
TD20 - base de données relationnelle

Exercice 1 ★ ★ ★ ★

Une société de matériel informatique possède un fichier qui archive les commandes effectuées par les clients et les produits achetés :

refcom	refprod	produit	prix	numclient	prénom	nom	adresse	date
1	3	pc	600	1	Jean	Bonnot	Lyon	10/01
2	2	souris	30	2	Amélie	Poulain	Valence	12/01
3	1	lampe	30	3	Robert	Poulain	Valence	12/01
4	4	clavier	20	1	Jean	Bonnot	Lyon	15/01
5	1	lampe	30	1	Jean	Bonnot	Lyon	16/01
6	2	souris	30	4	Robert	Lingot	Montélimar	20/01
7	4	clavier	20	2	Amélie	Poulain	Valence	21/01
8	5	lampe	50	4	Robert	Lingot	Montélimar	25/01
9	4	clavier	20	1	Jean	Bonnot	Lyon	25/01
10	3	pc	600	5	Désiré	Duval	Grenoble	30/01

1) Reporter ces données dans trois tables reliées entre elles :

- une table *clients* d'attributs *numclient*, *prénom*, *nom* et *adresse*,
- une table *produits* d'attributs *refprod*, *produit* et *prix*,
- une table *commandes* d'attributs *refcom*, *numclient*, *refprod* et *date*.

2) Préciser les clés primaires et étrangères de chaque table.

3) Pourquoi est-il mieux d'avoir 3 petites tables plutôt qu'une seule et grosse table ?

Exercice 2 ★ ★ ★ ★

On considère la table *clients* ci-dessous :

nom	prénom	date
Martin	Rémy	1970
Roux	Alice	1989
Roux	Philippe	1967
Durand	Arthur	1970
Dupont	Josette	1950
Martin	Blandine	1975

Quel est l'effet de la requête suivante sur la table *clients* ?

```
UPDATE clients SET date=1971 WHERE date=1970;
```

Exercice 3 ★ ★ ★ ★

On considère la table *clients* ci-dessous :

nom	prénom	date
Martin	Rémy	1970
Roux	Alice	1989
Roux	Philippe	1967
Durand	Arthur	1970
Dupont	Josette	1950
Martin	Blandine	1975

Quel est l'effet de la requête suivante sur la table *clients* ?

```
DELETE FROM clients WHERE nom="Roux" ;
```

Exercice 4 ★ ★ ★ ★

On considère la table *commandes* ci-dessous :

numcommande	marque	prix
1	Peugeot	20500
2	Audi	25300
3	Citroën	15700
4	Renault	12300
5	Peugeot	18000

Quel est le but de la requête suivante ?

```
SELECT numcommande,marque FROM commandes ;
```

Exercice 5 ★ ★ ★ ★

On considère la table *clients* ci-dessous :

numclient	ville	paiement
1	Nice	chèque
2	Montpellier	espèces
3	Marseille	cb
4	Lyon	cb
5	Nice	chèque

Quel est le but de la requête suivante ?

```
SELECT * FROM clients WHERE paiement="cb" ;
```

Exercice 6 ★ ★ ★ ★

On considère la table *clients* ci-dessous :

numclient	ville	paiement
1	Nice	chèque
2	Montpellier	espèces
3	Marseille	cb
4	Lyon	cb
5	Nice	cb

Quel est le but de la requête suivante ?

```
SELECT paiement,ville FROM clients WHERE ville="Nice" ;
```

Exercice 7 ★ ★ ☆ ☆

On considère les tables *etudiants* et *controles* ci-dessous :

Table etudiants

id_eleve	prénom	classe
1	Kilian	1A
2	Léo	1B
3	Zinédine	1A

Tables controles

id_controle	matière	note	id_eleve
M1	maths	15	1
M1	maths	7	2
M1	maths	12	3
P1	philo	10	1
P1	philo	8	2
P1	philo	16	3

1)Quels sont les enregistrements renvoyés par la requête ci-dessous ?

```
SELECT * FROM etudiants INNER JOIN controles ;
```

2)Quels sont les enregistrements renvoyés par la requête ci-dessous ?

```
SELECT * FROM etudiants INNER JOIN controles ON etudiants.id_eleve=controles.id_eleve
```

3)Laquelle des deux requêtes précédentes est la plus pertinente ?

4)Ecrire une requête qui affiche tous les couples (note,prénom) possibles en ordonnant les notes par ordre croissant.

5)Que renvoie la requête ci-contre : `SELECT AVG(note) FROM controles ;`

Exercice 8 ★ ★ ☆ ☆

Une compagnie aérienne dispose d'une base de données formée de deux tables :

- la table *pilotes* composée des attributs suivants :
 - numpil (de type INTEGER) : numéro du pilote,
 - nompil (de type TEXT) : nom du pilote,
 - aeroport (de type TEXT) : nom de l'aéroport où travaille le pilote,
 - salaire (de type INTEGER) : salaire du pilote.
- la table *avions* composée des attributs suivants :
 - numav (de type INTEGER) : numéro de l'avion,
 - marque (de type TEXT) : marque de l'avion,
 - capacite (de type INTEGER) : nombre de passagers maximal de l'avion,
 - aeroport (de type TEXT) : nom de l'aéroport où est basé l'avion.

Pour chacun des cas suivants, écrire une requête effectuant la tâche ci-dessous :

1)Extraire la liste des numéros d'avions dont la capacité est supérieure à 300.

2)Extraire les numéros et la marque des avions basés à Londres.

3)Extraire tous les enregistrements de la table *pilotes*.

4)Extraire le nom des pilotes qui travaillent à l'aéroport de Londres et dont le salaire est inférieur à 100000.

5)Mettre à jour la table *pilotes* en multipliant par 2 le salaire des pilotes parisiens.

Exercice 9 ★ ★ ☆ ☆

Dans la base de données d'une université, on dispose de trois tables gérant les notes des étudiants en L1 à l'examen de passage en L2 :

- la table *etudiants* composée des attributs suivants :
 - id_etudiant (de type INTEGER) : numéro d'identification de l'étudiant,
 - nom (de type TEXT) : nom de l'étudiant,
 - redoublant (de type TEXT) : oui ou non.
- la table *controles* composée des attributs suivants :
 - id_controle (de type TEXT) : numéro d'identification du controle,
 - matiere (de type TEXT),
 - coefficient (de type INTEGER).
- la table *evaluations* composée des attributs suivants :
 - id_etudiant,
 - id_controle,
 - note (de type INTEGER).

Les étudiants font deux épreuves d'anglais (A1 et A2), deux épreuves de philosophie (P1 et P2), deux épreuves d'économie (E1 et E2) et deux épreuves de maths (M1 et M2).

- 1) Imaginer un exemple d'enregistrement de la table *etudiants*.
- 2) Préciser les clés primaires et les clés étrangères de chaque table.
- 3) Pour chacun des cas suivants, écrire une requête SQL permettant d'effectuer la tâche ci-dessous :
 - a) créer les tables *etudiants*, *controles* et *evaluations*,
 - b) extraire le nombre total d'étudiants,
 - c) extraire parmi l'ensemble de toutes les notes récoltées à l'examen celle qui est la plus haute et celle qui est la plus basse,
 - d) extraire toutes les notes obtenues à l'examen, ordonnées dans l'ordre croissant,
 - e) obtenir le nom de tous les étudiants redoublants.
- 4) Expliquer les requêtes ci-dessous :
 - a) UPDATE *evaluations* SET note=10 WHERE note<10 ;
 - b) SELECT nom FROM *etudiants* ORDER BY nom ;
 - c) UPDATE *evaluations* SET note=note+1 FROM *controles* WHERE matiere="Maths" ;
 - d) SELECT note FROM *controles* INNER JOIN *evaluations* WHERE *controles*.id_controle=*evaluations*.id_controle AND *controles*.matiere="économie" ;
 - e) SELECT note FROM *etudiants* INNER JOIN *evaluations* WHERE nom="Durand" AND *etudiants*.id_etudiant=*evaluations*.id_etudiant ;

Exercice 10 ★ ★ ☆ ☆

Une boutique en ligne possède une base de données comportant les tables *articles*, *fournisseurs* et *achats* décrites ci-dessous.

articles(id_art,libelle,stock,prix)

fournisseurs(id_four,nom_four,adresse_four)

achats(id_four,id_art,prix,delai)

Pour chacun des cas suivants, écrire une requête effectuant la tâche ci-dessous :

-
- 1) Extraire les identifiants et libellés des articles dont le stock est inférieur à 10.
 - 2) Extraire la liste complète des articles dont le prix est compris entre 100 et 300.
 - 3) Extraire les noms et adresses des fournisseurs qui proposent des articles pour lesquels le délai d'approvisionnement est supérieur à 20 jours.
 - 4) Obtenir le nombre d'articles référencés.
 - 5) Obtenir la valeur financière du stock.

Exercice 11 ★ ★ ☆ ☆

On dispose d'une base de données comportant deux tables *vehicule* et *annonce* décrites ci-dessous.

- La table *vehicule* recense des informations sur les modèles de véhicules en vente sur le marché. Elle est composée des attributs suivants :
 - *id_vehicule* (de type INTEGER) : un code permettant d'identifier de façon unique chaque référence de véhicule (marque et modèle),
 - *marque* (de type TEXT) : le nom du constructeur du véhicule,
 - *modele* (de type TEXT) : le modèle du véhicule, le constructeur proposant en général plusieurs modèles de véhicules à la vente,
 - *prix_neuf* (de type INTEGER) : prix de vente du véhicule neuf.
- La table *annonce* regroupe des informations sur un grand nombre d'annonces de véhicules d'occasion. Chaque enregistrement correspond à une annonce et possède les attributs suivants :
 - *id_annonce* (de type INTEGER) : un code permettant d'identifier chaque annonce de façon unique,
 - *id_vehicule* (de type INTEGER) : l'identifiant du modèle de véhicule vendu, qui correspond à l'identifiant utilisé dans la table *vehicules*,
 - *annee* (de type INTEGER) : année de première mise en circulation du véhicule,
 - *km* (de type INTEGER) : nombre de kilomètres parcourus par le véhicule au moment de la revente,
 - *prix_occasion* (de type INTEGER) : prix de vente du véhicule d'occasion.

- 1) En justifiant brièvement, identifier une clef primaire dans chacune des tables *vehicule* et *annonce*, ainsi qu'une clef étrangère dans la table *annonce*.
- 2) Ecrire une requête SQL permettant d'extraire les noms de tous les modèles de véhicules mis en vente par le constructeur Dubreuil Motors.
- 3) Expliquer le fonctionnement de la requête SQL suivante et préciser l'effet éventuel de cette requête sur chacune des tables *vehicule* et *annonce*.

```
UPDATE annonce
SET prix_occasion = prix_neuf FROM annonce INNER JOIN vehicule
WHERE vehicule . id_vehicule = annonce . id_vehicule
AND vehicule . prix_neuf < annonce . prix_occasion
```

- 4) A l'aide d'une jointure, écrire une requête permettant d'obtenir, sur une même table, les enregistrements avec les attributs suivants :
 - l'identifiant de l'annonce *id_annonce*,
 - le kilométrage *km*,
 - le prix de vente du véhicule neuf *prix_neuf*,
 - le prix de l'annonce d'occasion *prix_occasion* ;

Réponses

Exercice 1

1)

Table clients

numclient	prenom	nom	adresse
1	Jean	Bonnot	Lyon
2	Amélie	Poulain	Valence
3	Robert	Poulain	Valence
4	Robert	Lingot	Montélimar
5	Désiré	Duval	Grenoble

Table produits

refprod	produit	prix
1	lampe	30
2	souris	30
3	pc	600
4	clavier	20
5	lampe	50

Table commandes

refcom	numclient	refprod	date
1	1	3	10/01
2	2	2	12/01
3	3	1	12/01
4	1	4	15/01
5	1	1	16/01
6	4	2	20/01
7	2	4	21/01
8	4	5	25/01
9	1	4	25/01
10	5	3	30/01

2) Table clients : clé primaire = numclient

Table produits : clé primaire = refprod

Table commandes : clé primaire = refcom, clés étrangères = numclient et refprod.

3) Quatre raisons de préférer les 3 petites tables :

- les 3 petites tables contiennent moins de données que la grosse (20+15+40=75 pour les petites contre 90 pour la grosse),
- la mise à jour d'une donnée est plus rapide (par exemple, changer l'adresse de Jean nécessite une mise à jour avec les petites tables contre 4 pour la grosse),
- une lecture des données facilitée avec les petites tables,
- les petites tables sont reliées entre elles, à l'aide des clés.

Exercice 2

A l'issue de cette commande, dans la colonne *date*, la valeur 1970 a été remplacée par 1971, ce qui donne :

nom	prénom	date
Martin	Rémy	1971
Roux	Alice	1989
Roux	Philippe	1967
Durand	Arthur	1971
Dupont	Josette	1950
Martin	Blandine	1975

Exercice 3

A l'issue de cette commande, on supprime les deux enregistrements qui possèdent la valeur "Roux" dans la colonne "nom", ce qui donne :

nom	prénom	date
Martin	Rémy	1971
Durand	Arthur	1971
Dupont	Josette	1950
Martin	Blandine	1975

Exercice 4

La requête sélectionne et extrait les enregistrements (1,Peugeot), (2,Audi), (3,Citroën), (4,Renault) et (5,Peugeot).

Exercice 5

Le programme recherche la valeur "cb" dans la colonne "paiement", puis extrait les deux enregistrements qui contiennent cette valeur : (3,Marseille,cb) et (4,Lyon,cb).

Exercice 6

Le programme recherche la valeur "Nice" dans la colonne "ville", puis extrait les deux enregistrements qui contiennent cette valeur, en se limitant au champ "paiement, ville", c'est-à-dire : (chèque,Nice) et (cb,Nice).

Exercice 7

1)La requête met en mémoire $3 \times 6 = 18$ enregistrements.

Ce sont les éléments du produit cartésien de la table *étudiants* par la table *contrôles* :

(1, Kilian, 1A, M1, maths, 15, 1), (1, Kilian, 1A', M1, maths, 7, 2),
(1, Kilian, 1A, M1, maths, 12, 3), (1, Kilian, 1A, P1, philo, 10, 1),
(1, Kilian, 1A, P1, philo, 8, 2), (1, Kilian, 1A, P1, philo, 16, 3),
(2, Léo, 1B, M1, maths, 15, 1), (2, Léo, 1B, M1, maths, 7, 2),
(2, Léo, 1B, M1, maths, 12, 3), (2, Léo, 1B, P1, philo, 10, 1),
(2, Léo, 1B, P1, philo, 8, 2), (2, Léo, 1B, P1, philo, 16, 3),
(3, Zinédine, 1A, M1, maths, 15, 1), (3, Zinédine, 1A, M1, maths, 7, 2),
(3, Zinédine, 1A, M1, maths, 12, 3), (3, Zinédine, 1A, P1, philo, 10, 1),
(3, Zinédine, 1A, P1, philo, 8, 2), (3, Zinédine, 1A, P1, philo, 16, 3).

2)La requête sélectionne dans les 18 enregistrements précédents ceux pour lesquels *id_eleve* est le même dans la table *étudiants* et dans la table *contrôles*

(1, Kilian, 1A, M1, maths, 15, 1), (1, Kilian, 1A, P1, philo, 10, 1),
(2, Léo, 1B, M1, maths, 7, 2), (2, Léo, 1B, P1, philo, 8, 2),
(3, Zinédine, 1A, M1, maths, 12, 3), (3, Zinédine, 1A, P1, philo, 16, 3).

3)La deuxième requête est la plus pertinente car elle met en relation les tables, grâce à l'attribut en commun "id_eleve".

4)SELECT note,prenom FROM etudiants INNER JOIN controles
ON etudiants.id_eleve = controles.id_eleve
ORDER BY note;

La requête extrait alors les couples (7,Léo), (8,Léo), (10,Kilian), (12,Zinédine), (15,Kilian) et (16,Zinédine).

5)La requête renvoie la moyenne des notes obtenues par les trois étudiants,

c'est-à-dire : $\frac{15 + 7 + 12 + 10 + 8 + 16}{6} = 11,33$.

Exercice 8

1)SELECT numav FROM avions WHERE capacite >300

2)SELECT numav,marque FROM avions WHERE aeroport="Londres"

3)SELECT * FROM pilotes

4)SELECT nompil FROM pilotes WHERE aeroport="Londres" AND salaire<100000

5)UPDATE pilotes SET salaire=2*salaire WHERE aeroport="Paris"

Exercice 9

1)(1,"Durand", "oui") est un exemple d'enregistrement de la table *etudiants*.

2)Recherche des clés :

– l'attribut id_etudiant est une clé primaire de la table *etudiants*, puisque pour un enregistrement donné, la valeur de cet attribut détermine de façon unique la valeur des autres attributs,

– l'attribut id_controle est une clé primaire de la table *controles*,

– le couple d'attributs (id_etudiant,id_controle) est une clé primaire de la table *evaluations*. En effet, pour un enregistrement donné, la valeur de ce couple détermine de façon unique la note de l'étudiant au controle,

– les attributs id_etudiant et id_controle sont des clés étrangères de la table *evaluation* car ce sont respectivement la clé primaire de la table *etudiants* et la clé primaire de la table *controles*.

3)a)CREATE TABLE etudiants(id_etudiant INTEGER,nom TEXT,redoublant TEXT)
CREATE TABLE controles(id_controle INTEGER,matiere TEXT,coefficient
INTEGER)

CREATE TABLE evaluations(id_etudiant INTEGER,id_controle INTEGER,note
INTEGER)

b)SELECT COUNT(*) FROM etudiants

c)Pour obtenir la note la plus haute : SELECT MAX(Note) FROM evaluations

Pour obtenir la note la plus basse : SELECT MIN(Note) FROM evaluations

d)SELECT note FROM evaluations ORDER BY note

e)SELECT nom FROM etudiants WHERE redoublant="oui"

4)a)La requête met à jour la table *evaluations* en remplaçant toutes les notes en-dessous de la moyenne par 10.

b)La requête extrait le nom de tous les étudiants par ordre alphabétique.

c)La requête fait une mise à jour de la table évaluations en mettant 1 point plus à toutes les notes de maths.

d) La requête effectue une jointure des tables *controles* et *evaluations* :

- elle extrait de cette jointure tous les sextuplets formés d’un enregistrement de la table *controles*, suivi d’un enregistrement de la table *evaluations*,
- de ces sextuplets, elle ne garde que ceux pour lesquels le numéro d’identification du controle est le même et pour lesquels la matière est l’économie, puis affiche la note de chacun de ces sextuplets.

Le but de la requête est d’extraire toutes les notes d’économie obtenues à l’examen.

e) Même idée qu’en 4)d). La requête extrait toutes les notes de l’élève Durand.

Exercice 10

1) `SELECT id_art, libelle FROM articles WHERE stock < 10`

2) `SELECT * FROM articles WHERE prix > 100 AND prix < 300`

3) `SELECT nom_four, adresse_four FROM fournisseurs INNER JOIN achats ON fournisseurs.id_four = achats.id_four AND delai > 20`

4) `SELECT COUNT(*) FROM articles`

5) `SELECT SUM(stock * prix) FROM articles`

Exercice 11

1) Recherche des clés :

- l’attribut `id_vehicule` est une clé primaire de la table *vehicule*,
- l’attribut `id_annonce` est une clé primaire de la table *annonce*,
- l’attribut `id_vehicule` de la table *annonce* est une clé étrangère car c’est la clé primaire d’une autre table (la table *vehicule*).

2) `SELECT modele FROM vehicule WHERE marque = "Dubreuil Motors"`

Remarque

La commande `SELECT DISTINCT` à la place de `SELECT` permettrait ici que les noms de modèle n’apparaissent qu’une seule fois (voir exercice 2025).

- 3) – la commande `UPDATE` signifie qu’on fait une mise à jour des enregistrements de la table *annonce*,
- la commande `INNER JOIN` fait une jointure des tables *annonce* et *vehicule*,
 - par la commande `WHERE`, la requête répertorie les enregistrements de la jointure qui ont le même identifiant de véhicule,
 - par la commande `AND`, la requête ne garde que les enregistrements pour lesquels le prix neuf de la table *vehicule* est inférieur au prix occasion de la table *annonce*,
 - par la commande `SET`, pour chacun des couples d’enregistrements précédents, la requête remplace le prix d’occasion de la table *annonce* par le prix neuf de la table *vehicule*.

Autrement dit, cette requête permet de plafonner le prix de vente d’occasion en s’assurant qu’il ne dépasse pas le prix neuf.

Remarque

A l’issue de la requête, la table *vehicule* n’est pas modifiée.

4)

```
SELECT id_annonce , km , prix_neuf , prix_occasion
      FROM annonce INNER JOIN vehicule
              jointure
      ON annonce . id_vehicule = vehicule . id_vehicule
```