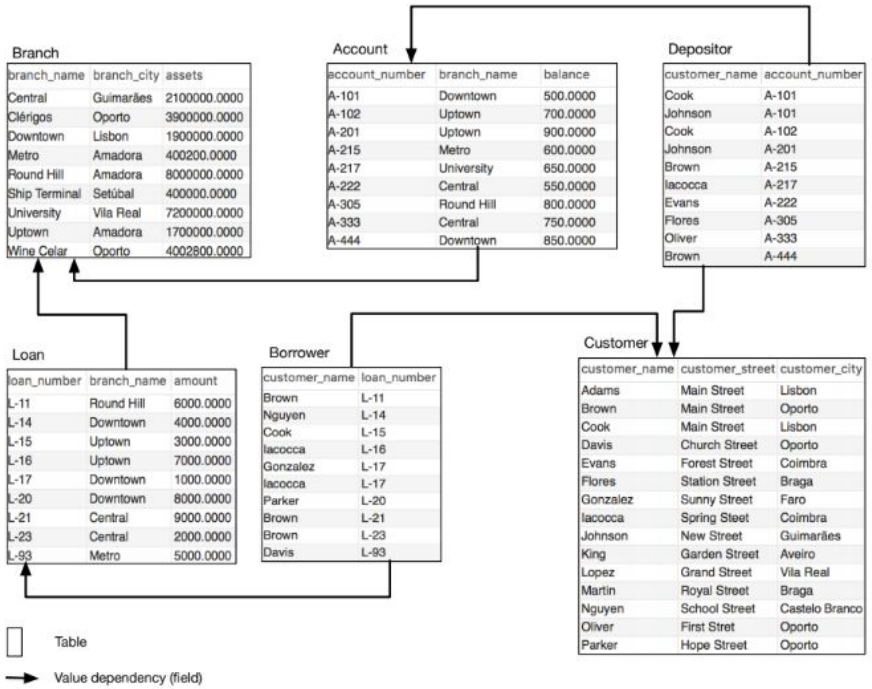
TÉCNICO LISBOA

SQL Avançado

Slides e Soluções do Laboratório 8

Base de Dados Bank



TÉCNICO LISBOA

Consultas à Base de Dados

a. Quem são os clientes que moram em cidades onde existem agências do banco?

```
SELECT customer_name
FROM customer WHERE customer_city IN
     (SELECT branch_city FROM branch);
```

ou

```
SELECT DISTINCT customer_name
FROM customer
     JOIN branch ON (branch_city = customer_city);
```

ou

```
SELECT DISTINCT customer_name
FROM customer, branch
WHERE branch_city = customer_city;
```

Consultas à Base de Dados

b. Quem são os clientes que têm um empréstimo numa agência da cidade onde moram?

```
SELECT customer_name
FROM customer
     JOIN borrower USING (customer_name)
     JOIN loan USING (loan_number)
     JOIN branch USING (branch_name)
WHERE branch_city = customer_city;
```

Consultas à Base de Dados

c. Quais são os nomes e moradas dos clientes que têm conta no banco, mas não tem empréstimos?

```
SELECT customer_name, customer_street, customer_city
FROM customer
WHERE customer_name IN (SELECT customer_name FROM depositor)
AND customer_name NOT IN (SELECT customer_name FROM borrower);
```

ou

```
SELECT c.customer_name, customer_street, customer_city
FROM customer c
WHERE EXISTS (SELECT * FROM depositor WHERE customer_name = c.customer_name)
AND NOT EXISTS (SELECT * FROM borrower WHERE customer_name = c.customer_name);
```

ou

```
SELECT DISTINCT customer_name, customer_street, customer_city
FROM depositor JOIN customer USING (customer_name)
EXCEPT
SELECT DISTINCT customer_name, customer_street, customer_city
FROM customer JOIN borrower USING (customer_name)
```

Consultas à Base de Dados

d. Quantos clientes moram na mesma cidade de uma agência onde têm conta?

```
SELECT COUNT(DISTINCT (customer_name)) AS num_cli
FROM customer
     JOIN depositor USING (customer_name)
     JOIN account USING (account_number)
     JOIN branch USING (branch_name)
WHERE branch_city = customer_city;
```

TÉCNICO LISBOA

Consultas à Base de Dados

e. Quantos clientes têm pelo menos uma conta e um empréstimo no banco?

```
SELECT COUNT(DISTINCT(customer_name)) AS num_cli
FROM depositor JOIN borrower USING (customer_name);
```

ou

```
SELECT COUNT(DISTINCT(customer_name)) AS num_cli
FROM depositor
WHERE customer_name IN
   (SELECT customer_name FROM borrower);
```

Consultas à Base de Dados

f. Qual o nome dos clientes que têm mais do que um empréstimo?

```
SELECT customer_name
FROM borrower
GROUP BY customer_name
HAVING COUNT(*) > 1;
```

ou

```
SELECT DISTINCT(customer_name)
FROM borrower b1
   JOIN borrower b2 USING (customer_name)
WHERE b1.loan_number!=b2.loan_number;
```

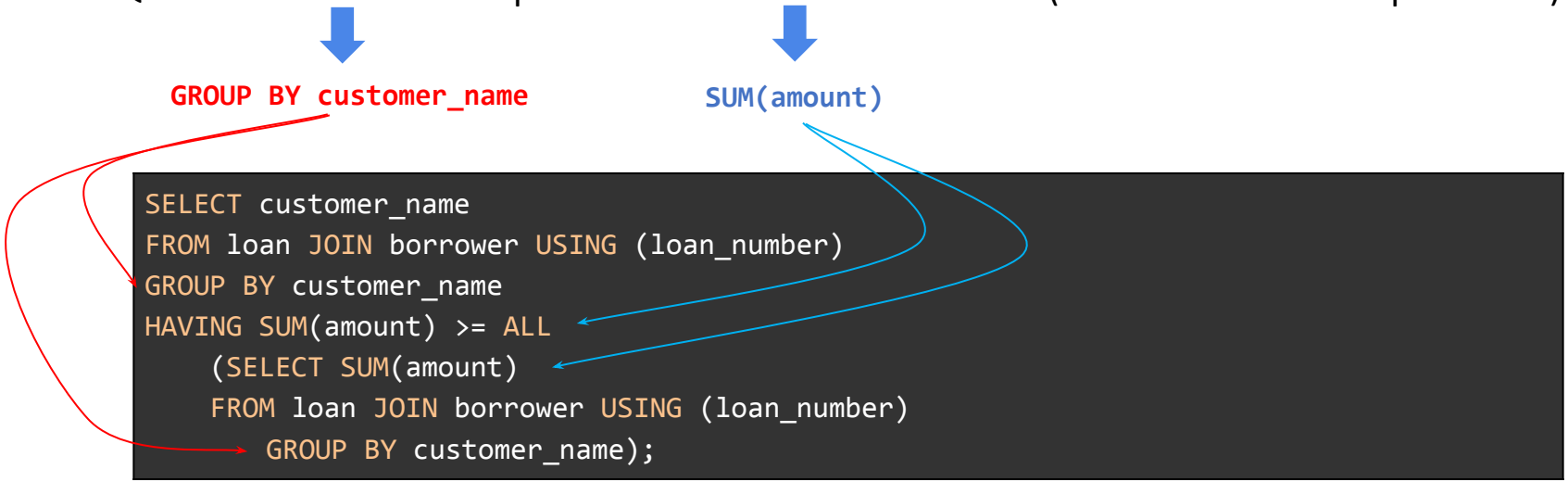
Consultas à Base de Dados

g. Liste por ordem alfabética os nomes dos clientes que têm mais de duas contas no banco.

```sql
SELECT customer_name
FROM depositor
GROUP BY customer_name
HAVING COUNT(*) > 2
ORDER BY customer_name ASC;
```

Consultas à Base de Dados

h. Qual o **nome do cliente** que **mais dinheiro deve** ao banco (no total dos seus empréstimos)?



Consultas à Base de Dados

i. Qual é o **nome e morada do cliente** que tem o **maior saldo no total das suas contas**?

GROUP BY customer_name

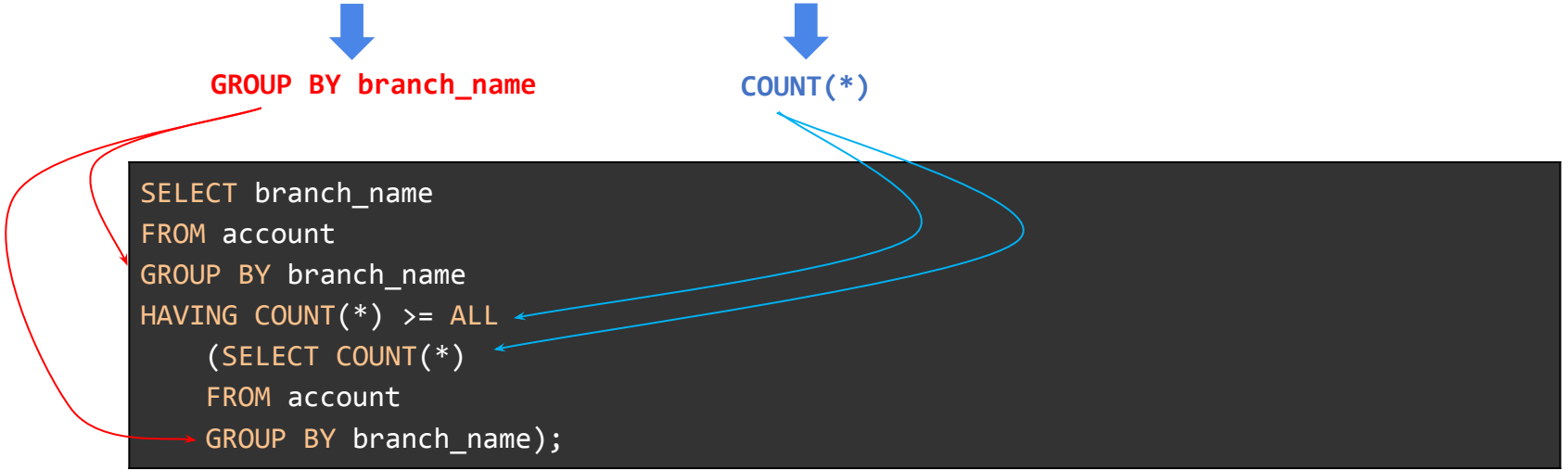
SUM(balance)

```
SELECT customer_name, customer_city, customer_street
FROM customer
   JOIN depositor USING (customer_name)
   JOIN account USING (account_number)
GROUP BY customer_name
HAVING SUM(balance) >= ALL
    (SELECT SUM(balance)
    FROM customer
       JOIN depositor USING (customer_name)
       JOIN account USING (account_number)
    GROUP BY customer_name);
```

TÉCNICO LISBOA

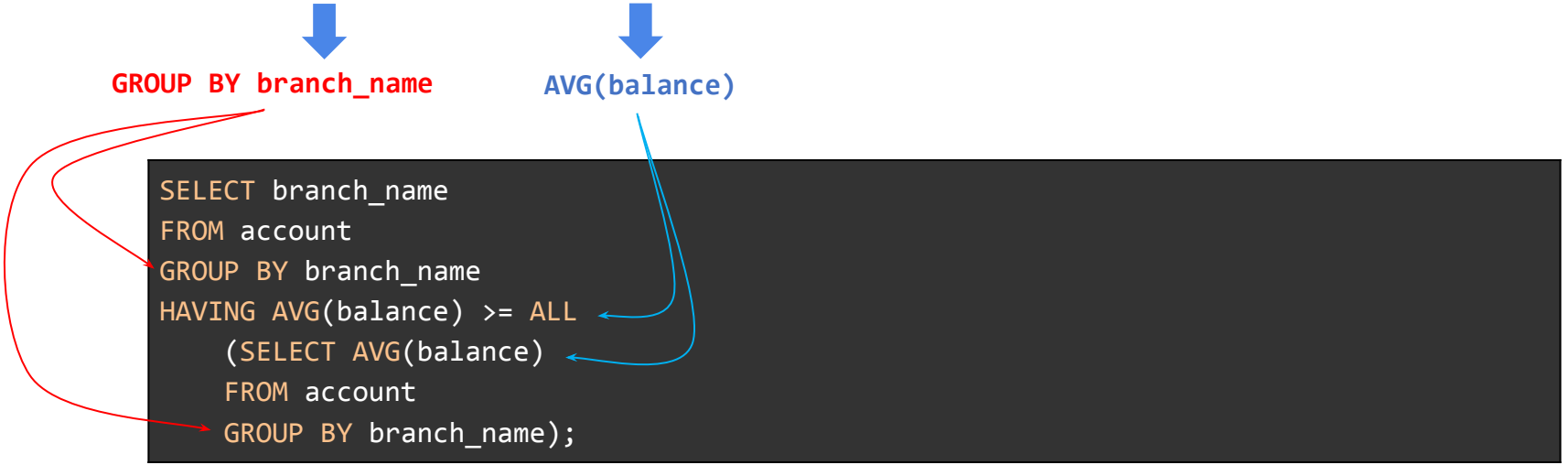
Consultas à Base de Dados

j. Qual a **agência do banco** que tem **mais contas**?



Consultas à Base de Dados

k. Qual é a **agência** cujo **saldo médio das contas** é o **maior** de todas as agências?



Consultas à Base de Dados

l. Liste o número de clientes por cada cidade onde existe pelo menos uma agência.

```
SELECT branch_city, COUNT(DISTINCT customer_name)
FROM
   (SELECT customer_name, branch_city
    FROM depositor
       JOIN account USING (account_number)
       JOIN branch USING (branch_name)
    UNION
    SELECT customer_name, branch_city
    FROM borrower
       JOIN loan USING (loan_number)
       JOIN branch USING (branch_name)
    ) AS customer_contracts
GROUP BY branch_city;
```

Consultas à Base de Dados

m. Qual é a **cidade** com **mais clientes**?

GROUP BY branch_city **COUNT(DISTINCT customer_name)**

```
WITH customer_contracts AS
   (SELECT customer_name, branch_city
   FROM depositor JOIN account USING (account_number) JOIN branch USING (branch_name)
   UNION
   SELECT customer_name, branch_city
   FROM borrower JOIN loan USING (loan_number) JOIN branch USING (branch_name))
SELECT branch_city
FROM customer_contracts
GROUP BY branch_city
HAVING COUNT(DISTINCT customer_name) >= ALL
   (SELECT COUNT(DISTINCT customer_name)
   FROM customer_contracts
   GROUP BY branch_city);
```

Consultas à Base de Dados

n. Quais são os **nomes dos clientes** que **têm contas** em **todas as agências do banco**?

Quociente **Dividendo** **Divisor**

```
SELECT DISTINCT customer_name
FROM customer c
WHERE NOT EXISTS (
   SELECT branch_name
   FROM branch
   EXCEPT
   SELECT branch_name
   FROM depositor d JOIN account USING (account_number)
   WHERE d.customer_name = c.customer_name);
```

Quociente

Divisor

Dividendo

Consultas à Base de Dados

n. Quais são os **nomes dos clientes** que **têm contas** em **todas as agências do banco**?

Evitar!

```
SELECT customer_name
FROM depositor JOIN account USING (account_number)
GROUP BY customer_name
HAVING COUNT(DISTINCT (branch_name)) =
   (SELECT COUNT(*) FROM branch);
```

Consultas à Base de Dados

o. Quais são os **nomes dos clientes** que **têm contas** em **todas as agências da cidade onde moram**?

Quociente **Dividendo** **Divisor**

```
SELECT DISTINCT customer_name
FROM customer c
WHERE NOT EXISTS (
   SELECT branch_name
   FROM branch
   WHERE branch_city = c.customer_city
   EXCEPT
   SELECT branch_name
   FROM depositor d JOIN account USING (account_number)
   WHERE d.customer_name = c.customer_name);
```

Quociente

Divisor

Dividendo