

1. Projeção

Escreva uma consulta à tabela account que retorne os IDs dos funcionários que abriram contas (com account.open_emp_id). Inclua uma única linha para cada funcionário específico.

```
SELECT DISTINCT open_emp_id  
FROM account  
WHERE open_emp_id IS NOT NULL;
```

2. Seleção

Recupere o ID da conta (account.account_id), o ID do cliente (account.cust_id) e o saldo disponível de todas as contas (account.avail_balance) cujo status seja igual a ACTIVE e cujo saldo disponível seja superior a R\$2.500.

```
SELECT account_id, cust_id, avail_balance  
FROM account  
WHERE status = 'ACTIVE' AND avail_balance > 2500;
```

3. Agrupamento e Agregação

Retorne a data de posse do funcionário (employee.start_date) mais antigo para cada departamento (employee.department).

```
SELECT dept_id, MIN(start_date) AS data_mais_antiga  
FROM employee  
GROUP BY dept_id;
```

4- Ordenação

Recupere o ID, o nome e o sobrenome de todos os funcionários (employee.emp_id, employee.fname, employee.lname). Ordene por primeiro nome e, depois, pelo sobrenome.

```
SELECT emp_id, fname, lname  
FROM employee  
ORDER BY fname ASC, lname ASC;
```

5. União

Escreva uma consulta que encontre nomes e sobrenomes de todos os clientes (individual.fname, individual.lname), juntamente com nomes e sobrenomes de todos os funcionários (employee.fname, employee.lname).

```
SELECT fname, lname  
FROM individual  
UNION  
SELECT fname, lname  
FROM employee;
```

6- Intersecção

Escreva uma consulta composta que retorne os IDs dos funcionários (employee.emp_id), que são também superiores (employee.superior_emp_id).

```
SELECT emp_id  
FROM employee  
INTERSECT  
SELECT superior_emp_id  
FROM employee  
WHERE superior_emp_id IS NOT NULL;
```

7- Diferença

Escreva uma consulta composta que retorne as cidades de clientes. (customer.city) onde não há agência (branch.city).

```
SELECT city  
FROM customer  
EXCEPT  
SELECT city
```

FROM branch;