

## Chapitre III

### Administration des BD SQL

## 3 - Administration de BD SQL

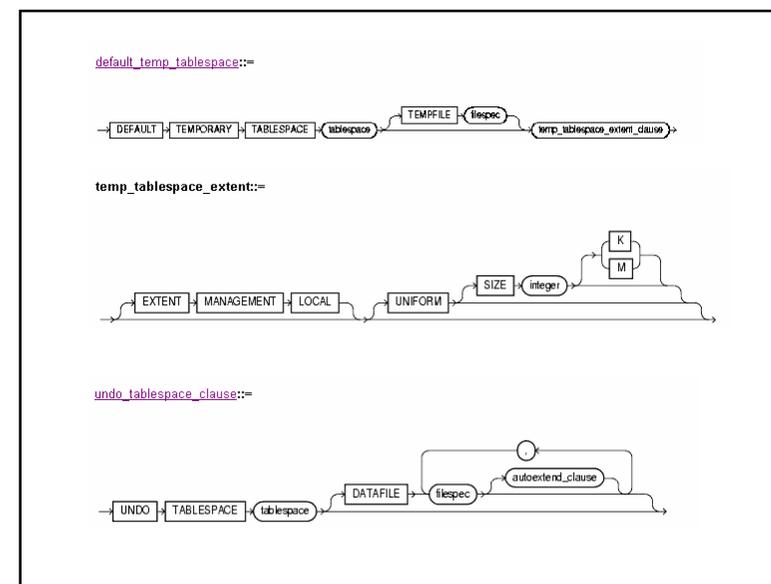
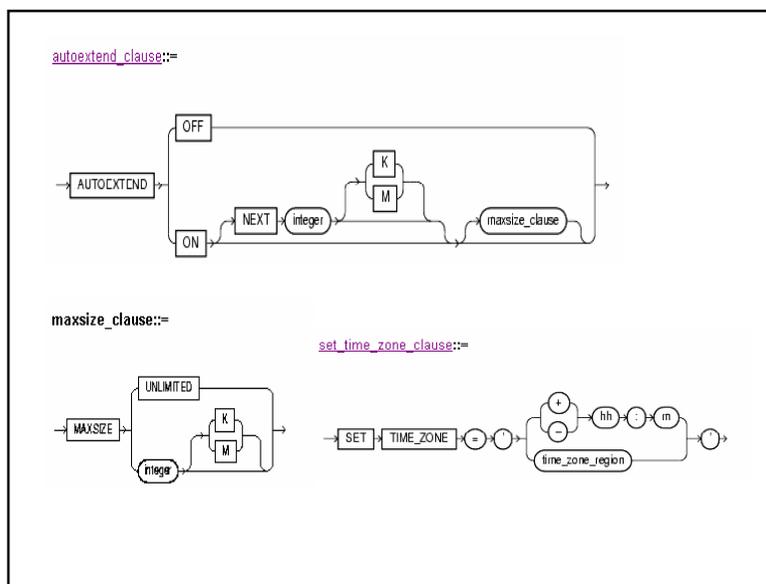
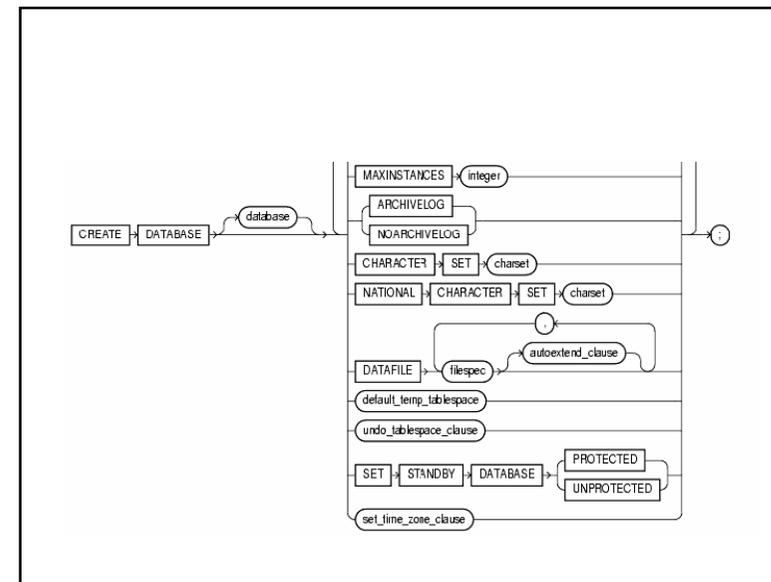
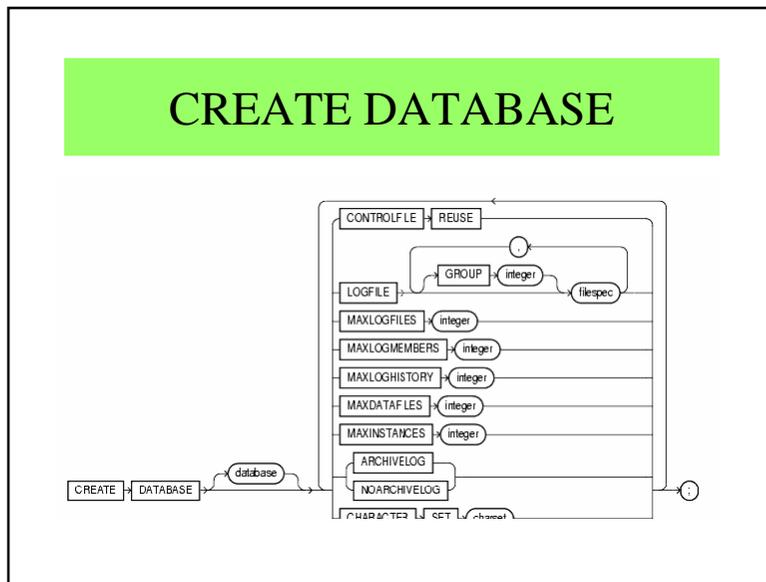
- 3.1 Création de BD
- 3.2 Gestion des utilisateurs
- 3.3 Administration des transactions
- 3.4 Accès à une BD distante
- 3.5 Conclusions

## Composants ORACLE

- Base de données
- Tables
- Index
- Cluster
- Tablespace
- Schéma
- Séquence
- Procédure
- Fonction
- Package
- Profil
- View
- Répertoire
- Trigger
- Bibliothèque
- Utilisateur
- Synonyme
- etc.

## Actions possibles

- CREATE
- DROP
- ALTER



## Exemple

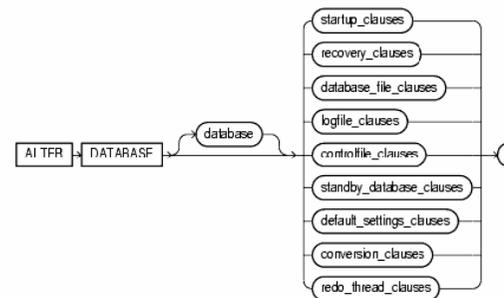
```

CREATE DATABASE sample

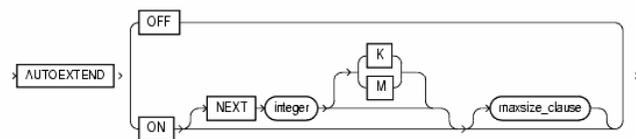
CONTROLFILE REUSE
LOGFILE
  GROUP 1 ('diskx:log1.log', 'disky:log1.log') SIZE 50K,
  GROUP 2 ('diskx:log2.log', 'disky:log2.log') SIZE 50K
MAXLOGFILES 5
MAXLOGHISTORY 100
MAXDATAFILES 10
MAXINSTANCES 2
ARCHIVELOG
CHARACTER SET UTF8
NATIONAL CHARACTER SET AL16UTF16
DATAFILE
  'disk1:df1.dbf' AUTOEXTEND ON,
  'disk2:df2.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
DEFAULT TEMPORARY TABLESPACE temp_ts
UNDO TABLESPACE undo_ts
SET TIME_ZONE = '+02:00';
    
```

## ALTER DATABASE

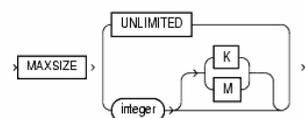
alter\_database::=



autoextend\_clause::=

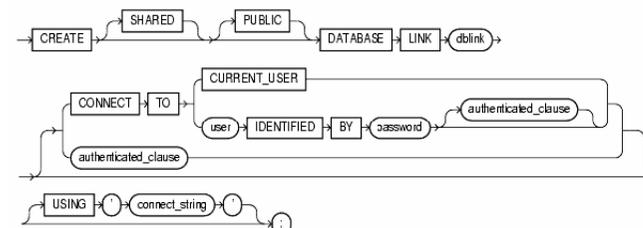


maxsize\_clause::=



## CREATE DATABASE LINK

create\_database\_link::=



authenticated\_clause::=



## Exemples

```
CREATE DATABASE LINK sales.hq.acme.com
CONNECT TO CURRENT_USER
USING 'sales';
```

```
CREATE DATABASE LINK sales.hq.acme.com SELECT * FROM employees@sales.hq.acme.com;
CONNECT TO hr IDENTIFIED BY hr
USING 'sales';
```

```
INSERT INTO orders@sales.hq.acme.com
(customer_id, order_id, order_total)
VALUES (5001, 1235, 2000);
```

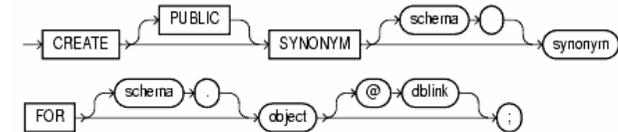
```
UPDATE orders@sales.hq.acme.com
SET order_total = order_total + 500;
```

```
DELETE FROM order_id@sales.hq.acme.com
WHERE order_id = 2443;
```

Travail sur  
une BD à distance

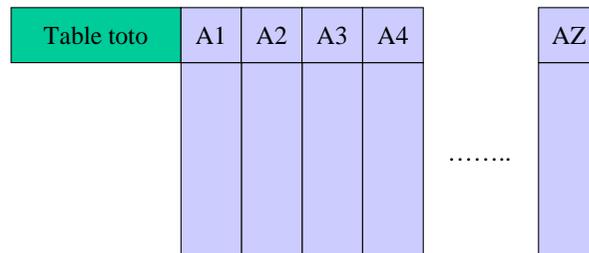
## CREATE SYNONYM

`create_synonym::=`



```
CREATE SYNONYM dept
FOR hr.departments@sales.hq.acme.com;
```

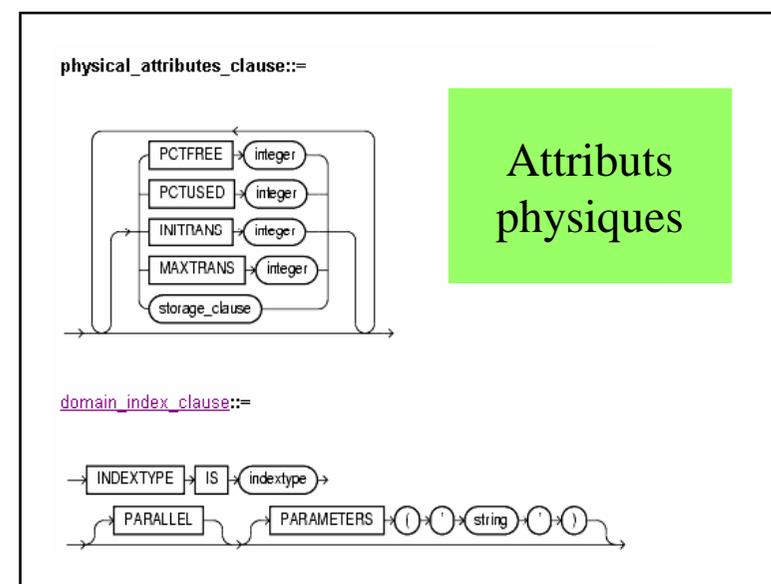
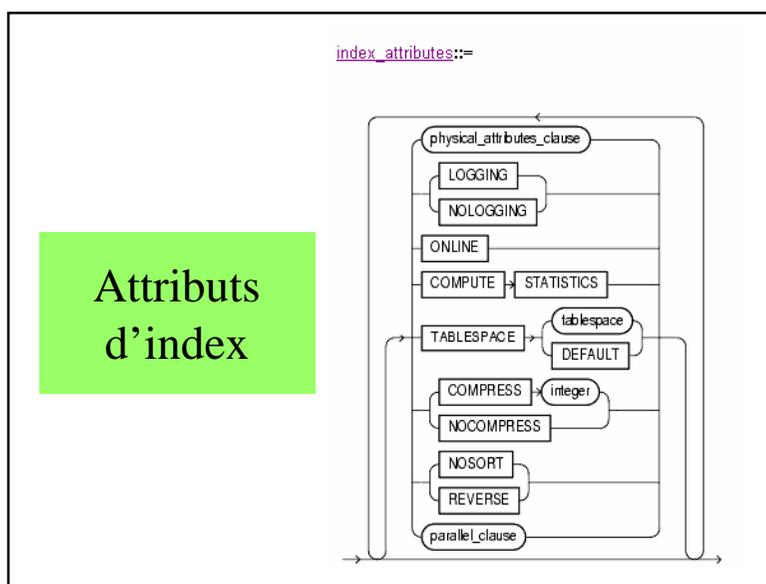
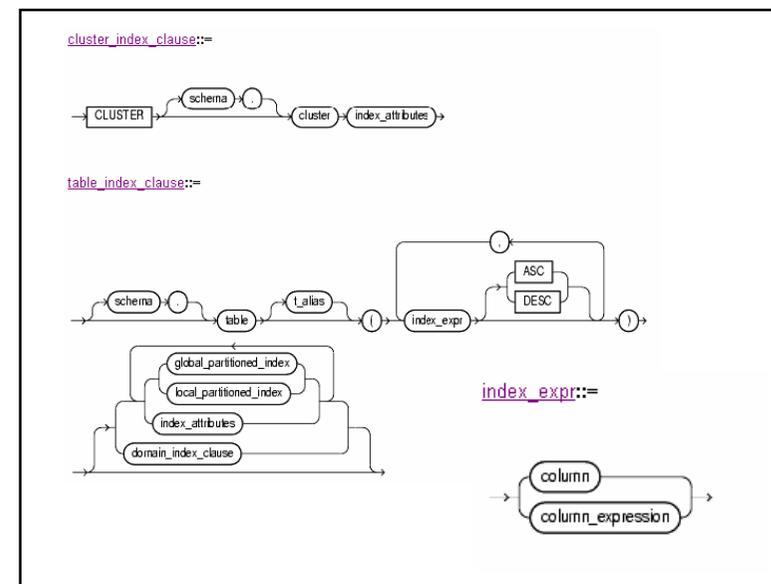
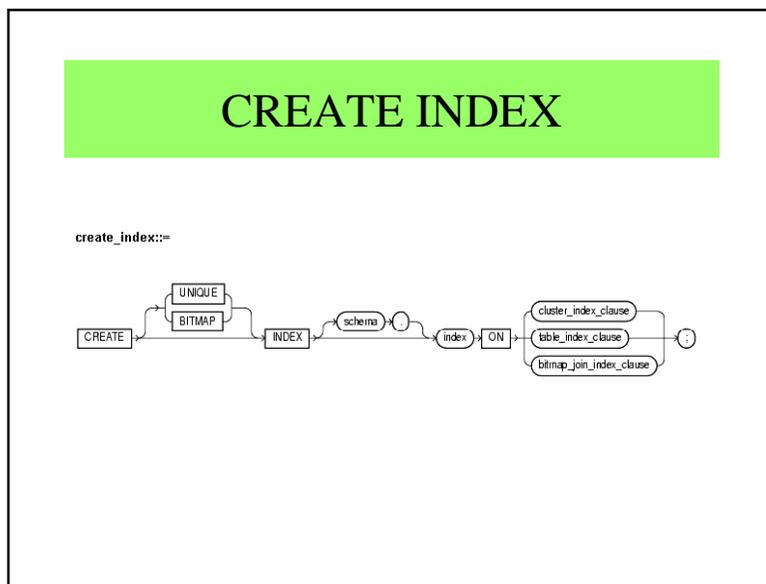
## INDEX



Possibilités :  
 index sur chaque attribut isolément  
 index sur des groupes : (A3,A1)  
 index sur des fonctions d'attributs : f(A2)

## INDEX

- Index : structure d'accélération aux accès
- Index normaux : B-tree
- Index de type bitmap : rowids et valeurs binaires de clés
- Index partitionnés
- Index d'expression (built-in / user defined)



## Exemple de création d'index

```
CREATE INDEX ord_customer_ix
ON orders (customer_id);
```

```
CREATE INDEX ord_customer_ix ON orders (customer_id, sales_rep_id)
COMPRESS 1;
```

```
CREATE INDEX ord_customer_ix ON orders(customer_id, sales_rep_id)
COMPUTE STATISTICS;
```

```
CREATE INDEX ord_customer_ix
ON orders (customer_id)
NOSORT
NOLOGGING
PARALLEL;
```

```
CREATE INDEX personnel_ix ON CLUSTER personnel;
```

```
CREATE INDEX upper_ix ON employees (UPPER(last_name));
```

```
SELECT first_name, last_name
FROM employees WHERE UPPER(last_name) IS NOT NULL
ORDER BY UPPER(last_name);
```

```
CREATE INDEX income_ix
ON employees(salary + (salary*commission_pct));
```

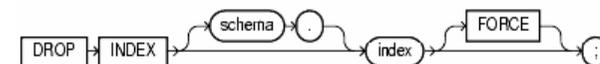
```
SELECT first_name||' '||last_name "Name"
FROM employees
WHERE (salary*commission_pct) + salary > 15000;
```

## Index partitionnés

```
CREATE INDEX cost_ix ON sales (amount_sold)
GLOBAL PARTITION BY RANGE (amount_sold)
(PARTITION p1 VALUES LESS THAN (1000),
PARTITION p2 VALUES LESS THAN (2500),
PARTITION p3 VALUES LESS THAN (MAXVALUE));
```

## DROP INDEX

**drop\_index::=**

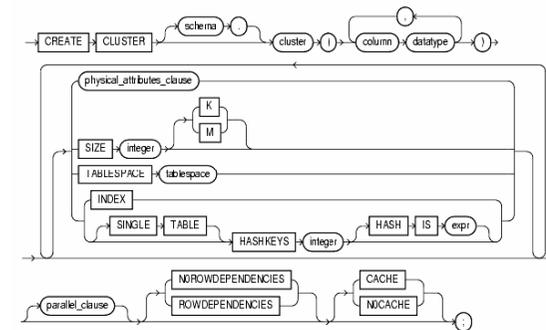


## CLUSTER

- Un cluster regroupe des données provenant de plusieurs tables ayant une ou plusieurs colonnes en commun
- Accélération des jointures

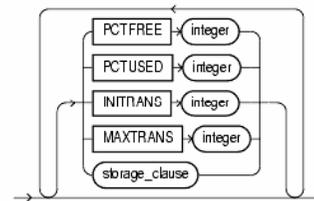
## CREATE CLUSTER

create\_cluster::=

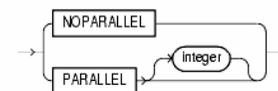


## Attributs physiques

physical\_attributes\_clause::=



parallel\_clause::=



## Exemple de création de cluster

```
CREATE CLUSTER personnel
  (department NUMBER(4))
  SIZE 512
  STORAGE (initial 100K next 50K);

CREATE INDEX idx_personnel ON CLUSTER personnel;

CREATE TABLE dept_10
  CLUSTER personnel (department_id)
  AS SELECT * FROM employees WHERE department_id = 10;

CREATE TABLE dept_20
  CLUSTER personnel (department_id)
  AS SELECT * FROM employees WHERE department_id = 20;
```

## DROP CLUSTER

drop\_cluster::=



```
DROP CLUSTER language;
```

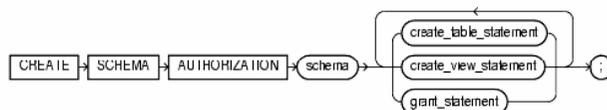
```
DROP CLUSTER personnel
INCLUDING TABLES
CASCADE CONSTRAINTS;
```

## SCHEMA

- C'est un sous-ensemble de la BD affecté à un utilisateur (CREATE USER)
- Un schéma peut contenir
  - tables, vues, etc.
- Nécessité d'autorisations (GRANT)

## CREATE SCHEMA

create\_schema::=



```
CREATE SCHEMA AUTHORIZATION oe
CREATE TABLE new_product
(color VARCHAR2(10) PRIMARY KEY, quantity NUMBER)
CREATE VIEW new_product_view
AS SELECT color, quantity FROM new_product WHERE color = 'RED'
GRANT select ON new_product_view TO hr;
```

## 3.2 - Gestion des utilisateurs

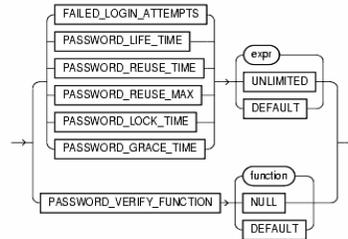
- Profil
- Utilisateur
- Gestion des droits / privilèges
  - GRANT
  - REVOKE

## CREATE PROFILE

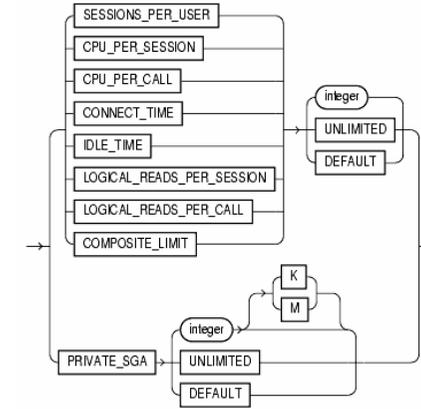
**create\_profile::=**



**password\_parameters::=**



**resource\_parameters::=**



## Exemple

```

CREATE PROFILE new_profile
  LIMIT PASSWORD_REUSE_MAX DEFAULT
        PASSWORD_REUSE_TIME UNLIMITED;

CREATE PROFILE app_user LIMIT
  SESSIONS_PER_USER      UNLIMITED
  CPU_PER_SESSION        UNLIMITED
  CPU_PER_CALL            3000
  CONNECT_TIME           45
  LOGICAL_READS_PER_SESSION DEFAULT
  LOGICAL_READS_PER_CALL 1000
  PRIVATE_SGA             15K
  COMPOSITE_LIMIT         5000000;
  
```

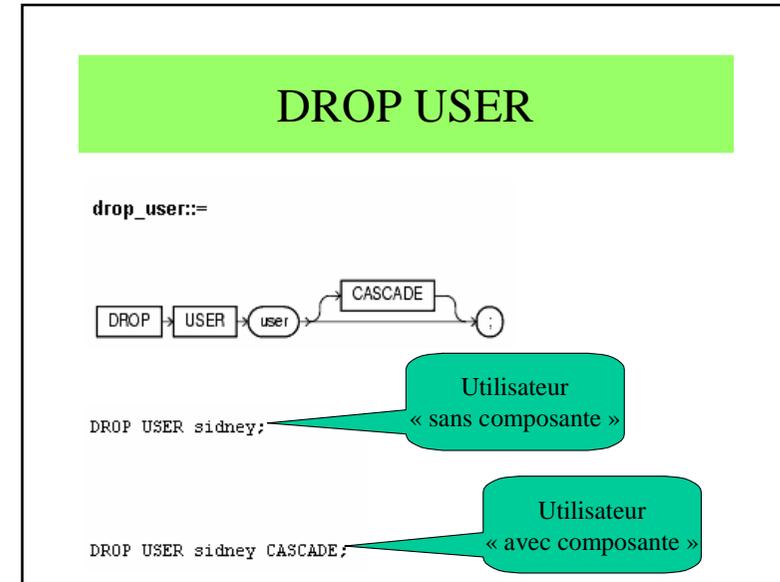
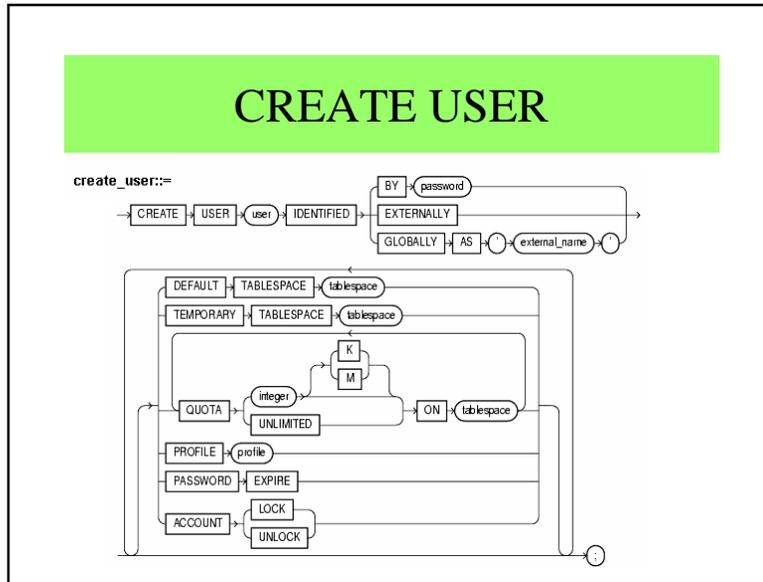
## DROP PROFILE

**drop\_profile::=**



```
DROP PROFILE app_user CASCADE;
```

Utilisateur  
profil par défaut

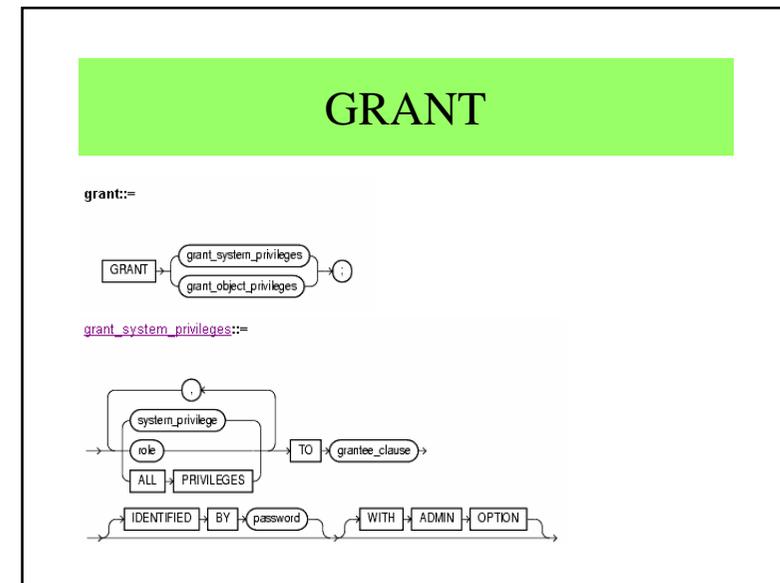


## Exemple

```

CREATE USER sidney
  IDENTIFIED BY welcome
  DEFAULT TABLESPACE demo
  QUOTA 10M ON demo
  TEMPORARY TABLESPACE temp
  QUOTA 5M ON system
  PROFILE app_user
  PASSWORD EXPIRE;

CREATE USER app_user1
  IDENTIFIED EXTERNALLY
  DEFAULT TABLESPACE tbs_1
  QUOTA 5M ON tbs_1
  PROFILE app_user;
    
```



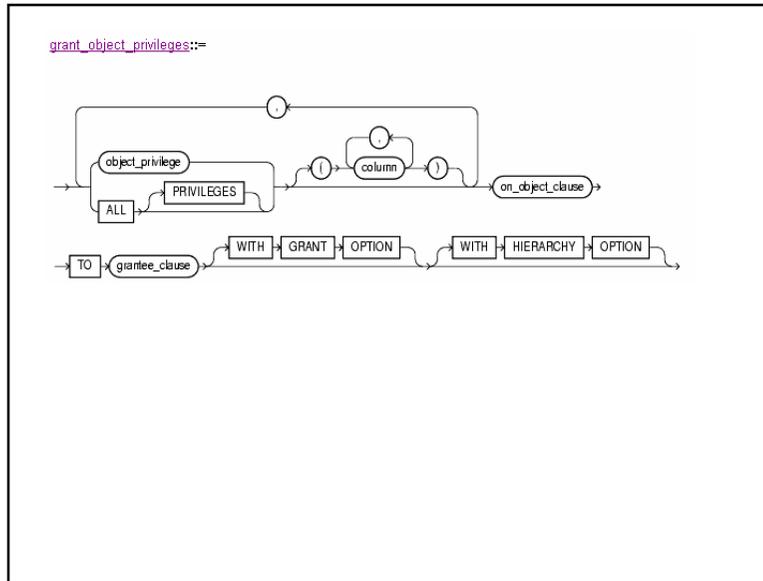


Table 16-3 Object Privileges Available for Particular Objects

Object Privilege	Table	View	Sequence	Procedures, Functions, Packages
ALTER	X		X	
DELETE	X	X		
EXECUTE				X
INDEX	X			
INSERT	X	X		
ON COMMIT REFRESH	X			
QUERY REWRITE	X			
READ				
REFERENCES	X	X		
SELECT	X	X	X	
UNDER		X		
UPDATE	X	X		
WRITE				

Extrait des privilèges possibles

## Exemples

```

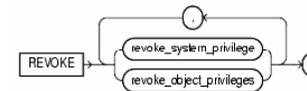
GRANT CREATE SESSION
TO hr;

GRANT dw_manager
TO sh
WITH ADMIN OPTION;

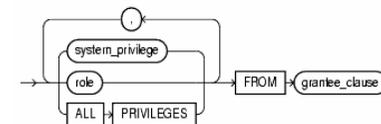
GRANT SELECT ON sh.sales TO warehouse_user;
  
```

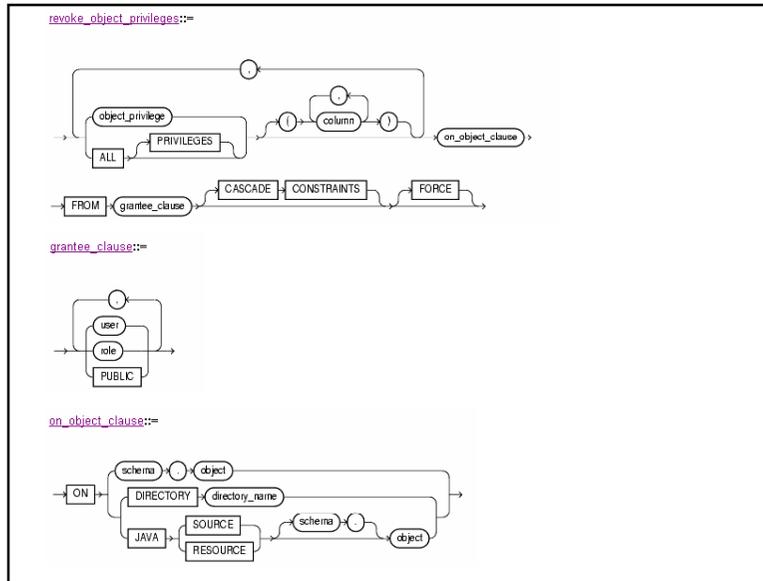
## REVOKE

`revoke::=`



`revoke_system_privileges::=`





## Exemples

```

REVOKE DROP ANY TABLE
  FROM hr, oe;

REVOKE dw_manager
  FROM sh;

REVOKE CREATE TABLESPACE
  FROM dw_manager;

REVOKE DELETE
  ON orders FROM hr;

REVOKE ALL
  ON orders FROM hr;
    
```

## 3.3 - Contrôle des transactions

- COMMIT
- ROLLBACK
- SAVEPOINT
- SET TRANSACTION

## COMMIT

*commit::=*

```

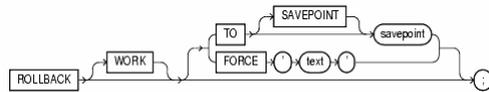
INSERT INTO regions VALUES (5, 'Antarctica');
COMMIT WORK;

COMMIT FORCE '22.57.53';
    
```

Numéro de la transaction forcée

## ROLLBACK

**rollback::=**



ROLLBACK;

ROLLBACK TO SAVEPOINT sp5;

ROLLBACK WORK  
FORCE '25.32.87';

## SAVEPOINT

**savepoint::=**



```
UPDATE employees
  SET salary = 7000
  WHERE last_name = 'Banda';
SAVEPOINT banda_sal;
```

```
UPDATE employees
  SET salary = 12000
  WHERE last_name = 'Greene';
SAVEPOINT greene_sal;
```

```
SELECT SUM(salary) FROM employees;
```

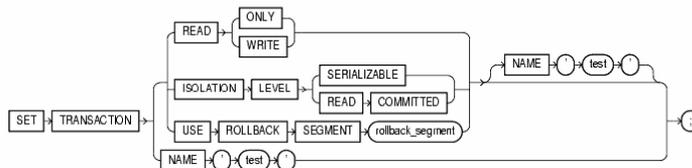
```
ROLLBACK TO SAVEPOINT banda_sal;
```

```
UPDATE employees
  SET salary = 11000
  WHERE last_name = 'Greene';
```

```
COMMIT;
```

## SET TRANSACTION

**set\_transaction::=**



## Exemples

```
COMMIT;
SET TRANSACTION READ ONLY NAME 'Toronto';
SELECT product_id, quantity_on_hand FROM inventories
  WHERE warehouse_id = 5;
COMMIT;
```

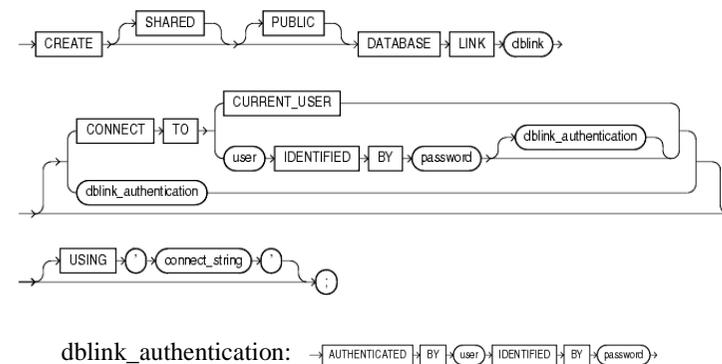
---

```
SET TRANSACTION USE ROLLBACK SEGMENT rs_1;
```

### 3.4 Accès à une BD distante

- Permet de travailler sur une BD située sur un autre site
- Permet de travailler sur plusieurs BD
- Mécanismes
  - Database Link

### Syntaxe



### Exemple « Public »

```
CREATE PUBLIC DATABASE LINK remote
  USING 'remote';
```

- En supposant que les privilèges soient créés:

```
UPDATE employees@remote
  SET salary=salary*1.1
  WHERE last_name = 'Dupont';
```

### Exemple pour un utilisateur précis

```
CREATE DATABASE LINK local CONNECT TO hr IDENTIFIED BY
  hr USING 'local';
```

```
SELECT * FROM employees@local;
```

```
INSERT INTO employees@local
  (employee_id, last_name, email, hire_date, job_id)
  VALUES (999, 'Claus', 'sclaus@oracle.com', SYSDATE,
  'SH_CLERK');
```

```
UPDATE jobs@local SET min_salary = 3000
  WHERE job_id = 'SH_CLERK';
```

```
DELETE FROM employees@local
  WHERE employee_id = 999;
```

## Pour l'utilisateur courant

```
CREATE DATABASE LINK
remote.us.oracle.com
CONNECT TO CURRENT_USER USING
'remote';
```

## Utilisation de synonyme

```
CREATE SYNONYM emp_table
FOR
oe.employees@remote.us.oracle
.com;
```

## Effacement



## 3.5 – Conclusions

- Administrer une BD et garantir son bon fonctionnement
  - utilisateurs
  - espaces