SQL- Estrutura Básica

- Uma consulta SQL básica tem a forma:

  select $A_1, A_2, ..., A_n$
  from $r_1, r_2, ..., r_m$
  where P

  - A_i*s* representam atributos
  - r_i*s* representam relações
  - P é um predicado.

- O resultado de uma consulta SQL é uma relação.

A cláusula select

- Listar os nomes de todas as agências de um banco

  select nome_agencia
  from *banco*

- Um asterisco na cláusula select denota “todos os atributos”

  select *
  from *banco*

- NOTA: O SQL não permite o carácter ‘-’ nos nomes, portanto deverá utilizar, por exemplo, nome_*banco* em vez de nome-*banco* num sistema existente.
- NOTA: As maiúsculas e minúsculas não são distinguidas em nomes da linguagem SQL.
  - Poderá utilizar maiúsculas nos sítios onde utilizamos **bold**.

A cláusula select (cont.)

- O SQL permite duplicados nas relações e nos resultados de consultas.
- Para forçar a eliminação de duplicados, inserir a palavra-chave **distinct** após **select.**
  Apresentar os nomes de todos os balcões onde foram efectuados empréstimos, sem repetições

  select distinct *nome_agencia*
  from *Emprestimo*

- A palavra-chave **all** indica que os duplicados não devem ser removidos.

  select all nome_*banco*
  from *Emprestimo*

A cláusula select (cont.)

- A cláusula **select** pode conter expressões aritméticas envolvendo as operações, +, –, ∗, and /, com argumentos constantes ou atributos de tuplos. Dependendo das implementações, encontram-se normalmente definidas uma biblioteca de funções.

- A consulta:

  select Num_emprestimo*, nome_balcao, valor* ∗ 100
  from Emprestimo

  devolve uma relação idêntica à relação *Emprestimo*, excepto que o atributo valor é multiplicado por 100.

A cláusula where

- A cláusula **where** é formada por um predicado envolvendo atributos de relações que aparecem na cláusula **from**.
- Para encontrar os números de contas da agência da Caparica com saldos superiores a 100.

  select num_*conta*
  from contas
  where nome_*balcao* = *'*Caparica*'* **and** saldo > 100
- Os resultados de comparações podem ser combinados por intermédio dos conectivos lógicos **and, or,** e **not.**
- Podem-se aplicar comparações ao resultado de expressões aritméticas.

A cláusula where (cont.)

- A linguagem SQL possui um operador de comparação **between** para especificar condições em que um valor deve estar contido num intervalo de valores (incluindo os seus extremos).

- Apresentar os números dos empréstimos de montantes entre \$90,000 e \$100,000 (ou seja, ≥\$90,000 e ≤\$100,000)

  select *num_emprestimo*
  from *Emprestimos*
  where *valor* **between** 90000 **and** 100000

- Para negar a condição pode-se colocar o conectivo **not** antes de **between**.

Operações com Cadeias de Caracteres

- O SQL inclui um mecanismo de concordância de padrões para comparações envolvendo cadeias de caracteres. Os padrões são descritos recorrendo a dois caracteres especiais:
  - percentagem(%). O carácter % concorda com qualquer subcadeia.
  - sublinhado (_). O carácter _ concorda com qualquer carácter.
- Listar todos os clientes cuja rua inclua a subcadeia “Main”.

```
select nome_cliente
from clientes
where nome_rua like ‘%Main%’
```

A cláusula from

- A cláusula **from** indica as relações a consultar na avaliação da expressão.
- Listar o nome, número de empréstimo e montante de todos os clientes que efectuaram um empréstimo na agência da Caparica.

select Clientes.*, *valor*
from *Clientes, Emprestimos*
where *Clientes.num_Emprestimo = Emprestimos.num_Emprestimo*
and nome_*agencia* = ‘Caparica’

A operação de Renomeação

- A linguagem SQL permite a renomeação de relações e atributos recorrendo à cláusula **as** :

  nome_*antigo* **as** *novo_nome*

- Listar o nome, número de empréstimo e montante de todos os clientes, renomeando o nome da coluna num_*emprestimo* para *id_emprestimo.*

  select *nome_cliente, clientes.num_emprestimo* **as** *id_emprestimo, valor*
  from Clientes*, Emprestimos*
  where Clientes*.num_emprestimo = Emprestimos.num_emprestimo*

- Caso se pretenda utilizar um nome com espaços, esse nome deverá ser colocado entre **aspas**.

Ordenando os tuplos

- Listar em ordem alfabética os nomes de todos os clientes que possuem um empréstimo na agência de Almada

  select distinct *nome_cliente*
  from *Clientes, Emprestimos*
  where *Clientes.num_Emprestimo* = *Emprestimos.num_Emprestimo* **and** *nome_agencia* = 'Almada'
  order by *nome_cliente*

- Pode-se especificar **desc** para ordenação descente ou **asc** para ordenação ascendente, para cada atributo; por omissão, assume-se ordem ascendente.
  - E.g. **order by** nome_*cliente* **desc**
- Pode-se ter mais do que uma chave de ordenação, separando-as com vírgulas

Funções de Agregação

- Estas funções aplicam-se a multiconjuntos de valores de uma coluna de uma relação, devolvendo um valor

  avg: valor médio
  min: valor mínimo
  max: valor máximo
  sum: soma dos valores
  count: número de valores

Funções de Agregação (cont.)

- Determinar o saldo médio das contas na agência de Almada.

  select avg *(saldo)*
  from *conta*
  where *nome_agencia* = ‘Almada’

- Calcular o número de clientes.

  select count (*)
  from *clientes*

- Encontrar o número de depositantes do banco.

  select count (distinct nome_*cliente)*
  from *depositantes*

Funções de agregação – Group By

- Listar o número de depositantes por agência.

  select nome_*agencia,* **count (distinct** *nome_cliente)*
  from *depositantes, contas*
  where *depositantes.num_conta = contas.num_conta*
  group by nome_*agencia*

Nota: Atributos na cláusula **select** fora de funções de agregação têm de aparecer na lista **group by**

Nota: Se aparecer mais do que um atributo em **group by**, então cada grupo é formado pelos tuplo com valores iguais *em todos* esses os atributos

Funções de Agregação – Cláusula Having

- Listar os nomes de todas as agências cujo valor médio dos saldos das contas é superior a $1,200.

  Select nome_agencia, **avg** *(saldo)*
  from *contas*
  group by *nome_agencia*
  having avg *(saldo)* > 1200

  Nota: predicados na cláusula **having** são aplicados depois da formação dos grupos, enquanto que os predicados na cláusula **where** são aplicados antes da formação dos grupos.