

www.dirasats.com



هذا الغلاف لا يعبر عن حقوق الملكية او فحوى الكتاب, فهو مجرد واجهة للموقع المحمل منه

شكرا لك على ثقتك بنا وعلى اختيار موقعنا

www.dirasats.com



من اجل تواصل معنا المرجو زيارة الموقع ستجد جميع المعلومات

www.dirasats.com

1

L'Ordre SELECT Elémentaire

Objectifs

A la fin de ce chapitre, vous saurez :

- **Enumérer toutes les possibilités de l'ordre SQL SELECT**
- **Exécuter un ordre SELECT élémentaire**
- **Faire la différence entre les ordres SQL et les commandes SQL*Plus**

1-2

Objectifs

Pour extraire des données d'une base de données, on utilise l'ordre SQL SELECT. Il est parfois nécessaire de restreindre le nombre de colonnes à l'affichage. Ce chapitre décrit tous les ordres SQL nécessaires à l'exécution de ces actions.

Il peut être utile de créer des ordres SELECT destinés à être réutilisés de nombreuses fois. Ce chapitre explique également comment utiliser les commandes SQL*Plus pour exécuter les ordres SQL.

Les Possibilités de l'Ordre SQL SELECT

Sélection

Table 1

Projection

Table 1

Jointure

Table 1



Table 2

1-3

Possibilités de l'ordre SQL SELECT

Un ordre *SELECT* permet d'extraire des informations d'une base de données. L'utilisation d'un ordre *SELECT* offre les possibilités suivantes :

- *Sélection* : SQL permet de choisir dans une table, les lignes que l'on souhaite ramener au moyen d'une requête. Divers critères de sélection sont disponibles à cet effet.
- *Projection* : SQL permet de choisir dans une table, les colonnes que l'on souhaite ramener au moyen d'une requête. Vous pouvez déterminer autant de colonnes que vous le souhaitez.
- *Jointure* : SQL permet de joindre des données stockées dans différentes tables, en créant un lien par le biais d'une colonne commune à chacune des tables. Les jointures seront décrites en détail dans la suite de ce cours.

Ordre SELECT Élémentaire

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

- **SELECT** indique *quelles* colonnes rapporter
- **FROM** indique dans *quelle* table rechercher

1-4

Ordre SELECT Élémentaire

Dans sa forme la plus simple, l'ordre SELECT comprend :

- Une clause SELECT précisant les colonnes à afficher
- Une clause FROM spécifiant la table qui contient les colonnes indiquées dans la clause SELECT

Syntaxe :

SELECT	liste d'une ou plusieurs colonnes
DISTINCT	suppression des doublons
*	sélection de toutes les colonnes
<i>column</i>	sélection de la colonne désignée
<i>alias</i>	attribue des en-têtes différents aux colonnes sélectionnées
FROM <i>table</i>	spécifie la table qui contient les colonnes

Remarque : nous utiliserons les termes mot-clé, clause et ordre tout au long de ce cours.

- Un *mot-clé* est un élément SQL individuel.
Par exemple, SELECT et FROM sont des mots-clés.
- Une *clause* est une partie d'un ordre SQL.
Par exemple, SELECT empno, ename, ... est une clause.
- Un *ordre* est la combinaison de deux clauses ou plus.
Par exemple, SELECT * FROM emp est un ordre SQL.

Ecriture des Ordres SQL

- **Les ordres SQL peuvent être écrits indifféremment en majuscules et/ou minuscules.**
- **Les ordres SQL peuvent être écrits sur plusieurs lignes.**
- **Les mots-clés ne doivent pas être abrégés ni scindés sur deux lignes différentes.**
- **Les clauses sont généralement placées sur des lignes distinctes.**
- **Les tabulations et indentations permettent une meilleure lisibilité.**

1-5

Ecriture des Ordres SQL

En suivant les règles et indications simples ci-dessous, vous pourrez créer des ordres corrects, simples à lire et à éditer.

- Sauf indication contraire, les ordres SQL peuvent être écrits indifféremment en majuscules ou en minuscules.
- Les ordres SQL peuvent être saisis sur plusieurs lignes.
- Les mots-clés ne doivent pas être scindés sur deux lignes différentes, ni abrégés.
- Les clauses se placent généralement sur des lignes distinctes pour en faciliter la lecture et l'édition.
- L'utilisation de tabulations et d'indentations permet une meilleure lisibilité.
- Généralement, les mots-clés sont saisis en majuscules, et tous les autres termes, tels que les noms de tables et de colonnes, sont saisis en minuscules.
- Dans le SQL*Plus, un ordre SQL est saisi au prompt SQL, et les lignes qui suivent sont numérotées. C'est ce qu'on appelle le *buffer SQL*. Il ne peut y avoir qu'un seul ordre courant à la fois dans le *buffer*.

Exécution d'Ordres SQL

- Placer un point-virgule (;) à la fin de la dernière clause.
- Placer un slash (/) seul sur la dernière ligne du *buffer*.
- Placer un slash (/) au prompt SQL.
- Entrer la commande RUN SQL*Plus au prompt SQL.

Sélection de Toutes les Colonnes

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

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

1-6

Sélection de Toutes les Colonnes et de Toutes les Lignes

Pour afficher *toutes* les colonnes d'une table, placez un astérisque à la suite du mot-clé SELECT (*). Dans l'exemple de la diapositive, la table des départements (DEPT) comporte trois colonnes : DEPTNO, DNAME et LOC. La table comporte également quatre lignes, une pour chaque département.

Vous pouvez aussi afficher *toutes* les colonnes de la table en les énumérant toutes à la suite du mot-clé SELECT. Par exemple, l'ordre SQL suivant affiche toutes les colonnes et toutes les lignes de la table DEPT :

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

Sélection d'Une ou Plusieurs Colonnes Spécifiques

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

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

1-7

Sélection de Colonnes Spécifiques et de Toutes les Lignes

L'ordre SELECT peut être utilisé pour afficher des colonnes spécifiques de la table. Pour cela, indiquez les noms de colonnes séparés par des virgules. L'exemple ci-dessus affiche tous les numéros de département de la table DEPT, ainsi que leur localisation.

Dans la clause SELECT, indiquez les colonnes dans l'ordre où vous souhaitez qu'elles vous soient rapportées. Par exemple, si vous voulez que la colonne "LOC" soit placée avant la colonne "DEPTNO", utilisez l'ordre suivant :

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

LOC	DEPTNO
NEW YORK	10
DALLAS	20
CHICAGO	30
BOSTON	40

Valeurs par Défaut des En-têtes de Colonne

- Justification par défaut
 - A gauche : date et données alphanumériques
 - A droite : données numériques
- Affichage par défaut : en majuscules

1-8

Valeurs par Défaut des En-têtes de Colonnes

Les en-têtes de colonnes, comme les données, alphanumériques et dates sont cadrées à gauche. Les en-têtes de colonnes, comme les données, numériques sont cadrées à droite.

```
SQL> SELECT ename, hiredate, sal
      2 FROM emp;
```

ENAME	HIREDATE	SAL
KING	17-NOV-81	5000
BLAKE	01-MAY-81	2850
CLARK	09-JUN-81	2450
JONES	02-APR-81	2975
MARTIN	28-SEP-81	1250
ALLEN	20-FEB-81	1600
...		
14 rows selected.		

Les en-têtes de colonnes alphanumériques peuvent être tronquées, mais pas les numériques. Par défaut, l'affichage se fait en majuscules. Vous pouvez substituer un alias à un en-tête de colonne à l'affichage. Les alias de colonnes sont traités un peu plus loin dans ce chapitre.

Expressions Arithmétiques

Possibilité de créer des expressions avec des données de type NUMBER et DATE au moyen d'opérateurs arithmétiques

Opérateur	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division

1-9

Expressions Arithmétiques

Si nécessaire, vous pouvez modifier l'affichage des données, effectuer des calculs ou étudier différentes hypothèses au moyen d'expressions arithmétiques. Une expression arithmétique peut contenir des noms de colonnes, des valeurs numériques constantes et des opérateurs arithmétiques.

Opérateurs Arithmétiques

Cette diapositive présente les opérateurs arithmétiques disponibles dans SQL. Vous pouvez les utiliser dans n'importe quelle clause d'un ordre SQL, excepté dans la clause FROM.

Utilisation des Opérateurs Arithmétiques

```
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.

1-10

Utilisation des Opérateurs Arithmétiques

L'exemple de la diapositive utilise l'opérateur d'addition pour calculer l'augmentation de salaire de \$300 pour tous les employés, puis affiche une nouvelle colonne SAL+300.

A noter que la colonne SAL+300 qui résulte de ce calcul n'est pas une nouvelle colonne de la table EMP ; elle n'est que l'affichage d'un résultat. Par défaut, le nom d'une nouvelle colonne est issu du calcul dont elle provient : dans ce cas précis, sal+300.

Remarque : SQL*Plus ignore les espaces situés avant et après l'opérateur arithmétique.

Priorité des Opérateurs



*** / + -**

- **La multiplication et la division ont priorité sur l'addition et la soustraction.**
- **A niveau de priorité identique, les opérateurs sont évalués de gauche à droite.**
- **Les parenthèses forcent la priorité d'évaluation et permettent de clarifier les ordres.**

1-11

Priorité des Opérateurs

Lorsqu'une expression comporte plusieurs opérateurs, les opérateurs de multiplication et de division sont évalués en premier. Lorsque les opérateurs d'une expression sont de priorité identique, ils sont évalués de la gauche vers la droite.

Pour forcer la priorité d'évaluation d'une expression, placez-la entre parenthèses.

Priorité des Opérateurs

```
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.

1-12

Priorité des Opérateurs (suite)

L'exemple de la diapositive présente le nom, le salaire et le revenu annuel des employés. Le revenu annuel est calculé en multipliant le salaire mensuel par 12, puis en ajoutant un bonus exceptionnel de \$100. Remarquez que la multiplication a été effectuée avant l'addition.

Remarque : l'utilisation des parenthèses renforce l'ordre normal de priorité des opérateurs et améliore la clarté. L'expression ci-dessus, par exemple, peut être écrite comme suit : (12*sal)+100, sans que le résultat en soit modifié.

Utilisation des Parenthèses

```
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.

1-13

Utilisation des Parenthèses

Vous pouvez modifier les règles de priorité en utilisant des parenthèses pour préciser l'ordre dans lequel les opérateurs doivent être évalués.

L'exemple de la diapositive présente les noms, salaires et revenus annuels des employés. Le calcul du revenu annuel est effectué en multipliant le salaire mensuel ainsi que le bonus de \$100 par 12. Les parenthèses rendent l'addition prioritaire sur la multiplication.

La Valeur NULL

- **NULL représente une valeur non disponible, non affectée, inconnue ou inapplicable.**
- **La valeur NULL est différente du zéro ou de l'espace.**

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

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

14 rows selected.

1-14

Les Valeurs NULL

Lorsqu'il manque une valeur dans une colonne sur une ligne, la valeur est dite *NULL*.

Une valeur NULL est une valeur non disponible, non affectée, inconnue ou inapplicable. Une valeur NULL est différente du zéro ou de l'espace. Le zéro est un chiffre et l'espace est un caractère.

Quel que soit le type de données d'une colonne, celle-ci peut contenir des valeurs NULL, excepté lorsque cette colonne a été définie comme NOT NULL ou comme CLE PRIMAIRE lors de sa création.

Dans la colonne COMM de la table EMP, vous pouvez remarquer que seul un employé occupant le poste de vendeur est habilité à toucher une commission. Ainsi, pour les non vendeurs COMM contient NULL. Turner, vendeur, n'a pas touché de commission : la valeur de sa commission est 0 et par conséquent non NULL.

Valeurs NULL dans les Expressions Arithmétiques

Les expressions arithmétiques comportant une valeur NULL sont évaluées à NULL

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

ENAME	12*SAL+COMM
-----	-----
KING	

1-15

Les Valeurs NULL (suite)

Lorsqu'une valeur NULL est utilisée dans une expression arithmétique, le résultat de cette expression est NULL. Si vous essayez de diviser par zéro, vous obtenez une erreur. En revanche, si vous divisez un nombre par NULL, le résultat sera NULL ou inconnu.

Dans l'exemple ci-dessus, l'employé KING ne fait pas partie de la catégorie SALESMAN et ne touche donc pas de commission. La valeur de la colonne COMM dans l'expression arithmétique étant NULL, le résultat est donc NULL.



Pour plus d'informations, reportez-vous à
Oracle8 Server SQL Reference, Release 8, "Elements of SQL."

L'Alias de Colonne

- **Renomme un en-tête de colonne**
- **Est utile dans les calculs**
- **Suit immédiatement le nom de la colonne ; le mot-clé AS placé entre le nom et l'alias est optionnel**
- **Doit obligatoirement être inclus entre guillemets s'il contient des espaces, des caractères spéciaux ou si les majuscules/minuscules doivent être différenciées**

1-16

Les Alias de Colonnes

Lors de l'affichage des résultats d'une requête, SQL*Plus prend le nom de la colonne sélectionnée comme en-tête de colonne. La plupart du temps, cet en-tête n'est pas explicite et par conséquent est difficile à comprendre. L'alias de colonne permet de modifier l'en-tête d'une colonne.

On spécifie l'alias à la suite de la colonne dans la liste SELECT en utilisant le caractère espace comme séparateur. Par défaut, les en-têtes de colonne sont en majuscules. Si l'alias contient des espaces ou des caractères spéciaux (tels que # ou \$), ou si la différence entre majuscules et minuscules est importante, placez l'alias entre guillemets (" ").

Utilisation des Alias de Colonnes

```
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
-----
...
```

1-17

Les Alias de Colonnes (suite)

Le premier exemple présente le nom et le salaire mensuel de tous les employés. Remarquez que le mot-clé optionnel AS a été placé avant l'alias de colonne. L'utilisation ou non de ce mot-clé ne modifie pas le résultat de la requête. Notez également que dans l'ordre SQL, les alias de colonne 'name' et 'salary' sont spécifiés en minuscules, alors que les en-têtes de colonne apparaissent en majuscules dans le résultat de la requête. Comme il a été précisé sur la diapositive précédente, c'est le mode d'affichage par défaut.

Le second exemple présente le nom et le salaire annuel de tous les employés. L'alias Annual Salary a été placé entre guillemets car il contient un espace. Remarquez que cette fois l'en-tête de colonne affiché est strictement identique à l'alias de colonne.

L'Opérateur de Concaténation

- **Concatène des colonnes ou chaînes de caractères avec d'autres colonnes**
- **Est représenté par deux barres verticales (||)**
- **La colonne résultante est une expression caractère**

1-18

L'Opérateur de Concaténation

L'opérateur de concaténation (||) permet de concaténer des colonnes à d'autres colonnes, à des expressions arithmétiques ou à des valeurs constantes afin de créer une expression caractère. Les colonnes situées de part et d'autre de l'opérateur se combinent pour donner une colonne unique lors de la restitution des données.

Utilisation de l'Opérateur de Concaténation

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

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

1-19

L'Opérateur de Concaténation (suite)

Dans cet exemple, ENAME et JOB sont concaténés et reçoivent l'alias Employees. On peut remarquer que noms et emplois sont combinés pour ne donner qu'une colonne à l'affichage.



Le mot-clé AS placé devant l'alias simplifie la lecture de la clause SELECT.

Littéral

- **Un littéral est un caractère, une expression, ou un nombre inclus dans la liste SELECT.**
- **Les valeurs littérales de type date et caractère doivent être placées entre simples quotes.**
- **Chaque littéral apparaît sur chaque ligne ramenée.**

1-20

Chaînes de Caractères Littérales

Un littéral est un caractère, une expression, ou un nombre quelconque inclus dans la liste SELECT, et qui n'est ni un nom de colonne ni un alias de colonne. Il apparaît sur chaque ligne ramenée. Des chaînes de texte littérales en format libre peuvent être intégrées au résultat de la requête. Elles sont traitées comme les colonnes dans une liste SELECT.

Les littéraux date et alphanumérique *doivent* être inclus entre simples quotes (' '), mais pas les littéraux numériques.

Utilisation des Chaînes de Caractères Littérales

```
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.
```

1-21

Chaînes de Caractères Littérales (suite)

L'exemple ci-dessus présente le nom et le poste de tous les employés. L'en-tête de la colonne est Employee Details. Remarquez l'espace situé entre les simples quotes dans l'ordre SELECT. Les espaces améliorent la lisibilité des résultats.

Dans l'exemple suivant, le nom et le salaire de chaque employé est concaténé avec un littéral pour que le résultat soit plus parlant.

```
SQL> SELECT ename || ': 1 Month salary = ' || sal Monthly
2 FROM      emp;
```

```
MONTHLY
-----
KING: 1 Month salary = 5000
BLAKE: 1 Month salary = 2850
CLARK: 1 Month salary = 2450
JONES: 1 Month salary = 2975
MARTIN: 1 Month salary = 1250
ALLEN: 1 Month salary = 1600
TURNER: 1 Month salary = 1500
...
14 rows selected.
```

Doublons

Par défaut, le résultat d'une requête affiche toutes les lignes, y compris les doublons.

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

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

1-22

Doublons

Par défaut, l'instruction SELECT retourne un résultat sans éliminer les doublons. L'exemple ci-dessus affiche la totalité des numéros de département de la table EMP. Vous pouvez remarquer que certains numéros de département apparaissent plusieurs fois.

Elimination des Doublons

Pour éliminer les doublons il faut ajouter le mot-clé DISTINCT à la clause SELECT.

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

DEPTNO
10
20
30

1-23

Doublons (suite)

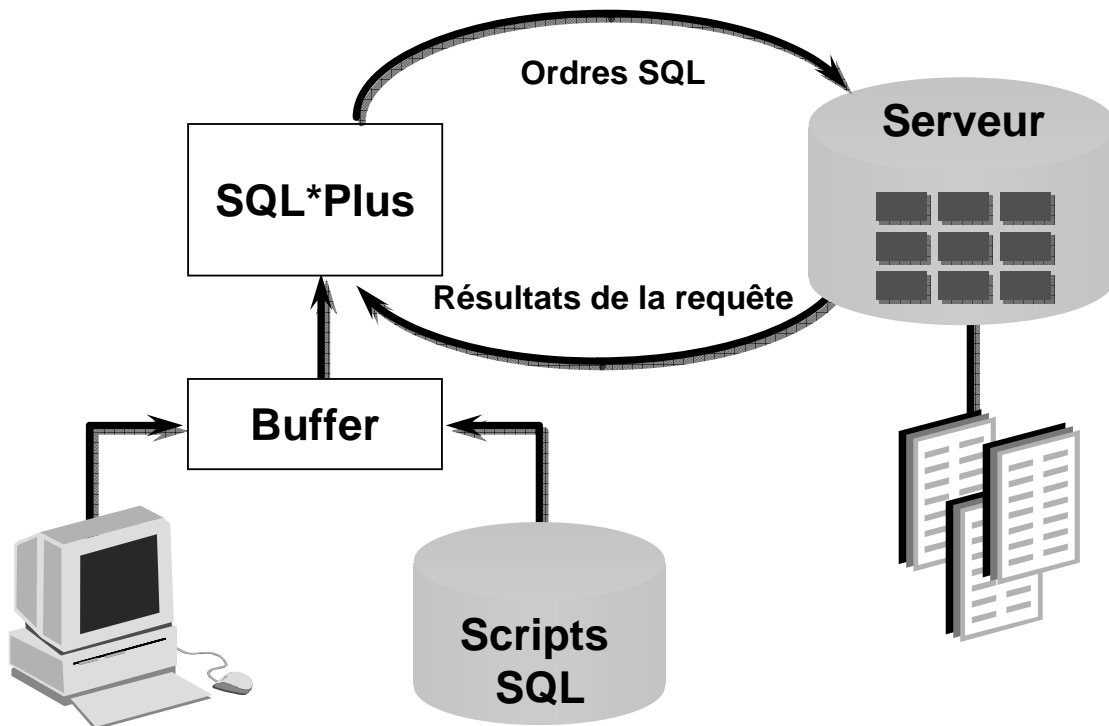
Pour éliminer les doublons du résultat d'une requête, ajoutez le mot-clé DISTINCT à la clause SELECT, directement à la suite du mot-clé SELECT. Dans l'exemple ci-dessus, la table EMP comporte 14 lignes, mais seulement trois numéros de département distincts.

Il est possible de spécifier plusieurs colonnes à la suite du paramètre DISTINCT. Ce paramètre agit sur toutes les colonnes sélectionnées, on obtient en résultat les combinaisons distinctes de ces colonnes.

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

```
DEPTNO JOB
-----
10 CLERK
10 MANAGER
10 PRESIDENT
20 ANALYST
...
9 rows selected.
```


Interaction entre SQL et SQL*Plus



1-24

SQL et SQL*Plus

SQL est un langage de commande qui permet de communiquer avec Oracle Server quels que soient l'outil ou l'application utilisés. Le SQL d'Oracle comprend de nombreuses extensions.

*SQL*Plus* est un outil Oracle possédant son propre langage de commande qui reconnaît les ordres SQL et les envoie à Oracle Server pour qu'ils soient exécutés. Lorsque vous saisissez un ordre SQL, celui-ci est stocké dans un espace mémoire appelé *buffer SQL* où il reste jusqu'à la saisie d'un nouvel ordre.

Caractéristiques de SQL

- Peut être employé par de nombreux utilisateurs, y compris ceux ayant peu ou pas d'expérience de programmation
- C'est un langage non procédural
- Il réduit le temps de création et de mise à jour des systèmes
- C'est un langage proche de la langue anglaise

Caractéristiques de SQL*Plus

- Accepte la saisie d'ordres ad hoc
- Accepte l'exécution d'ordres SQL à partir de fichiers
- Contient un éditeur de lignes pour la modification des ordres SQL
- Contrôle les paramètres d'environnement
- Formate les résultats des requêtes dans des états élémentaires
- Accède aux bases de données locales ou distantes

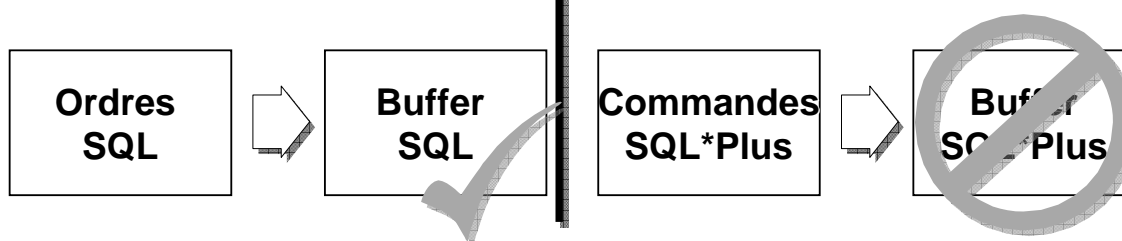
Comparatif entre Ordres SQL et Commandes SQL*Plus

SQL

- Un langage
- Standard ANSI
- Abréviation des mots-clés impossible
- Les ordres agissent sur le contenu et la définition des tables de la base de données

SQL*Plus

- Un environnement
- Produit propriétaire Oracle
- Abréviation des mots-clés possible
- Les commandes ne permettent d'agir ni sur le contenu, ni sur la définition des tables.



1-25

SQL et SQL*Plus (suite)

La table ci-dessous présente un comparatif de SQL et SQL*Plus.

SQL	SQL*Plus
Langage de communication avec Oracle Server pour accéder aux données	Reconnait les ordres SQL et les envoie au serveur
Basé sur le SQL standard ANSI (American National Standards Institute)	Interface propriétaire Oracle pour l'exécution des ordres SQL
Manipule les données et les définitions de tables dans la base de données	Les commandes SQL*Plus ne permettent pas la manipulation de valeurs dans la base de données
Est entré dans le buffer SQL sur une ou plusieurs lignes.	Entrée des lignes une par une, pas de stockage dans le buffer SQL
Ne comporte pas de caractère de continuation	Le tiret (-) est utilisé comme caractère de continuation lorsque la commande fait plus d'une ligne
Ne peut être abrégé	Peut être abrégé
Comporte un caractère de fin pour l'exécution immédiate des commandes	Ne nécessite pas de caractère de fin ; les commandes sont immédiatement exécutées
Utilise des fonctions pour effectuer les formats	Utilise des commandes pour formater les données

Présentation de SQL*Plus

- **Connexion à SQL*Plus.**
- **Affichage de la structure d'une table.**
- **Edition d'un ordre SQL.**
- **Exécution de SQL à partir de SQL*Plus.**
- **Enregistrement ou ajout d'ordres SQL dans un fichier.**
- **Exécution d'ordres contenus dans un fichier.**
- **Chargement dans le buffer de l'ordre contenu dans un fichier.**

1-26

SQL*Plus

L'environnement SQL*Plus permet :

- L'extraction, la modification, l'ajout et la suppression de données dans la base de données au moyen d'ordres SQL
- Le formatage, le calcul, le stockage et l'impression des résultats de requête sous la forme d'états
- La création de fichiers script pour le stockage d'ordres SQL dans le but d'une utilisation répétée

Les commandes SQL*Plus se divisent selon les catégories suivantes :

Catégorie	Objet
Environnement	Influencer le comportement général des ordres SQL pour la session en cours
Formatage	Formater les résultats de requêtes
Manipulation de fichiers	Sauvegarder, charger et exécuter des fichiers scripts
Exécution	Envoyer l'ordre SQL du buffer SQL vers Oracle Server
Edition	Modifier l' ordre SQL contenu dans le buffer
Interaction	Créer et transmettre des variables aux ordres SQL, imprimer les valeurs des variables et afficher des messages.
Divers	Nombreuses commandes permettant de se connecter à la base de données, de manipuler l'environnement SQL*Plus et d'afficher les définitions de colonnes

Connexion à SQL*Plus

• Depuis l'environnement Windows :

• Depuis une ligne de commande :

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

1-27

Connexion à SQL*Plus

La procédure d'appel de SQL*Plus varie en fonction de votre système d'exploitation ou de votre environnement Windows.

Pour se connecter depuis l'environnement Windows NT :

1. Cliquez sur Démarrer—>Programmes—>Oracle for Windows NT—>SQL*Plus
2. Entrez votre nom d'utilisateur, votre mot de passe et le nom de la chaîne de connexion à la base de données.

Pour se connecter depuis une ligne de commande :

1. Ouvrez une session sur votre machine
2. Saisissez la commande SQL*Plus comme indiqué sur la diapositive ci-dessus.

Syntaxe de la commande :

<i>username</i>	votre nom d'utilisateur de base de données
<i>password</i>	votre mot de passe ; à ce niveau, le mot de passe est visible
<i>@database</i>	la chaîne de connexion à la base de données

Remarque : pour garantir l'intégrité de votre mot de passe, ne le saisissez pas au prompt du système d'exploitation. Saisissez uniquement votre nom d'utilisateur, puis saisissez votre mot de passe au prompt Mot de passe.

Lorsque vous êtes bien connecté à SQL*Plus, un message équivalent au message suivant apparaît :

```
SQL*Plus : Release 8.0.3.0.0 - Production on Mon Oct 06 16:03:43
1997
```

(c) Copyright 1997 Oracle Corporation. All rights reserved.

Affichage de la Structure d'une Table

Utilisez la commande SQL*Plus DESCRIBE pour afficher la structure d'une table.

```
DESC[RIBE] tablename
```

1-28

Affichage de la Structure d'une Table

Dans SQL*Plus, vous pouvez afficher la structure d'une table à l'aide de la commande DESCRIBE. Cette commande affiche les noms de colonnes et les types de données, ainsi qu'une information indiquant si les colonnes *doivent* obligatoirement contenir des données ou non.

Syntaxe :

tablename

nom d'une table, d'une vue ou d'un synonyme existants,
accessibles à l'utilisateur

Affichage de la Structure d'une Table

```
SQL> DESCRIBE dept
```

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

1-29

Affichage de la Structure d'une Table (suite)

L'exemple ci-dessus affiche des informations sur la structure de la table DEPT.

Dans le résultat :

Null? indique si une colonne *doit ou non* contenir des données; NOT NULL signifie que la colonne doit contenir des données

Type affiche le type de données d'une colonne

Les types de données sont décrits dans le tableau suivant :

Type de données	Description
NUMBER(<i>p,s</i>)	Valeur numérique contenant au maximum <i>p</i> chiffres, dont <i>s</i> chiffres après la virgule
VARCHAR2(<i>s</i>)	Valeur caractère de longueur variable, d'une taille maximale <i>s</i>
DATE	Valeurs de date et heure situées entre le 1 ^{er} janvier 4712 av.J.-C. et le 31 décembre 9999 apr.J.-C.
CHAR(<i>s</i>)	Valeur caractère de longueur fixe <i>s</i>

Commandes d'Edition SQL*Plus

- **A[PPEND] *text***
- **C[HANGE] / *old* / *new***
- **C[HANGE] / *text* /**
- **CL[EAR] BUFF[ER]**
- **DEL**
- **DEL *n***
- **DEL *m n***

1-30

Commandes d'Editions SQL*Plus

Les commandes SQL*Plus sont saisies une à une et ne sont pas stockées dans le buffer SQL.

Commande	Description
A[PPEND] <i>text</i>	Ajoute du texte à la fin de la ligne courante
C[HANGE] / <i>old</i> / <i>new</i>	Remplace le texte <i>old</i> par le <i>new</i> dans la ligne courante
C[HANGE] / <i>text</i> /	Supprime le <i>text</i> dans la ligne courante
CL[EAR] BUFF[ER]	Supprime toutes les lignes du buffer SQL
DEL	Supprime la ligne courante

Conseils

- Si vous appuyez sur la touche [Entrée] avant d'avoir terminé une commande SQL, SQL*Plus vous demande un numéro de ligne.
- Pour terminer une saisie dans le buffer SQL, saisissez un des caractères de terminaison (point-virgule ou slash) ou appuyez deux fois sur la touche [Entrée]. Le prompt SQL apparaît alors.

Commandes d'Edition SQL*Plus

- I[NPUT]
- I[NPUT] *text*
- L[IST]
- L[IST] *n*
- L[IST] *m n*
- R[UN]
- *n*
- *n text*
- *0 text*

1-31

Commandes d'Edition SQL*Plus (suite)

Commande	Description
I[NPU T]	Insère un nombre indéfini de lignes
I[NPUT] <i>text</i>	Insère la ligne <i>text</i>
L[IST]	Affiche toutes les lignes du buffer SQL
L[IST] <i>n</i>	Affiche une ligne (indiquée par <i>n</i>)
L[IST] <i>m n</i>	Affiche une série de lignes (<i>m</i> à <i>n</i>)
R[UN]	Affiche et exécute l'ordre SQL actuellement dans le buffer
<i>n</i>	Indique quelle doit être la ligne courante
<i>n text</i>	Remplace la ligne <i>n</i> par <i>text</i>
<i>0 text</i>	Insère une ligne avant la ligne 1



On ne peut saisir qu'une seule commande SQL*Plus par prompt SQL. Les commandes SQL*Plus ne sont pas stockées dans le buffer. Pour poursuivre la saisie d'une commande SQL*Plus sur la ligne suivante, entrez un tiret (-) à la fin de la ligne courante.

Commandes de Fichiers SQL*Plus

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

1-32

Commandes de Fichiers SQL*Plus

Les ordres SQL communiquent avec Oracle. Les commandes SQL*Plus contrôlent l'environnement, formatent les résultats des requêtes et gèrent les fichiers. Pour gérer les fichiers, vous disposez des commandes décrites dans le tableau suivant :

Commande	Description
SAV[E] <i>filename</i> [.ext] [REP[LACE]APP[END]]	Sauvegarde dans un fichier le contenu actuel du buffer SQL. APPEND ajoute le contenu à un fichier existant ; REPLACE écrase un fichier existant. L'extension par défaut est .sql.
GET <i>filename</i> [.ext]	Charge le contenu d'un fichier déjà sauvegardé dans le buffer SQL. Par défaut, l'extension du fichier est .sql.
STA[RT] <i>filename</i> [.ext]	Exécute un fichier de commandes déjà sauvegardé.
@ <i>filename</i>	Exécute un fichier de commandes déjà sauvegardé (identique à START).
ED[IT]	Appelle l'éditeur et sauvegarde le contenu du buffer dans un fichier nommé <i>afiedt.buf</i> .
ED[IT] [<i>filename</i> [.ext]]	Appelle l'éditeur et édite le contenu d'un fichier déjà sauvegardé.
SPO[OL] [<i>filename</i> [.ext]] OFF[OUT]	Stocke les résultats de requêtes dans un fichier. OFF ferme le fichier spool. OUT ferme le fichier spool et envoie les résultats du fichier à l'imprimante du système.
EXIT	Quitte SQL*Plus.

Résumé

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

L'environnement SQL*Plus permet :

- **D'exécuter des ordres SQL**
- **D'éditer des ordres SQL**

1-33

Ordre SELECT

Dans ce chapitre, vous avez appris à extraire des données d'une table dans une base de données à l'aide de l'ordre SELECT.

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

Où :	SELECT	Est une liste contenant au moins une colonne
	DISTINCT	Supprime les doublons
	*	Sélectionne toutes les colonnes
	<i>column</i>	Sélectionne la colonne nommée
	<i>alias</i>	Remplace l'en-tête de la colonne sélectionnée
	FROM <i>table</i>	Spécifie la table qui contient les colonnes

SQL*Plus

SQL*Plus est un environnement d'exécution qui vous permet d'envoyer des commandes SQL au serveur de la base de données, ainsi que d'éditer et de sauvegarder des commandes SQL. Les commandes peuvent être exécutées à partir du prompt SQL ou depuis un fichier script.

Présentation des Exercices

- **Afficher les données de différentes tables.**
- **Afficher la structure des tables.**
- **Calculs arithmétiques et spécifications des noms de colonnes.**
- **Utilisation de l'éditeur SQL*Plus.**

1-34

Présentation des Exercices

Voici le premier d'une longue série d'exercices. Les solutions (si vous en avez besoin) sont dans l'annexe A. Les exercices ont pour objet de présenter tous les sujets abordés dans ce chapitre. Répondez aux questions 2 à 4 sur papier.

Certains exercices contiennent des questions "si vous avez du temps" ou "pour aller plus loin". Ne les faites que si vous avez répondu à toutes les autres questions dans le temps imparti et si vous souhaitez tester vos connaissances plus avant



Commencez les exercices sans vous presser et consciencieusement. Vous pouvez vous entraîner à sauvegarder et à exécuter des fichiers de commande. Si nécessaire, n'hésitez pas à poser des questions à votre instructeur.

Questions sur Papier

Pour les questions 2 à 4, entourez Vrai ou Faux.

Exercices 1

1. Initialisez une session SQL*Plus avec votre ID et le mot de passe que votre instructeur vous a remis.
2. Les commandes SQL*Plus accèdent aux bases de données.
Vrai/Faux
3. L'ordre SELECT suivant sera convenablement exécuté.
Vrai/Faux

```
SQL> SELECT      ename, job, sal Salary
2 FROM          emp;
```

4. L'ordre SELECT suivant sera convenablement exécuté.
Vrai/Faux

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

5. Cet ordre comporte trois erreurs de code ; pouvez-vous les trouver ?

```
SQL> SELECT      empno, ename
2                sal x 12 ANNUAL SALARY
3 FROM          emp;
```

6. Affichez la structure de la table DEPT. Sélectionnez toutes les données de la table DEPT.

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

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

Exercices 1

7. Affichez la structure de la table EMP . Créez une requête pour afficher le nom (ename), le poste (job), la date d'embauche (hiredate) et le matricule (empno) de chaque employé, en plaçant le matricule en premier. Enregistrez votre ordre SQL dans un fichier nommé *p1q7.sql*.

Name	Null?	Type
-----	-----	----
EMPNO	NOT NULL	NUMBER (4)
ENAME		VARCHAR2 (10)
JOB		VARCHAR2 (9)
MGR		NUMBER (4)
HIREDATE		DATE
SAL		NUMBER (7 , 2)
COMM		NUMBER (7 , 2)
DEPTNO		NOT NULL NUMBER (2)

8. Exécutez la requête que vous avez placée dans le fichier *p1q7.sql*.

EMPNO	ENAME	JOB	HIREDATE
-----	-----	-----	-----
7839	KING	PRESIDENT	17-NOV-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7499	ALLEN	SALESMAN	20-FEB-81
7844	TURNER	SALESMAN	08-SEP-81
7900	JAMES	CLERK	03-DEC-81
7521	WARD	SALESMAN	22-FEB-81
7902	FORD	ANALYST	03-DEC-81
7369	SMITH	CLERK	17-DEC-80
7788	SCOTT	ANALYST	09-DEC-82
7876	ADAMS	CLERK	12-JAN-83
7934	MILLER	CLERK	23-JAN-82
14 rows selected.			

Exercices 1

9. Créez une requête pour afficher les différents types de poste existant dans la table EMP.

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

Si vous avez du temps, faites les exercices suivants :

10. Editez *p1q7.sql*. Donnez respectivement les noms suivants aux en-têtes de colonne : Emp #, Employee, Job, et Hire Date. Exécutez à nouveau votre requête.

```
Emp # Employee   Job           Hire Date
-----
7839 KING        PRESIDENT    17-NOV-81
7698 BLAKE       MANAGER      01-MAY-81
7782 CLARK       MANAGER      09-JUN-81
7566 JONES       MANAGER      02-APR-81
7654 MARTIN      SALESMAN     28-SEP-81
7499 ALLEN       SALESMAN     20-FEB-81
7844 TURNER      SALESMAN     08-SEP-81
7900 JAMES       CLERK        03-DEC-81
7521 WARD        SALESMAN     22-FEB-81
7902 FORD        ANALYST      03-DEC-81
7369 SMITH       CLERK        17-DEC-80
7788 SCOTT       ANALYST      09-DEC-82
7876 ADAMS       CLERK        12-JAN-83
7934 MILLER      CLERK        23-JAN-82
14 rows selected.
```

Exercices 1

11. Affichez le nom concaténé avec le poste en les séparant par une virgule suivie d'un espace, puis donnez comme titre à la colonne Employee and Title.

```
Employee and Title
-----
KING, PRESIDENT
BLAKE, MANAGER
CLARK, MANAGER
JONES, MANAGER
MARTIN, SALESMAN
ALLEN, SALESMAN
TURNER, SALESMAN
JAMES, CLERK
WARD, SALESMAN
FORD, ANALYST
SMITH, CLERK
SCOTT, ANALYST
ADAMS, CLERK
MILLER, CLERK
14 rows selected.
```

Si vous souhaitez aller plus loin dans la difficulté, faites l' exercice suivant :

12. Créez une requête pour afficher toutes les données de la table EMP dans une seule colonne d'affichage. Séparez chaque colonne par une virgule. Nommez la colonne d'affichage THE_OUTPUT.

```
THE_OUTPUT
-----
7839,KING,PRESIDENT,,17-NOV-81,5000,,10
7698,BLAKE,MANAGER,7839,01-MAY-81,2850,,30
7782,CLARK,MANAGER,7839,09-JUN-81,2450,,10
7566,JONES,MANAGER,7839,02-APR-81,2975,,20
7654,MARTIN,SALESMAN,7698,28-SEP-81,1250,1400,30
7499,ALLEN,SALESMAN,7698,20-FEB-81,1600,300,30
7844,TURNER,SALESMAN,7698,08-SEP-81,1500,0,30
7900,JAMES,CLERK,7698,03-DEC-81,950,,30
7521,WARD,SALESMAN,7698,22-FEB-81,1250,500,30
7902,FORD,ANALYST,7566,03-DEC-81,3000,,20
7369,SMITH,CLERK,7902,17-DEC-80,800,,20
7788,SCOTT,ANALYST,7566,09-DEC-82,3000,,20
7876,ADAMS,CLERK,7788,12-JAN-83,1100,,20
7934,MILLER,CLERK,7782,23-JAN-82,1300,,10
14 rows selected.
```