Prof. Márcio Santos
http://www.marcio.site.vu
informarcio@gmail.com

SQL - STRUCTURED QUERY LANGUAGE

1 - Evolução histórica

A história da linguagem SQL começa em Junho de 1970 com a publicação por E. F. Cood, no ACM journal, de um artigo intitulado "A Relational Model of Data for Large Shared Data Banks".

A primeira implementação da linguagem SEQUEL foi desenvolvida pela IBM e tinha por objectivo a implementação do modelo de Cood. A evolução desta linguagem deu origem ao SQL.

A primeira implementação comercial de SQL foi realizada pela Relational Software, Inc., hoje conhecida por Oracle Corporation.

Nos dias de hoje, a linguagem SQL é considerada um Standard dos Sistemas Gestores de Base de Dados Relacionais. Por isso, todos os fabricantes a integram nos seus produtos.

Existem, até ao momento, 5 gerações de linguagem:

1ª geração - Código Máquina
2ª geração - Assembley
3ª geração -Pascal, C, Cobol, Fortran, Basic
4ª geração - SQL
5ª geração - C++, Java, Delphi, Visual Basic.

Embora se possa pensar que o SQL resulta de uma evolução das linguagens de 3ª geração, isso não é verdade.

As características abreviadas das linguagens de 3ª geração são:

- Existências de variáveis, vectores, ...
- Existências de instruções condicionais (If, Switch, Case)
- Existência de ciclos (For, While, Do..While, Repeat... Until)
- Possibilidade de escrita de funções e procedimentos.

Ora nenhuma destas características está presente na linguagem SQL, sendo uma linguagem que, pela sua simplicidade, não se destina apenas a informáticos, mas a todos os potenciais utilizadores de SGBDs.

2 - Definição

A SQL (Structured Query Language) é uma linguagem de interrogação para o sistema relacional. Sendo que, embora possua características que permitam criar, alterar e remover todas os componentes de uma base de dados, como tabelas, consultas, chaves, etc., será apresentada como linguagem de manipulação de dados.

A SQL que será apresentada, baseia-se na estrutura de funcionamento de um SGBDR em plataforma Windows, o Microsoft Access. No entanto, ao longo destas notas, serão apresentados os confrontos com outros SGBDR distintos: O Sql Server (Microsoft), o Sql AnyWhere (Sybase) e o Oracle (Oracle).

Estrutura básica de expressão de SQL:

SELECT (projecção da álgebra relacional)
FROM (lista das relações)
WHERE (predicado da selecção)

Ou seja:

SELECT a_1, a_2, a_3, ...a_n (campos pretendidos)
FROM r_1, r_2, r_3, ...r_n (tabelas dos campos pretendidos)
WHERE p (condições a serem satisfeitas)

Sendo que a cláusula WHERE é opcional.

Pretende-se seleccionar os nomes de balcões existentes na relação (A-CONTAS).

```
SELECT nome_balcão
FROM a-contas
```

Nota: Nas linguagens formais (álgebra relacional), usa-se a noção matemática de conjuntos para exprimir relações, pelo que não aparecem tuplos repetidos. Em SQL é possível a duplicação nas relações.

3 - Comando SELECT (Selecção Simples)

A interrogação de qualquer Base de Dados Relacional faz-se sempre utilizando o comando SELECT. A sintaxe do comando SELECT é a seguinte:

```
SELECT campo1, campo2, ... campon
FROM tabela 1 , tabela 2, .... tabela n
[WHERE Condição ]
[GROUP BY .....]
[HAVING …….]
[ORDER BY …….]
```

Os parêntesis rectos, em sintaxes de comandos, indicam que essa componente é opcional, ou seja poderá ou não ser utilizada.

3.1 - Seleccionar todos os registos

Considerando uma tabela Postal, com os atributos Código e Local, pretende-se seleccionar todos os registos da tabela Postal.

Access	Sql Server	Oracle	SyBase

```
SELECT Código, Local
FROM Postal
SELECT Código, Local   FROM Postal
```

Depois de executado o comando, o resultado obtido será:

Código	Local
1000	LISBOA
1100	LISBOA
1500	LISBOA
2000	SANTARÉM
2040	RIO MAIOR
2300	TOMAR
4000	PORTO

3.2 - Seleccionar todas as colunas

Considerando uma tabela Pessoa, com os atributos Id, Nome, Idade, Salário, Telefone e Cod_postal, pretende-se seleccionar todas as pessoas da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT  * FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade	Salário	Telefone	Cod_postal
42	António Dias	43	74000	789654	1000
5	Célia Morais	36	170000	123456	1100
32	Florinda Simões	35	147000		4000
37	Isabel Espada	28	86000		2040
49	José António	17	210000		1000
14	Nascimento Augusto	35	220000	456123	2300

A ordem pela qual as colunas são apresentadas é a ordem pela qual são colocadas na cláusula SELECT, ou, no caso de se colocar asterisco, é a ordem pela qual foram criadas na tabela.

◈ Pretende-se seleccionar o nome, salário e idade das pessoas da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT  Nome, Salário, Idade FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Nome	Salário	Idade
António Dias	74000	43
Célia Morais	170000	26
Florinda Simões	147000	35
Isabel Espada	86000	28
José António	210000	17
Nascimento Augusto	220000	35

3.3 - Restrição (Cláusula WHERE)

A operação de restrição permite restringir o número de linhas a apresentar, que satisfaçam a condição pretendida.

No entanto, para efectuar as restrições, é necessária a utilização de operadores que serão apresentados de seguida:

4 - Operadores

4.1 - Operadores Relacionais

Operador	Descrição	Exemplo	Resultado
=	Igual a	7=5	Falso
>	Maior que	7>5	Verdadeiro
<	Menor que	7<5	Falso
>=	Maior ou Igual que	7>=5	Verdadeiro
<=	Menor ou Igual que	7<=5	Falso
<> ou !=	Diferente	7<>5	Verdadeiro

◈ Pretende-se seleccionar, da tabela Pessoa, todas as pessoas com idade igual a 35 anos.

Access	Sql Server	Oracle	SyBase

```
SELECT *
FROM Pessoa
WHERE idade = 35
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade	Salário	Telefone	Cod_postal
32	Florinda Simões	35	147000		4000
14	Nascimento Augusto	35	220000	456123	2300

◈ Pretende-se seleccionar, da tabela Pessoa, a identificação (Id), o Nome e Salário das pessoas com idade maior ou igual a 18 anos.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Salário
FROM Pessoa
WHERE idade >= 18
```

4.2 - Operadores Lógicos

Operador	Exemplo
AND	condição ou atributo AND condição ou atributo
OR	condição ou atributo OR condição ou atributo
NOT	NOT condição

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome, Idade e Salário de todas as pessoas com idade entre os 30 e os 40 anos.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade, Salário
FROM Pessoa
WHERE idade >= 30   AND   idade   <= 40
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade	Salário
5	Célia Morais	36	170000
32	Florinda Simões	35	147000
14	Nascimento Augusto	35	220000

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome e Idade das pessoas com idades maiores que os 40 anos e menores que os 30 anos.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade
FROM Pessoa
WHERE idade < 30   OR   idade   > 40
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade
42	António Dias	43
37	Isabel Espada	28
49	José António	17

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome e Idade das pessoas com idades maiores que os 40 anos e menores que os 30 anos.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade
FROM Pessoa
WHERE NOT (idade >= 30   AND   idade  <= 40)
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade
42	António Dias	43
37	Isabel Espada	28
49	José António	17

No caso apresentado, caso não sejam colocados os parêntesis, o resultado obtido é diferente e incorrecto.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade
FROM Pessoa
```

WHERE **NOT** idade >= 30 **AND** idade <= 40

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade
37	Isabel Espada	28
49	José António	17

4.3 - Operador BETWEEN

Este operador permite especificar intervalos de valores.

SELECT $campo_1$, $campo_2$, ... $campo_n$
FROM tabela $_1$, tabela $_2$, ... tabela $_n$
WHERE valor [NOT] **BETWEEN** valor1 **AND** valor2

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome, Idade e Salário de todas as pessoas com idade entre os 30 e os 40 anos.

Access	Sql Server	Oracle	SyBase

SELECT Id, Nome, Idade, Salário
FROM Pessoa
WHERE idade **BETWEEN** 30 **AND** 40

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade	Salário
5	Célia Morais	36	170000
32	Florinda Simões	35	147000
14	Nascimento Augusto	35	220000

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome, Idade e Salário de todas as pessoas cuja idade não está compreendida entre os 30 e os 40 anos.

Access	Sql Server	Oracle	SyBase

SELECT Id, Nome, Idade, Salário
FROM Pessoa
WHERE idade **NOT BETWEEN** 30 **AND** 40

Ou

SELECT Id, Nome, Idade, Salário
FROM Pessoa
WHERE **NOT** idade **BETWEEN** 30 **AND** 40

4.4 - Operador IN

Este operador permite especificar conjuntos de valores.

SELECT $campo_1$, $campo_2$, ... $campo_n$
FROM tabela $_1$, tabela $_2$, ... tabela $_n$
WHERE valor [NOT] **IN** ($valor_1$, $valor_2$, $valor_n$)

◈ Pretende-se seleccionar, da tabela Postal, os códigos postais existentes para os locais (LISBOA e RIO MAIOR).

Access	Sql Server	Oracle	SyBase

SELECT *
FROM Postal
WHERE Local **IN** (“LISBOA”, “RIO MAIOR”)

Depois de executado o comando, o resultado obtido será:

Código	Local
1000	LISBOA
1100	LISBOA
1500	LISBOA
2040	RIO MAIOR

◈ Pretende-se seleccionar, da tabela Postal, os códigos postais existentes cujos os locais não sejam (LISBOA e RIO MAIOR).

Access	Sql Server	Oracle	SyBase

```
SELECT *
FROM Postal
WHERE Local  NOT IN  ("LISBOA", "RIO MAIOR")
```

Depois de executado o comando, o resultado obtido será:

Código	Local
2000	SANTARÉM
2300	TOMAR
4000	PORTO

4.5 - Operador IS

Este operador permite efectuar o tratamento de nulos (NULL) – strings vazias. NULL não é zero, é string vazia. A comparação com NULL terá de ser efectuada sempre com o operador IS, caso contrário devolve sempre Falso.

SELECT $campo_1$, $campo_2$, ... $campo_n$

FROM tabela $_1$, tabela $_2$, ... tabela $_n$

WHERE valor **IS** [NOT] **NULL**

◈ Pretende-se seleccionar da tabela Pessoa, os Nomes das pessoas sem telefone.

Access	Sql Server	Oracle	SyBase

```
SELECT Nome
FROM Pessoa
WHERE Telefone IS NULL
```

Depois de executado o comando, o resultado obtido será:

Nome
Florinda Simões
Isabel Espada
José António

◈ Pretende-se seleccionar, da tabela Pessoa, os dados das pessoas com telefone.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade, Salário, Telefone, Cod_postal
FROM Pessoa
WHERE Telefone IS NOT NULL
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade	Salário	Telefone	Cod_postal
42	António Dias	43	74000	789654	1000
5	Célia Morais	36	170000	123456	1100
14	Nascimento Augusto	35	220000	456123	2300

4.6 - Operador LIKE

Este operador permite resolver alguns problemas naturais que existem quando se pretende comparar strings.

A utilização do operador LIKE permite fazer comparações de partes da string. Para tal, utilizam-se dois WILDCARDS.

Access	Sql Server	Oracle	SyBase

WILDCARD	SIGNIFICADO
*	Qualquer string de zero ou mais caracteres
?	Um caracter qualquer

Access	Sql Server	Oracle	SyBase

WILDCARD	SIGNIFICADO
%	Qualquer string de zero ou mais caracteres
_ (underscore)	Um caracter qualquer

◈ Consideremos uma tabela Mensagem, com os atributos Id_Msg e Mensagem, referente a mensagens de uma empresa comercial.
Pretende-se seleccionar os códigos de mensagem e descrição das mensagens que comecem pela letra T.

Access	Sql Server	Oracle	SyBase

SELECT *
FROM Mensagem
WHERE Mensagem **LIKE "**T*"

Depois de executado o comando, o resultado obtido será:

Id_Msg	Mensagem
80	**T**ransportes
90	**T**elefonemas
105	**T**ratamentos

◈ Pretende-se seleccionar os códigos de mensagem e descrição das mensagens que terminem em "as".

Access	Sql Server	Oracle	SyBase

SELECT *
FROM Mensagem
WHERE Mensagem **LIKE** "*as"

Depois de executado o comando, o resultado obtido será:

Id_Msg	Mensagem
10	Comissão de Vend**as**
90	Telefonem**as**

◈ Pretende-se seleccionar os códigos de mensagem e descrição das mensagens que possuam a palavra "Vendas".

Access	Sql Server	Oracle	SyBase

SELECT *
FROM Mensagem
WHERE Mensagem **LIKE** "*Vendas*"

Depois de executado o comando, o resultado obtido será:

Id_Msg	Mensagem
10	Comissão de **Vendas**
40	**Vendas** Extra

◈ Pretende-se seleccionar os códigos de mensagem e descrição das mensagens cuja segunda letra seja "e".

Access	Sql Server	Oracle	SyBase

SELECT *
FROM Mensagem
WHERE Mensagem **LIKE "**?e*"

Depois de executado o comando, o resultado obtido será:

Id_Msg	Mensagem
40	V**e**ndas Extra
90	T**e**lefonemas

4.7 - Precedência dos operadores

ORDEM	OPERADOR	SÍMBOLO
1ª	Parêntesis	( )
2ª	Multiplicação / Divisão	* /
3ª	Adição /Subtracção	+ -
4ª	NOT	
5ª	AND	
6ª	OR	

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome e Idade das pessoas com idades maiores que os 29 anos e sem telefone e as pessoas com idades inferiores a 28 anos.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade
FROM Pessoa
WHERE Idade <= 27  OR
        Idade >= 30  AND
        Telefone IS NULL
```

É equivalente a:

```
SELECT Id, Nome, Idade
FROM Pessoa
WHERE Idade <= 27  OR
        (Idade >= 30  AND Telefone IS NULL)
```

◈ Pretende-se seleccionar, da tabela Pessoa, o Id, Nome e Idade das pessoas com idades maiores que os 29 anos e as pessoas com idades inferiores a 28 anos, que não possuam telefone.

Access	Sql Server	Oracle	SyBase

```
SELECT Id, Nome, Idade
FROM Pessoa
WHERE (Idade <= 27  OR Idade >= 30)
        AND Telefone IS NULL
```

5 - Ordenação

A ordenação pode ser realizada através da cláusula ORDER BY no comando SELECT. Esta cláusula, se existir, aparece sempre posicionada no final do comando SELECT.

SELECT $campo_1$, $campo_2$, ... $campo_n$
FROM tabela $_1$, tabela $_2$, ... tabela $_n$
[WHERE Condição]
[GROUP BY .....]
[HAVING …….]
[**ORDER BY Campo [ASC | DESC], Campo [ASC | DESC]…**]

Onde **Campo** representa o nome de um Campo, uma Expressão ou a Posição pela qual se pretende ordenar o resultado do SELECT.
ASC indica que a ordenação é **ASC**endente e **DESC** indica que a ordenação é **DESC**endente.
Por defeito, ou seja, caso não seja indicado um tipo de ordenação, esta é efectuada por ordem ascendente.

5.1 - Ordenando por uma coluna

Considerando a tabela Pessoa, pretende-se ordenar os dados por idade.

Access	Sql Server	Oracle	SyBase

```
SELECT  *                          SELECT  *
FROM Pessoa              ou        FROM Pessoa
ORDER BY Idade                     ORDER BY Idade ASC
```

Depois de executado o comando, o resultado obtido será:

Id	Nome	Idade	Salário	Telefone	Cod_postal
49	José António	17	210000		1000
37	Isabel Espada	28	86000		2040
32	Florinda Simões	35	147000		4000
14	Nascimento Augusto	35	220000	456123	2300
5	Célia Morais	36	170000	123456	1100
42	António Dias	43	74000	789654	1000

5.2 - Ordenando por várias colunas

Consideremos a tabela Comissões, com os atributos Id_comissão, Id_mensagem e valor. Pretende-se ordenar os dados da tabela comissão por Id_comissão e por Id_ mensagem.

Access	Sql Server	Oracle	SyBase

```
SELECT  *
FROM Comissão
ORDER BY Id_comissão, Id_mensagem
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Id_mensagem	Valor
14	10	10500
14	70	400
14	100	3750
25	10	2500
25	30	3370
37	40	14230
47	110	120

◈ Pretende-se ordenar os dados da tabela comissão, cujo Id_comissão seja inferior a 30, por Id_comissão ascendentemente e por valor descendentemente.

Access	Sql Server	Oracle	SyBase

```
SELECT  *
FROM Comissão
WHERE Id_comissão < 30
ORDER BY Id_comissão ASC, Valor DESC
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Id_mensagem	Valor
14	10	10500
14	100	3750
14	70	400
25	30	3370
25	10	2500

5.3 - Selecção de expressões

Consideremos a tabela Pessoa. Pretende-se o nome e idade de todas as pessoas, seleccionando a idade que irão ter daqui a 3 anos, ordenando os dados de saída por nome.

Access	Sql Server	Oracle	SyBase

```
SELECT  Nome, Idade, Idade + 3
FROM Pessoa
ORDER BY Nome
```

Depois de executado o comando, o resultado obtido será:

Nome	Idade	Expr1002
António Dias	43	46
Célia Morais	36	39
Florinda Simões	35	38
Isabel Espada	28	31
José António	17	20

O resultado do cálculo é atribuído a uma Expr 1002, sendo que esta varia de sistema para sistema e é atribuída automaticamente.

◈ Pretende-se o nome e idade de todas as pessoas, seleccionando a idade que irão ter daqui a 3 anos, ordenando os dados de saída por nome, e atribuíndo a idade actual a (Idade_actual) e a idade daqui a 3 anos a (Idade_em_2004).

Access	Sql Server	Oracle	SyBase

```
SELECT  Nome, Idade AS Idade_actual, Idade+3 AS Idade_em_2004
FROM Pessoa
ORDER BY Nome
```

Depois de executado o comando, o resultado obtido será:

Nome	Idade_actual	Idade_em_2004
António Dias	43	46
Célia Morais	36	39
Florinda Simões	35	38
Isabel Espada	28	31
José António	17	20

5.4 - Ordenação e NULLs

A forma como o NULL é colocado no resultado ordenado de um SELECT depende de sistema para sistema.

Alguns sistemas consideram o valor NULL menor que qualquer outro valor. Outros colocam o valor NULL sempre no topo dos valores, seja a ordenação ascendente ou descendente.

5.5 - Eliminação de repetições (DISTINCT e ALL)

A cláusula DISTINCT colocada a seguir ao SELECT, permitir retirar os dados repetidos.

A cláusula ALL colocada a seguir ao SELECT, mostra todos os conjuntos de valores existentes, repetidos ou não.

◈ Considerando uma tabela Postal, com os atributos Código e Local, pretende-se seleccionar todos os Locais da tabela Postal.

Access	Sql Server	Oracle	SyBase

```
SELECT ALL Local
FROM Postal
```

```
SELECT DISTINCT Local
FROM Postal
```

Depois de executados os comandos, os resultados obtidos serão:

Local
LISBOA
LISBOA
LISBOA
SANTARÉM
RIO MAIOR
TOMAR
PORTO

Local
LISBOA
SANTARÉM
RIO MAIOR
TOMAR
PORTO

Notas: As cláusulas DISTINCT e ALL só podem ser colocadas imediatamente a seguir ao SELECT.

Sintaxes inválidas:

SELECT Id, **DISTINCT** Id_mensagem
SELECT **DISTINCT** Id, **ALL** Id_mensagem

6 - Juntando várias tabelas

O Modelo Relacional estabelece claramente as regras para a divisão da informação entre tabelas, de forma a evitar a duplicação da informação.
Esta dispersão da informação por diferentes tabelas é facilmente manipulável através da linguagem SQL, um vez que a ligação entre estas será realizada através das chaves estrangeiras.
A junção entre tabelas faz-se colocando na cláusula FROM o conjunto de tabelas que se pretende juntar.

SELECT $campo_1$, $campo_2$, ... $campo_n$ *
FROM $tabela_1$, $tabela_2$, ... $tabela_n$

Consideremos as seguintes relações (Pessoa e Postal):

Id	Nome	Idade	Telefone	Cod_postal
71	António Dias	43	789654	1000
54	Célia Morais	36	123456	1000
12	Isabel Silva	28		2040
49	José António	17	333555	2000

Código	Local
1000	Lisboa
2000	Santarém
2040	Rio Maior
4000	Porto

6.1 - Produto Cartesiano de Tabelas

Access	Sql Server	Oracle	SyBase

SELECT * FROM Pessoa, Postal

Obtém-se um resultado bastante inesperado e que consiste no produto cartesiano de dois conjuntos de elementos.
O produto cartesiano entre as tabelas Pessoa e Postal associa a cada linha da tabela Pessoa o conjunto das linhas da tabela Postal.
Dado que na tabela Pessoa existem 4 registos e na tabela Postal existem 4 registos, o resultado do produto cartesiano será 4*4 = 16 registos.
O resultado do produto cartesiano será:

Id	Nome	Idade	Telefone	Cod_postal	Código	Local
71	António Dias	43	789654	1000	1000	Lisboa
71	António Dias	43	789654	1000	2000	Santarém
71	António Dias	43	789654	1000	2040	Rio Maior
71	António Dias	43	789654	1000	4000	Porto
54	Célia Morais	36	123456	1000	1000	Lisboa
54	Célia Morais	36	123456	1000	2000	Santarém
54	Célia Morais	36	123456	1000	2040	Rio Maior
54	Célia Morais	36	123456	1000	4000	Porto
12	Isabel Silva	28		2040	1000	Lisboa
12	Isabel Silva	28		2040	2000	Santarém
12	Isabel Silva	28		2040	2040	Rio Maior
12	Isabel Silva	28		2040	4000	Porto
49	José António	17	333555	2000	1000	Lisboa
49	José António	17	333555	2000	2000	Santarém
49	José António	17	333555	2000	2040	Rio Maior
49	José António	17	333555	2000	4000	Porto

O pretendido seria a junção entre as tabelas Pessoa e Postal. Para tal é necessário, na cláusula WHERE, indicar as chaves estrangeiras de ligação.

Access	Sql Server	Oracle	SyBase

```
SELECT *
FROM Pessoa, Postal
WHERE Cod_postal = Código
```

Neste caso, o resultado da selecção seria:

Id	Nome	Idade	Telefone	Cod_postal	Código	Local
71	António Dias	43	789654	1000	1000	Lisboa
54	Célia Morais	36	123456	1000	1000	Lisboa
12	Isabel Silva	28		2040	2040	Rio Maior
49	José António	17	333555	2000	2000	Santarém

6.2 - Equi-Join

Acontece quando todas as colunas das tabelas são apresentadas e a ligação entre as tabelas é feita através de uma igualdade, dando origem assim a duas colunas de conteúdos exactamente iguais.

Access	Sql Server	Oracle	SyBase

```
SELECT *
FROM Pessoa, Postal
WHERE Cod_postal = Código
```

Neste caso, o resultado da selecção seria:

Id	Nome	Idade	Telefone	Cod_postal	Código	Local
71	António Dias	43	789654	1000	1000	Lisboa
54	Célia Morais	36	123456	1000	1000	Lisboa
12	Isabel Silva	28		2040	2040	Rio Maior
49	José António	17	333555	2000	2000	Santarém

6.3 - Natural Join

Acontece quando todas as colunas envolvidas na ligação entre tabelas são apresentadas sem repetição de colunas.

Access	Sql Server	Oracle	SyBase

```
SELECT Pessoa.*, Postal.Local
FROM Pessoa, Postal
WHERE Cod_postal = Código
```

Neste caso, o resultado da selecção seria:

Id	Nome	Idade	Telefone	Cod_postal	Local
71	António Dias	43	789654	1000	Lisboa
54	Célia Morais	36	123456	1000	Lisboa
12	Isabel Silva	28		2040	Rio Maior
49	José António	17	333555	2000	Santarém

Estes dois tipos de ligação entre tabelas fazem parte de um tipo de ligação mais geral denominada INNER JOIN.
Num INNER JOIN, apenas são apresentados os registos em que exista ligação entre as tabelas.

6.4 - INNER JOIN

Embora existam diversos tipos de ligação entre tabelas (JOIN), este é o tipo mais comum e utilizado.

◈ Pretende-se seleccionar o nome e morada completa (código e localidade) de todas as pessoas da tabela Pessoa.

Access	**Sql Server**	**Oracle**	**SyBase**

```
SELECT Nome, Cod_postal, Local
FROM Pessoa, Postal
Where Cod_postal = Código
```

Depois de executados os comandos, os resultados obtidos serão:

Nome	Cod_postal	Local
António Dias	1000	Lisboa
Célia Morais	1000	Lisboa
Isabel Silva	2040	Rio Maior
José António	2000	Santarém

Embora ainda de uso pouco comum, é possível, em alguns sistemas, escrever o mesmo SELECT num formato em que se especifique a natureza do Join (INNER), no mesmo exemplo mas ordenando o resultado por Código Postal.

Access	**Sql Server**	Oracle	**SyBase**

```
SELECT Nome, Cod_postal, Local
FROM Pessoa INNER JOIN Postal
        ON Pessoa.Cod_postal = Postal.Código
ORDER BY Pessoa.Cod_postal
```

Depois de executados os comandos, os resultados obtidos serão:

Nome	Cod_postal	Local
António Dias	1000	Lisboa
Célia Morais	1000	Lisboa
José António	2000	Santarém
Isabel Silva	2040	Rio Maior

6.5 - OUTER JOIN

O conceito do Outer Join é obter numa ligação a totalidade das linhas de uma tabela, ainda que não exista o correspondente valor na outra tabela a que esta está ligada pela junção.

◈ Pretende-se seleccionar o nome e morada completa (código e localidade) de todas as pessoas da tabela Pessoa, assim como todos os código postais existentes na tabela Postal.

Access	**Sql Server**	Oracle	SyBase

```
SELECT Nome, Cod_postal, Código, Local
FROM Postal LEFT JOIN Pessoa
        ON Postal.Código = Pessoa.Cod_postal
```

Depois de executados os comandos, os resultados obtidos serão:

Nome	Cod_postal	Código	Local
António Dias	1000	1000	Lisboa
Célia Morais	1000	1000	Lisboa
		1100	Lisboa
José António	2000	2000	Santarém
Isabel Silva	2040	2040	Rio Maior
		2300	Tomar
		4000	Porto

Todas as linhas da tabela Postal são apresentadas. Se existir correspondente na coluna Cod_postal da tabela Pessoa, são mostrados os dados, senão as entradas da tabela Pessoa são preenchidas a NULL.

Access	Sql Server	**Oracle**	SyBase

```
SELECT Nome, Cod_postal, Código, Local
FROM Postal, Pessoa
WHERE Código  = Cod_postal (+)
```

Access	Sql Server	Oracle	**SyBase**

```
SELECT Nome, Cod_postal, Código, Local
FROM Postal, Pessoa
WHERE Código * = Cod_postal
```

6.6 - OUTER JOIN

Quando o Outer Join é efectuado à direita, são considerados todos os registos da tabela da direita e apenas os registos correspondentes na tabela da esquerda.
Consideremos as seguintes relações (Pessoa e Comissão):

Id	Nome	Idade
71	António Dias	43
12	Isabel Silva	28
49	José António	17
85	João Silva	49

Id	Id_msg	Valor
49	10	1250
49	70	750
71	12	100
71	15	200

◈ Pretende-se seleccionar todas as pessoas da tabela pessoa, assim como os correspondentes valores de comissão.

Access	**Sql Server**	Oracle	SyBase

```
SELECT Nome, Valor
FROM Comissão RIGHT JOIN Pessoa
        ON Comissão.Id = Pessoa.Id
```

Depois de executados os comandos, os resultados obtidos serão:

Nome	Valor
António Dias	100
António Dias	200
Isabel Silva	
José António	1250
José António	750
João Silva	

Access	Sql Server	**Oracle**	SyBase

```
SELECT Nome, Valor
FROM Comissão, Pessoa   WHERE  Comissão.Id (+) = Pessoa.Id
```

Access	Sql Server	Oracle	**SyBase**

```
SELECT Nome, Valor
FROM Comissão, Pessoa   WHERE  Comissão.Id = * Pessoa.Id
```

6.7 - SELF JOIN

O Self Join é uma variante do Inner Join, em que se comparam duas colunas da mesma tabela.
◈ Pretende-se saber quais os tipos de códigos postais existentes diferentes do local LISBOA.

Access	**Sql Server**	**Oracle**	**SyBase**

```
SELECT p1.Código, p2.Local
FROM Postal.p1, Postal.p2
WHERE p1.Código = p2.Código
        AND
         p2.Local <> “Lisboa”
```

6.8 - UNION

Uma união não é propriamente uma ligação entre tabelas. A **UNION** permite juntar o conteúdo de múltiplos comandos SELECT.
◈ Pretende-se juntar o código e a descrição da tabela mensagens aos códigos postais.

Access	**Sql Server**	**Oracle**	**SyBase**

SELECT Id_msg, Mensagem FROM Mensagem
UNION
SELECT Código, Local FROM Postal

Depois de executados os comandos, os resultados obtidos serão:

Id_msg	Mensagem
10	Comissão de vendas
30	Vendas extra
40	Refeições
1000	Lisboa
2040	Rio Maior
4000	Porto

Na UNION, o número de campos a seleccionar em cada um dos comandos SELECT tem de ser igual. O nome dos campos não é relevante, mas o tipo de dados que pode ser agrupado depende de sistema para sistema.

◈ Pretende-se juntar as tabelas Postal (cujos locais contenham a string “OR”) e Mensagem (cujas Id sejam inferiores a 30), ordenando o resultado por mensagem.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_msg, Mensagem FROM Mensagem
WHERE Id_msg < 30
UNION
SELECT Código, Local FROM Postal
WHERE Local LIKE “*OR*”
ORDER BY mensagem
```

6.9 - INTERSECT

O operador **INTERSECT** permite juntar o resultado de dois comandos SELECT, apresentando apenas as linhas que resultem de ambos os comandos.

◈ Pretende-se seleccionar as linhas da tabela Postal cujo Código é menor ou igual a 5000 e é maior ou igual a 3000.

Access	**Sql Server**	**Oracle**	**SyBase**

```
SELECT * FROM Postal
WHERE Código >= 3000 AND Código <= 5000
```

Access	Sql Server	**Oracle**	SyBase

```
SELECT * FROM Postal WHERE Código >= 3000
INTERSECT
SELECT * FROM Postal WHERE Código <= 5000
```

6.10 - MINUS

O operador **MINUS** devolve os registos que resultem do primeiro SELECT e que não aparecem no segundo.

◈ Pretende-se seleccionar as linhas da tabela Postal cujo Código é menor ou igual a 5000, ignorando todos os elementos cujo código está entre 2000 e 3000.

Access	**Sql Server**	**Oracle**	**SyBase**

```
SELECT * FROM Postal
WHERE Código <= 5000 AND
      Código NOT BETWEEN 2000 AND 3000
```

Access	Sql Server	**Oracle**	SyBase

```
SELECT * FROM Postal WHERE Código <= 5000
MINUS
SELECT * FROM Postal WHERE Código BETWEEN 2000 AND 3000
```

6.11 - RESUMO DAS JUNÇÕES

JOIN	Descrição
Produto Cartesiano	Juntar cada linha da tabela T1 com todas as linhas de T2.
INNER JOIN	Junção tradicional, em que apenas são apresentadas as linhas comuns às duas tabelas.
OUTER JOIN	Extensão do Inner Join ao proporcionar todos os registos de uma das tabelas, mesmo que sobre estes não exista qualquer ligação.
UNION	Todos os registos de qualquer das pesquisas (sem duplicados).
UNION ALL	Todos os registos de qualquer das pesquisas (com duplicados).
INTERSECT	Todos os registos comuns a ambas as pesquisas.
MINUS	Todos os registos da primeira pesquisa que não aparecem na segunda.

7 - Funções de agregação

Estas funções, também designadas por **Funções Estatísticas**, têm por objectivo obter informação sobre conjuntos de linhas especificadas na cláusula WHERE ou sobre grupos de linhas indicadas na cláusula GROUP BY.

Função	Descrição
COUNT	Devolve o número de linhas.
MAX	Devolve o maior valor da coluna.
MIN	Devolve o menor valor da coluna.
SUM	Devolve a soma de todos os valores da coluna.
AVG	Devolve a média (**AV**era**G**e) de todos os valores da coluna.

7.1 - FUNÇÃO COUNT

A função de agregação COUNT devolve o número de linhas que resultam num SELECT. Pode ser utilizada de três formas distintas:

Forma	Descrição
COUNT (*)	Devolve o número de linhas que resulta de um SELECT.
COUNT (coluna)	Devolve o número de ocorrências na coluna diferentes de NULL.
COUNT (DISTINCT coluna)	Devolve o número de ocorrências (sem repetições) na coluna.

Consideremos a seguinte relação Pessoa:

Id	Nome	Idade	Salário	Telefone	Cod_postal
42	**António Dias**	**43**	94000	7896544	1000
5	Célia Morais	36	170000	1234565	1000
32	Florinda Simões	35	147000		4000
37	Isabel Espada	28	**86000**		2040
49	José António	**17**	210000		1000
14	Nascimento Augusto	35	220000	4561233	2300
75	**Pedro Santos**	42	**235000**		2040

◈ Pretende-se saber quantas pessoas existem na tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT COUNT (*) AS Total
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Total
7

◈ Pretende-se saber quantas pessoas existem na tabela Pessoa e quantas pessoas possuem telefone.

Access	Sql Server	Oracle	SyBase

```
SELECT COUNT (*) AS [Total de pessoas],
       COUNT (Telefone) AS [Total de telefones]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Total de pessoas	Total de telefones
7	3

◈ Pretende-se saber quantos Códigos postais diferentes existem na tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT COUNT (DISTINCT Cod_postal) AS [Cód. postais diferentes]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Cód.postais diferentes
4

7.2 - MIN e MAX

As funções de agregação MIN e MAX permitem obter o menor e o maior valor de uma determinada coluna.

◈ Pretende-se saber qual o salário mais elevado e o salário mais baixo da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT MAX (salário) AS [Salário mais elevado],
       MIN (salário) AS [Salário mais baixo]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Salário mais elevado	Salário mais baixo
235000	86000

◈ Pretende-se saber a idade da pessoa mais nova e a idade da pessoa mais velha da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT MAX (idade) AS [Maior Idade], MIN (idade) AS [Menor Idade]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Maior Idade	Menor Idade
43	17

◈ Pretende-se saber o nome da primeira e da última pessoa da tabela Pessoa, se a lista fosse ordenada alfabeticamente.

Access	Sql Server	Oracle	SyBase

```
SELECT MAX (nome) AS [Primeiro], MIN (nome) AS [Último]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Primeiro	Último
António Dias	Pedro Santos

7.3 - SUM

A função SUM devolve a soma de uma determinada coluna.

◈ Pretende-se saber o total de salários da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT SUM (salário) AS [Total de Salários]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Total de Salários
1162000

◈ Pretende-se saber o total das idades da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT SUM (idade) AS [Total das idades]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Total das idades
236

◈ Pretende-se saber o total de salários da tabela Pessoa, assim como o total dos salários do próximo mês, sabendo que os salários serão aumentados em 3%.

Access	Sql Server	Oracle	SyBase

```
SELECT SUM (salário) AS [Total de Salários],
       SUM (salário) * 1.03 AS [Total de Salários com aumento]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Total de Salários	Total de Salários com aumento
1162000	1196860

7.4 - AVG

A função AVG devolve a média dos valores de uma determinada coluna.

◈ Pretende-se saber a média das idades da tabela Pessoa.

Access	Sql Server	Oracle	SyBase

```
SELECT AVG (idade) AS [Média de Idades]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Média de idades
33.714285714285714

Existem funções que permitem formatar o resultado do SELECT. Embora essas funções dependam de cada um dos sistemas, no Access a função FORMAT permite indicar sobre quantas casas decimais se pretende visualizar o resultado.

◈ Pretende-se saber a média das idades da tabela Pessoa, formatada a duas casas decimais.

Access	Sql Server	Oracle	SyBase

```
SELECT FORMAT (AVG (idade), '0.00') AS [Média de Idades]
FROM Pessoa
```

Depois de executado o comando, o resultado obtido será:

Média de idades
33.71

NOTA FINAL:

As funções MIN, MAX e COUNT podem ser utilizadas com qualquer tipo de dados. As funções SUM e AVG só podem ser aplicadas a campos numéricos. Se existirem valores NULL estes são ignorados por qualquer uma das funções.

8 - Agrupando a informação

As funções de agregação são uma ferramenta útil quando usada para obter informação resumida sobre o resultado de um comando SELECT.

No entanto, estas funções podem ser particularmente úteis no tratamento de informação de forma agrupada, não como um todo, mas em grupos mais pequenos.

◈ Pretende-se mostrar as comissões e respectivos valores, ordenando o resultado por Id da tabela comissão.

Access	Sql Server	Oracle	SyBase

```
SELECT  Id_comissão, Valor
FROM Comissão
ORDER BY Id_comissão
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Valor		Id_comissão	Valor
14	2600		37	120
14	400		37	5500
14	3750		37	14230
14	10500		37	20
25	370		40	20
25	2400		42	150
25	100		42	20

◈ Pretende-se saber o total de valores de comissões.

Access	Sql Server	Oracle	SyBase

```
SELECT  SUM(Valor) AS Total
FROM Comissão
```

Depois de executado o comando, o resultado obtido será:

Total
40180

No entanto, se o objectivo fosse obter a soma das comissões por cada Id_comissão, o resultado apresentado não seria o pretendido. Para resolver essas questões é necessário, antes de aplicar as funções de agregação, possuir a informação agrupada.

```
SELECT campo1, campo2, ... campon
FROM tabela 1 , tabela 2, .... tabela n
[WHERE Condição ]
[GROUP BY .....]
[HAVING ……]
[ORDER BY  .....]
```

8.1 - Cláusula GROUP BY

Esta cláusula divide o resultado de um SELECT em grupos de resultados que irão ser tratados com as funções de agregação.

- A cláusula GROUP BY é utilizada para agrupar informação.
- Os registos são processados em grupos de características semelhantes.
- As funções de agregação podem ser utilizadas para obter informação sobre cada um dos grupos.

◈ Pretende-se saber, para cada Id, o total de valores de comissões.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_comissão, SUM(Valor) AS Total
FROM Comissão
GROUP BY Id_comissão
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Total
14	17250
25	2870
37	19870
40	20
42	170

◈ Pretende-se saber, para cada Id, o maior valor de comissão.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_comissão, MAX(Valor) AS Maior
FROM Comissão
GROUP BY Id_comissão
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Maior
14	10500
25	2400
37	14230
40	20
42	150

Poderão ser utilizadas todas as outras funções de agregação, sobre os dados agrupados, sendo que também será possível efectuar ordenação sobre os dados agrupados e calculados.

◈ Pretende-se saber para cada Id, o maior valor de comissão, efectuando a ordenação descendente por maior valor de comissão.

Access	Sql Server	Oracle	SyBase

```
SELECT MAX(Valor) AS Maior, Id_comissão
FROM Comissão
GROUP BY Id_comissão
ORDER BY MAX(Valor) DESC
```

Depois de executado o comando, o resultado obtido será:

Maior	Id_comissão
14230	37
10500	14
2400	25
150	42
20	40

8.2 - Cláusula HAVING

Esta cláusula serve para fazer restrições ao nível dos grupos que são processados.

Esta cláusula actua sobre o resultado dos grupos, que resultam da função de agrupamento ORDER BY.

◈ Pretende-se saber, para cada Id, o total de valores de comissões. No entanto, só são relevantes os totais superiores a 1000.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_comissão, SUM(Valor) AS Total
FROM Comissão
GROUP BY Id_comissão
HAVING SUM(Valor) > 1000
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Total
14	17250
25	2870
37	19870

◈ Pretende-se saber, para cada Id, o maior valor de comissão. No entanto, só são relevantes os valores inferiores a 3000.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_comissão, MAX(Valor) AS Maior
FROM Comissão
GROUP BY Id_comissão
HAVING MAX(Valor) < 3000
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Maior
25	2400
40	20
42	150

8.3 - WHERE vs HAVING

Utiliza-se a cláusula WHERE sempre que se pretende restringir os registos a considerar na selecção. A cláusula HAVING serve para restringir os grupos que foram formados depois de aplicada a restrição da cláusula WHERE.

◈ Pretende-se saber qual o total de comissões para cada Id, considerando apenas aquelas cujo valor seja superior a 1000.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_comissão, SUM(Valor) AS Total
FROM Comissão
WHERE valor > 1000
GROUP BY Id_comissão
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Total	
14	16850	(17250-400)
25	2400	(2870-370-100)
37	19730	(19870-120-20)

◈ Pretende-se saber o total das comissões para cada Id, considerando apenas aquelas cujo valor total seja superior a 1000.

Access	Sql Server	Oracle	SyBase

```
SELECT Id_comissão, SUM(Valor) AS Total
FROM Comissão
GROUP BY Id_comissão
HAVING SUM(Valor) > 1000
```

Depois de executado o comando, o resultado obtido será:

Id_comissão	Total
14	17250
25	2870
37	19870

Nota Final:

Se um comando SELECT possuir a cláusula GROUP BY, **todas as colunas seleccionadas** (no SELECT) têm que estar presentes na cláusula GROUP BY.