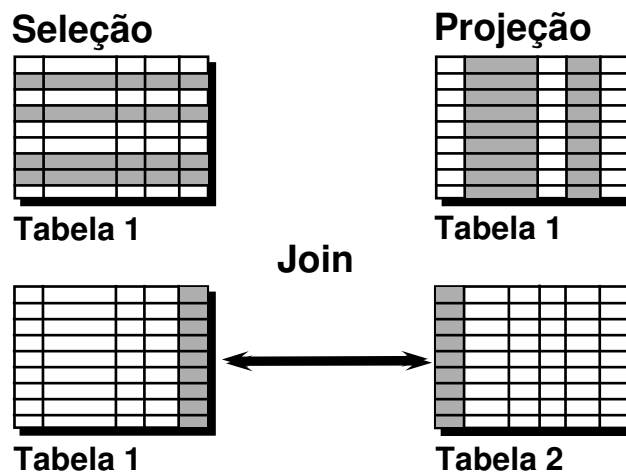


# Aula 16

## Introdução ao SQL Selecionando Informações

UNISUL – Computação - Banco de Dados

### Capacidades do comando SELECT



1-2

UNISUL – Computação - Banco de Dados

## SELECT Básico

```
SELECT  [DISTINCT] {*, column [alias],...}  
FROM    table;
```

- **SELECT** identifica *que* colunas
- **FROM** identifica *quais* tabelas

1-3

UNISUL – Computação - Banco de Dados

## Escrevendo Sentenças SQL

- Sentenças SQL não são case sensitive.
- Sentenças SQL podem ter uma ou mais linhas.
- Comandos chave não podem ser abreviados.
- Clausulas são usualmente colocadas em linhas específicas.
- O uso de tabulação é importante para o melhor entendimento do comando.

1-4

UNISUL – Computação - Banco de Dados

## Selecionando Todas as Colunas

```
SQL> SELECT *  
2 FROM dept;
```

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

1-5

UNISUL – Computação - Banco de Dados

## Selecionando Colunas Específicas

```
SQL> SELECT deptno, loc  
2 FROM dept;
```

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

1-6

UNISUL – Computação - Banco de Dados

## Padrão para Display de Colunas

- **Justificação Padrão**
  - Left: Data e caracteres
  - Right: Dados Numéricos
- **Display padrão: Uppercase**

1-7

UNISUL – Computação - Banco de Dados

## Expressões Aritméticas

**Criar expressões sobre dados tipo NUMBER e DATE usando operadores aritméticos.**

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão

1-8

UNISUL – Computação - Banco de Dados

## Utilização de Operadores Aritméticos

```
SQL> SELECT ename, sal, sal+300  
2 FROM emp;
```

ENAME	SAL	SAL+300
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
MARTIN	1250	1550
ALLEN	1600	1900
...		

14 rows selected.

1-9

UNISUL – Computação - Banco de Dados

## Precedência de Operadores

\* / + -

- Multiplicação e divisão têm prioridade sobre adição e subtração.
- Operadores com mesma prioridade são avaliados da esquerda para a direita.
- Faz-se o uso de parênteses para forçar maior prioridade e permitir clareza à sentença.

1-10

UNISUL – Computação - Banco de Dados

## Precedência de Operadores

```
SQL> SELECT ename, sal, 12*sal+100  
2 FROM emp;
```

ENAME	SAL	12*SAL+100
KING	5000	60100
BLAKE	2850	34300
CLARK	2450	29500
JONES	2975	35800
MARTIN	1250	15100
ALLEN	1600	19300
...		

14 rows selected.

## Utilização de Parênteses

```
SQL> SELECT ename, sal, 12*(sal+100)  
2 FROM emp;
```

ENAME	SAL	12*(SAL+100)
KING	5000	61200
BLAKE	2850	35400
CLARK	2450	30600
JONES	2975	36900
MARTIN	1250	16200
...		

14 rows selected.

## Definição de Valores Null

- O valor nulo é um valor que não pode ser avaliado, reconhecido.
- Um valor nulo não tem valor algum, não sendo um valor comparável ao valor zero ou ainda um espaço em branco.

```
SQL> SELECT  ename, job, comm
2 FROM      emp;
```

ENAME	JOB	COMM
KING	PRESIDENT	
BLAKE	MANAGER	
...		
TURNER	SALESMAN	0
...		

14 rows selected.

1-13

UNISUL – Computação - Banco de Dados

## Valores Null em Expressões Aritméticas

Expressões Aritméticas que contém um valor null é avaliado como null.

```
SQL> select  ename NAME, 12*sal+comm
2 from      emp
3 WHERE     ename='KING';
```

NAME	12*SAL+COMM
KING	

1-14

UNISUL – Computação - Banco de Dados

## Definição de um Alias para uma Coluna

- Renomeia o cabeçalho da coluna
- Tem muito uso com cálculos
- Colocados imediatamente após o nome da coluna
- Requer aspas duplas quando conter espaços em branco ou for case sensitive

1-15

UNISUL – Computação - Banco de Dados

## Utilização de Alias para Colunas

```
SQL> SELECT ename AS name, sal salary  
2 FROM emp;
```

NAME	SALARY
-----	-----
...	

```
SQL> SELECT ename "Name",  
2 sal*12 "Annual Salary"  
3 FROM emp;
```

Name	Annual Salary
-----	-----
...	

1-16

UNISUL – Computação - Banco de Dados



## Operador de Concatenação

- Concatena colunas ou caracteres strings para outras colunas
- É representado por duas barras verticais (||)
- Cria uma coluna resultante do tipo caracter

1-17

UNISUL – Computação - Banco de Dados

## Utilização do Operador de Concatenação

```
SQL> SELECT  ename||job AS "Employees"  
2  FROM      emp;
```

```
Employees  
-----  
KINGPRESIDENT  
BLAKEMANAGER  
CLARKMANAGER  
JONESMANAGER  
MARTINSALESMAN  
ALLENSALESMAN  
...  
14 rows selected.
```

1-18

UNISUL – Computação - Banco de Dados

## Caracteres Strings Literal

- Uma literal é um caracter, expressão ou número incluído num comando SELECT.
- Datas e caracteres literais devem estar representados entre aspas simples.

1-19

UNISUL – Computação - Banco de Dados

## Utilização de Caracteres Literais

```
SQL> SELECT ename || ' ' || 'is a' || ' ' || job  
2          AS "Employee Details"  
3 FROM    emp;
```

```
Employee Details  
-----  
KING is a PRESIDENT  
BLAKE is a MANAGER  
CLARK is a MANAGER  
JONES is a MANAGER  
MARTIN is a SALESMAN  
...  
14 rows selected.
```

1-20

UNISUL – Computação - Banco de Dados

## Linhas Duplicadas

Por padrão, uma consulta lista todas as linhas encontradas na base de dados, inclusive se estiverem duplicadas.

```
SQL> SELECT deptno  
2 FROM emp;
```

```
DEPTNO  
-----  
10  
30  
10  
20  
...  
14 rows selected.
```

1-21

UNISUL – Computação - Banco de Dados

## Eliminação de Linhas Duplicadas

Para eliminar linhas duplicadas utiliza-se a palavra-chave **DISTINCT** logo após o comando **SELECT**.

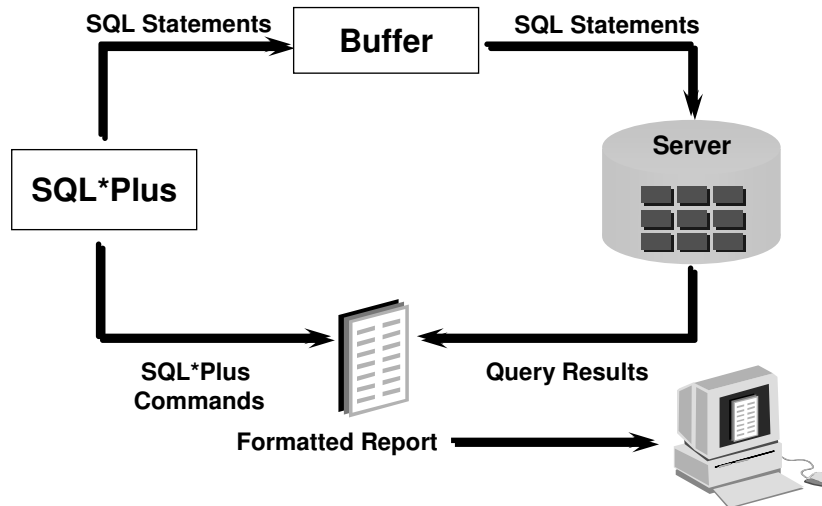
```
SQL> SELECT DISTINCT deptno  
2 FROM emp;
```

```
DEPTNO  
-----  
10  
20  
30
```

1-22

UNISUL – Computação - Banco de Dados

## Interação entre SQL e SQL\*Plus



1-23

UNISUL – Computação - Banco de Dados

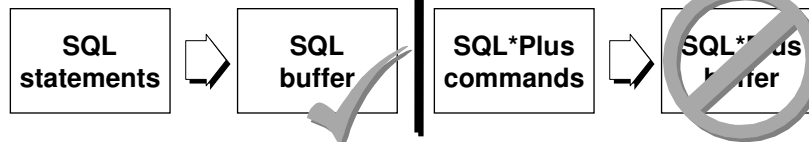
## Sentenças SQL X Comandos SQL\*Plus

### SQL

- Linguagem
- Padrão ANSI
- Palavras-chave não podem ser abreviadas
- Sentenças manipulam dados e objetos na base de dados

### SQL\*Plus

- Ambiente
- Propriedade da Oracle
- Palavras-chave podem ser abreviadas
- Não manipulam dados e objetos na base de dados

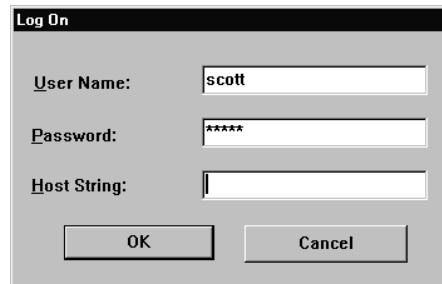


1-24

UNISUL – Computação - Banco de Dados

## Acesso ao SQL\*Plus

- Para ambientes Windows:



- Através de linha de comando:

```
sqlplus [username[/password  
[@database]]]
```

1-25

UNISUL – Computação - Banco de Dados

## Listar a Estrutura das Tabelas

Através do comando SQL\*Plus  
**DESCRIBE.**

```
DESC[RIBE] tablename
```

1-26

UNISUL – Computação - Banco de Dados

## Listando a Estrutura das Tabelas

```
SQL> DESCRIBE dept
```

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

1-27

UNISUL – Computação - Banco de Dados

## Comandos de Arquivo do SQL\*Plus

- **SAVE *filename***
- **GET *filename***
- **START *filename***
- **@ *filename***
- **EDIT *filename***
- **SPOOL *filename***

1-28

UNISUL – Computação - Banco de Dados

# Exercícios

1-29

UNISUL – Computação - Banco de Dados

# Restringindo e Ordenando Dados

1-30

UNISUL – Computação - Banco de Dados

## Limitando a Seleção de Linhas

EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

"...retornar todos os empregados no departamento 10"



EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7782	CLARK	MANAGER		10
7934	MILLER	CLERK		10

1-31

UNISUL – Computação - Banco de Dados

## Limitando a Seleção de Linhas

- Restrição de linhas retornadas através da cláusula WHERE.

```
SELECT      [DISTINCT] {*, column [alias], ...}
FROM        table
[WHERE      condition(s)];
```

- A cláusula WHERE segue a cláusula FROM.

1-32

UNISUL – Computação - Banco de Dados



## Utilização da Cláusula WHERE

```
SQL> SELECT ename, job, deptno  
2 FROM emp  
3 WHERE job='CLERK';
```

ENAME	JOB	DEPTNO
JAMES	CLERK	30
SMITH	CLERK	20
ADAMS	CLERK	20
MILLER	CLERK	10

1-33

UNISUL – Computação - Banco de Dados

## Caracteres tipo String e Date

- Caracteres string e date são descritos entre aspas simples
- Os valores tipo Character são case-sensitive e valores tipo Date são format-sensitive
- O formato padrão para um valor tipo Date é ' DD-MON-YY '

```
SQL> SELECT ename, job, deptno  
2 FROM emp  
3 WHERE ename = 'JAMES';
```

1-34

UNISUL – Computação - Banco de Dados

## Operadores de Comparação

Operador	Significado
=	Igual a
>	Maior que
>=	Maior que ou igual a
<	Menor que
<=	Menor que ou igual a
<>	Não igual a (diferente de)

1-35

UNISUL – Computação - Banco de Dados

## Utilização de Operadores de Comparação

```
SQL> SELECT ename, sal, comm
2 FROM emp
3 WHERE sal<=comm;
```

ENAME	SAL	COMM
MARTIN	1250	1400

1-36

UNISUL – Computação - Banco de Dados

## Outros Operadores de Comparação

Operador	Significado
BETWEEN	Entre dois valores (inclusive)
...AND...	
IN(list)	Pertence a lista de valores
LIKE	Pertence ao padrao informado
IS NULL	É um valor null

1-37

UNISUL – Computação - Banco de Dados

## Utilização do Operador BETWEEN

O uso do operador BETWEEN permite resgatar dados baseados numa faixa de valores.

```
SQL> SELECT  ename, sal
2 FROM      emp
3 WHERE     sal BETWEEN 1000 AND 1500;
```

ENAME	SAL		
MARTIN	1250	Limite Inferior	Limite Superior
TURNER	1500		
WARD	1250		
ADAMS	1100		
MILLER	1300		

1-38

UNISUL – Computação - Banco de Dados

## Utilização do Operador IN

Utiliza-se o operador IN para testar valores em uma lista.

```
SQL> SELECT empno, ename, sal, mgr
2 FROM emp
3 WHERE mgr IN (7902, 7566, 7788);
```

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

1-39

UNISUL – Computação - Banco de Dados

## Operador LIKE

- Utiliza-se o operador LIKE para realizar pesquisas com base em coringas em valores string.
- A condição de consulta pode conter qualquer caracter literal ou números.
  - % para zero ou qualquer caracter
  - \_ para um caracter qualquer

```
SQL> SELECT ename
2 FROM emp
3 WHERE ename LIKE 'S%';
```

1-40

UNISUL – Computação - Banco de Dados

## Utilização do Operador LIKE

- Podem ser combinados padrões de caracteres.

```
SQL> SELECT  ename
2 FROM      emp
3 WHERE     ename LIKE '_A%';
```

ENAME
JAMES
WARD

1-41

UNISUL – Computação - Banco de Dados

## Utilização do Operador NULL

### Testar valores null com o operador NULL

```
SQL> SELECT  ename, mgr
2 FROM      emp
3 WHERE     mgr IS NULL;
```

ENAME	MGR
KING	

1-42

UNISUL – Computação - Banco de Dados

## Operadores Lógicos

Operador	Significado
AND	Retorna TRUE caso ambos os componentes forem TRUE
OR	Retorna TRUE caso algum componente for TRUE
NOT	Retorna TRUE caso todas as componentes forem FALSE

1-43

UNISUL – Computação - Banco de Dados

## Utilização do Operador AND

**AND** requer ambas as condições serem TRUE.

```
SQL> SELECT empno, ename, job, sal
2 FROM emp
3 WHERE sal>=1100
4 AND job='CLERK';
```

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

1-44

UNISUL – Computação - Banco de Dados

## Utilização do Operador OR

OR requer alguma das condições ser TRUE.

```
SQL> SELECT empno, ename, job, sal
2 FROM emp
3 WHERE sal >= 1100
4 OR job = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7839	KING	PRESIDENT	5000
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7566	JONES	MANAGER	2975
7654	MARTIN	SALESMAN	1250
...			

14 rows selected.

1-45

UNISUL – Computação - Banco de Dados

## Utilização do Operador NOT

```
SQL> SELECT ename, job
2 FROM emp
3 WHERE job NOT IN ('CLERK', 'MANAGER', 'ANALYST');
```

ENAME	JOB
KING	PRESIDENT
MARTIN	SALESMAN
ALLEN	SALESMAN
TURNER	SALESMAN
WARD	SALESMAN

1-46

UNISUL – Computação - Banco de Dados

## Regras de Precedência

Ordem Avaliação	Operador
1	Todas os operadores de comparações
2	NOT
3	AND
4	OR

Para redefinir as regras de precedência faz-se a utilização de parênteses.

1-47

UNISUL – Computação - Banco de Dados

## Regras de Precedência

```
SQL> SELECT ename, job, sal
2 FROM emp
3 WHERE job='SALESMAN'
4 OR job='PRESIDENT'
5 AND sal>1500;
```

ENAME	JOB	SAL
KING	PRESIDENT	5000
MARTIN	SALESMAN	1250
ALLEN	SALESMAN	1600
TURNER	SALESMAN	1500
WARD	SALESMAN	1250

1-48

UNISUL – Computação - Banco de Dados



## Regras de Precedência

Uso de parênteses para forçar prioridades.

```
SQL> SELECT  ename, job, sal
2 FROM      emp
3 WHERE      (job='SALESMAN'
4 OR         job='PRESIDENT')
5 AND        sal>1500;
```

ENAME	JOB	SAL
KING	PRESIDENT	5000
ALLEN	SALESMAN	1600

1-49

UNISUL – Computação - Banco de Dados

## Cláusula ORDER BY

- Ordenar linhas com a cláusula ORDER BY
  - ASC: ordem ascendente, padrão
  - DESC: ordem descendente
- A cláusula ORDER BY é a última cláusula da sentença SELECT.

```
SQL> SELECT  ename, job, deptno, hiredate
2 FROM      emp
3 ORDER BY  hiredate;
```

ENAME	JOB	DEPTNO	HIREDATE
SMITH	CLERK	20	17-DEC-80
ALLEN	SALESMAN	30	20-FEB-81
...			

14 rows selected.

1-50

UNISUL – Computação - Banco de Dados

## Ordem Descendente

```
SQL> SELECT   ename, job, deptno, hiredate
2  FROM      emp
3  ORDER BY   hiredate DESC;
```

ENAME	JOB	DEPTNO	HIREDATE
ADAMS	CLERK	20	12-JAN-83
SCOTT	ANALYST	20	09-DEC-82
MILLER	CLERK	10	23-JAN-82
JAMES	CLERK	30	03-DEC-81
FORD	ANALYST	20	03-DEC-81
KING	PRESIDENT	10	17-NOV-81
MARTIN	SALESMAN	30	28-SEP-81
...			

14 rows selected.

1-51

UNISUL – Computação - Banco de Dados

## Ordenando por Alias de Colunas

```
SQL> SELECT   empno, ename, sal*12 annsal
2  FROM      emp
3  ORDER BY   annsal;
```

EMPNO	ENAME	ANNSAL
7369	SMITH	9600
7900	JAMES	11400
7876	ADAMS	13200
7654	MARTIN	15000
7521	WARD	15000
7934	MILLER	15600
7844	TURNER	18000
...		

14 rows selected.

1-52

UNISUL – Computação - Banco de Dados

## Ordenando por Múltiplas Colunas

- A ordem das colunas na cláusula ORDER BY define a ordem dos elementos.

```
SQL> SELECT      ename, deptno, sal
2 FROM          emp
3 ORDER BY deptno, sal DESC;
```

ENAME	DEPTNO	SAL
KING	10	5000
CLARK	10	2450
MILLER	10	1300
FORD	20	3000
...		

14 rows selected.

- You can sort by a column that is not in the SELECT list.

1-53

UNISUL – Computação - Banco de Dados

## Sentença Geral

```
SELECT          [DISTINCT] {*, column [alias], ...}
FROM            table
[WHERE          condition(s)]
[ORDER BY      {column, expr, alias} [ASC|DESC]];
```

1-54

UNISUL – Computação - Banco de Dados

# Exercícios

1-55

UNISUL – Computação - Banco de Dados