

# Disciplina de Modelação de Dados em Engenharia

**Material de Apoio às Aulas  
Teóricas (Português/Ingles)**

Resp. Disciplina: **Luis Camarinha Matos**

**João Rosas**

**Yves Rybarczyk**

**Pedro Santana**

# Tópicos para hoje

- SQL (3ª parte)
  - “Set operations”, joins
- PL/SQL (2ª parte)
  - Triggers,
  - Transacções: commit, rollback
  - recursividade
- Aplicação em contextos práticos
  - Odbc/jdbc
- Exemplos
  - Bd faculdade
  - Projecto electrico

# Joins, Sets

- Por vezes é necessário combinar os resultados de vários “select”.
  - Join, inner join, full join, cross join
  - Union, intersect, minus

# Operações JOIN

- Considerar as seguintes tabelas r1 e r2:

```
drop table r1;
```

```
create table r1 (
```

```
  id1  numeric(3) primary key,
```

```
  nome1 varchar2(20)
```

```
);
```

```
insert into r1 values(1, 'valor 1.1');
```

```
insert into r1 values(2, 'valor 1.2');
```

```
insert into r1 values(3, 'valor 1.3');
```

```
insert into r1 values(4, 'valor 1.4');
```

```
drop table r2;
```

```
create table r2 (
```

```
  id2  numeric(3) primary key,
```

```
  nome2 varchar2(20)
```

```
);
```

```
insert into r2 values(1, 'valor 2.1');
```

```
insert into r2 values(2, 'valor 2.2');
```

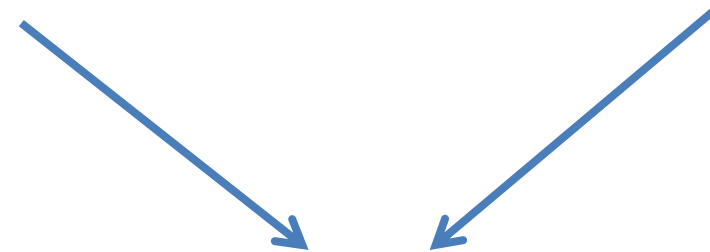
```
insert into r2 values(3, 'valor 2.3');
```

# Natural "Join"

```
select r1.nome1, r2.nome2  
from r1,r2  
where r1.id1 = r2.id2;
```

ID1	NOME1
1	valor 1.1
2	valor 1.2
3	valor 1.3
4	valor 1.4

ID2	NOME2
1	valor 2.1
2	valor 2.2
3	valor 2.3



NOME1	NOME2
1 valor 1.1	valor 2.1
2 valor 1.2	valor 2.2
3 valor 1.3	valor 2.3

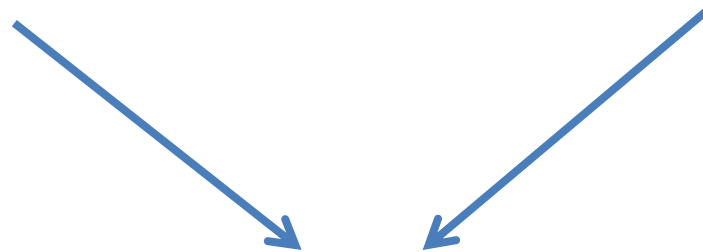
# Inner Join

```
select * from r1
```

```
inner join r2 on r1.id1=r2.id2;
```

ID1	NOME1
1	1 valor 1.1
2	2 valor 1.2
3	3 valor 1.3
4	4 valor 1.4

ID2	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3



ID1	NOME1	ID2	NOME2
1	1 valor 1.1	1	1 valor 2.1
2	2 valor 1.2	2	2 valor 2.2
3	3 valor 1.3	3	3 valor 2.3

# Full join

```
alter table r1 rename column id1 to id;  
alter table r2 rename column id2 to id;
```

```
select id , nome1, nome2  
from r1  
full join r2 using(id);
```

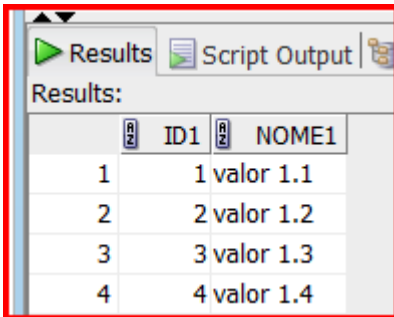
ID1	NOME1
1	1 valor 1.1
2	2 valor 1.2
3	3 valor 1.3
4	4 valor 1.4

ID2	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3

ID	NOME1	NOME2
1	1 valor 1.1	valor 2.1
2	2 valor 1.2	valor 2.2
3	3 valor 1.3	valor 2.3
4	4 valor 1.4	(null)

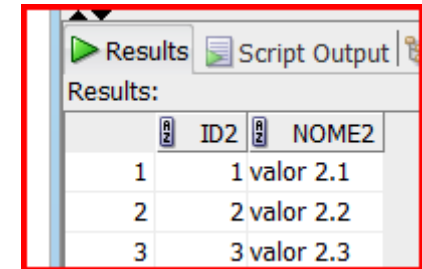
# Cross join

```
select * from r2  
cross join r1;
```



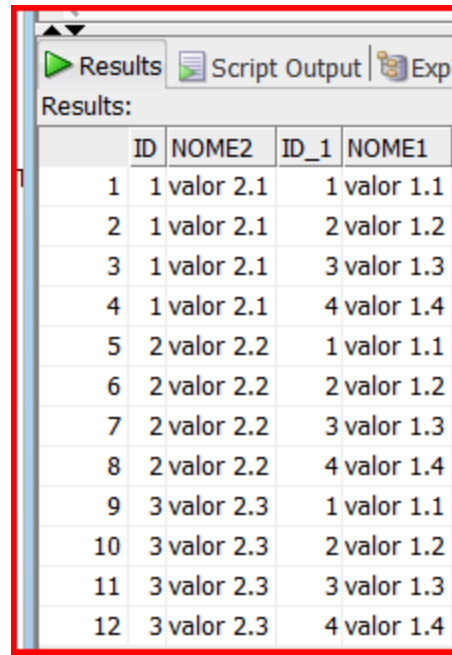
Results:

ID1	NOME1
1	1 valor 1.1
2	2 valor 1.2
3	3 valor 1.3
4	4 valor 1.4



Results:

ID2	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3



Results:

ID	NOME2	ID_1	NOME1
1	1 valor 2.1	1	1 valor 1.1
2	1 valor 2.1	2	2 valor 1.2
3	1 valor 2.1	3	3 valor 1.3
4	1 valor 2.1	4	4 valor 1.4
5	2 valor 2.2	1	1 valor 1.1
6	2 valor 2.2	2	2 valor 1.2
7	2 valor 2.2	3	3 valor 1.3
8	2 valor 2.2	4	4 valor 1.4
9	3 valor 2.3	1	1 valor 1.1
10	3 valor 2.3	2	2 valor 1.2
11	3 valor 2.3	3	3 valor 1.3
12	3 valor 2.3	4	4 valor 1.4



# Set operations: Union

```
select id1, nome1 from r1  
union  
select id2, nome2 from r2;
```

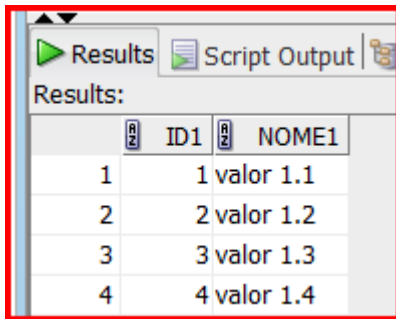
ID1	NOME1
1	1 valor 1.1
2	2 valor 1.2
3	3 valor 1.3
4	4 valor 1.4

ID2	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3

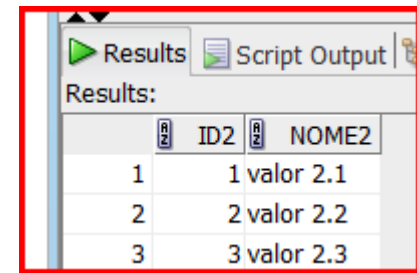
ID1	NOME1
1	1 valor 1.1
2	1 valor 2.1
3	2 valor 1.2
4	2 valor 2.2
5	3 valor 1.3
6	3 valor 2.3
7	4 valor 1.4

# Set operations: Intersection

```
select id1 from r1  
intersect  
select id2 from r2;
```

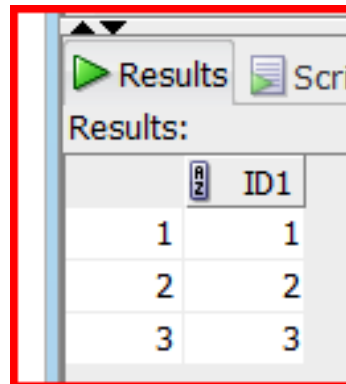


ID1	NOME1
1	1 valor 1.1
2	2 valor 1.2
3	3 valor 1.3
4	4 valor 1.4



ID2	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3

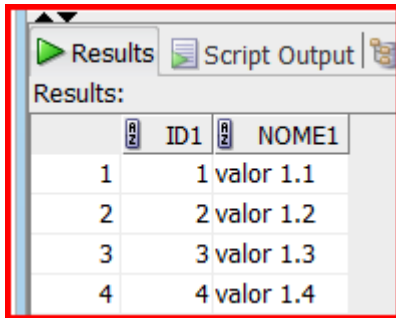
```
Select id1, nome1  
From r1 where id1 in  
    (select id1 from r1  
    intersect  
    select id2 from r2);
```



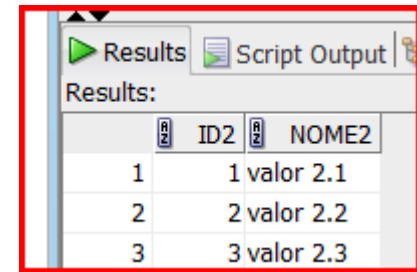
ID1
1
2
3

# Set operations: Minus

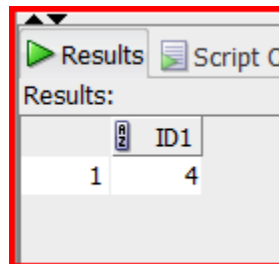
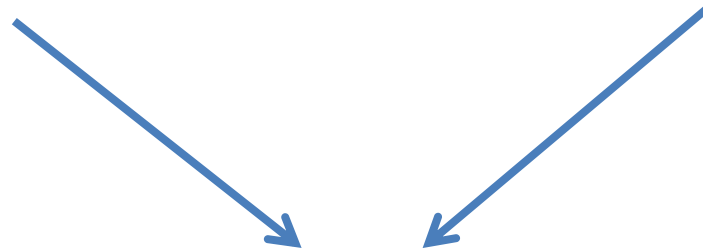
```
select id2 from r2  
minus  
select id1 from r1;
```



ID1	NOME1
1	1 valor 1.1
2	2 valor 1.2
3	3 valor 1.3
4	4 valor 1.4



ID2	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3



ID1
1

# Transactions

- **Ou faz tudo, ou não faz nada:**
- Begin transaction:
  - Insert into ...
  - While ...
    - Update ...
    - Delete ...
  - Update ...
- Commit transaction

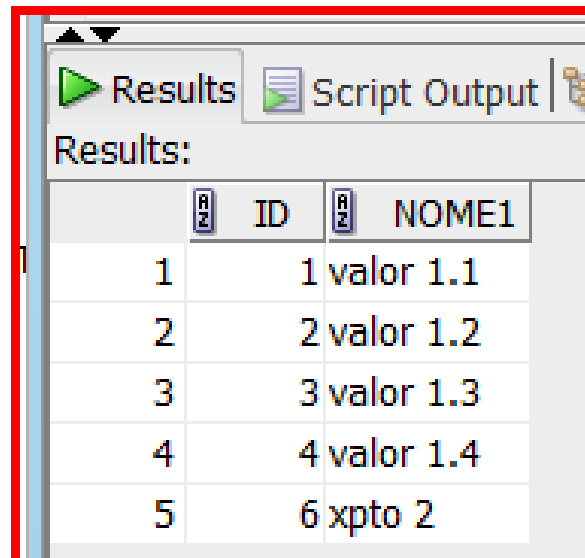
# Transactions: rollback

insert into R1 values (5, 'xpto 1');

rollback;

insert into R1 values (6, 'xpto 2');

commit;

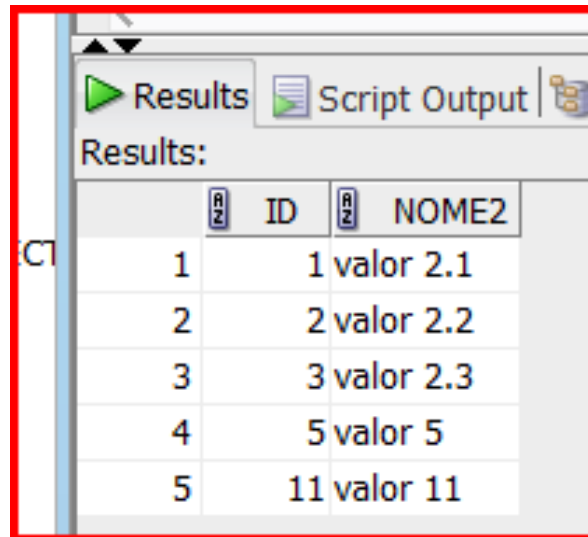


The screenshot shows a database query results window with two tabs: 'Results' (active) and 'Script Output'. The 'Results' tab displays a table with the following data:

	ID	NOME1
1	1	valor 1.1
2	2	valor 1.2
3	3	valor 1.3
4	4	valor 1.4
5	6	xpto 2

# Transactions: save point

```
insert into r2 values (5, 'valor 5');  
savepoint my_sp_1;  
insert into r2 values (7, 'valor 8');  
savepoint my_sp_2;  
insert into r2 values (9, 'valor 9');  
rollback to my_sp_1;  
insert into r2 values (11, 'valor 11');  
commit;
```

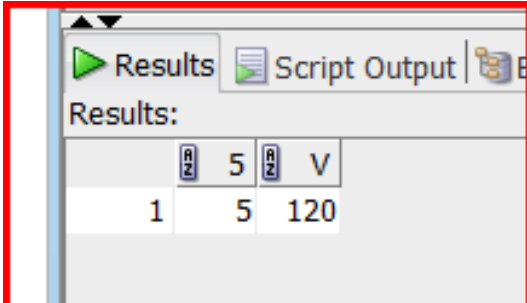


ID	NOME2
1	1 valor 2.1
2	2 valor 2.2
3	3 valor 2.3
4	5 valor 5
5	11 valor 11

# Recursividade

```
create or replace  
function factorial( NNN in number) return number is  
  auxiliar number;  
begin  
  if(NNN<=1) then return 1;  
  else  
    auxiliar := NNN * factorial(NNN-1);  
  end if;  
  return auxiliar;  
end;
```

```
select 5 as n, factorial(5) as ff from dual;
```

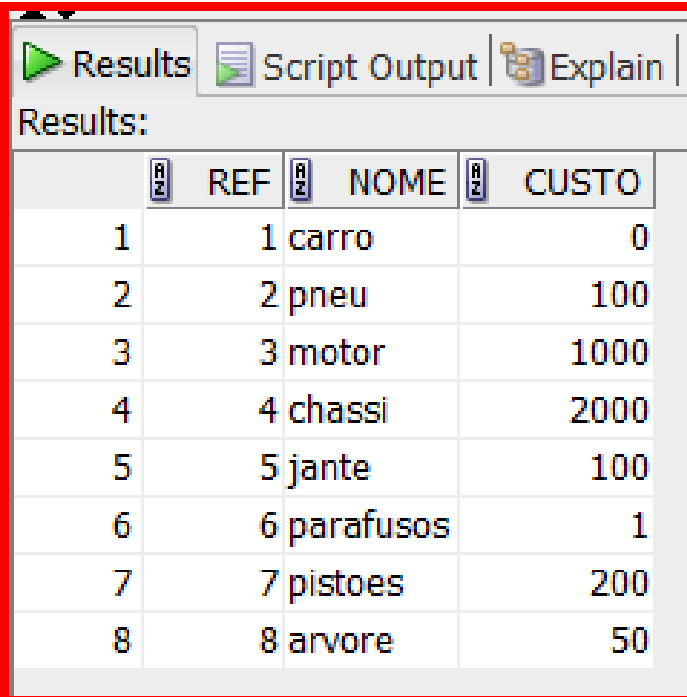


The screenshot shows a database query results window with a red border. The window has tabs for 'Results' (active), 'Script Output', and 'Error'. Below the tabs, the text 'Results:' is displayed. A table with three columns is shown. The first column has a value of 1, the second column has a value of 5, and the third column has a value of 120. The columns are labeled with 'R2', '5', 'R2', and 'V' above the values.

R2	5	R2	V
1	5		120

# Recursividade: exemplo (1)

```
create table artigo(  
  ref number primary key,  
  nome varchar(20),  
  custo numeric(10,2)  
);  
insert into artigo values(1, 'carro' , 0);  
insert into artigo values(2, 'pneu' , 100);  
insert into artigo values(3, 'motor' ,1000);  
insert into artigo values(4, 'chassi' ,2000);  
insert into artigo values(5, 'jante' , 100);  
insert into artigo values(6, 'parafusos', 1);  
insert into artigo values(7, 'pistoes' , 200);  
insert into artigo values(8, 'arvore' , 50);
```



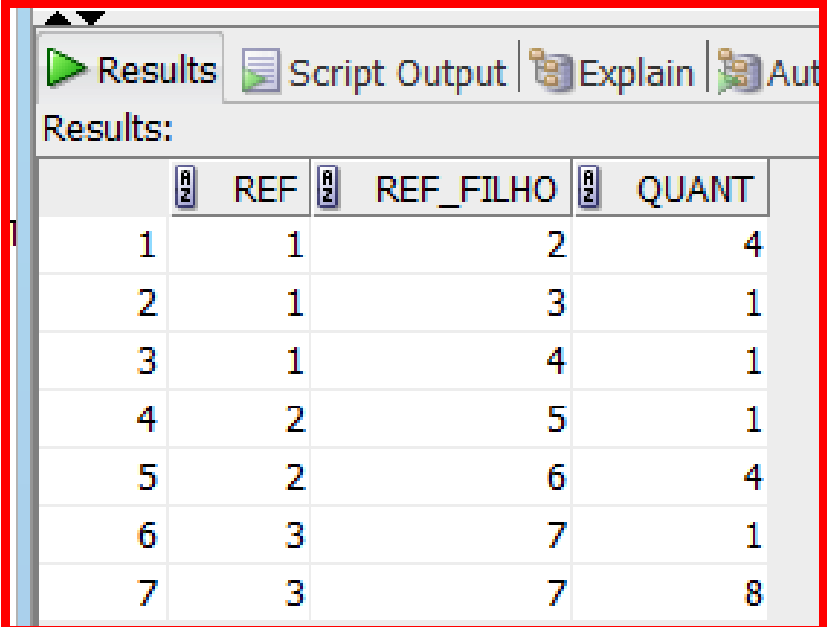
The screenshot shows a database query results window with a red border. The window has three tabs: 'Results' (active), 'Script Output', and 'Explain'. Below the tabs, the text 'Results:' is displayed. The results are shown in a table with the following columns: 'ID', 'REF', 'NOME', and 'CUSTO'. The data rows are as follows:

ID	REF	NOME	CUSTO
1	1	carro	0
2	2	pneu	100
3	3	motor	1000
4	4	chassi	2000
5	5	jante	100
6	6	parafusos	1
7	7	pistoes	200
8	8	arvore	50



# Recursividade: exemplo (2)

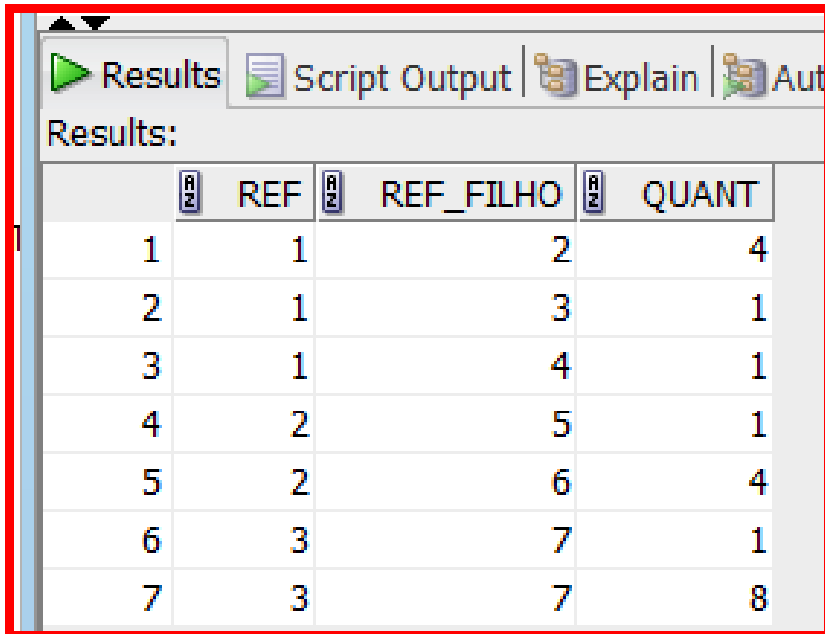
```
create table gasta(  
  ref      references artigo(ref),  
  ref_filho references artigo(ref),  
  quant   number  
);  
insert into gasta(1,2,4);  
insert into gasta(1,3,1);  
insert into gasta(1,4,1);  
insert into gasta(2,5,1);  
insert into gasta(2,6,4);  
insert into gasta(3,7,1);  
insert into gasta(3,7,8);
```



The screenshot shows a database query results window with a table containing 7 rows. The table has columns for ID, REF, REF\_FILHO, and QUANT. The data represents a recursive structure where each row's REF is the ID of its parent row, and its REF\_FILHO is the ID of its child row. The QUANT column shows the quantity for each row.

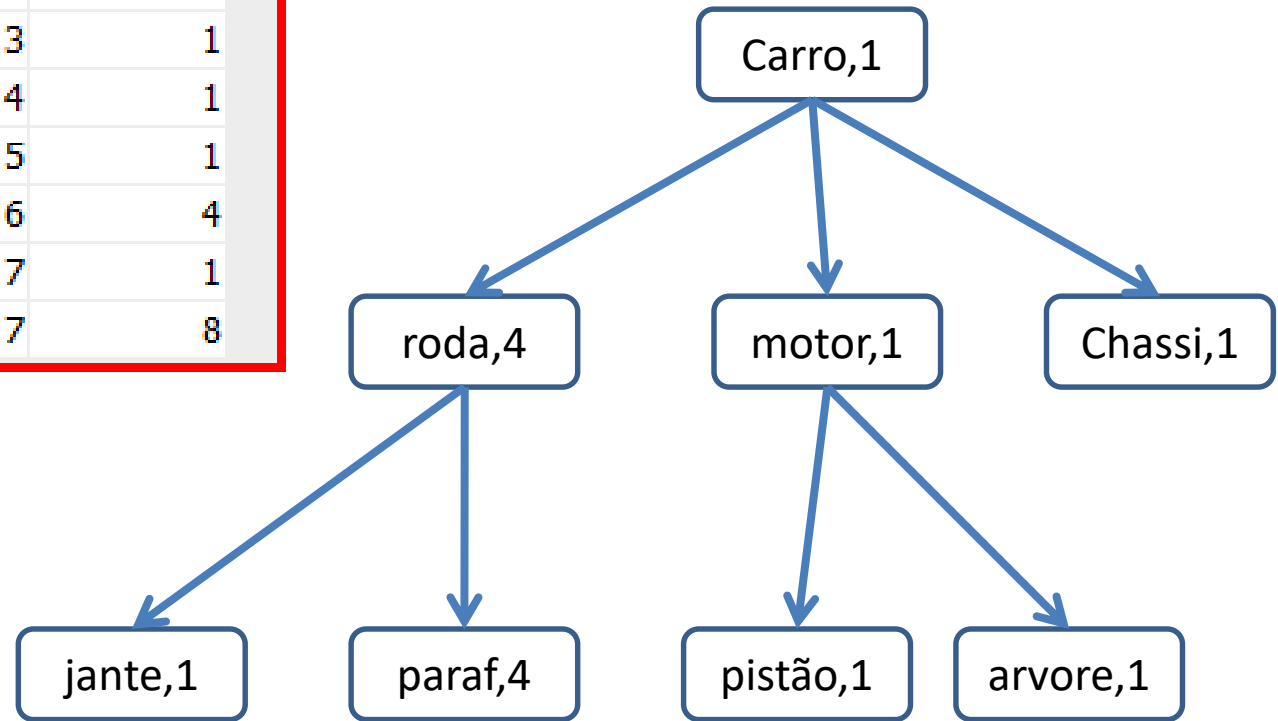
ID	REF	REF_FILHO	QUANT
1	1	2	4
2	1	3	1
3	1	4	1
4	2	5	1
5	2	6	4
6	3	7	1
7	3	7	8

# Recursividade: exemplo (3)



Results

	REF	REF_FILHO	QUANT
1	1	2	4
2	1	3	1
3	1	4	1
4	2	5	1
5	2	6	4
6	3	7	1
7	3	7	8



# Recursividade: exemplo (4)

```
create or replace
function custo(ref_pai in artigo.ref%type) return number is

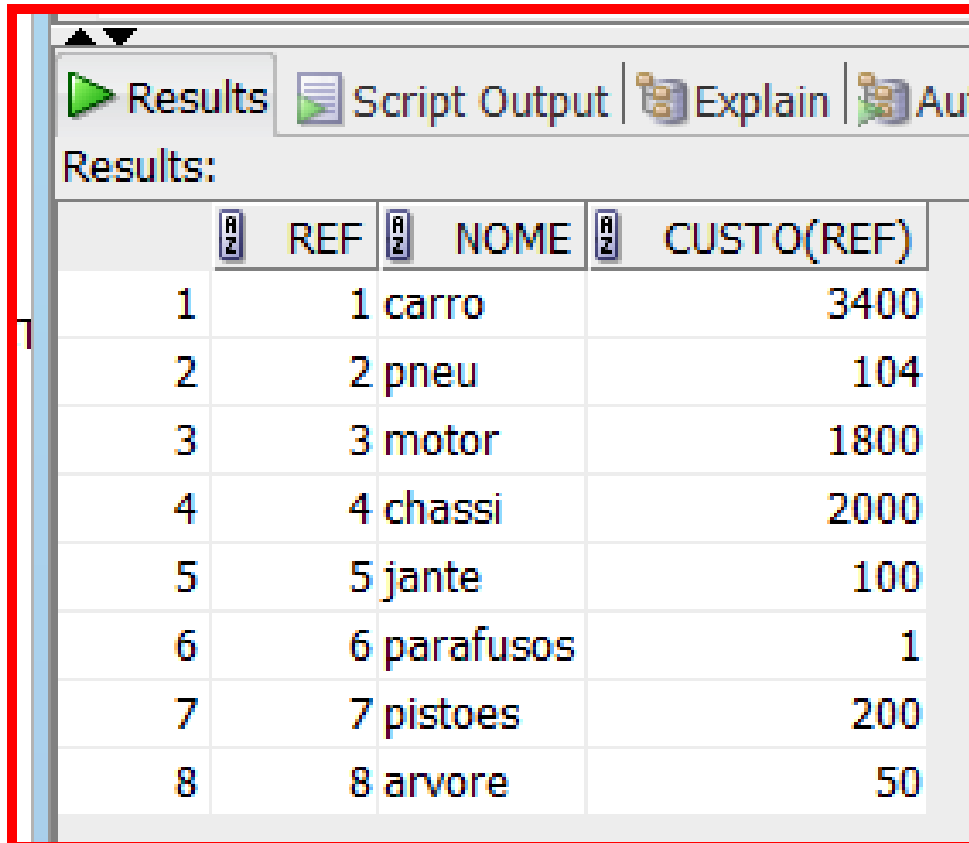
total NUMBER default 0;
ccc number default 0;
tem_filhos number default 0;

begin
select count(*) into tem_filhos from gasta where gasta.ref=ref_pai;

if(tem_filhos > 0) then
for g in (select * from gasta where gasta.ref= ref_pai)
loop
select artigo.custo into ccc from artigo where ref=g.ref_filho;
total := total + g.quant*ccc;
end loop;
else
select artigo.custo into total from artigo where artigo.ref = ref_pai;
end if;
return total;
end;
```

# Recursividade: exemplo (5)

```
select ref, nome, custo(ref) from artigo;
```



The screenshot shows a database query results window with a red border. The window has tabs for 'Results', 'Script Output', 'Explain', and 'Aut'. The 'Results' tab is active, displaying a table with 8 rows. The table has columns for 'REF', 'NOME', and 'CUSTO(REF)'. The data is as follows:

	REF	NOME	CUSTO(REF)
1	1	carro	3400
2	2	pneu	104
3	3	motor	1800
4	4	chassi	2000
5	5	jante	100
6	6	parafusos	1
7	7	pistoes	200
8	8	arvore	50

# Recursividade: exemplo (6)

Determinar custo sempre a partir dos filhos:

```
create or replace
function custo2(ref_pai in artigo.ref%type) return number is

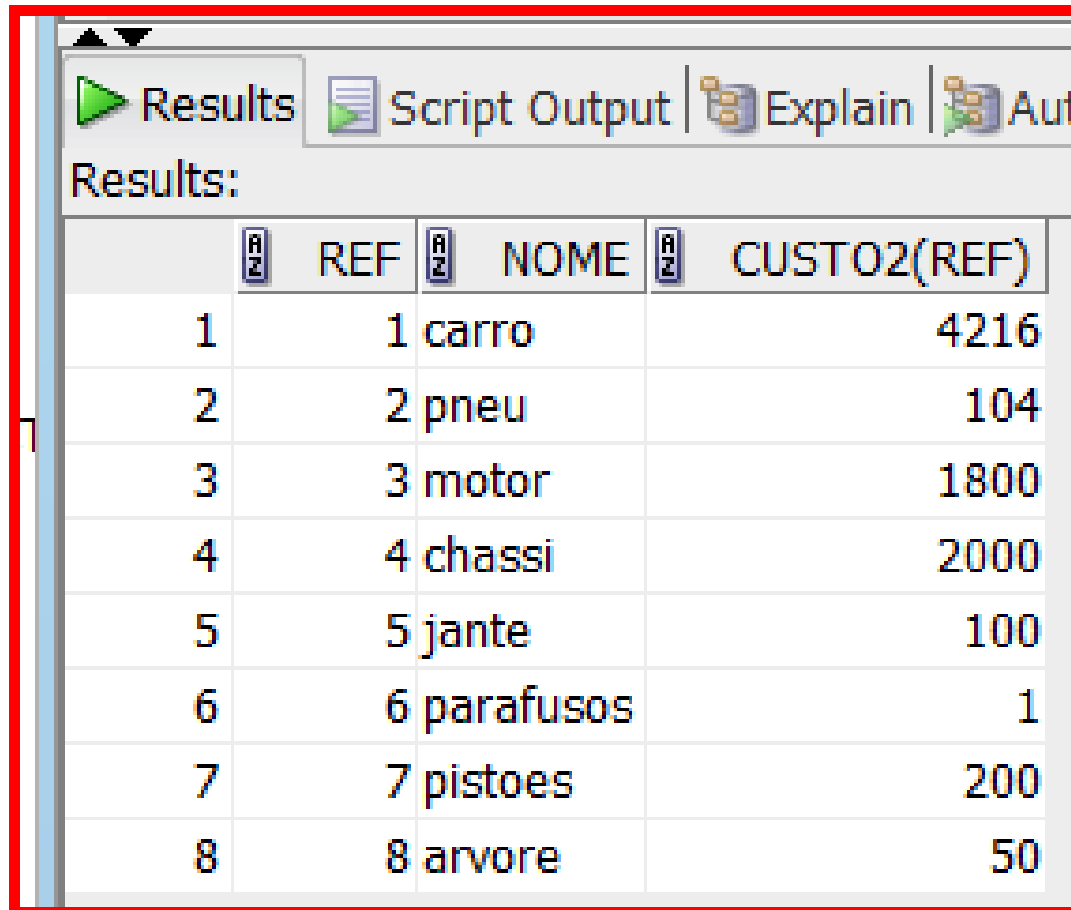
total NUMBER default 0;
ccc number default 0;
tem_filhos number default 0;

begin
select count(*) into tem_filhos from gasta where gasta.ref=ref_pai;

if(tem_filhos > 0) then
  for g in (select * from gasta where gasta.ref= ref_pai)
  loop
    total := total + g.quant*custo2(g.ref_filho);
  end loop;
else
  select artigo.custo into total from artigo where artigo.ref = ref_pai;
end if;
return total;
end;
```

# Recursividade: exemplo (7)

```
select ref, nome, custo2(ref) from artigo;
```



The screenshot shows a database query results window with a red border. The window has tabs for 'Results', 'Script Output', 'Explain', and 'Aut'. Below the tabs, the text 'Results:' is displayed. The results are presented in a table with 8 rows and 3 columns: 'REF', 'NOME', and 'CUSTO2(REF)'. Each column has a small 'A-Z' icon to its left. The data in the table is as follows:

	A-Z	REF	A-Z	NOME	A-Z	CUSTO2(REF)
1		1		carro		4216
2		2		pneu		104
3		3		motor		1800
4		4		chassi		2000
5		5		jante		100
6		6		parafusos		1
7		7		pistoes		200
8		8		arvore		50

# ODBC/JDBC

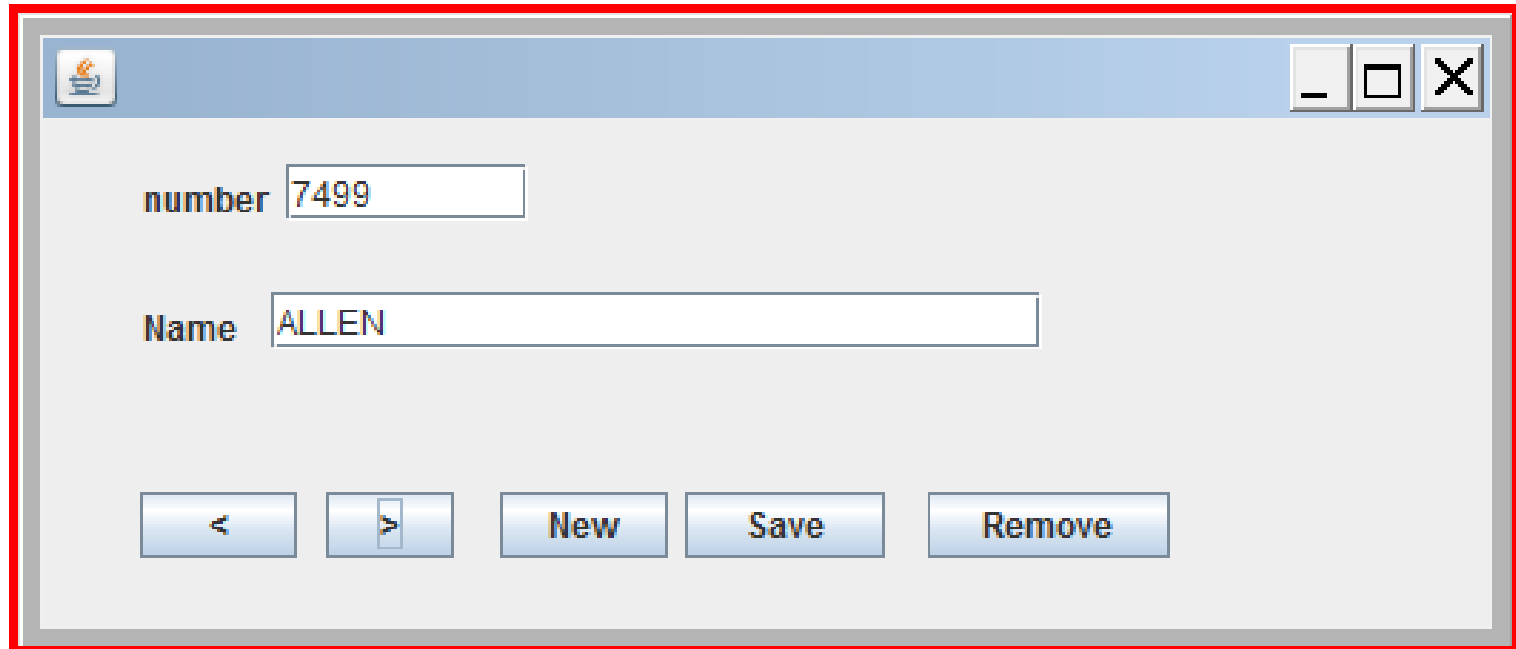
- ODBC (Open database connectivity)
  - C, C++, pascal...
- JDBC (Java Database Connectivity)
- Outros:
  - Oracle/asp.net
  - Oracle/php
  - ...

# ODBC/JDBC

- Usado para comunicar com um dbms
- A partir do programa C++, java
  - Abrir uma conexão com o dbms
  - enviar queries
  - Receber os resultados



# Um exemplo Java



A Java Swing window with a title bar. The window contains the following elements:

- A text field labeled "number" containing the value "7499".
- A text field labeled "Name" containing the value "ALLEN".
- A set of five buttons at the bottom: "<", ">", "New", "Save", and "Remove".

# Acesso Java

```
public class FrameDemo extends javax.swing.JFrame {
    public static String USERNAME = new String("scott");
    public static String PASSWORD = new String("tiger");
    Connection con = null;
    static {
        //IMPORTANTE
        try {
            // Load the Oracle Driver
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }catch(Exception e) {
            e.printStackTrace();
        }
    }
}
.../...
```

# Estabelecendo a conexão

```
/** Creates new form FrameDemo */
public FrameDemo() {
    initComponents();
    try {
        //IMPORTANTE
        con = DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:mde2011",USERNAME, PASSWORD);
    }catch(Exception e) {
        e.printStackTrace();
    }
    this.showFirstEmployee();
    this.setSize(500,200);
}
```

# Inserir um empregado

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButton4ActionPerformed
    try {
        //INSERIR UM EMPREGAGO
        PreparedStatement pstmt = null;
        String empno = this.jNumberTextField.getText();
        String name = this.jNameTextField.getText();
        pstmt = con.prepareStatement("insert into emp(empno,ename) values(?,?) ");
        pstmt.setString(1, empno);
        pstmt.setString(2, name);
        pstmt.executeUpdate();
        con.commit();
    } catch (SQLException e) {
        e.printStackTrace();
    }
} //GEN-LAST:event_jButton4ActionPerformed
```

# Apagar um empregado

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButton5ActionPerformed
    // TODO add your handling code here:
    try {
        //APAGAR UM EMPREGAGO
        PreparedStatement pstmt = null;
        String empno = this.jNumberTextField.getText();
        pstmt = con.prepareStatement("delete from emp where empno=?");
        pstmt.setString(1, empno);
        if(pstmt.executeUpdate()>=1) {
            this.jNameTextField.setText("Employee "+empno+" has been deleted...");
        }
        con.commit();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

# “Navegar” para o empregado anterior

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButton2ActionPerformed
    try {

        //OBTENHA EMPREGADO COM NUMERO IMEDIATAMENTE INFERIOR AO ATUAL
        Statement stmt = con.createStatement();
        String emp_number = this.jNumberTextField.getText();
        String sql = "select * from emp where empno in (select min(empno) from (select empno from emp where empno > " + emp_number + "))";
        ResultSet rs = stmt.executeQuery(sql);
        //displayResultSet(rs);
        if(rs.next()) {
            this.jNumberTextField.setText(rs.getString("EMPNO"));
            this.jNameTextField.setText(rs.getString("ENAME"));
        }
        rs.close();
        stmt.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
} //GEN-LAST:event_jButton2ActionPerformed
```

# Avec PHP

## Configuration (pour XAMPP) :

- Dans php.ini, il faut enlever le “;” devant :  
**extension=php\_oci8.dll**
- On peut vérifier (optionel) que la config est bien prise en compte à travers l’interface phpinfo() de XAMPP. Ce module doit apparaitre:



### oci8

OCI8 Support	enabled
Version	1.3.5
Revision	\$Revision: 289423 \$
Active Persistent Connections	0
Active Connections	0
Temporary Lob support	enabled
Collections support	enabled

Directive	Local Value	Master Value
oci8.connection_class	no value	no value
oci8.default_prefetch	100	100
oci8.events	Off	Off
oci8.max_persistent	-1	-1
oci8.old_oci_close_semantics	Off	Off
oci8.persistent_timeout	-1	-1
oci8.ping_interval	60	60
oci8.privileged_connect	Off	Off
oci8.statement_cache_size	20	20

# 'hello world' avec PHP

```
1 <?php
2 try
3 {
4     // On se connecte à Oracle (username, password)
5     $bdd = oci_connect('system', 'manager');
6
7     // On récupère tout le contenu du champs 'ename' la table scott.emp
8     $reponse = oci_parse($bdd, 'SELECT ename FROM scott.emp');
9     oci_execute($reponse);
10
11     // On affiche chaque entrée une à une
12     while ($donnees = oci_fetch_array($reponse))
13     {
14         ?>
15         <p>
16         <?php echo $donnees['ENAME']; ?> <!-- Attention de bien respecter la casse du titre du champs -->
17         </p>
18     <?php
19     }
20
21     oci_close($bdd);
22
23 }
24 catch(Exception $e)
25 {
26     // En cas d'erreur précédemment, on affiche un message et on arrête tout
27     die('Erreur de connexion à la BDD');
28 }
29
30
31 ?>
```



# Autre exemple avec PHP (formulaire)

- Le code CSS pour la présentation :

```
1 <html>
2   <style type="text/css">
3     .resultats
4     {
5       margin-top: 0px;
6     }
7     .edition
8     {
9       margin-top: -58px;
10      margin-left: 200px;
11    }
12  </style>
13
```

- Le code html pour le formulaire :

```
14 <body>
15 <form method="post" action="test_4.php">
16 <p>
17 <!-- un menu déroulant avec groupes d'options -->
18 <label for="donnees">Information recherchée</label><br />
19 <select name="donnees" onChange=this.form.submit()>
20 <option value="vide"></option><!-- pour faire apparaître une ligne blanche à chaque chargement de page -->
21 <optgroup label="Scott">
22 <option value="nom_emp">Noms d'employés</option>
23 <option value="job">Fonctions</option>
24 <option value="salaire">Salaires</option>
25 </optgroup>
26 <optgroup label="Books">
27 <option value="titre">Titres</option>
28 <option value="auteur">Auteurs</option>
29 </optgroup>
30 <optgroup label="Clients">
31 <option value="nom_client">Noms de clientes</option>
32 <option value="prenom">Prénoms</option>
33 </optgroup>
34 </select>
35 </p>
36 <p class="edition">
37 Ajouter le nom d'un employé : <input type="text" name="ajout" /><input type="submit" value="Ajouter" /><br />
38 Enlever le nom d'un employé : <input type="text" name="enlever" /><input type="submit" value="Enlever" />
39 </p>
40
41 <p class="resultats">
```

- Le code php pour le traitement (lire) :

```
43 <?php
44 // LIRE LES DONNEES
45 if($_POST['donnees'] != 'vide') {
46     // On se connecte d'abord à Oracle
47     $bdd = oci_connect('system', 'manager');
48
49     $info = $_POST['donnees'];
50
51     // On récupère les données
52     if($info == 'nom_emp') {
53         $reponse = oci_parse($bdd, 'SELECT ename FROM scott.emp');
54         echo 'Les employers sont :'; }
55     elseif ($info == 'job') {
56         $reponse = oci_parse($bdd, 'SELECT job FROM scott.emp');
57         echo 'Les fonctions sont :'; }
58     elseif ($info == 'salaire') {
59         $reponse = oci_parse($bdd, 'SELECT sal FROM scott.emp');
60         echo 'Les salaires sont :'; }
61     elseif ($info == 'titre') {
62         $reponse = oci_parse($bdd, 'SELECT title FROM books');
63         echo 'Les titres de livres sont :'; }
64     elseif ($info == 'auteur') {
65         $reponse = oci_parse($bdd, 'SELECT author_last_name FROM books');
66         echo 'Les auteurs sont :'; }
67     elseif ($info == 'nom_client') {
68         $reponse = oci_parse($bdd, 'SELECT last_name FROM customers');
69         echo 'Les noms de clients sont :'; }
70     elseif ($info == 'prenom') {
71         $reponse = oci_parse($bdd, 'SELECT first_name FROM customers');
72         echo 'Les prénoms de clients sont :'; }
73
74     oci_execute($reponse);
```

...la suite de la lecture :

```
76 // Puis on fait une boucle pour afficher tous les résultats
77 while ($donnees = oci_fetch_array($reponse))
78 {
79     ?>
80     <br />
81     <?php
82     echo $donnees['ENAME'];
83     echo $donnees['JOB'];
84     echo $donnees['SAL'];
85     echo $donnees['TITLE'];
86     echo $donnees['AUTHOR_LAST_NAME'];
87     echo $donnees['LAST_NAME'];
88     echo $donnees['FIRST_NAME'];
89 }
90
91 oci_close($bdd);
92 }
```

# ...le reste pour l'écriture :

```
94 // EDITER LA BD
95 if(isset($_POST['ajout']) OR isset($_POST['enlever'])) {
96     if($_POST['ajout'] != NULL OR $_POST['enlever'] != NULL) {
97         $bdd = oci_connect('system', 'manager');
98
99         if($_POST['ajout']) {
100             $ajout = $_POST['ajout'];
101             $editer = "INSERT INTO scott.emp (empno, ename) VALUES ((SELECT max(EMPNO)+1 as nextVal from scott.emp), '$ajout')";
102             $ordre = oci_parse($bdd, $editer);
103             echo 'Employé ajouté au registre.';
104         }
105
106         if($_POST['enlever']) {
107             $enlever = $_POST['enlever'];
108             $editer = "DELETE FROM scott.emp WHERE ename = '$enlever'";
109             $ordre = oci_parse($bdd, $editer);
110             echo 'Employé retiré du registre.';
111         }
112
113         oci_execute($ordre);
114         oci_commit($bdd);
115         oci_free_statement($ordre);
116         oci_close($bdd);
117     }
118 }
119 ?>
120 </p>
121 </form>
122 </body>
123 </html>
```

# Outro exemplo

- run SQLRunner

```
EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO
-----
1,aaaaa,null,null,null,null,null
7369,SMITH,CLERK,7902,1980-12-17 00:00:00,800,null,20
7499,ALLEN,SALESMAN,7698,1981-02-20 00:00:00,1600,300,30
7521,WARD,SALESMAN,7698,1981-02-22 00:00:00,1250,500,30
7566,JONES,MANAGER,7839,1981-04-02 00:00:00,2975,null,20
7654,MARTIN,SALESMAN,7698,1981-09-28 00:00:00,1250,1400,30
7698,BLAKE,MANAGER,7839,1981-05-01 00:00:00,2850,null,30
7782,CLARK,MANAGER,7839,1981-06-09 00:00:00,2450,null,10
7788,SCOTT,ANALYST,7566,1987-04-19 00:00:00,3000,null,20
7839,KING,PRESIDENT,null,1981-11-17 00:00:00,5000,null,10
7844,TURNER,SALESMAN,7698,1981-09-08 00:00:00,1500,0,30
7876,ADAMS,CLERK,7788,1987-05-23 00:00:00,1100,null,20
7900,JAMES,CLERK,7698,1981-12-03 00:00:00,950,null,30
7902,FORD,ANALYST,7566,1981-12-03 00:00:00,3000,null,20
7934,MILLER,CLERK,7782,1982-01-23 00:00:00,1300,null,10
```

# Exemplos

- Projectar uma BD faculdade
- Fazer uma BD para suportar o projecto eléctrico.

Search

[Advanced Search](#) • [Master Book List](#) • [Master Index](#) • [Master Glossary](#) • [Error Messages](#)

[Expand All](#) | [Collapse All](#) | [Help](#)

- [-] Installing and Upgrading
- [-] Getting Started
- [-] Database Administration
- [-] Application Development
- [-] Grid Computing
- [-] Performance
- [-] High Availability
- [-] Data Warehousing and Business Intelligence
- [-] Unstructured Data and Content Management
- [-] Information Integration
- [-] Security
- [-] Favorites

New and changed books:

[HTML documents](#) [PDF files](#)

Manage all aspects of your Oracle databases with the Enterprise Manager GUI.

2 Day + Data Replication and Integration Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + Data Warehousing Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + Performance Tuning Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + Real Application Clusters Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + Security Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day DBA	<a href="#">HTML</a>	<a href="#">PDF</a>

**New to Oracle Database 11g**  
Information you need for the latest release.

Master Glossary	<a href="#">HTML</a>
New Features Guide	<a href="#">HTML</a> <a href="#">PDF</a>
Readme	<a href="#">HTML</a> <a href="#">PDF</a>

**Supporting Documentation**  
Supporting documentation provides in-depth conceptual, task-based and reference material beyond the scope of the *2 Day DBA*, *2 Day Developer* and *2 Day + Guides*.

Administrator's Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
Concepts	<a href="#">HTML</a>	<a href="#">PDF</a>
Error Messages	<a href="#">HTML</a>	
Performance Tuning Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
Readme	<a href="#">HTML</a>	<a href="#">PDF</a>
Reference	<a href="#">HTML</a>	<a href="#">PDF</a>
SQL Language Reference	<a href="#">HTML</a>	<a href="#">PDF</a>

**Upgrade Information**  
If you are familiar with earlier Oracle releases and are moving to Oracle Database 11g, these books describe the new features, and explain how to upgrade your

Build applications around Oracle databases using the languages and platforms of your choice.

2 Day Developer's Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + .NET Developer's Guide for Microsoft Windows	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + Java Developer's Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + Application Express Developer's Guide	<a href="#">HTML</a>	<a href="#">PDF</a>
2 Day + PHP Developer's Guide	<a href="#">HTML</a>	<a href="#">PDF</a>

**Tips**

Each page in the library lists the essential books in a specific topic area first. The *2 Day* and *2 Day +* guides cover the most essential tasks and concepts in a specific topic area. The remainder of the page lists books that contain more detailed or advanced information.

**Online Resources**

[Oracle Technology Network](#)

- [Getting Started with Oracle Database](#)
- [Getting Started for Database Administrators](#)
- [Getting Started for Developers](#)



# Quanto custa?

## Oracle Database 11g: Administration Workshop II DBA Release 2

In this course, students learn how to perform backup and recovery, how to diagnose and repair data failures, how to manage major database components, such as memory, performance, resources and space using Oracle Database 11g. The lesson topics are reinforced with structured hands-on practices. This course covers the key features and enhancements of both Oracle Database 11g Release 1 and Release 2.



Oracle University provides a downloadable eKit containing the training materials for this course.



Oracle University eKit now viewable on the iPad

[Ver detalhes dos Cursos](#)

Formação em Sala - Calendário escondido		Live Virtual Class + Ver Calendário		Cursos Privados + Ver Detalhes						
Local	Ordenar	Data do Curso	Ordenar	Duração	Horário do Curso	Preço	Linguagem de apresentação	Materiais pedagógicos do Curso	Lugares	Adicione ao carrinho de compras
PT- <a href="#">Lisbon</a>		28-Mar-2011		5 Dias	09:00 AM-5:30 PM	€ 2000	European Portuguese	English	Disponível	
		16-Mai-2011		5 Dias	09:00 AM-5:30 PM	€ 2000	European Portuguese	English	Disponível	
		16-Mai-2011		5 Dias	09:00 AM-5:30 PM	€ 2000	European Portuguese	English	Disponível	

Can't find it, click [here](#)