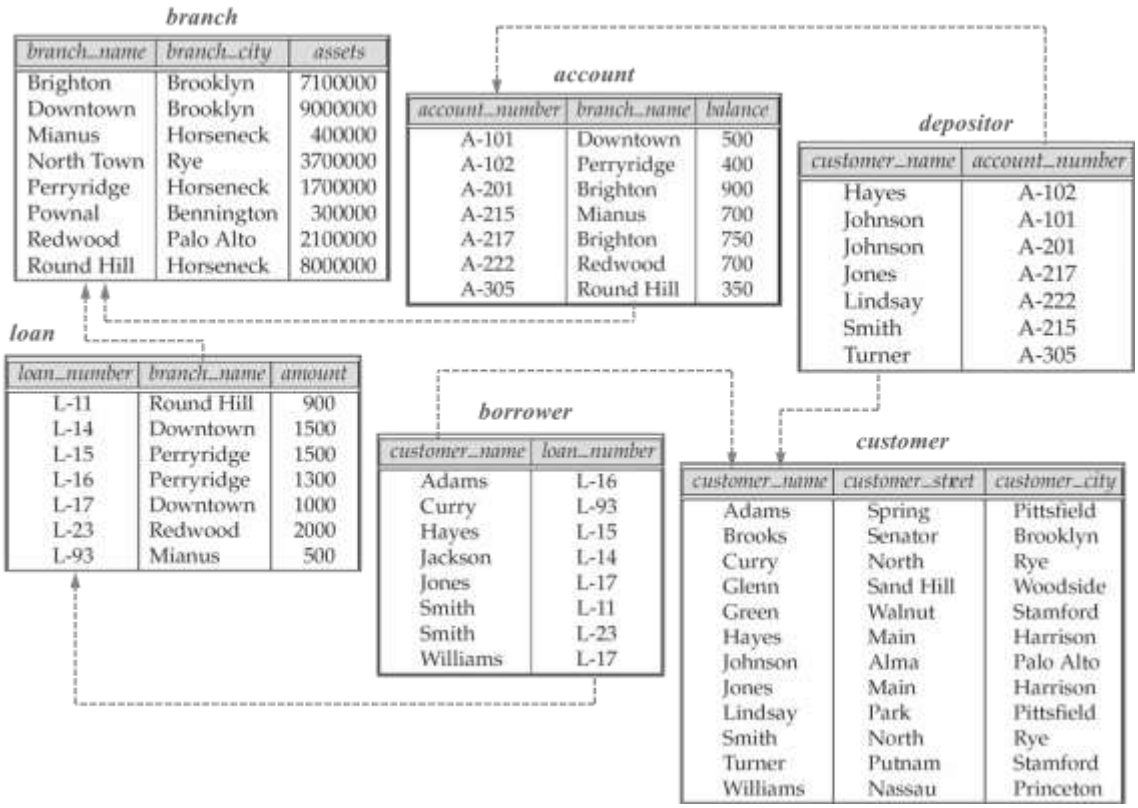
Lab Session 4: Introduction to SQL

In Lab 1, you created the simple bank database that is shown in the following figure:



branch

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

account

account_number	branch_name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

depositor

customer_name	account_number
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

loan

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Figure 1. Example database

This lab assumes that this database has already been created, and can be accessed with a tool such as the **mysql command**.

⚠ In case you have not yet created this database, you should go back to Lab 1 and follow the instructions, in particular steps 1-12.

⚠ In case you have forgotten your MySQL password, go back to Lab 1 and follow steps 3-5.

In this lab we are going to query the database using the SQL language. First we will begin with some basic experiments, and then we will move on to more interesting queries.

Part I: Some basic experiments

1. Open an SSH connection to the computing cluster at **sigma.tecnico.ulisboa.pt**
   - On Windows, try the **PuTTY** tool.
   - On MacOS or Linux, open a terminal window and use the **ssh** command**:**
     ssh sigma.tecnico.ulisboa.pt –l <your-identity>

2. At the command prompt in the cluster **sigma.tecnico.ulisboa.pt**, run the command below:
   mysql -h db. tecnico.ulisboa.pt -u istxxxxxx -p
   - The parameter **-h db.tecnico.ulisboa.pt** specifies the MySQL server.
   - The parameter **-u istxxxxxx** specifies the username.
   - The parameter **-p** requests the program to prompt your password. Give the password that you have just obtained from **mysql_reset**.

3. At the **mysql>** prompt, select your default database with the following command, replacing **istxxxxxx** by your username:
   USE istxxxxxx

4. Run the command:
   SHOW TABLES;
   To output the names of the tables in your database

5. Obtain information about the structure of the **account** table:
   DESCRIBE account;
   SHOW CREATE TABLE account;
   The meaning of the columns should be clear to you, but what about the primary key and foreign key? Find the primary key and foreign key.

6. At the **mysql>** prompt, execute the following query and inspect the contents of table account:
   SELECT * FROM account;

7. Now execute the following query and inspect the contents of table depositor:
   SELECT * FROM depositor;

8. Run the following query and explain, in your own words, what this query is doing:
   SELECT * FROM account, depositor;

9. Now explain what this query is doing:
   SELECT *
   FROM account, depositor
   WHERE account.account_number = depositor.account_number;

10. What are the differences between the result of the previous query and this one:
    SELECT * FROM account NATURAL JOIN depositor;

11. Try joining 3 tables:
    SELECT * FROM account, depositor, customer;

12. If table account has 7 rows, table depositor has 7 rows, and table customer has 13 rows, can you predict how many rows will appear in the result of the previous query?

13. Now run a similar query, but with some additional criteria:
    SELECT *
    FROM account, depositor, customer
    WHERE account.account_number = depositor.account_number
    AND depositor.customer_name = customer.customer_name;
    What information is this query showing?

14. What are the differences between the result of the previous query and this one:
    SELECT *
    FROM account NATURAL JOIN depositor NATURAL JOIN customer;

15. This query works:
    SELECT account_number
    FROM account NATURAL JOIN depositor NATURAL JOIN customer;
    But this query does not work:
    SELECT account_number
    FROM account, depositor, customer
    WHERE account.account_number = depositor.account_number
    AND depositor.customer_name = customer.customer_name;
    How can you fix the second query to make it work?

16. In the previous queries, you used mostly **SELECT *** to show all columns. However, in practice you should select only those columns which are relevant to answer the given question. For example, if the question is “Which accounts have a balance of 700?” then the query is:
    SELECT account_number FROM account WHERE balance = 700;
    As another example, if the question is “Which customers live in Brooklyn?” then the query should be:

SELECT customer_name FROM customer WHERE customer_city = 'Brooklyn';
Note that, in both cases, the column being selected is the primary key of the corresponding table. The question may also ask for a specific column (e.g. balance of table account) which is not the primary key of a table.

Part II: Querying the database

Write a single SQL query to answer each of the following questions:

 In this lab, you should avoid using natural joins. Instead, you should indicate explicitly (i.e. in the WHERE clause) which joining criteria are being used to join the tables.

1. Some customers have accounts with balance over 500€. Find their names (but, for privacy reasons, do not show their balances). There should be no repeated names in the result.
2. Some customers have loans with amounts between 1000€ and 2000€. We need to find the cities where those customers live.
3. What would be the new balance of the accounts in Perryridge, if this branch decided to charge a commission of 1% to each account? (You do not need to update the table, just show what the new balance of each account would be)
4. What is the account number and balance for each account of the customer associated with loan L-15?
5. Who are the customers who live in cities where the bank has branches? (Do not show duplicates in the result)
6. What are the assets of the branch where customer ‘Jones’ has an account?
7. In this question we are interested only in those customers with a name beginning with ‘J’ and ending with ‘s’. In which branches do these customers have an account?
8. In this question we are interested only in those customers who live in a street with 4 letters. If these customers have a loan, what is the amount of their loans? (In the results, show the customer name, the customer street, the loan numbers, and the amounts of their loans)
9. Who is the customer who has both an account and a loan in the same branch?