

SQL

Exercices

Exercice 1: Gestion de livres

Soit la table tblLivre, utilisée pour stocker les livres d'une bibliothèque:

Nom du champ	Type de données	Description
idLivre	Numérique	Numéro du livre
dtTitre	Texte	Titre du livre
dtAuteur	Texte	Auteur du livre
dtGenre	Texte	Catégorie, p.ex. roman, conte...
dtLieu	Texte	Salle où se trouve le livre; p.ex. S1, S2
dtLangue	Texte	Langue du livre
dtDateAchat	Date/Heure	Date d'achat du livre
dtPrixAchat	Numérique	Prix d'achat du livre ; 0 si le livre était un cadeau
dtNPages	Numérique	Nombre de pages du livre

Trouvez le code SQL pour les requêtes suivantes:

1. Affichez le numéro, le titre et l'auteur pour tous les romans.
2. Affichez les livres où le prix d'achat n'est pas indiqué.
3. Affichez les livres moins chers que 10€.
4. Affichez les romans allemands triés suivant l'auteur.
5. Affichez tous les romans allemands et français (deux solutions).
Pour tester vos requêtes, veillez à ce qu'il y ait dans votre base de données des
 - romans français
 - romans allemands
 - romans ni français ni allemands
 - 'non-romans' allemands ou français
 - 'non-romans' ni allemands ni français
6. Affichez les livres dont le prix d'achat se situe entre 10€ et 20€ (deux solutions).
7. Quelles données sont nécessaires pour tester entièrement la requête précédente ?
8. Affichez les livres dont le prix d'achat ne se situe pas entre 10€ et 20€ (deux solutions).
9. Affichez tous les livres à l'exception des livres informatiques français de moins de 50€
10. Quelles données sont nécessaires pour tester entièrement la requête précédente ?
11. Affichez toutes les langues présentes parmi les livres de la bibliothèque.
12. Affichez le numéro du livre "Windows XP" de "Pierre Godefroid" en version française.
13. Supposons que ce numéro soit 38285, augmentez de 10 le nombre de pages de ce livre.
14. Insérez le nouveau livre suivant:

340	MS-Access 2003	Ken Getz	Informatique	S01	ANG	15/1/03	35.25	985
-----	----------------	----------	--------------	-----	-----	---------	-------	-----

15. Insérez le nouveau livre suivant:

341	MS-Office	Ken Getz	Informatique	ANG
-----	-----------	----------	--------------	-----

16. Tous les livres des genres Informatique et Histoire reçoivent une réduction de 5%.

17. Comptez le nombre de livres qui ont moins de 300 pages.

18. Affichez la date du dernier achat.

19. Affichez une liste qui contient toutes les langues dont il existe des romans plus chers que 8€ (au moins un roman plus cher que 8€).

20. Affichez pour chaque livre le prix d'une page.

21. Supprimez tous les livres informatiques achetés il y a plus de deux ans.

22. Supprimez tous les romans à l'exception de ceux dont le prix se situe entre 10€ et 20€

23. Supprimez tous les livres à l'exception des romans français et anglais.

24. Expliquez la différence entre les trois variantes de l'option COUNT ci-dessous:

SELECT COUNT(*) FROM tblLivre;	SELECT COUNT(dtLangue) FROM tblLivre;	SELECT COUNT(DISTINCT dtLangue) FROM tblLivre;
-----------------------------------	--	---

25. Affichez le nombre de salles dans lesquelles il y a des livres

26. Affichez le nombre de livres

27. Affichez le nombre de romans

28. Affichez le nombre de livres pour chaque langue

29. Affichez le nombre de romans pour chaque langue

30. Affichez le nombre de langues pour chaque catégorie

31. Affichez pour chaque salle le prix moyen des livres.

32. Affichez pour chaque salle la moyenne des prix de page des livres de cette salle.

33. Affichez les salles dans lesquelles il y a plusieurs livres

34. Affichez le nombre de livres par salle en commençant par la salle qui présente le plus de livres

35. Affichez pour chaque catégorie le nombre de livres achetés par jour en commençant par l'achat le plus récent

36. Affichez le nombre de romans dans chaque salle

37. Affichez les catégories dont la valeur dépasse 200€

38. Affichez pour chaque salle le nombre de livres français sans pour autant prendre en compte les salles où il y en a moins de 10
39. Affichez les auteurs ayant plusieurs livres dans la bibliothèque et dont la somme ne dépasse pas 50€
40. Affichez les catégories pour lesquelles le livre le plus cher ne dépasse pas 50€
41. Affichez les titres uniques
42. Affichez les salles dans lesquelles tous les livres sont plus chers que 100€
43. Affichez les catégories pour lesquelles il n'y a qu'un seul livre
44. Affichez les auteurs dont il y a plusieurs livres dans la bibliothèque
45. Affichez les auteurs dont il y a plusieurs fois le même titre dans la bibliothèque
46. Affichez les auteurs qui n'ont qu'un seul titre dans la bibliothèque
47. Affichez les auteurs dont il y a des livres dans plusieurs catégories
48. Affichez les auteurs qui ont plusieurs livres dans une même catégorie
49. Affichez les auteurs dont tous les titres sont dans une même catégorie, sans afficher pour autant les auteurs n'ayant qu'un seul livre dans la bibliothèque
50. Décrivez quels livres sont affichés par les requêtes suivantes :
 - a)

```
SELECT *  
FROM tblLivre  
WHERE dtGenre='Roman'  
      AND dtLangue='Français' OR dtLangue='Anglais';
```
 - b)

```
SELECT *  
FROM tblLivre  
WHERE dtGenre='Roman'  
      AND (dtLangue='Français' OR dtLangue='Anglais');
```
51. Expliquez ce qui est affiché par la requête suivante :

```
SELECT COUNT(dtAuteur)  
FROM tblLivre;
```
52. Expliquez ce qui est affiché par la requête suivante et donnez une solution alternative :

```
SELECT COUNT(*) - COUNT(dtAuteur)  
FROM tblLivre  
WHERE dtGenre='Roman' ;
```

53. Supposez disposer des données suivantes :

idLivre	...	dtGenre	dtLangue
100		Roman	Chinois
101		Roman	Français
102		Roman	Anglais
103		Conte	Chinois
104		Conte	Français
105		Conte	Anglais

Donnez les numéros des livres non supprimés par les requêtes suivantes :

- a)

```
DELETE FROM tblLivre
WHERE dtGenre<>'Roman' OR dtLangue<>'Français' AND
      dtLangue<>'Anglais' ;
```
- b)

```
DELETE FROM tblLivre
WHERE dtGenre<>'Roman' AND dtLangue<>'Français' OR
      dtGenre<>'Roman' AND dtLangue<>'Anglais' ;
```
- c)

```
DELETE FROM tblLivre
WHERE dtGenre<>'Roman' AND (dtLangue<>'Français' OR
      dtLangue<>'Anglais') ;
```
- d)

```
DELETE FROM tblLivre
WHERE dtGenre<>'Roman' AND (dtLangue='Français' OR
      dtLangue='Anglais') ;
```
- e)

```
DELETE FROM tblLivre
WHERE dtGenre<>'Roman' AND dtLangue='Français' OR
      dtLangue='Anglais' ;
```
- f)

```
DELETE FROM tblLivre
WHERE dtGenre<>'Roman' AND dtLangue='Français' AND
      dtLangue='Anglais' ;
```
- g)

```
DELETE FROM tblLivre
WHERE NOT (dtGenre='Roman' AND dtLangue='Français')
      AND NOT (dtGenre='Roman' AND dtLangue='Anglais' ;
```

54. Affichez les titres français commençant par un article

55. Affichez les titres constitués par un seul mot

56. Expliquez la commande

```
SELECT DISTINCT dtGenre
FROM tblLivre
WHERE dtPrixAchat>200;
```

57. Affichez les catégories dont un seul livre dépasse 200€
58. Affichez les catégories dont tous les livres dépassent 200€
59. De quels auteurs y a-t-il plusieurs livres dans une même salle ?
60. Affichez les salles dans lesquelles il y a plusieurs livres de même prix.
61. Affichez les livres qui sont seuls dans leur catégorie
62. Affichez le nombre de livres français par salle sans pour autant prendre en compte les salles où il y a moins de 10 livres (toutes langues confondues)
63. Affichez les livres qui furent achetés en même temps que d'autres livres
64. Affichez les auteurs dont des livres furent achetés en même temps que des livres d'autres auteurs
65. Affichez les auteurs qui ont écrit exclusivement des livres français

Exercice 2: Gestion des clients

Une société utilise la table tblClient pour gérer ses clients.

Nom du champ	Type de données	Description
idClient	Numérique	Numéro du client
dtNom	Texte	Nom du client
dtPrenom	Texte	Prénom du client
dtSexe	Texte	Sexe du client. valeurs possibles: 'F' et 'M'
dtAdresse	Texte	Rue et numéro
dtCP	Texte	Code postal
dtLocalite	Texte	Localité
dtNoTel	Texte	Numéro de téléphone
dtNoFax	Texte	Numéro de fax ; vide si le client n'a pas de Fax
dtDateNaiss	Date/Heure	Date de naissance du client
dtBonClient	Booléen (Logique)	
NombreEnfants	Numérique	Nombre d'enfants ; vide si non connu

Trouvez le code SQL pour les requêtes suivantes:

1. Affichez les adresses postales des clients habitant à Walferdange.
2. Affichez les clients mineurs (*minderjährig*) de Walferdange.
3. Quelles données sont nécessaires pour tester entièrement la requête précédente ?
4. Affichez tous les clients sauf les mineurs de Walferdange.
5. Insérez le client suivant dans la table:

6	Heinen	Edmond	M	12, Am Gaard	2021	Ettelbruck	217070	217071	12/01/1954	non
---	--------	--------	---	--------------	------	------------	--------	--------	------------	-----

Indiquez la requête correcte pour ajouter ce client lorsque vous ignorez le numéro de fax

Indiquez la requête correcte pour ajouter ce client lorsque vous savez que le client ne possède pas de fax.

6. Affichez les localités dans lesquelles il y a des clients.
7. Afficher les bons clients masculins, à l'exception de ceux habitant à Luxembourg, Esch/Alzette et Ettelbruck.
8. Affichez par sexe, le nombre de clients nés après le 31/12/1985. Le champ qui affiche le nombre de clients doit porter l'en-tête 'qcpNombreClients'.
9. Afficher les clients habitant à Luxembourg. Triez le résultat de sorte à afficher en premier les femmes les plus jeunes et en terminant par les hommes les plus âgés.
10. Affichez le numéro client, le nom, le prénom et le code 'bon client' pour tous les clients masculins habitant à Diekirch ou Mersch. En ce qui concerne le code bon client, affichez l'en-tête 'Code spécial' au lieu d'afficher le nom du champ.

11. Tous les clients habitant à Ettelbruck, dans la Cité Patton, auront le nouveau code postal 8897.
12. Nous voulons "nettoyer" notre BD. En fait, il y a un certain nombre de clients dont l'adresse, le code postal, la localité le numéro de téléphone et le numéro de fax sont indéterminés. Ces enregistrements sont sans aucune valeur commerciale pour nous. Formulez une requête qui garde uniquement les clients pour lesquels on connaît:
soit le numéro de téléphone, soit le numéro de fax, soit l'adresse complète (*dtAdresse*, *dtCP*, *dtLocalite*).
Tous les autres clients sont effacés de la BD. ***
13. Comptez les clients masculins nés après le 1.1.1985.
14. Affichez pour chaque localité le nombre de bons clients ainsi que le nombre des autres clients. Triez la liste par ordre ascendant sur les localités.
15. Déterminez la date de naissance du client le plus vieux habitant à Luxembourg.
16. Affichez pour chaque localité le nombre de clients, en commençant par la localité qui en compte le plus.
17. On voudrait savoir s'il y a des clients qui sont nés un même jour. Affichez les dates respectives.
18. Affichez le nombre de localités dans lesquelles habitent des clients.
19. Affichez les localités où des clients ont un fax.
20. Affichez le nombre de clients sans fax par localité.
21. Affichez le nombre de fax par localité.
22. On voudrait afficher les clients dans l'ordre de distribution du courrier à leur adresse. En effet, le facteur parcourt une rue sur un côté (p.ex. numéros pairs) et la redescend sur le côté opposé (numéros impairs). Scindez le champ *dtAdresse* en deux (*dtNo* et *dtRue*). ***

Exercice 3: Gestion de concerts

Une agence de concerts utilise la table tblConcert :

Nom du champ	Type de données	Description
idConcert	Numérique	Numéro d'identification (Clé primaire)
dtArtiste	Texte	Artiste ou groupe qui donne le concert
dtDate	Date/Heure	Date du concert
dtDebut	Date/Heure	Début du concert (Heure)
dtLocalite	Texte	Localité du concert
dtLieu	Texte	Lieu du concert
dtTypeLieu	Texte	(Centre Culturel , Hall Sportif , Club etc.)
dtPrixTicket	Numérique	Prix d'un ticket
dtPlaces	Numérique	Total des places disponibles pour le concert
dtTicketsVendus	Numérique	Nombre de tickets déjà vendus pour le concert

Exprimez les requêtes suivantes en langage SQL:

1. Affichez toutes les informations pour les concerts à au moins 1000 places qui ne sont pas à Luxembourg.
2. Affichez toutes les informations pour les concerts à au moins 1000 places qui ne sont pas à Luxembourg.
3. Affichez l'artiste, la date et la localité des concerts qui se tiendront à Luxembourg pendant la deuxième moitié de l'année.
4. Insérez le concert suivant:

No.	Artiste	Date	Début	Localité	Lieu	Type Lieu	Places
710	Chris Rea	20/2/2006	20:00	Esch/Belval	Rockhal	Salle concert	5400

5. Affichez l'artiste, la date, la localité et le prix des concerts qui ont lieu dans un hall sportif ou un club à partir du 1/8/1998. Triez cette liste par ordre ascendant sur les types des lieux et à l'intérieur d'un type par ordre descendant sur le prix.
6. Quel était le prix moyen pour un concert en mois de juillet 1998 ? On ignorera les fêtes sous tente.
7. Classez les localités par ordre descendant sur le montant des recettes des concerts pour l'année 1998 (Recette d'un concert=Tickets vendus*Prix d'un ticket). Ignorez les localités pour lesquelles il n'y a pas encore de recettes.
8. Comptez le nombre de localités dans lesquelles a eu lieu un concert pendant les mois de juillet et août 1998.
9. Effacez tous les concerts qui ont eu lieu avant le 1/8/1998.
10. Affichez le nom de l'artiste, la date, la localité ainsi que le nombre de places encore disponibles pour les concerts qui auront lieu au mois de juillet 1998.

11. Affichez les artistes qui furent deux fois au Grand-Duché pendant une même année
12. Affichez la liste des localités, à l'exception de Luxembourg, dans lesquelles ont eu lieu au moins 2 concerts pendant la première moitié de l'année 1998.
13. Le concert numéro 108 est transféré du Centre Culturel Schlindermanderscheid au club "Den Atelier" à Luxembourg à la date et l'heure prévue initialement. Les tickets déjà vendus gardent leur validité et le prix d'un nouveau ticket ne change pas. Effectuez les modifications correspondantes dans la table.

Exercice 4: Assurance bagages

Une société d'assurances offre une formule 'Assurance Bagages'. Cette formule garantit pendant une durée limitée un remboursement intégral de la valeur des bagages avec contenu en cas de vol ou de perte.

Voici la BD utilisée pour gérer ce type d'assurances.

tblAgent
<u>idAgent</u>
dtNom
dtPrenom
dtAdresse
dtCP
dtLocalite
fiAgentGeneral

tblContrat
<u>idContrat</u>
dtDate
dtDebut
dtFin
dtPrime
dtPays
fiAgent
fiClient

tblClient
<u>idClient</u>
dtNom
dtPrenom
dtAdresse
dtCP
dtLocalite
dtTel

Remarque:

- dtDate est la date à laquelle le contrat est conclu ; dtDebut et dtFin définissent la période de validité du contrat.
- A chaque agent correspond un agent général ; les agents généraux sont leur propre agent général.

- 4-1. Affichez les numéros des contrats, le pays de destination et le nom de l'agent.
- 4-2. Affichez pour chaque contrat couvrant la France comme pays de destination le nom de l'agent.
- 4-3. Affichez les agents ayant conclu un contrat pour la France.
- 4-4. Affichez le numéro de contrat, les dates de début et de fin du contrat ainsi que le nom, prénom, adresse, code postal et localité du client pour tous les contrats qui couvrent entièrement la période du 14 juillet au 20 juillet 2014 et dont le pays de destination était l'Italie.
- 4-5. Déterminez la plus grande prime parmi celles où le pays de destination est la Belgique et l'agent n'est pas un agent général.
- 4-6. Affichez toutes les informations concernant les clients ayant un agent qui habite à Capellen.
- 4-7. Affichez le numéro de contrat, la prime, le nom et prénom du client ainsi que le nom et prénom de l'agent pour tous les contrats où l'agent a le même nom que le client.
- 4-8. Affichez les agents qui ont fait leur propre contrat
- 4-9. Affichez pour chaque client, le numéro de client, son nom, le nom de son agent et la somme des primes de tous les contrats qu'il a conclu avec cet agent.
- 4-10. Elaborez une liste qui affiche pour chaque agent le nombre de contrats par pays de destination.

- 4-11. Triez les agents par ordre descendant sur le nombre de contrats qu'ils ont conclus. En tenant uniquement compte des agents qui ont conclu au moins 2 contrats, affichez pour chaque agent, son numéro, son nom et prénom ainsi que le nombre de contrats qu'il a conclus.
- 4-12. Elaborez une liste qui affiche pour chaque agent le nombre de pays de ses contrats.
- 4-13. Affichez les numéros des clients dont toutes les primes dépassent 2000.-
- 4-14. Affichez les clients ayant conclu des contrats chez différents agents.
- 4-15. Affichez pour chaque agent la date du premier contrat conclu après le 1.1.2014
- 4-16. Affichez les agents avec leur agent général respectif. Triez sur les agents généraux.
- 4-17. Supposez que pour chaque contrat on voudrait indiquer un agent de réserve. Comment devriez-vous modifier la structure des données afin de pouvoir stocker cette information? Affichez pour chaque contrat l'agent principal et l'agent de réserve.
- 4-18. Affichez les numéros et la prime de tous les contrats conclus, suivis du nom de l'agent traitant. Utilisez les deux syntaxes possibles pour la jointure.
- 4-19. Affichez les numéros et la prime de tous les contrats conclus, suivis du nom de l'agent traitant, y compris les contrats traités directement avec l'assurance, c'est-à-dire sans agent.
- 4-20. Affichez tous les agents suivis du numéro et de la prime de ses contrats (une ligne par contrat). Veillez à afficher également les agents n'ayant pas fait de contrats.
- 4-21. On voudrait afficher pour chaque agent le nombre de ses contrats. Les agents sans contrat devront donc apparaître avec le total 0.
- 4-22. Affichez noms et adresses des agents et clients triés par ordre alphabétique.
- 4-23. Affichez noms et adresses des agents et clients en distinguant s'il s'agit de clients ou d'agent.
- 4-24. Affichez pour chaque ville le nombre de contacts (agents et clients confondus). Utilisez la dernière requête comme source.
- 4-25. Affichez le nom et prénom des clients habitant dans la même localité que le client numéro 11.
- 4-26. Affichez le nom et prénom des agents habitant dans la même localité que le client numéro 11.
- 4-27. Affichez, sans utiliser une sous-requête, le nom et prénom de tous les agents ayant conclu un contrat avec un client habitant à Mersch.
- 4-28. Affichez le nom et prénom de tous les agents ayant conclu un contrat avec un client habitant dans la même localité que le client numéro 11.

- 4-29. Indiquez le nom et la localité des clients ayant conclu un contrat dont la prime est strictement inférieure à celle du contrat numéro 1003.
- 4-30. Affichez les numéros des contrats dont la prime est inférieure à la prime moyenne.
- 4-31. Affichez la date début du contrat le plus cher.
- 4-32. Affichez les noms des clients habitant dans une ville où il y a un agent.
- 4-33. Affichez les noms des clients habitant dans une ville où il n'y a pas d'agent.
- 4-34. Affichez tous les agents ayant conclu un¹ contrat (2 solutions)
- 4-35. Affichez les agents qui n'ont pas conclu de contrat (2 solutions).
- 4-36. a) Affichez les numéros des clients ayant conclu des contrats chez plusieurs agents.
b) Affichez les noms des clients ayant conclu des contrats chez plusieurs agents.
c) Affichez les noms des clients ayant conclu des contrats chez plusieurs agents, ainsi que leurs agents respectifs et le nombre de contrats conclus avec chaque agent.
- 4-37. Affichez les numéros des clients dont la prime moyenne est supérieure à la moyenne des primes de tous les contrats.
- 4-38. Affichez la localité dans laquelle il y a le plus de clients.
- 4-39. a) Affichez les contrats dont la prime dépasse la prime moyenne
b) Affichez les contrats dont la prime dépasse la prime moyenne pour ce pays
- 4-40. Utilisez la technique de *requête corrélée* pour afficher les agents sans contrat.
- 4-41. Affichez pour chaque pays la localité comptant le plus de clients. (Vous devez ajouter un champ dtPays dans tblClient)
- 4-42. Affichez pour chaque localité le client avec le plus de contrats.
- 4-43. Affichez les localités dans lesquelles il y a plus d'agents que de clients
- 4-44. Expliquez la clause EXISTS
- 4-45. Affichez les noms des clients habitant dans une ville où il y a un agent, en employant la clause EXISTS.
- 4-46. Affichez les clients habitant dans une ville où il y a un agent au même prénom que ce client.
- 4-47. Affichez pour chaque pays la prime et la date du contrat le plus cher.

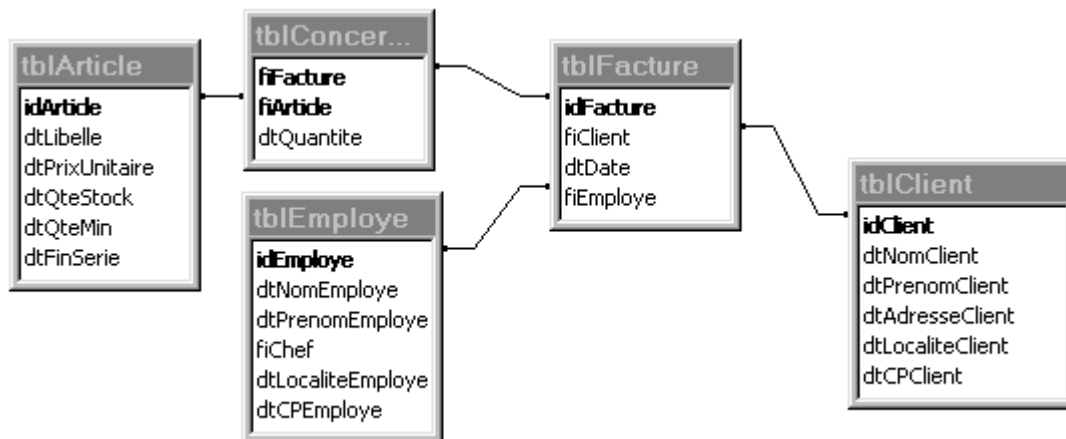
¹ un ou plusieurs

- 4-48. Affichez le nom et prénom des agents ayant conclu un contrat avec un client, qui a encore conclu un contrat avec au moins un autre agent.
- 4-49. Affichez les agents qui ont conclu uniquement des contrats avec des clients de leur propre localité.
- 4-50. Affichez les agents qui ont conclu un ou plusieurs contrats avec chaque client de leur propre localité.
- 4-51. Affichez les agents qui ont conclu tous les contrats des clients de leur propre localité.
- 4-52. Affichez les localités dans lesquelles des clients ont plusieurs agents.
- 4-53. Affichez les localités dans lesquelles tous les clients ont plusieurs agents.
- 4-54. Affichez les localités dans lesquelles aucun client n'a plusieurs agents.
- 4-55. Affichez les clients habitant dans une ville d'un agent qui n'ont pas fait de contrat.
- 4-56. Donnez une requête qui fournit le même résultat que la requête suivante, mais sans utiliser EXISTS.

```
SELECT *  
FROM tblAgent Main  
WHERE NOT EXISTS  
  (SELECT *  
   FROM tblContrat  
   WHERE fiAgent<>Main.idAgent)
```

Exercice 5: Facturation

Un magasin spécialisé dans la vente d'appareils électroménagers entretient la BD suivante afin de gérer la facturation.



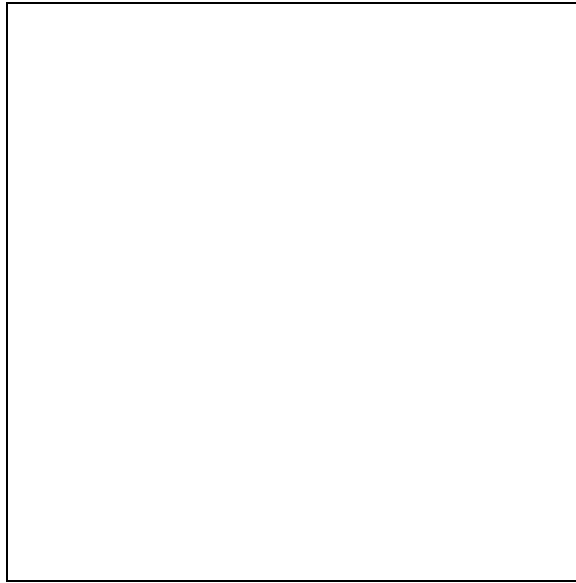
1. Affichez le libellé et le prix unitaire de l'article (des articles) qui est le plus cher.
2. Affichez le numéro de l'article ainsi que le libellé pour les articles moins cher que le prix moyen de tous les articles.
3. Affichez le numéro et la date de toutes les factures dont le client habite à Luxembourg.
4. Affichez le nom et le prénom des clients qui habitent à Luxembourg et qui sont concernés par une facture établie au cours du mois d'août 2012.
5. Affichez le numéro et le libellé des articles plus chers que le prix moyen, et dont furent achetés plusieurs à la fois.
6. Affichez le no des clients ayant déjà acheté un article plus cher que 300€.
7. Affichez le no des clients n'ayant pas acheté d'article plus cher que 300€.
8. Affichez le no des clients ayant acheté plusieurs articles plus chers que 300€.
9. Affichez le no des clients ayant uniquement acheté des articles plus cher que 300€
10. Affichez le no des clients ayant eu une ou plusieurs factures dépassant 300€.
11. Affichez le no des clients n'ayant pas eu de factures dépassant 300€.
12. Affichez le no des clients ayant eu plusieurs factures dépassant 300€.
13. Affichez le no des clients ayant une facture, qui concerne un seul article. La facture ne doit donc ni concerner plusieurs articles différents ni avoir une quantité >1 pour un seul article.
14. Affichez les articles qui n'ont pas encore été vendus.
15. Affichez le chiffre d'affaires par jour en commençant par le chiffre le plus élevé.

16. Affichez les journées présentant un chiffre d'affaire identique. (à partir de la requête précédente stockée ou par CREATE VIEW AS SELECT):
17. Affichez combien de clients n'ont pas encore de facture.
18. Affichez les numéros des factures suivis du nom de l'employé traitant.
19. Même question, mais en employant la syntaxe avec INNER JOIN.
20. Affichez toutes les factures, suivies, le cas échéant, du nom de l'employé traitant.
21. Affichez nom et adresse des clients et employé dans une seule liste, triée par ordre alphabétique.
22. Affichez les articles achetés par plusieurs clients.
23. Affichez les articles achetés par plusieurs clients, suivis par le nom des clients respectifs.
24. Affichez les articles non encore vendus.
25. Affichez les clients qui n'ont acheté qu'un seul article par facture. (un seul type d'article)
26. Affichez les clients qui ont une facture ne comprenant qu'un seul article. (un seul type d'article)
27. Affichez les articles qui ont déjà été vendus seuls. (sans autre article sur la même facture)
28. Affichez les clients qui ont acheté plusieurs articles par facture. (également plusieurs exemplaires du même article)
29. Affichez les clients qui ont toujours acheté plusieurs articles par facture. (également plusieurs exemplaires du même article)
30. Affichez pour chaque localité le client ayant le plus de factures.
31. Affichez les articles qui ont été achetés plusieurs fois par un même client.
32. Affichez les articles qui n'ont pas été achetés plusieurs fois par un même client.
33. Affichez pour chaque localité le meilleur client (celui qui a dépensé le plus).
34. Affichez les articles qui figurent sur chaque facture.
35. Affichez les articles qui n'ont jamais été vendus seuls. (pas d'autres articles sur la même facture)
36. Affichez pour chaque facture le ou les articles achetés en la plus grande quantité.
37. Affichez les clients ayant acheté la plus grande quantité jamais achetée pour cet article.
38. Affichez les clients ayant acheté un autre article à chaque facture.

39. Affichez les clients ayant acheté en moyenne une plus grande quantité que la quantité moyenne de l'article respectif.
40. Affichez les articles qui ont été achetés plusieurs fois par un même client.
41. Affichez les articles qui ont été achetés chaque fois en un seul exemplaire.

Exercice 6: Bibliothèque

Une bibliothèque utilise la BD suivante.



Remarques:

- Un auteur peut rédiger plusieurs livres et un livre peut être rédigé par plusieurs auteurs.
- La bibliothèque peut disposer de plusieurs exemplaires du même livre.
- Un prêt concerne un seul exemplaire d'un livre.
- Le champ *dtDateRetour* de la table *tblPrêts* reste indéterminé (NULL) tant que l'exemplaire emprunté n'a pas été retourné à la bibliothèque.

Formulez en SQL les requêtes suivantes:

1. Affichez le numéro, le titre et le genre de tous les livres allemands (Code=ALL). Classez la liste par ordre alphabétique sur le genre (p.ex. 'Roman' avant 'Technique') et à l'intérieur d'un genre par ordre ascendant sur les numéros.
2. Affichez une liste triée par ordre alphabétique de tous les genres de livres disponibles.
3. Affichez une liste de toutes les localités où habite un membre dont l'adresse contient l'abréviation 'bvd' , indiquant que le membre habite sur un boulevard.

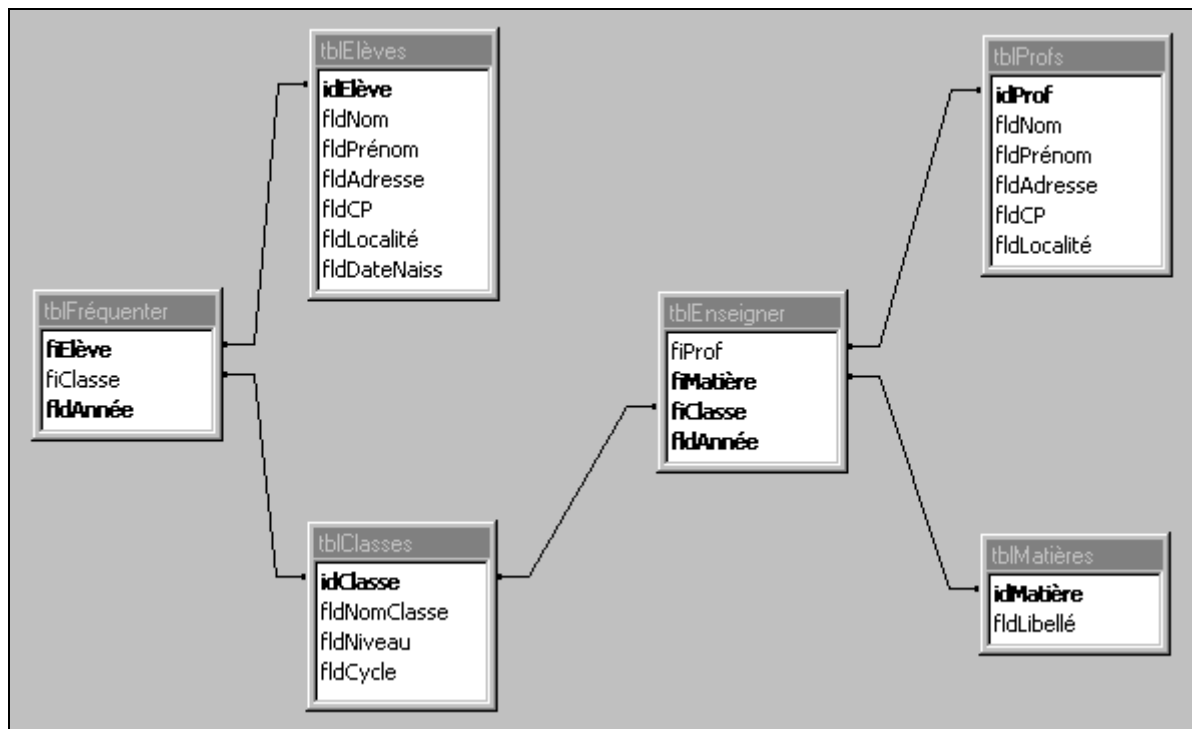
4. Affichez toutes les informations de la table *tblAuteurs* concernant les auteurs ayant une des nationalités suivantes.

Nationalité (Pays d'origine)	Code
Allemagne	ALL
Angleterre	ANG
France	FRA
Autriche	AUT
Italie	ITA
Suisse	SUI
Russie	RUS

5. Effacez tous les membres n'ayant pas encore effectué un prêt.
6. Affichez le nom, le prénom, l'adresse, le code postal et la localité de tous les membres habitant à Luxembourg ou à Esch-s-Alzette, n'ayant pas encore retourné un exemplaire emprunté.
7. Affichez pour chaque livre le titre, le genre, la langue et le nombre d'exemplaires disponibles (emprunté ou non). Triez la liste par ordre alphabétique sur la langue, sur le genre et finalement sur le titre. Le champ indiquant le nombre d'exemplaires disponibles doit avoir l'en-tête 'Exemplaires disponibles'.
8. Affichez le nom et le prénom des auteurs ayant écrit un livre français dont le titre contient le mot 'passage', et dont la bibliothèque possède au moins 3 exemplaires.
9. Affichez tous les livres (Titre et genre) de l'auteur 'Alexandre Dumas'. Triez la liste par ordre alphabétique sur le titre.
10. Affichez le nom, le prénom et le nombre de prêts effectués, pour tous les membres qui habitent à Esch-s-Alzette ou à Luxembourg, ayant déjà effectué au moins 2 prêts. Triez la liste par ordre alphabétique sur le nom.
11. Créez une liste qui affiche pour chaque exemplaire actuellement emprunté (pas encore retourné), le numéro du prêt, le numéro, le nom et le prénom du membre ayant emprunté le livre ainsi que le titre et le genre du livre en question. Triez la liste par ordre alphabétique sur le nom et le prénom du membre.
12. Quels sont les auteurs (Nom et prénom) ayant déjà écrit un livre ensemble avec l'auteur 'Margaret Gibson' ?
13. Quels sont les auteurs (Nom et prénom) n'ayant pas encore écrit un livre ensemble avec l'auteur 'Margaret Gibson' ?

Exercice 7: Gestion d'une école

Voici une BD qui représente une gestion simplifiée des cours d'un lycée technique.



Remarques:

- Une classe est représentée par un code interne (*idClasse*) , un nom de classe (*dtNomClasse*) tel que '13CG2' ou '11CM1' , un niveau (*dtNiveau*) tel que 10 pour la classe '10GE2' ou 13 pour '13CG1' , et un champ indiquant le cycle (*dtCycle*) avec les valeurs possibles 'Inférieur', 'Moyen' et 'Supérieur'.
- Nous supposons qu'un élève ne change pas de classe pendant l'année scolaire. Les champs *fIdElève* et *dtAnnée* forment donc la clé primaire de la table *tblFréquenter*. Cependant, un élève peut fréquenter la même classe pendant plusieurs années consécutives (redoublants).
- De même nous supposons qu'une matière est enseignée pendant une année par un seul prof dans une classe. Les champs *fIdMatière*, *fIdClasse* et *dtAnnée* forment donc la clé primaire de la table *tblEnseigner*. Toutefois, un prof peut enseigner la même matière pendant plusieurs années dans une même classe ou la même matière pendant une année dans plusieurs classes.
- Les champs *dtAnnée* des tables *tblFréquenter* et *tblEnseigner* font référence à des années scolaires. On y retrouve des valeurs telles que '97/98' ou '95/96'. La BD ne contient pas uniquement la situation de l'année scolaire actuelle, mais également celle des années précédentes.

Formulez en SQL les requêtes suivantes:

1. Affichez pour l'année scolaire '97/98' , le nom de chaque classe ainsi que le nombre d'élèves.
2. Affichez par année scolaire et par niveau le nombre d'élèves. Triez la liste par ordre ascendant sur l'année scolaire et par ordre ascendant sur le niveau.

3. Affichez le nom et le prénom de tous les profs ayant enseigné une matière dans une classe de 13^{ème} pendant les 5 dernières années scolaires (à partir de l'année scolaire '97/98'). Triez la liste par ordre alphabétique sur le nom du prof.
4. Dressez une liste avec le nom, le prénom, l'adresse, le code postal, et la localité pour tous les élèves qui ont fréquenté la classe '08TH1' pendant l'année scolaire '96/97'. La liste doit être triée par ordre alphabétique sur le nom des élèves. Utilisez au maximum possible le mécanisme des requêtes imbriquées.
5. Créez une liste, qui montre pour l'année scolaire '97/98', pour chaque classe, les matières enseignées avec les noms et prénoms des profs correspondants. Triez la liste par ordre alphabétique sur les noms des classes et à l'intérieur d'une classe par ordre alphabétique sur les matières. Utilisez uniquement des jointures en définissant des alias pour toutes les tables impliquées.
6. Créez une liste des profs (nom & prénom) qui est triée par ordre descendant sur le nombre de cours enseignés pendant les 3 dernières années scolaires (à partir de l'année scolaire '97/98'). La notion de cours est définie par le fait d'enseigner une matière dans une classe.
7. Affichez le nom et le prénom des profs qui enseignent au moins une matière dans une classe pendant l'année scolaire '97/98'.

Formulez la même requête en utilisant le mécanisme de la requête imbriquée corrélée.

8. Affichez le nom, le prénom, l'adresse, le code postal et la localité de tous les élèves ayant fréquenté pendant l'année scolaire 96/97 une classe du cycle inférieur. Utilisez au maximum les requêtes imbriquées.
9. Affichez le nom, le prénom et la dénomination de la classe actuelle des élèves qui sont actuellement (Année '97/98') des redoublants. Attention: Un élève est actuellement un redoublant s'il a fréquenté l'année scolaire passée une classe de même niveau, mais pas nécessairement la même classe.
10. Sachant qu'une classe ne devrait avoir un effectif supérieur à 21 élèves, le directeur vous demande d'établir une liste avec les noms des classes du cycle inférieur, qui pourraient encore accepter des nouveaux élèves pendant l'année scolaire '97/98'. Utilisez uniquement des requêtes imbriquées.

Formulez la même requête en utilisant uniquement des jointures

Formulez la même requête en utilisant le mécanisme de la requête corrélée.

11. Affichez le nom et le prénom, ainsi que le nom, le niveau et le cycle de leur classe actuelle (année = '97/98') de tous les élèves qui n'ont jamais redoublé une classe dans notre lycée.
12. Affichez parmi tous les profs, qui ont déjà enseigné la même matière que le prof numéro 10001, ceux n'ayant pas encore enseigné la même matière au même niveau que le prof numéro 10001 pendant les années scolaires '96/97' et '97/98'.

Transparents

La commande SELECT

SELECT [DISTINCT] */<colonnes>
 FROM <table/view> [alias]
 [,<table/view>[alias]...]
 [**WHERE** <condition>]
 [**GROUP BY** <colonne>[,<colonne>...]
 [**HAVING** <condition>]
 [**UNION** <SELECT>]
 [**ORDER BY** <colonne>/<integer> [ASC/DESC]
 [,<colonne>/<integer> [ASC/DESC]...]

exemples:

SELECT DISTINCT *
FROM tblEmploye
WHERE dtPays="France"
ORDER BY dtDateEmbauche,dtNom;

SELECT fiNoClient, count(*)
FROM tblCommande
GROUP BY fiNoClient;

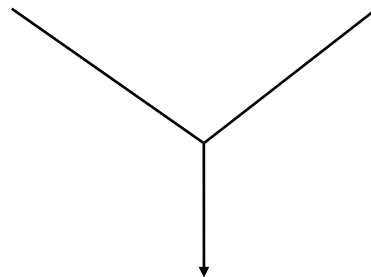
La jointure

tblProduit
<u>idProd</u>
dtLibProd
fiCat

tblCategorie
<u>idCat</u>
dtLibCat

<u>idProd</u>	dtLibProd	fiCat
55	<i>Pâté chinois</i>	6
56	<i>Gnocchi</i>	5
57	<i>Ravioli Angelo</i>	5

<u>idCat</u>	dtNomCat
5	<i>Pâtes et céréales</i>
6	<i>Viandes</i>



idProd	dtLibProd	fiCat	idCat	dtNomCat
55	<i>Pâté chinois</i>	6	5	<i>Pâtes et céréales</i>
55	<i>Pâté chinois</i>	<u>6</u>	<u>6</u>	<i>Viandes</i>
56	<i>Gnocchi</i>	<u>5</u>	<u>5</u>	<i>Pâtes et céréales</i>
56	<i>Gnocchi</i>	5	6	<i>Viandes</i>
57	<i>Ravioli Angelo</i>	<u>5</u>	<u>5</u>	<i>Pâtes et céréales</i>
57	<i>Ravioli Angelo</i>	5	6	<i>Viandes</i>

L'auto-jointure

Exemple: Les employés d'une entreprise et leur supérieur

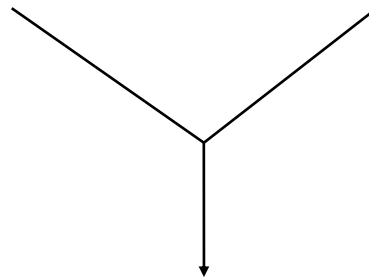
tblPersonne
<u>idPers</u>
dtNom
fiSup

tblPersonnel

<u>idPers</u>	dtNom	fiSup
55	Muller	6
56	Meier	5
57	Weber	5

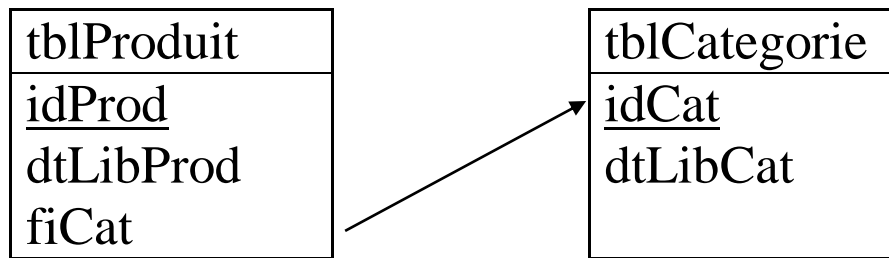
tblPersonnel

<u>idPers</u>	dtNom
5	Grandchef
6	Superchef



idPers	dtNom			dtNom
55	Muller	6	5	Grandchef
55	Muller	<u>6</u>	<u>6</u>	Superchef
56	Meier	<u>5</u>	<u>5</u>	Grandchef
56	Meier	5	6	Superchef
57	Weber	<u>5</u>	<u>5</u>	Grandchef
57	Weber	5	6	Superchef

La jointure externe



<u>idProd</u>	dtLibProd	fiCat
55	<i>Pâté chinois</i>	6
56	<i>Gnocchi</i>	
57	<i>Ravioli Angelo</i>	5

<u>idCat</u>	dtNomCat
5	<i>Pâtes et céréales</i>
6	<i>Viandes</i>

idProd	dtLibProd	fiCat	idCat	dtNomCat
55	<i>Pâté chinois</i>	6	5	<i>Pâtes et céréales</i>
55	<i>Pâté chinois</i>	<u>6</u>	<u>6</u>	<i>Viandes</i>
56	<i>Gnocchi</i>		<u>5</u>	<i>Pâtes et céréales</i>
56	<i>Gnocchi</i>		6	<i>Viandes</i>
57	<i>Ravioli Angelo</i>	<u>5</u>	<u>5</u>	<i>Pâtes et céréales</i>
57	<i>Ravioli Angelo</i>	5	6	<i>Viandes</i>

idProd	dtLibProd	fiCat
55	<i>Pâté chinois</i>	6
56	<i>Gnocchi</i>	
57	<i>Ravioli Angelo</i>	5

idCat	dtNomCat
5	<i>Pâtes et céréales</i>
6	<i>Viandes</i>

Jointure interne:

```
SELECT dtLibProd, dtNomCat
FROM tblProduit, tblCategorie
WHERE fiCat=idCat;
```

ou bien

```
SELECT dtLibProd, dtNomCat
FROM tblProduit INNER JOIN tblCategorie
ON fiCat=idCat;
```

idLibProd	dtNomCat
Pâté chinois	Viandes
Ravioli Angelo	Pâtes et céréales

Jointure externe:

```
SELECT dtLibProd, dtNomCat
FROM tblProduit JOIN tblCategorie
ON fiCat=idCat;
```

idLibProd	dtNomCat
Pâté chinois	Viandes
<i>Gnocchi</i>	
Ravioli Angelo	Pâtes et céréales

Tableau des commandes SQL

SQL interactif			SQL intégré	SQL dynamique
LDD	LMD	LCD	DECLARE CURSOR FETCH	PREPARE DESCRIBE EXECUTE
CREATE DROP ALTER	SELECT INSERT DELETE UPDATE	GRANT REVOKE CONNECT COMMIT ROLLBACK SET		

LDD: Langage de définition des données (Data Definition Language, DDL)
 LMD: Langage de manipulation des données (Data Manipulation Language, DML)
 LCD: Langage de contrôle des données (Data Control Language, DCL)