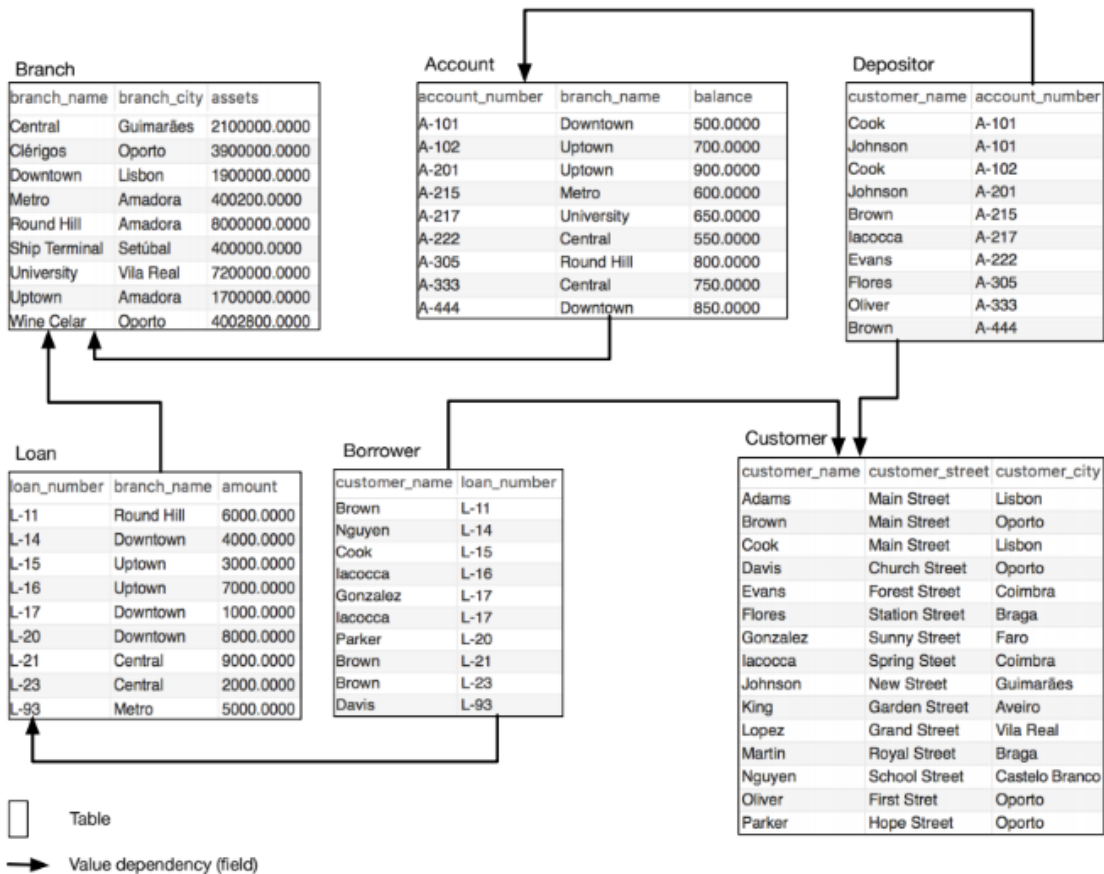


Bases de Dados

Lab 02: Consultas em SQL (Parte 1) [Solução]



Escreva uma consulta em SQL para responder a cada uma das seguintes questões:

(a) Quem são os clientes que têm contas com saldo superior a 500€?

Solução com produto cartesiano:

```
SELECT DISTINCT d.customer_name
FROM depositor d, account a
WHERE d.account_number = a.account_number
AND a.balance > 500;
```

Solução com join:

```
SELECT DISTINCT customer_name
FROM depositor INNER JOIN account USING (account_number)
WHERE a.balance > 500;
```

- (b) Em que cidades moram os clientes que têm empréstimos entre 1000€ e 2000€?

Solução com produto cartesiano e operadores de comparação:

```
SELECT DISTINCT c.customer_city
FROM customer AS c, borrower AS b, loan as l
WHERE c.customer_name = b.customer_name
      AND b.loan_number = l.loan_number
      AND l.amount <= 2000
      AND l.amount >= 1000;
```

Solução com join e predicado “between... and...”:

```
SELECT DISTINCT customer_city
FROM customer INNER JOIN borrower USING (customer_name)
      INNER JOIN loan USING (loan_number)
WHERE l.amount BETWEEN 1000 AND 2000;
```

- (c) Quais seriam os novos saldos das contas na agência de ‘Downtown’, se esta oferecesse um bônus de 10% sobre o saldo atual dos seus clientes?

Solução:

```
SELECT account_number, balance*1.1 AS newBalance
FROM account
WHERE branch_name = 'Downtown';
```

- (d) Qual é o saldo de todas as contas do cliente que tem o empréstimo L-15?

Solução com produto cartesiano:

```
SELECT DISTINCT a.account_number, a.balance
FROM account a, depositor d, borrower b
WHERE a.account_number = d.account_number
      AND d.customer_name = b.customer_name
      AND b.loan_number = 'L-15';
```

Solução com join:

```
SELECT DISTINCT account_number, balance
FROM account INNER JOIN depositor USING (account_number)
      INNER JOIN borrower USING (customer_name)
WHERE loan_number = 'L-15';
```

- (e) Quais são as agências onde têm conta os clientes cujo nome começa por 'J' e acaba em 'n'?

Solução com produto cartesiano:

```
SELECT DISTINCT a.branch_name
FROM account a, depositor d
WHERE a.account_number = d.account_number
      AND d.customer_name LIKE 'J%n';
```

Solução com join:

```
SELECT DISTINCT branch_name
FROM account INNER JOIN depositor USING (account_number)
WHERE d.customer_name LIKE 'J%n';
```

- (f) Quais são as quantias dos empréstimos de todos os clientes que moram numa cidade cujo nome tem exatamente 6 caracteres?

Solução com produto cartesiano e string pattern matching:

```
SELECT DISTINCT l.loan, l.amount
FROM loan l, borrower b, customer c
WHERE l.loan_number = b.loan_number
      AND b.customer_name = c.customer_name
      AND c.customer_city LIKE '_____';
```

Solução com join e função length:

```
SELECT DISTINCT loan, amount
FROM loan INNER JOIN borrower USING (loan_number)
      INNER JOIN customer USING (customer_name)
WHERE LENGTH(c.customer_city) = 6;
```

- (g) Quais são as quantias dos empréstimos de todos os clientes que moram numa cidade cujo nome tem pelo menos um espaço no meio (e não no início nem no final)?

Solução com produto cartesiano:

```
SELECT DISTINCT l.loan_number, l.amount
FROM loan l, borrower b, customer c
WHERE l.loan_number = b.loan_number
      AND b.customer_name = c.customer_name
      AND c.customer_city LIKE '% %_'
      AND c.customer_city NOT LIKE '% '
      AND c.customer_city NOT LIKE '% %';
```

Solução com join:

```
SELECT DISTINCT loan_number, amount
FROM loan INNER JOIN borrower USING (loan_number)
      INNER JOIN customer USING (customer_name)
WHERE customer_city LIKE '% %'
      AND customer_city NOT LIKE '% '
      AND customer_city NOT LIKE '% %';
```

- (h) Quais os ativos (“assets”) das agências onde o Johnson tem conta?

Solução com produto cartesiano:

```
SELECT DISTINCT b.branch_name, b.assets
FROM branch b, account a, depositor d
WHERE b.branch_name = a.branch_name
      AND a.account_number = d.account_number
      AND d.customer_name = 'Johnson';
```

Solução com join:

```
SELECT DISTINCT branch_name, assets
FROM branch INNER JOIN account USING (branch_name)
      INNER JOIN depositor USING (account_number)
WHERE d.customer_name = 'Johnson';
```

- (i) Quem são os clientes que têm um empréstimo numa agência da mesma cidade onde moram?

Solução com produto cartesiano:

```
SELECT DISTINCT c.customer_name
FROM customer c, borrower bo, loan l, branch br
WHERE c.customer_name = bo.customer_name
      AND bo.loan_number = l.loan_number
      AND l.branch_name = br.branch_name
      AND c.customer_city = br.branch_city;
```

Solução com join:

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

Outra solução SÓ com join:

```
SELECT DISTINCT customer_name
FROM customer INNER JOIN borrower USING (customer_name)
      INNER JOIN loan l USING (loan_number)
      INNER JOIN branch b ON (l.branch_name = b.branch_name
      AND customer_city = branch_city);
```

Notem que a query é circular, juntando N tabelas com N cláusulas (em vez das habituais N-1 cláusulas) pelo que podemos:

- Fazer quaisquer N-1 joins do círculo e verificar o último join no WHERE.
- Usar o ON para expressar duas cláusulas num join, fechando o círculo sem necessidade de WHERE

- (j) Qual é a quantia total em saldos de contas existentes em agências da cidade de Lisboa ('Lisbon')?

Solução com produto cartesiano:

```
SELECT SUM(balance) AS total_balance
FROM account a, branch b
WHERE b.branch_name = a.branch_name
      AND b.branch_city = 'Lisbon';
```

Solução com join:

```
SELECT SUM(balance) AS total_balance
FROM account INNER JOIN branch USING (branch_name)
WHERE b.branch_city = 'Lisbon';
```

- (k) Quem são os clientes que moram em cidades onde existem agências do banco?

Solução com produto cartesiano:

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

Solução com join:

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