

# ACH2025

## Laboratório de Bases de Dados

### Aula 6

**SQL Oracle – Visão geral – SELECT**

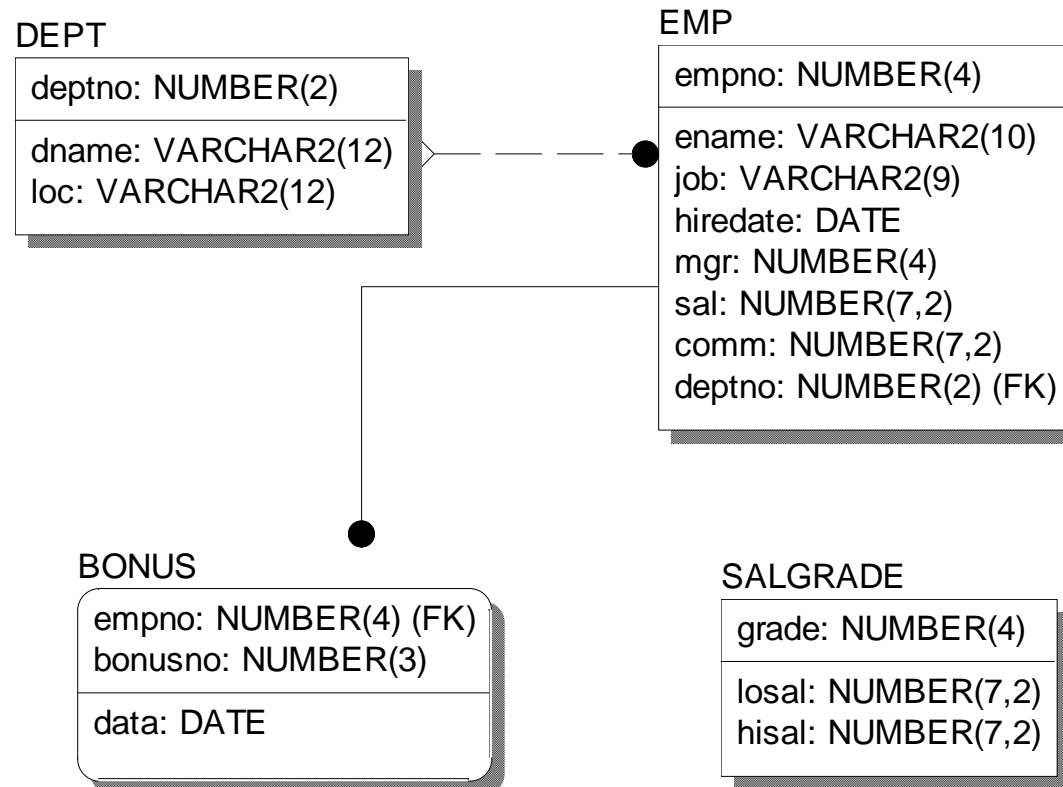
**Professora:**

➤ **Fátima L. S. Nunes**



# SQL – Comandos para seleção de dados

✓ **Modelo utilizado nos exemplos a seguir:**



# SQL – Comandos para seleção de dados

- ✓ **SELECT** - recupera informação do Banco de Dados.

- ✓ **Formato mais simples:**

SELECT colunas

FROM tabelas;

- ✓ Vamos considerar a tabela EMP para os exemplos a seguir



# SQL – Comandos para seleção de dados

```
SQL> set linesize 100;  
SQL> select * from emp;
```

EMPNO	JOB	ENAME	HIREDATE	SAL	COMM	DEPTNO	MGR
7839	PRESIDENT	KING	17/11/81	5000		10	
7566	MANAGER	JONES	02/04/81	2975		20	7839
7788	ANALYST	SCOTT	19/04/87	3000		20	7566
7876	CLERK	ADAMS	23/05/87	1100		20	7788
7902	ANALYST	FORD	03/12/81	3000		20	7566
7369	CLERK	SMITH	17/12/80	800		20	7902
7698	MANAGER	BLAKE	01/05/81	2850		30	7839
7499	SALESMAN	ALLEN	20/02/81	1600	300	30	7698
7521	SALESMAN	WARD	22/02/81	1250	500	30	7698
7654	SALESMAN	MARTIN	28/09/81	1250	1400	30	7698
7844	SALESMAN	TURNER	08/09/81	1500	0	30	7698
7900	CLERK	JAMES	03/12/81	950		30	7698
7782	MANAGER	CLARK	09/06/81	2450		10	7839
7934	CLERK	MILLER	23/01/82	1300		10	7782

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ Exemplo:

```
SQL> select deptno, ename, mgr  
2   from emp;
```

DEPTNO	ENAME	MGR
10	KING	
20	JONES	7839
20	SCOTT	7566
20	ADAMS	7788
20	FORD	7566
20	SMITH	7902
30	BLAKE	7839
30	ALLEN	7698
30	WARD	7698
30	MARTIN	7698
30	TURNER	7698

DEPTNO	ENAME	MGR
30	JAMES	7698
10	CLARK	7839
10	MILLER	7782

14 linhas selecionadas.



EACH

# SQL – Comandos para seleção de dados

✓ É possível incluir outros itens no comando **SELECT**:

- expressões aritméticas
- apelidos (aliases)
- colunas concatenadas
- literais

# SQL – Comandos para seleção de dados

✓ **SELECT** - expressões aritméticas.

✓ Operador	Operação
+	soma
-	subtração
*	multiplicação
/	divisão

# SQL – Comandos para seleção de dados

## ✓ SELECT - expressões aritméticas.

```
SQL> select ename, sal*12, comm  
2 from emp;
```

ENAME	SAL*12	COMM
-----	-----	-----
KING	60000	
JONES	35700	
SCOTT	36000	
ADAMS	13200	
FORD	36000	
SMITH	9600	
BLAKE	34200	
ALLEN	19200	300
WARD	15000	500
MARTIN	15000	1400
TURNER	18000	0

ENAME	SAL*12	COMM
-----	-----	-----
JAMES	11400	
CLARK	29400	
MILLER	15600	

14 linhas selecionadas.





# SQL – Comandos para seleção de dados

## ✓ **SELECT** - expressões aritméticas.

– Se a expressão aritmética contém mais de uma operador:

- prioridade: \* e / primeiro, + e - depois
- esquerda para a direita, em caso de operadores de prioridade igual.

# SQL – Comandos para seleção de dados

## ✓ **SELECT - expressões aritméticas.**

```
SQL> select ename, sal + 250 * 12  
2   from emp;
```

ENAME	SAL+250*12
-----	-----
KING	8000
JONES	5975
SCOTT	6000
ADAMS	4100
FORD	6000
SMITH	3800
BLAKE	5850
ALLEN	4600
WARD	4250
MARTIN	4250
TURNER	4500

ENAME	SAL+250*12
-----	-----
JAMES	3950
CLARK	5450
MILLER	4300

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ SELECT - expressões aritméticas.

- Uso de parênteses: especificar ordem na qual operadores são executados.

```
SQL> select ename, sal + 250 * 12  
2 from emp;
```

ENAME	SAL+250*12
-----	-----
KING	8000
JONES	5975
SCOTT	6000
ADAMS	4100
FORD	6000
SMITH	3800
BLAKE	5850
ALLEN	4600
WARD	4250
MARTIN	4250
TURNER	4500

ENAME	SAL+250*12
-----	-----
JAMES	3950
CLARK	5450
MILLER	4300

```
SQL> select ename, (sal + 250) * 12  
2 from emp;
```

ENAME	(SAL+250)*12
-----	-----
KING	63000
JONES	38700
SCOTT	39000
ADAMS	16200
FORD	39000
SMITH	12600
BLAKE	37200
ALLEN	22200
WARD	18000
MARTIN	18000
TURNER	21000

ENAME	(SAL+250)*12
-----	-----
JAMES	14400
CLARK	32400
MILLER	18600

# SQL – Comandos para seleção de dados

## ✓ **SELECT – ALIASES**

- Apelido para colunas.
- Cabeçalho alternativo que será mostrado.
- Padrão: maiúsculas e sem espaços.
- Mais que uma palavra: aspas duplas.

# SQL – Comandos para seleção de dados

```
SQL> select ename, sal*12 "salario anual", comm comissao  
2 from emp;
```

ENAME	salario anual	COMISSAO
KING	60000	
JONES	35700	
SCOTT	36000	
ADAMS	13200	
FORD	36000	
SMITH	9600	
BLAKE	34200	
ALLEN	19200	300
WARD	15000	500
MARTIN	15000	1400
TURNER	18000	0

ENAME	salario anual	COMISSAO
JAMES	11400	
CLARK	29400	
MILLER	15600	

14 linhas selecionadas.

# SQL – Comandos para seleção de dados

## ✓ **SELECT – concatenação de colunas**

- Permite unir colunas, expressões aritméticas ou constantes.
- Colunas de cada lado do operador são unidas, criando uma única coluna.

# SQL – Comandos para seleção de dados

```
SQL> select empno ||  
2      2      from emp;
```

```
EMPNO || ENAME
```

```
-----
```

```
---
```

```
7839KING  
7566JONES  
7788SCOTT  
7876ADAMS  
7902FORD  
7369SMITH  
7698BLAKE  
7499ALLEN  
7521WARD  
7654MARTIN  
7844TURNER
```

```
EMPNO || ENAME
```

```
-----
```

```
---
```

```
7900JAMES  
7782CLARK  
7934MILLER
```

```
14 linhas selecionadas.
```



# SQL – Comandos para seleção de dados

## ✓ **SELECT – literais**

- Qualquer caracter, expressão ou número incluído na lista do **SELECT** e que não é coluna ou *alias*
- É mostrado para cada linha retornada.
- *Strings* podem ser incluídas no resultado da consulta e tratadas como uma coluna na lista do **SELECT**.
- Datas e literais devem vir entre ‘apóstrofes’, exceto literais numéricos.





# SQL – Comandos para seleção de dados

```
SQL> set pagesize 50
SQL> select empno || '-' || ename || ' trabalha no depto ' || deptno
       2 from emp;
```

```
EMPNO || '-' || ENAME || 'TRABALHANODEPTO' || DEPTNO
```

```
-----
7839-KING trabalha no depto 10
7566-JONES trabalha no depto 20
7788-SCOTT trabalha no depto 20
7876-ADAMS trabalha no depto 20
7902-FORD trabalha no depto 20
7369-SMITH trabalha no depto 20
7698-BLAKE trabalha no depto 30
7499-ALLEN trabalha no depto 30
7521-WARD trabalha no depto 30
7654-MARTIN trabalha no depto 30
7844-TURNER trabalha no depto 30
7900-JAMES trabalha no depto 30
7782-CLARK trabalha no depto 10
7934-MILLER trabalha no depto 10
```

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT – valores nulos**

- Quando falta valor em uma coluna ► valor nulo (*null*).
- *SQL* trata corretamente tais valores.
- Não é zero!
- Valores nulos usam somente um byte para armazenamento interno.
- Se uma coluna em uma expressão é nula, a expressão é nula.

# SQL – Comandos para seleção de dados

```
SQL> select ename, sal, sal * 12 + comm anosal  
2 from emp;
```

ENAME	SAL	ANOSAL
KING	5000	
JONES	2975	
SCOTT	3000	
ADAMS	1100	
FORD	3000	
SMITH	800	
BLAKE	2850	
ALLEN	1600	19500
WARD	1250	15500
MARTIN	1250	16400
TURNER	1500	18000
JAMES	950	
CLARK	2450	
MILLER	1300	

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT – valores nulos**

- Para converter o resultado para valores nulos, usa-se a função NVL.
- A função NVL tem dois argumentos:
  1. uma expressão
  2. um valor nulo
- Pode ser usada para converter um número, data ou *string* nulos para outro número, data ou *string*, desde que os tipos de dados combinem.

## ✓ **Exemplos:**

- ✓ NVL (hiredate, '01-JAN-99');
- ✓ NVL (comm, 999);



# SQL – Comandos para seleção de dados

```
SQL> select ename, sal, sal * 12 + nvl(comm,0) anosal  
2 from emp;
```

ENAME	SAL	ANOSAL
-----	-----	-----
KING	5000	60000
JONES	2975	35700
SCOTT	3000	36000
ADAMS	1100	13200
FORD	3000	36000
SMITH	800	9600
BLAKE	2850	34200
ALLEN	1600	19500
WARD	1250	15500
MARTIN	1250	16400
TURNER	1500	18000
JAMES	950	11400
CLARK	2450	29400
MILLER	1300	15600

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT – *DISTINCT***

- qualificador usado para eliminar valores duplicados no resultado
- se múltiplas colunas forem especificadas após o **DISTINCT**, o qualificador age em todas elas.

# SQL – Comandos para seleção de dados

```
SQL> select deptno  
2 from emp;
```

DEPTNO
10
20
20
20
20
20
30
30
30
30
30
30
10
10

14 linhas selecionadas.

```
SQL> select distinct deptno  
2 from emp;
```

DEPTNO
10
20
30

# SQL – Comandos para seleção de dados

```
SQL> select distinct deptno, job  
2 from emp;
```

DEPTNO	JOB
10	CLERK
10	MANAGER
10	PRESIDENT
20	ANALYST
20	CLERK
20	MANAGER
30	CLERK
30	MANAGER
30	SALESMAN

9 linhas selecionadas.





# SQL – Comandos para seleção de dados

## ✓ **SELECT – ORDER BY**

- Classifica a ordem das linhas exibidas.
- Deve ser sempre a última cláusula do **SELECT**
- A ordem *default* é ascendente (ASC):
  - números menores primeiro
  - datas mais antigas primeiro
  - *strings* em ordem alfabética
- Para mudar *default*: palavras DESC (descendente) após o nome da coluna na cláusula **ORDER BY**.

# SQL – Comandos para seleção de dados

```
SQL> select ename, job, sal*12, deptno  
2   from emp  
3   order by ename;
```

ENAME	JOB	SAL*12	DEPTNO
ADAMS	CLERK	13200	20
ALLEN	SALESMAN	19200	30
BLAKE	MANAGER	34200	30
CLARK	MANAGER	29400	10
FORD	ANALYST	36000	20
JAMES	CLERK	11400	30
JONES	MANAGER	35700	20
KING	PRESIDENT	60000	10
MARTIN	SALESMAN	15000	30
MILLER	CLERK	15600	10
SCOTT	ANALYST	36000	20
SMITH	CLERK	9600	20
TURNER	SALESMAN	18000	30
WARD	SALESMAN	15000	30

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT – ORDER BY**

- É possível usar várias colunas na ordenação.
- Limite: quantidade de colunas da tabela.
- Para ordenar por uma coluna, não é necessário selecioná-la.

# SQL – Comandos para seleção de dados

```
SQL> select ename, job, sal*12, deptno  
2   from emp  
3   order by deptno, ename;
```

ENAME	JOB	SAL*12	DEPTNO
CLARK	MANAGER	29400	10
KING	PRESIDENT	60000	10
MILLER	CLERK	15600	10
ADAMS	CLERK	13200	20
FORD	ANALYST	36000	20
JONES	MANAGER	35700	20
SCOTT	ANALYST	36000	20
SMITH	CLERK	9600	20
ALLEN	SALESMAN	19200	30
BLAKE	MANAGER	34200	30
JAMES	CLERK	11400	30
MARTIN	SALESMAN	15000	30
TURNER	SALESMAN	18000	30
WARD	SALESMAN	15000	30

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT – ORDER BY**

- Valores nulos podem ter resultados diferentes dependendo do servidor de BD.
- Oracle: mostrados por último quando usa-se ordem ascendente e primeiro quando usa-se descendente.

# SQL – Comandos para seleção de dados

```
SQL> select ename, job, sal*12, deptno, comm  
2   from emp  
3   order by comm;
```

ENAME	JOB	SAL*12	DEPTNO	COMM
-----	-----	-----	-----	-----
TURNER	SALESMAN	18000	30	0
ALLEN	SALESMAN	19200	30	300
WARD	SALESMAN	15000	30	500
MARTIN	SALESMAN	15000	30	1400
KING	PRESIDENT	60000	10	
JONES	MANAGER	35700	20	
JAMES	CLERK	11400	30	
MILLER	CLERK	15600	10	
CLARK	MANAGER	29400	10	
SCOTT	ANALYST	36000	20	
ADAMS	CLERK	13200	20	
FORD	ANALYST	36000	20	
SMITH	CLERK	9600	20	
BLAKE	MANAGER	34200	30	

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

```
SQL> select ename, job, sal*12, deptno, comm  
2  from emp  
3  order by comm desc;
```

ENAME	JOB	SAL*12	DEPTNO	COMM
-----	-----	-----	-----	-----
KING	PRESIDENT	60000	10	
JONES	MANAGER	35700	20	
SCOTT	ANALYST	36000	20	
ADAMS	CLERK	13200	20	
FORD	ANALYST	36000	20	
SMITH	CLERK	9600	20	
JAMES	CLERK	11400	30	
MILLER	CLERK	15600	10	
CLARK	MANAGER	29400	10	
BLAKE	MANAGER	34200	30	
MARTIN	SALESMAN	15000	30	1400
WARD	SALESMAN	15000	30	500
ALLEN	SALESMAN	19200	30	300
TURNER	SALESMAN	18000	30	0

14 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT – cláusula *WHERE***

- Corresponde ao operador de restrição da Álgebra Relacional.
- Contém uma condição que as linhas devem satisfazer para serem selecionadas.
- Deve vir logo após a cláusula FROM.

## ✓ **Formato:**

SELECT colunas

FROM tabelas

WHERE expressão de condição;





# SQL – Comandos para seleção de dados

## ✓ **SELECT** – cláusula **WHERE**

- Pode comparar valores em colunas, literais, expressões aritméticas ou funções.
- Tem 3 elementos:
  - 1. um nome de coluna
  - 2. um operador de comparação
  - 3. um nome de coluna, constante ou lista de valores
- Os operadores podem ser divididos em 2 categorias: operadores lógicos e operadores SQL.

# SQL – Comandos para seleção de dados

✓ **SELECT** – cláusula **WHERE**

✓ Operadores lógicos

Operador	Operação
----------	----------

=	igual a
---	---------

>	maior que
---	-----------

>=	maior que ou igual a
----	----------------------

<	menor que
---	-----------

<=	menor que ou igual a
----	----------------------

# SQL – Comandos para seleção de dados

## ✓ **SELECT** – cláusula **WHERE**

- **Strings** e datas devem estar entre apóstrofos (' ').
- Se os caracteres de uma *string* forem maiúsculos ou minúsculos devem combinar com o conteúdo da colunas de dados, a menos que sejam alterados por uma função.

# SQL – Comandos para seleção de dados

```
SQL> select ename, empno, job, deptno
2   from emp
3   where job = 'CLERK';
```

ENAME	EMPNO	JOB	DEPTNO
ADAMS	7876	CLERK	20
SMITH	7369	CLERK	20
JAMES	7900	CLERK	30
MILLER	7934	CLERK	10

```
SQL> select ename, empno, job, deptno
2   from emp
3   where deptno > 20;
```

ENAME	EMPNO	JOB	DEPTNO
BLAKE	7698	MANAGER	30
ALLEN	7499	SALESMAN	30
WARD	7521	SALESMAN	30
MARTIN	7654	SALESMAN	30
TURNER	7844	SALESMAN	30
JAMES	7900	CLERK	30

6 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ **SELECT** – cláusula **WHERE**

- É possível comparar duas colunas da mesma linha.

```
SQL> select ename, sal, comm  
2  from emp  
3  where comm > sal;
```

ENAME	SAL	COMM
MARTIN	1250	1400

# SQL – Comandos para seleção de dados

✓ **SELECT** – cláusula **WHERE**

✓ **Operadores SQL**

**Operador**

**Operação**

**BETWEEN ... AND ...**

entre dois valores (inclusive)

**IN (*list*)**

combina com qualquer um da lista de valores

**LIKE**

combina com uma padrão

**IS NULL**

é um valor nulo



# SQL – Comandos para seleção de dados

✓ **SELECT** – cláusula *WHERE*

✓ **Operador BETWEEN**

– Testa valores de um intervalo, inclusive nos limites superior e inferior, sendo o limite inferior especificado antes.

# SQL – Comandos para seleção de dados

```
SQL> select ename, sal, comm  
2   from emp  
3   where sal between 1000 and 2000;
```

ENAME	SAL	COMM
ADAMS	1100	
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400
TURNER	1500	0
MILLER	1300	

6 linhas selecionadas.





# SQL – Comandos para seleção de dados

✓ **SELECT** – cláusula **WHERE**

✓ **Operador IN**

- Compara valores de uma lista
- Se os valores são do tipo *string* ou *data*, devem estar entre apóstrofos.

# SQL – Comandos para seleção de dados

```
SQL> select empno, ename, sal, mgr  
2   from emp  
3   where mgr in (7902, 7566, 7788);
```

EMPNO	ENAME	SAL	MGR
7788	SCOTT	3000	7566
7876	ADAMS	1100	7788
7902	FORD	3000	7566
7369	SMITH	800	7902



# SQL – Comandos para seleção de dados

## ✓ **SELECT** – cláusula **WHERE**

## ✓ **Operador LIKE**

- Usado quando o valor exato de uma *string* não é conhecido.
- Permite selecionar linhas cujas colunas combinem em parte com o argumento da pesquisa.
- Podem ser usados caracteres coringas para construir a *string* de pesquisa.



# SQL – Comandos para seleção de dados

✓ **SELECT** – cláusula *WHERE*

✓ **Operador LIKE - Coringas**

**Símbolo**

**Significado**

%

qualquer seqüência de zero ou mais caracteres

\_

qualquer caracter único

# SQL – Comandos para seleção de dados

```
SQL> select ename  
2   from emp  
3   where ename like 'S%';
```

ENAME

-----

SCOTT

SMITH

```
SQL> select ename  
2   from emp  
3   where ename like 'S_';
```

não há linhas selecionadas



# SQL – Comandos para seleção de dados

✓ **SELECT – cláusula *WHERE***

✓ **Operador IS NULL**

✓ **Testa valores que sejam nulos.**

```
SQL> select ename, mgr
2   from emp
3   where mgr is null;
```

ENAME	MGR
-----	-----
KING	

# SQL – Comandos para seleção de dados

## ✓ **SELECT** – expressões negativas

### Operador

### Operação

!=

diferente (VAX, UNIX, PC)

^=

diferente (IBM)

< >

diferente (todos SO)

NOT coluna

= diferente

NOT coluna >

não maior (menor ou igual) que

# SQL – Comandos para seleção de dados

✓ **SELECT** – expressões negativas (oper. SQL)

## Operador SQL

## Operação

NOT BETWEEN

não entre 2 valores dados

NOT IN

não na lista de valores dados

NOT LIKE

não semelhante a uma *string* dada

IS NOT NULL

não é valor nulo



# SQL – Comandos para seleção de dados

```
SQL> select ename, sal  
2   from emp  
3   where sal not between 1000 and 2000;
```

ENAME	SAL
KING	5000
JONES	2975
SCOTT	3000
FORD	3000
SMITH	800
BLAKE	2850
JAMES	950
CLARK	2450

8 linhas selecionadas.

# SQL – Comandos para seleção de dados

```
SQL> select ename, mgr
      2  from emp
      3  where mgr is not null;
```

ENAME	MGR
JONES	7839
SCOTT	7566
ADAMS	7788
FORD	7566
SMITH	7902
BLAKE	7839
ALLEN	7698
WARD	7698
MARTIN	7698
TURNER	7698
JAMES	7698
CLARK	7839
MILLER	7782

13 linhas selecionadas.



# SQL – Comandos para seleção de dados

```
SQL> select empno, ename, sal, mgr  
2   from emp  
3   where mgr not in (7902, 7566, 7788);
```

EMPNO	ENAME	SAL	MGR
7566	JONES	2975	7839
7698	BLAKE	2850	7839
7499	ALLEN	1600	7698
7521	WARD	1250	7698
7654	MARTIN	1250	7698
7844	TURNER	1500	7698
7900	JAMES	950	7698
7782	CLARK	2450	7839
7934	MILLER	1300	7782

9 linhas selecionadas.



# SQL – Comandos para seleção de dados

- ✓ **Buscas com condições múltiplas**

- ✓ **Operadores AND e OR:**

AND - todas as condições satisfeitas

OR - pelo menos uma condição satisfeita

# SQL – Comandos para seleção de dados

## ✓ Exemplo:

```
SQL> select empno, ename, job, sal  
2   from emp  
3   where sal between 1000 and 2000  
4   and job = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7876	ADAMS	CLERK	1100
7934	MILLER	CLERK	1300

# SQL – Comandos para seleção de dados

## ✓ Exemplo:

```
SQL> select empno, ename, job, sal  
2   from emp  
3   where sal between 1000 and 2000  
4   or job = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7876	ADAMS	CLERK	1100
7369	SMITH	CLERK	800
7499	ALLEN	SALESMAN	1600
7521	WARD	SALESMAN	1250
7654	MARTIN	SALESMAN	1250
7844	TURNER	SALESMAN	1500
7900	JAMES	CLERK	950
7934	MILLER	CLERK	1300

8 linhas selecionadas.

# SQL – Comandos para seleção de dados

## ✓ SELECT - AND e OR

– AND tem precedência em relação ao OR

```
SQL> select empno, ename, job, sal, deptno
2   from emp
3   where  sal > 1500
4     and   job = 'MANAGER'
5     or    job = 'SALESMAN';
```

EMPNO	ENAME	JOB	SAL	DEPTNO
7566	JONES	MANAGER	2975	20
7698	BLAKE	MANAGER	2850	30
7499	ALLEN	SALESMAN	1600	30
7521	WARD	SALESMAN	1250	30
7654	MARTIN	SALESMAN	1250	30
7844	TURNER	SALESMAN	1500	30
7782	CLARK	MANAGER	2450	10

7 linhas selecionadas.

# SQL – Comandos para seleção de dados

## ✓ **SELECT - AND e OR**

– AND tem precedência em relação ao OR

```
SQL> select empno, ename, job, sal, deptno
2    from emp
3    where sal > 1500
4      and (job = 'MANAGER'
5      or    job = 'SALESMAN' );
```

EMPNO	ENAME	JOB	SAL	DEPTNO
7566	JONES	MANAGER	2975	20
7698	BLAKE	MANAGER	2850	30
7499	ALLEN	SALESMAN	1600	30
7782	CLARK	MANAGER	2450	10



# SQL – Comandos para seleção de dados

## ✓ **SELECT - Precedência de operadores**

- Em uma expressão os operadores são executados na ordem de sua precedência.
- Se houver operadores de mesma precedência ► executa-se da esquerda para a direita.
- Ordem:
  1. Operadores de comparação e operadores SQL:  
=, <>, <, >, <=, >=, BETWEEN, IN, LIKE, IS NULL
  2. NOT
  3. AND
  4. OR
- Se houver dúvida ou falta de clareza ► usar parênteses.



# SQL – Comandos para seleção de dados

## ✓ SELECT - AND e OR

- Às vezes os parênteses são desnecessários, mas tornam a leitura mais clara.

```
SQL> select ename, job,deptno
2   from emp
3   where job = 'MANAGER'
4   or    ( job = 'CLERK'
5   or      deptno = 10);
```

ENAME	JOB	DEPTNO
-----	-----	-----
KING	PRESIDENT	10
JONES	MANAGER	20
ADAMS	CLERK	20
SMITH	CLERK	20
BLAKE	MANAGER	30
JAMES	CLERK	30
CLARK	MANAGER	10
MILLER	CLERK	10

8 linhas selecionadas.



# SQL – Comandos para seleção de dados

## ✓ RESUMO DO SELECT

### ✓ Formato geral:

SELECT [DISTINCT] { \*, coluna [alias], ... }

FROM tabela

WHERE condições

ORDER BY {coluna, expressão,...} [ASC|DESC]

onde: >>>>>

# SQL – Comandos para seleção de dados

## ✓ RESUMO DO SELECT

<b>SELECT</b>	Seleciona pelo menos uma coluna
<b>Alias</b>	Pode ser usado para colunas da lista do SELECT
<b>*</b>	Todas as colunas da tabela
<b>DISTINCT</b>	Elimina duplicatas
<b>FROM tabela</b>	Identifica a tabela de onde as colunas são extraídas.
<b>WHERE</b>	Restringe as linhas que satisfazem uma condição. Pode conter colunas, valores, expressões e literais.
<b>AND/OR</b>	Podem ser usados em uma cláusula WHERE para construir condições múltiplas.
<b>( )</b>	Pode ser usados para forçar uma prioridade.
<b>ORDER BY</b>	Aparece sempre por último. Identifica a ordem de classificação. Uma ou mais colunas podem ser especificadas.
<b>ASC</b>	Ascendente – ordem de classificação padrão. Não precisa ser especificada.
<b>DESC</b>	Inverte a ordem padrão de classificação e deve ser especificadas após o nome da coluna.

# SQL – Comandos para seleção de dados

## ✓ JOINS

- Usado quando uma *query* necessita de dados de mais de uma tabela.
- As linhas de uma tabela podem ser juntadas às linhas de outra tabela.
- *JOIN* é especificado na cláusula **WHERE**.
- Sua forma mais simples é o *equijoin*, quando os valores em uma tabela são iguais aos desejados da outra tabela. Corresponde geralmente à igualdade entre chaves do Modelo Relacional.

# SQL – Comandos para seleção de dados

## ✓ JOINS

- Se a condição do *join* é omitida ou inválida, o resultado é um produto, que gera um grande número de linhas, normalmente inúteis. Corresponde ao resultado do Produto Cartesiano da Álgebra Relacional.

# SQL – Comandos para seleção de dados

## ✓ JOINS

- Exemplos: Tabelas EMP e DEPT

```
SQL> DESCRIBE DEPT;
```

Name	Null?	Type
-----	-----	-----
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

```
SQL> SELECT * FROM DEPT;
```

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



# SQL – Comandos para seleção de dados

## ✓ JOINS

- Exemplos: Tabelas EMP e DEPT

```
SELECT ENAME, JOB, DNAME  
FROM EMP, DEPT  
WHERE EMP.DEPTNO = DEPT.DEPTNO;
```

ENAME	JOB	DNAME
-----	-----	-----
SMITH	CLERK	RESEARCH
ALLEN	SALESMAN	SALES
WARD	SALESMAN	SALES
JONES	MANAGER	RESEARCH
MARTIN	SALESMAN	SALES
BLAKE	MANAGER	SALES
CLARK	MANAGER	ACCOUNTING
SCOTT	ANALYST	RESEARCH
KING	PRESIDENT	ACCOUNTING
TURNER	SALESMAN	SALES
ADAMS	CLERK	RESEARCH
JAMES	CLERK	SALES
FORD	ANALYST	RESEARCH
MILLER	CLERK	ACCOUNTING





# SQL – Comandos para seleção de dados

## ✓ JOINS

- Exemplos: Tabelas EMP e DEPT

```
SELECT DEPT.DEPTNO, ENAME, JOB, DNAME
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
ORDER BY DEPT.DEPTNO;
```

DEPTNO	ENAME	JOB	DNAME
10	CLARK	MANAGER	ACCOUNTING
10	KING	PRESIDENT	ACCOUNTING
10	MILLER	CLERK	ACCOUNTING
20	SMITH	CLERK	RESEARCH
20	ADAMS	CLERK	RESEARCH
20	FORD	ANALYST	RESEARCH
20	SCOTT	ANALYST	RESEARCH
20	JONES	MANAGER	RESEARCH
30	ALLEN	SALESMAN	SALES
30	BLAKE	MANAGER	SALES
30	MARTIN	SALESMAN	SALES
30	JAMES	CLERK	SALES
30	TURNER	SALESMAN	SALES
30	WARD	SALESMAN	SALES

# SQL – Comandos para seleção de dados

## ✓ APELIDOS (ALIASES)

- Usados para simplificar os comandos SQL que utilizam *joins*.
- Os nomes das tabelas podem ser substituídos por *aliases* na cláusula FROM.
- Os *aliases* podem ser referenciados nas demais partes do comando.
- Os *aliases* podem ser longos (30 caracteres), mas nomes curtos facilitam o uso posterior.



# SQL – Comandos para seleção de dados

## ✓ APELIDOS (ALIASES)

- Exemplo:

```
SELECT E.ENAME, D.DEPTNO, D.DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
ORDER BY D.DEPTNO;
```

ENAME	DEPTNO	DNAME
-----	-----	-----
CLARK	10	ACCOUNTING
KING	10	ACCOUNTING
MILLER	10	ACCOUNTING
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
FORD	20	RESEARCH
SCOTT	20	RESEARCH
JONES	20	RESEARCH
ALLEN	30	SALES
BLAKE	30	SALES
MARTIN	30	SALES
JAMES	30	SALES
TURNER	30	SALES
WARD	30	SALES

# SQL – Comandos para seleção de dados

## ✓ NON-EQUI-JOIN

- Algumas vezes não há colunas comuns em tabelas envolvidas em um relacionamento.
- Neste caso, utiliza-se um *non-equi-join*, que consiste em um relacionamento obtido com um operador que não é a igualdade (=), como, por exemplo, o *between*.

# SQL – Comandos para seleção de dados

## ✓ NON-EQUI-JOIN

- Exemplo: Tabelas EMP e SALGRADE

```
DESCRIBE SALGRADE;
```

Name	Null?	Type
-----	-----	----
GRADE		NUMBER
LOSAL		NUMBER
HISAL		NUMBER

```
SELECT * FROM SALGRADE;
```

GRADE	LOSAL	HISAL
-----	-----	-----
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999



# SQL – Comandos para seleção de dados

## ✓ NON-EQUI-JOIN

- Exemplo: Tabelas EMP e SALGRADE

```
SELECT E.ENAME, E.SAL, S.GRADE
FROM EMP E, SALGRADE S
WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

ENAME	SAL	GRADE
SMITH	800	1
ADAMS	1100	1
JAMES	950	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
ALLEN	1600	3
TURNER	1500	3
JONES	2975	4
BLAKE	2850	4
CLARK	2450	4
SCOTT	3000	4
FORD	3000	4
KING	5000	5

# SQL – Comandos para seleção de dados

## ✓ JOIN na mesma tabela

- É possível fazer um *join* de uma tabela com ela mesma, como se fossem duas tabelas separadas (auto-relacionamento).
- Isto permite que tuplas de uma relação sejam combinadas com tuplas da mesma relação.
- Exemplo:

# SQL – Comandos para seleção de dados

## ✓ JOIN na mesma tabela

- Exemplo:

```
SELECT E.ENAME "Nome Func.", E.SAL "Sal Func",  
       G.ENAME "Nome Ger.", G.SAL "Sal. Ger."  
FROM EMP E, EMP G  
WHERE E.MGR = G.EMPNO AND E.SAL < G.SAL;
```

Nome Func.	Sal Func	Nome Ger.	Sal. Ger.
-----	-----	-----	-----
SMITH	800	FORD	3000
ALLEN	1600	BLAKE	2850
WARD	1250	BLAKE	2850
JONES	2975	KING	5000
MARTIN	1250	BLAKE	2850
BLAKE	2850	KING	5000
CLARK	2450	KING	5000
TURNER	1500	BLAKE	2850
ADAMS	1100	SCOTT	3000
JAMES	950	BLAKE	2850
MILLER	1300	CLARK	2450



# SQL – Comandos para seleção de dados

## ✓ OPERADORES DE CONJUNTO

- São úteis em *queries* que referenciam diferentes tabelas.
- Combinam resultados de dois ou mais comandos *select* em um único resultado.
- São também conhecidos como *joins* verticais.

# SQL – Comandos para seleção de dados

## ✓ OPERADORES DE CONJUNTO

### ► Operador UNION

- Retorna todas as linhas distintas recuperadas pelas *queries*.
- **Exemplo:**

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 30;
```

```
JOB  
-----  
CLERK  
MANAGER  
PRESIDENT  
SALESMAN
```



# SQL – Comandos para seleção de dados

## ✓ OPERADORES DE CONJUNTO

### ► Operador INTERSECT

- Retorna somente as linhas recuperadas em ambas as *queries*.
- **Exemplo:**

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
INTERSECT  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 30;
```

```
JOB  
-----  
CLERK  
MANAGER
```



# SQL – Comandos para seleção de dados

## ✓ OPERADORES DE CONJUNTO

### ► Operador MINUS

- Retorna somente as linhas da primeira *query* que não estão presentes no resultado da segunda *query*.
- **Exemplo:**

```
SELECT JOB
FROM EMP
WHERE DEPTNO = 10
MINUS
SELECT JOB
FROM EMP
WHERE DEPTNO = 30;

JOB
-----
PRESIDENT
```

# SQL – Comandos para seleção de dados

## ✓ OPERADORES DE CONJUNTO

### ► Regras no uso de operadores de conjunto

- **Compatibilidade:**
  - Comandos SELECT devem selecionar o mesmo número de colunas.
  - As colunas correspondentes devem ser do mesmo *datatype*.
- Linhas duplicadas são automaticamente eliminadas e a cláusula DISTINCT não pode ser usada.
- Os nomes das colunas da primeira *query* são os que aparecem no resultado.



# SQL – Comandos para seleção de dados

## ✓ OPERADORES DE CONJUNTO

### ► Regras no uso de operadores de conjunto

- A cláusula ORDER BY pode aparecer no final do comando.
- Os operadores de conjunto podem ser usados em *subqueries*.
- Os comandos SELECT são executados de cima para baixo.
- Vários operadores de conjunto podem ser usados com parênteses.

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

- Operam em um conjunto de linhas.
- Retornam resultados baseados em grupos de linhas, ao invés de um resultado por linha.

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

Função	Valor Retornado
<b>AVG (DISTINCT   ALL   n)</b>	Média de <i>n</i> , ignorando <i>nulls</i>
<b>COUNT (DISTINCT   ALL   expr)</b>	Número de vezes que <i>expr</i> não retorna nulo
<b>MAX (DISTINCT   ALL   expr)</b>	Máximo valor de <i>expr</i>
<b>MIN (DISTINCT   ALL   expr)</b>	Mínimo valor de <i>expr</i>
<b>STDDEV (DISTINCT   ALL   n)</b>	Desvio padrão de <i>n</i> , ignorando <i>nulls</i> .
<b>SUM (DISTINCT   ALL   n)</b>	Soma valores de <i>n</i> , ignorando <i>nulls</i> .
<b>VARIANCE (DISTINCT   ALL   n)</b>	Variância de <i>n</i> , ignorando <i>nulls</i> .





# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Observações:

- Cláusula **DISTINCT**: faz a função considerar somente valores distintos.
- Cláusula **ALL**: opção *default*, inclui os valores duplicados.
- Tipos de dados de *expr* podem ser **CHAR**, **NUMBER** ou **DATE**
- Todas as funções, exceto **COUNT(\*)** ignoram valores nulos.



# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

- Exemplos:

```
SQL> SELECT AVG(SAL) FROM EMP;
```

```
AVG(SAL)
```

```
-----
```

```
2073,2143
```

```
SQL> SELECT MIN(SAL) FROM EMP;
```

```
MIN(SAL)
```

```
-----
```

```
800
```



# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

- Exemplos:

```
SQL>  SELECT COUNT ( * )  
      2  FROM EMP  
      3  WHERE DEPTNO = 20 ;
```

```
      COUNT ( * )
```

```
-----
```

```
              5
```

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Cláusula GROUP BY

- Pode ser usada para dividir um conjunto de linhas em grupos menores, enquanto que as funções de grupo podem ser usadas para retornar informação sobre cada grupo.

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Cláusula GROUP BY

- Exemplo:

```
SQL>  SELECT JOB, AVG(SAL)
      2  FROM EMP
      3  GROUP BY JOB;
```

JOB	AVG(SAL)
-----	-----
ANALYST	3000
CLERK	1037,5
MANAGER	2758,3333
PRESIDENT	5000



EACH

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Exclusão de linhas da seleção

- Se alguma linha não deve ser incluída na seleção, ela deve ser excluída com a cláusula **WHERE**.

- Exemplo:

```
SQL> SELECT JOB, AVG(SAL)
      2 FROM EMP
      3 WHERE JOB <> 'MANAGER'
      4 GROUP BY JOB;
```

JOB	AVG(SAL)
ANALYST	3000
CLERK	1037,5

ACH



# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Grupos dentro de grupos

- A cláusula **GROUP BY** também pode ser usada para fornecer resultados de grupos dentro dos grupos.
- Exemplo:

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Grupos dentro de grupos

- Exemplo:

```
SQL> SELECT DEPTNO, JOB, AVG(SAL)
      2 FROM EMP
      3 GROUP BY DEPTNO, JOB;
```

DEPTNO	JOB	AVG(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	3000
20	CLERK	950





# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Cláusula HAVING

- Especifica quais grupos devem ser mostrados.
- Observação: a cláusula WHERE é usada somente para restringir as linhas que serão selecionadas para a formação dos grupos, enquanto a cláusula HAVING é usada para *restringir* os grupos já formados.

# SQL – Comandos para seleção de dados

## ✓ FUNÇÕES DE GRUPO

### ► Cláusula HAVING

- Exemplo:

```
SQL> SELECT DEPTNO, AVG(SAL)
      2 FROM EMP
      3 GROUP BY DEPTNO
      4 HAVING COUNT(*) > 3;
```

DEPTNO	AVG(SAL)
20	2175
30	1566,6667

# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

- Comando **SELECT** aninhado dentro de outro comando **SELECT**.

- **Formato geral:**

**SELECT** coluna1, coluna2,..

**FROM** tabela 1

**WHERE** colunaN = (**SELECT** colunaM

**FROM** tabela 2

**WHERE** condição)



# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

- Exemplos:

```
SQL> SELECT ENAME, JOB, SAL  
      FROM EMP  
      WHERE SAL = (SELECT MIN(SAL) FROM EMP);
```

ENAME	JOB	SAL
SMITH	CLERK	800

# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

- Exemplos:

```
SQL>  SELECT ENAME, JOB
        FROM    EMP
        WHERE   JOB = (SELECT JOB
                        FROM EMP
                        WHERE ENAME = 'BLAKE' );
```

ENAME	JOB
JONES	MANAGER
BLAKE	MANAGER
CLARK	MANAGER

# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

- Retorno de mais de uma linha
- Observação: uso do operador IN no exemplo.

- Exemplo:

```
SQL> SELECT ENAME, SAL, DEPTNO
      FROM EMP
      WHERE SAL IN (SELECT MIN(SAL)
                    FROM EMP
                    GROUP BY DEPTNO;
```

ENAME	SAL	DEPTNO
-------	-----	--------

-----	-----	-----
-------	-------	-------

SMITH	800	20
-------	-----	----

JAMES	950	30
-------	-----	----

MILLER	1300	10
--------	------	----

# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

- Comparação de mais de um valor
- Observação: ordem das colunas

- Exemplo:

```
SQL> SELECT ENAME, SAL
      FROM EMP
      WHERE (SAL,DEPTNO) IN (SELECT MIN(SAL), DEPTNO
                             FROM EMP
                             GROUP BY DEPTNO);
```

ENAME	SAL
SMITH	800
JAMES	950
MILLER	1300



# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

### – Operadores ANY e ALL

- Usados em *subqueries* que retornam mais de uma linha.
- Usados nas cláusulas WHERE ou HAVING em conjunto com operadores lógicos.
- ANY compara um valor a cada valor retornado pela *subquery*.



# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

### – Operadores ANY e ALL

- Exemplo:

```
SQL> SELECT ENAME, SAL, JOB, DEPTNO
      FROM EMP
      WHERE SAL > ANY (SELECT DISTINCT SAL
                        FROM EMP
                        WHERE DEPTNO=30);
```

ENAME	SAL	JOB	DEPTNO
ALLEN	1600	SALESMAN	30
WARD	1250	SALESMAN	30
JONES	2975	MANAGER	20
MARTIN	1250	SALESMAN	30
BLAKE	2850	MANAGER	30
CLARK	2450	MANAGER	10



# SQL – Comandos para seleção de dados

## ✓ SUBQUERIES

### – Operadores ANY e ALL

- ALL compara um valor a todos os valores retornados pela *subquery*.

- Exemplo:

```
SQL> SELECT ENAME, SAL, JOB, DEPTNO
      FROM EMP
      WHERE SAL > ALL (SELECT DISTINCT SAL
                        FROM EMP
                        WHERE DEPTNO=30);
```

ENAME	SAL	JOB	DEPTNO
JONES	2975	MANAGER	20
SCOTT	3000	ANALYST	20



# ACH2025

## Laboratório de Bases de Dados

### Aula 6

**SQL Oracle – Visão geral – SELECT**

**Professora:**

➤ **Fátima L. S. Nunes**

