

**TP 1 de Bases de données H-2019**

**OUSSAMA LOURHMATI**

**Date de remise :** **Voir Léa**

**Modalité de remise :** fichiers .sql de création de la base et .doc pour les requêtes SQL sur Léa.

La base de données concerne la gestion d'une bibliothèque scientifique. Elle permet de répertorier les articles et l

es livres écrit par des scientifiques membres d'organismes de recherche. Une liste de mot-clé est associée à chaque article. Les articles sont publiés dans des journaux scientifiques. Les journaux et les livres sont publiés par un éditeur.

Les attributs soulignés appartiennent à la clé primaire, les attributs précédés de **#** sont des clés étrangères. Le schéma relationnel de cette base est le suivant :

ARTICLE **( idArticle**, #idJournal, titre, pageDebut, pageFin, volume, serie )

JOURNAL ( **idJournal**, nomJournal, #idEdition, anneeCreation, proprietaire )

AUTEUR ( **idAuteur**, nomAuteur, prenom, courriel, siteWeb )

ORGANISME ( **idOrganisme**, nomOrganisme, telephone, pays )

LIVRE ( **idLivre**, titre, #idEdition, annee, nbrePage )

MOTCLES ( **idMot**, mot, descriptif )

EDITION ( **idEdition**, nom, telephone, fax, pays ) ARTICLE\_AUTEUR ( #**idArticle, #idAuteur** )

AUTEUR\_ORGANISME ( #**idAuteur, #idOrganisme** )

LIVRE\_AUTEUR ( #**idLivre, #idAuteur** ) ARTICLE\_MOTCLES ( #**idArticle, #idMot** )

# Langage de définition des données /20

Ecrire les fichiers de commandes SQL permettant de créer et de supprimer les tables précédentes. Le type de chacun des attributs est spécifié dans le fichier TP1\_Bibliotheque\_etudiant.sql

Ne pas oublier de préciser pour chaque table les clés primaires et clés étrangères.

# Langage de manipulation des données /5

1. Insérer les données du fichier insertBiblio.sql dans les tables correspondantes. 2. Insérer un nouvel auteur en donnant une valeur pour chacun des attributs. /1

INSERT INTO AUTEUR(IDAUTEUR, NOMAUTEUR, PRENOM, EMAIL, SITEWEB)

VALUES (17, 'Lourhmati', 'Oussama', 'lourhmati@gmail.com','https://www.w3schools.com');

1. Insérer un nouvel auteur en indiquant uniquement son nom et son idAuteur.

/1

INSERT INTO AUTEUR(IDAUTEUR, NOMAUTEUR, PRENOM)

VALUES (18, 'Trump', 'Donald');

1. Supprimer en une commande les deux auteurs que vous venez d’insérer. /1

DELETE FROM AUTEUR

WHERE idauteur in (17,18);

1. Modifier les noms des éditeurs pour la première lettre du nom soit en majuscule et les autres lettres en minuscule (fonction INITCAP). /2

UPDATE EDITION

SET EDITION.nom = INITCAP (EDITION.nom);

**III) Interrogation de la base /13**

Ecrire les requêtes SQL permettant d’obtenir les résultats suivants:

# Requête simple

1. La liste des noms d'éditeurs;

SELECT nom nom\_editeur

FROM EDITION;

1. Les pays des éditeurs, chaque pays n’apparaitra qu’une fois dans l’ensemble résultat ;

SELECT DISTINCT pays pays\_edition

FROM EDITION;

1. Tous les titres d'articles triés par ordre alphabétique;

SELECT titre titre\_article

FROM ARTICLE

ORDER BY titre;

1. Les livres parus après 1990;

SELECT \*

FROM livre

WHERE annee > 1990;

1. Les journaux dont le nom contient le mot «pattern», sans tenir compte de la casse (UPPER, LOWER) ;

SELECT \*

FROM journal

WHERE UPPER(nomjournal) like UPPER('%pattern%');

1. Le nom des auteurs et le site web des auteurs ayant un site web;

SELECT nomauteur, siteweb

FROM AUTEUR

WHERE siteweb IS NOT NULL;

1. Le nombre de journaux;

SELECT COUNT(idjournal) nombre\_journaux

FROM JOURNAL;

1. Pour chaque article, son titre et sa longueur en nombre de pages ;

SELECT titre titre\_article, (pagefin - pagedebut) nombre\_pages

FROM ARTICLE;

# Sous-requête, Jointure /13

1. Les identifiants des articles écrits par ‘*Adams’*;

SELECT ARTICLE.idarticle, ARTICLE.idjournal, article\_auteur.idauteur

FROM ARTICLE

JOIN ARTICLE\_AUTEUR ON ARTICLE.idarticle = ARTICLE\_AUTEUR.idarticle

JOIN AUTEUR ON article\_auteur.idauteur = auteur.idauteur

WHERE auteur.nomauteur = 'Adams';

1. Pour chaque article, son titre et le nom du journal dans laquelle il a été publié;

SELECT ARTICLE.titre, journal.nomjournal

FROM ARTICLE

JOIN JOURNAL ON article.idjournal = journal.idjournal;

1. Tous les auteurs qui n’ont pas publiés de livre;

SELECT \*

FROM AUTEUR

LEFT JOIN LIVRE\_AUTEUR ON LIVRE\_AUTEUR.IDAUTEUR = AUTEUR.IDAUTEUR

WHERE AUTEUR.IDAUTEUR NOT IN (SELECT IDAUTEUR FROM LIVRE\_AUTEUR);

1. La liste des articles dont un des mots clés est ‘*segmentation’* ;

SELECT \*

FROM ARTICLE

JOIN article\_motcles ON article.idarticle = article\_motcles.idarticle

JOIN motcles ON article\_motcles.idmot = motcles.IDMOT

WHERE MOTCLES.MOT LIKE '%segmentation%';

1. Tous les auteurs qui ont publiés dans le journal ‘*Information Processing Letters’* ;

SELECT \*

FROM AUTEUR

JOIN ARTICLE\_AUTEUR ON auteur.idauteur = article\_auteur.idauteur

JOIN ARTICLE ON article\_auteur.idarticle = article.idarticle

JOIN JOURNAL ON article.idjournal = journal.idjournal

WHERE journal.nomjournal LIKE 'Information Processing Letters';

1. Les noms des éditeurs qui ont publiés des livres ces 23 dernières années (

EXTRACT*(*YEAR *FROM* SYSDATE*)* );

SELECT edition.nom, livre.annee

FROM EDITION

JOIN LIVRE ON edition.idedition = livre.idedition

WHERE LIVRE.ANNEE > (SELECT EXTRACT(YEAR FROM SYSDATE) - 33

FROM DUAL);

1. Les éditeurs qui publient au moins un journal et qui ont déjà publié au moins un livre;

SELECT \*

FROM EDITION

JOIN JOURNAL ON edition.idedition = journal.idedition

JOIN LIVRE ON edition.idedition = livre.idedition

WHERE edition.idedition IN (SELECT IDEDITION FROM JOURNAL)

AND edition.idedition IN (SELECT IDEDITION FROM LIVRE);

1. Le titre des articles et le nom du journal pour les articles publiés dans la plus ancienne des journaux;

SELECT ARTICLE.TITRE, JOURNAL.NOMJOURNAL

FROM ARTICLE

JOIN JOURNAL ON ARTICLE.IDJOURNAL = journal.idjournal

ORDER BY journal.annee

FETCH FIRST ROW ONLY;

# Group By, Having /12

17. Toutes les paires d’auteur qui ont écrit un article ensemble. Chaque couple d’auteur n’apparaît qu’une fois dans l’ensemble résultat;

SELECT auteur.nomauteur, article.titre

FROM AUTEUR

JOIN ARTICLE\_AUTEUR ON article\_auteur.idauteur = auteur.idauteur

JOIN ARTICLE ON ARTICLE\_AUTEUR.IDARTICLE = article.idarticle

WHERE article.titre IN (

SELECT ARTICLE.TITRE

FROM ARTICLE

JOIN ARTICLE\_AUTEUR ON article.idarticle = ARTICLE\_AUTEUR.IDARTICLE

GROUP BY ARTICLE.TITRE

HAVING COUNT(ARTICLE\_AUTEUR.IDAUTEUR) = 2)

GROUP BY AUTEUR.NOMAUTEUR, ARTICLE.TITRE

ORDER BY ARTICLE.TITRE;

18. Le nombre de livres par auteur;

SELECT COUNT(LIVRE.IDLIVRE) NOMBRE\_LIVRE, auteur.nomauteur

FROM AUTEUR

JOIN LIVRE\_AUTEUR ON livre\_auteur.idauteur = auteur.idauteur

JOIN LIVRE ON livre.idlivre = livre\_auteur.idlivre

GROUP BY auteur.nomauteur;

1. Pour chaque auteur, son identifiant, son nom ainsi que le nombre d'organismes auxquels il est rattaché;

SELECT AUTEUR.IDAUTEUR, AUTEUR.NOMAUTEUR, COUNT(organisme.idorganisme) NOMBRE\_ORGANISMES

FROM AUTEUR

JOIN AUTEUR\_ORGANISME ON AUTEUR.IDAUTEUR = AUTEUR\_ORGANISME.IDAUTEUR

JOIN ORGANISME ON auteur\_organisme.idorganisme = organisme.idorganisme

GROUP BY auteur.nomauteur, AUTEUR.IDAUTEUR;

1. Le nombre d’articles publiés pour chaque membre d'un organisme Canadien;

SELECT DISTINCT AUTEUR.NOMAUTEUR, COUNT(ARTICLE\_AUTEUR.IDARTICLE) NOMBRE\_ARTICLES

FROM AUTEUR

JOIN auteur\_organisme ON auteur.idauteur = auteur\_organisme.idauteur

JOIN ORGANISME ON AUTEUR\_ORGANISME.idorganisme = organisme.idorganisme

JOIN ARTICLE\_AUTEUR ON auteur.idauteur = article\_auteur.idauteur

WHERE organisme.pays LIKE 'Canada'

GROUP BY auteur.nomauteur, organisme.nom;

1. Les articles qui sont associés à deux mots clés;

SELECT article.titre

FROM ARTICLE

JOIN ARTICLE\_MOTCLES ON article.idarticle = article\_motcles.idarticle

JOIN motcles ON article\_motcles.idmot = motcles.idmot

GROUP BY article.titre

HAVING COUNT(motcles.mot)=2;

1. Les livres qui ont plus de deux auteurs;

SELECT livre.titre

FROM livre

JOIN LIVRE\_AUTEUR ON livre.idlivre = livre\_auteur.idlivre

JOIN AUTEUR ON livre\_auteur.idauteur = auteur.idauteur

GROUP BY LIVRE.TITRE

HAVING COUNT(AUTEUR.NOMAUTEUR) > 2;