



## **Big Data for Managers & Analytics**

**Submitted To:**

Prof. Amarnath Mitra

**Report on Project on Database Management using MySQL**

**Topic:** Financial Database

**Submitted by:**

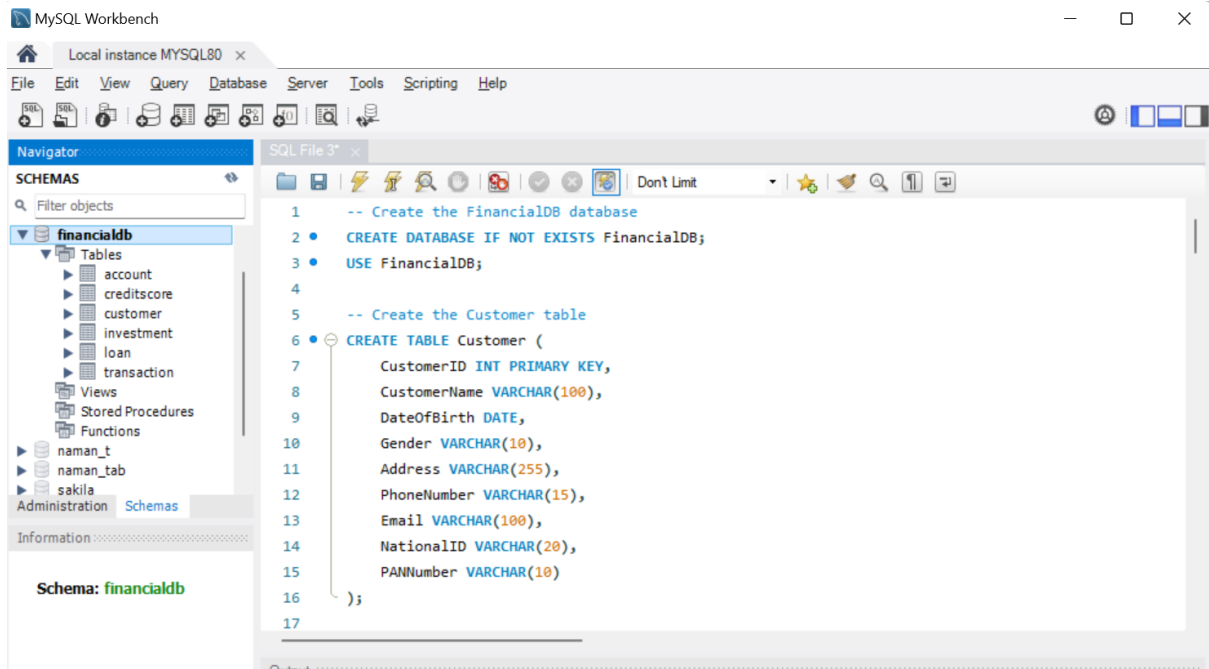
**Naman Agarawal (064030)**

## Database

This schema “the FinancialDB” database is to maintain a comprehensive and organized repository of financial data for customers, including their accounts, credit scores, investments, loans, and transaction histories. This database aims to facilitate efficient management, analysis, and reporting of financial information, enabling better decision-making and customer service for financial institutions.

### Description of Each Table in FinancialDB

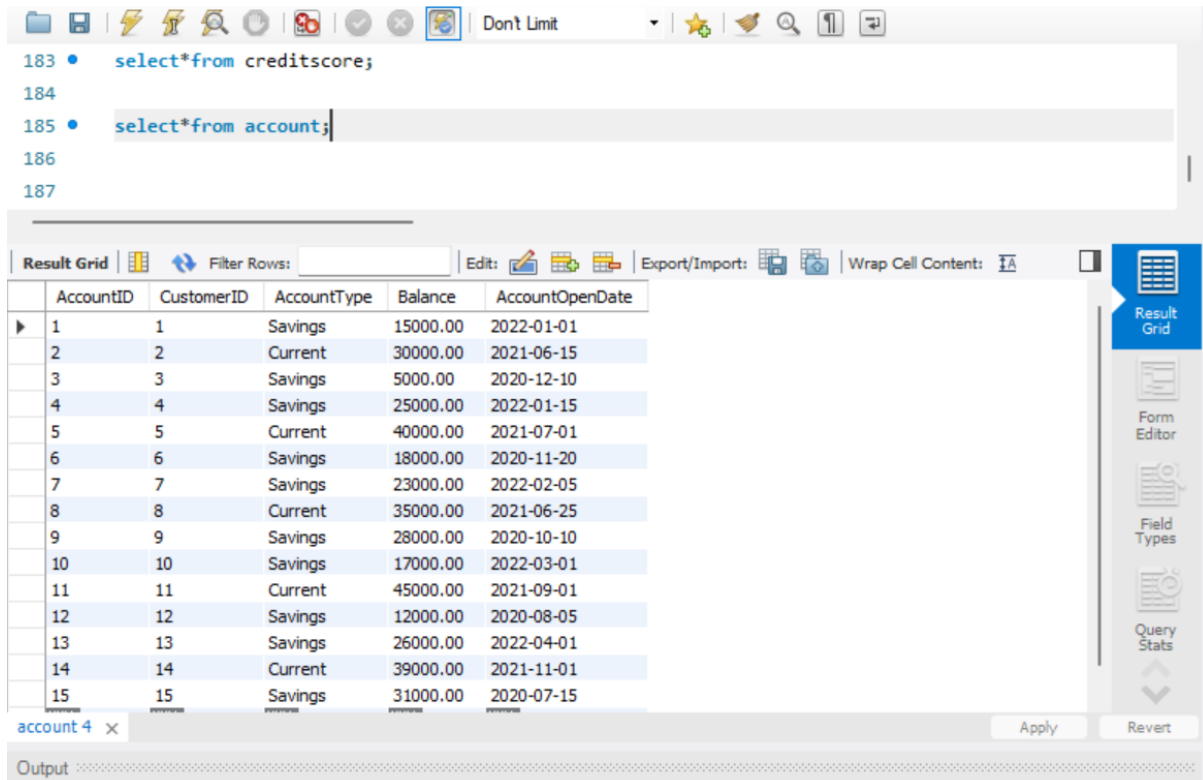
1. **Account:** Stores detailed information about each customer's bank account, including account type, balance, and associated customer ID.
2. **CreditScore:** Records the credit scores of customers, tracking their financial reliability and risk profile over time.
3. **Customer:** Contains personal and contact information for each customer, serving as the central entity to which all financial activities are linked.
4. **Investment:** Holds data related to customer investments, including types of investments, amounts invested, and performance metrics.
5. **Loan:** Tracks information about loans taken by customers, including loan amount, interest rate, payment schedule, and status.
6. **Transaction:** Captures the details of all financial transactions, such as deposits, withdrawals, and transfers, for each customer account.



## Tables

The database consists of six tables, each serving a specific role and these tables are interconnected, with **Account** being the central entity, linking to **Credit Score**, **Customer**, **Investment**, **Loan**, and **Transaction**.

- **Account:** This table stores information about each customer's bank account, including details like account number, type of account (savings, checking, etc.), current balance, and the date the account was opened.



The screenshot shows a database query tool interface. The top section contains a SQL query editor with the following queries:

```
183 • select*from creditscore;
184
185 • select*from account;
186
187
```

The bottom section displays the results of the query in a table format. The table has the following columns: AccountID, CustomerID, AccountType, Balance, and AccountOpenDate. The results are as follows:

AccountID	CustomerID	AccountType	Balance	AccountOpenDate
1	1	Savings	15000.00	2022-01-01
2	2	Current	30000.00	2021-06-15
3	3	Savings	5000.00	2020-12-10
4	4	Savings	25000.00	2022-01-15
5	5	Current	40000.00	2021-07-01
6	6	Savings	18000.00	2020-11-20
7	7	Savings	23000.00	2022-02-05
8	8	Current	35000.00	2021-06-25
9	9	Savings	28000.00	2020-10-10
10	10	Savings	17000.00	2022-03-01
11	11	Current	45000.00	2021-09-01
12	12	Savings	12000.00	2020-08-05
13	13	Savings	26000.00	2022-04-01
14	14	Current	39000.00	2021-11-01
15	15	Savings	31000.00	2020-07-15

The interface also includes a toolbar with various icons for editing and viewing the data, and a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

- **CreditScore:** This table maintains a record of the customers' credit scores, along with the date when each score was recorded, helping to assess creditworthiness and track changes over time.

179 (15, 15, 725, '2022-06-01');  
 180  
 181 • select\*from customer;  
 182  
 183 • select\*from creditscore;

	CreditScoreID	CustomerID	CreditScore	ScoreDate
1	1	1	750	2022-02-01
2	2	2	680	2022-02-01
3	3	3	720	2022-02-01
4	4	4	770	2022-03-01
5	5	5	690	2022-03-01
6	6	6	740	2022-03-01
7	7	7	760	2022-04-01
8	8	8	675	2022-04-01
9	9	9	710	2022-04-01
10	10	10	780	2022-05-01
11	11	11	695	2022-05-01
12	12	12	730	2022-05-01
13	13	13	755	2022-06-01
14	14	14	685	2022-06-01
15	15	15	725	2022-06-01

creditscore 3 x Apply Revert

- **Customer:** This table contains key personal information for each customer, such as name, address, contact details, and customer ID, which uniquely identifies each customer across the database.

179 (15, 15, 725, '2022-06-01');  
 180  
 181 • select\*from customer;  
 182  
 183

	CustomerID	CustomerName	DateOfBirth	Gender	Address	PhoneNumber	Email
3	3	Robert Johnson	1975-12-10	Male	789 Pine Road, Springfield	9876543212	robert.johnson@example.com
4	4	Emily Davis	1982-03-22	Female	321 Maple Street, Springfield	9876543213	emily.davis@example.com
5	5	Michael Brown	1978-07-05	Male	654 Cedar Avenue, Springfield	9876543214	michael.brown@example.com
6	6	Sarah Wilson	1995-09-18	Female	987 Oakwood Drive, Springfield	9876543215	sarah.wilson@example.com
7	7	David Miller	1988-11-25	Male	159 Spruce Road, Springfield	9876543216	david.miller@example.com
8	8	Sophia Moore	1992-04-10	Female	753 Birch Lane, Springfield	9876543217	sophia.moore@example.com
9	9	James Anderson	1981-01-30	Male	258 Pineapple Street, Springfield	9876543218	james.anderson@example.com
10	10	Olivia Taylor	1993-12-15	Female	369 Oakwood Avenue, Springfield	9876543219	olivia.taylor@example.com
11	11	Christopher Harris	1986-05-20	Male	147 Walnut Street, Springfield	9876543220	christopher.harris@example.com
12	12	Ava Martinez	1997-02-25	Female	753 Cherry Drive, Springfield	9876543221	ava.martinez@example.com
13	13	Matthew Thompson	1976-08-05	Male	951 Pine Cone Lane, Springfield	9876543222	matthew.thompson@example.com
14	14	Isabella White	1989-10-22	Female	369 Oakwood Avenue, Springfield	9876543223	isabella.white@example.com
15	15	Daniel Lewis	1983-07-30	Male	852 Maplewood Avenue, Springfield	9876543224	daniel.lewis@example.com
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Investment:** This table records customer investment activities, including details of the types of investments (stocks, bonds, mutual funds), investment dates, amounts, and current value of investments.

SQL File 3\* x

183 • `select*from creditscore;`  
 184  
 185 • `select*from account;`  
 186  
 187 • `select*from investment;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	InvestmentID	CustomerID	InvestmentType	InvestmentAmount	InvestmentDate	MaturityDate	InterestRate
▶	1	1	Fixed Deposit	20000.00	2022-01-01	2025-01-01	6.50
	2	2	Mutual Fund	30000.00	2021-06-15	2024-06-15	8.00
	3	3	Bonds	10000.00	2020-12-10	2025-12-10	7.00
	4	4	Fixed Deposit	25000.00	2022-02-01	2026-02-01	6.70
	5	5	Mutual Fund	40000.00	2021-07-01	2025-07-01	8.20
	6	6	Bonds	15000.00	2020-11-10	2025-11-10	7.10
	7	7	Fixed Deposit	30000.00	2022-03-01	2026-03-01	6.60
	8	8	Mutual Fund	35000.00	2021-06-25	2024-06-25	7.90
	9	9	Bonds	12000.00	2020-10-01	2025-10-01	7.30
	10	10	Fixed Deposit	28000.00	2022-04-01	2025-04-01	6.80
	11	11	Mutual Fund	45000.00	2021-09-01	2024-09-01	8.10
	12	12	Bonds	18000.00	2020-08-05	2025-08-05	7.20
	13	13	Fixed Deposit	26000.00	2022-05-01	2026-05-01	6.90
	14	14	Mutual Fund	39000.00	2021-11-01	2025-11-01	8.30
	15	15	Bonds	31000.00	2020-07-15	2025-07-15	7.40

investment 5 x Apply Revert

- **Loan:** This table holds data on loans issued to customers, including loan type (personal, home, auto, etc.), principal amount, interest rate, repayment schedule, and current loan status.

SQL File 3\* x

185 • `select*from account;`  
 186  
 187 • `select*from investment;`  
 188  
 189 • `select*from loan;`

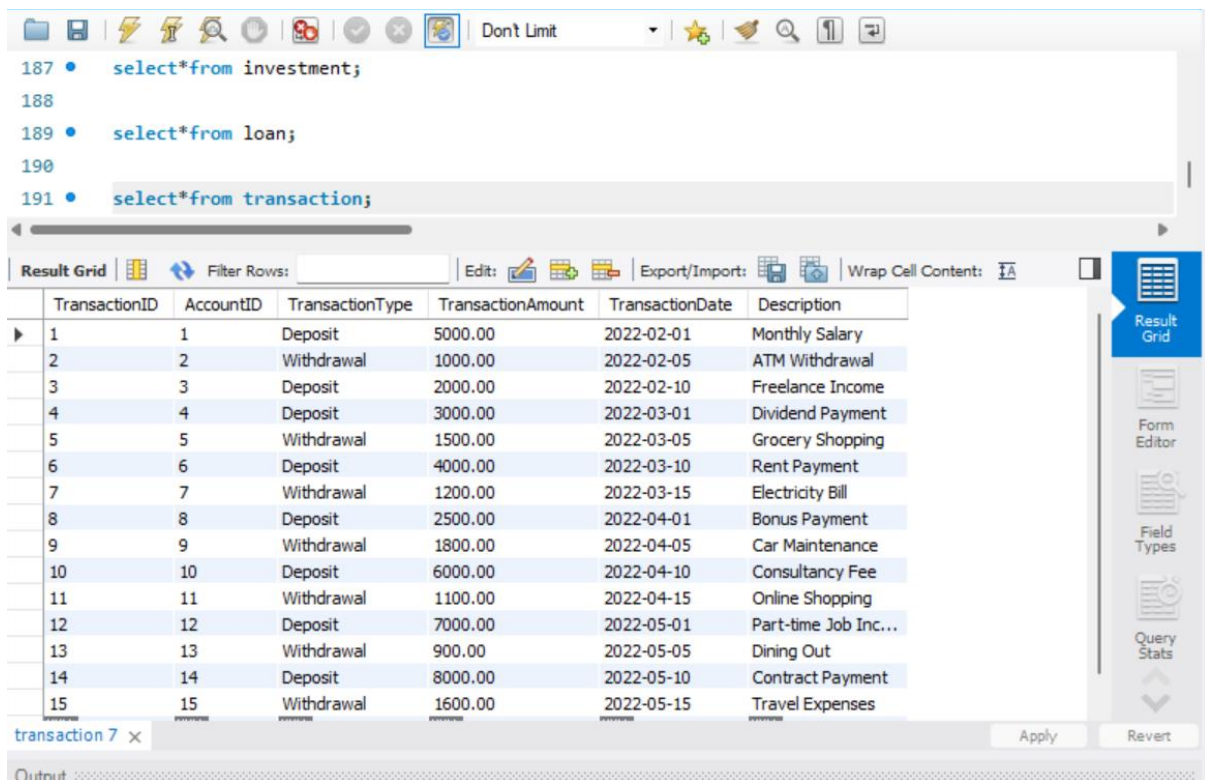
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	LoanID	CustomerID	LoanType	LoanAmount	InterestRate	LoanStartDate	LoanEndDate	LoanStatus
▶	1	1	Personal	50000.00	7.50	2022-01-01	2025-01-01	Active
	2	2	Mortgage	150000.00	6.50	2021-06-15	2031-06-15	Active
	3	3	Auto	25000.00	8.00	2020-12-10	2024-12-10	Paid
	4	4	Education	30000.00	5.50	2022-02-01	2026-02-01	Active
	5	5	Business	100000.00	9.00	2021-07-15	2028-07-15	Active
	6	6	Personal	40000.00	7.00	2020-11-01	2024-11-01	Paid
	7	7	Mortgage	180000.00	6.20	2022-03-01	2032-03-01	Active
	8	8	Auto	22000.00	7.80	2021-06-10	2025-06-10	Active
	9	9	Education	35000.00	5.20	2020-09-01	2025-09-01	Paid
	10	10	Personal	45000.00	7.40	2022-04-01	2026-04-01	Active
	11	11	Business	90000.00	8.50	2021-09-10	2029-09-10	Active
	12	12	Auto	28000.00	7.90	2020-07-01	2024-07-01	Paid
	13	13	Mortgage	160000.00	6.70	2022-05-01	2032-05-01	Active
	14	14	Education	32000.00	5.40	2021-10-01	2026-10-01	Active
	15	15	Personal	55000.00	7.30	2020-08-01	2024-08-01	Paid

loan 6 x Apply Revert

Output

- **Transaction:** This table logs all financial transactions made by customers, such as deposits, withdrawals, transfers, and payments, including transaction dates, amounts, and associated account numbers.



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons. Below it, a SQL query is entered in a text area:

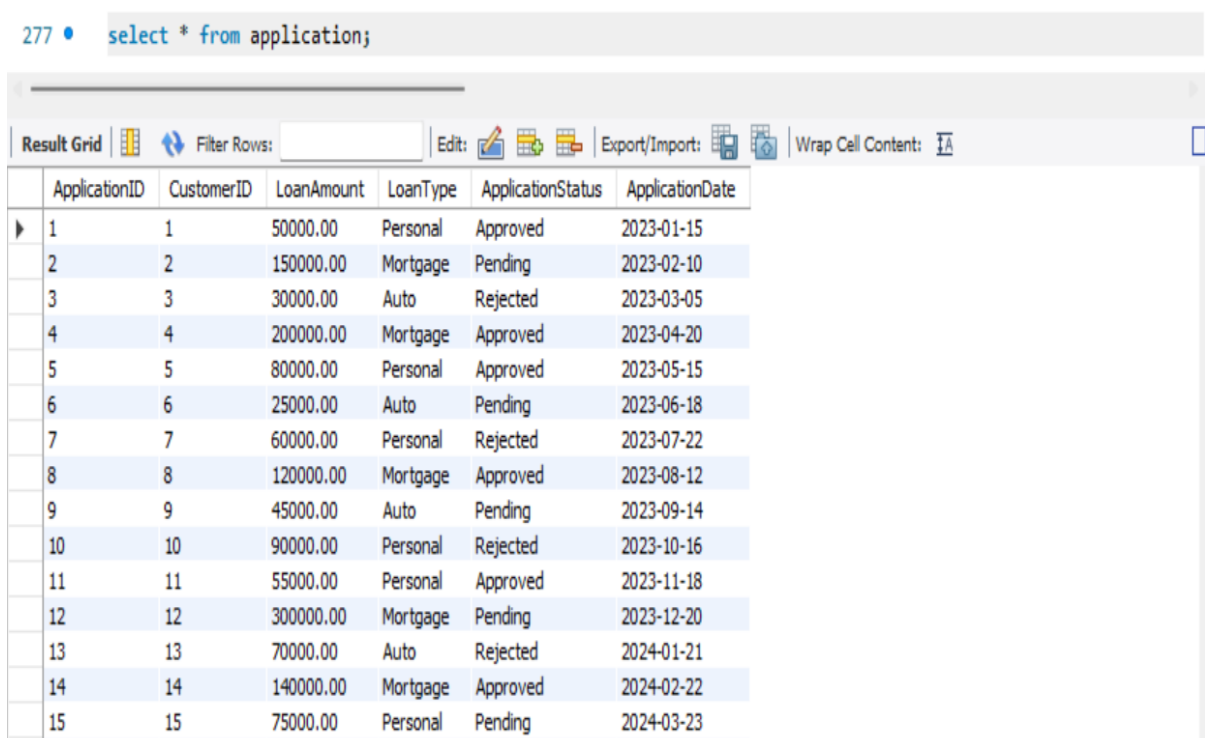
```
187 • select*from investment;
188
189 • select*from loan;
190
191 • select*from transaction;
```

The results are displayed in a table with the following columns: TransactionID, AccountID, TransactionType, TransactionAmount, TransactionDate, and Description. The table contains 15 rows of data.

TransactionID	AccountID	TransactionType	TransactionAmount	TransactionDate	Description
1	1	Deposit	5000.00	2022-02-01	Monthly Salary
2	2	Withdrawal	1000.00	2022-02-05	ATM Withdrawal
3	3	Deposit	2000.00	2022-02-10	Freelance Income
4	4	Deposit	3000.00	2022-03-01	Dividend Payment
5	5	Withdrawal	1500.00	2022-03-05	Grocery Shopping
6	6	Deposit	4000.00	2022-03-10	Rent Payment
7	7	Withdrawal	1200.00	2022-03-15	Electricity Bill
8	8	Deposit	2500.00	2022-04-01	Bonus Payment
9	9	Withdrawal	1800.00	2022-04-05	Car Maintenance
10	10	Deposit	6000.00	2022-04-10	Consultancy Fee
11	11	Withdrawal	1100.00	2022-04-15	Online Shopping
12	12	Deposit	7000.00	2022-05-01	Part-time Job Inc...
13	13	Withdrawal	900.00	2022-05-05	Dining Out
14	14	Deposit	8000.00	2022-05-10	Contract Payment
15	15	Withdrawal	1600.00	2022-05-15	Travel Expenses

The interface also includes a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are 'Apply' and 'Revert' buttons.

1. **Application Table:** Records each loan application, detailing the amount requested, type of loan, status, and the date it was submitted.



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons. Below it, a SQL query is entered in a text area:

```
277 • select * from application;
```

The results are displayed in a table with the following columns: ApplicationID, CustomerID, LoanAmount, LoanType, ApplicationStatus, and ApplicationDate. The table contains 15 rows of data.

ApplicationID	CustomerID	LoanAmount	LoanType	ApplicationStatus	ApplicationDate
1	1	50000.00	Personal	Approved	2023-01-15
2	2	150000.00	Mortgage	Pending	2023-02-10
3	3	30000.00	Auto	Rejected	2023-03-05
4	4	200000.00	Mortgage	Approved	2023-04-20
5	5	80000.00	Personal	Approved	2023-05-15
6	6	25000.00	Auto	Pending	2023-06-18
7	7	60000.00	Personal	Rejected	2023-07-22
8	8	120000.00	Mortgage	Approved	2023-08-12
9	9	45000.00	Auto	Pending	2023-09-14
10	10	90000.00	Personal	Rejected	2023-10-16
11	11	55000.00	Personal	Approved	2023-11-18
12	12	300000.00	Mortgage	Pending	2023-12-20
13	13	70000.00	Auto	Rejected	2024-01-21
14	14	140000.00	Mortgage	Approved	2024-02-22
15	15	75000.00	Personal	Pending	2024-03-23

The interface also includes a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are 'Apply' and 'Revert' buttons.

## Stress Testing

Stress testing of database using the CRUD operations:

- **Create:** Used to insert new records.
- **Read:** Used to retrieve and view data.
- **Update:** Used to modify existing data.
- **Delete:** Used to remove data.

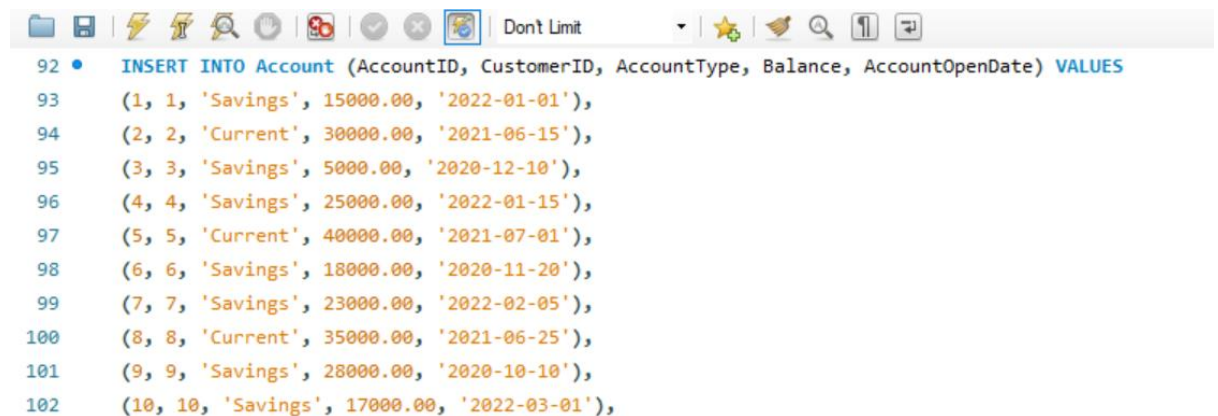
### Create Operation:

#### Inserting into Account Table:

```
INSERT INTO Account (AccountID, CustomerID, AccountType, Balance, AccountOpenDate)
```

```
VALUES
```

```
(1, 1, 'Savings', 15000.00, '2022-01-01'),
```



The screenshot shows a SQL query editor window with a toolbar at the top. The query is an INSERT statement for the Account table, listing 10 rows of data. The data includes AccountID, CustomerID, AccountType, Balance, and AccountOpenDate. The query is as follows:

```
92 • INSERT INTO Account (AccountID, CustomerID, AccountType, Balance, AccountOpenDate) VALUES
93 (1, 1, 'Savings', 15000.00, '2022-01-01'),
94 (2, 2, 'Current', 30000.00, '2021-06-15'),
95 (3, 3, 'Savings', 5000.00, '2020-12-10'),
96 (4, 4, 'Savings', 25000.00, '2022-01-15'),
97 (5, 5, 'Current', 40000.00, '2021-07-01'),
98 (6, 6, 'Savings', 18000.00, '2020-11-20'),
99 (7, 7, 'Savings', 23000.00, '2022-02-05'),
100 (8, 8, 'Current', 35000.00, '2021-06-25'),
101 (9, 9, 'Savings', 28000.00, '2020-10-10'),
102 (10, 10, 'Savings', 17000.00, '2022-03-01'),
```

#### Inserting into Credit Score Table:

```
INSERT INTO CreditScore (CreditScoreID, CustomerID, CreditScore, ScoreDate)
```

```
VALUES
```

```
(1, 1, 750, '2022-02-01'),
```



```
164 • INSERT INTO CreditScore (CreditScoreID, CustomerID, CreditScore, ScoreDate) VALUES
165     (1, 1, 750, '2022-02-01'),
166     (2, 2, 680, '2022-02-01'),
167     (3, 3, 720, '2022-02-01'),
168     (4, 4, 770, '2022-03-01'),
169     (5, 5, 690, '2022-03-01'),
170     (6, 6, 740, '2022-03-01'),
171     (7, 7, 760, '2022-04-01'),
172     (8, 8, 675, '2022-04-01'),
173     (9, 9, 710, '2022-04-01'),
174     (10, 10, 780, '2022-05-01'),
```

## Read (Select) Operation: -

**Retrieving from Account Table where account type is Savings:**

`SELECT * FROM account WHERE accounttype = "savings";`

The screenshot shows a SQL IDE window titled 'SQL File 3\*'. The query editor contains the following SQL code:

```
188
189 • select*from loan;
190
191 • select*from transaction;
192
193 • SELECT * FROM account WHERE accounttype = "savings";
194
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has the following columns: AccountID, CustomerID, AccountType, Balance, and AccountOpenDate. The results are as follows:

AccountID	CustomerID	AccountType	Balance	AccountOpenDate
1	1	Savings	15000.00	2022-01-01
3	3	Savings	5000.00	2020-12-10
4	4	Savings	25000.00	2022-01-15
6	6	Savings	18000.00	2020-11-20
7	7	Savings	23000.00	2022-02-05
9	9	Savings	28000.00	2020-10-10
10	10	Savings	17000.00	2022-03-01
12	12	Savings	12000.00	2020-08-05
13	13	Savings	26000.00	2022-04-01
15	15	Savings	31000.00	2020-07-15
NULL	NULL	NULL	NULL	NULL

The 'Result Grid' tab is active, and the 'Filter Rows' field is empty. The 'Wrap Cell Content' option is checked. The 'Output' tab is also visible at the bottom.

**Retrieving from Customer where gender is Male:**

`select * from customer where gender = "male";`



The screenshot shows a database management interface. At the top, there's a toolbar with various icons and a 'Don't Limit' dropdown. Below the toolbar, a SQL editor contains three queries:

```

190
191 • select*from transaction;
192
193 • SELECT * FROM account WHERE accounttype = "savings";
194
195 • select * from customer where gender = "male";
196

```

Below the editor is a 'Result Grid' section. It includes a 'Filter Rows' input, an 'Edit' button, an 'Export/Import' button, and a 'Wrap Cell Content' checkbox. The grid displays data for the 'customer' table where gender is 'male'.

CustomerID	CustomerName	DateOfBirth	Gender	Address	PhoneNumber	Email
1	John Doe	1985-06-15	Male	123 Elm Street, Springfield	9876543210	john.doe@example.c
3	Robert Johnson	1975-12-10	Male	789 Pine Road, Springfield	9876543212	robert.johnson@exa
5	Michael Brown	1978-07-05	Male	654 Cedar Avenue, Springfield	9876543214	michael.brown@exa
7	David Miller	1988-11-25	Male	159 Spruce Road, Springfield	9876543216	david.miller@example
9	James Anderson	1981-01-30	Male	258 Pineapple Street, Springfield	9876543218	james.anderson@exa
11	Christopher Harris	1986-05-20	Male	147 Walnut Street, Springfield	9876543220	christopher.harris@e
13	Matthew Thompson	1976-08-05	Male	951 Pine Cone Lane, Springfield	9876543222	matthew.thompson@
15	Daniel Lewis	1983-07-30	Male	852 Maplewood Avenue, Springfield	9876543224	daniel.lewis@exampl
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

On the right side of the grid, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom, there's a tab labeled 'customer 11' and buttons for 'Apply' and 'Revert'.

## Update Operation:

### Updating From Customer Table:

UPDATE Customer

SET Email = 'newemail@example.com',

PhoneNumber = '123-456-7890'

WHERE CustomerID = 1;

### Updating From Account Table:

UPDATE account

SET AccountType = 'current',

Balance = 5000

WHERE AccountID = 1;

The screenshot shows a database management interface with a SQL editor and an output window. The SQL editor contains two update queries. The first query updates the 'Customer' table, setting the 'Email' to 'newemail@example.com' and 'PhoneNumber' to '123-456-7890' for 'CustomerID = 1'. The second query updates the 'account' table, setting the 'AccountType' to 'current' and 'Balance' to 5000 for 'AccountID = 1'. The output window shows the execution results for these queries.

```

196
197 • UPDATE Customer
198   SET Email = 'newemail@example.com',
199       PhoneNumber = '123-456-7890'
200   WHERE CustomerID = 1;
201
202 • UPDATE account
203   SET AccountType = 'current',
204       Balance = 5000
205   WHERE AccountID = 1;
206
207
208
209

```

Output

Action Output

#	Time	Message	Duration / Fetch
39	21:30:34	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that us...	0.000 sec
40	21:31:16	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
41	21:32:05	15 row(s) returned	0.000 sec / 0.000 sec
42	21:33:31	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec

## Delete Operation:

### **Deleting from Loan Table:**

Delete from Loan where LoanId = 1;

### **Deleting From Application Table:**

Delete from Application where Applicationid = 1;

```
299 Where applicationid = "10";
300
301
302
303
304 • Delete from Loan where LoanId = 1 ;
305 • Delete from Application where Applicationid = 1;
306
307
308
309
310
311
312
313
```

Context Help Snippets

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✖ 53	03:01:48	Delete from Application where Applicationid = 10	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ...	0.016 sec
✔ 54	03:02:00	Delete from Loan where LoanId = 1	0 row(s) affected	0.000 sec
✔ 55	03:02:00	Delete from Application where Applicationid = 1	1 row(s) affected	0.000 sec