

Bases de dados

- › **ISUTC INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES**



**DEPARTAMENTO DE TECN. DA
INFORMAÇÃO E COMUNICAÇÃO
Ano Lectivo 2024**

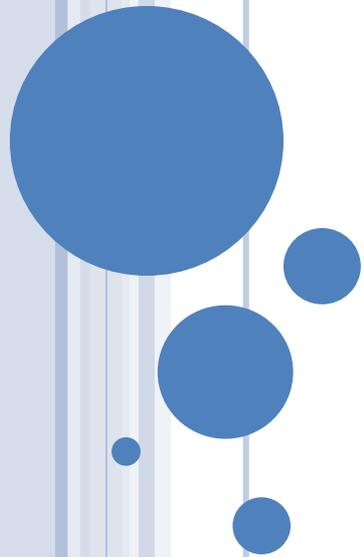
Bases de Dados

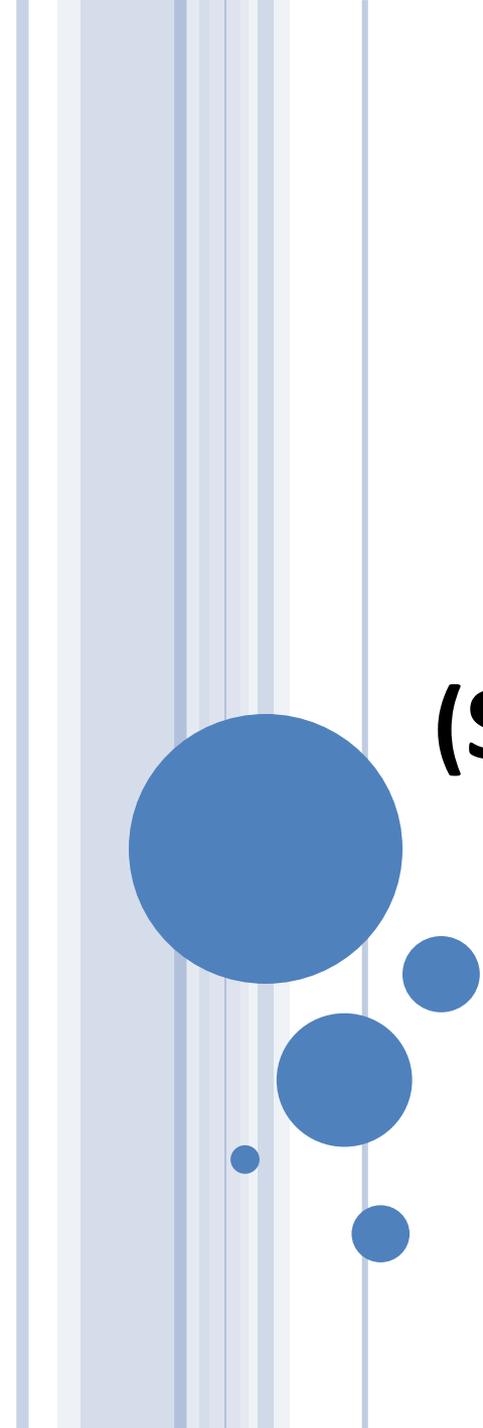
1º semestre

CONTEÚDO

- **Consultas Com Uma Tabela**
 - Cláusula WHERE
 - Condições Compostas
 - Ordenação dos Resultados
 - Funções de Agregação
 - Consultas Agrupadas
- **Actividades práticas 15 e 16 (TPC)**

MODELO FISICO





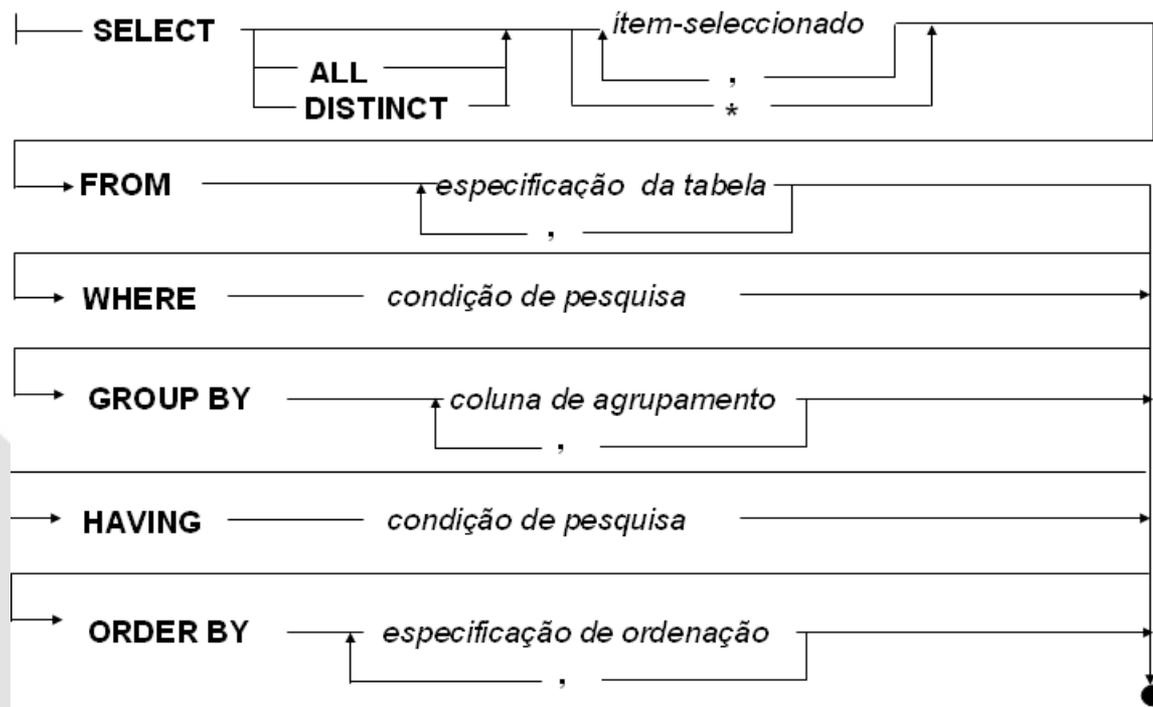
SQL

(Structured Query Language)

1. CONSULTAS NUMA ÚNICA TABELA

1. Consultas Com Uma Tabela

- Sintax:



1. Consultas Com Uma Tabela

- **SELECT** - especifica os atributos cujos valores interessa conhecer.
- **FROM** - especifica as tabelas envolvidas no processamento em questão
- **WHERE** - especifica a condição de restrição que permite restringir o número de linhas a apresentar.
- **GROUP BY** - permite agrupar informação.
- **HAVING** - especifica as restrições ao nível dos grupos que são processados.
- **ORDER BY** - especifica a ordenação dos resultados.

Comandos SQL

1. Consultas Com Uma Tabela

- **Exemplos:**

ESTUDANTE (est_num, est_apelido, est_nome, estsexo, est_bi, est_nacionalidade, est_morada, est_telef, est_email)

INSCRICAO(num_est, cod_disc, insc_data, nota)

DISCIPLINA (disc_cod, disc_nome)

- **(a) Imprimir o número e o nome de todos os estudantes.**

- SELECT est_num, est_nome
- FROM estudante;

- **(b) Imprimir todos os dados de estudantes.**

- SELECT *
- FROM estudante;

- **(c) Imprimir os números dos inscritos.**

- SELECT DISTINCT num_est
- FROM inscricao;

1.1 Cláusula WHERE

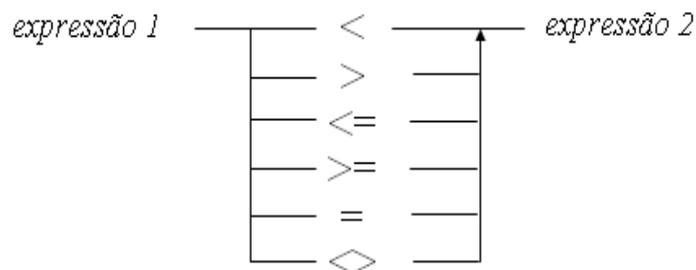
- Utilizamos para fazer filtragem de resultado.

Testes feitos:

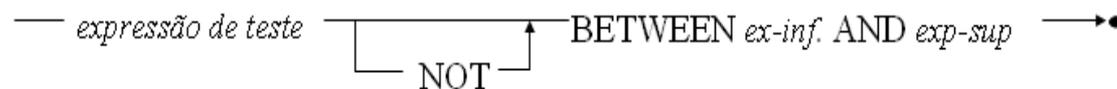
- Comparação
- Intervalo
- Conjunto
- Valores nulos
- Correspondência com padrões

- 1.1 Cláusula WHERE

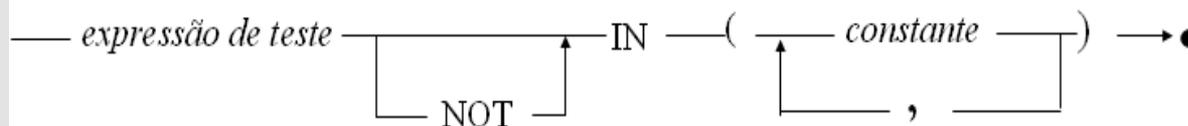
Teste de comparação



Teste de intervalo (permite especificar intervalos de valores)

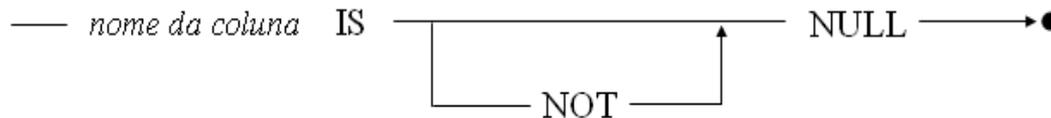


Teste de conjunto (permite especificar conjunto de valores)

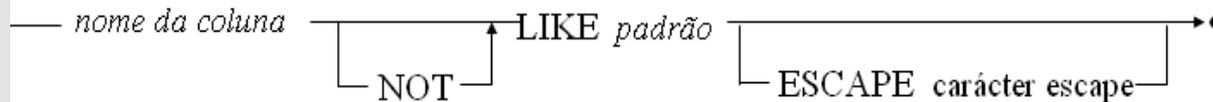


- 1.1 Cláusula WHERE

Teste de valores nulos



Teste de correspondência com padrões



% → Qualquer string de zero ou mais caracteres

_ (underscore) → UM caracter qualquer

Comandos SQL

- **1.1 Cláusula WHERE**

- **Exemplos:**
- **Dadas as seguintes tabelas**

Empregado

emp_num	emp_bi	emp_nome	emp_data_nasc	emp_cel	cod_dept	emp_sexo	emp_telef
0001	77777Y	Herminia	17/12/1982	829093221	03	F	21577700
0002	11111K	Tuaha	20/10/1976	826843220	02	M	null
0003	33333Z	Ossufo	01/09/1973	827342348	02	M	21000426
0004	22222X	Aderito	18/03/1982	null	02	M	21323290
0005	99999U	Aquiliama	07/10/1985	null	03	F	null
0006	55555Y	Daniela	11/10/1985	847623637	01	F	null

Departamento

Dep._cod	dept_nome	dept_localizacao
01	Recursos Humanos	Maputo Cidade
02	Contabilidade	Xai-Xai
03	Recursos Minerais	Pemba

Comandos SQL

• 1.1 Cláusula WHERE

- **Exemplos:**
- **(a) Imprimir todos os dados de empregados que estão no departamento 01.**
- **SELECT ***
- **FROM empregado**
- **WHERE cod_dept='01';**

- **(b) Imprimir o número, bi, nome e data de nascimento de todos os empregados cuja data de nascimento esta entre 01/01/1980 a 31/12/1985.**
- **SELECT emp_num, emp_bi, emp_nome, emp_data_nasc**
- **FROM empregado**
- **WHERE emp_data_nasc BETWEEN '01/01/1980'AND '31/12/1985';**

- **1.1 Cláusula WHERE**

Exemplos:

- **(c) Imprimir todos os empregados que estão nos departamentos 01, 02 e 04.**
- SELECT *
- FROM empregado
- WHERE cod_dept IN ("01","02","04");

- **(d) Imprimir empregados sem número de telefone**
- SELECT *
- FROM empregado
- WHERE emp_telef IS NULL;

- **(e) Imprimir bi e nome de empregados com número de telefone.**
- SELECT emp_bi, emp_nome
- FROM empregado
- WHERE emp_telef IS NOT NULL;

Comandos SQL

• 1.1 Cláusula WHERE

○ **Exemplos:**

- **(f) Imprimir os empregados cujo nome contém a palavra “ca”.**

- SELECT *

- FROM empregado

- WHERE emp_nome LIKE “%ca%”;

- **(g) Imprimir os empregados cujo nome tem como segunda letra “o”**

- SELECT *

- FROM empregado

- WHERE emp_nome LIKE “_o%”;

- **(h) Imprimir os empregados cujo nome começa pela letra “b”**

- SELECT *

- FROM empregado

- WHERE emp_nome LIKE “b %”;

ACTIVIDADE 15

ESTUDANTE								
est_nu m	est_apeli do	est_no me	est_se xo	est_bi	est_nacionali dade	est_morad a	est_tel ef	est_email
0001	Bobo	Dércio	M	100111 M	Moçambicana	B. Leberdade	8220128 4	derciobobe@transco m.co.mz
0002	Sulude	Joyce	F	101200Y	Malawiana	B. Nyerere	0287391 2	Joyce.sulude@gmail. com
0005	Chaúma	Ivan	M	104000 N	Portuguesa	C. Madeira	3200000 1	Ivan.chauma@trans com.co.mz
0003	Mussá	Jéssica	F	100050I	Moçambicana	B. Alto Maé	8200010 0	Jessica.mussa@trans com.co.mz
0006	Laimoni	Yara	F	100200 K	Moçambicana	B. Xipamanine	8400100 0	Yara.laimoni@transc om.co.mz
0007	Razak	Imran	M	100510L	Moçambicana	B. Alto Maé	8200101 0	Imran.razak@transc om.co.mz
0009	Parruque	Paula	F	110020 B	Moçambicana	B. Alto Maé	8601000 2	Paula.parruque@tra nscm.co.mz

DISCIPLINA	
disc_cod	disc_nome
SGBD	Sistemas de Gestão de Base de Dados
ASI	Aplicações e Serviços de Internet
ANT	Antenas
RIP	Redes IP
PDS	Processamento Digital de Sinais
SCII	Sistemas de Comunicação II

INSCRICAO

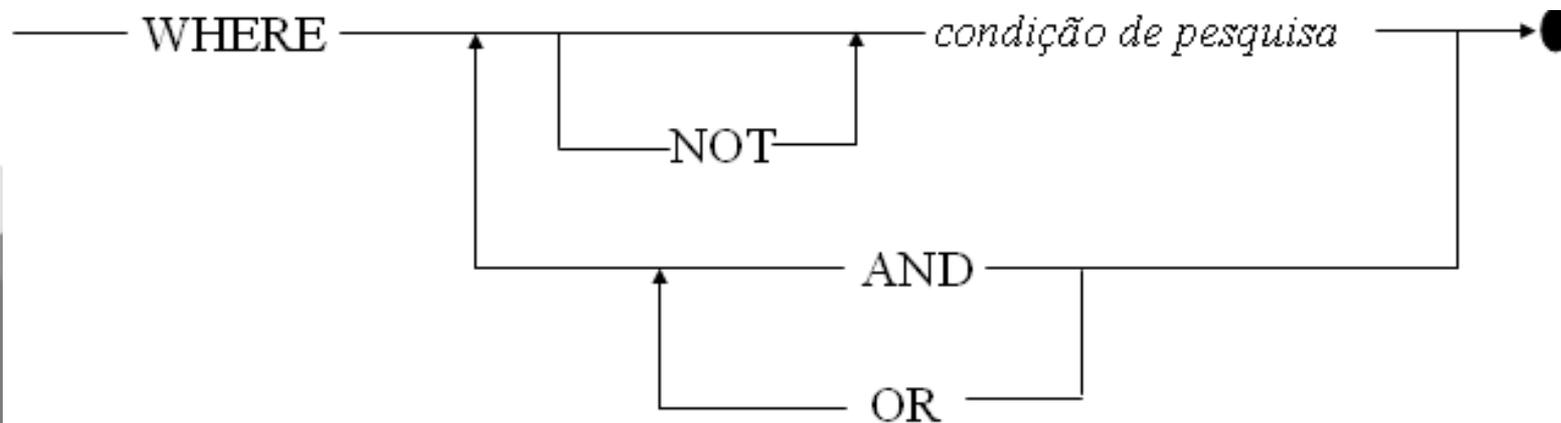
Num_est	Cod_disc	Insc_data	nota
0003	SGBD	20/06/2014	17
0005	SGBD	10/06/2015	7
0003	SGBD	02/06/2013	5
0001	ANT	17/07/2015	18
0006	RIP	12/06/2014	11
0007	SGBD	29/06/2015	15
0006	PDS	30/06/2013	4
0005	RIP	12/06/2012	8
0009	SGBD	13/06/2013	13
0009	ASI	19/06/2015	2
0001	ANT	25/02/2011	2

Com recurso a sql apresente consultas para impressão de:

1. Todos os dados de estudantes.
2. Numero, apelido e nome de estudantes.
3. Numero, apelido e nome de estudantes de nacionalidade moçambicana.
4. Numero, apelido e nome de estudantes residentes no Alto Maé.
5. Numero, apelido e nome de estudantes com número de telefone da Tmcel.
6. Numero, apelido e nome de estudantes com email da transcom
7. Numero, apelido e nome de estudantes cujo nome tem como segunda letra "a".
8. Numero, apelido e nome de estudantes cujo nome termina pela letra "a".
9. Numero, apelido e nome de estudantes estrangeiros.
10. Actualizar a nacionalidade da estudante Joyce para "Brasileira".

- **1.2 Condições Compostas**

- **Syntax**

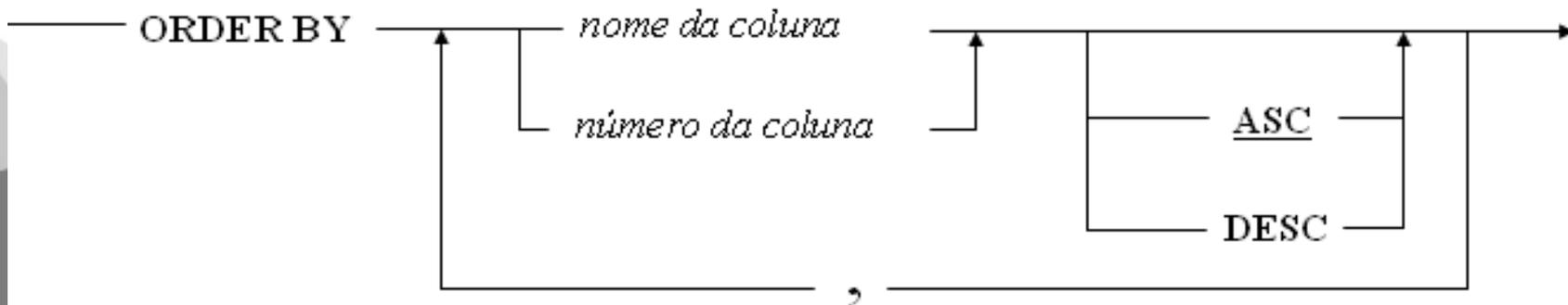


• 1.2 Condições Compostas

- Exemplo:
- Imprimir os nomes dos empregados que estão nos departamentos de 1 a 3 e sem número de telefone?
- `SELECT emp_nome`
- `FROM empregado`
- `WHERE (cod_dept BETWEEN 1 AND 3) AND (emp_telef IS NULL);`

• 1.3 Ordenação dos Resultados

○ Sintax



○ ASCendente, DESCendente

• 1.3 Ordenação dos Resultados

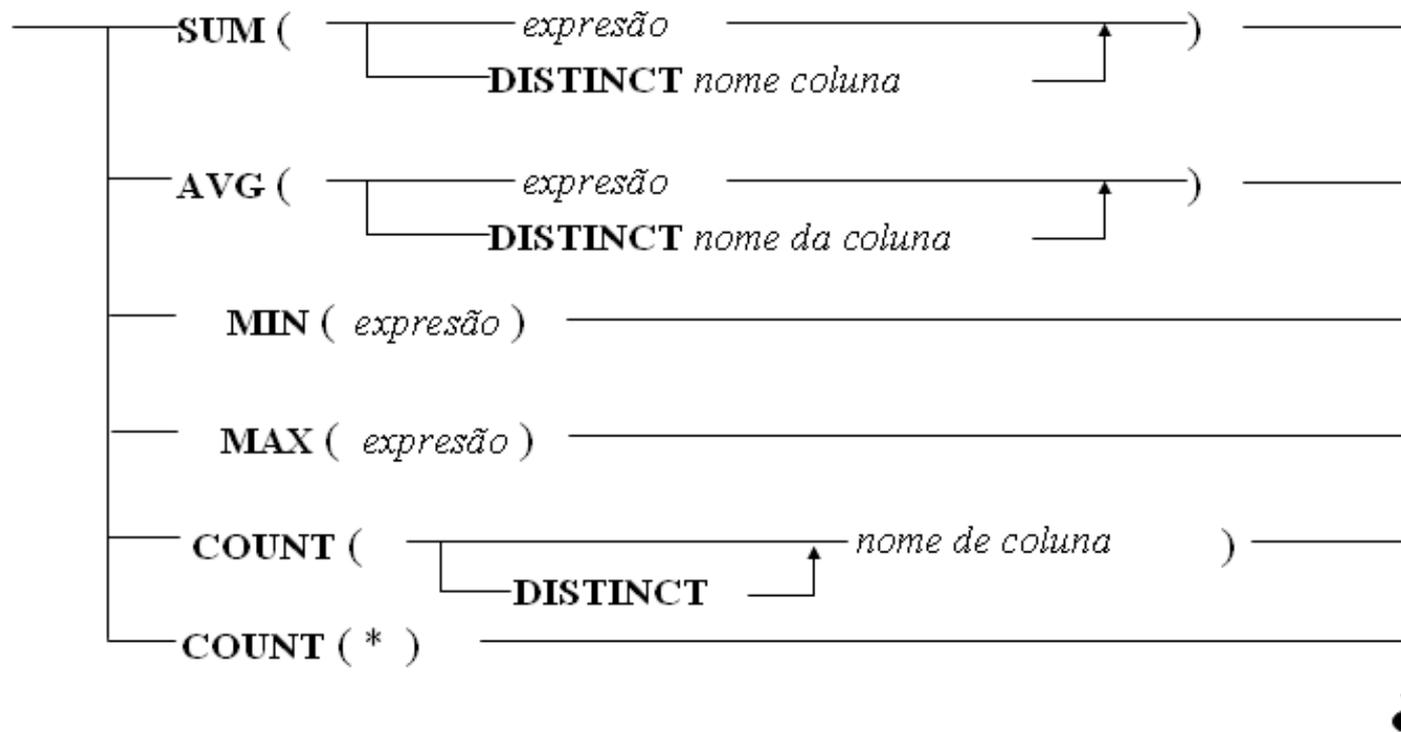
- Exemplos:
- (a) Imprimir todos empregados com número de telefone na ordem crescente de data de nascimento.
- SELECT *
- FROM empregado
- WHERE emp_telef IS NOT NULL
- ORDER BY emp_data_nasc ASC;
- (b) Imprimir todos os empregados ordenado por idade e departamento.
- SELECT *
- FROM empregado
- ORDER BY emp_data_nasc, cod_dept;

• 1.4 Funções de Agregação

- Têm por objectivo executar cálculos sobre o resultado de um comando **SELECT** (resumir) e são elas:
- **SUM ()** - Devolve a soma de todos os valores da coluna
- **AVG ()** - Devolve a média de todos os valores da coluna
- **MAX ()** - Devolve o maior valor da coluna
- **MIN ()** - Devolve o menor valor da coluna
- **COUNT ()** - Devolve o número de valores de uma coluna
- **COUNT (*)** - Devolve o número de linhas

• 1.4 Funções de Agregação

○ Sintax



- **1.4 Funções de Agregação**

- **Exemplos:**
- **(a) Quantos departamentos possuem empregados?**
- `SELECT COUNT(DISTINCT cod_dept)`
- `FROM empregado;`

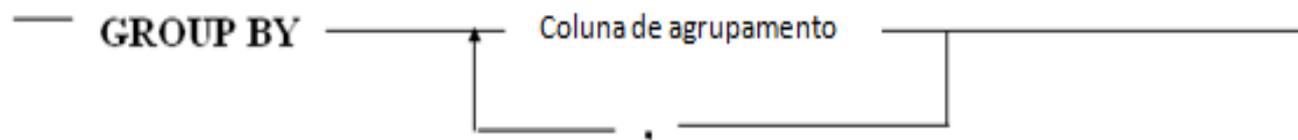
- **(b) Quantos empregados foram registados na base de dados?**
- `SELECT COUNT(*)`
- `FROM empregado;`

- **(c) Qual é o numero total de empregados e quantos tem telefone?**
- `SELECT COUNT(*), COUNT(emp_telef)`
- `FROM empregado;`

- **NB: Nas funções de agregação SUM(), AVG(), MIN(), MAX(), COUNT() os valores NULL são ignorados.**

• 1.5 Consultas Agrupadas

- Em algumas ocasiões é necessário agrupar registos.
- **Exemplo:** Calcular a media de salários para cada departamento.
- **Sintax:**



- **Exemplo:**
- **Imprimir o total de empregados por cada departamento.**
- SELECT cod_dept, COUNT(*)
- FROM empregado
- GROUP BY cod_dept;

NB: Qualquer coluna que não seja uma função de agregação só pode estar na cláusula SELECT se estiver na cláusula GROUP BY.
(rigidamente obrigatório no Oracle- vede <http://raelcunha.com/oracle.php>)

• 1.5 Consultas Agrupadas

Restrições sobre os Grupos: Cláusula HAVING

- **HAVING** actua unicamente sobre o resultado dos grupos e esta é utilizada sempre que se pretende restringir o conjunto dos **GRUPOS** de registos a considerar.

- **Syntax:**

— **HAVING** ————— *condição de pesquisa* —————→

- **Exemplo:**

- **Imprimir o total de empregados por idade, mas somentes para idades que ocuparam no mínimo três departamentos.**

- SELECT emp_data_nasc, COUNT (*)
- FROM empregado
- GROUP BY emp_data_nasc
- HAVING COUNT (DISTINCT cod_dept)>=3;

ACTIVIDADE 16 (TPC)

Com recurso a sql apresente consultas para impressão de:

1. Numero, apelido e nome de estudantes ordenados por apelido.
2. Total de estudantes por sexo.
3. Numero, apelido e nome de estudantes moçambicanos com numero de telefone de Tmcel.
4. Média das notas por disciplina.
5. Total de notas por disciplina, somente para as disciplinas com mais de 2 notas.
6. Total de estudantes por disciplina.
7. Média de notas positivas por disciplina.
8. Máxima nota na disciplina de SGBD.
9. Disciplina onde registou-se maior nota no ano de 2013.

Extra:

- i) Numeros de estudantes inscritos nas disciplinas de SGBD e RIP.
- ii) Total de inscrições por estudante
- iii) Total de disciplinas por estudante
- iv) Total de notas positivas por disciplina
- v) Media global de cada estudante (use somente notas positivas)
- vi) Numero de vezes que cada estudante reprovou.

GARANTE O TEU FUTURO
COM UMA FORMAÇÃO SÓLIDA



Prolong. da Av. Kim Il Sung (IFT/TDM) Edifício
D1
Maputo, Moçambique

www.facebook.com/isutc

www.transcom.co.mz/isutc