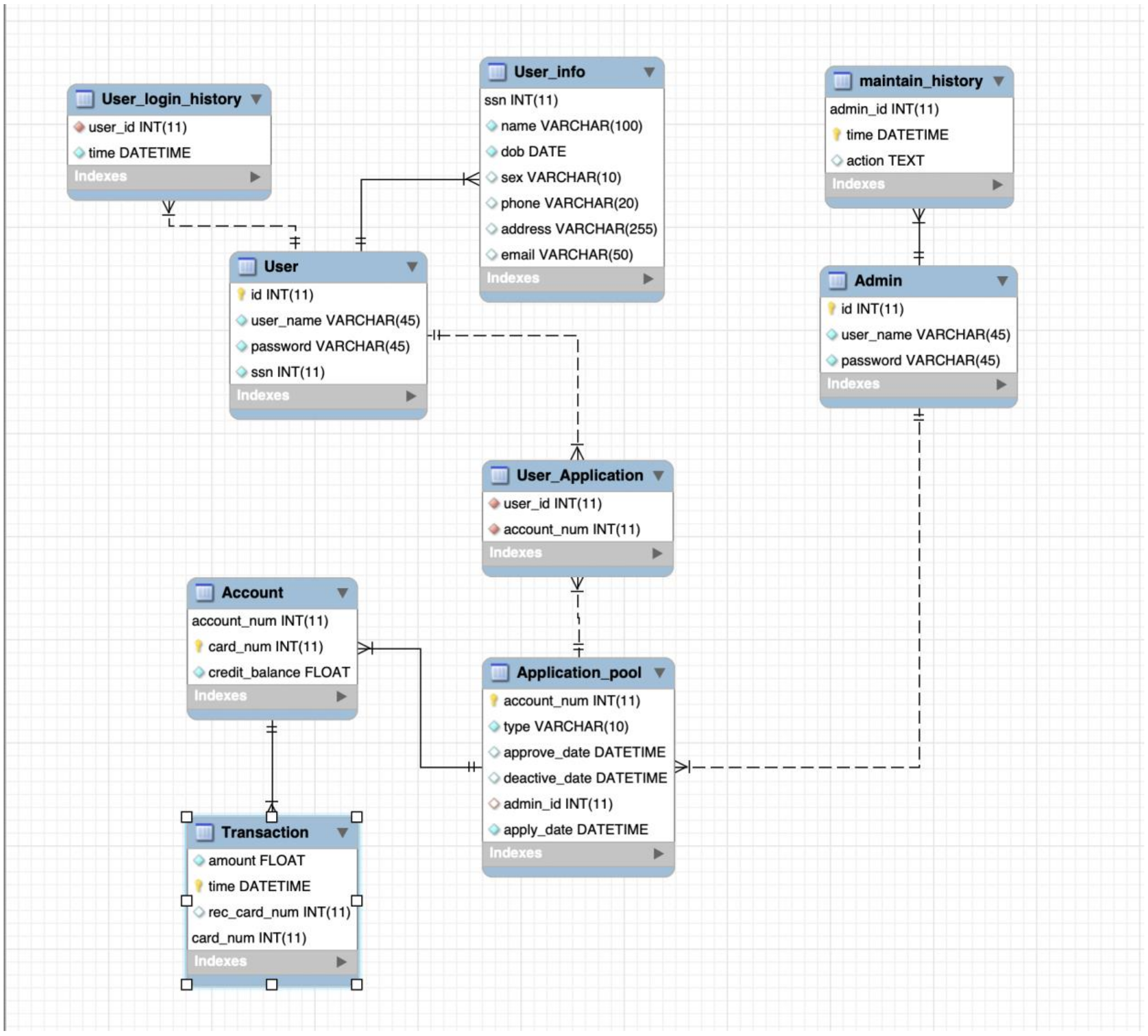
From Our ER diagram, we generate our relational model by following two main ideas. First, we convert entity with attributes to relation with fields. Second, we generate our foreign key based on the relationship in our ER diagram. Then we normalize our relational model up to 3NF so that we could minimize redundancy as well as minimize the insertion, deletion and update anomalies. Below is our final Relational Model.



1

5. EER Diagram in MySQL workbench

Below is our online bank system EER diagram generated on MySQL Workbranch.



6. SQL Statement

Below is the SQL statement to generate our online bank database.

```
CREATE SCHEMA IF NOT EXISTS `bank` DEFAULT CHARACTER SET utf8 ;
USE `bank` ;

-- -----------------------------------------------------
-- Table `bank`.`Admin`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`Admin` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `user_name` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `user_name_UNIQUE` (`user_name` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`Application_pool`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`Application_pool` (
  `account_num` INT(11) NOT NULL AUTO_INCREMENT,
  `type` VARCHAR(10) NOT NULL,
  `approve_date` DATETIME NULL DEFAULT NULL,
  `deactive_date` DATETIME NULL DEFAULT NULL,
  `admin_id` INT(11) NULL DEFAULT NULL,
  `apply_date` DATETIME NOT NULL,
  PRIMARY KEY (`account_num`),
  INDEX `admin_idx` (`admin_id` ASC),
  CONSTRAINT `admin`
    FOREIGN KEY (`admin_id`)
    REFERENCES `bank`.`Admin` (`id`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 22
DEFAULT CHARACTER SET = utf8;
```

```
-- -----------------------------------------------------
-- Table `bank`.`Account`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`Account` (
  `account_num` INT(11) NOT NULL,
  `card_num` INT(11) NOT NULL,
  `credit_balance` FLOAT NOT NULL,
  PRIMARY KEY (`account_num`, `card_num`),
  UNIQUE INDEX `card_num_UNIQUE` (`card_num` ASC),
  CONSTRAINT `acc_num`
    FOREIGN KEY (`account_num`)
    REFERENCES `bank`.`Application_pool` (`account_num`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`Transaction`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`Transaction` (
  `amount` FLOAT NOT NULL,
  `time` DATETIME NOT NULL,
  `rec_card_num` INT(11) NULL DEFAULT NULL,
  `card_num` INT(11) NOT NULL,
  PRIMARY KEY (`time`, `card_num`),
  INDEX `card_num_idx` (`card_num` ASC),
  CONSTRAINT `card`
    FOREIGN KEY (`card_num`)
    REFERENCES `bank`.`Account` (`card_num`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`User`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`User` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `user_name` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `ssn` INT(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `user_name_UNIQUE` (`user_name` ASC),
  UNIQUE INDEX `ssn_UNIQUE` (`ssn` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 13
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`User_Application`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`User_Application` (
  `user_id` INT(11) NOT NULL,
  `account_num` INT(11) NOT NULL,
  INDEX `userid_idx` (`user_id` ASC),
  INDEX `account_num_idx` (`account_num` ASC),
  CONSTRAINT `account_num`
    FOREIGN KEY (`account_num`)
    REFERENCES `bank`.`Application_pool` (`account_num`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `userid`
    FOREIGN KEY (`user_id`)
    REFERENCES `bank`.`User` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`User_info`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`User_info` (
  `ssn` INT(11) NOT NULL,
  `name` VARCHAR(100) NOT NULL,
  `dob` DATE NOT NULL,
  `sex` VARCHAR(10) NULL DEFAULT NULL,
  `phone` VARCHAR(20) NULL DEFAULT NULL,
  `address` VARCHAR(255) NULL DEFAULT NULL,
  `email` VARCHAR(50) NULL DEFAULT NULL,
  PRIMARY KEY (`ssn`),
  INDEX `ssn_idx` (`ssn` ASC),
  CONSTRAINT `ssn`
    FOREIGN KEY (`ssn`)
    REFERENCES `bank`.`User` (`ssn`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`User_login_history`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`User_login_history` (
  `user_id` INT(11) NOT NULL,
  `time` DATETIME NOT NULL,
  INDEX `user_id_idx` (`user_id` ASC),
  CONSTRAINT `user_id`
    FOREIGN KEY (`user_id`)
    REFERENCES `bank`.`User` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `bank`.`maintain_history`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bank`.`maintain_history` (
  `admin_id` INT(11) NOT NULL,
  `time` DATETIME NOT NULL,
  `action` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`admin_id`, `time`),
  CONSTRAINT `admin_id`
    FOREIGN KEY (`admin_id`)
    REFERENCES `bank`.`Admin` (`id`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

7. Summary

In this project, we first designed our database by generating ER diagram. Then we created the MySQL database based on our design. Finally, we learned and used Flask and Flask SQLAlchemy framework to create the server and user interface for customer accessing into the system to manage their account.

After the entire design and developing phase, we deeply understand what database is and how to design, create and query data from it. We also gained knowledge of how to connect our database and extract data from it by using Flask framework.

Appendix I

Individual Progress Report

Name: Chao Jiang

Group ID: Database 101

Reporting Period: 09/12/2018-12/07/2018

Date	Hours	Activity
09/21 – 10/12	10	Database design
10/12 – 11/12	10	Generate database in MySQL workbench
11/12 – 11/30	20	Build up server and user interface
12/01 – 12/07	5	Final report

Signature: Chao Jiang

Appendix II

Group Progress Report

Group ID: Database 101

Members: Chao Jiang, Ran Xu, Xuecen Zhang, Zihao Liu

Reporting Period: 09/12/2018-12/07/2018

Name	Total Hours in each Period	Total Hours (Cumulative)
Chao Jiang	15	45
Ran Xu	15	45
Xuecen Zhang	15	45
Zihao Liu	15	45
Group Totals	60	180

Appendix III

Contents of attached files:

1. Bank.mwb – To be used to do forward engining to export the schema design to a MySQL server;
2. Python code and html templates – To be used to build up server to connect to the database and user interface.

To run the app, you need to run **bankServer.py** file.