

Introduction to Databases

Basic SQL

Dr Paolo Guagliardo

(v26.0)

CHANGELOG

v26.0 Initial version.

Database schema. Consider the following schema:

CUSTOMER : *ID, Name, City*

ACCOUNT : *Number, Branch, CustID, Balance*

and assume that:

- the values for *ID* in CUSTOMER are all distinct (i.e., no two rows in CUSTOMER can have the same value for *ID*);
- the values for *Number* in ACCOUNT are all distinct (i.e., no two rows in ACCOUNT can have the same value for *Number*).

We will see how to write the above constraints in SQL later on in the course.

Problem 1. Write query the following queries in SQL:

- (1) Return the value of attribute *Number* of all accounts owned by customers called “John Doe”.
- (2) Return *Number* and *Branch* of all accounts owned by a customer with *ID* “xyz123”, only if there is such a customer in the CUSTOMER table.
- (3) Return *Number* and *Balance* of all overdrawn accounts in the “London” branch.
- (4) Return all pairs (*Name*, *Number*) where *Name* is the name of a customer and *Number* is the number of an account owned by that customer, such that the branch of the account is in a different city than the one where the customer lives.

Problem 2. Given an instance of the schema above, what is the answer to the following query?

```
SELECT C1.id, C1.name, C1.city
FROM   Customer C1, Customer C2
WHERE  C1.id = C2.id AND ( C1.name <> C2.name OR C1.city <> C2.city ) ;
```

Problem 3. Describe a situation in which a query keeps working without errors after a change of the given schema, but produces a different answer than what was originally intended, and discuss how the situation could be avoided with an appropriate use of table aliases and qualification of attributes.