

UE 21 M2106

**PROGRAMMATION ET
ADMINISTRATION DES BASES
DE DONNEES**

**Volume horaire : 48h
1 DS de groupe, 1 DS de promo**

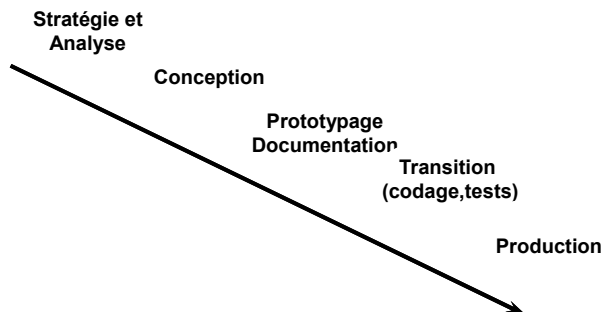
<http://docs.oracle.com/database/121/index.htm#>

1



**BASES DE DONNÉES
RELATIONNELLES
Rappels**

**Cycle de vie de développement
d'une application**



**Concept de Base de Données
Relationnelles**

- Dr. Edgar Frank Codd proposa le modèle relationnel en 1970
- Base pour les Systèmes de Gestion de Bases de Données Relationnelles (SGBDR)
- Le modèle relationnel consiste en :
 - Collection de relations
 - Ensemble d'opérateurs pour agir sur les relations
 - Contraintes d'intégrité pour l'exactitude et la cohérence des données

BASES DE DONNÉES RELATIONNELLES - RAPPELS

CARACTÉRISTIQUES DES SYSTÈMES DE GESTION DE BASES DE DONNÉES RELATIONNELLES (SGBDR)

Stockage et gestion de gros volumes de données

Données persistantes partagées

Technologies :

- recherches efficaces
- mises à jour fiables

BASES DE DONNÉES RELATIONNELLES : RAPPELS

Consultation

Efficacité de l'accès

Accès à des données

- persistantes
- partagées
- résistantes aux pannes
- pouvant être accédées et mises à jour de façon concurrente par d'autres applications

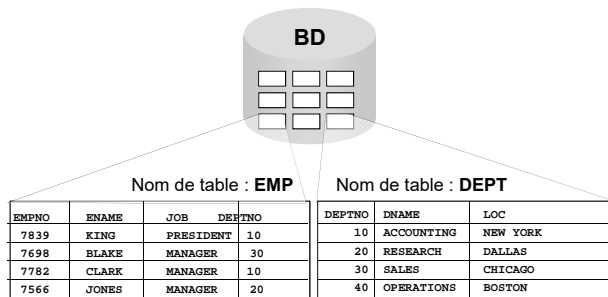
Contraintes

Exprimées dans le langage de définition de données

Langage de définition et de manipulation de données orienté utilisateurs : SQL (Structured Query Language)

Définition d'une Base de Données Relationnelle

Une BD relationnelle est une collection de relations ou de tables bi-dimensionnelles



STRUCTURE RELATIONNELLE

CONCEPTS DE BASE DU MODÈLE RELATIONNEL

- **Domaine:** ensemble de valeurs
Exemples : Domaine produits = {aspirine, alcool, vitamine,...}
 Domaine D1 = ensemble des entiers
- **Attribut:** variable prenant ses valeurs dans un domaine
Exemple : l'attribut "n° représentant" prend ses valeurs dans D1
- **Relation:** une relation R sur un ensemble d'attributs $U=\{A_1, A_2, \dots, A_n\}$ de domaines respectifs D_1, D_2, \dots, D_n est un sous-ensemble du produit cartésien $D_1 \times D_2 \times \dots \times D_n$
Exemple : commande(n° commande, date commande, date livraison, n° client)

STRUCTURE RELATIONNELLE - suite

Terminologie :

Chaque ligne d'une relation (table) est appelée **n-uplet** ou **tuple**

L'**ordre d'une relation** est le nombre d'attributs (de colonnes) de la relation

La **cardinalité d'une relation** est le nombre de tuples de la relation

Quelques tables utilisées dans le cours

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
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
7566			7698	22-FEB-81	1250	500	30
7566			7566	03-DEC-81			20

DEPT

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

SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Terminologie des BD relationnelles

1	2	3	4
EMPNO	ENAME	JOB	MGR
7839	KING	PRESIDENT	6
7698	BLAKE	MANAGER	7839
7782	CLARK	MANAGER	7839
7566	JONES	MANAGER	7839
7654	MARTIN	SALESMAN	7698
7499	ALLEN	SALESMAN	7698
7844	TURNER	SALESMAN	7698
7900	JAMES	CLERK	7698
7521	WARD	SALESMAN	7698
7902	FORD	ANALYST	7566
7369	SMITH	CLERK	7902
7788	SCOTT	ANALYST	7566
7876	ADAMS	CLERK	7788
7934	MILLER	CLERK	7782

STRUCTURE RELATIONNELLE - suite

Dépendance Fonctionnelle (DF) : exprime une contrainte

Soit X et Y deux sous-ensembles d'attributs de l'ensemble des attributs d'une relation R, il existe une dépendance fonctionnelle de X vers Y, notée $X \rightarrow Y$, si pour une valeur donnée à chaque attribut de X, il correspond **au plus une** valeur pour chaque attribut de Y

⇔ lorsque 2 n-uplets ont les mêmes valeurs pour les attributs de X, ils ont les mêmes valeurs sur les attributs de Y

Exemple: n° commande → date commande, date livraison, n° client

STRUCTURE RELATIONNELLE - suite

Type de dépendance fonctionnelle

⇒ **quelconque**

⇒ **élémentaire**

U : ensemble d'attributs

$X, Y \subset U$, $X \rightarrow Y$ est élémentaire $\Leftrightarrow \nexists X' \subset X$ tel que $X' \rightarrow Y$

Exemple : n° commande, n° produit \rightarrow n° client, n'est pas élémentaire

⇒ **directe**

$X, Y \subset U$, $X \rightarrow Y$ est directe $\Leftrightarrow \nexists Z \subset U$ tel que $X \rightarrow Z$ et $Z \rightarrow Y$

Exemple : n° commande \rightarrow n° client

n° client \rightarrow nom client

n° commande \rightarrow nom client, n'est pas directe

STRUCTURE RELATIONNELLE - suite

◦ Clé candidate, clé primaire, clé étrangère

Une **clé candidate** est un sous-ensemble minimal d'attributs tel qu'il existe une dépendance fonctionnelle de ce sous-ensemble vers tout autre attribut de la relation

Une **clé primaire** est une des clés candidate (exemple : n° commande)

Une **clé étrangère** est un ensemble d'attributs, clé primaire d'une autre relation (exemple : n° client dans la table commande) ou faisant référence à la clé primaire de la relation à laquelle il appartient (exemple : mgr dans la table scott.EMP)

STRUCTURE RELATIONNELLE - suite

◦ Contraintes

D'unicité de la clé : il ne peut exister deux n-uplets ayant la même valeur pour une occurrence de relation

D'intégrité référentielle : toute valeur d'une clé étrangère dans une relation doit exister dans une autre relation comme valeur de clé primaire (exemple : toute valeur de n° client dans la table commande doit exister dans la table client)

D'entité : tout attribut participant à une clé primaire ne doit pas avoir de valeur nulle

STRUCTURE RELATIONNELLE - suite

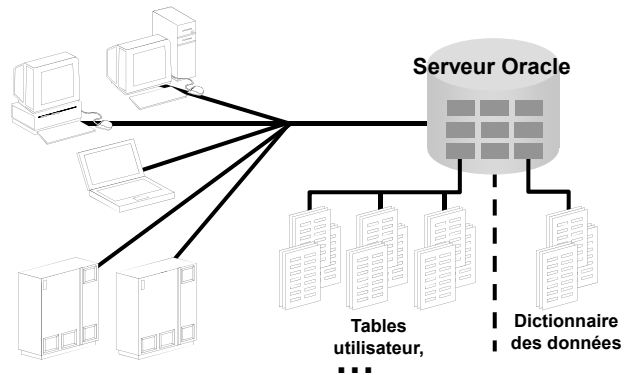
◦ Forme normale d'une relation

1FN : tous les attributs sont atomiques (monovalués et non composés) et la relation possède une clé primaire

2FN : 1FN et tous les attributs non clé sont en DF élémentaire avec la clé

3FN : 2FN et tous les attributs non clé sont en DF directe avec la clé

SGBDR

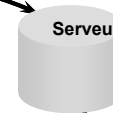


Communiquer avec un SGBDR en utilisant SQL via l'environnement SQL*Plus ou l'outil SQL Developer

Un ordre SQL est entré

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

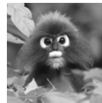
L'ordre est envoyé au serveur



Les données résultats sont affichées

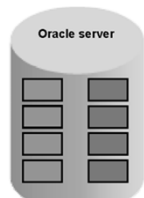
```
LOC
-----
NEW YORK
DALLAS
CHICAGO
BOSTON
```

2



DICTIONNAIRE DE DONNEES

Tables containing business data:
EMPLOYEES
DEPARTMENTS
LOCATIONS
JOB_HISTORY
...

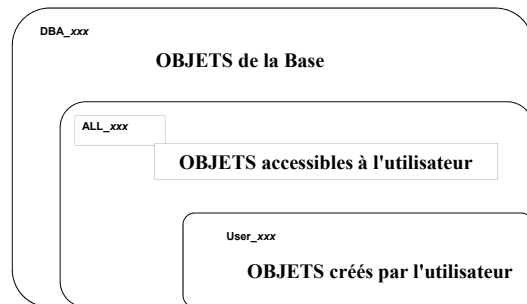


Data dictionary views:
DICTIONARY
USER_OBJECTS
USER_TABLES
USER_TAB_COLUMNS
...

DICTIONNAIRE DE DONNÉES

- Ensemble central des tables et vues utilisées comme référence en lecture seule concernant une base de données :
 - Structure logique et physique de la base de données
 - Informations relatives aux contraintes d'intégrité
 - Quantité d'espace allouée pour chaque objet de schéma, ainsi que la proportion actuellement utilisée
 - Utilisateurs
- Créé en même temps que la base de données et est automatiquement mis à jour lorsque la structure de la base est mise à jour
- Vue **DICTIONARY** : descriptions des tables et vues du DD

VUES DU DICTIONNAIRE DE DONNÉES



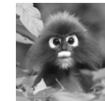
VUES DU DICTIONNAIRE Exemples et Catégories

VUES	Description
dictionary dict_columns	Informations générales
dba_tables dba_objects dba_lobs dba_tab_columns dba_constraints	Informations sur la structure des objets de la base
dba_users dba_sys_privs dba_roles	Information sur les Privilèges, Utilisateurs, Rôles

VUES DU DICTIONNAIRE Exemples et Catégories

VUES	Description
dba_extents dba_free_space dba_segments	Structure physique de la Base
dba_rollback_segs dba_data_files dba_tablespaces	Structure logique de la Base
dba_audit_trail dba_audit_objects dba_audit_obj_opts	Information de d'audit

3



RÉVISIONS SQL

PRINCIPAUX ORDRES SQL

SELECT
INSERT
UPDATE
DELETE

Langage de manipulation de
données (DML)

CREATE
ALTER
DROP
RENAME
TRUNCATE

Langage de définition de
Données (DDL)

COMMIT
ROLLBACK
SAVEPOINT

Contrôle des transactions

GRANT
REVOKE

Langage de contrôle des
données (DCL)



ORDRE SELECT OPÉRATEURS DE COMPARAISON

=, <, >, <=, >=, <> ou !=

BETWEEN ... AND ... : entre deux valeurs (inclusif)

IN (liste) : appartenance à une liste

LIKE : correspondance avec un pattern (% , _)

IS NULL : est une valeur nulle

Liaison entre tables relationnelles

- Chaque ligne dans une table est identifiée de façon unique par une clé primaire
- Vous pouvez lier logiquement des données de plusieurs tables en utilisant les clés étrangères

Nom de la table : EMP

Nom de la table : DEPT

EMPNO	ENAME	JOB	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	PRESIDENT	10	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	30	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	10	30	SALES	CHICAGO
7566	JONES	MANAGER	20	40	OPERATIONS	BOSTON



Clé primaire



Clé étrangère



Clé primaire



ORDRE SELECT JOINTURE

☐ PRODUIT CARTÉSIEN (CROSS JOIN)

Pas de condition de jointure (ou condition invalide)

Toutes les lignes de la première table sont reliées à toutes les lignes de la deuxième table.

À éviter !!

☐ TYPES DE JOINTURES

Équijointure Thêtajointure Jointure externe Autojointure



ÉQUIJOINTURE (NATURAL | [INNER] JOIN)

Condition de sélection (pivot de jointure) :

égalité de valeur entre la clé primaire de la première table et la clé étrangère de la deuxième (qui référence la clé primaire de la 1^{ère} table)

EMP			DEPT		
EMPNO	ENAME	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	10	10	ACCOUNTING	NEW YORK
7698	BLAKE	30	30	SALES	CHICAGO
7782	CLARK	10	10	ACCOUNTING	NEW YORK
7566	JONES	20	20	RESEARCH	DALLAS
7654	MARTIN	30	30	SALES	CHICAGO
7499	ALLEN	30	30	SALES	CHICAGO
7844	TURNER	30	30	SALES	CHICAGO
7900	JAMES	30	30	SALES	CHICAGO
7521	WARD	30	30	SALES	CHICAGO
7902	FORD	20	20	RESEARCH	DALLAS
7369	SMITH	20	20	RESEARCH	DALLAS
...			...		
14 rows selected.			14 rows selected.		

Foreign key

Primary key

Bases de données - © Christine Bonnet

29

THÊTAJOINTURE ([INNER] JOIN)

Pivot de jointure :

utilise des opérateurs autres que l'égalité

EMP			SALGRADE		
EMPNO	ENAME	SAL	GRADE	LOSAL	HISAL
7839	KING	5000	1	700	1200
7698	BLAKE	2850	2	1201	1400
7782	CLARK	2450	3	1401	2000
7566	JONES	2975	4	2001	3000
7654	MARTIN	1250	5	3001	9999
7499	ALLEN	1600			
7844	TURNER	1500			
7900	JAMES	950			
...					
14 rows selected.					

salary

Pivot : "salary dans la table EMP est entre le plus bas et le plus haut salaire de la table SALGRADE"

30

Bases de Données - © Christine Bonnet

JOINTURE EXTERNE (LEFT|RIGHT|FULL [OUTER] JOIN)

Les lignes de la table dominante (privilégiée) sont sélectionnées même si la condition de jointure n'est pas vérifiée

EMP		DEPT	
ENAME	DEPTNO	DEPTNO	DNAME
KING	10	10	ACCOUNTING
BLAKE	30	30	SALES
CLARK	10	10	ACCOUNTING
JONES	20	20	RESEARCH
...		...	
		40	OPERATIONS

Aucun employé dans le département OPERATIONS

31

Bases de Données - © Christine Bonnet

JOINTURE EXTERNE Exemple 1

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
...		
De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

20 rows selected

Bases de données - © Christine Bonnet

32

JOINTURE EXTERNE Exemple 2

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e RIGHT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Davies	50	Shipping
...		
Kochhar	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
	190	Contracting

20 rows selected.

JOINTURE EXTERNE Exemple 3

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e FULL OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
...		
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		
	190	Contracting

21 rows selected.

AUTOJOINTURE ([INNER] JOIN)

Jointure d'une table à elle-même

EMP (WORKER)			EMP (MANAGER)	
EMPNO	ENAME	MGR	EMPNO	ENAME
7839	KING		7839	KING
7698	BLAKE	7839	7839	KING
7782	CLARK	7839	7839	KING
7566	JONES	7839	7839	KING
7654	MARTIN	7698	7698	BLAKE
7499	ALLEN	7698	7698	BLAKE

Pivot : "MGR dans la table WORKER est égal à EMPNO dans la table MANAGER"

SYNTAXE

```
SELECT table1.column, table2.column
FROM   table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
ON (table1.column_name = table2.column_name)] |
[CROSS JOIN table2];
```



DDL - CONTRAINTE FOREIGN KEY

```
SQL> CREATE TABLE emp(
2      empno      NUMBER(4) CONSTRAINT emp_empno_pk PRIMARY KEY,
3      ename       VARCHAR2(10) NOT NULL,
4      job        VARCHAR2(9),
5      mgr        NUMBER(4),
6      hiredate    DATE,
7      sal         NUMBER(7,2),
8      comm        NUMBER(7,2),
9      deptno      NUMBER(7,2) NOT NULL,
10     CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
11     REFERENCES dept );
```

37

Bases de Données - © Christine Bonnet

ACTIONS DÉFINIES PAR LES CONTRAINTES FOREIGN KEY

☐ ON DELETE NO ACTION (par défaut)

Une suppression est impossible dans la table parente si des lignes en sont dépendantes

☐ ON DELETE SET NULL

Une suppression dans la table parente entraîne la mise à jour des clés étrangères dépendantes à NULL dans la table secondaire

☐ ON DELETE CASCADE

Une suppression dans la table parente entraîne la suppression des lignes dépendantes dans la table secondaire

38

Bases de Données - © Christine Bonnet



ORDRE SELECT : FONCTION Fonction NVL

Conversion d'une valeur NULL en une valeur significative

Types de données convertissables : DATE, NUMBER, CHAR ou VARCHAR2

- NVL(comm,0)
- NVL(hiredate,'01-JAN-2010')
- NVL(job,'No Job Yet')

Bases de données - © Christine Bonnet

39

ORDRE SELECT : FONCTION Fonction DECODE

DECODE ≡ instruction CASE ou IF-THEN-ELSE

```
DECODE(col/expression, valeur1, résultat1
                        [, valeur2, résultat2,...,]
                        [, valeur par défaut])
```

40

Bases de Données - © Christine Bonnet



ORDRE SELECT GROUPEMENT DE DONNÉES

CLAUSE GROUP BY

Diviser les lignes d'une table en groupes plus petits

Toutes les colonnes du SELECT qui ne sont pas dans des fonctions de groupe doivent être dans la clause GROUP BY

EMP		
DEPTNO	SAL	
10	2450	2916.6667
10	5000	
10	1300	
20	800	2175
20	1100	
20	3000	
20	3000	
20	2975	
30	1600	1566.6667
30	2850	
30	1250	
30	950	
30	1500	
30	1250	

DEPTNO	AVG(SAL)
10	2916.6667
20	2175
30	1566.6667

41

Bases de Données - © Christine Bonnet

FONCTIONS DE GROUPE

AVG	COUNT
MAX	MIN
SUM	VARIANCE
STDDEV	



Ignorent les valeurs nulles dans la colonne à l'exception de COUNT(*)



La fonction NVL force les fonctions de groupe à inclure les valeurs nulles (select AVG(NVL(comm,0)))

42

Bases de Données - © Christine Bonnet



CONTRÔLE DES TRANSACTIONS

TRANSACTION :

- ✓ Des ordres DML qui changent les données
- ✓ Un (seul) ordre DDL
- ✓ Un (seul) ordre DCL

Bases de données - © Christine Bonnet

43

TRANSACTION

Débuté quand le premier ordre SQL est exécuté

Se termine avec un des événements suivants :

- ✓ COMMIT ou ROLLBACK
- ✓ Un ordre DDL ou DCL (commit automatique)
- ✓ Déconnexion de l'utilisateur
- ✓ Crash system

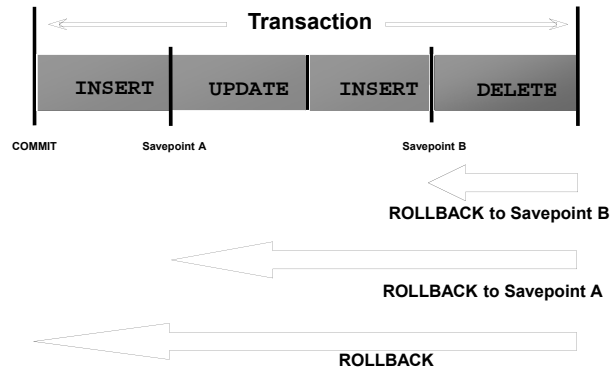
Le prochain ordre SQL exécuté débute automatiquement la transaction suivante



Bases de données - © Christine Bonnet

44

CONTRÔLE



Exercices