
 PRODSET Groupe SORIA	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
	Objet : Cours de base ORACLE	Date : 27/10/2021


COURS ORACLE DE BASE

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>


Cours de base Oracle

V0. 4/1/05

1	Qu'est-ce qu'une base de données et un SGBD ?	4
2	Qu'est-ce qu'une base de données relationnels et un SGBDR ?.....	4
3	Les bases Oracles et Oracle par rapport aux autres bases (DB2, SYBASE ...)	5
4	Le SQL et SQLPLUS	8
5	Notions de tables, index, vues.....	9
5.1	La table, description.....	9
5.2	Les types de données.....	9
5.3	La clause NULL	10
5.4	Les noms de TABLE.....	11
6	Les vues.....	11
6.1	Les Index.....	12
6.2	Les Synonymes.....	15
7	L'ordre SELECT	16
7.1	Généralités	16
7.2	Définition des colonnes.....	17
7.3	Définition clause FROM.....	21
7.4	Définition clause WHERE.....	22
7.5	Clause WHERE et la structure des critères de sélection.....	23
7.6	Définition clause GROUP BY.....	26
7.7	Définition clause HAVING.....	27
7.8	Définition clause ORDER BY.....	28
7.9	Définition du SUBSELECT.....	29
7.10	Définition d'une jointure.....	30
8	Le langage de manipulation	31
8.1	L'ordre INSERT	31
8.2	L'ordre UPDATE.....	33
8.3	L'ordre DELETE.....	34
9	Le langage de contrôle	35
9.1	La commande GRANT.....	35
9.2	La commande REVOKE.....	36
10	Le langage de définition	38
10.1	La commande CREATE.....	38
10.2	Le Create VIEW.....	40

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

10.3	Le Create INDEX.....	40
10.4	La commande DROP.....	41
10.5	La commande ALTER.....	42
11	Annexe : notions d'appel de SQL dans un autre programme (« PRO*SQL », exemple Pro*C, Pro*Cobol ...)	43
11.1	Structure du programme appelant.....	43
11.2	La zone de déclaration SQLCA.....	44
11.3	La déclaration des objets.....	44
11.4	La gestion des erreurs.....	44
11.5	La gestion des HOST VARIABLES.....	45
11.6	Le SELECT INTO.....	46
11.7	L'ordre INSERT.....	48
11.8	L'ordre UPDATE.....	49
11.9	L'ordre DELETE.....	50
11.10	Le SELECT de plusieurs lignes (CURSEURS).....	50
11.11	La déclaration du CURSOR.....	51
11.12	Le traitement (OPEN, FETCH, CLOSE).....	52
11.13	La mise à jour avec CURSOR.....	55
12	Notion de PL/SQL.....	56
12.1	Exemples de programmes PL/SQL	56
13	Exemple de script de création de table.....	58
14	Exemple de script de création de synonymes.....	59
15	Exemple de scripts d'insertions de données dans une table.....	60
16	Exemple de scripts de création / gestion de droits	62

	<p>PLAN de FORMATION</p>	<p>cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p>Objet : Cours de base ORACLE</p>	<p>Date : 27/10/2021</p>

1 Qu'est-ce qu'une base de données et un SGBD ?

C'est un ensemble de données en relation entre elles, et de fichiers associés.
Ces fichiers peuvent être indexés ou non

Des programmes gestionnaires permettent aux utilisateurs et des programmes d'accéder à ces données et fichiers. L'ensemble de ces programmes sont dénommés habituellement « Système de gestion de base de données » (SGBD).

- ◆ Un SGBD assure :
 - l'INTEGRITE,
 - la COHERENCE,
 - La SECURITE.

Des données de cette base.

- ◆ Le SGBD permet la gestion des accès concurrents aux données entre plusieurs utilisateurs.


Historiquement, il y a eu au départ des bases de données sans systèmes SGBDL de mise en relation des données.

Puis est apparu ce que l'on appelle les « bases réseau », vers les années 70, où les données étaient mis en relation entre elle par des pointeurs ou index. En général, ces pointeurs ou ces index étaient créées une fois pour toute, à la création de la base. Et ensuite, il n'était pas facile de les modifier.

2 Qu'est-ce qu'une base de données relationnels et un SGBDR ?

C'est une façon de présenter les données sous la forme de tableaux et de vues sur ces tableaux, à l'aide d'un langage relationnel, simple proche du langage naturel, dit de « 4^o génération » (en général, le langage SQL).

Avec ce genre de bases relationnel, on peut gérer souplement les relations entre données, voire les modifier en cours de route (bien qu'il faut le faire avec prudence, et que cela ne soit pas souvent recommandé).

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

- ◆ Les données sont regroupées en TABLE.

TABLCONC

DEPT	CODE CONCESSION	NOM CONCESSION	MARQUE	CLIENTS 1996
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	A.V.A.	FORD	150


TABLEDEP

CODE DEPARTEMENT	NOM DEPARTEMENT	HABITANTS (M)
29	FINISTERE	1,2
22	COTES D'ARMOR	0,5
56	MORBIHAN	0,8
85	VENDEE	0,4
44	LOIRE-ATLANTIQUE	2,1

- ◆ L'intersection d'une LIGNE et d'une COLONNE est un CHAMP.
- ◆ Un CHAMP non renseigné est NULL.
- ◆ Ordre quelconque des LIGNES et des COLONNES.

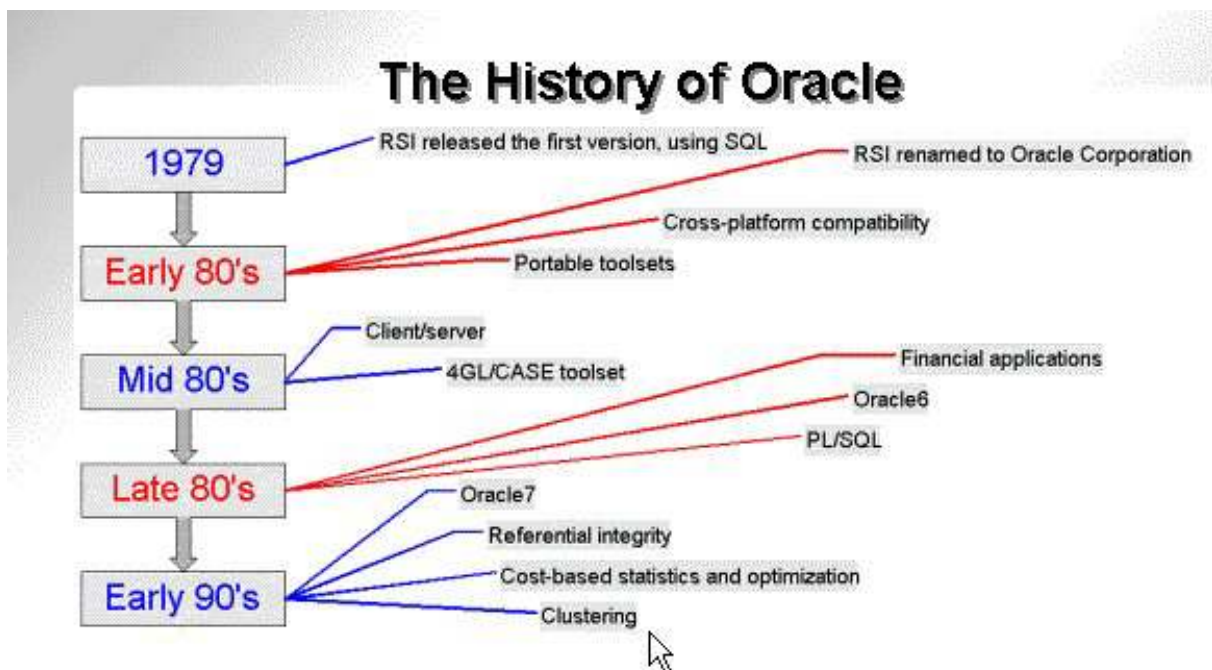
3 Les bases Oracles et Oracle par rapport aux autres bases (DB2, SYBASE ...).

- ◆ Les données d'ORACLE sont manipulables par un langage de haut niveau (SQL), ce qui permet :

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

- La portabilité des applications,
- Une plus grande durée de vie des applications,
- La communication entre systèmes,
- Une formation des intervenants à un seul interface (SQL ...).


Ce langage a été créé par IBM dans les années 1976 et mis dans le domaine public.
ORACLE qui a repris le langage SQL a été créé vers 1979.

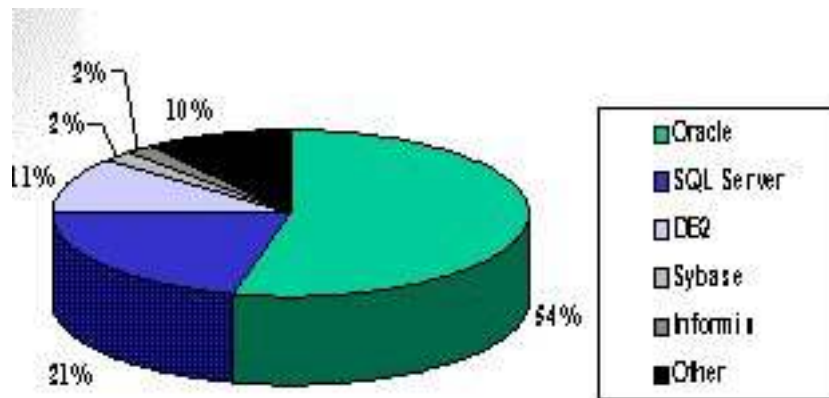


Petit schéma de l'historique d'Oracle jusqu'aux années 90.

Plusieurs sociétés concurrentes, à la société ORACLE, ont créé leur propre base de données relationnelle, vers la même époque :

- IBM, avec la base DB2, sur gros systèmes IBM.
- INGRES,
- SYBASE.
- ...

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>



Parts de marché d'Oracle par rapport à la concurrence en 2004.

Puis des bases relationnelles ont été créées sur micros _ comme ACCESS de Microsoft etc ...
_, et sur serveurs Windows _ comme SQLServer de Microsoft ...

Le relationnel avec le langage SQL est devenu une norme universelle largement répandue.


Oracle se situe dans la gamme des bases de données de haut de gamme et haut niveau, très puissante. Son offre logiciel est immense. Et beaucoup d'autres progiciels tourne sur un noyau Oracle (des outils de CRM, de gestions du personnel _ par exemple SAP ... _, de gestion de parc informatique ...).

ORACLE est disponible sur pratiquement toutes les machines du marché (UNIX, gros système, Windows, MAC OS, Tandem etc ...).

Toute application ORACLE est entièrement portable (à ~100 %) d'une machine ou plate-forme matérielle à l'autre, du moment que c'est la même version d'ORACLE installée sur ces machines. La très grande force d'ORACLE est sa portabilité intégrale d'une machine à l'autre (quelque soit le système d'exploitation de cette dernière).

Sa seconde force est la fiabilité d'ORACLE. Sinon, les messages d'erreurs d'ORACLE sont numérotés, parfaitement documentés, expliquées dans le détail (comme pour la base de donnée concurrente gros système DB2).

Sa troisième force est qu'ORACLE peut aussi bien fonctionner avec des petites bases ou des bases gigantesques de l'ordre du Terabytes (du milliard de milliard d'octets, 10^{12} octets). Avec un petit nombre d'utilisateurs, ou avec un nombre d'utilisateurs immense (de l'ordre du million d'utilisateurs).

	<p align="center">PLAN de FORMATION</p>	<p align="center">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="center">Date : 27/10/2021</p>

4 Le SQL et SQLPLUS

SQL signifie Structured Query Language (langage d'interrogation structurée).

Pour ce langage SQL, une base est un regroupement logique de TABLES.

Dans le langage SQL, les notions les plus importantes sont celles de tables (TABLE), de colonnes, de lignes dans ces tables, d'index sur ces tables, de vues sur ces tables (VIEW). Un ordre important dans ce langage, est l'ordre d'interrogation de table, ou de plusieurs tables, de vues, de lignes et de colonnes : SELECT (dont nous verrons la syntaxe au paragraphe suivant).

Le langage SQL est une sorte de « JCL » d'accès aux données de la base.

L'interface la plus simple pour taper ou exécuter en batch, les ordres SQL est le programme SQLPLUS d'ORACLE.

On y tape des commandes sous la forme de lignes de commandes.

Les résultats apparaissent sous la forme d'états (de rapports) sous SQLPLUS, qu'on peut formater, en plus, avec des commandes SQLPLUS.

Mais on peut aussi passer des ordres SQL à travers des écrans de saisies et d'interrogation (l'outil natif Oracle de gestion d'écrans transactionnels étant SQL*Forms) ou de génération de rapports (l'outil natif Oracle de gestion de rapports / d'états étant SQL*Report). Et bien d'autres interfaces encore (Business Object, VB, PowerBuilder ... etc. etc. ...).

- ◆ Langage de définition (DDL : Data Definition Language).


CREATE
DROP
ALTER

- ◆ Langage de manipulation (DML : Data Manipulation Language).

SELECT
INSERT
UPDATE
DELETE

- ◆ Langage de contrôle.

GRANT
REVOKE

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

5 Notions de tables, index, vues.

5.1 La table, description.

- ◆ L'objet des bases de données relationnelles.
- ◆ Elle contient les données organisées en LIGNES et COLONNES


```

CREATE TABLE table-name
           (column-definition)
           [IN database-name.tablespace-name]
           ...

```

5.2 Les types de données

- ◆ CHAR (n)
- ◆ VARCHAR (n)
- ◆ SMALLINT
(Entiers compris entre -32768 et 32767)
- ◆ INTEGER
(Entiers compris entre -2 147 483 648 et 2 147 483 647)
- ◆ DECIMAL (p , s)
(p = nombre total de digits, s = digits après la virgule)
- ◆ FLOAT (n)
- ◆ DATE
(Exemple : '1997-01-31')
- ◆ TIME
(Exemple : '18.15.24')
- ◆ TIMESTAMP
(Exemple : '1997-01-31-18.15.24.456789')

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

- ◆ GRAPHIC (n)
- ◆ VARGRAPHIC (n)

5.3 La clause NULL

- ◆ NULL indique un champ non renseigné.

TABLCONC

DEPT	CODE CONCESSION	NOM CONCESSION	MARQUE	CLIENTS 1996
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	--
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	A.V.A.	FORD	150

Champ NULL ←

- ◆ Si aucune précision au niveau du champ, alors un champ NULL est accepté.
- ◆ Si NOT NULL est précisé à la définition du CHAMP, alors le Champ doit être renseigné.
- ◆ Si NOT NULL WITH DEFAULT est précisé à la définition du CHAMP, alors le Champ prend les valeurs :

0, pour les numériques,


blanc, pour alphanumériques.

- ◆ NULL est ignoré dans les fonctions COLONNES.

Exemple :

MARQUE	STOCK
RENAULT	12
PEUGEOT	--
FIAT	12

AVG(STOCK) = 12

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

NOT NULL WITH DEFAULT donnerait :
AVG(STOCK) = 8

5.4 Les noms de TABLE.

- ◆ Format général

CREATEUR.NOM_TABLE

- ◆ Format implicite

NOM_TABLE

Exemple :

⇒ DBADM.TABLCONC


⇒ TABLCONC est accessible par le USER DBADM ou par une personne autorisée par DBADM.

⇒ DBADM.TABLCONC et ETUD.TABLCONC sont deux tables différentes et l'appel implicite est identique.

6 Les vues.

TABLCONC

DEPT	CODE CONCESSION	NOM CONCESSION	MARQUE	CLIENTS 1996
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	A.V.A.	FORD	150

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

TABLEDEP

CODE DEPARTEMENT	NOM DEPARTEMENT	HABITANTS (M)
29	FINISTERE	1,2
22	COTES D'ARMOR	0,5
56	MORBIHAN	0,8
85	VENDEE	0,4
44	LOIRE-ATLANTIQUE	2,1

DEPARTEMENT


NOM DEPT	NOM	HABITANT	NBCLIENT
FINISTERE	AUTOMOBILE BRESTOISE	1,2	520
FINISTERE	CONCESSION PEUGEOT QUIMPER	1,2	325
MORBIHAN	LA LORIENTAISE DE L'AUTOMOBILE	0,8	150
MORBIHAN	VANNES EUROPE AUTOMOBILE	0,8	246
VENDEE	A.V.A.	0,4	150

- ◆ Restrictions en mise à jour, pas de mise à jour si compaction de l'information.

```
CREATE VIEW view-name
      (column-name, column-name, ...)
AS SELECT ...
...
```

6.1 Les Index.

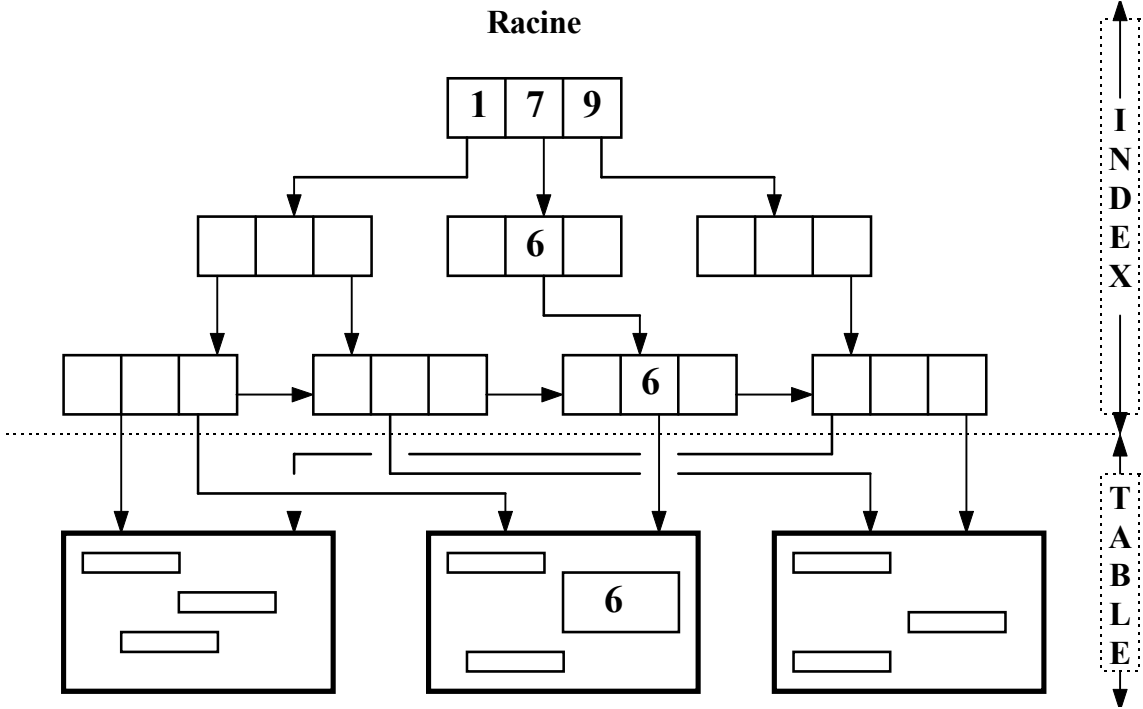
- ◆ En ORACLE, un index ne sert qu'à améliorer la performance des accès.
- ◆ L'index permet l'accès direct à l'information.
- ◆ Il est géré par l'administrateur de la base de donnée.


	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

```

CREATE INDEX index-name
ON table-name (column-name ASC, ...)
...

```



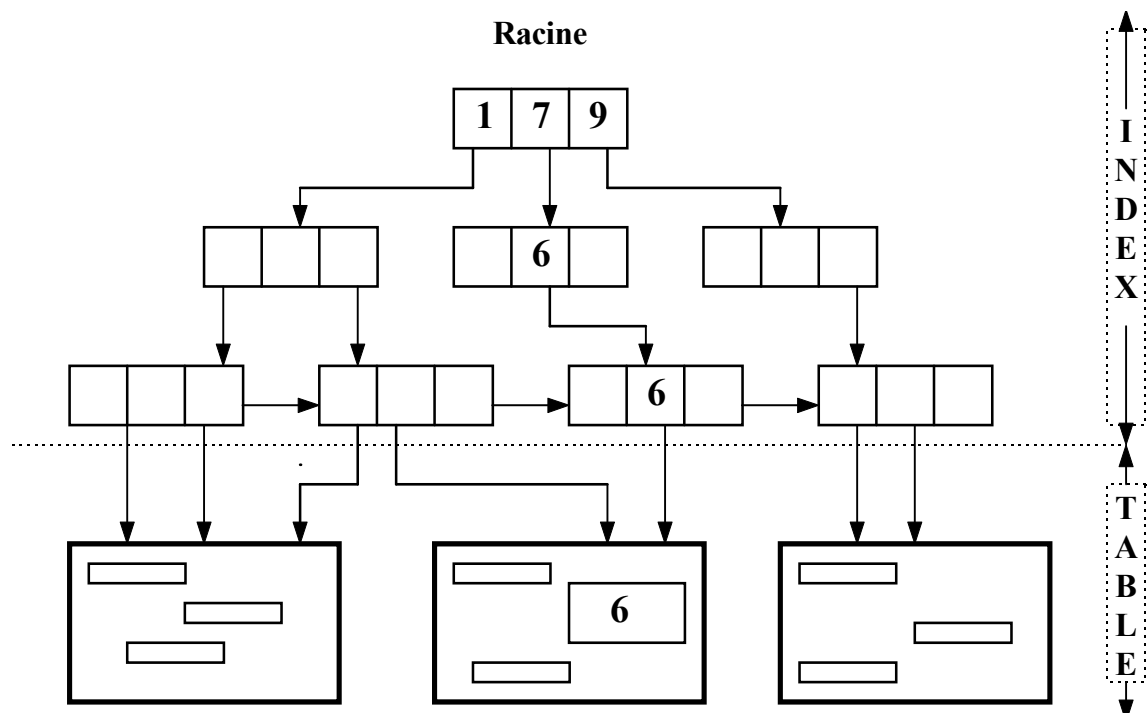
	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>


- ◆ L'INDEX UNIQUE est obligatoire si une clé primaire a été définie sur la table.

```
CREATE UNIQUE INDEX index-name
ON table-name (column-name ASC,
...)
...
```

- ◆ L'INDEX CLUSTER (un seul par table) assure un rangement physique des lignes dans l'ordre de cet index.

```
CREATE INDEX index-name
ON table-name (column-name ASC, ...)
CLUSTER
...
```



	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

6.2 Les Synonymes.


- ◆ Ils permettent de nommer une TABLE par un autre nom.

```

CREATE      SYNONYM      synonym
              FOR table-name

CREATE      SYNONYM      synonym
              FOR view-name

```

 PRODSET Groupe SORIA	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
	Objet : Cours de base ORACLE	Date : 27/10/2021

7 L'ordre SELECT

7.1 Généralités

- ◆ L'ordre SELECT est structuré par clause.

```
SELECT      Column-name, column-name, ...

            FROM ...

            WHERE ...

            GROUP BY ...

            HAVING ...


            ORDER BY ...
```

- ◆ Clauses obligatoires.

```
SELECT ... FROM ...
```

- ◆ Dépendances des clauses.

```
SELECT ... GROUP BY ...
```


	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

7.2 Définition des colonnes.

```

SELECT   Column-name, column-name, ...

        FROM ...

        WHERE ...

        GROUP BY ...

        HAVING ...

        ORDER BY ...

```

π La clause SELECT est suivie de la liste des colonnes que l'on veut extraire de la table, ces colonnes définissent les éléments de la TABLE résultat.

Exemple :


TABLCONC

DEPT	CODE	NOM	MARQUE	NBCLIENT
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	A.V.A.	FORD	150

```
SELECT NOM, NBCLIENT FROM TABLECONC
```

Résultat :

NOM	NBCLIENT
AUTOMOBILE BRESTOISE	520
CONCESSION PEUGEOT QUIMPER	325
LA LORIENTAISE DE L'AUTOMOBILE	150
VANNES EUROPE AUTOMOBILE	246
A.V.A.	150

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

π Un ou plusieurs éléments de cette liste peut être une manipulation de colonnes de la table origine.

π Les manipulations possibles sont :

- l'opérateur de concaténation : '||',

Il sert à regrouper deux chaînes de caractères en une seule.

Requête : SELECT NOM || ' SUR LE DEPARTEMENT ' ||
DEPT
FROM TABLCONC

Résultat :

AUTOMOBILE BRESTOISE SUR LE DEPARTEMENT 29
CONCESSION PEUGEOT QUIMPER SUR LE DEPARTEMENT 29
....

- les fonctions colonnes,

MIN renvoie le minimum de la colonne

MAX renvoie le maximum de la colonne

SUM renvoie la somme des valeurs


AVG renvoie la moyenne des valeurs de la colonne

COUNT renvoie le nombre de lignes sélectionnées

Requête : SELECT SUM(NBCLIENT)
FROM TABLCONC
WHERE DEPT = 29

Résultat :

SUM(NBCLIENT)
845

	<p align="center">PLAN de FORMATION</p>	<p align="center">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="center">Date : 27/10/2021</p>

- les fonctions scalaires,

Résultat sur chaque champ de la colonne

CHAR	transforme une DATE en une chaîne de caractères
DECIMAL	transforme un nombre en format décimal
DIGIT	transforme un nombre en sa chaîne de caractères
FLOAT	transforme un nombre en format flottant
HEX	chaîne de caractères représentant le champ en hexadécimal
INTEGER	retourne la valeur entière du nombre
LENGTH	longueur d'un champ quelconque
SUBSTR	extraction de chaîne de caractères
VALUE	substitution du champ NULL par deuxième argument

Combinaison possible de plusieurs fonctions.

Exemple : SUBSTR (DIGIT (. . .) , LENGTH (. . .) , 7)

- les fonctions de date et d'heure,

DATE
YEAR
MONTH
DAY
DAYS
HOUR
MINUTE
SECOND
MICROSECOND
TIME
TIMESTAMP


Plusieurs types d'argument possibles pour chaque fonction.

Combinaison possible de plusieurs fonctions.

Fonction :

TIME (TIMESTAMP ('1997-01-31-15.47.25.123456'))

Résultat :

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

' 15.47.25'

- les fonctions arithmétiques : +, -, *, /

Le résultat dépend des types de données en entrée. Il faut distinguer les calculs effectués avec : les ENTIERS, les DECIMAUX et les FLOTTANTS.

Exemple :

```
4 / 3                1
4.0 / 3             1.333333333
```

- les opérations sur les dates et heures : +, -

Exemples :

'1997-02-28' - '1996-11-25' Invalide

(soustraction de caractères)

DATE('1997-02-28') - '1996-11-25' VALIDE

ORACLE reconnaît une expression arithmétique sur les dates.

- une combinaison des manipulations précédentes.

π Le SELECT * FROM, permet la substitution du nom de toutes les colonnes de la TABLE :


- Diminution de la clarté du programme,
- Affecte les performances,

Exemple :

```
SELECT * FROM TABLCONC
```

est équivalent à

```
SELECT DEPT, CODE, NOM, MARQUE, NBCLIENT FROM TABLCONC
```

	<p align="center">PLAN de FORMATION</p>	<p align="center">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="center">Date : 27/10/2021</p>

π Le SELECT DISTINCT ... FROM, permet l'élimination des lignes en double de la table.

Exemple :

```
SELECT DEPT FROM TABLCONC
```

Résultat :

DEPT
29
29
56
56
85

```
SELECT DISTINCT DEPT FROM TABLCONC
```

Résultat :

DEPT
29
56
85

7.3 Définition clause FROM.

```

SELECT      Column-name, column-name, ...

           FROM ...


           WHERE ...

           GROUP BY ...

           HAVING ...

           ORDER BY ...

```

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

π Elle définit la TABLE ou les TABLES dans lesquelles se trouvent les colonnes spécifiées en *column-name* de la clause SELECT.

π La TABLE peut être une VUE.

π Les noms de TABLES ou de VUES sont séparés par des virgules.

π Si un nom de colonne est commun à deux tables, ce nom de colonne doit être préfixé par le nom de la TABLE ou par une chaîne de caractères associée à la TABLE.

Exemple :

Le Champ DEPT est défini dans la TABLE TABLCONC et dans la TABLE TABLEDEP.

```
SELECT TABLECONC.DEPT, NOM_DEPT
      FROM TABLCONC , TABLEDEP
      WHERE ...
```

ou

```
SELECT A.DEPT, NOM_DEPT
      FROM TABLCONC A, TABLEDEP
      WHERE ...
```

7.4 Définition clause *WHERE*.

```
SELECT      Column-name, column-name, ...


           FROM ...

           WHERE ...

           GROUP BY ...

           HAVING ...

           ORDER BY ...
```

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

π Elle définit les critères de sélections sur les lignes des TABLES de la clause FROM.

π Une ligne est sélectionnée si elle vérifie l'ensemble des critères de la clause WHERE.

π Un critère peut être associé avec d'autres critères par l'intermédiaire des opérations logiques **AND** et **OR**.

Exemple :

```
SELECT  DEPT, NOM, MARQUE, NBCLIENT
FROM TABLCONC
WHERE  NBCLIENT > 350
      AND DEPT = '29'
```


Résultat :

DEPT	NOM	MARQUE	NBCLIENT
29	AUTOMOBILE BRESTOISE	RENAULT	520

7.5 Clause WHERE et la structure des critères de sélection.

=	Egal
<>, ^=	Différent
<	Inférieur
<=, ^>	Inférieur ou égal
^<, >=	Supérieur ou égal
>	Supérieur
LIKE	Est similaire à
BETWEEN	Compris entre les valeurs
IN	Appartient à l'ensemble

π Tout critère peut être infirmé par l'opérateur logique NOT.

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

Exemple :

```
SELECT    DEPT, NOM, MARQUE, NBCLIENT
FROM TABLCONC
WHERE NOT DEPT IN ('29', '85')
      AND NBCLIENT > 200
```

Résultat :

DEPT	NOM	MARQUE	NBCLIENT
56	VANNES EUROPE AUTOMOBILE	RENAULT	246

π LIKE identifie une colonne à une chaîne de caractères, il peut être précédé de l'opérateur NOT.

% : masque de longueur quelconque,

_ : masque de 1 caractère.

Exemple :

```
SELECT    DEPT, NOM, MARQUE, NBCLIENT
FROM TABLCONC
WHERE NOM LIKE '%AUTO%'
      AND DEPT LIKE '5_'
```

Résultat :


DEPT	NOM	MARQUE	NBCLIENT
56	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	VANNES EUROPE AUTOMOBILE	RENAULT	246

π **BETWEEN ... AND ...** , vérifie si une expression est dans l'intervalle de deux valeurs calculées.

Exemple :

```
WHERE NBCLIENT BETWEEN 200 AND 300
```

est équivalent à

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODESET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

(NBCLIENT >= 200) AND (NBCLIENT <= 300)

π **IN** ... , vérifie si une expression calculée se trouve dans un ensemble de valeurs. L'ensemble peut-être :

- Une liste finie de valeurs
(. . . , . . . , . . . , . . . , . . .)
- Commande SELECT (SUBSELECT)

Exemple :

```


SELECT    DEPT , NOM , MARQUE , NBCLIENT
FROM      TABLCONC
WHERE     DEPT IN
          (
            SELECT DEPT
              FROM TABLEDEP
              WHERE HABITANT > 1
          )

```

Résultat :

DEPT	NOM	MARQUE	NBCLIENT
29	AUTOMOBILE BRESTOISE	RENAULT	520
29	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325

π Remarques : Dans un critère, une colonne est définie de la même façon que dans la clause SELECT.

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

7.6 Définition clause **GROUP BY**.

```

SELECT      Column-name, column-name, ...

            FROM ...

            WHERE ...

            GROUP BY ...

            HAVING ...

            ORDER BY ...

```

π Elle permet le regroupement des lignes en correspondance avec une fonction colonne (MIN, AVG, ...).

π Cette clause est suivie par une liste de colonnes pour spécifier des ruptures et des sous ruptures.

Exemple :


```

SELECT      DEPT, SUM (NBCLIENT) , AVG (NBCLIENT)
            FROM TABLCONC
            GROUP BY DEPT

```

Résultat :

DEPT	SUM(NBCLIENT)	AVG(NBCLIENT)
29	845	422,5
56	396	198
85	150	150

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

7.7 Définition clause *HAVING*.

```

SELECT      Column-name, column-name, ...

            FROM ...

            WHERE ...

            GROUP BY ...

            HAVING ...

            ORDER BY ...

```

π Le *HAVING* est à la table résultat ce que le *WHERE* est à la table initiale.

π L'ordre *HAVING* intervient après application du *GROUP BY*.

π Les critères du *HAVING* s'appliquent sur les colonnes de la clause *SELECT*.

Exemple :


```

SELECT      DEPT, SUM (NBCLIENT) , AVG (NBCLIENT)
            FROM TABLCONC
            GROUP BY DEPT
            HAVING SUM (NBCLIENT) > 200

```

Résultat :

DEPT	SUM(NBCLIENT)	AVG(NBCLIENT)
29	845	422,5
56	396	198

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

7.8 Définition clause ORDER BY.

```

SELECT      Column-name, column-name, ...

           FROM ...

           WHERE ...

           GROUP BY ...

           HAVING ...

           ORDER BY ...

```

π L'ORDER BY permet le tri des lignes de la TABLE sur les colonnes spécifiées.

π La colonne ou les colonnes spécifiées sont celles de la clause SELECT (Si pas de calcul).

π Le tri peut-être ASCendant ou DESCendant.

Exemple :


```

SELECT      DEPT, NOM, MARQUE, NBCLIENT
           FROM TABLCONC
           ORDER BY NBCLIENT DESC, DEPT ASC

```

Résultat :

DEPT	NOM	MARQUE	NBCLIENT
29	AUTOMOBILE BRESTOISE	RENAULT	520
29	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	VANNES EUROPE AUTOMOBILE	RENAULT	246
56	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
85	A.V.A.	FORD	150

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODESET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

7.9 Définition du **SUBSELECT**.

π Le **SUBSELECT** est un ordre **SELECT** imbriqué dans un autre ordre **SELECT**.

π Le **SUBSELECT** se positionne :


- Derrière un opérateur de comparaison,
- Derrière l'opérateur **IN**.

π Le **SUBSELECT** s'inclus dans une clause **WHERE** ou **HAVING**.

```

SELECT      ...
            FROM ...
            WHERE ... IN (SELECT ...
                          FROM ...
                          WHERE ... > (SELECT ...
                                       FROM ...
                                       WHERE ...
                                       )
                          )

```

 PRODSET Groupe SORIA	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
	Objet : Cours de base ORACLE	Date : 27/10/2021

7.10 Définition d'une jointure.

π Les colonnes de la TABLE résultante appartiennent à plusieurs TABLES.

π Mise en place d'une égalité de deux colonnes de tables différentes dans la clause WHERE.

π Le coût n'est pas à négliger à la formulation de la requête.


Exemple :

```

SELECT
      CONC.DEPT, CONC.NOM, DEP.HABITANT, CONC.NBCLIEN
T
      FROM TABLCONC CONC, TABLEDEP DEP
      WHERE CONC.DEPT = DEP.DEPT
      ORDER BY CONC.DEPT
  
```

Résultat :

DEPT	NOM	HABITANT	NBCLIENT
29	AUTOMOBILE BRESTOISE	1,2	520
29	CONCESSION PEUGEOT QUIMPER	1,2	325
56	LA LORIENTAISE DE L'AUTOMOBILE	0,8	150
56	VANNES EUROPE AUTOMOBILE	0,8	246
85	A.V.A.	0,4	150

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

8 Le langage de manipulation

8.1 L'ordre INSERT

π Commande SQL d'ajout de lignes selon deux formats possibles :

π Insertion d'une ligne

```
INSERT      INTO ...
           VALUES ( ... )
```

π Insertion de plusieurs lignes

```
INSERT      INTO ...
           SELECT ...
```


π La clause INTO spécifie la TABLE ou la VUE.

π La liste des colonnes peut-être spécifiée dans la clause INTO.

π La clause VALUE impose une valeur pour chaque champ.

π La clause SELECT ... doit spécifier le même type de champ que la TABLE de la clause INTO.

π L'INSERT est effectif au COMMIT.

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

Exemples :

```

INSERT INTO TABLECONC
VALUES ('44',
        '010'
        'FORD NANTES AUTOMOBILE',
        'FORD',
        0)

```

Résultat :

TABLCONC


DEPT	CODE	NOM	MARQUE	NBCLIENT
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	A.V.A.	FORD	150
44	010	FORD NANTES AUTOMOBILE	FORD	0

```

INSERT INTO TABLEDEP
SELECT *
FROM TABLDEP75
WHERE DEPT = '75')

```

DEPT	NOM DEPT	HABITANT
29	FINISTERE	1,2
22	COTES D'ARMOR	0,5
56	MORBIHAN	0,8
85	VENDEE	0,4
44	LOIRE-ATLANTIQUE	2,1
75	PARIS	7,8

	<p>PLAN de FORMATION</p>	<p>cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p>Objet : Cours de base ORACLE</p>	<p>Date : 27/10/2021</p>

8.2 L'ordre UPDATE

- ◆ Commande SQL de modification de lignes.

```

UPDATE      ...

            SET ...

            WHERE ...

```

π La clause WHERE est optionnelle, la mise à jour portera sur toute la table si celle-ci est omise.

π La mise à jour de la TABLE est effective au COMMIT.

π La clause UPDATE spécifie la TABLE ou la VUE sur laquelle porte l'ordre de mise à jour.


π La clause SET définit les colonnes mises à jour, la liste est partielle ou totale.

Syntaxe : SET colonne = expression,
colonne = expression,
...

L'expression est identique à l'expression de l'ordre SELECT.

π La clause WHERE définit les lignes.

Utilisation identique à la clause WHERE de l'ordre SELECT.

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

Exemples :

```
UPDATE TABLECONC
SET NOM = 'ANJOU VENDEE AUTOMOBILE'
WHERE DEPT = '85'
AND CODE = '001'
```

Résultat :

TABLCONC

DEPT	CODE	NOM	MARQUE	NBCLIENT
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	ANJOU VENDEE AUTOMOBILE	FORD	150
44	001	FORD NANTES AUTOMOBILE	FORD	0

8.3 L'ordre DELETE


π Commande SQL de suppression de lignes.

```
DELETE FROM ...
WHERE ...
```

π FROM spécifie la table concernée

π WHERE indique les lignes à supprimer, si la clause n'est pas spécifiée toutes les lignes de la table seront supprimées.

π La suppression des lignes de la TABLE est effective au COMMIT.

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

Exemples :

```
DELETE FROM TABLECONC
WHERE DEPT = '44'
AND CODE = '001'
```

Résultat :

TABLCONC

DEPT	CODE	NOM	MARQUE	NBCLIENT
29	001	AUTOMOBILE BRESTOISE	RENAULT	520
29	002	CONCESSION PEUGEOT QUIMPER	PEUGEOT	325
56	001	LA LORIENTAISE DE L'AUTOMOBILE	FIAT	150
56	002	VANNES EUROPE AUTOMOBILE	RENAULT	246
85	001	ANJOU VENDEE AUTOMOBILE	FORD	150


9 Le langage de contrôle

9.1 La commande GRANT.

π Tout accès à un objet ORACLE n'est possible que par un privilège accordé (par la commande GRANT).

π Plusieurs privilèges :

- DATABASE
- PLAN
- SYSTEM
- TABLE ou VUE
- USE privilège

	<p align="center">PLAN de FORMATION</p>	<p align="center">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="center">Date : 27/10/2021</p>

π Privilèges :

■ Nominatif

```
GRANT  SELECT
        ON TABLE TABLCONC
        TO XUSERX
```

■ Public

```
GRANT  SELECT
        ON TABLE TABLCONC
        TO PUBLIC
```

■ Avec option GRANT


```
GRANT  SELECT
        ON TABLE TABLCONC
        TO XUSERX
        WITH GRANT OPTION
```

9.2 La commande **REVOKE**.

π Tout privilège accordé peut être supprimé par la commande REVOKE.

π Plusieurs privilèges révocables :

- DATABASE
- PLAN
- SYSTEM
- TABLE ou VUE
- USE privilège

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

π Privilèges :

■ Nominatif

```

REVOKE  SELECT
        ON TABLE TABLCONC
        FROM YUSERY

```

■ Public


```

REVOKE  SELECT
        ON TABLE TABLCONC
        FROM PUBLIC

```

π Les privilèges disparaissent hiérarchiquement, la suppression d'un privilège sur un utilisateur qui avait donné lui-même des privilèges à d'autres utilisateurs supprime ces privilèges.

π Les privilèges sont supprimés sur la commande DROP.

	<p>PLAN de FORMATION</p>	<p>cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODESET Groupe SORIA</p>	<p>Objet : Cours de base ORACLE</p>	<p>Date : 27/10/2021</p>

10 Le langage de définition

10.1 La commande CREATE.

π Définition des objets ORACLE.


π Une commande par objet : DATABASE, STOGROUP, TABLE, TABLESPACE, INDEX, SYNONYM, ALIAS, VIEW.

π Les informations sont stockées au catalogue.

π Le SQL-id du créateur des objets définit le préfixe du nom des TABLES, VUES, ALIAS.

π La création des objets se fait hiérarchiquement.

1. La DATABASE et le STORAGE GROUP,
2. Les TABLESPACE,
3. Les TABLES,
4. Les INDEX et les VUES.

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>


3.2.1.1. Le Create TABLE.

```

CREATE TABLE Table name
(Column definition, column definition, ...,
[PRIMARY KEY (Column name, ...)],
[UNIQUE (Column name, ...)],
[FOREIGN KEY Constraint name (Column name, ...)
REFERENCES Table name [ON DELETE rules] ])
IN Database name.tablespace name ...

```

- ✓ **Primary key** : Identification des colonnes identifiant une clé primaire sur la table, la définition de la table est incomplète sans la création de l'index associé.
- ✓ **Unique** : Identification des colonnes identifiant une unicité sur la table, la définition de la table est incomplète sans la création de l'index associé.
- ✓ **Foreign key**: Identification des colonnes identifiant une clé étrangère, identification d'une relation avec une autre table et avec des contraintes associées.
- ✓ **ON DELETE rules** : Identification des règles de suppression entre la table parent et les tables dépendantes.
 - ON DELETE RESTRICT : Erreur sur la suppression d'une ligne de la table parent.
 - ON DELETE CASCADE : Suppression en cascade des lignes des tables dépendantes.
 - ON DELETE SET NULL : Mise à la valeur NULL des clés étrangères.

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

Exemple :

```

CREATE TABLE ETU.TW0DEPT
(DEPT          CHAR(2)      NOT NULL,
NOMDEPT       CHAR(20)     NOT NULL,
NBHABIT       DECIMAL(5,2) NOT NULL,
PRIMARY KEY (DEPT));

```

```

CREATE TABLE ETU.TW0CONC
(DEPT          CHAR(2)      NOT NULL,
CODECONC      CHAR(3)      NOT NULL,
NOMCONC       CHAR(20)     NOT NULL,
ILCONC        CHAR(20)     NOT NULL,
NBCLIENT      INTEGER,
PRIMARY KEY (DEPT, CODECONC),
FOREIGN KEY CONTI (DEPT) REFERENCES (ETU.TW0DEPT) );

```

10.2 Le Create VIEW.

```

CREATE VIEW   View name
                (Column name, column name, ...)
                AS Subselect

```

Exemple :

```

CREATE VIEW WEMP1
AS SELECT * FROM DSN8230.EMP WHERE SALARY < 35000.00;


```

10.3 Le Create INDEX.

```

CREATE [UNIQUE] INDEX Index name
                      ON Table name (Column name ASC/DESC, ...)
                      [USING STOGROUP Stogroup name
                        PRIQTY integer
                        SECQTY integer]

```


	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

```
[CLUSTER [PART integer VALUES (constant, ...)]]
```

Exemple : CREATE UNIQUE INDEX DSN8310.XDEPT1
 ON DSN8310.DEPT
 (DEPTNO ASC)
 USING STOGROUP DSN8G310 ;

10.4 La commande **DROP**.

π Suppression des objets ORACLE.

π Une commande par objet : DATABASE, STOGROUP, TABLE, TABLESPACE, INDEX, SYNONYM, ALIAS, VIEW.


π Un Drop TABLE

```
DROP    TABLE    Table name
```

π Un Drop INDEX

```
DROP    INDEX    Index name
```

Exemple : DROP TABLE DSN8230.DEPT;
 DROP TABLESPACE DSN8D23A.DSN8S23D;
 DROP VIEW DSN8230.VPROJRE1;

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

π Les informations sont enlevées du catalogue.

π La disparition des OBJETS se fait hiérarchiquement.

10.5 La commande ALTER.

π Modification des objets ORACLE.

π Une commande par objet : STOGROUP, TABLE, TABLESPACE, INDEX.

π Les informations sont stockées au catalogue. Cette modification peut nécessiter l'exécution d'utilitaires pour être pris en compte


(ALTER ... PRIQTY ...)

π Un ALTER TABLE.

```

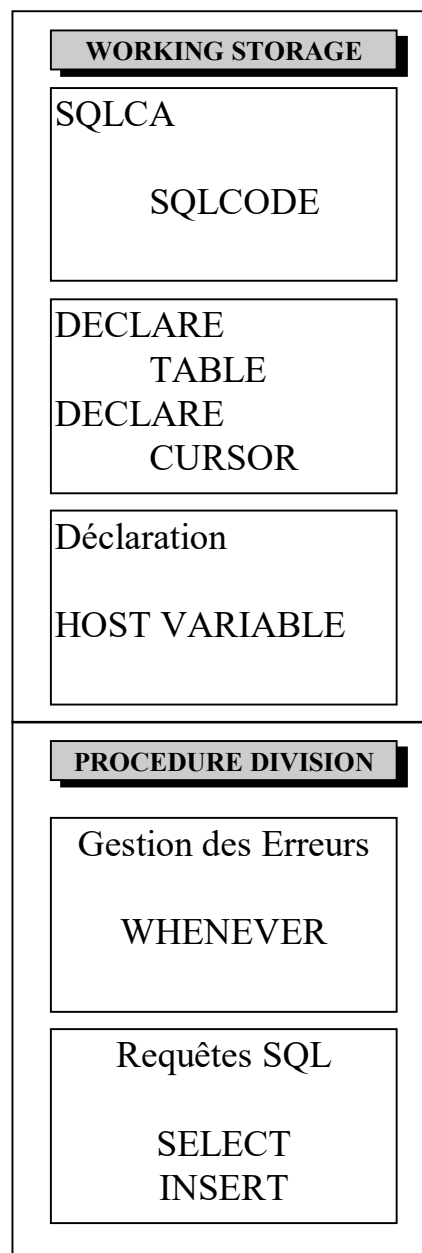
ALTER TABLE Table name
                [ADD Column definition]
                [ADD PRIMARY KEY (Column name, column name,
...)]
                [DROP PRIMARY KEY]
                [DROP FOREIGN KEY Constraint name ]


```

	<p>PLAN de FORMATION</p>	<p>cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p>Objet : Cours de base ORACLE</p>	<p>Date : 27/10/2021</p>

11 Annexe : notions d'appel de SQL dans un autre programme Pro*C, Pro*Cobol ...

11.1 Structure du programme appelant



	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

11.2 La zone de déclaration SQLCA.

π Zone de communication entre le programme et ORACLE.

π Apporte au programme des indications sur le déroulement de la requête SQL.
La zone SQLCODE contient le code retour à tester après un ordre EXEC SQL.

π L'insertion dans le programme se fait par un ordre :

```
EXEC SQL INCLUDE SQLCA END-EXEC.
```

11.3 La déclaration des objets.

π TABLE, VUE ou CURSEUR


```
...
EXEC SQL
      DECLARE TABLE ...
END-EXEC.
...
```

11.4 La gestion des erreurs.

π Le SQLCODE indique l'état de la requête SQL.

π L'ordre WHENEVER permet de tester :

- NOT FOUND (SQLCODE = 100)
- SQLERROR (SQLCODE < 0)
- SQLWARNING (SQLCODE > 0)

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

```

...
EXEC SQL WHENEVER NOTFOUND GOTO NONTROUVE END-
EXEC.
...
EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
...
EXEC SQL WHENEVER SQLERROR GO TO ERREUR END-EXEC.
...

```

11.5 La gestion des HOST VARIABLES.

π Une HOST VARIABLE est une zone mémoire qui contient la donnée récupérée par un ordre SELECT ou qui produit la valeur traitée par UPDATE, INSERT, DELETE.

π Une HOST VARIABLE est générée par le DCLGEN.


π Une HOST VARIABLE correspond à la définition de la colonne.

Exemple :

```

...
01 WPAYS      PIC X(30) VALUE 'FRANCE' .
...
EXEC SQL UPDATE TABLEDEP
SET PAYS = :WPAYS
WHERE PAYS = ' '
END-EXEC.
...

```

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

ORACLE	COBOL
CHAR(N)	PIC X(N)
DATE	PIC X(10)
TIME	PIC X(8)
TIMESTAMP	PIC X(26)
SMALLINT	PIC S9(4) COMP
INTEGER	PIC S9(8) COMP
DECIMAL(N,M)	PIC S9(N-M)V9(M) COMP-3
FLOAT simple	COMP-1
FLOAT double	COMP-2
VARCHAR(N)	01
LONG VARCHAR	49 PIC S9(4) COMP 49 PIC X(N)

11.6 Le **SELECT INTO**.

π Le **SELECT INTO** permet la recherche d'une ligne de la TABLE (ou VUE).
La sélection de plusieurs lignes doit se faire par un **CURSOR**.

Si le résultat du **SELECT** aboutit à renvoyer plusieurs lignes alors


SQLCODE = -811

Aucune lignes de disponible pour le traitement.

```

...
EXEC SQL  SELECT  ...
          INTO    <Host variables>
          FROM    ...
          WHERE   ...
END-EXEC.
...

```


	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

Exemple :

```

...
77 W-DEPT          PIC X(2) .
77 W-CODE          PIC X(3) .
01 LIG-TABCONC .
   02 DEPT         PIC X(2) .
   02 CODE         PIC X(3) .
   02 NOM          PIC X(40) .
   02 MARQUE       PIC X(10) .
   02 NBCLIENT    PIC S9(8) COMP .
...
MOVE '85'          TO W-DEPT .
MOVE '001'        TO W-CODE .
EXEC SQL
      SELECT  DEPT, CODE, NOM, MARQUE, NBCLIENT
      INTO    :LIG-TABCONC
      FROM    TABLCONC
      WHERE   DEPT = :W-DEPT
             AND CODE = :W-CODE
      END-EXEC .
IF SQLCODE < 0 ...
...

```

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

π Un INDICATEUR de HOST VARIABLES permet de tester la valeur NULL.

```

EXEC SQL
    SELECT  DEPT, CODE, NOM, MARQUE, NBCLIENT
          INTO :DEPT, :CODE, :NOM:NOMIND,
              :MARQUE, :NBCLIENT
          FROM TABLCONC
          WHERE DEPT = :W-DEPT
              AND CODE = :W-CODE
          END-EXEC.
IF SQLCODE < 0 ...
...

```

Indicateur de
la variable
NOM

Si NOMIND < 0 **alors** NOM est NULL.

Si NOMIND > 0 **alors** NOM est tronqué.

Si NOMIND = 0 **alors** NOM est correct.

11.7 L'ordre INSERT.


```

...
EXEC SQL  INSERT INTO table-name
          (column-name, column-name, ...)
          VALUES (host-variable, host-variable, ...)
          END-EXEC.
...

```

π Une host-variable correspond à une valeur à affecter à la column-name correspondante. Toutes les colonnes doivent être affectées d'une valeur. Si une colonne restante n'est pas affectée d'une valeur elle prendra une valeur par défaut, attention si la colonne est définie NOT NULL.

π INSERT de plusieurs lignes (MASSINSERT)

	<p>PLAN de FORMATION</p>	<p>cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p>Objet : Cours de base ORACLE</p>	<p>Date : 27/10/2021</p>

π INDICATEUR VARIABLE pour spécifier NULL sur une colonne spécifiée.

π Effectif au COMMIT

Exemple :

```

...
EXEC SQL INSERT INTO TABLCONC
      (DEPT, CODE, NOM, MARQUE, NBCLIENT)
      VALUES (:DEPT, :CODE, :NOM, :MARQUE, :NBCLIENT)
END-EXEC.
...

```

L'ordre UPDATE.


```

...
EXEC SQL UPDATE table-name
      SET column-name = host-variable,
          column-name = host-variable
          ...
      WHERE ...
END-EXEC.
...

```

π Mise à jour des lignes de la TABLE sélectionnées par la clause WHERE.

π Effectif au COMMIT.

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

11.8 L'ordre *DELETE*.

```

...
EXEC SQL DELETE
        FROM table-name
        WHERE ...
END-EXEC.
...

```


π Suppression des lignes de la TABLE sélectionnées par la clause WHERE.

π Effectif au COMMIT

11.9 Le *SELECT* de plusieurs lignes (*CURSEURS*).

π Le Curseur permet de traiter une sélection de plusieurs lignes en séquentielle avec les ordres suivants :

- DECLARE CURSOR
- OPEN
- FETCH
- CLOSE

	<p>PLAN de FORMATION</p>	<p>cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p>PRODSET Groupe SORIA</p>	<p>Objet : Cours de base ORACLE</p>	<p>Date : 27/10/2021</p>

11.10 La déclaration du CURSOR

π Dans un programme on pourra trouver plusieurs CURSOR et donc une déclaration pour chacun.

```


...
EXEC SQL DECLARE cursor-name CURSOR
      [WITH HOLD]
      FOR SELECT ...
      [FOR FETCH ONLY]
      [FOR UPDATE OF column-name,column-name, ...]
END-EXEC.
...

```

π **WITH HOLD** : permet de conserver la position du curseur après un COMMIT.

π **FOR UPDATE OF ...** : indique une intention de mise à jour ou de suppression par un ordre SQL.

π **FOR FETCH ONLY** : Pas d'ordre UPDATE ou DELETE référençant le cursor.

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

π Lecture uniquement si dans le SELECT du DECLARE CURSOR apparaissent les ordres suivants :

- DISTINCT
- GROUP BY
- HAVING
- FONCTIONS ou EXPRESSIONS
- JOINTURE
- UNION de SELECT
- SUBSELECT
- ORDER BY

(FOR FETCH ONLY est implicite dans ce cas)

11.11 Le traitement (**OPEN, FETCH, CLOSE**)

```

...
EXEC SQL DECLARE cursor-name CURSOR
...
EXEC SQL OPEN      cursor-name
EXEC SQL FETCH ... INTO ...
TANTQUE SQLCODE = 0
    Traiter la ligne
    ...
    EXEC SQL FETCH ... INTO ...
    ...
FINTANTQUE
EXEC SQL CLOSE   cursor-name
...


```

π C'est lors de l'OPEN CURSOR que le SELECT est exécuté.

```

...
EXEC SQL OPEN   cursor-name END-EXEC.
...

```

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

π l'ordre FETCH permet d'accéder à une ligne

```

...
EXEC SQL  FETCH cursor-name
           INTO  host-variables
END-EXEC.
...


```

π Par défaut le COMMIT implique le CLOSE du curseur.

```

...
EXEC SQL  CLOSE cursor-name END-EXEC.
...

```

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

Exemple :

```


01 LIG_TABLCONC
02 DEPT    PIC X(2) .
02 NOM     PIC X(40) .
02 MARQUE  PIC X(15) .
...
EXEC SQL  DECLARE  BRETAGNE  CURSOR
          FOR SELECT DEPT,NOM,MARQUE
             FROM TABLCONC
             WHERE DEPT IN ('29','56',...)

END-EXEC.
...
EXEC SQL  OPEN    BRETAGNE
END-EXEC.
...
EXEC SQL  FETCH  BRETAGNE
          INTO  :DEPT, :NOM, :MARQUE
END-EXEC.

TANTQUE  SQLCODE = 0
    Traiter la ligne
    ...
    EXEC SQL    FETCH  BRETAGNE
              INTO  :DEPT, :NOM, :MARQUE
    END-EXEC.
...
FINTANTQUE

EXEC SQL  CLOSE  BRETAGNE
...

```

	<p style="text-align: center;">PLAN de FORMATION</p>	<p style="text-align: right;">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p style="text-align: center;">PRODSET Groupe SORIA</p>	<p style="text-align: center;">Objet : Cours de base ORACLE</p>	<p style="text-align: right;">Date : 27/10/2021</p>

11.12 La mise à jour avec CURSOR.

π la déclaration du curseur (FOR UPDATE OF ...).

```

...
EXEC SQL DECLARE cursor-name CURSOR
FOR SELECT ...
FOR UPDATE OF column-name,column-name,...
END-EXEC.
...

```

π la mise à jour par UPDATE.

```

...
EXEC SQL UPDATE table-name
SET ...
WHERE CURRENT OF cursor-name
END-EXEC.
...


```

π la suppression par DELETE.

```

...
EXEC SQL DELETE FROM table-name
WHERE CURRENT OF cursor-name
END-EXEC.
...

```

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

12 Notion de PL/SQL

Le PL/SQL est une extension structurée L3G du SQL (proche d'un L4G), pour des traitements batch sans passer par un programme C, COBOL etc. ... (Il évite aussi les ordres OPEN ou CLOSE de fichier).

Ce langage est interprété, sauf dans le cas où les programmes PL/SQL sont pré-compilés sous la forme de procédures stockées dans la base.

Les règles de déclaration de variables, de curseurs, d'appels de curseurs, de programmation etc ... sont les même qu'au chapitre 11 précédent.

12.1 Structure d'un programme PL/SQL

DECLARE

Déclarations de variables et/ou de curseurs ;

BEGIN

Instructions SQL ou PL/SQL ;

EXCEPTION

Instructions SQL ou PL/SQL à exécuter en cas d'erreur ;


END

12.2 Exemples de programmes PL/SQL

```

DECLARE
    dummy char(1);
BEGIN
    select 'X' into dummy
    from vcv.vcv_coloris
    where cod_coloris = :block.cod_coloris;
EXCEPTION
    when no_data_found then
        message('.....');
        raise form_trigger_failure;
    when others then
        message('Error '||to_char(sqlcode)||' <OVF> Contact DBA');
        raise form_trigger_failure;

```


	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

```

* Si SCALE -> 0 -> format '999.9999' dans lequel la partie *
* avant le point decimal a une longueur de *
* DATA_PRECISION-DATA_SCALE et la partie *
* aprs le point d,imal a une longueur de *
* DATA_SCALE *
* R,alis, avec un DECODE bas, sur la valeur de DATA_SCALE *
*-----*/
UPDATE VCV.VCV_TEMPOVALCOL A
SET A.STR_FORMAT =
(SELECT DECODE(B.data_scale,
              0,RPAD('9',B.data_precision,'9'),
              RPAD('9',B.data_precision-B.data_scale,'9')
              ||'.'||
              RPAD('9',B.data_scale,'9'))
 FROM ALL_TAB_COLUMNS B
 WHERE A.cod_table = B.table_name
       AND A.cod_col = B.column_name
       AND B.owner = 'VCV')
 WHERE A.COD_TYPE = 'N'
       AND A.cod_table = :tablename1.cod_table;
/*-----*/
* UPDATE de la colonne STR_FORMAT si c'est une colonne date *
* Toujours le format A8, parce que la fonction *
* TO_CHAR(---,'DD/MM/YY') sera mise dans le SELECT-Statement *
*-----*/
UPDATE VCV.VCV_TEMPOVALCOL A
SET A.STR_FORMAT = 'A8'
WHERE A.COD_TYPE = 'D'
      AND A.cod_table = :tablename1.cod_table;
/*-----*/
* Les colonnes de type 'LONG' seront format, par la commande *
* SET LONG = 80 de SQL*Plus *
* *
* Les colonnes de type 'CHAR' seront format, avec le format *
* A##, selon la valeur se trouvant dans la colonne FORMAT-MODEL *
* de la table VCV_TEMPOSELCOL *
*-----*/
UPDATE VCV.VCV_TEMPOVALCOL
SET STR_FORMAT = 'A'||STR_FORMAT
WHERE COD_TYPE = 'C'
      AND cod_table = :tablename1.cod_table;
END;
END;

```


13 Annexe : Exemple de script SQL de création de table

```

rem =====
rem Table : T_ACT_R
rem =====
DROP TABLE T_ACT_R;

CREATE TABLE T_ACT_R
(
  CO_APE                CHAR(4)                NOT NULL,
  LI_APE                CHAR(40)               NOT NULL,
  PRIMARY KEY (CO_APE) CONSTRAINT PK_T_ACT_R
)

```

	<p align="center">PLAN de FORMATION</p>	<p align="right">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="right">Date : 27/10/2021</p>

```
);

COMMENT ON TABLE T_ACT_R IS 'Activite des entreprises (ACT)';

COMMENT ON COLUMN T_ACT_R.CO_APE IS 'Code A.P.E';
COMMENT ON COLUMN T_ACT_R.LI_APE IS 'Libelle';

rem =====
rem Index : T_ACT_R_PK
rem =====
DROP INDEX T_ACT_R_PK;

CREATE UNIQUE INDEX T_ACT_R_PK ON T_ACT_R (CO_APE ASC);
```

14 Annexe : Exemple de script SQL de création de synonymes

```
SET DOC OFF
/*
*-----
* Module.....: dropsyn.sql
* Domaine.....: DBA
* Auteur.....: ATT-DATAID
* Description g,n,rale.: suppress des synonymes des tables pour tous les usagers
* Date de cr,ation.....: <19/01/94>
* Version.....: <01>
* R,visions.....:
*   Date.....: <>
*   Auteur.....: <ffe>
*   Descriptif.....:
*   R,f,rence.....: SYO/PRO/FFA/CPT/SFD/008 du 06/01/94
*-----
*/
REM Note : script ... lancer par le DBA ("SYSTEM") de l'application FFA
DEFINE nom = OPS$FFA ;

SET HEAD OFF ;
SET VER OFF ;
SET PAGES 9999;
SET FEED OFF ;

SPOOL drop_syn2.sql


SELECT 'DROP PUBLIC SYNONYM OPS$FFA..' || TABLE_NAME || ' ' ;'
FROM ACCESSIBLE_TABLES
WHERE OWNER = 'OPS$FFA';

SPOOL OFF

HOST vi drop_syn2.sql
START drop_syn2.sql

HOST rm drop_syn2.sql

SET DOC OFF
```

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

```

/*
-----
* Module.....: cresyn.sql
* Domaine.....: DBA
* Auteur.....: ATT-DATAID
* Description g,n,rale.: cr,ation des synonymes des tables pour tous les usagers
* Date de cr,ation.....: <19/01/94>
* Version.....: <01>
* R,visions.....:
*   Date.....: <>
*   Auteur.....:
*   Descriptif.....:
*   R,f,rence.....: SYO/PRO/FFA/CPT/SFD/008 du 06/01/94
*   Note          : script ... lancer par le DBA ("SYSTEM") de l'application
-----
*/

DEFINE nom = OPS$FFA ;

SET HEAD OFF
SET VER OFF
SET PAGES 9999
SET FEED OFF

SET TERM OFF
SPOOL grant_syn2.sql

SELECT 'CREATE PUBLIC SYNONYM '||TABLE_NAME||' FOR &nom..'||TABLE_NAME||' ';
FROM ACCESSIBLE_TABLES
WHERE OWNER = '&nom';

SPOOL OFF
SET TERM ON

ACCEPT ok CHAR PROMPT 'visualisation sous "vi" pour verification, taper "Return"'
PROMPT =====

HOST vi grant_syn2.sql

PROMPT creation des synonymes des tables pour tous les usagers
PROMPT =====

SET ECHO ON

START grant_syn2.sql

SET ECHO OFF

HOST rm grant_syn2.sql


```

15 Annexe : Exemple de scripts ou d'ordres SQL d'insertions de données dans une table

```

INSERT INTO T_MESS VALUES ('M001','Ce numero de DAEDO n'existe pas');
INSERT INTO T_MESS VALUES ('M002','Ce numero de DAP n'existe pas');

```

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

```

INSERT INTO T_MESS VALUES ('M003','Ce numero de CIB n'existe pas');
INSERT INTO T_MESS VALUES ('M004','Ce code programme n'existe pas');
INSERT INTO T_MESS VALUES ('M005','Ce code sous programme n'existe pas');
INSERT INTO T_MESS VALUES ('M006','Ce numero de SDAEDO n'existe pas');
INSERT INTO T_MESS VALUES ('M007','Ce numero de SDAP n'existe pas');

```

```


/*-----*
| Modifications :
| Auteur : B.LISAN   Date : 21/12/92
| Raison : remplissage de la table VCV_TEMPOVALCOL.
| But      : pour rendre plus convial l'utilisation de l',cran de selection des
| colonnes propos,es (VCVTVM.inp)
|-----*
| * 3) PRE-INITIALISATION DE LA TABLE VCV_TEMPOVALCOL :
|
| On inserre les noms des tables et des colonnes
| de VCV dans la table VCV_TEMPOVALCOL.
| On n'inserre ces colonnes que si elles ne sont pas encore
| pr,sents dans cette dernišre table.
| si elles ne sont pas utilis,es dans une Foreign key
| le
|
|         AND EXISTS (SELECT 'x'
|                       FROM ALL_OBJECTS
|                       WHERE object_name = table_name
|                       AND object_type = 'TABLE')
| est utilis, pour ,viter de s,lectionner des nom des vues dans la liste des
| noms de tables s,lectionn,es par all_tab_columns
|-----*/

```

```

INSERT INTO vcv.vcv_tempovalcol
(cod_table,cod_col,lib_col,COD_TYPE,STR_FORMAT,chx_sel)
SELECT table_name,
       column_name,
/*-----*
| OBTENTION DU LIBELLE :
| remplacement des 3 ler caractšres du nom de la colonne et
| du 4iŠme caractšre '_' par un blanc pour obtenir le libell,
| (et mise en majuscule du 1er caractšre de chaque mot du lib.)
| On remplace NUM_ par ' '
| "      COD_ par 'Code '
| "      ... par ...
| "      BEX par Bon d'Examen
|-----*
REPLACE(
INITCAP(
DECODE(SUBSTR(column_name,1,4), 'NUM_', '',
                                     'COD_', 'Code ',
                                     'LIB_', 'Libell, de ',
                                     'TPS_', 'Date de ',
                                     'SUB_', 'Sous-',
                                     'CHX_', 'Choix ',
                                     'STR_', ''
                                     substr(column_name,1,3) || ' ')
|| LOWER(REPLACE(REPLACE(SUBSTR(column_name,5),
                           'COMMENT', 'COMMENTAIRE '), 'BEX', 'BON D'EXAMEN '))
), ' De ', ' de '
),
/*-----*
| TRADUCTION DU TYPE (de data_type en cod_type)
|-----*/

```

	PLAN de FORMATION	cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN
PRODSET Groupe SORIA	Objet : Cours de base ORACLE	Date : 27/10/2021

```

DECODE(data_type,'CHAR' , 'C',
          'NUMBER', 'N',
          'LONG'   , 'L',
          'DATE'   , 'D'),
/*-----*
| CALCUL DU FORMAT :
| But : donne les formats par defaut des ordres SQLPLUS "COL"
|       de formatage des colonnes du listing "LISTE DYNAMIQUE"
| Si DATE      :-> FORMAT = 'A8'
| Si CHAR ou LONG :-> FORMAT = 'A' || data_length
| Si NUMBER    :
|     Si SCALE = 0 -> FORMAT = '9...9' avec
|                               nbre de '9' = data_precision
|     Si SCALE > 0 -> FORMAT = '9""9.9~~9' avec
|                               '9' = partie entière dont le nbre de
|                               '9' = data_precision - data_scale
|                               '.' = point d,cimal,
|                               '9~~9' = partie d,cimale dont nbre de
|                               '9' = data_scale
|
| Exemple:
| Si NUM_SPECIMEN d,clar, NUMBER(5,0) alors
|     data_precision = 5, data_scale = 0 et FORMAT = '99999'
|-----*/
DECODE(data_type,'CHAR' , 'A' || data_length,
          'LONG'   , 'A' || data_length,
          'NUMBER', DECODE(data_scale,
                           0,RPAD('9',data_precision,'9')
                           ,RPAD('9',data_precision -
                                data_scale , '9')
                           || '.' ||
                           RPAD('9',data_scale , '9')),
          'DATE' , 'A8'),
'0'
FROM all_tab_columns
WHERE owner = 'VCV'
AND table_name NOT LIKE 'VCV_TEMPO%'
AND NOT EXISTS (SELECT 'x'
                 FROM vcv.vcv_tempofkey,
                 vcv.vcv_tempoconscol
                 WHERE cod_detail_table = 'VCV.' || table_name
                   AND cod_col = column_name
                   AND cod_foreign_key = cod_constraint)
AND EXISTS (SELECT 'x'
            FROM ALL_OBJECTS
            WHERE object_name = table_name
              AND object_type = 'TABLE')
;


```

16 Annexe : Exemple de scripts Sql de gestion de droits

```

SET DOC OFF
/*
*-----*
* Module.....: grant_tabs.sql
* Domaine.....: DBA
* Auteur.....: ATT-DATAID
* Description generale.: creation droits d'accès DML,connection ... ORACLE pour

```


	<p align="center">PLAN de FORMATION</p>	<p align="center">cb1e-7979-496b-a4e7 Auteur : Benjamin LISAN</p>
<p align="center">PRODSET Groupe SORIA</p>	<p align="center">Objet : Cours de base ORACLE</p>	<p align="center">Date : 27/10/2021</p>

```

* Domaine.....: DBA
* Auteur.....: ATT-DATAID
* Description generale.: creation droits d'accès aux tables pour les usagers
* Date de creation.....: <19/01/94>
* Version.....: <01>
* R,visions.....:
*   Date.....: <>
*   Auteur.....: <ffe>
*   Descriptif.....:
*   Reference.....: SYO/PRO/FFA/CPT/SFD/008 du 06/01/94
*-----
*/
REM Note : script ... lancer par le OWNER (Proprietaire) de l'application FFA
REM      (OPS$FFA)

SET HEADING    OFF;
SET VERIFY    OFF;
SET PAGESIZE  9999;
SET FEEDBACK  OFF;

ACCEPT nom CHAR PROMPT 'Usager dont on accorde des droit sur les tables FFA : '

SPOOL grant_tab2.sql

SELECT 'GRANT SELECT,DELETE,INSERT,UPDATE ON '||TABLE_NAME||' TO OPS$&nom ;'
FROM   USER_TABLES;

REM WHERE  OWNER = 'OPS$FFA';

SPOOL OFF;

HOST vi grant_tab2.sql

START grant_tab2.sql

HOST rm grant_tab2.sql

```