Loan Management Database System

Final Project Report

Group Number: 2

Project number: 7

Course: CIS 9340

Semester: Fall 2025

Name	Email
Christopher Chan	Christopher.Chan2@baruchmail.cuny.edu
Roshani Hada	Roshani.Hada@baruchmail.cuny.edu
Fabian Guzman	Fabian.Guzman@baruchmail.cuny.edu
Evan Kravitz	Evan.Kravitz@baruchmail.cuny.edu
Shilpa Koley	Shilpa.Koley@baruchmail.cuny.edu

Table of Contents

1. Introduction

For our final database project, we chose to design a system for a local bank that allows both individual and organizational customers to request and manage different types of loans. These loans can include auto loans, personal loans, home-repair loans, and business loans. No matter which type of loan the customer applies for, they must provide a collateral that covers at least 80% of the loan amount.

The process begins with the customer logging into the system, submitting a loan request, and attaching all the required documents. They also need to specify their collateral, which can be an account, a property, a car, or a life insurance policy. Certain collateral items like properties and vehicles must also include proof of insurance. After the request is submitted, a bank employee reviews the information, validates the collateral, and either approves or rejects the loan. If the loan is approved, the customer will see it on their dashboard and can start making payments toward it. If it is not approved, the customer will still be able to view the request along with the reason for denial. The system also keeps track of each customer's background information, including their annual income history, which helps the bank make better decisions.

Overall, this project focuses on creating a database that supports the entire loan process from the customer's initial request to collateral approval, loan management, and payment tracking while keeping the data organized, secure, and easy to access.

Why is the database system required for your organization?

A database system is required to keep all customer, loan, collateral, and payment information organized, accurate, and secure. It helps the bank manage the entire loan process efficiently, avoid mistakes, support faster decisions, and give customers a smooth way to request and track their loans.

The goal of this project is to build a reliable database that supports the bank's loan process from request to payment. The motivation is to improve efficiency, reduce errors, and give both customers and staff a simple, organized system to manage loans.

The system is expected to streamline the loan process, reduce errors, and improve accuracy in customer and loan records. It will help bank employees make faster decisions, give customers clearer access to their loan information, and provide a secure, organized way to manage all loan-related data.

2. Requirements & Business Overview

2.1 Business Description

The business is a local bank that offers loans to individuals and organizations. Customers submit loan requests, upload documents, and provide collateral, while bank employees review, approve, or reject these requests and track payments. The bank relies on accurate data such as customer details, income history, loan records, and payments to make decisions and manage daily operations efficiently.

2.2 Functional Requirements

List required features such as:

- Registration & Login: profile & KYC on file.
- Loan Application
- Collateral Submission
- Loan Review
- Loan Booking (if approved): amortization schedule generated
- Denial Visibility: Customers and certain bank staff see request & denial reasons
- Backups & replication

2.3 Non-Functional Requirements

- Security (encryption),
- Performance (Number of queries)
- Availability/Replication (99.9%)
- Scalability (Number of Individual/organization)
- Storage
- Budget constraints

3. Cost and Benefit Analysis

3.1 Costs

The estimated cost of developing a database system for a Bank's loan management project. The rational cost estimation depends on the software development, testing and deployment, maintenance of a secure database system, and server costs.

Breakdown of cost:

- Requirement analysis and design (1 week)
    - Business analysis and database architectures to define schemas and workflow and compliances needed.
    - Cost: $2000 (One analyst and one architect at $50/hour which is approximately 40 hours)
- Database Development: (Approximately 1 month)
    - Relational database (SQL) for customer data, loan details, collateral information, and transactional history. Includes schema for individuals or organizations, Loan type, collateral type, and payment tracking.
    - Security: Encryption for sensitive data.
    - Cost: Approximately $12,800 (One database developer, 80/hour which is approximately 160 hours (about 13 days) total)
- Document management system (2 weeks)
    - Integration of secure file storage system for documents like collateral proofs and incurrence records.
    - Cost: $6,400 (development and cloud storage setup)
- Quality analyst/ Testing (Approximately 1 month)
    - Functional testing (loan request, approval/denial workflows), security testing (penetration testing), and performance testing.

    - Cost: Approximately $12,800 (One QA tester, 80/hour which is approximately 160 hours)
- Compliance and Security (2-3 weeks)
    - Compliance with financial regulations and data protection laws.
    - Cost: Approximately $9,600 (If bank takes a third-party security service)
- Training and Documentation (1 week)
    - Training for bank staff and user guides for customers.
    - Cost: $2,000
- Cloud services setup and membership: $50 per month

Total Rational Cost:

- **Requirement Analysis and Design**: $2000
- **Database Development**: $12,800
- **Document Management System**: $6,400
- **Quality Analyst/Testing**: $12,800
- **Compliance and Security**: $9,600
- **Training and Documentation**: $2,000
- **Cloud services setup**: $50 per month

Total Rational Cost Calculation: The total rational cost is approximately **$45,600 + $50 per month**

Timeline: The total estimated timeline would be approximately 3 months.

3.2 Benefits

- **Efficiency & Productivity**: Automated workflows (loan application, collateral verification, payment tracking) reduce manual paperwork and staff overhead.
- **Regulatory Compliance**: Built-in checks (collateral coverage, insurance validation, audit logs) minimize risk of fraud and regulatory violations.

- **Better Risk Management**: Access to structured income histories and collateral data improves underwriting decisions, reducing defaults.
- Individual **Satisfaction & Retention**: Transparent and faster services improve the bank's reputation and competitive edge.
- **Scalability**: Supports growth in Individual base without requiring proportional increases in staff.
- **Data Insights**: Analytics on loan trends, repayment patterns, and portfolio risks help in strategic planning.

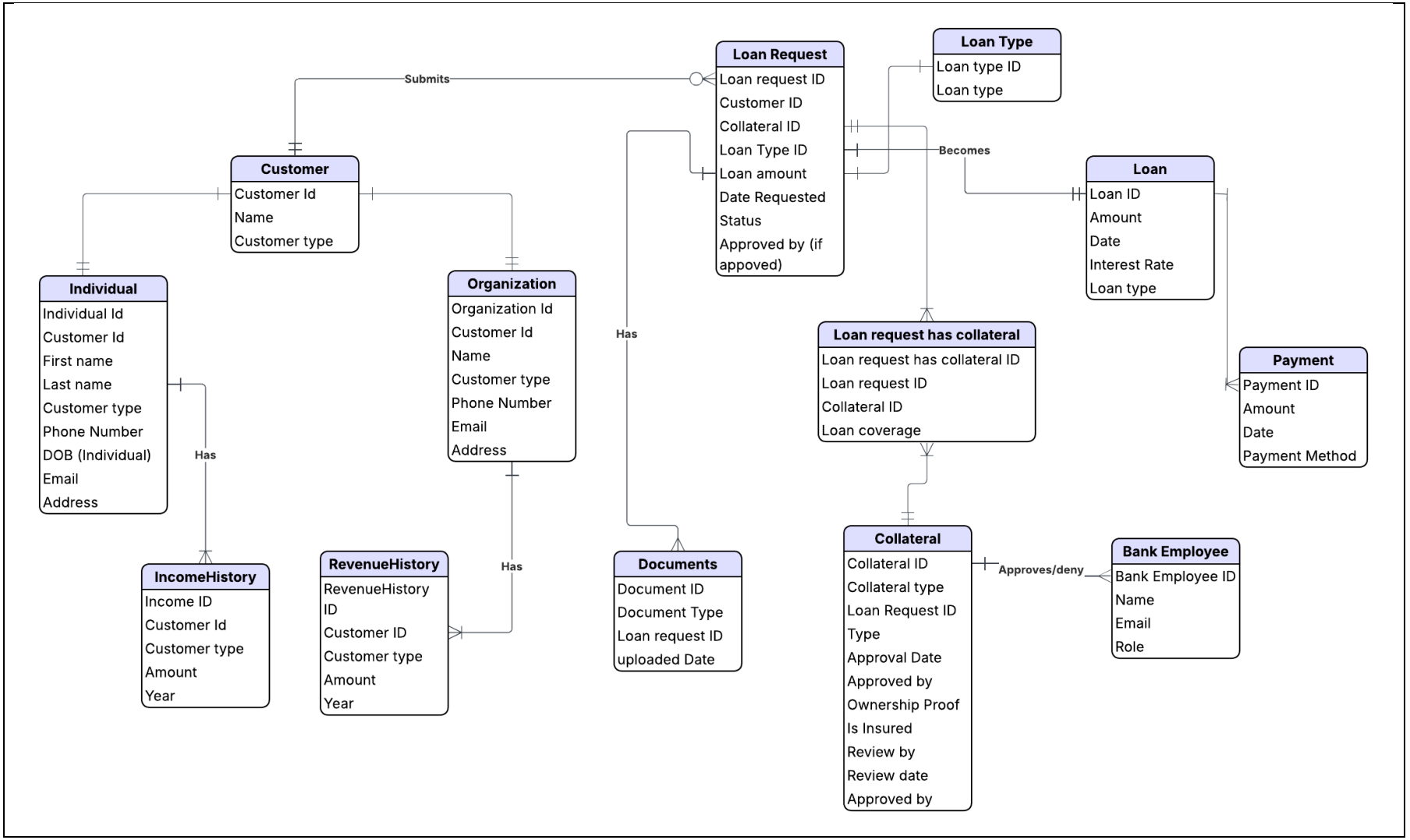
Challenges

- **Initial Investment**: Resource cost, training, and infrastructure setup.
- **Maintenance & Security**: Ongoing costs for IT support, database maintenance, and cybersecurity safeguards.
- **Integration with Legacy Systems**: Existing systems may require costly upgrades or connectors.

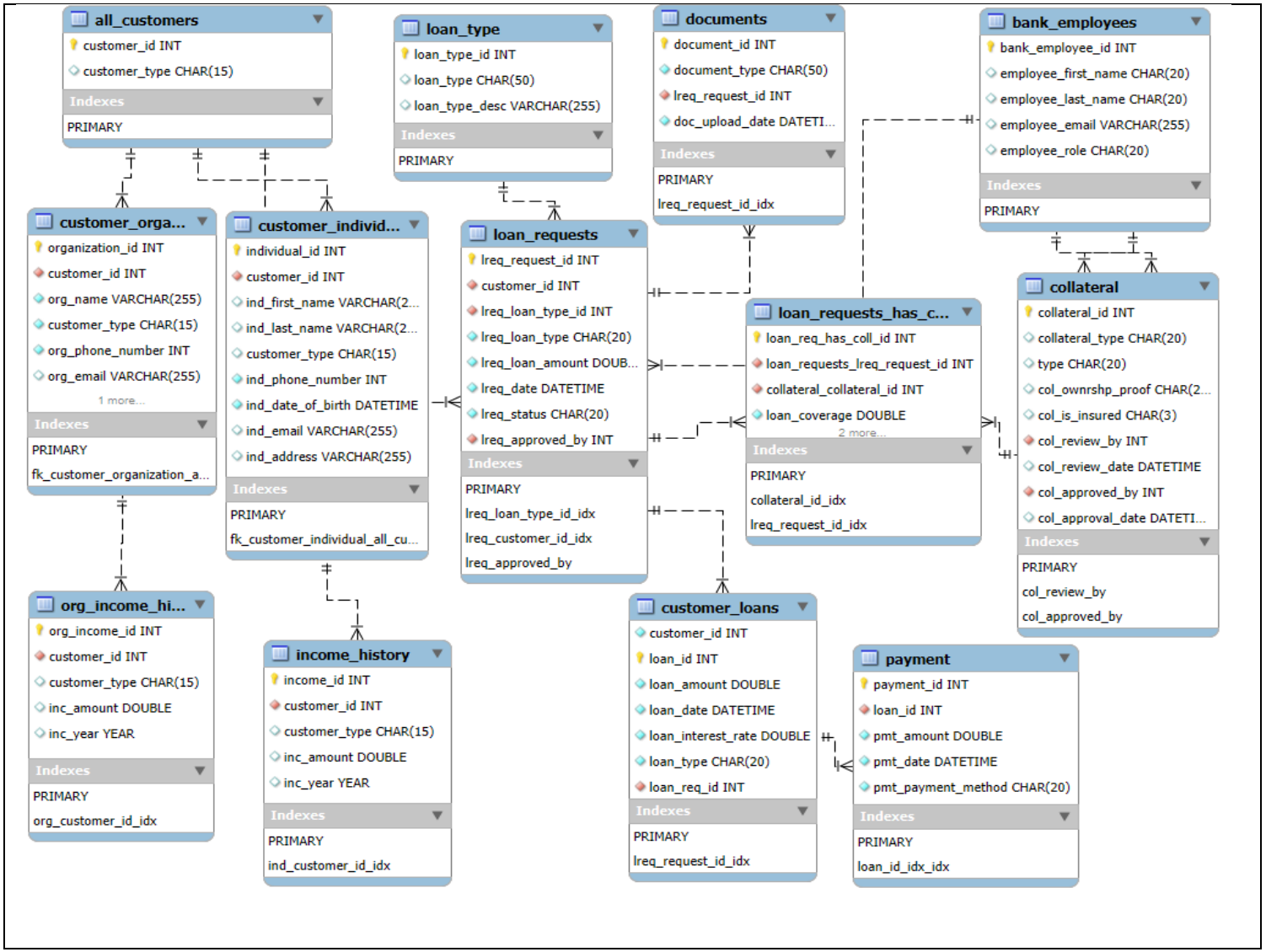
3.3 ROI Summary

The investment is justified because the system improves efficiency, reduces manual errors, speeds up loan processing, and gives customers a smoother experience. By automating key tasks and organizing all loan data in one place, the bank saves time, lowers operational costs, and can serve more customers with better accuracy and reliability.

4. Conceptual Diagram



5. Logical/Physical Diagram



6. Normalization, Relations

6.1 Normalized Relations

- Customer(CustomerID, FirstName, LastName, Phone, Email)
- Product(ProductID, ProductName, UnitPrice, SupplierID)
- Order(OrderID, CustomerID, OrderDate, StatusID)
- OrderDetails(OrderID, ProductID, Quantity, UnitPrice, Discount)

6.2 Normal Forms Achieved

All of our tables achieved 3NF. We removed partial and transitive dependencies by separating into multiple tables and creating new ID's. No denormalization was done for performance or reporting and we do not think this would be a good strategy. If needed, a reporting layer could be built with some denormalization on top of the base normalized tables

7. Functional Dependencies

7.1. all_customers(customer_id, customer_type)

- **customer_id --> customer_type**

7.2. customer_individual(individual_id, customer_id, ind_first_name, ind_last_name, customer_type, ind_phone_number, ind_date_of_birth, ind_email, ind_address)

- **individual_id → customer_id, ind_first_name, ind_last_name, customer_type, ind_phone_number, ind_date_of_birth, ind_email, ind_address**

- Often also: **customer_id → individual_id, ind_first_name, ind_last_name, customer_type, ind_phone_number, ind_date_of_birth, ind_email, ind_address**
- **customer_id --> customer_type**

7.3. customer_org(organization_id, customer_id, org_name, customer_type, org_phone_number, …)

- **organization_id → customer_id, org_name, customer_type, org_phone_number, …**
- Often also: **customer_id --> organization_id, org_name, customer_type, org_phone_number, …**
- all_customers: **customer_id --> customer_type**

7.4. loan_type(loan_type_id, loan_type_name, …)

- **loan_type_id --> (all other loan_type attributes)**
- Usually also: **loan_type_name --> (all other loan_type attributes)**

7.5. documents (document_id, …)

- **document_id --> (all other document attributes, including the related request/loan id)**

7.6. bank_employees(bank_employee_id, employee_first_name, employee_last_name, employee_email, employee_role)

- **bank_employee_id --> employee_first_name, employee_last_name, employee_email, employee_role**
- Often also: **employee_email --> bank_employee_id, employee_first_name, employee_last_name, employee_role**

7.7. loan_requests(lreq_request_id, customer_id, lreq_loan_type_id, lreq_loan_type, lreq_loan_amount, lreq_date, lreq_status, lreq_approved_by)

- **lreq_request_id --> customer_id, lreq_loan_type_id, lreq_loan_type, lreq_loan_amount, lreq_date, lreq_status, lreq_approved_by**
- From reference tables:
    - **lreq_loan_type_id --> lreq_loan_type** (via loan_type)
    - **customer_id --> customer_type** (via all_customers)

7.8. loan_requests_has_collateral(loan_req_has_coll_id, lreq_request_id, collateral_id, …)

- **loan_req_has_coll_id --> lreq_request_id, collateral_id, …**
- Usually also: **(lreq_request_id, collateral_id) --> loan_req_has_coll_id, …** (a given loan request + collateral pair is unique)

7.9. collateral(collateral_id, collateral_type_id, type, col_ownshp_proof, col_is_insured, col_review_by, col_review_date, col_approved_by, col_approval_date)

- **collateral_id --> collateral_type_id, type, col_ownshp_proof, col_is_insured, col_review_by, col_review_date, col_approved_by, col_approval_date**

7.10. income_history(income_id, customer_id, customer_type, inc_amount, inc_year)

- **income_id --> customer_id, customer_type, inc_amount, inc_year**
- Business-wise you also expect:
    - **(customer_id, inc_year) --> inc_amount, customer_type**

7.11. org_income_history(org_income_id, customer_id, customer_type, inc_amount, inc_year)

- **org_income_id --> customer_id, customer_type, inc_amount, inc_year**
- And similarly:
    - **(customer_id, inc_year) --> inc_amount, customer_type**

7.12. customer_loans(customer_id, loan_id, loan_amount, loan_date, loan_interest_rate, loan_type, loan_req_id)

- **loan_id --> customer_id, loan_amount, loan_date, loan_interest_rate, loan_type, loan_req_id**
- If each request can generate at most one loan:
    - **loan_req_id --> loan_id, customer_id, loan_amount, loan_date, loan_interest_rate, loan_type**

7.13. payment(payment_id, loan_id, pmt_amount, pmt_date, pmt_payment_method)

- **payment_id --> loan_id, pmt_amount, pmt_date, pmt_payment_method**
- Often also in business terms:
    - **(loan_id, pmt_date) --> pmt_amount, pmt_payment_method**

8. SQL DDL & Data Entry Methods

8.1 Data Entry Methods and DDL Examples

Data entry was initially performed by first exporting the conceptual mode from LucidChart to a csv document, which was adjusted in MS Word for table creation use in MySQL. However, this script was not meant to reflect the end state. The script was meant to serve as a good starting point for creating the bulk of the required tables and defining the Primary Keys and Foreign Keys within them. Manual adjustments had to be implemented directly to the model via updates to the schema through the MySQL GUI or through ad-hoc SQL queries.

Indexes were automatically assigned by MySQL when Primary Keys were defined in the table creation script.

- AcmeBankloandb SQL Table Creation INSERT sample script:

```
CREATE TABLE all_customers (
  customer_id INT NOT NULL,
  customer_type CHAR (1) NOT NULL,
  PRIMARY KEY (customer_id) );


CREATE TABLE  loan_requests (
  lreq_request_id INT NOT NULL,
  customer_id INT NOT NULL,
  collateral_id INT,
  lreq_loan_type CHAR(10),
```

```
  lreq_loan_amount DOUBLE,
  lreq_date  DATETIME NOT NULL,
  lreq_status CHAR(10) NOT NULL,
  lreq_approved_by INT,
  PRIMARY KEY (lreq_request_id),
  FOREIGN KEY (customer_id) REFERENCES all_customers (customer_id),
  FOREIGN KEY (collateral_id) REFERENCES collateral (collateral_id),
  FOREIGN KEY (lreq_approved_by) REFERENCES bank_employees
(bank_employee_id) );
CREATE TABLE collateral (
  collateral_id INT NOT NULL,
  collateral_type CHAR (10),
  lreq_request_id INT NOT NULL,
  type CHAR(10),
  col_ownrshp_proof CHAR (20),
  col_is_insured CHAR(1),
  col_review_by INT,
  col_review_date DATETIME,
  col_approved_by INT,
  col_approval_date DATETIME,

PRIMARY KEY (collateral_id),
FOREIGN KEY (lreq_request_id) REFERENCES loan_requests (lreq_request_id),
```

```
FOREIGN KEY (col_review_by) REFERENCES bank_employees
(bank_employee_id),
FOREIGN KEY (col_approved_by) REFERENCES bank_employees
(bank_employee_id) );

CREATE TABLE bank_employees (
  bank_employee_id INT NOT NULL,
  employee_first_name CHAR (20),
employee_last_name CHAR (20),
  employee_email CHAR (20),
  employee_role CHAR (20),
  PRIMARY KEY (bank_employee_id) );

CREATE TABLE customer_organization (
  organization_id INT NOT NULL,
  customer_id INT NOT NULL,
  org_name CHAR (30) NOT NULL,
  customer_type CHAR (1) NOT NULL,
  org_phone_number INT NOT NULL,
  org_email CHAR (20),
  org_address CHAR (50) NOT NULL,
   PRIMARY KEY (organization_id),
   FOREIGN KEY (customer_id ) REFERENCES all_customers (customer_id) );
```

```
CREATE TABLE income_history (
  income_id INT NOT NULL,
  customer_id INT NOT NULL,
  customer_type CHAR (1),
  inc_amount DOUBLE,
  inc_year YEAR,
   PRIMARY KEY (income_id),
   FOREIGN KEY (customer_id) REFERENCES all_customers (customer_id) );

CREATE TABLE customer_individual (
  individual_id INT NOT NULL,
  customer_id INT NOT NULL,
  ind_first_name CHAR (20),
  ind_last_name CHAR (20),
  customer_type CHAR (1),
  ind_phone_number INT NOT NULL,
  ind_date_of_birth DATETIME NOT NULL,
  ind_email CHAR (20),
  ind_address CHAR (50),
   PRIMARY KEY (individual_id),
  FOREIGN KEY (customer_id) REFERENCES all_customers (customer_id));
  CREATE TABLE customer_loans (
```

```
loan_id INT NOT NULL,
customer_id INT NOT NULL,
loan_amount DOUBLE NOT NULL,
loan_date DATETIME NOT NULL,
loan_interest_rate DOUBLE NOT NULL,
loan_type CHAR (20) NOT NULL,
PRIMARY KEY (loan_id),
FOREIGN KEY (customer_id) REFERENCES all_customers(customer_id) );

CREATE TABLE payment (
payment_id INT NOT NULL,
loan_id INT NOT NULL,
pmt_amount DOUBLE NOT NULL,
pmt_date DATETIME NOT NULL,
pmt_payment_method CHAR (20) NOT NULL,
 PRIMARY KEY (payment_id),
 FOREIGN KEY (loan_id ) REFERENCES customer_loans (loan_id));

CREATE TABLE documents (
document_id INT NOT NULL,
collateral_id INT NOT NULL,
document_type CHAR (20),
lreq_request_id INT NOT NULL,
doc_upload_date DATETIME NOT NULL,
PRIMARY KEY (document_id),
```

```
  FOREIGN KEY (collateral_id) REFERENCES collateral (collateral_id),
  FOREIGN KEY (lreq_request_id) REFERENCES loan_requests (lreq_request_id));

COMMIT;
```

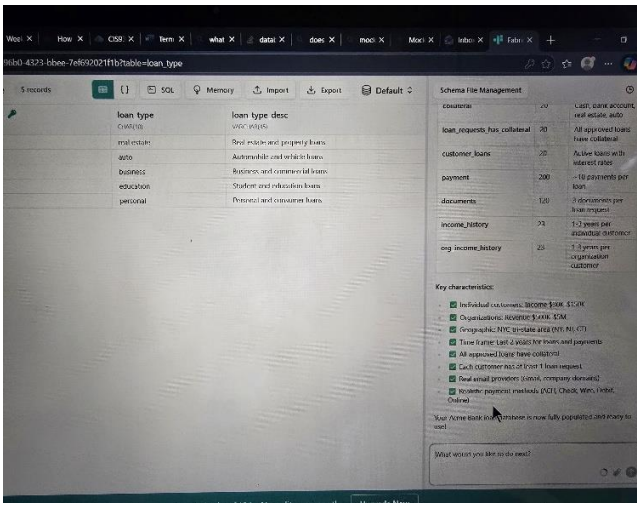
- Mockaroo Data Import

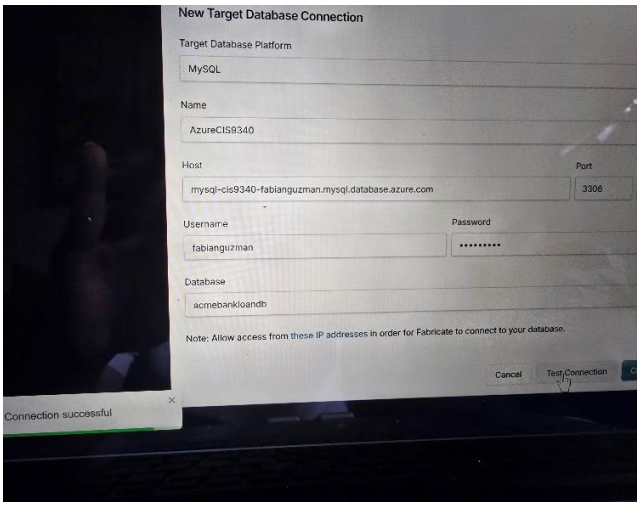
  Listed below is a URL link to the file with an export of the data that was imported directly from Mockaroo to the Acmebankloandb in MySQL via the Mockaroo interface. A URL link for a file documenting the prompting of AI to generate the Mockaroo sample data is also listed below.

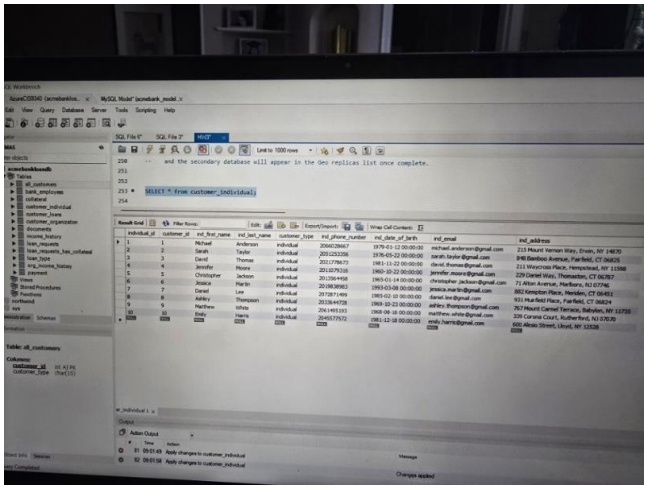
  - URL: Mockaroo data export for MySQL
  - URL: Prompting Mockaroo AI for sample data generation

The Acmebankloandb database in MySQL was populated directly from Mockaroo using its functionality to link directly to the db. (pics attached)

- MySQL Workbench Data Import Wizard







9. Forms, Reports, Queries & Navigation Structure

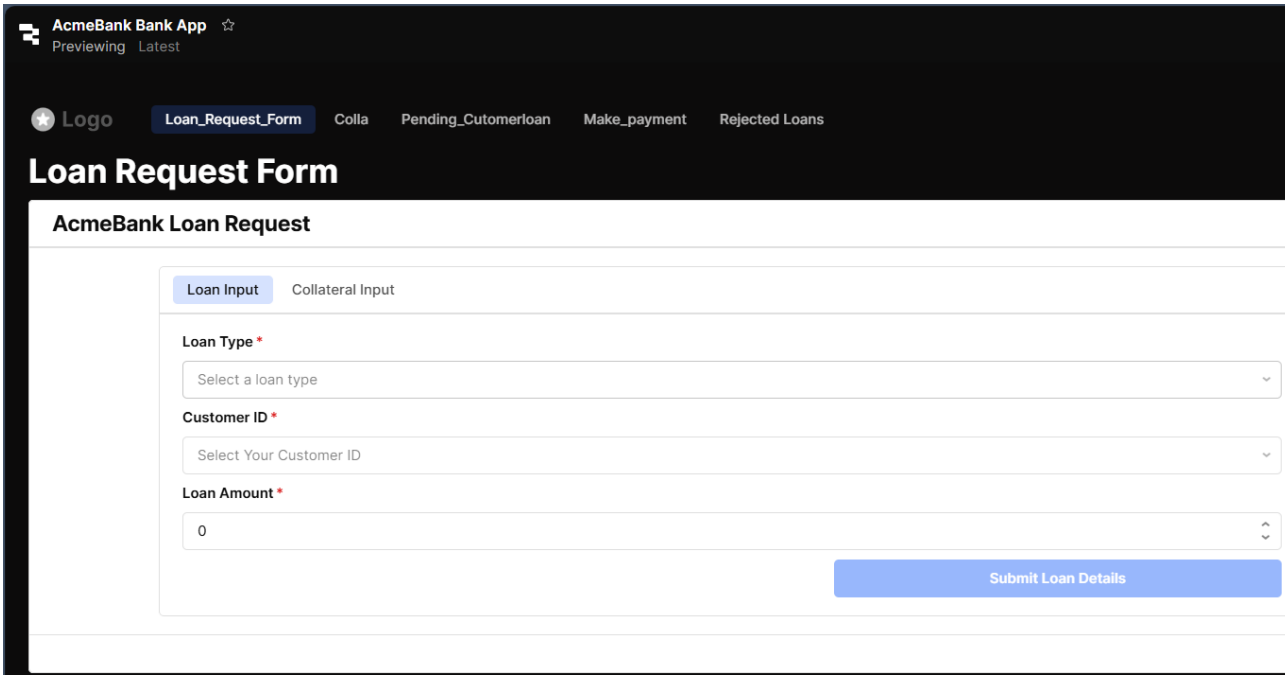
9.1 Loan Request Form and Related Queries

The AcmeBank Bank App has 5 forms, Loan Request Form, Colla, Pending_Customerloan, Make_payment, and Rejected Loans, which on the app's front end are reflected as tabs.

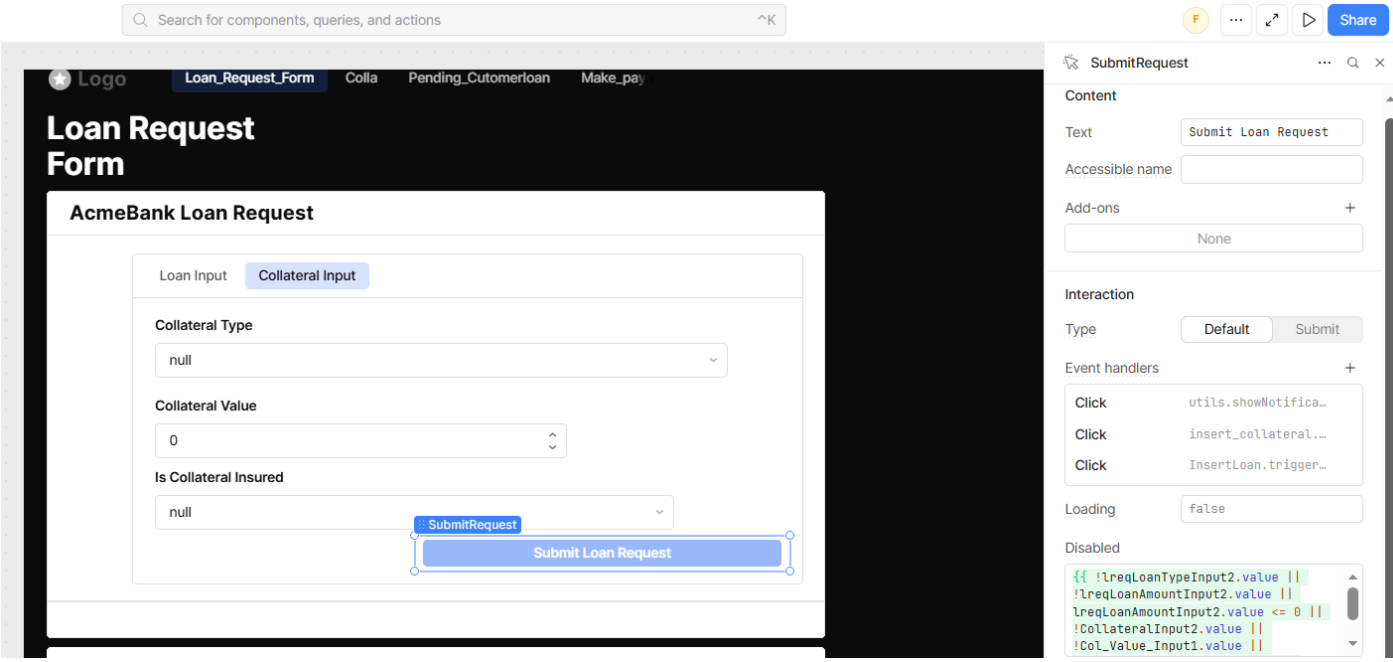
Loan Request Form

9.1.1 Tabbed Component - Loan Input Tab

The Loan request form in the applications provides users with the ability to create a loan request and provide collateral details using two tabs. The form provides a Loan Input tab where, via a dropdown, one can select the loan type and customer id. It also has a field where a user can enter a loan amount. Once these loan related inputs are entered, a submit button is enabled for the user to submit the loan details. Once the loan details are submitted, the tab view changes focus from the Loan Input tab to the Collateral Input tab.

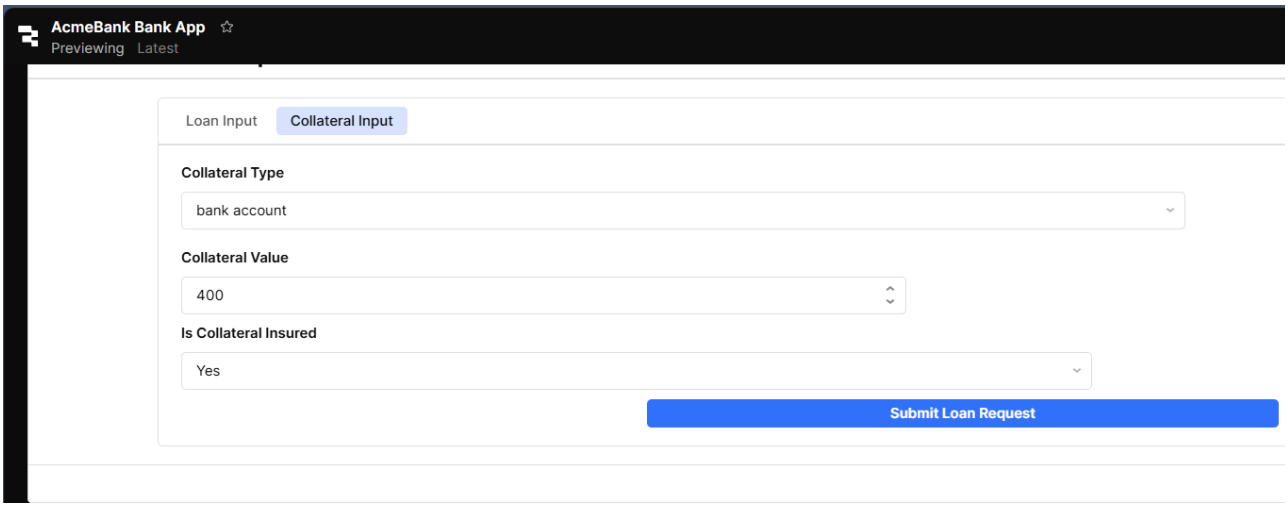


‘Submit Loan Request’ Retool constraints located on the lower right-hand corner



9.1.2 Tabbed Component - Collateral Input Tab

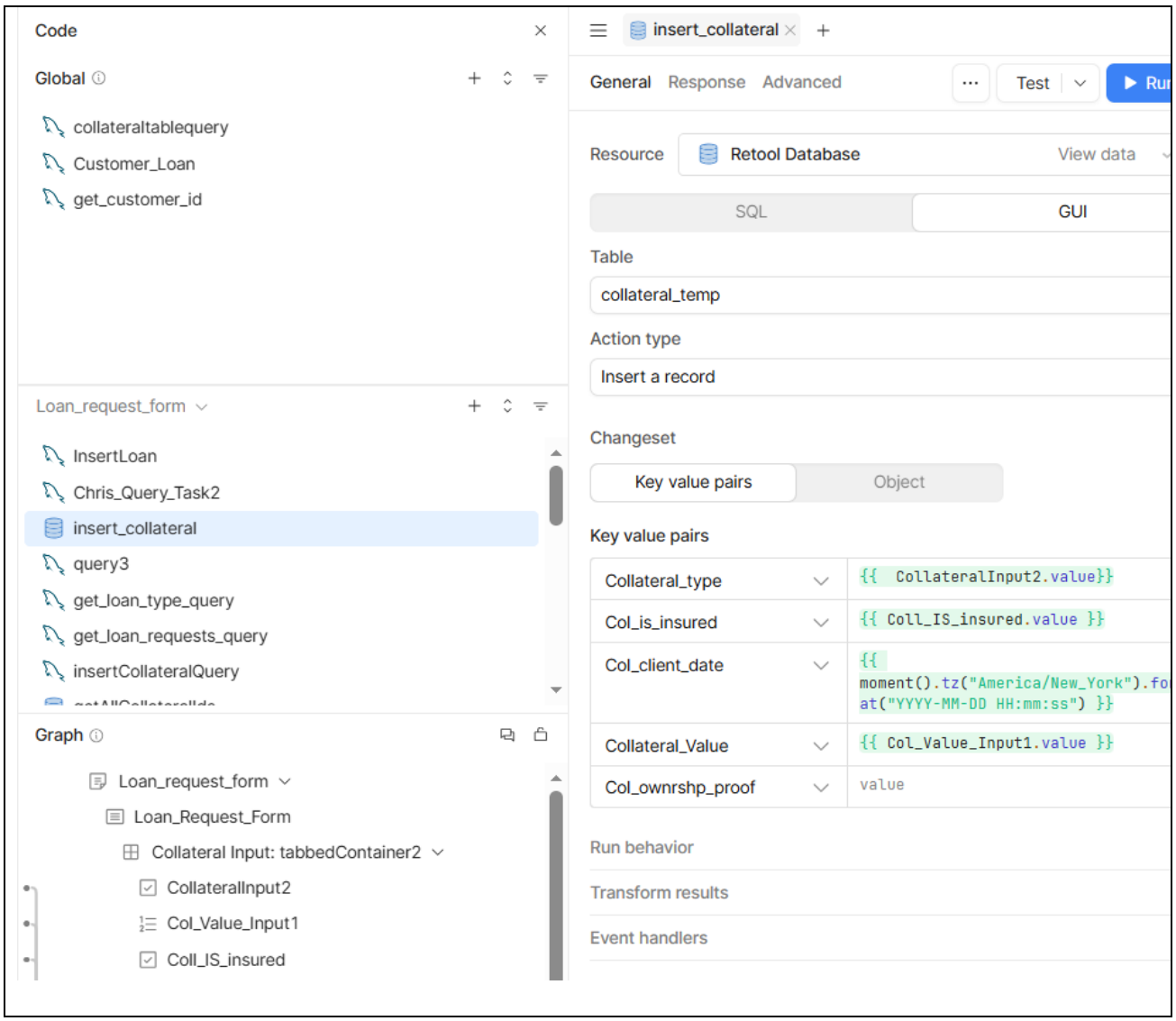
In the Collateral Input tab, the user must select the collateral type, enter a collateral value, and specify if the collateral is insured. When the collateral details are entered, the Submit Loan Request button is enabled, and the user can click on it to submit the loan request.



9.1.3 Loan request insertion SQL to MySQL – Creation loan request prior to approval

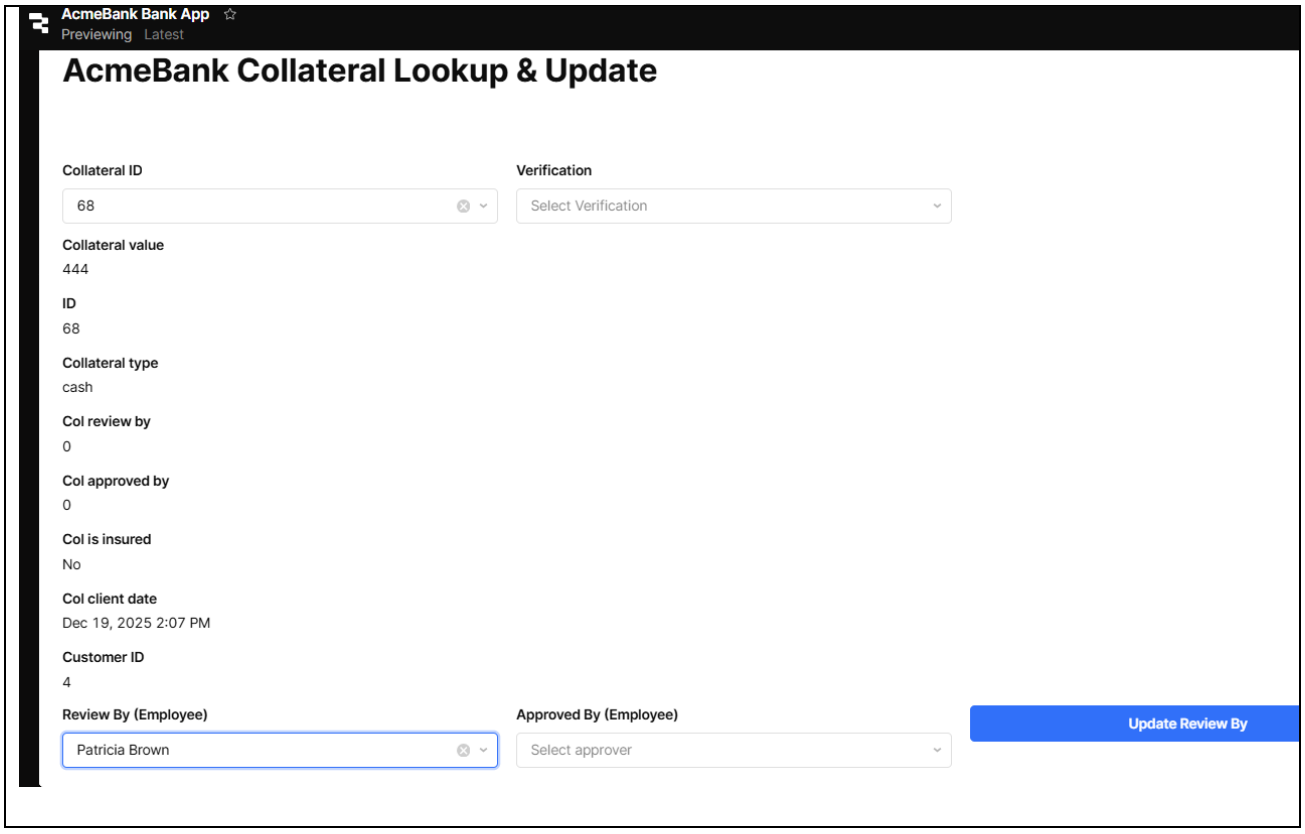
```
INSERT INTO acmebankloandb.loan_requests (
  customer_id,
  lreq_loan_type_id,
  lreq_loan_type,
  lreq_loan_amount,
  lreq_date,
  lreq_status,
  lreq_approved_by
)
VALUES (
  {{ customerID.value }},
  {{ lreqLoanTypeInput2.selectedItem.loan_type_id}},
{{ lreqLoanTypeInput2.value}},
  {{ lreqLoanAmountInput2.value }},
  {{ moment().tz("America/New_York").format("YYYY-MM-DD HH:mm:ss") }},
    'pending',
  1);
COMMIT;
```

9.1.4 Unapproved Collateral (temp table) insertion to Retool DB was created with Retool GUI



9.2 Component: AcmeBank Collateral Lookup & Update

The AcmeBank Collateral Lookup & Update component of the Loan Request form provides staff with a way to select unapproved collateral using an id saved in a Retool db temp table for unapproved collateral. Once the collateral details are visible on the screen, the user can select a dropdown with the name of a reviewer, which changes the button name to reflect that the collateral reviewer can be saved. At this point, a user can also save the approver nam, which also changes the button to indicate that the user can save Reviewer and Approver. Clicking on the button to 'Update Reviewer and Approver' triggers a series of queries to timestamp the update action and save the approved collateral and related details to another table that serves to link loan requests to collateral in the MySQL database, acmebankloandb.



9.2.1 Collateral SQL to insert data to MySQL table – Post collateral review and approval

```
INSERT INTO
  acmebankloandb.collateral (collateral_type,
    type,
    col_ownrshp_proof,
    col_is_insured,
    col_review_by,
    col_review_date,
    col_approved_by,
    col_approval_date
  )
VALUES
  (    {{ collateralDetailsDisplay.data.Collateral_type }},
    {{ collateralDetailsDisplay.data.Collateral_type }},
    {{ collateralDetailsDisplay.data.Col_ownrshp_proof }},
    {{ collateralDetailsDisplay.data.Col_is_insured}},
    {{ colReviewBySelect.value }},
    {{  moment().tz('America/New_York').format('YYYY-MM-DD HH:mm:ss')}},
    {{colApprovedBySelect.value }},
  {{  moment().tz('America/New_York').format('YYYY-MM-DD HH:mm:ss')}}
  );
COMMIT;
```

9.2.2 Loan has collateral insertion SQL – Post collateral review and approval

```
INSERT INTO acmebankloandb.loan_requests_has_collateral
(loan_requests_lreq_request_id,
collateral_collateral_id,
loan_coverage,
 collateral_col_review_by,
 collateral_col_approved_by
 )
 VALUES
({{ get_max_lreq_request_id.data.max_lreq_id }},
{{ get_max_collateral_id.data.max_coll_id }},
{{getCollateralById.data.Collateral_Value/get_coverage.data.lreq_loan_amount}},
COALESCE({{collateralDetailsDisplay.data.Col_review_by === 0 ? null :
collateralDetailsDisplay.data.Col_review_by}}, {{  colReviewBySelect.value}}),
{{colApprovedBySelect.value}});
COMMIT;
```

9.3 Pending_Customerloan

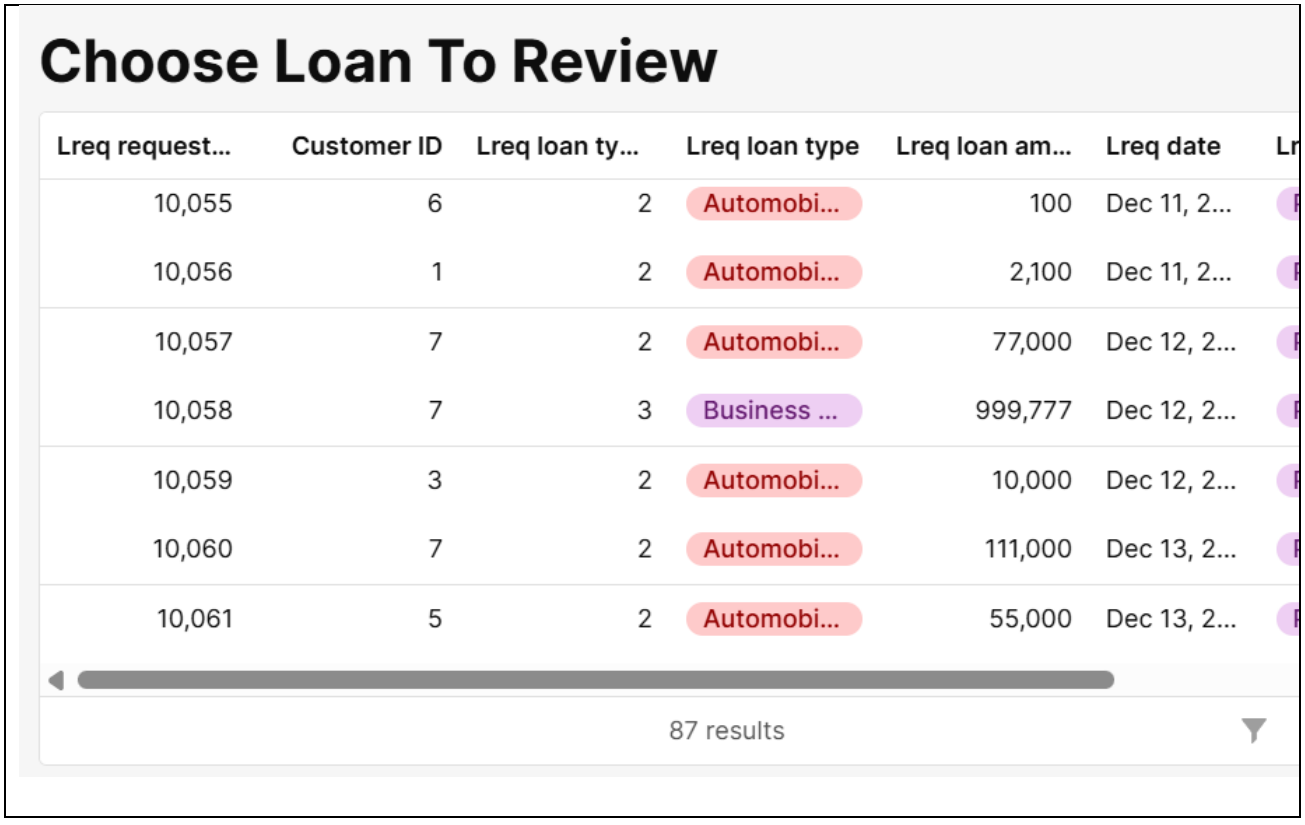
Queries on This Page:

9.3.1. To get unapproved loans

1st Fetches all pending loan requests

```
SELECT * FROM acmebankloandb.loan_requests
 WHERE lreq_status = 'PENDING'
```

This is how table 5 is created.



Choose Loan To Review

Lreq request...	Customer ID	Lreq loan ty...	Lreq loan type	Lreq loan am...	Lreq date	Lr
10,055	6	2	Automobi...	100	Dec 11, 2...	
10,056	1	2	Automobi...	2,100	Dec 11, 2...	
10,057	7	2	Automobi...	77,000	Dec 12, 2...	
10,058	7	3	Business ...	999,777	Dec 12, 2...	
10,059	3	2	Automobi...	10,000	Dec 12, 2...	
10,060	7	2	Automobi...	111,000	Dec 13, 2...	
10,061	5	2	Automobi...	55,000	Dec 13, 2...	

87 results

9.3.2. getBankEmployeesForPendingPage

Here we Fetch bank employees for the approver dropdown

```
SELECT bank_employee_id,
    CONCAT(employee_first_name, ' ', employee_last_name) AS employee_name
FROM acmebankloandb.bank_employees
ORDER BY employee_last_name ASC
```

9.3.3. updateLoanRequest

To Updates loan status to APPROVED and sets approver

```
UPDATE acmebankloandb.loan_requests
SET lreq_status = 'approved',
   lreq_approved_by = [selected approver ID]
WHERE lreq_request_id = [selected loan ID]
```

9.4.4. insertCustomerLoan

To Inserts approved loan into customer_loans table

```
INSERT INTO acmebankloandb.customer_loans
(customer_id, loan_amount, loan_date, loan_interest_rate, loan_type, loan_req_id)
VALUES ([data from form])
```

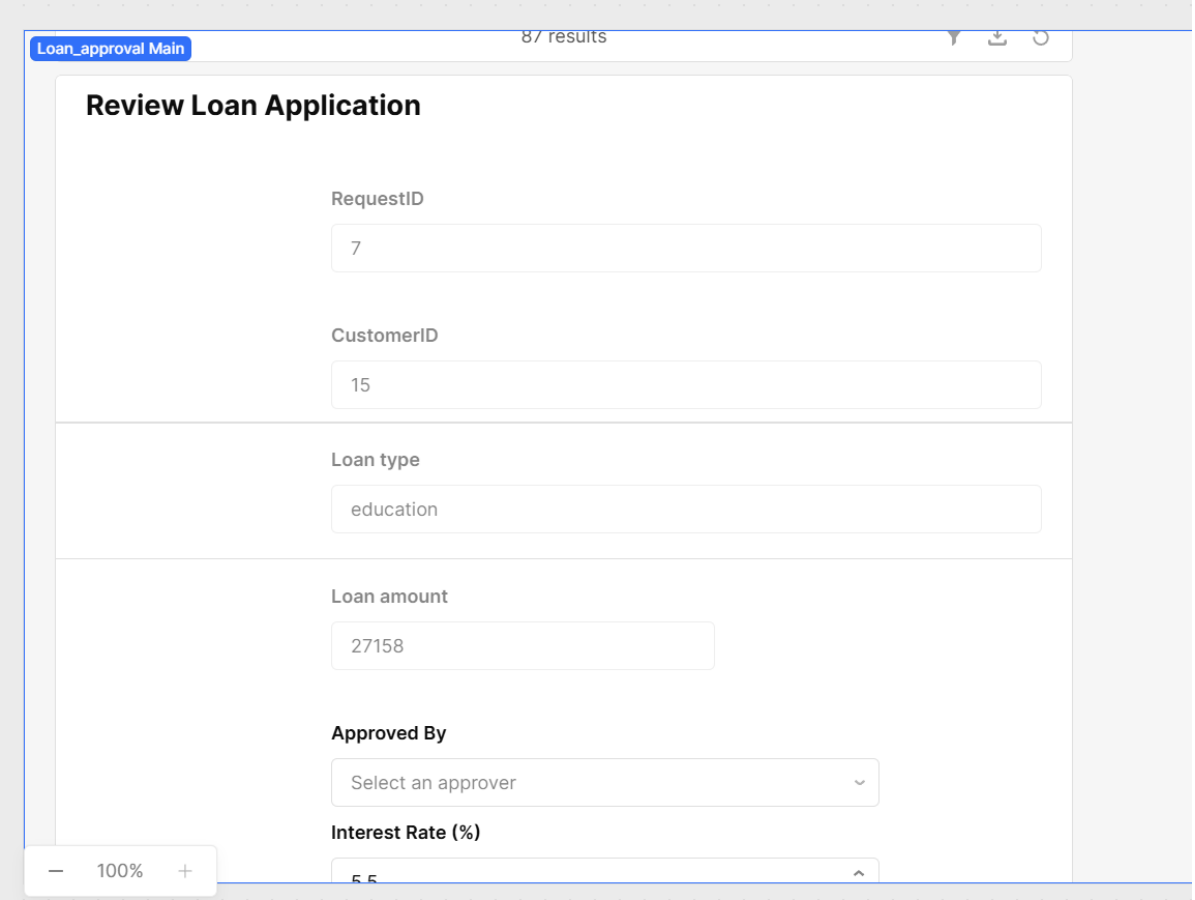
9.5.5. rejectLoanRequest (UPDATE)

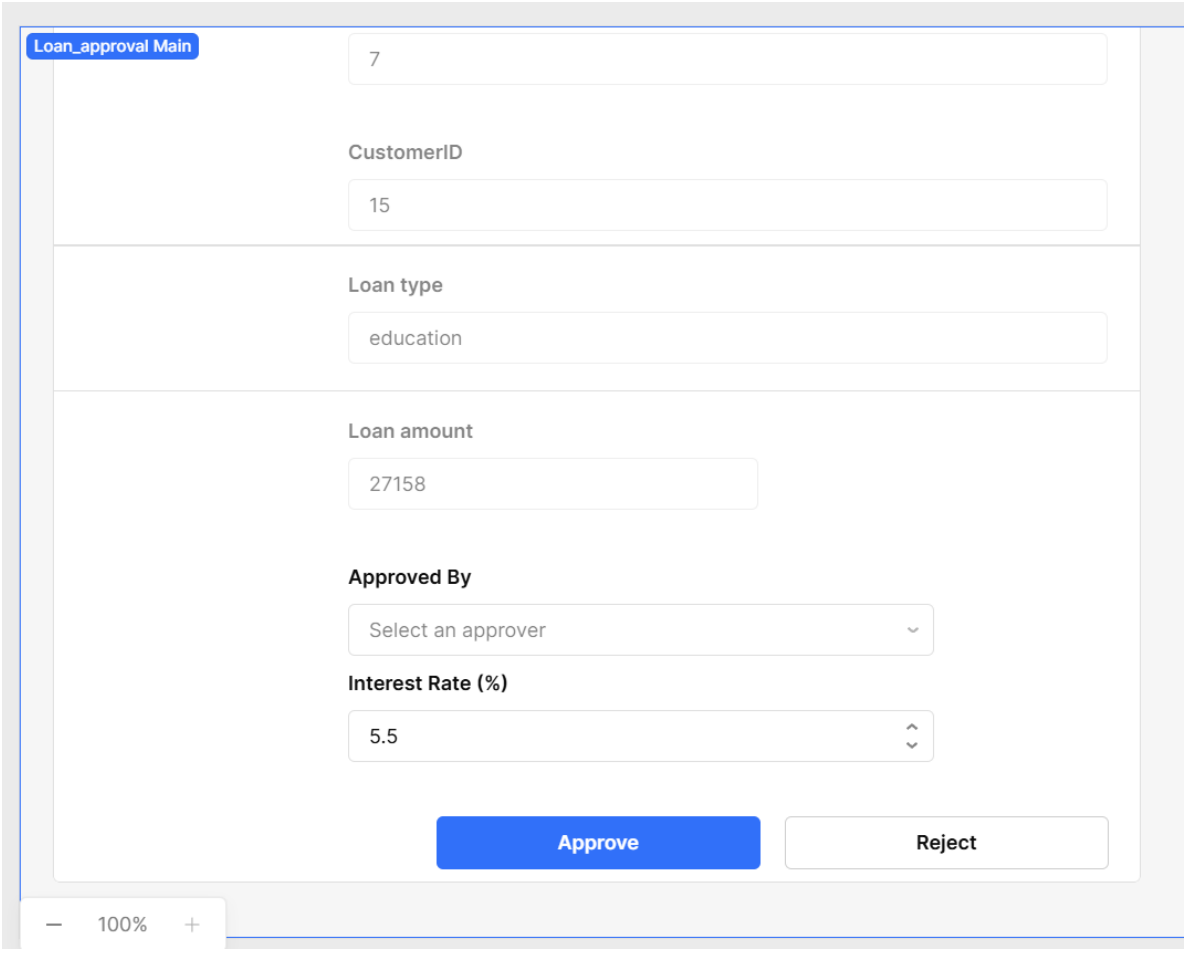
To Updates loan status to REJECTED

```
UPDATE acmebankloandb.loan_requests
SET lreq_status = 'rejected',
```

lreq_approved_by = [selected approver ID]

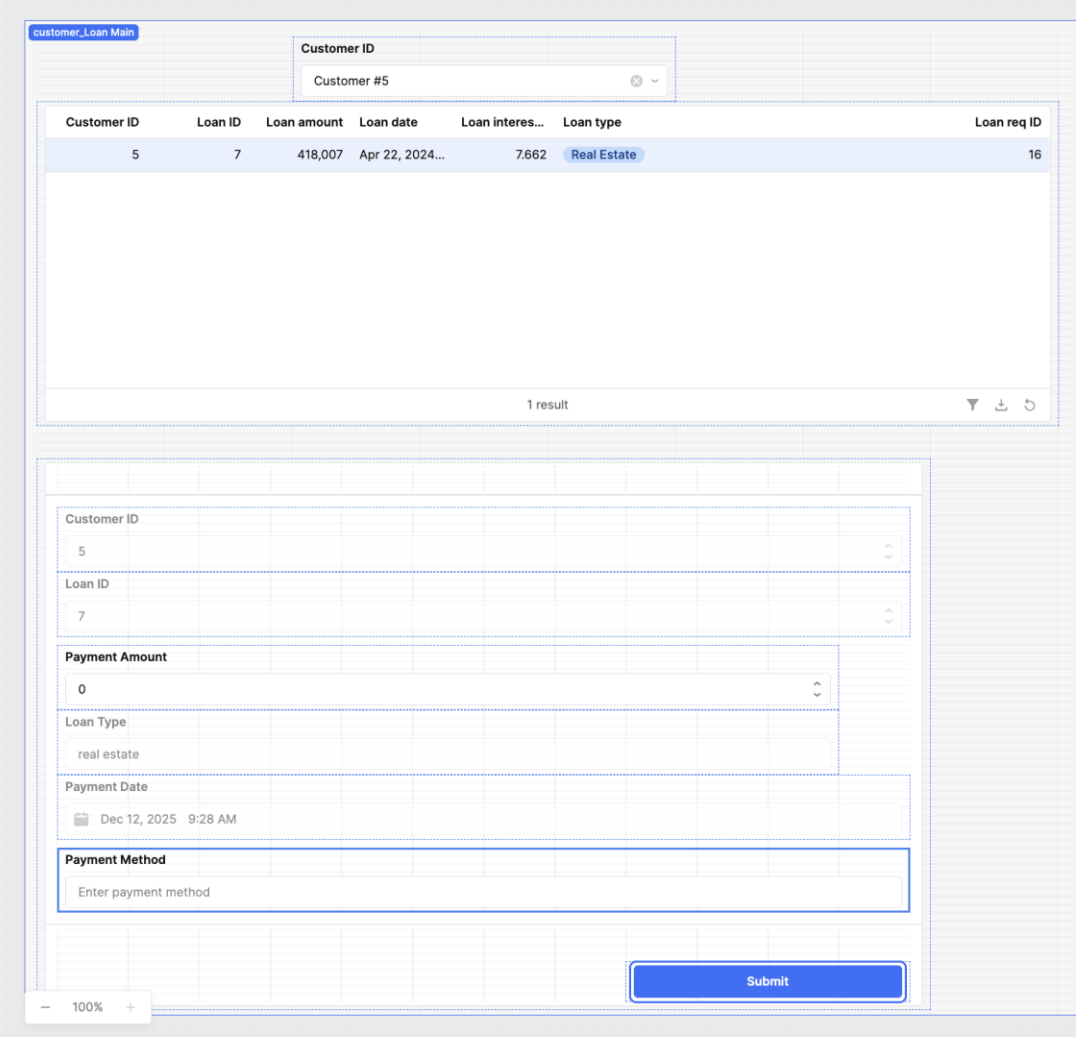
WHERE lreq_request_id = [selected loan ID]




Loan_approval Main
7
CustomerID
15
Loan type
education
Loan amount
27158
Approved By
Select an approver
Interest Rate (%)
5.5
Approve
Reject
100%

9.4 Make a Payments

Once the request is approved, the customer will see the loan on his page and will be able to make payments towards it. Table with applicable customer loans (customer selects their ID in the dropdown options and sees relevant loans from customer loan table)



If you see the above picture, it shows that specific filtered "customer ID" in the drop-down menu which is only populating on the table. In this system, if you click on that row, it's going to display the same "Customer ID" in the form we have made.

9.4.2. Payment FORM:

In the same picture, if you check that “FROM” shows the same “customer ID” that system fetched after filtering on the table. Then customer can put payment amount, Payment date (Current date) and payment method. If a customer hits on the submit button, it’s automatically populated in our main database which is connected integrated here in Retool.

a. **Sql script to write to payment table:**

{{ loanLoanIdInput.value }}

{{ paymentAmountInput.value }}

{{ moment().format('YYYY-MM-DD HH:mm:ss') }}

{{ paymentMethodInput.value }}

Payment table connection in the form:

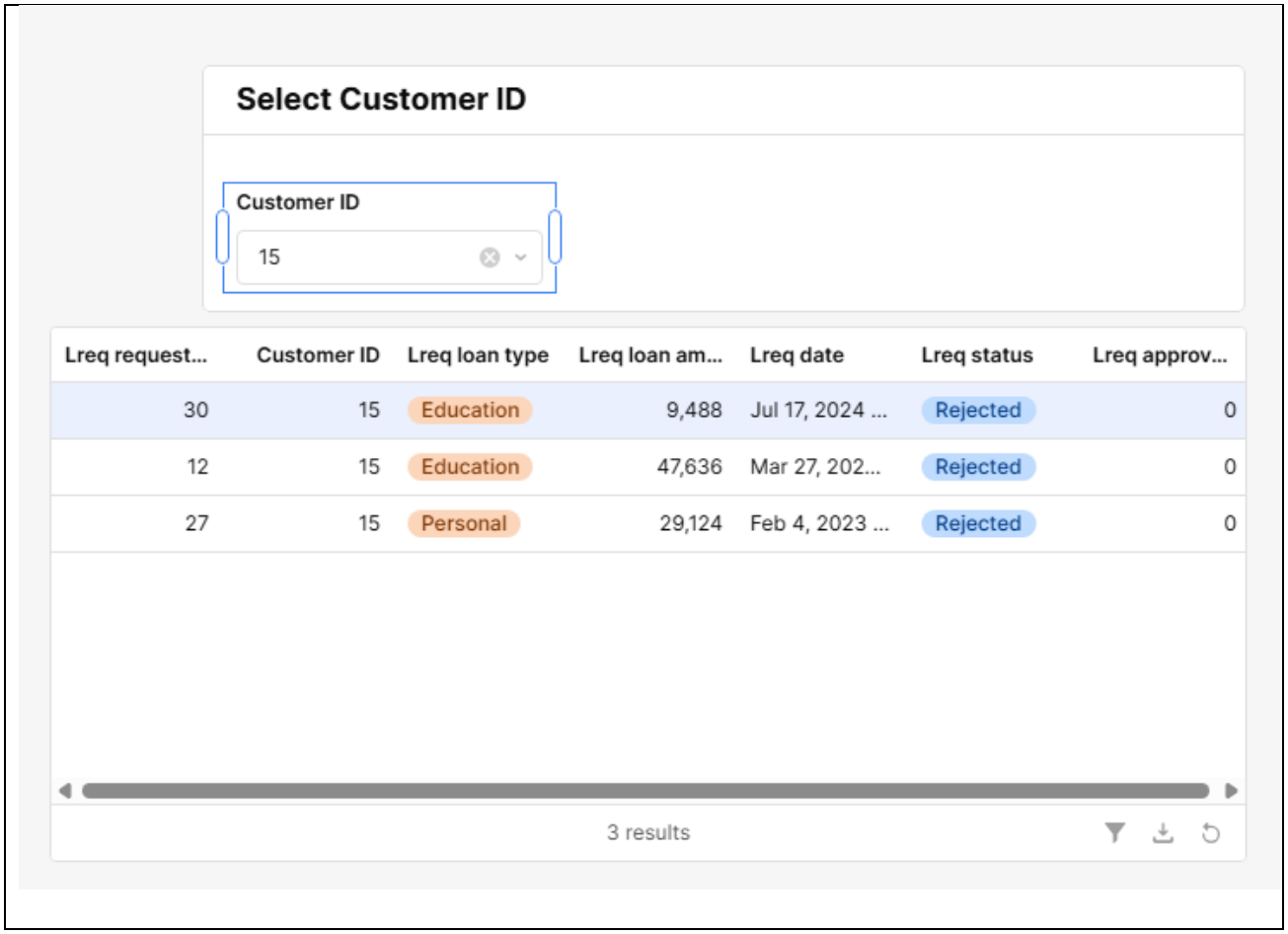
INSERT INTO acmebankloandb.payment (loan_id, pmt_amount, pmt_date, pmt_payment_method)

VALUES ({{ loanLoanIdInput.value }}, {{ paymentAmountInput.value }}, NOW(), {{ paymentMethodInput.value }});

9.5 Rejected Loans Queries

9.5.1. Table with customerID showcasing rejected loans

```
SELECT
lreq_request_id,
  customer_id,
  lreq_loan_type,
  lreq_loan_amount,
  lreq_date,
  lreq_status,
  lreq_approved_by
FROM acmebankloandb.loan_requests
WHERE lreq_status = 'rejected'
AND ({{ !numberInput2.value }} OR customer_id = {{ numberInput2.value }})
ORDER BY lreq_date DESC;
```


Select Customer ID
Customer ID
15
Lreq request...
Customer ID
Lreq loan type
Lreq loan am...
Lreq date
Lreq status
Lreq approv...
30
15
Education
9,488
Jul 17, 2024 ...
Rejected
0
12
15
Education
47,636
Mar 27, 202...
Rejected
0
27
15
Personal
29,124
Feb 4, 2023 ...
Rejected
0
3 results

10. Conclusion & Team Reflection

10.1 Group Experience

Discuss:

- What steps were the easiest?
    - Understanding the requirements and translating them into a roadmap and project timeline
    - The concept of the bank app and the functionality was relatively clear

- What steps were hardest
    - Normalization took a few tries to implement properly
    - Keeping a focus on the requirement and not adding items that were nice to have instead of must haves.
    - Learning and building in retool required a new type of thinking and coding that we were unused to
    - Managing schedules between people with different levels of availability

- What did we learn that we did not expect?

    - The coding engine behind the retool application and its integration with the database was certainly unexpected
    - The costing was also not at all what we initially hypothesized

- How we overcame technical challenges
    - Some use of AI tools in retool and Mockaroo were very helpful
    - Ultimately, the best way to approach the technical challenges was trial and error. We would try something out and see if the result matched what we wanted
- If we had to do it all over again, what would we have done differently?
    - We would have started getting familiar with Retool much earlier.
    - We would have aligned on schedules and expectations more clearly so that no one was saddled with unexpected work

10.2 Final Comments

In conclusion, this project helped us design a clear and organized database system that supports the entire loan process from customer requests to approvals and payments.

We learned how proper data modeling, normalization, and SQL design can make a real business workflow more efficient and reliable. The final system provides a solid foundation for managing customers, collateral, income history, and loan records in a way that is accurate, consistent, and easy to maintain.

Group Meeting Log Sheet

Meeting 1

Date of Meeting: 9_/18___/_ **Time of Meeting: ___9 a.m.__**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Christopher and Fabian

Absent: **__EVAN_____** **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Overview of the project requirement**
2. **Understanding our 1st step**
3. **Strategy about how to proceed further as a team**
4. **Project folder created in the OneDrive for an easy access**
5. **Will put our thoughts and questions as per the requirement**
6. **Will discuss our progress in next meeting tomorrow morning**
7. **Talk to professor tomorrow with our understanding**

Tasks Assigned	Team Member	Delivery Date
Working on a Project requirement	**Shilpa, Roshani, Fabian, Christopher**	**9/24**

Meeting Ending Time: _12:16 p.m

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	

Meeting 2

Date of Meeting: 9_/24___/_ **Time of Meeting: 8pm**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Christopher, Fabian, Evan and Professor Jefferson

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

For Project

1. **Finalized with the requirement gathering document along with risk, benefit and cost**
2. **Modification done on the document (Everyone contributed)**
3. **Discussed about the 1st phase submission**
4. **Meeting members: Shilpa, Roshani, Fabian, Christopher and Evan**

Tasks Assigned	**Team Member**	**Delivery Date**
Updated project requirements as per professor feedback. **Review the document once we have a meeting with a professor**	**All** **All**	

Meeting Ending Time: 9pm_

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 3

Date of Meeting: 9___/28___/_ **Time of Meeting: ___10:00a.m__**

Group : 2 **Recorder:** Evan Kravitz

Attending: All

Absent: **__None_____** **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

- **Updates and comments on the requirements document**
- **Editing functional requirements**
- **Finalizing cost estimates**

Tasks Assigned	Team Member	Delivery Date
Submit document **Fabian start the lucid chart and share on teams for everyone to work on conceptual model for HW 1**	**Shilpa** **All**	**9/28**

Meeting Ending Time: _11:15p.m

Performance Appraisal & Sign-off

Team member name (print)_____ Signature_____ Weekly Contribution

Meeting 4

Date of Meeting: 10_/13__/_ **Time of Meeting: 9 pm**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

For HW 1

1. **Finalized the conceptual module**
2. **Discussed about the Logical module**
3. **Discuss about next step "External Views".**
4. **Thursday will have a final meeting**

Tasks Assigned	Team Member	Delivery Date

Evan start the logical model and forward it to teams and others should update as we go **Review the Business rules and give the feedback**	**All** **All**	

Meeting Ending Time: 9:55 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 5

Date of Meeting: 10_/14___/_ **Time of Meeting: 9pm**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Fabian and Evan

Absent: Christopher _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Logical Model finalized for HW1**
2. **How to get started with the External View**

Tasks Assigned	Team Member	Delivery Date
Shilpa Start the draft of External view and share it in teams for the team to update and edit	**All**	

Meeting Ending Time: 10 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Evan	**EK**	

Meeting 6

Date of Meeting: _10/_18__/_ **Time of Meeting: 4:00 pm**

Group : 2 **Recorder:**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Finalized the External view for the HW1**
2. **How to prepare the document**

Tasks Assigned	**Team Member**	**Delivery Date**
Roshani prepare the document for the HW1 for the Submission and forward it to the group for a review **Once everyone is done with the review submit the document.**	**All** **Roshani**	

Meeting Ending Time: 5pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 7

Date of Meeting: 10_/23___/_ **Time of Meeting: 8:00 pm**

Group : 2 **Recorder:**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**
_____ **YES / NO**
_____ **YES / NO**

Topics Discussed:

For Project

1. **Getting started with conceptual model**
2. **Reviewed the Business Requirement**

Tasks Assigned	**Team Member**	**Delivery Date**
Get Started with the Conceptual model **Send the model in teams for others to review and add comments**	**Roshani** **All**	**ASAP**

Meeting Ending Time: 8;30 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 8

Date of Meeting: 11_/7___/_ **Time of Meeting: 8:00 pm**

Group : 2 **Recorder: Team**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**
_____ **YES / NO**
_____ **YES / NO**

Topics Discussed:

For project

1. Updated Conceptual Model based on Professor's feedback.

2. Start with logical Model.

3. Meetup with professor once we are done with logical model

Tasks Assigned	Team Member	Delivery Date
Update the Conceptual model based on professor's feedback **Get Started with Logical Model** **Share the logical model to teams for everyone to work and update as we go**	**All** **Fabian** **All**	**ASAP**

Meeting Ending Time: 9:00 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 9

Date of Meeting: 11_/_19__/_ **Time of Meeting: 8:30**

Group : 2 **Recorder: Team**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Discussed execution plan about HW3**
2. **Discussed about progress and next step on Logical and Phyical modeling**
3. **Discussed about project report progress**
4. **Nov 24th will have next meeting to go through the work done so far**
5.

Tasks Assigned	Team Member	Delivery Date
Work on HW 3 on your own so we all understand and can divide the sections for the recording.	**All**	

Meeting Ending Time: 11 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 10

Date of Meeting: _11/ 24___/_ **Time of Meeting: 9:30 pm**

Group : 2 **Recorder: Team**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Doubts regarding homework3**
2. **Getting approval for logical model for our project from the professor**

Tasks Assigned	Team Member	Delivery Date
Prepare for the recording for HW3 **Updated the logical model once we get an approval from the professor**	**All** **All**	

Meeting Ending Time: 11:00 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 10

Date of Meeting: 11_/28___/_ **Time of Meeting: 8:00 pm**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: Evan,_____ **Excused (circle)?: YES / NO**

Christopher _____ **YES /NO**

_____ **YES / NO**

Topics Discussed:

In today's meeting, we discussed the following items:

- **The homework 3 recording process**
- **Indexing and its requirements**
- **How to organize and structure the recording**
- **The logical model and conceptual model for the main project**

Based on this discussion, we plan to schedule another meeting with the entire team to conduct a rehearsal before the final recording.

Tasks Assigned	Team Member	Delivery Date

Meeting Ending Time: 9:00pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	

Meeting 11

Date of Meeting: 11_/_29__/_ **Time of Meeting: 7:00 pm**

Group : 2 **Recorder:**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Recorded the video for HW 3 submission.**

Tasks Assigned	**Team Member**	**Delivery Date**
Forward the sql scripts on teams. **Update the Document for HW 3** **Submit the HW3 Doc**	**All** **Shilpa and Roshani** **Shilpa**	

Meeting Ending Time: 9:00 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 12

Date of Meeting: _12/ 4___/_ **Time of Meeting: 8:00 pm**

Group : 2 **Recorder: Roshani**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Update the logical model after professor feedback.**
2. **Use Mackaroo to create a Sample data**
3. **Get started with Retool**

Tasks Assigned	Team Member	Delivery Date
Everyone use the Mackaroo and retool for learning	**All**	
Work on mackaroo and connect to the Database	**Fabian and Shilpa**	
Get Started with retool and share with teammates	**Evan**	
Updated the conceptual model as per updated logical model and reserve the Study room	**Roshani**	
Create a template for the Final report	**Christopher**	

Meeting Ending Time: 8:30 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 13

Date of Meeting: 12/_8__/_ **Time of Meeting: 9:30pm**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. **Discussed about the retool and how we can work with it.**
2. **Assigned sections for all of all**

Tasks Assigned	**Team Member**	**Delivery Date**
We divided the sections to work on retool. **Part 1** **Part 2** **Part 3** **Part 4** **Part 5** **Part 6** **Part 7**	 **Evan** **Fabian** **TBD** **Roshani** **Shilpa** **Christopher**	**12/11**

Meeting Ending Time: _10:45 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	

Meeting 14

Date of Meeting: 12 _/ 11___/_ **Time of Meeting: 5:00 pm**

Group : 2 **Recorder: Roshani Hada**

Attending: Shilpa, Roshani, Christopher, Fabian and Evan

Absent: _____ **Excused (circle)?:** **YES / NO**

_____ **YES / NO**

_____ **YES / NO**

Topics Discussed:

1. A few remaining items to finalize in Retool, we're close to completion.
2. The project report is currently in progress.
3. Work on the log sheet
4. The presentation still needs to be prepared.

Tasks Assigned	**Team Member**	**Delivery Date**
Complete the Retool work **Update the log Sheet** **Start working on presentation and send it to teams** **Update the report**	**Evan and Fabian** **Roshani** **Shilpa** **All**	

Meeting Ending Time: _8:00 pm

Performance Appraisal & Sign-off

Team member name (print)	Signature	Weekly Contribution
Shilpa	**SK**	
Roshani	**RH**	
Fabian	**FG**	
Christopher	**CC**	
Evan	**EK**	