Banking Data Management with MySQL

PIYUSH TIGAONKAR

Problem Statement

The ICIC Bank database system lacked a structured and efficient solution for managing critical banking operations, such as:

- Retrieving employee and customer information based on specific criteria.
- Analyzing account details, balances, and transaction patterns.
- Ensuring seamless integration of multiple departments and job roles within the organization.
- Generating dynamic reports for operational insights.
- To address these challenges, a robust relational database was designed and implemented using MySQL to store, manage, and retrieve banking data efficiently while maintaining data integrity and consistency.

Problem Statement

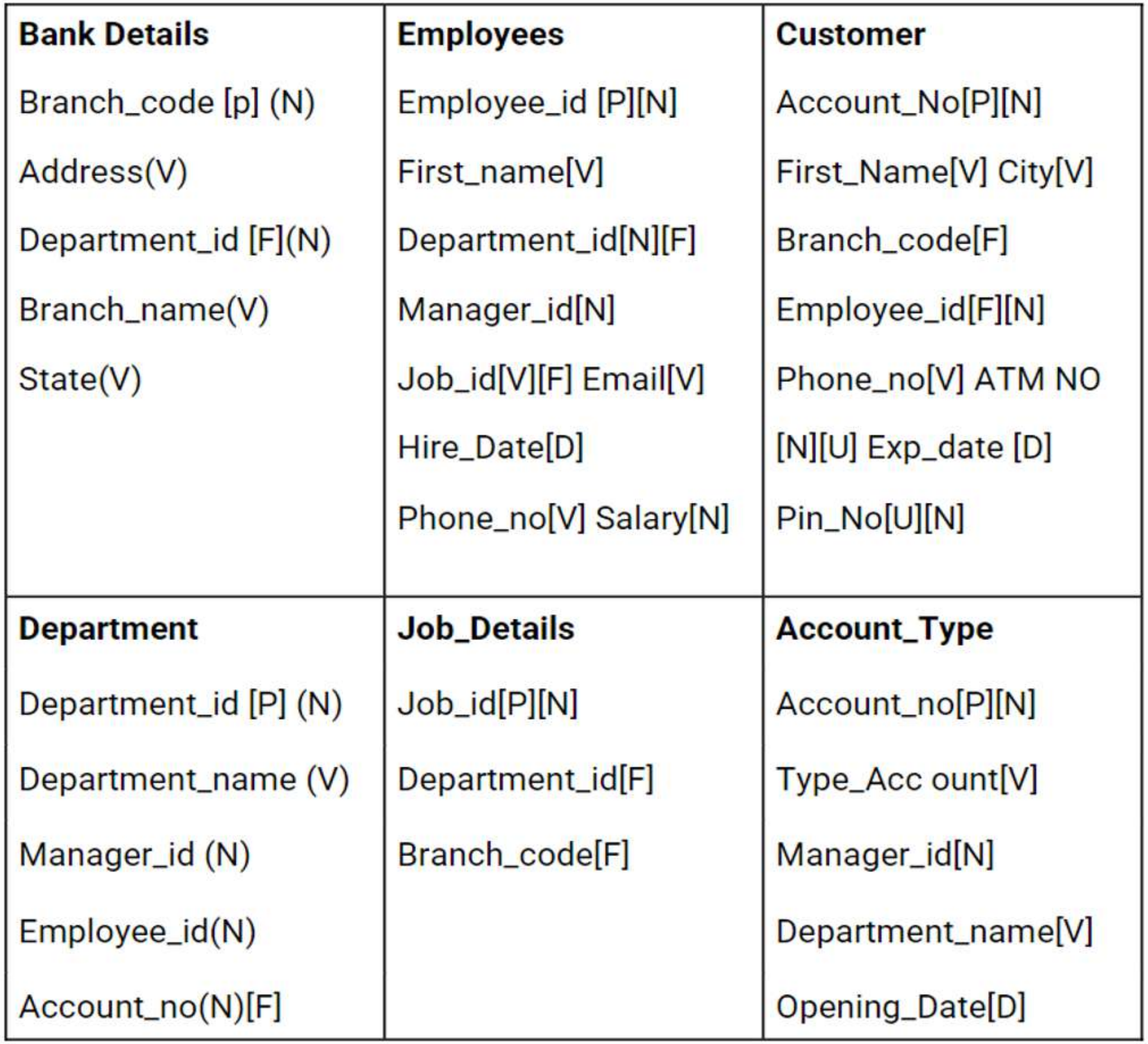
Challenges Addressed :-

1. ***Data Redundancy***:
    - Ensuring **no duplicate records** across different departments and customer accounts.
2. ***Efficient Data Retrieval***:
    - Managing complex queries to **retrieve customer and employee details** across multiple tables.
3. ***Data Integrity***:
    - Maintaining **accurate relationships** between customers, accounts, employees, and departments.
4. ***Scalability***:
    - Handling a large dataset across multiple branches with **flexibility for future growth**.
5. ***Query Optimization***:
    - Designing **optimized queries** for fast retrieval of data, minimizing response time.
6. ***Security***:
    - Ensuring sensitive customer information like **ATM numbers and PINs** are securely managed.

About the Data

The data consists of six key tables:-

1. **ACCOUNT_TYPE -**
2. **DEPARTMENT**
3. **BANK DETAILS**
4. **JOB_DETAILS**
5. **EMPLOYEES**
6. **CUSTOMER**



Bank Details	**Employees**	**Customer**

Branch_code [p] (N)
Address(V)
Department_id [F](N)
Branch_name(V)
State(V)
	Employee_id [P][N]
	First_name[V]
	Department_id[N][F]
	Manager_id[N]
	Job_id[V][F] Email[V]
	Hire_Date[D]
	Phone_no[V] Salary[N]
		Account_No[P][N]
		First_Name[V] City[V]
		Branch_code[F]
		Employee_id[F][N]
		Phone_no[V] ATM NO
		[N][U] Exp_date [D]
		Pin_No[U][N]

Department	**Job_Details**	**Account_Type**
Department_id [P] (N) Department_name (V) Manager_id (N) Employee_id(N) Account_no(N)[F]	Job_id[P][N] Department_id[F] Branch_code[F]	Account_no[P][N] Type_Acc ount[V] Manager_id[N] Department_name[V] Opening_Date[D]

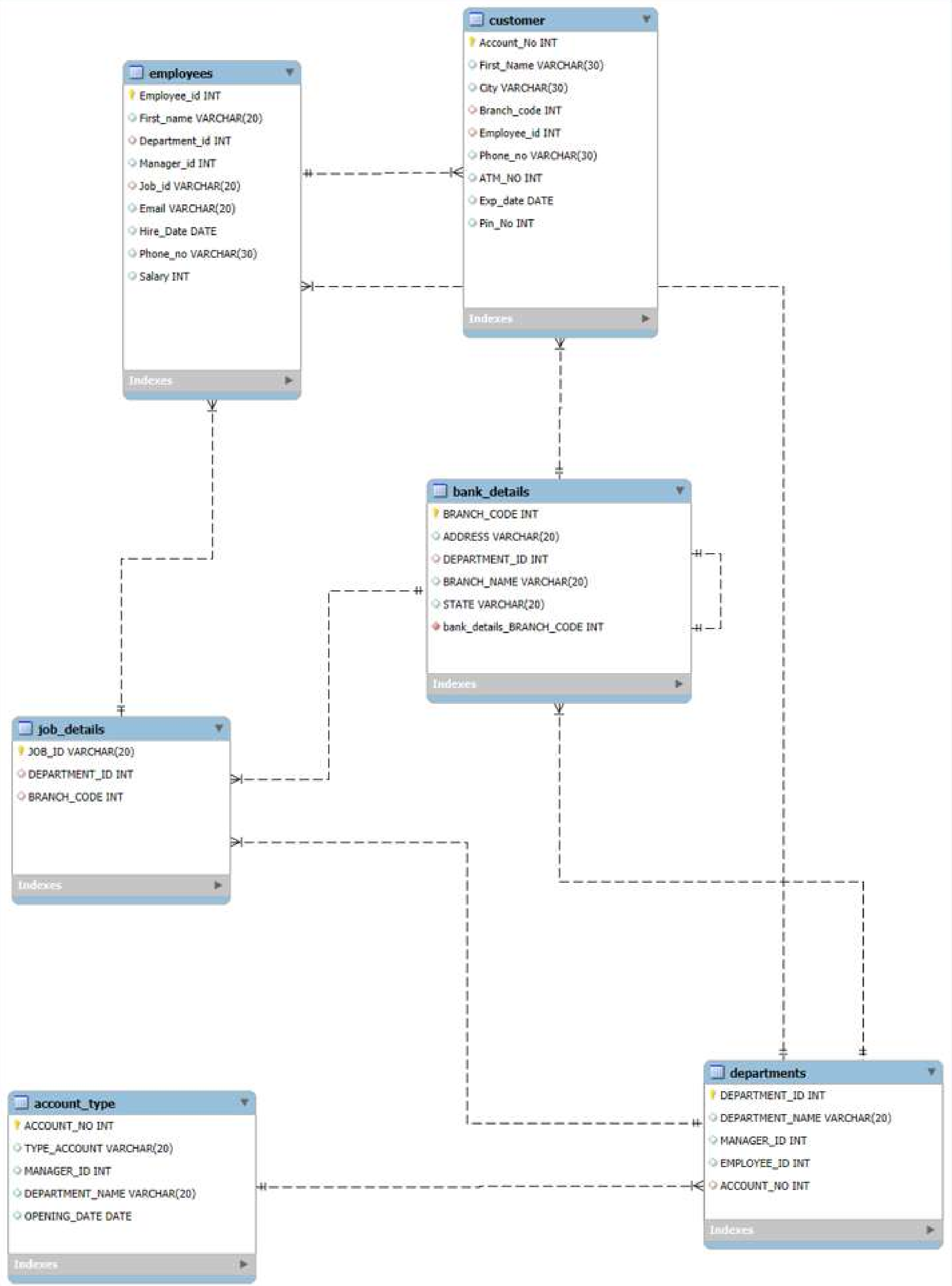
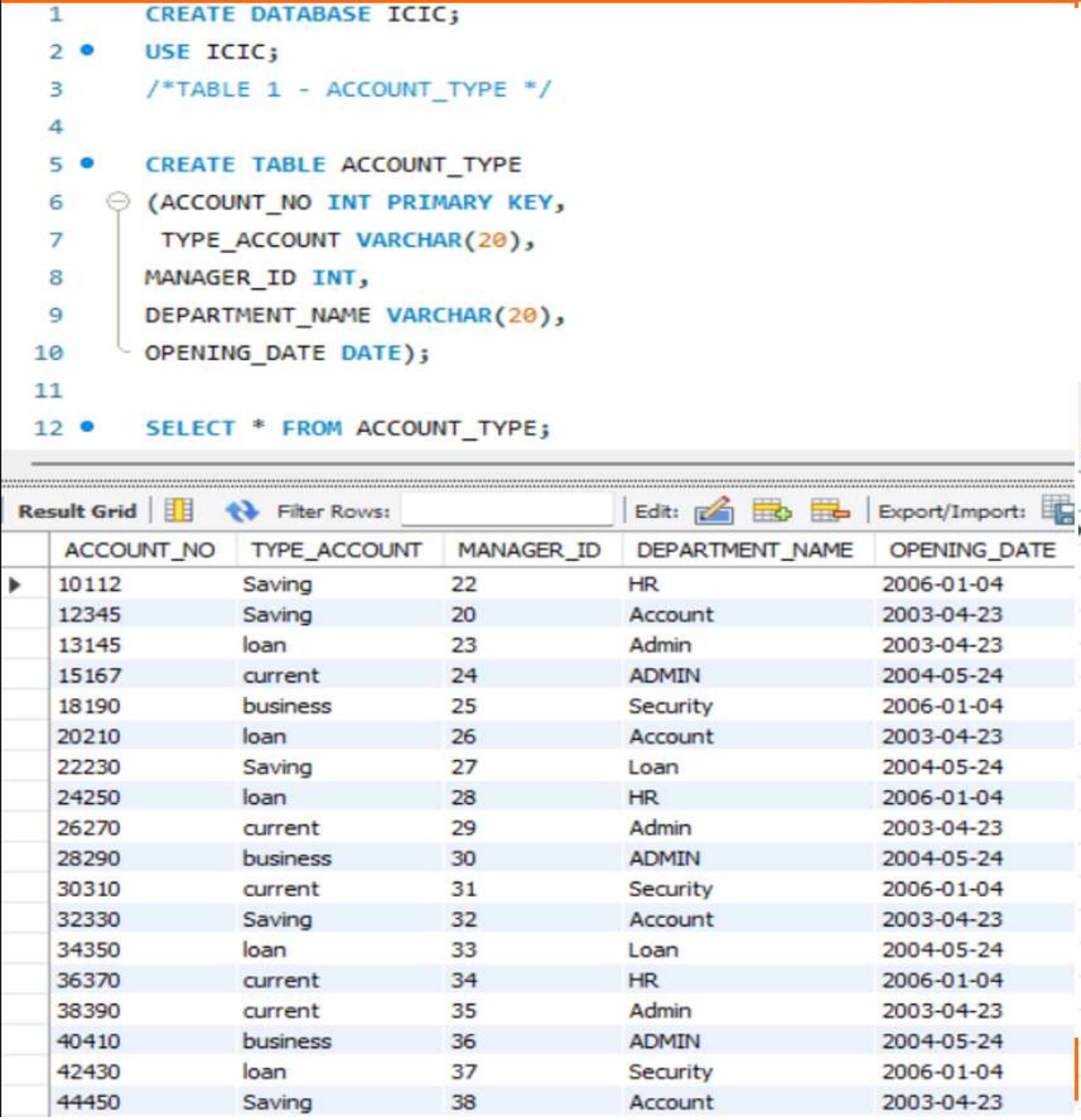
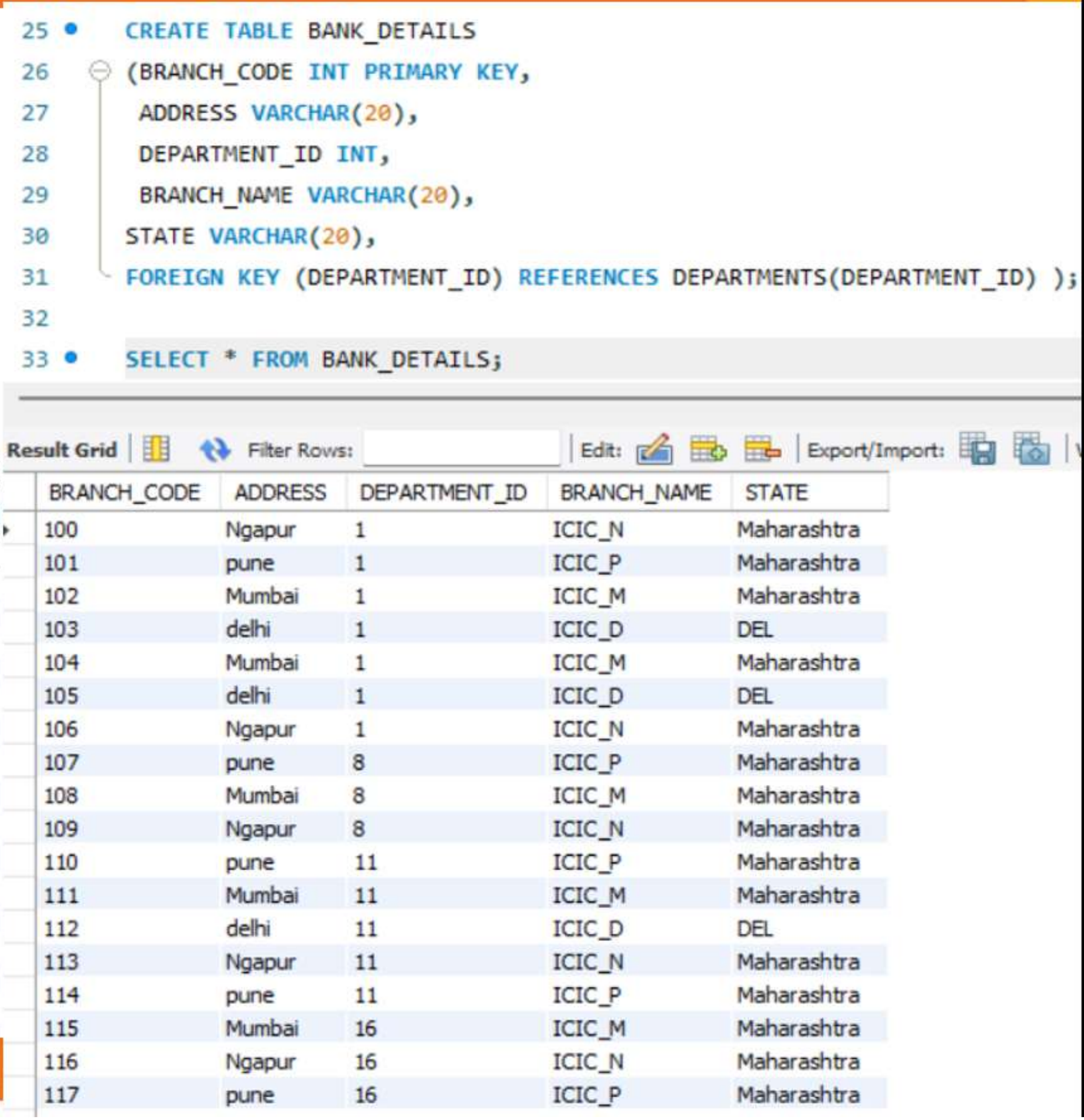

customer
Account_No INT
First_Name VARCHAR(30)
City VARCHAR(30)
Branch_code INT
Employee_id INT
Phone_no VARCHAR(30)
ATM_NO INT
Exp_date DATE
Pin_No INT
Indexes
employees
Employee_id INT
First_name VARCHAR(20)
Department_id INT
Manager_id INT
Job_id VARCHAR(20)
Email VARCHAR(20)
Hire_Date DATE
Phone_no VARCHAR(30)
Salary INT
Indexes
bank_details
BRANCH_CODE INT
ADDRESS VARCHAR(20)
DEPARTMENT_ID INT
BRANCH_NAME VARCHAR(20)
STATE VARCHAR(20)
bank_details_BRANCH_CODE INT
Indexes
job_details
JOB_ID VARCHAR(20)
DEPARTMENT_ID INT
BRANCH_CODE INT
Indexes
departments
DEPARTMENT_ID INT
DEPARTMENT_NAME VARCHAR(20)
MANAGER_ID INT
EMPLOYEE_ID INT
ACCOUNT_NO INT
Indexes
account_type
ACCOUNT_NO INT
TYPE_ACCOUNT VARCHAR(20)
MANAGER_ID INT
DEPARTMENT_NAME VARCHAR(20)
OPENING_DATE DATE
Indexes

TABLE AND DATA



```sql
CREATE DATABASE ICIC;
USE ICIC;
/*TABLE 1 - ACCOUNT_TYPE */

CREATE TABLE ACCOUNT_TYPE
(ACCOUNT_NO INT PRIMARY KEY,
 TYPE_ACCOUNT VARCHAR(20),
MANAGER_ID INT,
DEPARTMENT_NAME VARCHAR(20),
OPENING_DATE DATE);

SELECT * FROM ACCOUNT_TYPE;
```

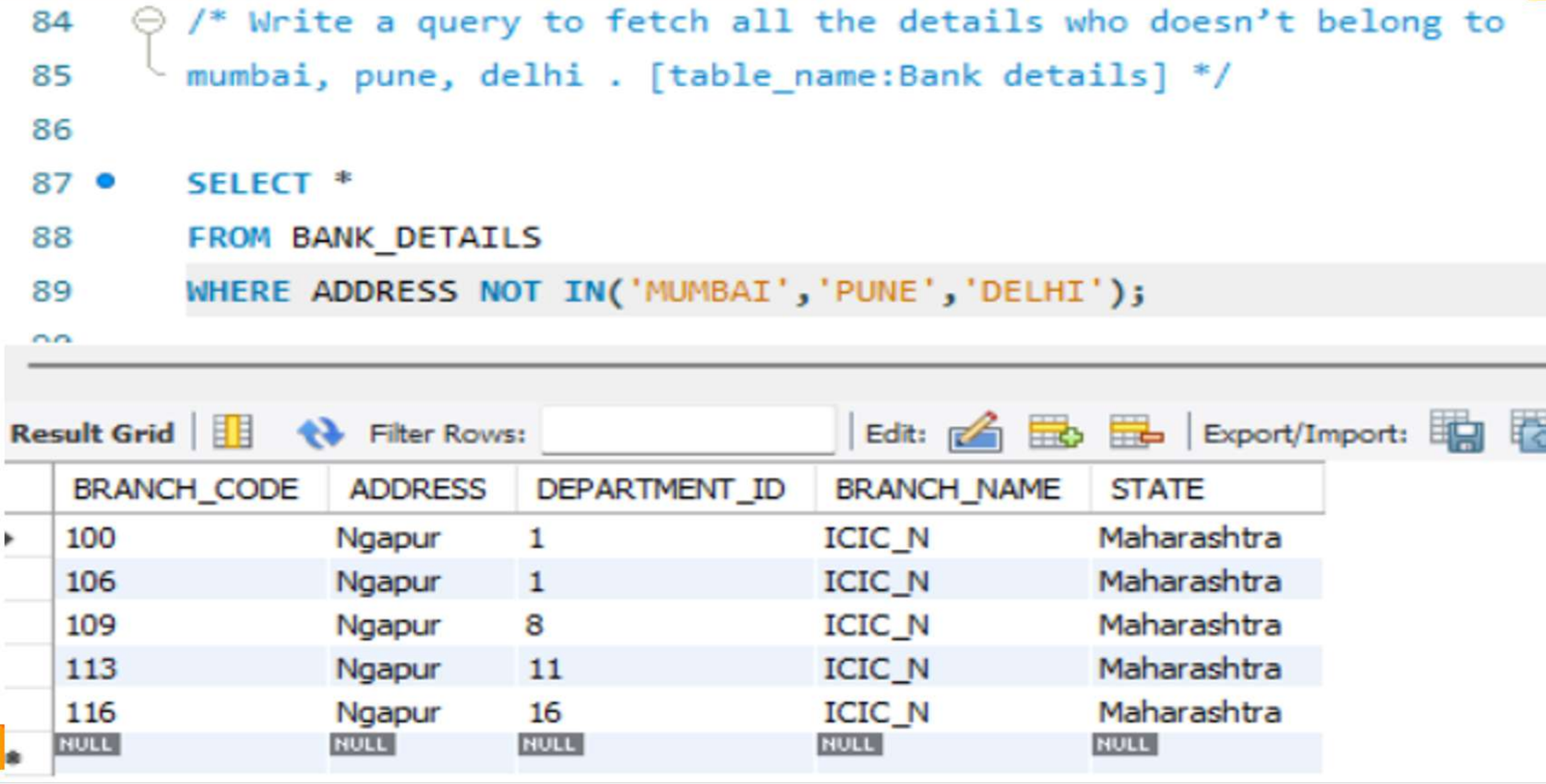
ACCOUNT_NO	TYPE_ACCOUNT	MANAGER_ID	DEPARTMENT_NAME	OPENING_DATE
10112	Saving	22	HR	2006-01-04
12345	Saving	20	Account	2003-04-23
13145	loan	23	Admin	2003-04-23
15167	current	24	ADMIN	2004-05-24
18190	business	25	Security	2006-01-04
20210	loan	26	Account	2003-04-23
22230	Saving	27	Loan	2004-05-24
24250	loan	28	HR	2006-01-04
26270	current	29	Admin	2003-04-23
28290	business	30	ADMIN	2004-05-24
30310	current	31	Security	2006-01-04
32330	Saving	32	Account	2003-04-23
34350	loan	33	Loan	2004-05-24
36370	current	34	HR	2006-01-04
38390	current	35	Admin	2003-04-23
40410	business	36	ADMIN	2004-05-24
42430	loan	37	Security	2006-01-04
44450	Saving	38	Account	2003-04-23



```sql
CREATE TABLE BANK_DETAILS
(BRANCH_CODE INT PRIMARY KEY,
 ADDRESS VARCHAR(20),
 DEPARTMENT_ID INT,
 BRANCH_NAME VARCHAR(20),
STATE VARCHAR(20),
FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENTS(DEPARTMENT_ID) );

SELECT * FROM BANK_DETAILS;
```

BRANCH_CODE	ADDRESS	DEPARTMENT_ID	BRANCH_NAME	STATE
100	Ngapur	1	ICIC_N	Maharashtra
101	pune	1	ICIC_P	Maharashtra
102	Mumbai	1	ICIC_M	Maharashtra
103	delhi	1	ICIC_D	DEL
104	Mumbai	1	ICIC_M	Maharashtra
105	delhi	1	ICIC_D	DEL
106	Ngapur	1	ICIC_N	Maharashtra
107	pune	8	ICIC_P	Maharashtra
108	Mumbai	8	ICIC_M	Maharashtra
109	Ngapur	8	ICIC_N	Maharashtra
110	pune	11	ICIC_P	Maharashtra
111	Mumbai	11	ICIC_M	Maharashtra
112	delhi	11	ICIC_D	DEL
113	Ngapur	11	ICIC_N	Maharashtra
114	pune	11	ICIC_P	Maharashtra
115	Mumbai	16	ICIC_M	Maharashtra
116	Ngapur	16	ICIC_N	Maharashtra
117	pune	16	ICIC_P	Maharashtra

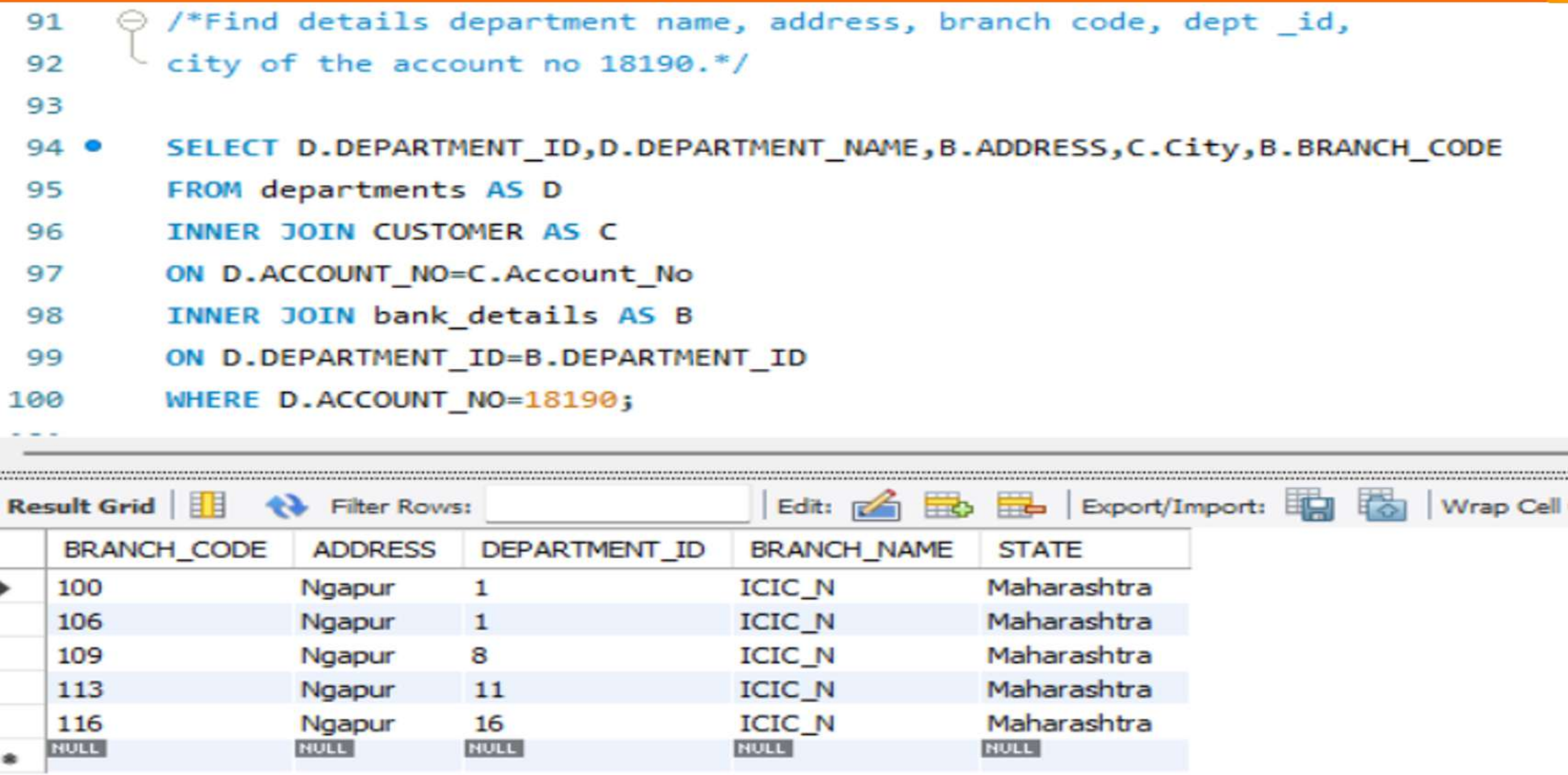
Proposed Solution



```
/* Write a query to fetch all the details who doesn't belong to
mumbai, pune, delhi . [table_name:Bank details] */

SELECT *
FROM BANK_DETAILS
WHERE ADDRESS NOT IN('MUMBAI','PUNE','DELHI');
```

BRANCH_CODE	ADDRESS	DEPARTMENT_ID	BRANCH_NAME	STATE
100	Ngapur	1	ICIC_N	Maharashtra
106	Ngapur	1	ICIC_N	Maharashtra
109	Ngapur	8	ICIC_N	Maharashtra
113	Ngapur	11	ICIC_N	Maharashtra
116	Ngapur	16	ICIC_N	Maharashtra
NULL	NULL	NULL	NULL	NULL

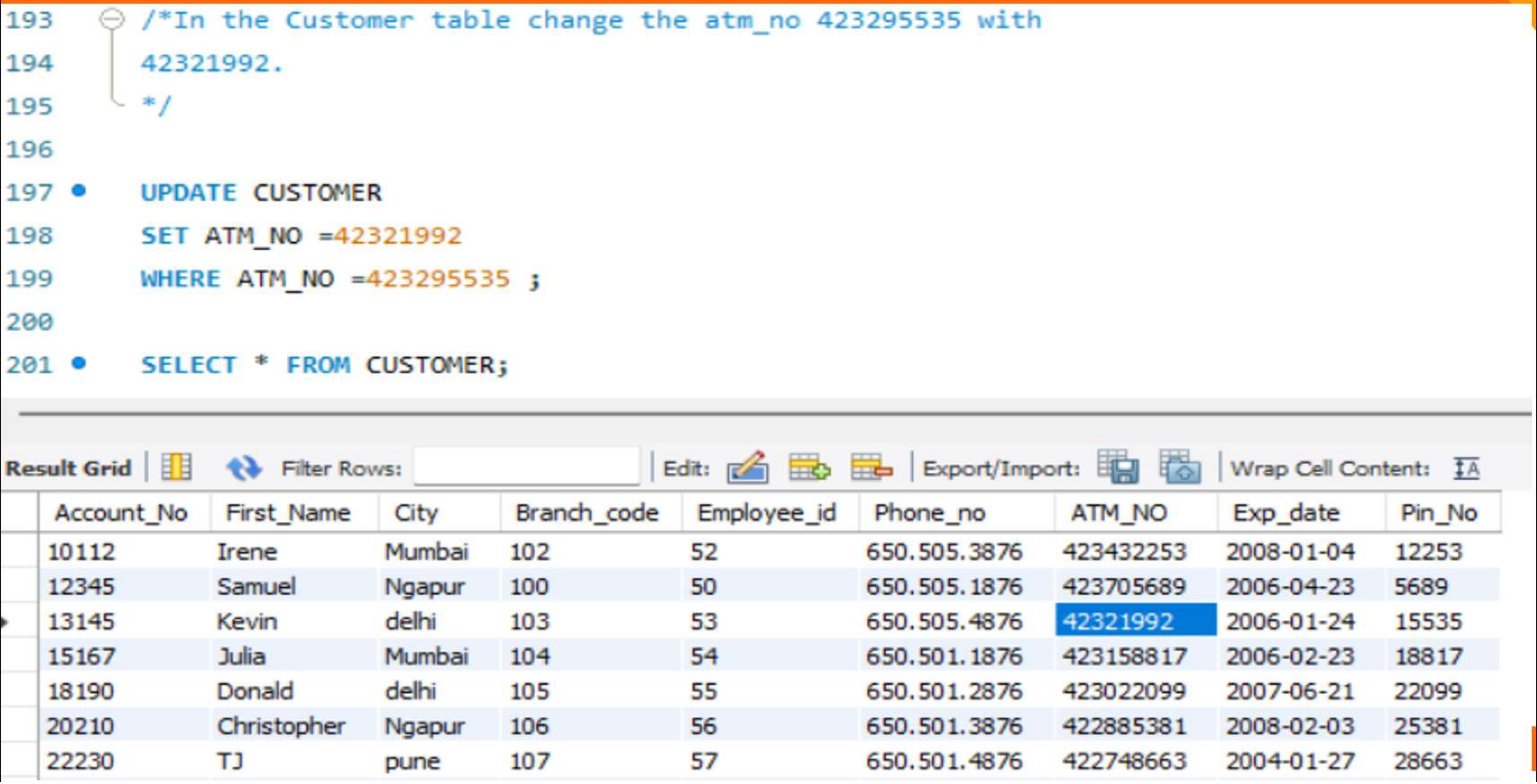
Proposed Solution



```sql
/*Find details department name, address, branch code, dept _id,
city of the account no 18190.*/

SELECT D.DEPARTMENT_ID,D.DEPARTMENT_NAME,B.ADDRESS,C.City,B.BRANCH_CODE
FROM departments AS D
INNER JOIN CUSTOMER AS C
ON D.ACCOUNT_NO=C.Account_No
INNER JOIN bank_details AS B
ON D.DEPARTMENT_ID=B.DEPARTMENT_ID
WHERE D.ACCOUNT_NO=18190;
```

BRANCH_CODE	ADDRESS	DEPARTMENT_ID	BRANCH_NAME	STATE
100	Ngapur	1	ICIC_N	Maharashtra
106	Ngapur	1	ICIC_N	Maharashtra
109	Ngapur	8	ICIC_N	Maharashtra
113	Ngapur	11	ICIC_N	Maharashtra
116	Ngapur	16	ICIC_N	Maharashtra
NULL	NULL	NULL	NULL	NULL

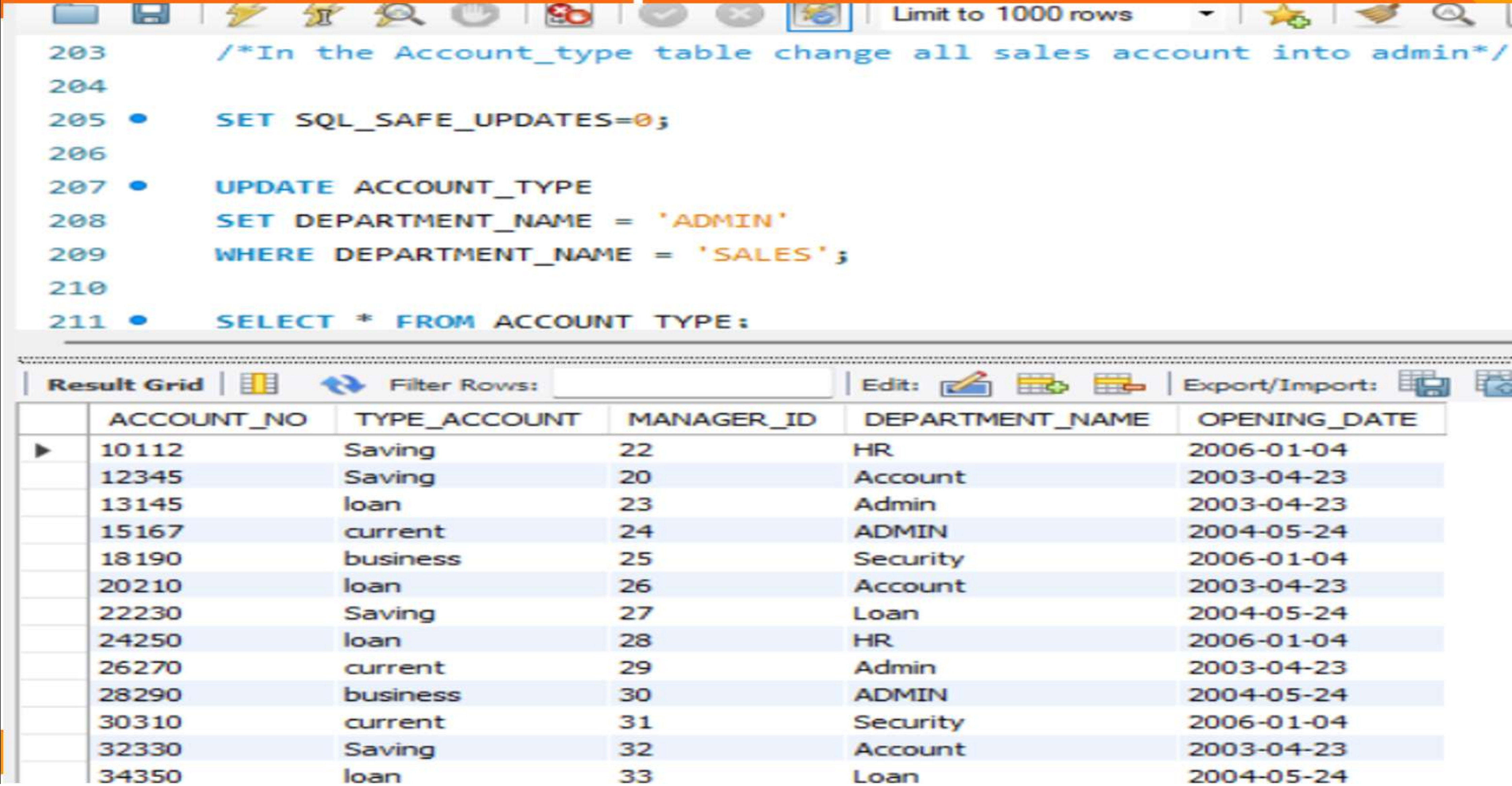
Proposed Solution



```sql
/*In the Customer table change the atm_no 423295535 with
42321992.
*/

UPDATE CUSTOMER
SET ATM_NO =42321992
WHERE ATM_NO =423295535 ;

SELECT * FROM CUSTOMER;
```

Account_No	First_Name	City	Branch_code	Employee_id	Phone_no	ATM_NO	Exp_date	Pin_No
10112	Irene	Mumbai	102	52	650.505.3876	423432253	2008-01-04	12253
12345	Samuel	Ngapur	100	50	650.505.1876	423705689	2006-04-23	5689
13145	Kevin	delhi	103	53	650.505.4876	42321992	2006-01-24	15535
15167	Julia	Mumbai	104	54	650.501.1876	423158817	2006-02-23	18817
18190	Donald	delhi	105	55	650.501.2876	423022099	2007-06-21	22099
20210	Christopher	Ngapur	106	56	650.501.3876	422885381	2008-02-03	25381
22230	TJ	pune	107	57	650.501.4876	422748663	2004-01-27	28663

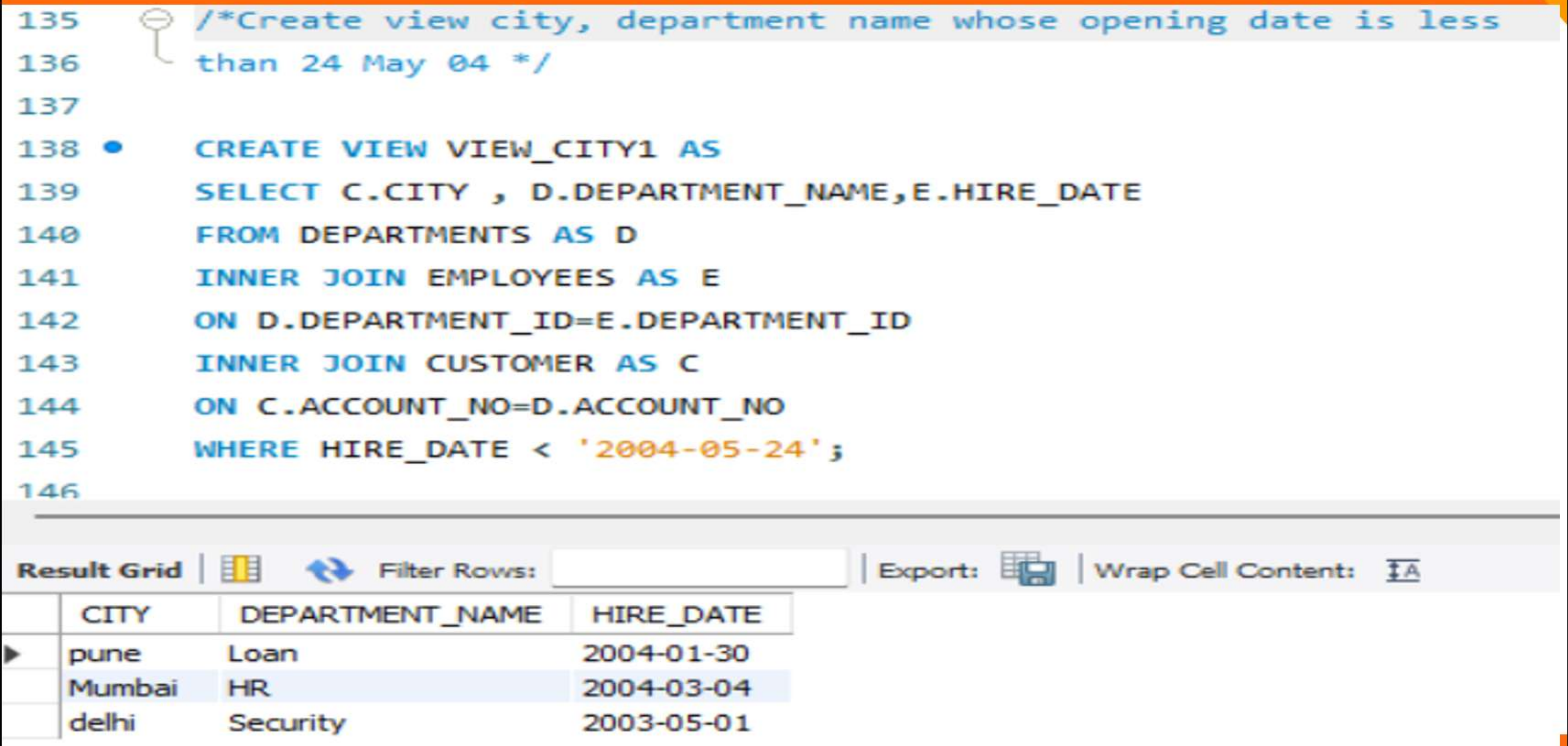
Proposed Solution



```sql
/*In the Account_type table change all sales account into admin*/

SET SQL_SAFE_UPDATES=0;

UPDATE ACCOUNT_TYPE
SET DEPARTMENT_NAME = 'ADMIN'
WHERE DEPARTMENT_NAME = 'SALES';

SELECT * FROM ACCOUNT_TYPE;
```

ACCOUNT_NO	TYPE_ACCOUNT	MANAGER_ID	DEPARTMENT_NAME	OPENING_DATE
10112	Saving	22	HR	2006-01-04
12345	Saving	20	Account	2003-04-23
13145	loan	23	Admin	2003-04-23
15167	current	24	ADMIN	2004-05-24
18190	business	25	Security	2006-01-04
20210	loan	26	Account	2003-04-23
22230	Saving	27	Loan	2004-05-24
24250	loan	28	HR	2006-01-04
26270	current	29	Admin	2003-04-23
28290	business	30	ADMIN	2004-05-24
30310	current	31	Security	2006-01-04
32330	Saving	32	Account	2003-04-23
34350	loan	33	Loan	2004-05-24

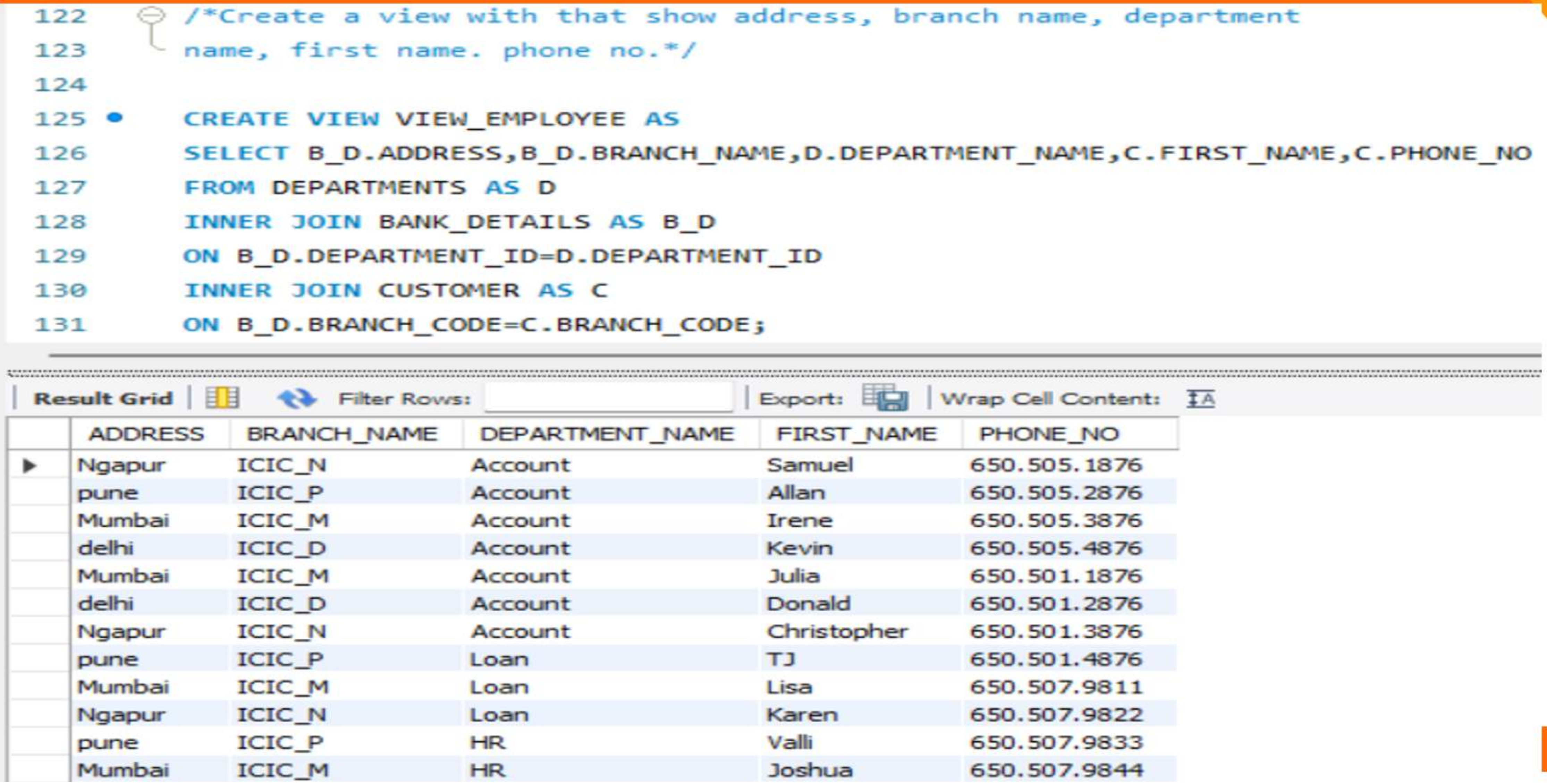
Proposed Solution



```
/*Create view city, department name whose opening date is less
than 24 May 04 */

CREATE VIEW VIEW_CITY1 AS
SELECT C.CITY , D.DEPARTMENT_NAME,E.HIRE_DATE
FROM DEPARTMENTS AS D
INNER JOIN EMPLOYEES AS E
ON D.DEPARTMENT_ID=E.DEPARTMENT_ID
INNER JOIN CUSTOMER AS C
ON C.ACCOUNT_NO=D.ACCOUNT_NO
WHERE HIRE_DATE < '2004-05-24';
```

CITY	DEPARTMENT_NAME	HIRE_DATE
pune	Loan	2004-01-30
Mumbai	HR	2004-03-04
delhi	Security	2003-05-01

Proposed Solution



```
/*Create a view with that show address, branch name, department
name, first name. phone no.*/

CREATE VIEW VIEW_EMPLOYEE AS
SELECT B_D.ADDRESS,B_D.BRANCH_NAME,D.DEPARTMENT_NAME,C.FIRST_NAME,C.PHONE_NO
FROM DEPARTMENTS AS D
INNER JOIN BANK_DETAILS AS B_D
ON B_D.DEPARTMENT_ID=D.DEPARTMENT_ID
INNER JOIN CUSTOMER AS C
ON B_D.BRANCH_CODE=C.BRANCH_CODE;
```

ADDRESS	BRANCH_NAME	DEPARTMENT_NAME	FIRST_NAME	PHONE_NO
Ngapur	ICIC_N	Account	Samuel	650.505.1876
pune	ICIC_P	Account	Allan	650.505.2876
Mumbai	ICIC_M	Account	Irene	650.505.3876
delhi	ICIC_D	Account	Kevin	650.505.4876
Mumbai	ICIC_M	Account	Julia	650.501.1876
delhi	ICIC_D	Account	Donald	650.501.2876
Ngapur	ICIC_N	Account	Christopher	650.501.3876
pune	ICIC_P	Loan	TJ	650.501.4876
Mumbai	ICIC_M	Loan	Lisa	650.507.9811
Ngapur	ICIC_N	Loan	Karen	650.507.9822
pune	ICIC_P	HR	Valli	650.507.9833
Mumbai	ICIC_M	HR	Joshua	650.507.9844

Conclusion

Title: Conclusion

- Created a working database system to manage bank operations smoothly.
- Used SQL queries to easily find and update important informatio about customers and employees.
- Improved how the bank handles data, making it easier for different departments to access and use the information.
- This project helps the bank work more efficiently, leading to better decisions and improved customer service.

Future Scope

- **Advanced Analytics**: Use AI to analyze customer behavior and predict trends.
- **Task Automation**: Automate routine tasks like loan approvals with AI.
- **Data Security**: Improve security to protect customer information.
- **Mobile Integration**: Create a mobile-friendly system for easier access.
- **Scalability**: Expand the database to support more customers and services.