

Administration des bases de données

Jean-Yves Antoine

<http://www.info.univ-tours.fr/~antoine/>

Administration des bases de données

II – Objets avancés
dans les bases de données

OBJECTIFS

2.1. NOTIONS

2.1.1. Vues

2.1.2. Vues concrètes ou matérialisées

2.1.3. Déclencheurs (*triggers*) et index : introduction



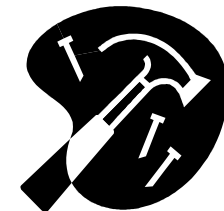
2.2. PRATIQUES

2.2.1. Savoir créer, manipuler et administrer une vue

2.2.2. Savoir créer, manipuler et administrer une vue concrète

2.2.3. Synonymes (ORACLE)

2.2.4. Savoir créer, manipuler et administrer des triggers simples
(ordre SQL pour action)





Objets « avancés »

Dès que l'on quitte le domaine des SGBD mono-utilisateurs, un certain nombre d'objets n'ayant pas de lien direct avec le modèle relationnel est nécessaire pour une utilisation et une administration efficace

- Vue
- Vue matérialisée
- Index
- Déclencheur (*trigger*)

Objets spécifiques à un SGBD

Exemple : ORACLE

- Synonyme
- Séquence (*sequence*)
- Cluster



Principe

- Fenêtre sur le contenu de la base de données
- Relation virtuelle : pas de duplication de données
- Mémorisation de l'expression définissant la vue, et non son contenu

Exemple

```
CREATE VIEW VilleFr AS  
SELECT Nom FROM Ville WHERE Etat='France' ;
```

No	Nom	Etat	Population
1	Lyon	France	1,2
2	Paris	France	10
3	Mexico	Mexique	18
4	Paris	Texas	0,05



Nom
Lyon
Paris



Voir les données de différentes manières

Exemple : calcul de données de synthèse avec fonctions d'agrégat

```
CREATE VIEW Pop_Pays AS      SELECT Etat, Avg(Pop)
                             FROM Ville
                             GROUP BY Etat;
```

Sécurité vis à vis des données cachées

Exemple : n'autoriser que l'accès aux vues, pas aux tables

Assurer l'indépendance logique

Extension : vue reprenant le schéma d'une relation à qui on a rajouté des attributs

Restructuration : vue de même nom qu'une relation divisée en deux

Attention : indépendance imparfaite (exemple : mise à jour)



Abréviation de requête

- Décomposer une requête en sous-requêtes (vue: résultat intermédiaire)
- Vue intermédiaire : peut avoir une influence bénéfique sur le temps d'exécution d'une requête dans un environnement client/serveur

VILLE	No	Nom	Etat	Pop
	1	Lyon	France	1,2
	2	Paris	France	10
	3	Mexico	Mexique	18
	4	Paris	Texas	0,05

PAYS	Etat	Pop
	France	62
	Turquie	90
	Mexique	120
	Texas	30

Exemple : rechercher les villes de plus de 1 million d'habitants dans les pays de plus de cinquante millions d'habitants

- requête avec jointure
- avec requête imbriquée
- avec vue intermédiaire : rôle de macro ou fonction



Consultation (SELECT)

aucun problème

Mise à jour d'une table via une vue (UPDATE, INSERT, DELETE)

- Mise à jour avec ou sans vérification des contraintes d'intégrité
- Problèmes éventuels de complétion des données

Exemple

No	Nom	Etat	Population
1	Lyon	France	1,2
2	Paris	France	10
3	Mexico	Mexique	18
4	Paris	Texas	0,05

```
CREATE VIEW Ville_Fr AS  
SELECT Nom  
FROM Ville  
WHERE Etat='France'  
WITH CHECK OPTION;
```

```
INSERT INTO Ville_Fr  
VALUES ('Blois');
```

SQL3 **vue multitable** – mise à jour impossible

vue monotable – possible si attribut clé (voire champs avec contrainte NOT NULL) présent dans la vue + absence de clause DISTINCT ou fonction d'agrégat (GROUP BY en particulier)



ORACLE

- Utilisation des vues telle qu'autorisée par le standard SQL3.
- Dictionnaire de données : USER_VIEWS, ALL_VIEWS

MySQL

- Support depuis la version 5.0
- Utilisation des vues telle qu'autorisée par le standard SQL3.
- Dictionnaire de données : INFORMATION_SCHEMA.VIEWS

DB2

- Supportés

Access

- Tables temporaires (non nommées) dans les requêtes imbriquées
- Pas de vues permanentes



Création

Définition comme résultat d'une requête de sélection SQL

Syntaxe `CREATE VIEW <nom_vue> AS <requête SQL>`
 `[WITH { CHECK OPTION | READ ONLY }] ;`

Vérification des contraintes sur mise à jour (CHECK OPTION)

- Par défaut, pas de contrôle des contraintes de définition de la vue
- Option de vérification fortement recommandée

Suppression

Syntaxe `DROP VIEW <nom_vue> [RESTRICT | CASCADE] ;`

CASCADE : suppression en cascade des vues reliées



Principe

Vue « réelle » : fenêtre sur le contenu de la base de données MAIS ...

- **Objet réel** : recopie physique des données concernées
- **Actualisation périodique** : mise à jour automatique avec les données physiques de la BD

	Vue	Vue concrète
Données observées	virtuelles	réelles
Accès aux données sources	direct	indirect
Mise à jour des données	immédiate	périodique / sur demande

Utilisation

- **Pré-agrégation des données** - data warehouse (entrepôt de données)
- **Pré-récupération des données** - bases de données distribuées
- **Sécurisation des données** - plus d'accès aux données sources



Cas d'école : ORACLE

Paramétrisation des vues concrètes

- Chargement des données à la création ou lors de la première requête
- Fréquence d'actualisation des données
- Mode d'actualisation des données : recopie complète ou seulement incrémentale (plus rapide).
- Optimisation par réécriture de la requête de définition : forcer l'optimisateur de requête à réécrire son plan d'exécution pour accélérer l'exécution

```
CREATE MATERIALIZED VIEW <nom_vue_concrete>  
TABLESPACE <nom_tablespace>  
BUILD [IMMEDIATE | DEFERRED]  
[ENABLE QUERY REWRITE]  
REFRESH [FAST | COMPLETE | FORCE | NEVER]  
[START WITH date] [NEXT date]  
AS <requête SQL> ;
```



ORACLE

- MATERIALIZED VIEW à partir de la version 10g (avant : SNAPSHOT)
- Dictionnaire de données USER_MVIEWS, ALL_MVIEWS

MySQL

- Non supportées
- Emulation plus ou moins satisfaisante sous la forme de procédure stockées

DB2

- Supportées (SQL3)

MS ACCESS

- Pas de vues concrètes



ORACLE : Création

Sans réactualisation périodique (syntaxe simplifiée)

```
CREATE MATERIALIZED VIEW <nom_vue_concrete>
TABLESPACE <nom_tablespace>
BUILD [IMMEDIATE | DEFERRED]
[ENABLE QUERY REWRITE]
REFRESH [FAST | COMPLETE | FORCE | NEVER]
        [ON COMMIT | ON DEMAND]
AS <requête SQL> ;
```

Avec réactualisation périodique (syntaxe simplifiée)

```
CREATE MATERIALIZED VIEW <nom_vue_concrete>
TABLESPACE <nom_tablespace>
BUILD [IMMEDIATE | DEFERRED]
[ENABLE QUERY REWRITE]
REFRESH [FAST | COMPLETE | FORCE | NEVER]
        [START WITH date] [NEXT date]
AS <requête SQL> ;
```

START WITH SYSDATE NEXT SYSDATE+7

ORACLE : Réactualisation incrémentale et journalisation

- Modes de réactualisation

FAST	mise à jour incrémentale
COMPLETE	duplication intégrale
FORCE	FAST pas défaut et COMPLETE si impossible
NEVER	pas de réactualisation !

- Obligation de créer un journal (MATERIALIZED VIEW LOG) pour utiliser les modes FAST ET FORCE

```
CREATE MATERIALIZED VIEW LOG <nom_journal_vue>  
TABLESPACE DATA  
STORAGE (INITIAL <size> NEXT <size_next> PCTINCREASE 0) ;
```

Dictionnaire de données ALL_MVIEWS_LOG

ORACLE : Suppression

Syntaxe DROP MATERIALIZED VIEW <nom_vue_concrete> ;



Dictionnaire Oracle

- ALL_MVIEWS
- ALL_MVIEWS_LOG
- USER_MVIEWS



OWNER
MVIEW_NAME
CONTAINER_NAME
QUERY
QUERY_LEN
UPDATABLE
UPDATE_LOG
MASTER_ROLLBACK_SEG
MASTER_LINK
REWRITE_ENABLED
REWRITE_CAPABILITY
REFRESH_MODE
REFRESH_METHOD
BUILD_MODE
FAST_REFRESHABLE
LAST_REFRESH_TYPE
LAST_REFRESH_DATE
STALENESS
AFTER_FAST_REFRESH
UNKNOWN_PREBUILT
UNKNOWN_PLSQL_FUNC
UNKNOWN_EXTERNAL_TABLE
UNKNOWN_CONSIDER_FRESH
UNKNOWN_IMPORT
COMPILE_STATE
USE_NO_INDEX



Utilisation

- Alias d'un objet Oracle : table, vue, séquence ...
- Confidentialité
- Raccourci pour simplifier l'accès aux objets
- Maintenabilité

Création

```
CREATE [OR REPLACE] [PUBLIC] SYNONYM [<schema>].  
    <nom_synonym> FOR [<schema>].<nom_objet>;
```

- ▶ synonyme public ou privé

Suppression

```
DROP SYNONYM <nom_synonym>;
```



Dictionnaire Oracle

- ALL_SYNONYMS
- USER_SYNONYMS



OWNER
SYNONYM_NAME
TABLE_OWNER
TABLE_NAME
DB_LINK



Triggers / Déclencheurs

- Action se déclenchant automatiquement suite à un événement ou au passage à un état donné de la base de données ou par un ordre SQL
- Action : ordre SQL ou procédure stockée
- Porte sur des tables ou des vues
- Trois parties : a) description de l'évènement traqué, b) conditions éventuelles de déclenchement sur l'évènement, c) action à réaliser

Différents types de déclencheurs

- sur ordre LMD (INSERT, UPDATE, DELETE).
- sur ordre LDD (CREATE, ALTER, DROP, GRANT, COMMENT...)
- sur événement SGBD (démarrage, arrêt..) : déclencheur d'instance

Déclencheur LMD

- sur l'état global de l'objet considéré : déclencheur d'état
- sur chaque tuple modifié : déclencheur de ligne

Applications

- Règles de gestion non modélisables par des contraintes d'intégrité
- Chargement automatique de tables d'audit sur l'évolution de la BD
- Implémenter des alertes ...



Création (SQL 3 : syntaxe simplifiée)

```
CREATE TRIGGER nom_trigger
BEFORE | AFTER
INSERT | DELETE | UPDATE [ OF liste_colonne ]
ON nom_table
FOR EACH STATEMENT | ROW
[WHEN condition]
BEGIN
    code_trigger
END
```

Activation / Désactivation (SQL 3 : syntaxe simplifiée)

```
ALTER TRIGGER nom_trigger ENABLE | DISABLE
```

Suppression (SQL 3 : syntaxe simplifiée)

```
DROP TRIGGER nom_trigger
ON nom_table
[CASCADE | RESTRICT]
```



ORACLE

- Programme PL/SQL (langage propre à ORACLE)
- Etude approfondie dans le chapitre du cours consacré à PL/SQL
- **Dictionnaire de données** : ALL_TRIGGERS, USER_TRIGGERS

MySQL

- Supportés depuis la version 5.0.2.
- Support encore rudimentaire en terme d'actions et d'administration
- **Dictionnaire de données** : INFORMATION_SCHEMA.TRIGGERS

DB2

- Supportés depuis la version 6
- Peuvent être déclenchés en cascade (16 niveaux)

MS ACCESS

- Pas de support → MS SQL Server



Dictionnaire Oracle

- ALL_TRIGGERS
- USER_TRIGGERS



OWNER
TRIGGER_NAME
TRIGGER_TYPE
TRIGGERING EVENT
TABLE_OWNER
BASE_OBJECT_TYPE
TABLE_NAME
COLUMN_NAME
REFERENCING_NAMES
WHEN_CLAUSE
STATUS
DESCRIPTION
ACTION_TYPE
TRIGGER_BODY



Optimisation

- BD réelle : très grosse masse de données impliquant une exécution des requêtes coûteuses d'un point de vue computationnel
- Les index sont une des solutions d'optimisation, en accélérant l'accès aux données les plus souvent utilisées

Index

- **Principe** : structure de données associant directement à une valeur de clé donnée l'adresse physique du tuple considéré dans la base de données
 - **Organisation d'index** : brut, trié, hiérarchisé, avec hachage
 - Index sur clé primaire mais également sur champ très utilisés (« clé » secondaire)
- ↳ Index étudiés dans la partie « Optimisation » du cours (P. Marcel)

Dictionnaire Oracle

- ALL_INDEXES, USER_INDEXES, ALL_IND_COLUMNS

BIBLIOGRAPHIE

Notions générales

DATE C. J. (2000) *Introduction aux bases de données* (7^e édition), Vuibert, Paris.

Ouvrages disponibles à la B.U.

BRIARD G. (2006) *Oracle 10g sous Windows*, Eyrolles, 2-212-11707-8

HEURTEL O. (2008) *Oracle 11i*, ENI, St Herblain, France (catalogue : 005.75 HEU)

HEURTEL O. (2005) *Oracle 10g*, ENI, St Herblain, France (005.75 HEU)

LONEY K., BRYLA B. (2005) *Oracle 10g, guide du DBA*. Oracle Press (005.75 LON)

MySQL AB (2005) *MySQL 5: Guide de l'administrateur*, Campus Pr., Paris (005.75 MYS)

NOIRAULT C. (2008) *Oracle 11g, entraînez-vous à administrer une base de données* ENI, St Herblain (005.75 NOI)

SOUTOU C. (2005) *SQL pour Oracle*, Eyrolles, ISBN 2-212-11697-7. (005.75 SOU)

THIBAUT C. (2005) *MySQL 5: Installation, mise en œuvre, administration, programmation*, ENI, St Herblain (005. THI)

RESSOURCES ELECTRONIQUES

Sur la Toile

Référence MySQL en français

<http://dev.mysql.com/doc/refman/5.0/fr/index.html>

Comparatif fonctionnalités SGBD

www.answers.com/topic/comparison-of-relational-database-management-systems

Serveur de documentation Emery du département

- Oracle SQL Reference