

ACH2025

Laboratório de Bases de Dados

Aula 5

SQL Oracle – Visão geral

Professora:

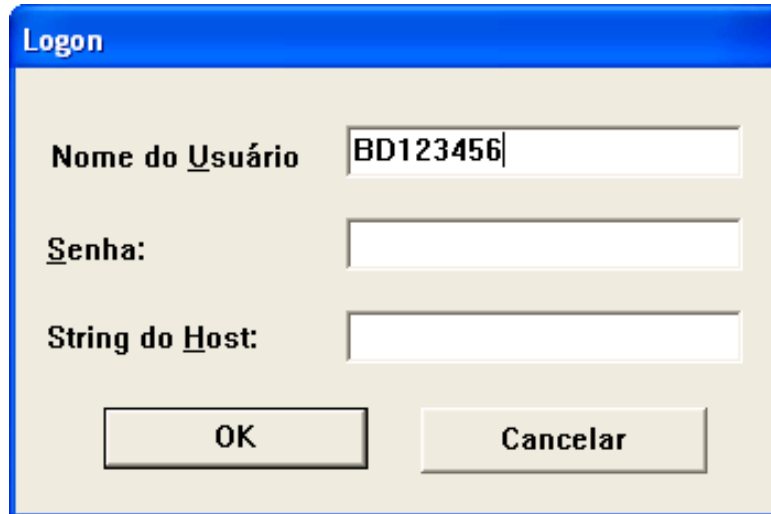
➤ **Fátima L. S. Nunes**



Características SQL

- ✓ Os comandos podem ser agrupados em 4 classes:
 - **DDL** - Comandos para a Definição de Relações
 - **DML** - Comandos para a Manipulação de Relações
 - **DCL** - Comandos para o Controle do Gerenciador, Conexão e Usuário
 - **SQL** - Comandos para recuperação de dados

- Aplicativo Oracle que permite executar todas as classes de comandos SQL.
- Permite comandos do SQL padrão e outros adicionais, próprios do SGBD Oracle.



Logon

Nome do Usuário

Senha:

String do Host:

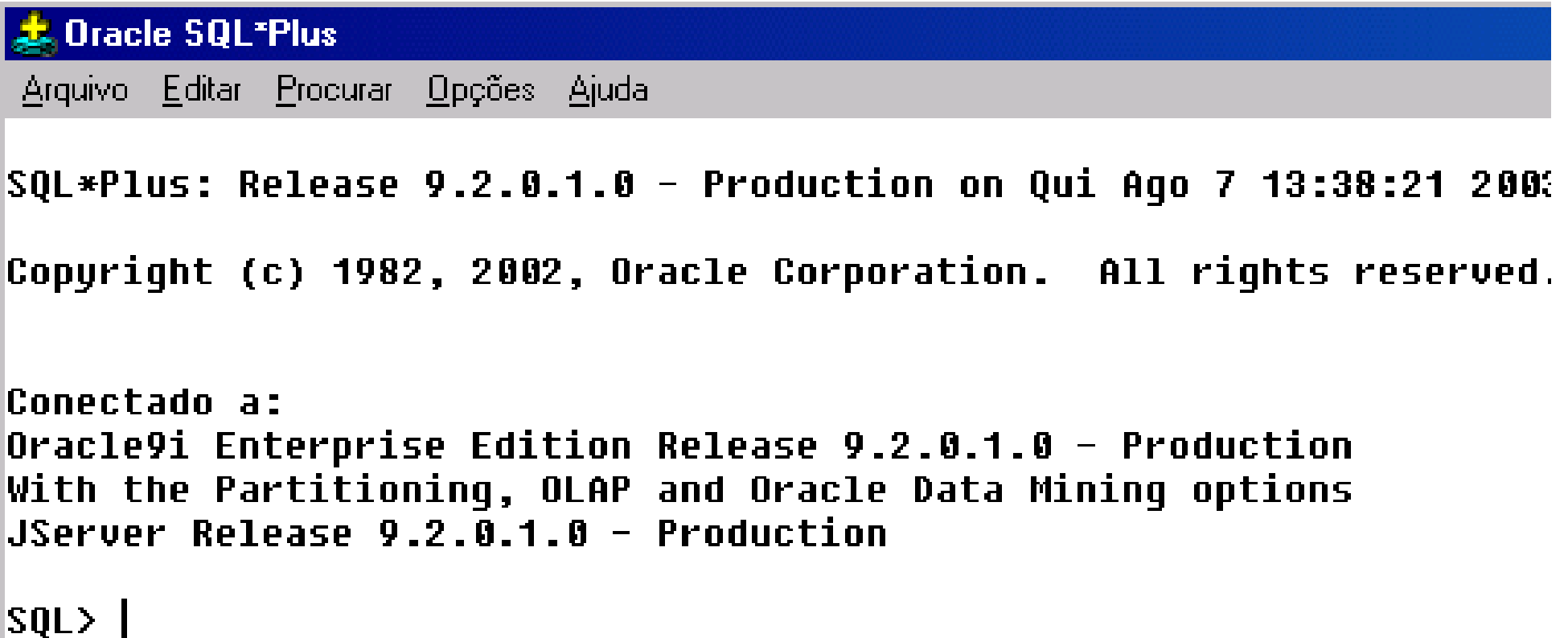
OK Cancelar

✓ Acesso (Oracle 8):

- Nome: scott
- Senha: tiger
- String do Host: oracle9i (definida na instalação)

SQLPlus

✓ Acesso:



The screenshot shows the Oracle SQL*Plus application window. The title bar reads "Oracle SQL*Plus". The menu bar includes "Arquivo", "Editar", "Procurar", "Opções", and "Ajuda". The main text area displays the following information:

```
SQL*Plus: Release 9.2.0.1.0 - Production on Qui Ago 7 13:38:21 2003  
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.  
  
Conectado a:  
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production  
With the Partitioning, OLAP and Oracle Data Mining options  
JServer Release 9.2.0.1.0 - Production  
  
SQL> |
```

SQLPlus

✓ Mudança de senha:

**alter user <nome-usuario>
identified by <nova-senha>;**

```
SQL> alter user fatima identified by senha;
```

Usuário alterado.



SQLPlus

- ✓ Comandos iniciais:
- ✓ Gravar o que está sendo feito:

```
SQL> spool nome-do-arquivo
```

SQLPlus

✓ Comandos iniciais:

✓ Listar todos os usuários:

```
SQL> select * from all_users;
```

USERNAME	USER_ID	CREATED
SYS	0	14/03/03
SYSTEM	5	14/03/03
OUTLN	11	14/03/03
DBSNMP	19	14/03/03
GD285171	354	07/08/03
GD285641	334	07/08/03
GD280720	335	07/08/03
GD282741	336	07/08/03



SQLPlus

✓ Observações

Digitação: contínua (uma única linha) ou ocupar mais de uma linha, teclando-se <enter> ao final de cada linha.

```
SQL> select * from all_users;
```

```
SQL> select  
2      *  
3      from  
4      all_users;
```

```
SQL> select * from all_users;
```

USERNAME	USER_ID	CREATED
SYS	0	14/03/03
SYSTEM	5	14/03/03
OUTLN	11	14/03/03
DBSNMP	19	14/03/03

✓ Funcionamento do SQLPlus



SQLPlus

- ✓ Todos os comandos digitados são colocados na **área de Buffer** para transmissão ao banco de dados.
- ✓ SQLPlus possui comandos próprios para:
 - Editar o comando SQL armazenado no Buffer;
 - Formatar os resultados retornados pelo BD;
 - Armazenar os comandos de SQL para o disco e recuperá-los para execução;
 - Modificar o modo de trabalhar do SQLPlus;
 - Enviar mensagens e receber respostas de outros usuários;
 - Listar a definição de qualquer tabela;
 - Fazer acesso e copiar dados entre banco de dados.



SQLPlus

✓ Comandos de Edição

- **LIST** - *Exibe uma ou mais linhas do buffer*

- **Sintaxe:**

L[IST] [n| n m| n*| n LAST| * | * n| * LAST| LAST]

```
SQL> select
      2   'teste'
      3   from
      4   dual;
```

```
'TEST
-----
teste
```



SQLPlus

✓ Comandos de Edição

– **LIST** - *Exibe uma ou mais linhas do buffer*

- Sintaxe:

L[IST] [n| n m| n*| n LAST| * | * n| * LAST| LAST]

```
SQL> 12                // 2a. linha
      2* 'teste'
SQL> 1 * 4              // linha atual até linha 4
      2 'teste'
      3 from
      4* dual
SQL> 1 last             // última linha
      4* dual
SQL> 1 1 *              // linha 1 até final
      1 select
      2 'teste'
      3 from
```



SQLPlus

✓ Comandos de Edição

- **APPEND** - *adiciona um trecho de texto ao final da linha corrente*

- Sintaxe:

A[PPEND] text

```
SQL> 1
      1  select
      2  'teste'
      3  from
      4*  dual
SQL> 12
      2*  'teste'
SQL> A, ' mais um teste'
      2*  'teste', ' mais um teste'
```



SQLPlus

✓ Comandos de Edição

- **APPEND** - *adiciona um trecho de texto ao final da linha corrente*

- **Sintaxe:**

A[PPEND] text

```
SQL> /  
  
'TEST' 'MAISUMTESTE'  
-----  
teste  mais um teste
```



SQLPlus

✓ Comandos de Edição

- **APPEND** - *adiciona um trecho de texto ao final da linha corrente*
 - Para incluir um branco (ou mais) entre o texto antigo e o novo: digitar o comando *Append*, dois brancos (ou mais) e o texto a ser adicionado.
 - Para incluir ponto-e-vírgula: incluir dois símbolos seguidos.

```
SQL> a ;;  
      2* 'teste', ' mais um teste', ;
```



SQLPlus

✓ Comandos de Edição

– **CHANGE** - *Altera texto na linha contida no buffer*

- **Sintaxe:**

C[CHANGE] sepchar old [sepchar [new [sepchar]]]

- Substitui parte do texto (ou todo) por outro.
- O separador **<sepchar>** pode ser qualquer caracter especial que não esteja presente no texto **<old>** nem no texto **<new>**.
- Uma substituição por linha: a linha é analisada da esquerda para a direita. Quando for encontrado um trecho de texto igual ao texto **<old>**, é feita a substituição pelo texto **<new>** e o comando é encerrado.
- Para repetir a substituição para outro trecho da linha: digitar o comando novamente.



SQLPlus

✓ Comandos de Edição

- **CHANGE** - *Altera texto na linha contida no buffer*

```
SQL> 12
      2* 'teste'
SQL> a, 'maisumteste'
      2* 'teste', 'maisumteste'
SQL> 12
      2* 'teste', 'maisumteste'
SQL> c/teste/tst
      2* 'tst', 'maisumteste'
SQL> /
      'TS  'MAISUMTEST
-----
tst maisumteste
```



SQLPlus

✓ Comandos de Edição

– **DEL** - *Elimina uma ou mais linhas no buffer*

• Sintaxe:

DEL [n| n m| n*| n LAST| * | * n| * LAST| LAST]

Onde:

n	Deleta linha n.
n m	Deleta linha n até m.
n *	Deleta linha n até a linha corrente.
n LAST	Deleta linha n até a última linha.
*	Deleta linha corrente.
* n	Deleta a corrente linha até linha n.
* LAST	Deleta todas as linhas.
LAST	Deleta a última linha.



SQLPlus

✓ Comandos de Edição

– **DEL** - *Elimina uma ou mais linhas no buffer*

• Sintaxe:

```
SQL> 1
      1  select
      2    'tst', 'maisumteste'
      3  from
      4*   dual
SQL> del 2
SQL> 1
      1  select
      2  from
      3*   dual
```



SQLPlus

✓ Comandos de Edição

- **INPUT** - *Adiciona uma ou mais linhas após a linha corrente no SQL buffer*

- **Sintaxe:**

I[INPUT] [text]

- caso não sejam digitados parâmetros, uma linha em branco será acrescentada.
- linhas inseridas apresentam um *i* após a numeração.



SQLPlus

✓ Comandos de Edição – INPUT

```
SQL> 1
      1 select
      2 from
      3* dual
SQL> 1
      1* select
SQL> i
      2i 'teste'
      3i , 'outro teste'
      4i
SQL> 1
      1 select
      2 'teste'
      3 , 'outro teste'
      4 from
      5* dual
SQL> /

' TEST  ' OUTROTESTE
-----
teste outro teste
```



SQLPlus

✓ Comandos de Edição

- **EDIT** - *aciona um editor registrado no Windows e passa como parâmetro o nome de um arquivo ou o texto presente no SQL Buffer, de acordo com o comando executado.*

- **Sintaxe:**

ED[IT] [file_name[.ext]]

- O editor em uso pode ser obtido através de uma variável chamada **_Editor**.
- Para saber seu valor:
 - **DEFINE _EDITOR** ou apenas **DEFINE**.



SQLPlus

✓ Comandos de Edição

– EDIT

```
SQL> DEFINE
DEFINE _CONNECT_IDENTIFIER = "fatima" (CHAR)
DEFINE _SQLPLUS_RELEASE = "902000100" (CHAR)
DEFINE _EDITOR = "Notepad" (CHAR)
DEFINE _O_VERSION = "Oracle9i Enterprise
Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data
Mining options
JServer Release 9.2.0.1.0 - Production" (CHAR)
DEFINE _O_RELEASE = "902000100" (CHAR)
```

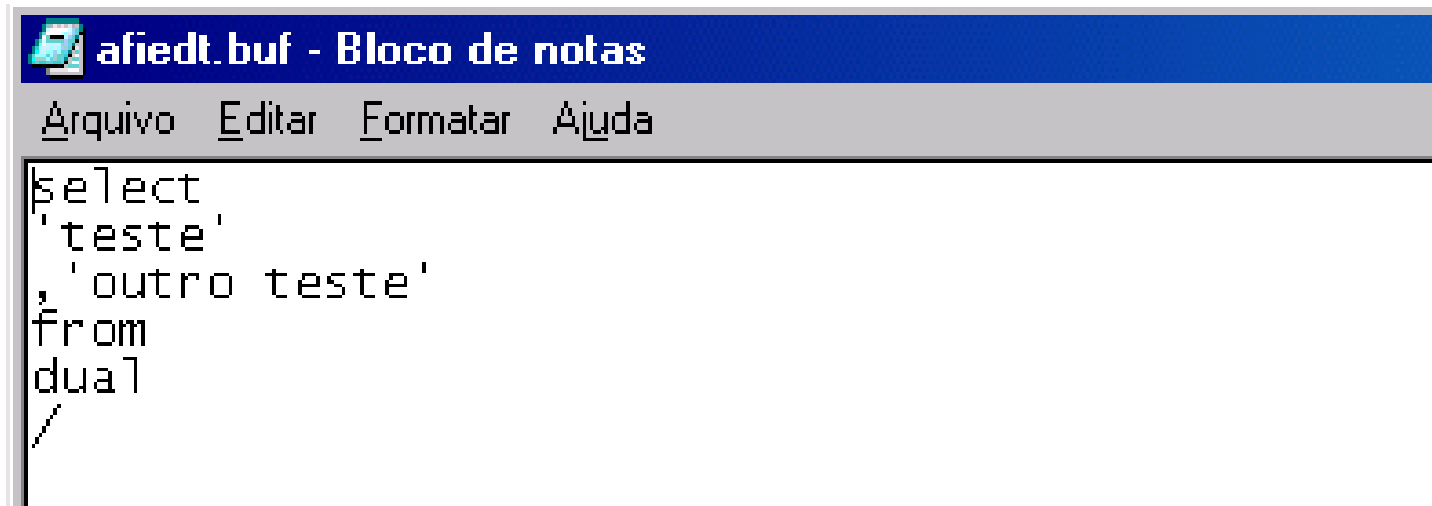


SQLPlus

✓ Comandos de Edição

– EDIT

```
SQL> ED  
Gravou arquivo afiedt.buf
```



The screenshot shows a Windows Notepad application window with the title bar 'afiedt.buf - Bloco de notas'. The menu bar includes 'Arquivo', 'Editar', 'Formatar', and 'Ajuda'. The text area contains the following SQL query:

```
select  
'teste'  
, 'outro teste'  
from  
dual  
/
```

SQLPlus

✓ Comandos de Edição

– EDIT

- a mensagem na linha imediatamente abaixo da linha de prompt indica que o SQL*Plus deveria passar como parâmetro para o editor o conteúdo do SQL Buffer.
- O que ele faz: grava um arquivo em disco (cujo nome default é Afiedt.buf) e passa o nome deste arquivo como parâmetro para o editor.
- Quando efetuamos as modificações desejadas, salvamos e fechamos o editor (Notepad), o SQL*Plus refaz a leitura do arquivo Afiedt.buf carregando o texto modificado para o SQL Buffer.



SQLPlus

✓ Comandos de Edição

– EDIT

- O editor apresenta o texto seguido de uma barra.
- Essa barra indica ao SQL*Plus que o comando está completo.
- Caso retiremos a barra do arquivo, ao retornarmos para o SQL*Plus, ele apresentará uma linha vazia imediatamente após a última linha lida para que continuemos a digitação.



SQLPlus

✓ Comandos de Edição

– EDIT

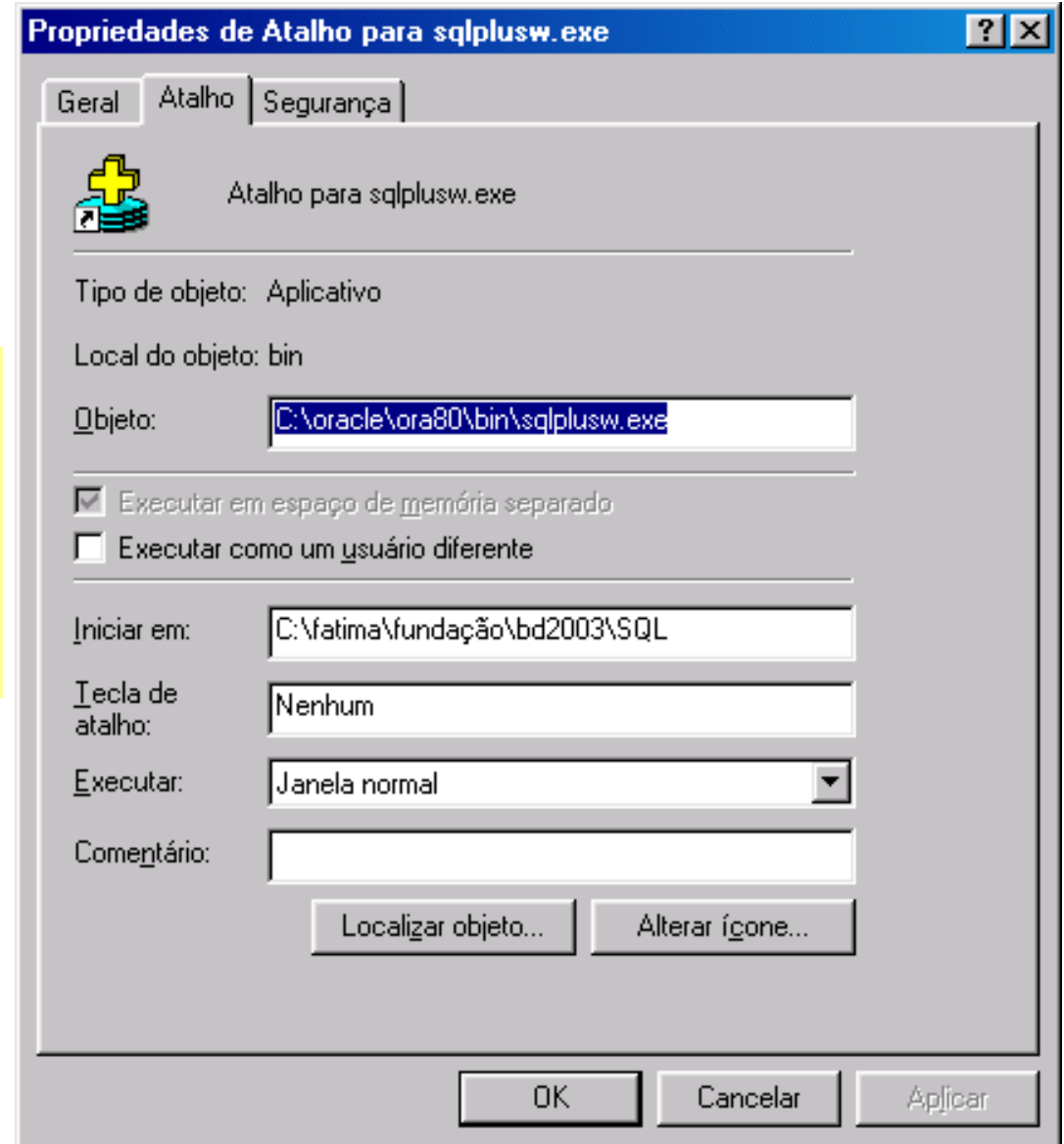
- O diretório do Windows onde serão gravados (ou lidos) os arquivos de trabalho pode ser modificado (default é **C:\<Oracle Home>\Bin**).
- Se criarmos um atalho para o SQL*Plus (o executável se encontra na pasta C:\<Oracle Home>\Bin e se chama SQLPlusw.exe), modificamos o diretório de trabalho.
- Basta que modifiquemos o diretório “Iniciar em” da pasta Atalho, obtido ao pressionarmos o botão direito do mouse sobre o ícone do SQL*Plus e escolhermos a opção Propriedade.
- Para que a modificação tenha efeito, o SQL*Plus deve ser reiniciado.



SQLPlus

✓ Comandos de Edição – EDIT

Qualquer arquivo a ser lido ou gravado pelo SQL*Plus será no novo diretório default.



SQLPlus

✓ Comandos de Edição

- **RUN** - *Envia o conteúdo do SQL buffer para o banco de dados e apresenta no vídeo as linhas enviadas (lista o SQL Buffer). Adiciona uma ou mais linhas após a linha corrente no SQL buffer*

- Sintaxe:
R[UN]

```
SQL> r
1  select
2  'teste'
3  from
4*  dual
```

```
'TEST'
-----
teste
```



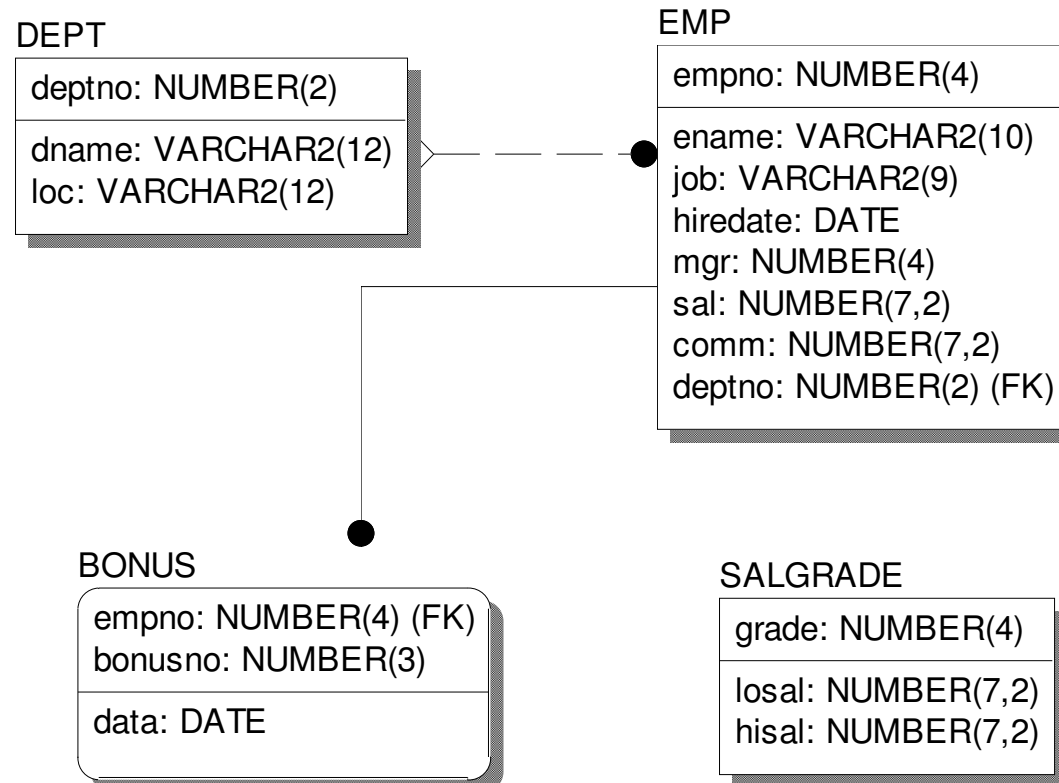
Configuração do SQLPlus

- **set time on** – liga o relógio no prompt
 - **set time on**
- **set sqlprompt <prompt>** - muda prompt
 - **set sqlprompt eusoulegal>**
- **set pagesize NN** – quantidade de linhas exibidas antes de repetir cabeçalho
 - **set pagesize 40**
- **set linesize NNN** – quantidade de caracteres exibidos por linha
 - **set linesize 100**



Criando tabelas

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



SQLPlus

✓ Criando tabelas

– CREATE TABLE

- **Sintaxe:**

CREATE TABLE tabela

(coluna1 type (size) [NULL / NOT NULL],

coluna2 type (size) [NULL / NOT NULL],

...)

SQLPlus

✓ Criando tabelas

– CREATE TABLE

```
SQL> CREATE TABLE DEPT (  
2          deptno          NUMBER(2) NOT NULL,  
3          dname          VARCHAR2(12) NULL,  
4          loc            VARCHAR2(12) NULL  
5  );
```

Tabela criada.

– ALTER TABLE - para definição de chave primária

```
ALTER TABLE DEPT  
ADD ( PRIMARY KEY (deptno) );
```



Criando tabelas

✓ Criando tabelas – definindo PK e FK na criação

```
CREATE TABLE EMP (  
    empno          NUMBER(4) NOT NULL,  
    job            VARCHAR2(20) NULL,  
    ename          VARCHAR2(40) NULL,  
    hiredate       DATE NULL,  
    sal            NUMBER(7,2) NULL,  
    comm           NUMBER(7,2) NULL,  
    deptno         NUMBER(2) NULL,  
    mgr            NUMBER(4) NULL,  
    PRIMARY KEY (empno),  
    FOREIGN KEY (deptno) REFERENCES DEPT );
```



SQLPlus

✓ Criando tabelas

✓ ALTER TABLE - para definição de chave estrangeira

```
SQL> CREATE TABLE EMP (  
  2      empno                NUMBER(4) NOT NULL,  
  3      job                  VARCHAR2(20) NULL,  
  4      ename                 VARCHAR2(40) NULL,  
  5      hiredate              DATE NULL,  
  6      sal                   NUMBER(7,2) NULL,  
  7      comm                  NUMBER(7,2) NULL,  
  8      deptno                NUMBER(2) NULL,  
  9      mgr                   NUMBER(4) NULL  
 10 );
```

Tabela criada.

```
SQL> ALTER TABLE EMP  
  2      ADD ( FOREIGN KEY (deptno)  
  3                               REFERENCES DEPT )  
  ;
```

Tabela alterada.



SQLPlus

✓ Verificando estruturas

- **DESCRIBE** - *apresenta a definição de um objeto criado na base de dados Oracle. O objeto pode ser uma tabela (table), visão (view), sinônimo (synonym) especificado ou ainda as especificações de um procedimento (procedure), função (function) ou pacote (package).*

- **Sintaxe:**

```
DESC[RIBE] {[ user.] table[@ database_link_name] [column] |  
[user.] object[. Subobject]};
```



SQLPlus

✓ Verificando estruturas

– DESCRIBE

```
SQL> desc emp;
```

Nome	Nulo?	Tipo
EMPNO	NOT NULL	NUMBER (4)
JOB		VARCHAR2 (20)
ENAME		VARCHAR2 (40)
HIREDATE		DATE
SAL		NUMBER (7, 2)
COMM		NUMBER (7, 2)
DEPTNO		NUMBER (2)
MGR		NUMBER (4)



SQLPlus

✓ Eliminando tabelas

– DROP TABLE

- **Sintaxe:**

DROP TABLE tabela;

```
SQL> drop table salgrade;
```

Tabela eliminada.

```
SQL> drop table emp;
```

Tabela eliminada.

```
SQL> drop table dept;
```

Tabela eliminada.



SQLPlus

✓ Executando arquivos de lote (script)

- Os comandos vistos podem ser armazenados em um arquivo texto - denominados *scripts*, podendo ser executados através do comando **@nome-arquivo**.

```
SQL> @apaga_empresa;
```

```
Tabela eliminada.
```

```
Tabela eliminada.
```

```
Tabela eliminada.
```



Criando tabelas

✓ Principais tipos de dados em Oracle

<i>Datatype</i>	<i>Descrição</i>
VARCHAR2(tam)	<i>String de caracteres de tamanho variável com um máximo de tam byte. Máximo permitido para tam: 2000.</i>
NUMBER(p,s)	<i>Número de precisão (tamanho) p, variando entre 1 e 38 e escala (decimais) s variando entre -84 e 127.</i>
LONG	<i>Caracteres de dados de tamanho variável até 2 gigabytes ou $2^{31} - 1$.</i>
DATE	<i>Datas válidas entre 1 de janeiro de 4712 AC e 31 de dezembro de 4712 DC.</i>
RAW(tam)	<i>Dados binários brutos (raw) de tamanho variável. Um tam deve ser especificado e o máximo permitido é 255.</i>
LONG RAW	<i>Dados binários brutos (raw) de tamanho variável até 2 gigabytes.</i>
ROWID	<i>String hexadecimal que representa o endereço único de uma linha em uma tabela.</i>
CHAR(tam)	<i>String de caracteres de tamanho tam fixo. O máximo é 255 e o default é 1.</i>

Exemplo de MER

Exemplo de utilização dos tipos de dados

CD

cd_codigo: NUMBER(4)
cd_nome: VARCHAR2(25)
cd_gravadora: VARCHAR2(15)
cd_data: DATE

MUSICA

cd_codigo: NUMBER(4) (FK)
mu_codigo: INTEGER
mu_titulo: VARCHAR2(25)
mu_tempo: NUMBER(4)
genero: VARCHAR2(15)
ca_código: DECIMAL(v) (FK)

CANTOR

ca_código: DECIMAL(v)
Nome: VARCHAR2(20)
ca_biografia: VARCHAR2(500)



DDL - Comandos para Definição de Relações

✓ Comandos para Definição de Dados

- CREATE INDEX
- DESCRIBE
- ALTER TABLE
- DROP



DDL - Comandos para Definição de Relações

✓ Criação de índices

- Auxiliar o gerenciador em pesquisa de um ou mais campos na base de dados, quando for solicitado;
- Auxiliar o gerenciador em ordenação de um ou mais campos na base de dados, quando for solicitado.
- Aumentam a performance de consultas que aum pequeno percentual de linhas;
- Índice são mantidos pelo Oracle mesmo que não sejam usados – manutenção demanda significativa quantidade de CPU e I/O;
- Regra geral – para consultas que resultam em 2 a 4% das linhas.



DDL - Comandos para Definição de Relações

✓ Criação de índices

– Oracle: índices BTree e Bitmap

- **Btree:** Mais usado em atualização online; árvores balanceadas – rápida localização de chaves; usam ROWID para construir árvore.
- **Bitmap:** tabelas muito grandes (milhões de linhas); tabelas com cardinalidade baixa (pequeno conjunto de valores distintos na coluna usada para indexação); milhares de linha são retornadas na consulta.
 - Mapa de bits para cada valor de chave: cada bit corresponde d um possível ROWID. Se bit estiver marcado: linha contém o valor da chave.
 - Oracle faz MERGE dos diferentes mapas de bits antes que e filtra linhas antes de qualquer acesso à tabela.



DDL - Comandos para Definição de Relações

✓ **CREATE INDEX** - Cria um índice para uma tabela existente.

✓ **Formato:**

```
CREATE [UNIQUE]  
INDEX indice ON tabela ( campo [, campo...]  
[ASC | DESC]);
```

DDL - Comandos para Definição de Relações

✓ Exemplos:

```
SQL> create index iempnome  
      2  on emp(ename);
```

Índice criado.

```
SQL> create index idhiredate  
      2  on emp(hiredate desc);
```

Índice criado.

DDL - Comandos para Definição de Relações

✓ Visualização da estrutura - DESCRIBE

✓ Formato:

DESCRIBE tabela;

✓ Exemplo:

```
SQL> describe dept
```

Nome	Nulo?	Tipo
-----	-----	-----
DEPTNO	NOT NULL	NUMBER (2)
DNAME		VARCHAR2 (12)
LOC		VARCHAR2 (12)



DDL - Comandos para Definição de Relações

✓ Alteração de uma tabela - ALTER TABLE

✓ **Formato:**

ALTER TABLE tabela

ADD (coluna type (size) [NULL/NOT NULL]);

ALTER TABLE tabela

MODIFY (coluna type (size) [NULL/NOT NULL]);

ALTER TABLE tabela

DROP (coluna);



DDL - Comandos para Definição de Relações

- ✓ Alteração de uma tabela - ALTER TABLE
- ✓ Exemplos:

```
SQL> alter table dept  
      2  add (chefe varchar2(15));
```

Tabela alterada.

```
SQL> describe dept;
```

Nome	Nulo?	Tipo
-----	-----	-----
DEPTNO	NOT NULL	NUMBER (2)
DNAME		VARCHAR2 (12)
LOC		VARCHAR2 (12)
CHEFE		VARCHAR2 (15)

DDL - Comandos para Definição de Relações

- ✓ Alteração de uma tabela - ALTER TABLE
- ✓ Exemplos:

```
SQL> alter table dept  
      2 modify (chefe varchar2(20));
```

Tabela alterada.

```
SQL> describe dept;
```

Nome	Nulo?	Tipo
-----	-----	-----
DEPTNO	NOT NULL	NUMBER (2)
DNAME		VARCHAR2 (12)
LOC		VARCHAR2 (12)
CHEFE		VARCHAR2 (20)

DDL - Comandos para Definição de Relações

- ✓ Alteração de uma tabela - ALTER TABLE
- ✓ Exemplos:

```
SQL> alter table dept  
      2 drop (chefe);
```

Tabela alterada.

```
SQL> descr dept;
```

Nome	Nulo?	Tipo
-----	-----	-----
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(12)
LOC		VARCHAR2(12)

DDL - Comandos para Definição de Relações

✓ Exclusão de tabelas e índices - DROP.

✓ **Formato:**

DROP {TABLE tabela | INDEX índice}

✓ Exemplos:

```
SQL> drop table musica;
```

Tabela eliminada.

```
SQL> drop index iempnome;
```

Índice eliminado.

DML - Comandos para a Manipulação de Dados

✓ Comandos DML:

- INSERT
- UPDATE
- DELETE

✓ COMMIT e ROLLBACK



DML - Comandos para a Manipulação de Dados

✓ **INSERT** - Adiciona linhas em uma tabela.

✓ **Formato:**

```
INSERT INTO tabela [ (coluna1, coluna2, ....)]  
VALUES (valor1, valor2, ....)
```

- Se não for especificada uma lista ► todas as colunas serão usadas
- Valores de colunas do tipo CHARACTER e DATE devem estar entre apóstrofos.



DML - Comandos para a Manipulação de Dados

✓ Exemplos:

```
SQL> INSERT INTO DEPT VALUES  
      2      (10, 'ACCOUNTING', 'NEW YORK');
```

1 linha criada.

```
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK



DML - Comandos para a Manipulação de Dados

✓ Exemplos:

```
SQL> INSERT INTO EMP VALUES  
2 (7839, 'PRESIDENT', 'KING', to_date('17-11-1981', 'dd-mm-yyyy'), 5000, NULL, 10, NULL);
```

1 linha criada.

```
SQL> select * from emp;
```

EMPNO	JOB	ENAME	HIREDATE	SAL	COMM	DEPTNO

	MGR					

7839	PRESIDENT	KING	17/11/81	5000		10

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

EMPNO	JOB	ENAME	HIREDATE	SAL	COMM	DEPTNO	MGR

7839	PRESIDENT	KING	17/11/81	5000		10	



DML - Comandos para a Manipulação de Dados

✓ Exemplos:

```
SQL> INSERT INTO EMP (EMPNO,ENAME)  
2 VALUES (4566,'JOAO');
```

1 linha criada.

```
SQL> select * from emp;
```

	EMPNO	JOB	ENAME	HIREDATE	SAL	COMM	DEPTNO
MGR							
	7839	PRESIDENT	KING	17/11/81	5000		10
	4566		JOAO				



DML - Comandos para a Manipulação de Dados

✓ Exemplos:

```
SQL> INSERT INTO SALGRADE VALUES  
(1, 700, 1200);
```

1 linha criada.

```
SQL> select * from salgrade;
```

GRADE	LOSAL	HISAL
1	700	1200



DML - Comandos para a Manipulação de Dados

✓ **UPDATE** - Altera valores em uma tabela.

✓ **Formato:**

UPDATE tabela [alias]

SET coluna [, coluna,] = {expressão, subquery}

[WHERE condição]

- **Cláusula WHERE** ► especifica quais linhas serão alteradas.
- Se for omitida, todas as linhas da tabela serão alteradas



DML - Comandos para a Manipulação de Dados

✓ Exemplos:

```
SQL> update emp
      2  set ename = 'MARK'
      3  where ename = 'KING';
```

1 linha atualizada.

```
SQL> select * from emp;
```

EMPNO	JOB	ENAME	HIREDATE	SAL	COMM	DEPTNO
MGR						
7839	PRESIDENT	MARK	17/11/81	5000		10



DML - Comandos para a Manipulação de Dados

✓ **DELETE** - Remove linhas de uma tabela.

✓ **Formato:**

DELETE FROM tabela
[WHERE condição]

- Cláusula WHERE ► especifica quais linhas serão excluídas.
- Se for omitida, **todas as linhas da tabela** serão excluídas.



DML - Comandos para a Manipulação de Dados

✓ Exemplos:

```
SQL> delete from emp  
      2  where ename = 'MARK';
```

1 linha deletada.

```
SQL> delete from emp  
      2  where empno = 7566;
```

1 linha deletada.



DML - Comandos para a Manipulação de Dados

✓ Comando COMMIT

- Transação: operação no BD que realiza uma série de alterações em uma ou mais tabelas.
- Pode ser do tipo DML (Data Manipulation Language) ou DDL (Data Definition Language).
- Transação DML pode ser encerrada por:
 - Comando COMMIT ou ROLLBACK
 - Saída do SQL
 - Comando DDL
 - Interrupção brusca do BD.



DML - Comandos para a Manipulação de Dados

✓ Comando COMMIT

- Uma transação não é atualizada no BD até receber um comando (COMMIT, por exemplo).
- Este comando provoca a atualização em todas as transações pendentes até a sua execução.
- Transação DDL:
 - Atualização imediata
 - Não precisam de COMMIT



DML - Comandos para a Manipulação de Dados

✓ Comando COMMIT

✓ Formato:

COMMIT [WORK]

- Cláusula WORK ► desnecessária ► mantida por questões de compatibilidade com o padrão ANSI.



DML - Comandos para a Manipulação de Dados

✓ Comando ROLLBACK

- Oposto do COMMIT

- **desfaz** alterações pendentes até a sua execução, retornando as tabelas ao estado original no início da transação.

✓ **Formato:**

ROLLBACK [WORK]



DML - Comandos para a Manipulação de Dados

✓ Comando ROLLBACK

- Oposto do COMMIT

- Reverte alterações pendentes até a sua execução

**Voltaremos ao assunto
TRANSAÇÕES mais adiante
no curso**

ROLLBACK [WORK]

ACH2025

Laboratório de Bases de Dados

Aula 5

SQL Oracle – Visão geral

Professora:

➤ **Fátima L. S. Nunes**

