

Introduction aux Bases de Données

Ecole Centrale Paris - Cours électif IS1230

Exercices et Travaux Pratiques

version 2.3
24 septembre 2015

François Boisson, Abal-Kassim Cheik Ahamed, Pascale Mourier, Nicolas Travers

TABLE DES MATIÈRES

1	BDD	3
1.1	Analyse des besoins	3
1.2	Modèle Entité/Association	3
1.3	Schéma relationnel	3
1.4	Contraintes	3
1.5	LDD : Langage de Description de Données	4
2	SQLDeveloper	5
2.1	Modélisation	5
2.2	Création de la base de données	5
3	SQL	6
3.1	Chargement des données	6
3.2	Algèbre et SQL	6
3.3	SQL	6
4	PL/SQL	8
4.1	Procédure simple	8
4.2	Fonction	8
4.3	Exceptions	8
4.4	Curseurs	8
4.5	Procédure complexe	9
5	Trigger	10
5.1	Responsable d'un cours	10
5.2	Incompatibilité de salles	10
5.3	Triggers bonus	11
6	Bases NoSQL : MongoDB	12
6.1	Gestion de collection	12
6.2	Interrogation simple	12
6.3	Agrégats	12
6.4	Map/Reduce	13

1 CRÉATION D'UNE BASE DE DONNÉES

1.1 Analyse des besoins

Soit le texte suivant décrivant les besoins pour une base de données gérant les enseignements d'un établissement d'enseignement supérieur.

Nous avons besoin d'une base de données permettant de gérer les enseignements de notre établissement. Il faut prendre en compte les étudiants avec leur nom, prénom, date de naissance et numéro d'identification, qui lors de leurs études dans le supérieur s'inscrivent à différents parcours (L1, L2, L3, M1, M2). Ces parcours ont des dénominations, sont reliés entre eux (représentant les possibilités de passage entre parcours) et sont composés d'un ensemble de cours. A noter que des étudiants de différents parcours peuvent suivre un même cours qui doit correspondre au niveau du parcours associé.

Ces mêmes cours chaque année ont un enseignant responsable, un coefficient et la moyenne des notes des étudiants. Cette moyenne sera calculée en fin d'année par la moyenne des notes obtenues par les étudiants à ce cours. La moyenne des notes d'un étudiant pondérée par les coefficients sera calculée pour donner une mention à son parcours (Refusé/Passable/Assez bien/Bien/Très bien).

Afin de gérer la réservation des salles pour chaque TD d'un parcours, des groupes sont créés dans lesquels seront répartis les étudiants. Les séances de cours forment un emploi du temps qui se verront associer la plage horaire, le type (cours/TD/TP) et une salle attribuée. Les salles sont localisées par un étage et un numéro dans un bâtiment avec une capacité maximale et un type (Amphi, TD, TP).

Les enseignants interviennent dans différentes séances, il faut prendre en compte, chaque année, leur service en nombre d'heures de cours, de TD et de TP qu'il effectue. Un enseignant peut être titulaire, moniteur ou vacataire

- 1.1.1 Donner les acteurs principaux de ce texte ;
- 1.1.2 Pour chaque acteur, définir leurs caractéristiques ;

1.2 Modèle Entité/Association

- 1.2.1 Créer le schéma Entité/Association correspondant ;

1.3 Schéma relationnel

- 1.3.1 Transformer le schéma E/A en schéma relationnel ;
- 1.3.2 Définir les dépendances fonctionnelles dans chaque Relation ;
- 1.3.3 Normaliser 3^o Forme Normale ;

1.4 Contraintes

- 1.4.1 Donner la clé primaire de chaque relation ;
- 1.4.2 Donner la/les clés étrangères de chaque relation ;

1.4.3 Donner les contraintes de domaines spécifiques ;

1.5 LDD : Langage de Description de Données

1.5.1 Donner les requêtes SQL de création de tables pour : Etudiant, Parcours et Inscription ;

2 MISE EN PRATIQUE AVEC *SQLDeveloper*

Pour la suite des TPs, l'utilisation de Oracle Express et SQLDeveloper est requise. Veuillez suivre le guide d'installation/connexion disponible sur Caroline "Oracle_SQLDeveloper_Guide.pdf".

2.1 Modélisation

A l'aide de *DataModeler*¹, créer le modèle des tables à partir du schéma Entité/Association réalisé à la question 1.2. Pour cela, il faut ouvrir un nouveau modèle (Menu "File/New Model"), une fois la fenêtre ouverte double-click sur "Add diagram". Vous pouvez maintenant créer les entités et les cardinalités qui les relient.

2.2 Création de la base de données

Générer les scripts de création de la Base de Données (cf. Guide / DataModeler). Enregistrer le fichier, puis fermer le modèle.

Lancer le script à l'aide d'une console SQL de SQLDeveloper (cf. Guide / SQL). Pour exécuter l'ensemble des requêtes utilisez la touche F5.

1. *DataModeler* est un outils de SQLDeveloper, le guide détailles les fonctionnalités utiles dans le cadre de ce TP

3 INTERROGATION DE LA BASES DE DONNÉES

3.1 Chargement des données

Afin de travailler sur une base de données homogènes, nous travaillerons sur la base de données dont voici le script de création se trouve le fichier 'BDD_Ecole.sql' de l'espace de téléchargement du cours. Une fois le script exécuté, veuillez exécuter le script d'insertion des données 'BDD_Ecole_insert.sql'.

Pour tester une requête SQL sur la base de données, ouvrir un onglet 'SQL', écrire la requête SQL, puis exécuter avec la touche F5 (Cf. Guide / SQL).

3.2 Algèbre et SQL

Pour chacune des questions suivantes, produire l'expression algébrique correspondante puis la requête SQL. Tester ensuite la requête SQL sur la base de données.

- 3.2.1 Donner la liste des noms d'enseignants ;
- 3.2.2 Donner la liste des noms et prénoms des étudiants nés avant le '01/01/1990'.
Sous Oracle, une chaîne de caractère n'est pas une date. Pour cela, il faut typer le texte avec la commande : `TO_DATE('01/01/1990', 'DD/MM/YYYY')` ;
- 3.2.3 Donner le nom des enseignants responsables d'un cours ;
- 3.2.4 Donner les notes obtenues par l'étudiant 'Kevin Dupont' en 2014.
N'ayant pas de notes, le résultat est vide. Créer une requête SQL qui insère une note de 15 pour l'étudiant n°1 (Kevin Dupont) en 2014 pour le cours n°2. Exécuter à nouveau la requête ;
- 3.2.5 Donner le nom et le status des enseignants dont le service (nombre d'heures enseignées) dépasse 70 heures ;

3.3 SQL

Pour chacune des requêtes suivantes, donner la requête SQL correspondante et la tester sur la base de données.

- 3.3.1 Donner l'intitulé des parcours de M2 ;
- 3.3.2 Donner les noms des inscrits en 2014 ;
- 3.3.3 Donner les intitulés des cours du parcours de Master en Informatique (M1 ou M2, contenant le mot 'Informatique').
La clause "LIKE" est sensible à la casse sous Oracle (et beaucoup d'autres systèmes). Il est possible de contourner le problème avec des variables de sessions

```
ALTER SESSION SET NLS_COMP=LINGUISTIC;  
ALTER SESSION SET NLS_SORT=BINARY_CI;
```

Ensuite il faut exécuter la requête ;

- 3.3.4 Donner le nom des enseignants qui n'enseignent pas dans un cours dont ils sont responsables. Le résultat étant vide, créer une requête SQL qui modifie le responsable du cours 3 (Production et distribution de biens et de services) par l'enseignant n°4 (Eric Mathieur). Exécuter à nouveau la requête pour tester ;
- 3.3.5 Donner les intitulés distincts de cours dans lesquels intervient Mr 'Travers'. Proposer deux manières de le faire (avec et sans DISTINCT) ;
- 3.3.6 Donner les moyennes pour chaque année d'inscription de l'étudiant 'Olivier Jaquemin'. Ordonner le résultat par date croissante ;
- 3.3.7 Donner le nombre de séances de cours de 'Bases de Données' ayant lieu dans un Amphi ;
- 3.3.8 Donner le nom des étudiants qui ont réussi à toutes les matières (toutes les notes au dessus de la moyenne et pas de valeurs nulles) ;
- 3.3.9 Donner les noms des enseignants qui n'interviennent pas dans un parcours de M2 (2 requêtes SQL différentes) ;
- 3.3.10 Donner les noms des enseignants qui interviennent dans au moins un parcours de M2 et au moins un parcours de M1 ;
- 3.3.11 Donner la dernière année d'inscription de l'étudiant 'Guillard'. Donner deux manières de le faire (avec et sans MAX) ;
- 3.3.12 Donner le nombre d'étudiants inscrits en M1 en 2014 ;
- 3.3.13 Donner le nom des enseignants distincts qui sont responsables de plus d'un cours. Donner deux manières de le faire ;
- 3.3.14 Donner la moyenne pondérée (coeff) des notes de l'étudiant 'Jaquemin' en 2013 (sans utiliser la table Inscription) ;
- 3.3.15 Donner le nom des étudiants dont ayant plus de 4 notes supérieures à 15 dans toute leur scolarité ;
- 3.3.16 Donner les intitulés des enseignements se trouvant dans plus de 2 parcours différents ;
- 3.3.17 Donner le nom de l'enseignant qui a fait le plus grand service d'enseignement (sommes des heures) en 2013 ;
- 3.3.18 Donner les intitulés des cours ayant lieu dans le bâtiment A au premier étage le '2014-09-18' ;
- 3.3.19 Donner pour chaque parcours de 2013, le nombre d'étudiants, la moyenne et l'écart-type de leurs notes ;
- 3.3.20 Donner le nom des étudiants ayant suivi tous les parcours d'informatique de l'école ;

4 PROGRAMMATION DANS LA BASE DE DONNÉES

La création d'une procédure et d'une fonction est décrit dans le guide disponible sous claroline.

Pour exécuter une procédure, une console SQL suffit pour y faire l'appel de celle-ci avec le mot-clé 'CALL' avec le nom de la procédure et ses paramètres.

Pour chaque procédure créée, tester le résultat.

4.1 Procédure simple

- 4.1.1 Créer une procédure 'affEtudiants' qui affiche la liste des noms et prénoms d'étudiants inscrits d'une année donnée en paramètre ;
- 4.1.2 Créer une procédure 'affNotes' qui affiche les notes avec l'intitulé du cours correspondant pour un étudiant (nom et prénom) et année donnés ;
- 4.1.3 Créer une procédure 'affMatières' qui affiche les intitulés et coefficients des matières qu'un étudiant (nom et prénom) suit durant une année donnée. Trier les matières par coefficients décroissants ;

4.2 Fonction

- 4.2.1 Créer une fonction 'retEtudiant' qui renvoie l'identifiant d'un étudiant pour un nom et prénom donné ;
- 4.2.2 Créer une fonction 'moyennePonderee' qui renvoie la moyenne pondérée des résultats d'un étudiant (utiliser la fonction 'retEtudiant') et année donnés ;

4.3 Exceptions

- 4.3.1 Modifier la fonction 'retEtudiant' pour qu'elle puisse lever une exception si aucun étudiant n'est trouvé ;
- 4.3.2 Modifier la procédure 'affMatières' en utilisant la fonction 'retEtudiant' et afficher un message d'erreur lorsqu'une exception est levé ;

4.4 Curseurs

- 4.4.1 Créer une procédure 'affBulletins' qui pour un étudiant donné, affiche l'intitulé de chaque parcours (avec l'année d'inscription croissante) qu'il a suivi et pour chaque parcours les résultats obtenus pour chaque matière (avec intitulé) ;
- 4.4.2 Créer une procédure 'majBulletins' qui met à jour toutes les moyennes des inscriptions (utiliser la fonction 'moyennePondere') des étudiants d'un parcours donné (id) ;

4.5 Procédure complexe

4.5.1 Créer une procédure 'affPlanning' qui affiche le planning d'un parcours donné (niveau et intitulé), organisé par enseignement ;

5 DÉCLENCHEURS

La création d'un Trigger est décrit dans le guide disponible sous claroline.

On souhaite ajouter à la base de données les contraintes qui ne sont pas exprimable sous formes de clés primaires, clés étrangères, types, domaines de valeurs ou CHECK. Créer les Trigger correspondant aux contraintes suivantes :

5.1 Le responsable d'un enseignement est un enseignant titulaire

Jeu de test :

```
delete from enseignement where idEnseignement = 10;
delete from enseignement where idEnseignement = 11;

select * from enseignement

call insert_into_enseignement (10,'pratique de mysql_Boisson', 'Boisson','M1',20,2);
show errors;
call insert_into_enseignement (11,'pratique de mysql_Mourier', 'Mourier','M1',20,2);
show errors;

select * from enseignement;
```

la procédure 'insert_into_enseignement' est définie ci-dessous :

```
DROP PROCEDURE IF EXISTS insert_into_enseignement;
CREATE PROCEDURE insert_into_enseignement
    (id Integer, intitule varchar, nomResp varchar, niveau varchar, duree integer, coeff integer)
IS
    idResp INTEGER;
BEGIN
    select idEnseignant into idResp from Enseignant where Nom = nomResp ;
    insert into enseignement values (id,intitule,idResp,niveau,duree,coeff) ;
    DBMS_OUTPUT.PUT_LINE('insertion du cours <' || intitule || '> réussie ');
EXCEPTION
    WHEN OTHERS
    DBMS_OUTPUT.PUT_LINE('insertion du cours <' || intitule || '> refusée mais l exécution peut se
        poursuivre ');
END;
/
```

5.2 On ne peut avoir de séances de 'Cours' dans une salle de 'TP', et on ne peut avoir de 'TP' que dans des salles de 'TP'

Jeu de test :

```
/* procedure d'insertion avec traitement des refus d'insertion dans Mysql
----- */
DROP PROCEDURE IF EXISTS insert_into_cours;
CREATE PROCEDURE `insert_into_cours`
    (idEns integer, idGroupe integer, date date, heure time, duree time ,idSalle integer,idE integer,type
    varchar)
BEGIN
    insert into cours values (idEns,idGroupe, date, heure, duree,idSalle,idE,type) ;
EXCEPTION
    WHEN OTHERS
        DBMS_OUTPUT.PUT_LINE('insertion du cours refusée. ' || SQLERRM ||' : ' || SQLCODE);
END

/* Requetes de test */
delete from cours where date = '2014-10-15' limit 3;
select * from cours where date = '2014-10-15';

-- cours CM en salle de TP
call insert_into_cours (1,1,'2014-10-15','09:00:00','01:30:00',5,1,'Cours');
show errors;

-- cours TP en salle de amphi
call insert_into_cours (1,1,'2014-10-15','10:00:00','01:30:00',2,1,'TP');
show errors;

-- cours CM en salle de cours
call insert_into_cours (1,1,'2014-10-15','11:00:00','01:30:00',2,1,'Cours');
show errors;

select * from cours where date = '2014-10-15';
```

5.3 Triggers bonus

5.3.1 L'étage d'une salle ne peut être supérieur au nombre d'étages du bâtiment

5.3.2 Deux séances de cours ne peuvent avoir lieu dans la même salle en même temps

5.3.3 Le total cumulé des séances de cours ne peut dépasser le nombre d'heures maximum d'un enseignement

6 BASES NoSQL : MONGODB

Pour l'installation de l'environnement, des instructions de connexion et de requêtes, veuillez vous référer au guide disponible sur Claroline.

6.1 Gestion de collection

- 6.1.1 Se connecter à la base DBLP (cf Guide);
- 6.1.2 Créer un collection `publis`;
- 6.1.3 Créer un document (définition sur une ligne) : `w={"type": "Book", "title": "Modern Database Systems: The Object Model, Interoperability, and Beyond.", "year": 1995, "publisher": "ACM Press and Addison-Wesley", "authors": ["Won Kim"], "source": "DBLP"}`
- 6.1.4 Insérer le document dans la collection `"publis"` : `db.publis.save(w)`;
- 6.1.5 Créer et insérer deux autres publications à partir de cette page de conférence (type de publication) : <http://www.informatik.uni-trier.de/~ley/db/journals/vldb/vldb23.html>
- 6.1.6 Consulter le contenu de la collection ;
- 6.1.7 Télécharger et décompresser le fichier contenant les données DBLP.zip (disponible sous Claroline);
- 6.1.8 Importer les données du TP dans MongoDB (cf Guide);
- 6.1.9 Dans la console *MongoDB*, compter le nombre de documents insérés.

6.2 Interrogation simple

Créer des requêtes simples pour les phrases suivantes :

- 6.2.1 Liste de tous les livres ('Book');
- 6.2.2 Liste des publications depuis 2011 ;
- 6.2.3 Liste des livres depuis 2014 ;
- 6.2.4 Liste des publications de "Toru Ishida" ;
- 6.2.5 Liste de tous les publieur ("publisher"), distinctes ;

6.3 Agrégats

- 6.3.1 Trier et projeter les titre des publications de "Toru Ishida" par titre de livre et par pages de début ;
- 6.3.2 Compter le nombre de ses publications ;
- 6.3.3 Compter le nombre de publications par auteur. Trier le résultat par ordre croissant ;

6.4 Map/Reduce

Produire une fonction Map et Reduce pour chacune des questions suivantes :

- 6.4.1 Exécuter la requête du Guide ;
- 6.4.2 Pour chacun des documents de type livres, donner le nombre de ses auteurs ;
- 6.4.3 Pour chaque document ayant "booktitle" (chapitre) publié par Springer, donner le nombre de ses chapitres en a plus d'un.
Attention, la fonction reduce n'est évaluée que lorsqu'il y a au moins 2 documents pour une même clé. Il est donc nécessaire d'appliquer un filtre après génération du résultat.
- 6.4.4 Pour chaque publieur "Springer", donner le nombre de publication par année
- 6.4.5 Pour chaque clé "publisher & année" (pour ceux qui ont un publisher), donner le nombre de publications
Attention, la clé du emit doit être un document
- 6.4.6 Pour l'auteur "Toru Ishida", donner le nombre de publication par année
- 6.4.7 Pour l'auteur "Toru Ishida", donner le nombre moyen de pages pour ses articles (type Article)
- 6.4.8 Pour chaque auteur, donner le titre de ses publications
Attention, la sortie de map et du reduce doivent être des documents (pas des tableaux)
- 6.4.9 Pour chaque auteur, donner le nombre de publications associé à chaque année
- 6.4.10 Pour le publieur "Springer", donner le nombre d'auteurs différents par année
- 6.4.11 Compter les publications de plus de 3 auteurs ;
- 6.4.12 Donner pour chaque publieur, donner le nombre moyen de pages par publication ;
- 6.4.13 Pour chaque auteur, donner le minimum et le maximum des années, ainsi que le nombre de publication totale ;