

# Interfaces de base de données



# Objectifs

**A la fin de ce chapitre, vous pourrez :**

- **utiliser SQL\*Plus et *i*SQL\*Plus pour accéder à la base de données Oracle Database 10g**
- **décrire la structure logique des tables**
- **utiliser le langage SQL pour interroger, manipuler et définir des données**
- **identifier les interfaces de base de données courantes**

# Qu'est-ce que le langage SQL ?

**Le langage SQL offre des instructions pour un large éventail de tâches, telles que :**

- **Interroger les données**
- **Insérer, mettre à jour et supprimer des lignes dans une table**
- **Créer, remplacer, modifier et supprimer des objets**
- **Contrôler l'accès à la base de données et à ses objets**

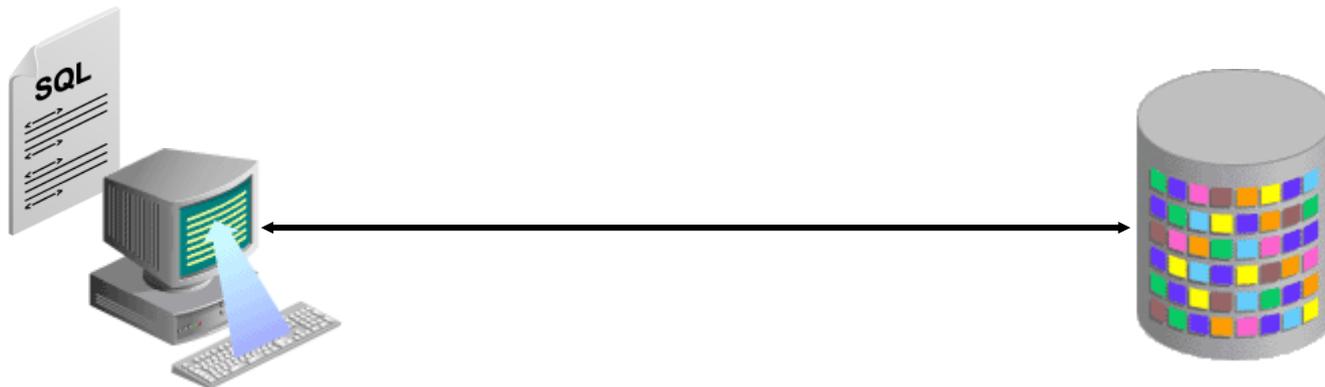
**Le langage SQL unifie toutes ces tâches dans un même langage cohérent.**



# Utiliser le langage SQL

Plusieurs outils permettent d'interagir avec la base de données via le langage SQL :

- Oracle SQL\*Plus et *iSQL\*Plus*
- Oracle Forms, Reports et Discoverer
- Oracle Enterprise Manager
- Outils tiers



# Enterprise Manager : afficher le code SQL

Create Tablespace

Show SQL Cancel OK

General Storage Thresholds

\* Name INVENTORY

Database: orcl.us.oracle.com > Tablespaces > Create Tablespace Logged in As SYS

SQL

Return

```
CREATE SMALLFILE TABLESPACE "INVENTORY" DATAFILE
'/u01/app/oracle/product/10.1.0/oradata/orcl/inv01.dbf' SIZE
100M LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO
BEGIN DBMS_SERVER_ALERT.SET_THRESHOLD
(9000, NULL, NULL, NULL, NULL, 1, 1, NULL, 5, 'INVENTORY'); END;
```

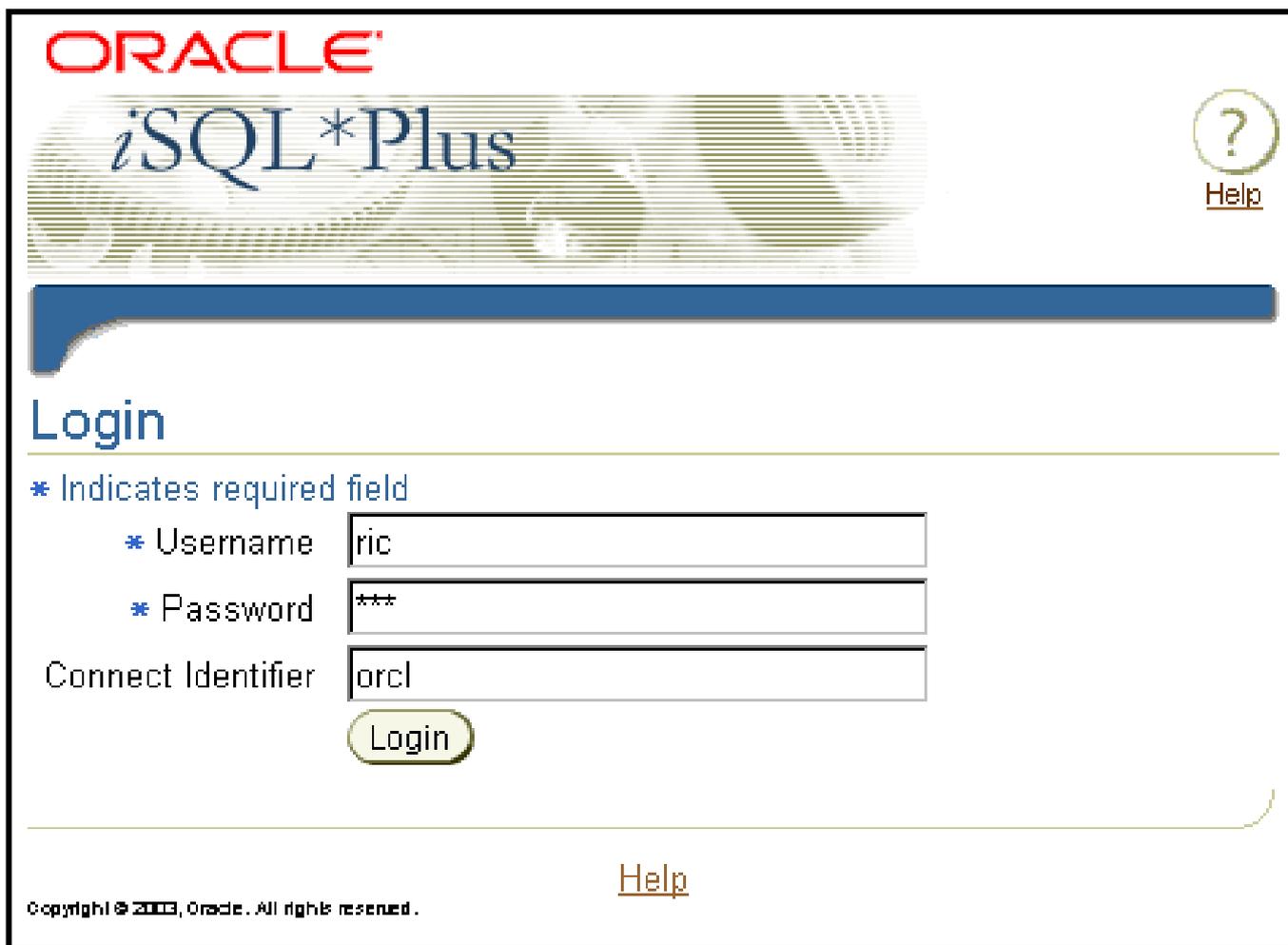
# Qu'est-ce que SQL\*Plus ?

- Outil en mode ligne de commande
- Utilisé en mode interactif ou en mode batch

```
$ sqlplus /nolog
SQL*Plus: Release 10.1.0.2.0 - Production on Tue Feb
17 06:17:14 2004
Copyright (c) 1982, 2004, Oracle. All rights
reserved.
SQL> connect ric
Enter password:
Connected.
SQL> SELECT * FROM dual;

D
-
X
SQL>
```

# Qu'est-ce que *iSQL\*Plus* ?



**ORACLE**  
*iSQL\*Plus*

[?](#)  
Help

## Login

\* Indicates required field

\* Username

\* Password

Connect Identifier

[Help](#)

Copyright © 2003, Oracle. All rights reserved.

# Utiliser iSQL\*Plus

ORACLE<sup>®</sup>  
iSQL\*Plus

Logout Preferences Help

Workspace History

Connected as RIC@orcl

Workspace

Enter SQL, PL/SQL and SQL\*Plus statements. Clear

```
describe dual;  
select * from dual;
```

Execute Load Script Save Script Cancel

Execute	Name	Null?	Type
	DUMMY		VARCHAR2(1)
		D	
	X		

# Décrire les données

**Workspace**

Enter SQL, PL/SQL and SQL\*Plus statements. Clear

```
desc hr.employees
```

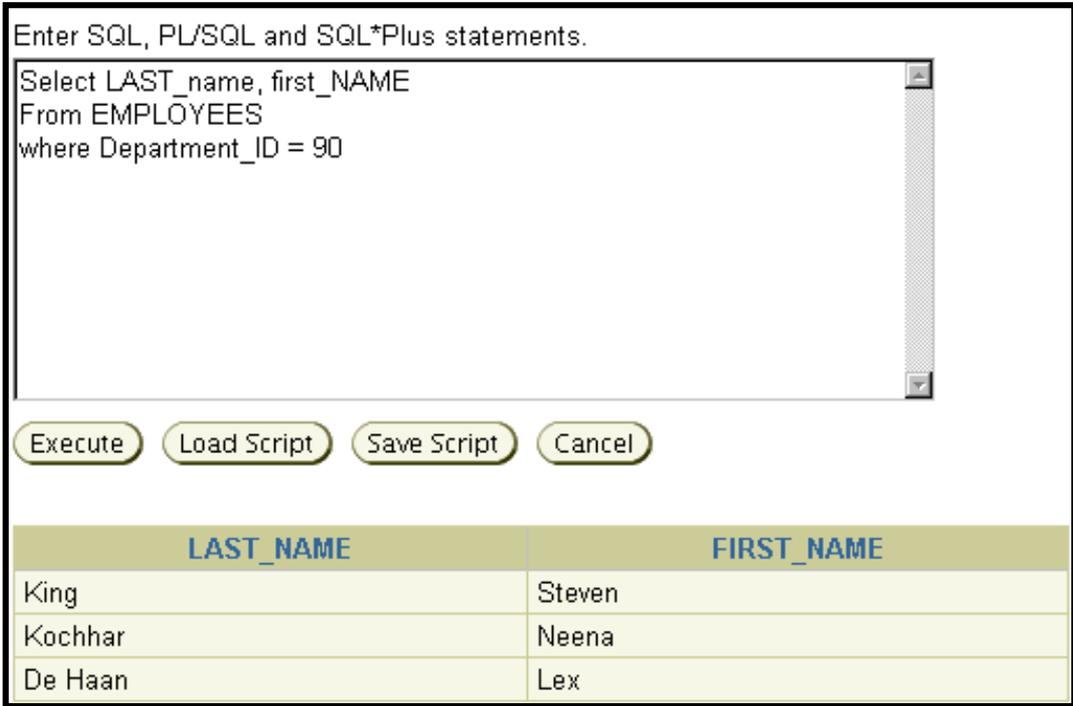
Execute Load Script Save Script Cancel

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

# Interroger les données

L'instruction **SELECT** comporte trois parties fondamentales :

- **La liste SELECT**
- **La clause FROM**
- **La condition WHERE (facultative)**



Enter SQL, PL/SQL and SQL\*Plus statements.

```
Select LAST_name, first_NAME  
From EMPLOYEES  
where Department_ID = 90
```

Execute Load Script Save Script Cancel

LAST_NAME	FIRST_NAME
King	Steven
Kochhar	Neena
De Haan	Lex

# Trier les données

```
SQL> SELECT last_name, department_id, phone_number  
2 FROM employees  
3 ORDER BY last_name;
```

LAST_NAME	DEPARTMENT_ID	PHONE_NUMBER
Abel	80	011.44.1644.429267
Ande	80	011.44.1346.629268
Atkinson	50	650.124.6234
Austin	60	590.423.4569
Baer	70	515.123.8888
Baida	30	515.127.4563
Banda	80	011.44.1346.729268

# Joindre des tables

## Obtenir des données de plusieurs tables

**Workspace**

Enter SQL, PL/SQL and SQL\*Plus statements. Clear

```
SELECT last_name, department_name  
FROM employees, departments  
WHERE employees.department_id = departments.department_id
```

Execute Load Script Save Script Cancel

LAST_NAME	DEPARTMENT_NAME
Whalen	Administration
Hartstein	Marketing
Fay	Marketing
Raphaely	Purchasing
Khao	Purchasing

# Manipuler les données

```
SQL> INSERT INTO employees
  2  (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER,
  3  HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT,
  4  MANAGER_ID, DEPARTMENT_ID)
  5  VALUES
  6  (9999, 'Bob', 'Builder', 'bob@abc.com', NULL, sysdate,
  7  'IT_PROG', NULL, NULL, 100, 90);
```

1 row created.

```
SQL> UPDATE employees SET SALARY=6000
  2  WHERE EMPLOYEE_ID = 9999;
```

1 row updated.

```
SQL> DELETE from employees
  2  WHERE EMPLOYEE_ID = 9999;
```

1 row deleted.

# Définir des données

Enter SQL, PL/SQL and SQL\*Plus statements.

```
CREATE TABLE local_temp  
(temp_date DATE,  
hi_temp NUMBER(4,1),  
lo_temp NUMBER(4,1));  
  
ALTER TABLE local_temp ADD (mean_temp NUMBER(4,1));  
  
DROP TABLE local_temp;
```

Execute Load Script Save Script Cancel

Table created.

Table altered.

Table dropped.

# Présentation des transactions



# Instructions de contrôle des transactions

```
SQL> SELECT * FROM local_temp;  
no rows selected
```

```
SQL> INSERT INTO local_temp VALUES  
  2  (SYSDATE, 76, 58);  
1 row created.
```

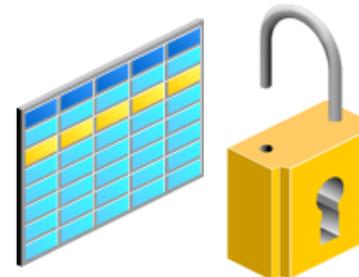
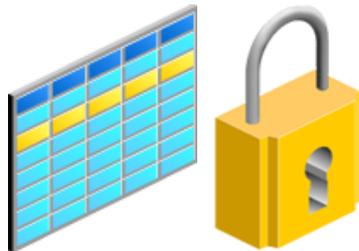
```
SQL> SELECT * from local_temp;  
TEMP_DATE      HI_TEMP      LO_TEMP  
-----  
27-OCT-03           76           58
```

```
SQL> ROLLBACK;  
Rollback complete.
```

```
SQL> SELECT * FROM local_temp;  
no rows selected
```

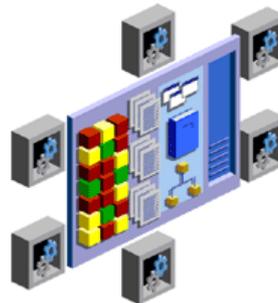
# Verrouiller les données

**Oracle Database 10g verrouille automatiquement les données, de sorte qu'un seul utilisateur à la fois puisse apporter des modifications.**



# Autres catégories d'instruction

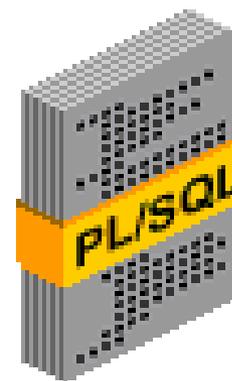
- **Instructions de contrôle de session : gèrent les propriétés d'une session utilisateur**
- **Instruction de contrôle système : gère les propriétés d'une instance Oracle**
- **Instructions SQL intégrées : instructions SQL dans un programme utilisant un langage procédural**



# Qu'est-ce que le langage PL/SQL ?

Le langage PL/SQL est un langage structuré à base de blocs, qui étend le langage SQL avec :

- **Des déclarations :**
  - Variables
  - Constantes
  - Curseurs
- **Des structures de contrôle :**
  - Contrôle conditionnel
  - Contrôle itératif
  - Contrôle séquentiel
- **Traitement des erreurs**



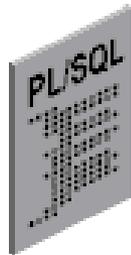
# Exemple de bloc PL/SQL

```
DECLARE
  qty_on_hand NUMBER(5);
BEGIN
  SELECT quantity INTO qty_on_hand FROM
    inventory
  WHERE product = 'TENNIS RACKET'
  FOR UPDATE OF quantity;
  IF qty_on_hand > 0 THEN -- check quantity
    UPDATE inventory SET quantity = quantity - 1
    WHERE product = 'TENNIS RACKET';
    INSERT INTO purchase_record
    VALUES ('Tennis racket purchased', SYSDATE);
  ELSE
    INSERT INTO purchase_record
    VALUES ('Out of tennis rackets', SYSDATE);
  END IF;
  COMMIT;
END;
```

# Utilisations du langage PL/SQL

**Les blocs de code PL/SQL sont utilisés dans :**

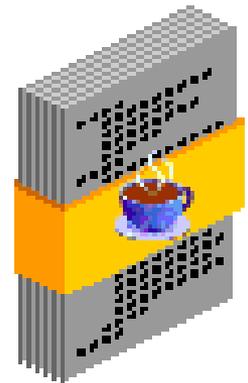
- **Les blocs anonymes**
- **Les fonctions**
- **Les procédures**
- **Les packages**
- **Les déclencheurs**
- **Les types d'objet**



# Qu'est-ce que Java ?

**Le langage Java est un langage de programmation standard orienté objet. Il inclut les concepts suivants :**

- **Une JVM (Java Virtual Machine), qui permet l'indépendance par rapport à la plate-forme**
- **Des techniques de gestion automatisée du stockage**
- **Une syntaxe issue du langage C et fiable**



# Oracle et le langage Java

## Une fonction PL/SQL :

```
FUNCTION balance (acct_id NUMBER) RETURN NUMBER IS
acct_bal NUMBER;
BEGIN
  SELECT bal INTO acct_bal FROM accts
    WHERE acct_no = acct_id;
  RETURN acct_bal;
END;
```

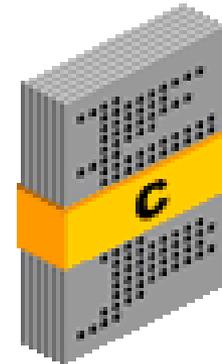
## Appeler la fonction en langage Java :

```
CallableStatement
cstmt = conn.prepareCall("{? = CALL balance(?) }");
cstmt.registerOutParameter(1, Types.FLOAT);
cstmt.setInt(2, acctNo);
cstmt.executeUpdate();
float acctBal = cstmt.getFloat(1);
```

# Qu'est-ce que OCI ?

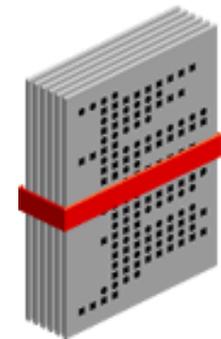
**Les caractéristiques d'OCI sont les suivantes :**

- **Oracle Call Interface (OCI) permet de mettre toutes les fonctionnalités de base de données à la disposition des développeurs d'applications.**
- **OCI permet la création d'applications évolutives et hautes performances.**
- **Les API et outils de plus haut niveau utilisent OCI indirectement pour l'accès à la base de données.**



# Autres API

- **JDBC (Java Database Connectivity)**
- **Pro\*C/C++**
- **Pro\*COBOL**
- **Oracle C++ Interface (OCCI)**
- **ODBC (Open Database Connectivity)**
- **ODP.NET (Oracle Data Provider for .NET)**
- **OO4O (Oracle Objects for OLE)**



# Synthèse

**Ce chapitre vous a permis d'apprendre à :**

- **utiliser SQL\*Plus et *i*SQL\*Plus pour accéder à la base de données Oracle Database 10g**
- **décrire la structure logique des tables**
- **utiliser le langage SQL pour interroger, manipuler et définir des données**
- **identifier les interfaces de base de données courantes**

# Exercice 4 : Utiliser le langage SQL

**Cet exercice traite de l'utilisation de *iSQL\*Plus* pour :**

- **décrire des tables**
- **sélectionner des données dans des tables**
- **mettre à jour une table**
- **supprimer des données d'une table**
- **annuler des modifications**