

# Requêtes SQL

## Exercices corrigés

Frédéric Gava (MCF)

*[gava@univ-paris12.fr](mailto:gava@univ-paris12.fr)*

LACL, bâtiment P2 du CMC, bureau 223

Université de Paris XII Val-de-Marne

61 avenue du Général de Gaulle

94010 Créteil cedex



# Exercise 1

# Les livraisons, la BD

## ■ Soit la base relationnelle de données PUF de schéma :

- ▶ U(NumU, NomU, VilleU)
- ▶ P(NumP, NomP, Couleur, Poids)
- ▶ F(NumF, NomF, Statut, VilleF)
- ▶ PUF(NumP, NumU, NumF, Quantité)

## ■ décrivant le fait que (avec des DF évidentes) :

- ▶ U : une usine est décrite par son numéro NumU, son nom NomU et la ville VilleU où elle est située
- ▶ P : un produit est décrit par son numéro NumP, son nom NomP, sa couleur et son poids
- ▶ F : un fournisseur est décrit par son numéro NumP, son nom NomF, son statut (sous-traitant, client...) et la ville VilleF où il est domicilié
- ▶ PUF : le produit de numéro NumP a été livré à l'usine de numéro NumU par le fournisseur de numéro NumF dans une quantité donnée

# Exprimez en SQL (1)

- 1) Ajouter un nouveau fournisseur avec les attributs de votre choix
- 2) Supprimer tous les produits de couleur noire et de numéros compris entre 100 et 1999
- 3) Changer la ville du fournisseur 3 par Toulouse
- 4) Donnez le numéro, le nom, la ville de toutes les usines
- 5) Donnez le numéro, le nom, la ville de toutes les usines de Paris
- 6) Donnez les numéros des fournisseurs qui approvisionnent l'usine de numéro 2 en produit de numéro 100
- 7) Donnez les noms et les couleurs des produits livrés par le fournisseur de numéro 2
- 8) Donnez les numéros des fournisseurs qui approvisionnent l'usine de numéro 2 en un produit rouge
- 9) Donnez les noms des fournisseurs qui approvisionnent une usine de Paris ou de Créteil en produit rouge
- 10) Donnez les numéros des produits livrés à une usine par une fournisseur de la même ville

# Exprimez en SQL (2)

- 11) Donnez les numéros des produits livrés à une usine de Paris par un fournisseur de Paris.
- 12) Donnez les numéros des usines qui ont au moins un fournisseur qui n'est pas de la même ville
- 13) Donnez les numéros des fournisseurs qui approvisionnent à la fois des usines de numéros 2 et 3
- 14) Donnez les numéros des usines qui utilisent au moins un produit disponible chez le fournisseur de numéro 3 (c'est-à-dire un produit que le fournisseur livre mais pas nécessairement à cette usine)
- 15) Donnez le numéro du produit le plus léger (les numéros si plusieurs produits ont ce même poids)
- 16) Donnez le numéro des usines qui ne reçoivent aucun produit rouge d'un fournisseur parisien
- 17) Donnez les numéros des fournisseurs qui fournissent au moins un produit fourni par au moins un fournisseur qui fournit au moins un produit rouge

# Exprimez en SQL (3)

- 18) Donnez tous les triplets (VilleF, NumP, VilleU) tels qu'un fournisseur de la première ville VilleF approvisionne une usine de la deuxième ville VilleU avec un produit NumP
- 19) Même question que précédemment mais sans les triplets où les deux villes sont identiques
- 20) Donnez les numéros des produits qui sont livrés à toutes les usines de Paris
- 21) Donnez les numéros des fournisseurs qui approvisionnent toutes les usines avec un même produit
- 22) Donnez les numéros des usines qui achètent au fournisseur de numéro 3 tous les produits qu'il fournit
- 23) Donnez les numéros des usines qui s'approvisionnent uniquement chez le fournisseur de numéro 3

# Exercise 2

# La société Gavasoft

## ■ Soit les relations suivantes de la société Gavasoft

► Emp(NumE, NomE, Fonction, NumS, Embauche, Salaire, Comm, **NumD**)

► Dept(NumD, NomD, Lieu)

► Exemple

NumD	NomD	Lieu
1	Droit	Créteil
2	Commerce	Boston

NomE	Fonction	NumS	Embauche	Salaire	Comm	NumD
Gava	Président	NULL	10/10/1979	10000	NULL	NULL
Guimezanes	Doyen	1	01/10/2006	5000	NULL	1
Toto	Stagiaire	1	01/10/2006	0	NULL	1
Al-Capone	Commercial	2	01/10/2006	5000	100	2



# Exprimez en SQL (4)

- 1) Donnez la liste des employés ayant une commission (non NULL) classé par commission décroissante
- 2) Donnez les noms des personnes embauchées depuis le 01-09-2006
- 3) Donnez la liste des employés travaillant à Créteil
- 4) Donnez la liste des subordonnés de "Guimezanes"
- 5) Donnez la moyenne des salaires
- 6) Donnez le nombre de commissions non NULL
- 7) Donnez la liste des employés gagnant plus que la moyenne des salaires de l'entreprise

# Exercise 3

# Une médiathèque (1)

- On considère le schéma relationnel suivant qui modélise une application sur la gestion de livres et de disques dans une médiathèque
- Les disques :
  - ▶ Disque(CodeOuv, Titre, Style, Pays, Année, Producteur)
  - ▶ Cette relation regroupe un certain nombre d'informations sur un disque : le code d'ouvrage CodeOuv qui est la clé de la relation, le titre, le style (Jazz, Rock etc.), le pays, l'année de sortie et le producteur (par exemple Barclay) ; ces informations sont générales et pour un enregistrement de la relation Disque, on aura  $n > 1$  enregistrements dans la relation E\_Disque correspondant aux exemplaires de ce disque possédés par la médiathèque
- Les exemplaires :
  - ▶ E\_Disque(CodeOuv, NumEx, DateAchat, Etat)
  - ▶ Cette relation contient un enregistrement pour chaque exemplaire de disque possédé par la médiathèque ; chaque exemplaire est identifié par son code (CodOuv) et un numéro d'exemplaire (NumEx) ; on trouve également la date d'achat et l'état du disque (intact, abîmé etc.)

# Une médiathèque (2)

## ■ Les livres :

- ▶ Livre(CodeOuv, Titre, Editeur, Collection)
- ▶ Cette relation regroupe un certain nombre d'informations sur un livre : le code de l'ouvrage (CodeOuv) qui est la clé de la relation, le titre, le genre (par exemple polar ou SF), l'éditeur (par exemple Glénat) et la collection (par exemple « livre de poche ») ; ces informations sont générales et pour un enregistrement de la relation Livre, on aura  $n > 1$  enregistrement dans la relation E\_Livre correspondant aux exemplaires de ce livre possédés par la médiathèque

## ■ Les exemplaires :

- ▶ E\_Livre(CodeOuv, NumEx, DateAchat, Etat)
- ▶ Cette relation contient un enregistrement pour chaque exemplaire de livre possédé par la médiathèque ; chaque exemplaire est identifié par son code (CodeOuv) et un numéro d'exemplaire (NumEx) ; on trouve également la date d'achat et l'état du livre (intact, abîmé etc.)

# Une médiathèque (3)

## ■ Les auteurs :

- ▶ Auteurs(CodeOuv, Identité)
- ▶ Chaque enregistrement de cette relation correspond à l'un des auteurs d'un ouvrage particulier (livre ou disque) ; l'attribut Identité peut avoir pour valeur un nom de personne (par exemple Isaac Asimov) ou un nom de groupe (par exemple Noir Désir)

## ■ Les abonnés :

- ▶ Abonne(NumAbo, Nom, Prénom, Rue, Ville, CodeP, Téléphone)
- ▶ Cette relation regroupe les informations sur les abonnées de la médiathèque : NumAbo qui identifie tout abonné de manière individuelle, le nom, le prénom de l'abonné, son adresse et son numéro de téléphone

# Une médiathèque (4)

## ■ Les prêts :

- ▶ Prêt(CodeOuv, NumEx, DisqueOuLivre, NumAbo, DatePret)
- ▶ Cette relation contient un enregistrement par prêt effectué ; pour chaque prêt, on trouve l'identifiant du livre ou du disque (code ouvrage et numéro d'exemplaire), le numéro de l'abonné effectuant le prêt, un attribut explicitant si le prêt est celui d'un livre ou d'un disque (« D » pour un disque et « L » pour un livre) et enfin la date du prêt ; cette relation ne contient des informations que pour les prêts en cours c'est-à-dire pour les emprunts non encore rendus

## ■ Le Personnel :

- ▶ Personnel(NumEmp, Nom, Prénom, Adresse, Fonction, Salaire)
- ▶ Cette relation contient un enregistrement par employé de la médiathèque ; chaque employé est identifié par un numéro et pour chaque employé, la relation donne son nom, son prénom, son adresse, sa fonction et son salaire annuel

# Une médiathèque (5)

- Traduisez en SQL les question suivantes :
  - 1) Quel est le contenu de la relation Livre ?
  - 2) Quels sont les titres des romans édités par Gava-Editor ?
  - 3) Quelle est la liste des titres que l'on retrouve à la fois comme titre de disque et titre de livre ?
  - 4) Quelle est l'identité des auteurs qui ont fait des disques et écrit des livres ?
  - 5) Quels sont les différents style de disques proposés ?
  - 6) Quel est le salaire annuel des membres du personnel gagnant plus de 20000 euros en ordonnant le résultat par salaire descendant et nom croissant ?

# Une médiathèque (6)

## ■ Suite :

- 7) Donnez le nombre de prêts en cours pour chaque famille en considérant qu'une famille regroupe des personnes de même nom et possédant le même numéro de téléphone ?
- 8) Quel est le code du disque dont la médiathèque possède le plus grand nombre d'exemplaire ?
- 9) Quels sont les éditeurs pour lesquels l'attribut Collection n'a pas été renseigné ?
- 10) Quels sont les abonnés dont le nom contient la chaîne « ALDO » et habitant en Isère ?
- 11) Quel est le nombre de prêts en cours ?
- 12) Quels sont les salaires minimum, maximum et moyen des employés exerçant une fonction de bibliothécaire ?
- 13) Quel est le nombre de genres de livres différents ?
- 14) Quel est le nombre de disque acheté en 1998 ?



# Une médiathèque (7)

## ■ Suite :

- 15) Quel est le salaire annuel des membres du personnel gagnant plus de 20000 euros ?
- 16) Quel est le nom, prénom et l'adresse des abonnés ayant emprunté un disque le '12/01/2006' ?
- 17) Quels sont les titres des livres et des disques actuellement empruntés par Frédéric Gava ?
- 18) Quels sont les titres des ouvrages livres policiers ou disques de Jazz empruntés par Frédéric Gava ?
- 19) Quel est l'identité des auteurs qui n'ont écrit que des romans policiers (genre=policier) ?
- 20) Quel sont les codes des ouvrages des livres pour lesquels il y a au moins un exemplaire emprunté et au moins un exemplaire disponible ?

# Correction exercice 1

# Les livraisons (1)

- 1) **INSERT INTO F VALUES** (45, 'Alfred', 'Sous-traitant', 'Chalon')
- 2) **DELETE P WHERE**  $N_p \geq 100$  **AND**  $N_p \leq 199$  **AND** Couleur='Noire'
- 3) **UPDATE F SET** Ville='Nice' **WHERE**  $N_f = 1$
- 4) **SELECT \* FROM** U
- 5) **SELECT \* FROM** U **WHERE** Ville="Créteil"
- 6) **SELECT**  $N_f$  **FROM** PUF **WHERE**  $N_u = 1$  **AND**  $N_p = 1$
- 7) **SELECT DISTINCT** NomP, Couleur **FROM** P, PUF **WHERE** PUF. $N_p = P.N_p$  **AND**  $N_f = 1$ 
  - ▶ Ou bien **SELECT** NomP, Couleur **FROM** P **WHERE**  $N_p$  **IN** (**SELECT**  $N_p$  **FROM** PUF **WHERE**  $N_f = 1$ )

# Les livraisons (2)

- 8) **SELECT DISTINCT Nf FROM PUF, P WHERE**  
**Couleur="Rouge" AND PUF.Np=P.Np AND Nu=1**
- Ou bien **SELECT DISTINCT Nf FROM PUF WHERE Np IN**  
**(SELECT Np FROM P WHERE Couleur="Rouge") AND**  
**Nu=1**
- 9) **SELECT NomF FROM PUF, P, F, U WHERE**  
**Couleur='Rouge' AND PUF.Np=P.Np AND PUF.Nf=F.Nf AND**  
**PUF.Nu=U.Nu AND (U.Ville IN ('Paris','Créteil'))**
- Ou bien **SELECT NomF FROM F WHERE Nf IN (SELECT**  
**Nf FROM PUF WHERE Np IN (SELECT Np FROM P**  
**WHERE Couleur='Rouge') AND Nu IN (SELECT Nu FROM**  
**U WHERE Ville IN ('Paris', 'Créteil'))**
- 10) **SELECT DISTINCT Np FROM PUF, F, U WHERE PUF.Nf=F.Nf**  
**AND PUF.Nu=U.Nu AND U.Ville=F.Ville**

# Les livraisons (3)

- 11) **SELECT DISTINCT Np FROM PUF, F, U WHERE PUF.Nf=F.Nf AND PUF.Nu=U.Nu AND U.Ville=F.Ville AND U.Ville='Paris'**
- Ou bien **SELECT DISTINCT Np FROM PUF WHERE Nf IN (SELECT Nf FROM F WHERE Ville='Paris') AND Nu IN (SELECT Nu FROM U WHERE Ville='Paris')**
- 12) **SELECT DISTINCT PUF.Nu FROM PUF, F, U WHERE PUF.Nf=F.Nf AND PUF.Nu=U.Nu AND U.Ville<>F.ville**
- Ou bien **SELECT DISTINCT Nu FROM PUF WHERE Nf=ANY(SELECT Nf FROM F, U WHERE PUF.Nf=F.Nf AND PUF.Nu=U.Nu AND F.Ville<>U.Ville)**
- 13) **SELECT DISTINCT First.Nf FROM PUF First, PUF Second WHERE First.Nf=Second.Nf AND First.Nu=1 AND Second.Nu=2**
- Ou bien **SELECT DISTINCT Nf FROM PUF WHERE Nf IN (SELECT Nf FROM PUF WHERE Nu=1) AND Nu=2**
- 14) **SELECT DISTINCT Nu FROM PUF WHERE Np IN (SELECT Np FROM PUF WHERE Nf=3)**

# Les livraisons (4)

- 15) **SELECT** Np **FROM** P **WHERE** Poids **IN** (**SELECT** MIN(Poids) **FROM** P)
- Ou bien **SELECT** Np **FROM** P p1 **WHERE** NOT EXISTS (**SELECT** \* **FROM** P **WHERE** P1.Poids>Poids)
- 16) **SELECT** Nu **FROM** U **WHERE** Nu NOT **IN** (**SELECT** Nu **FROM** PUF, F, P **WHERE** PUF.Np=P.Np **AND** PUF.Nf=F.Nf **AND** Couleur='Rouge' **AND** Ville='Paris')
- 17) **SELECT** DISTINCT PUF.Nf **FROM** PUF, PUF PUF1, PUF PUF2, P **WHERE** Couleur='Rouge' **AND** P.Np=PUF2.Np **AND** PUF2.Nf=PUF1.Nf **AND** PUF1.Np=PUF.Np
- Ou bien **SELECT** DISTINCT Nf **FROM** PUF **WHERE** Np **IN** (**SELECT** Np **FROM** PUF **WHERE** Nf **IN** (**SELECT** Nf **FROM** PUF **WHERE** Np **IN** (**SELECT** Np **FROM** P **WHERE** Couleur='Rouge'))))
- 18) **SELECT** DISTINCT F.Ville, Np, U.Ville **FROM** PUF, U, F **WHERE** PUF.Nf=F.Nf **AND** PUF.Nu=U.Nu

# Les livraisons (5)

- 19) **SELECT DISTINCT** F.Ville, NP, U.Ville **FROM** PUF, U, F  
**WHERE** F.Ville<>U.Ville **AND** PUF.Nf=F.Nf **AND**  
PUF.Nu=U.Nu
- 20) **SELECT** Np **FROM** PUF **WHERE NOT EXISTS**(**SELECT** Nu  
**FROM** U **WHERE NOT EXISTS** (**SELECT** \* **FROM** PUF **WHERE**  
**NOT** (Ville='Paris') **OR** (P.Np=PUF.Np **AND** U.Nu=PUF.Nu))
- 21) **SELECT** NF **FROM** PUF **WHERE NOT EXISTS** (**SELECT** Nu  
**FROM** U **WHERE NOT EXISTS** (**SELECT** \* **FROM** PUF PUF1  
**WHERE** F.Nf=PUF1.NF **AND** U.Nu=PUF1.Nu **AND**  
PUF.Np=PUF1.Np))
- **SELECT** Nf **FROM** F **WHERE EXISTS** (**SELECT** Np **FROM** P  
**WHERE NOT EXISTS** (**SELECT** Nu **FROM** U **WHERE NOT**  
**EXISTS** (**SELECT** \* **FROM** PUF **WHERE** F.Nf=PUF.Nf **AND**  
U.Nu=PUF.Nu **AND** P.Np=PUF.Np)))
- 23) **SELECT** Nu **FROM** U **WHERE** Nu **NOT IN** (**SELECT** Nu **FROM**  
PUF **WHERE** Nf<>3)

# Correction exercice 2



# La société Gavasoft

- 1) **SELECT** Nom, Comm "Commission" **FROM** Emp **WHERE** Comm **IS NOT NULL AND** Comm!=0 **ORDER BY** Comm **DESC**
- 2) **SELECT** Nom, Embauche, N\_Dept **FROM** Emp **WHERE** Embauche > '01/10/2006'
- 3) **SELECT** Nom, Embauche, N\_Dept **FROM** Emp, Dept **WHERE** Emp.N\_Dept=Dept.N\_Dept **AND** Lieu="Créteil"
- 4) **SELECT** a.Nom "Nom", Lieu **FROM** Emp a, Emp b **WHERE** a.NumSup=b.NumSup **AND** b.NumSup="Gava"
- 5) **SELECT AVG**(Salaire) « Moyenne des salaires » **FROM** Emp
- 6) **SELECT COUNT**(Comm) « Nb. Commissions non-Null » **FROM** Emp **WHERE** Comm **IS NOT NULL**
- 7) **SELECT** Nom, Fonction, Salaire **FROM** Emp **WHERE** Salaire>(SELECT **AVG**(Salaire) **FROM** Emp)

# Correction exercice 3

# Une médiathèque (6)

1) **SELECT \* FROM** Livre

- Dans ce premier exemple, notons l'utilisation du symbole \* pour spécifier que l'on souhaite conserver dans le résultat tous les attributs de la relation Livre

2) **SELECT** Titre **FROM** Livre **WHERE** Editeur="Droit-Edition"  
**AND** Genre="Polar"

- Dans cette requête, la condition porte sur les attributs Editeur et Genre et le résultat retourné par la requête est la liste des titres. Il n'y a en effet pas nécessairement de liens entre les attributs retournés et ceux sur lesquels portent la condition

3) **SELECT** D.Titre **FROM** Disque D, Livre L **WHERE**  
D.Titre=L.Titre

4) **SELECT** A1.Identité **FROM** Disque D, Livre L, Auteur A1,  
Auteur A2 **WHERE** D.CodeOuv=A1.CodeOuv **AND**  
L.CodeOuv=A2.CodeOuv **AND** A1.Identité=A2.Identité

# Une médiathèque (7)

## 5) **SELECT DISTINCT** Style **FROM** Disque

- La clause **DISTINCT** permet de supprimer les doublons au niveau du résultat ; par défaut, SQL conserve les doublons pour optimiser le temps d'exécution et pour répondre à une éventuelle attente de l'utilisateur

## 6) **SELECT** Nom, Prénom, Salaire\*12 **AS** Salaire\_Annuel **FROM** Personnel **WHERE** Salaire\_Annuel>20000 **ORDER BY** Salaire **DESC**, Nom **ASC**

- La clause **ORDER BY** permet le tri du résultat avant affichage

## 7) **SELECT** Nom, Téléphone, **COUNT**(\*) **FROM** Abonne A, Prêt P **WHERE** A.NumAbo=P.NimAbo **GROUP BY** Nom, Téléphone

## 8) **SELECT** CodeOuv **FROM** E\_Disque **GROUP BY** CodeOuv **HAVING** **COUNT**(\*)=(**SELECT** **MAX**(**COUNT**(\*)) **FROM** E\_Disque **GROUP BY** CodeOuv)

## 9) **SELECT** Editeur **FROM** Livre **WHERE** Collection **IS NULL**

- La présence de valeurs nulles dans une relation peut être autorisée mais rarement souhaitable car leur interprétation est ambiguë...

# Une médiathèque (8)

10) **SELECT \* FROM** Abonne **WHERE** Nom='%ALDO%' **AND** CodeP='38--'

- Certains opérateurs SQL permettent une recherche approximative pour les attributs de type chaîne : le caractère de remplacement % indique la possibilité d'avoir 0 ou plusieurs caractères quelconques

11) **SELECT COUNT(\*) FROM** Prêt

12) **SELECT MIN(Salaire), MAX(Salaire), AVG(Salaire) FROM** Personnel **WHERE** Fonction=« bibliothécaire »

13) **SELECT COUNT(DISTINCT Genre) FROM** Livre

14) **SELECT COUNT(\*) FROM** E\_Disque **WHERE** DateAchat **BETWEEN** '01-Jan-2006' **AND** '10-Dec-2007'

15) **SELECT** Nom, Prénom, Salaire\*12 **AS** Salaire\_Annuel **FROM** Personnel **WHERE** Salaire\_Annuel>20000

- Il est possible d'utiliser les opérateur arithmétique à la fois au niveau de la clause SELECT et de la clause WHERE

# Une médiathèque (9)

- 16) **SELECT** Nom, Prénom, Rue, Ville, CodeP **FROM** Abonne A, Prêt P, Disque D **WHERE** A.NumAbo=P.NumAbo **AND** P.CodeOuv=D.CodeOuv **AND** DatePret='12-Jan-2006'
- 17) (**SELECT** Titre **FROM** Abonne A, Prêt P, Disque D **WHERE** A.NumAbo=P.NumAbo **AND** P.CodeOuv=D.CodeOuv **AND** NOM="Gava" **AND** Prénom="Frédéric") **UNION** (**SELECT** Titre **FROM** Abonne A, Prêt P, Livre L **WHERE** A.NumAbo=P.NumAbo **AND** P.CodeOuv=L.CodeOuv **AND** NOM="Gava" **AND** Prénom="Frédéric")
- 18) **SELECT** CodeOuv **FROM** Prêt P, Abonne A **WHERE** P.NumAbo=A.NumAbo **AND** Prénom="Frédéric" **AND** Nom="Gava" **AND** CodeOuv **IN** (**SELECT** CodeOuv **FROM** Livre **WHERE** Genre="Policier") **OR** CodeOuv **IN** (**SELECT** CodeOuv **FROM** Disque **WHERE** Style="Jazz")

# Une médiathèque (10)

- **SELECT** Identité **FROM** Auteur A, Livre L **WHERE** A.CodeOuv=L.CodeOuv **AND** Genre="Policier" **AND NOT ALL**(**SELECT** Identité **FROM** Auteur A, Livre L **WHERE** A.CodeOuv=L.CodeOuv **AND** Genre<>"Policier")
- (**SELECT** P.CodeOuv **FROM** E\_Livre E, Prêt P **WHERE** E.CodeOuv=P.CodeOuv) **INTERSECT** (**SELECT** CodeOuv **FROM** E\_Livre E **WHERE NOT EXISTS**(**SELECT** \* **FROM** Prêt P **WHERE** E.CodeOuv=P.CodeOuv **AND** E.NumEx=P.NumEx