

## Exercice 1.

On considère la BD suivante.

**Table Vendeur**

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

**Table Client**

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

1. Écrivez une requête SQL pour préparer une liste avec le nom du vendeur, le nom du client et leurs villes pour les vendeurs et les clients qui appartiennent à la même ville.
2. Estimer le nombre d'opérations que fait la requête.
3. Votre requête est-elle optimisée ?

## Exercice 2.

On considère la BD suivante.

**Table Commandes**

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001

### Table Client

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

1. Écrivez une requête SQL pour extraire une liste avec le numéro de commande, le montant de l'achat, le nom du client et leurs villes pour les commandes dont le montant de la commande est compris entre 500 et 2000.
2. Estimer le nombre d'opérations que fait la requête.
3. Votre requête est-elle optimisée ?
4. Proposer une requête optimisée au cas où.

### Exercice 3.

On considère la BD suivante.

#### Table Client

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

#### Table Vendeur

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

5. Écrivez une instruction SQL pour savoir quel vendeur travaille pour quel client.
6. Estimer le nombre d'opérations que fait la requête.
7. Dessiner l'arbre d'exécution de la requête.
8. Votre requête est-elle optimisée ? Proposer une requête optimisée au cas où.

#### Exercice 4.

On considère la BD suivante.

**Table Client**

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

**Table Vendeur**

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

1. Proposer une requête SQL pour trouver la liste des clients qui ont désigné un vendeur pour leur travail qui reçoit une commission supérieure à 12%.
2. Estimer le nombre d'opérations que fait la requête.
3. Dessiner l'arbre d'exécution de la requête.
4. Votre requête est-elle optimisée ? Proposer une requête optimisée au cas où.
5. Proposer une requête SQL pour trouver la liste des clients ayant désigné un vendeur pour leur travail qui ne vit pas dans la même ville où vit leur client et ayant reçu une commission supérieure à 12%.

#### Exercice 5.

On considère la BD suivante.

**Table Client**

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

**Table Vendeur**

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

1. Proposer une fragmentation correcte de la base de données par rapport au grade.
2. Pourrait-on fragmenter la table CLIENT par rapport au à "customer\_id"?
3. On suppose dans un autre cas de figure dont l'objectif c'est de garantir l'accessibilité des données au niveau géographique. Est-ce qu'une fragmentation verticale permettrait de résoudre le problème?
4. Proposer une alternative de fragmentation au cas où ce serait nécessaire.