

## Voici 5 énoncés d'exercices avec leurs réponses.

### SUJET N°1

Soit le schéma relationnel de la société française d'archéologie :

OBJET (num-obj, description, type, datation, num-ville, num-site, num-musee)

VILLE (num-ville, nom-ville)

MUSEE (num-musee, num-ville, nom)

SITE (num-ville, num-site, description, civilisation)

PUBLICATION (num-pub, titre, date, editeur)

AUTEUR (num-aut, nom, prenom)

COOPERATION (num-aut, num-pub)

REFERENCE (num-pub, num-obj)

Cette base gère des sites archéologiques et des publications sur ces objets. La relation OBJET décrit les objets proprement dits avec leur type (par exemple vase), leur datation qui est une année, le site où ils ont été découverts et le musée où ils se trouvent actuellement. La relation VILLE comprend le nom des villes. La relation SITE indique la ville à laquelle se rattache le site et un numéro qui est un numéro d'ordre du site dans la ville. La civilisation du site est une grande catégorie comme « romaine » ou « crétoise ».

### Questions :

Exprimez en SQL les requêtes suivantes :

- 1) Quelles sont les frises du troisième siècle
- 2) Même question avec en plus le nom du musée où elles sont exposées
- 3) Noms et prénoms des auteurs de publication référençant des objets de la civilisation romaine
- 4) Quelles statues sont exposées dans la ville où elles ont été découvertes ? donner le numéro et la description
- 5) Noms et prénoms des auteurs des ouvrages qui référencent tous les objets trouvés dans le site 2 de Thèbes
- 6) Donner le nombre de statues trouvées dans chaque site de la civilisation phénicienne

### REPONSES

- 1) `select num-obj, description from OBJET where type = « frise » and datation = 3`
- 2) `select num-obj, description, nom from OBJET, MUSEE where type = « frise » and datation = 3 and OBJET.num-musee = MUSEE.num-musee`
- 3) `select nom, prenom from AUTEUR A, COOPERATION C, REFERENCE R, OBJET O, SITE S where civilisation = « romaine » and A.num-aut = c.num-aut and C.num-pub = R.num-pub and R.num-obj = O.num-obj and O.num-ville = S.num-ville and O.num-site = S.num-site`
- 4) `select num-obj, description from OBJET O where type = « statue » and num-ville in (select num-ville from MUSEE M where O.num-musee = M.num-musee)`  
ou `select num-obj, description from OBJET, MUSEE where type = « statue » and OBJET.num-ville = MUSEE.num-ville and OBJET.num-musee = MUSEE.num-musee`
- 5) `Select nom, prenom from AUTEUR A where NOT EXISTS (select num-objet from OBJET O, SITE S where O.num-ville = S.num-ville and O.num-site = S.num-site and S.num-site = 2 and S.numville = « Thèbes ») and NOT EXISTS (select * from REFERENCE R where R.num-obj = O.num-objet and num-pub in (select num-pub from COOPERATION C where c.num-aut = A.num-aut)`
- 6) `Select num-ville, num-site, count(*) from OBJET O, SITE S where type = « statue » and O.num-ville = S.num-ville and O.num-site = S.num-site civilisation = « phénicienne » group by num-ville, num-site`

## SUJET N°2

On considère la structure relationnelle suivante :

ARTICLE (ref\_art, designation, Prix\_unit, famille)  
ARTICLE\_SUBSTITUTION (ref\_art, ref\_art\_replacement)  
CUMUL\_VENTES (ref\_art, code\_magasin, mois\_annee, cumulvente)  
STOCK\_MAGASIN (ref\_art, code\_magasin, qtestk)  
HISTO\_PRIX\_ARTICLE (ref\_art, date\_prix, prix)  
MAGASIN (code\_magasin, adresse\_magasin)

### Questions :

- 1) Déclarer en SQL les structures des tables ARTICLE, ARTICLE\_SUBSTITUTION avec leurs attributs et leurs contraintes d'intégrité
- 2) Exprimer les requêtes suivantes avec le langage SQL
  - a) Donner la liste des articles (leur référence) qui n'ont pas d'article de remplacement
  - b) Donner la liste des produits dont le prix a diminué au moins une fois
  - c) Donner les références et désignations des articles qui ont un cumul mensuel de leurs ventes supérieur à 50 000
  - d) Donner les références et désignations des articles qui ont un cumul mensuel de leurs ventes supérieur à 50 000 et une quantité en stock supérieure à 256 dans le magasin 15
  - e) Donner les articles qui sont disponibles dans tous les magasins
- 3) Définir une vue permettant de comparer facilement le prix d'un article et le prix de ses articles de substitution
- 4) A partir de la vue définie en 3), exprimer une requête permettant d'obtenir les articles qui ont des articles de substitution moins chers.

## REPONSES

1) Déclaration SQL des tables ARTICLE et ARTICLE\_SUBSTITUTION

**CREATE TABLE ARTICLE**

(reCart NUMERIC(5) **PRIMARY KEY**  
designation CHAR(15)  
prix\_uoitNUMERIC (6), **NOT NULL**  
famille CHAR)

**CREATE TABLE ARTICLE SUBSTITUTION**

(reCartNUMERIC(5) REFERENCES ARTICLE  
reCartJemplacement NUMERIC (5) REFERENCE ARTICLE  
**PRIMARY KEY** (reCart, reCartJemplacement))

2) Requêtes SQL

- a) Select reCart from ARTICLE wherereCart not in (Select reCart from ARTICLE\_SUBSTITUTION) ;
- b) Select reCart from HISTO]RIX\_ARTICLE H1, HISTO]RIX\_ARTICLE H2 where H1.reCart = H2.reCart and H1. date-prix < H2.date--.prix and H1prix > H2.prix

- c) Select ref-art, designation from ARTICLE, CUMUL\_VENTES where ARTICLE.reC art = CUMUL\_VENTES.ref mi and cumulvente > 50 000  
Ou Select ref-art, designation from ARTICLE where reCart in select reCart from CUMUL\_VENTES where cumul vente > 50 000
  - d) Select reC art, designation from ARTICLE, CUMUL\_VENTES, STOCK\_MAGASIN where ARTICLE.reCart = CUMUL\_VENTES.reCart and ARTICLE.reCart = STOCK\_MAGASIN.reCart and cumul vente > 50 000 and qtestk = 256 and code\_magasin = 15
  - e) Select reC mi from STOCK\_MAGASIN where qtestk <> 0 group by reC art having couot (\*) = select count (\*) from MAGASIN
- 3) CREATE VIEW COMPARE (reCart, prix\_art, rem\_art, rem--'prix) AS select A reCart, Aprix\_uoit, S.reCart, S.prix\_uoit from ARTICLE A, ARTICLE\_SUBSTITUTION AS, ARTICLE S where AreCart = AS.reCart and AS.reCartJemplacement = S.reCart
- 4) select reCart from COMPARE where prix\_art > remp -prix

## SUJET N°3

On considère un système centralisé de réservation de chambres d'hôtels dans une région de vacances qui englobe plusieurs stations ayant chacune plusieurs hôtels.

Tout demandeur peut téléphoner au système, afin de réserver des chambres. L'opérateur chargé de la réservation lui demande plusieurs renseignements : nom, adresse, numéro de téléphone et diverses indications sur sa demande : période de réservation, le nombre de chambres, la catégorie d'hôtel, la station demandée. Il affecte à chaque demande un numéro d'ordre. Pour deux périodes distinctes, on considère qu'il y a deux demandes distinctes.

L'opérateur est chargé de vérifier si la demande peut être satisfaite; s'il n'y a pas de possibilité, la demande peut être mise en attente.

S'il est possible de satisfaire la demande, il y a confirmation par l'opérateur, avec création d'une réservation qui comporte tous les renseignements qui vont permettre sa réalisation.

Une réservation est associée à un client. Un client est une personne qui a, ou a eu, au moins une réservation établie par le système.

Chaque hôtel est décrit par son nom, son numéro, la station à laquelle il appartient, sa catégorie, le nombre de chambres disponibles, le prix unitaire de chaque chambre et les périodes de disponibilité pour chaque chambre. Chaque station est décrite par son nom et son numéro.

On considère qu'une demande est relative à un seul demandeur et la réservation à un seul client. Chaque période est repérée par une date de début et une date de fin.

On vous fournit ci-dessous une collection de relations normalisées ainsi que le dictionnaire des attributs.

**DEMANDE** (numdem, nomdeur, adrdeur, teldeur, datedebdem, datefindem, nbchamdem, cathotdem, datedem, etatdem, numstat)

**STATION** (numstat, nomstat)

**HOTEL** (numhot, nomhot, adrhot, cathot, numstat)

**CHAMBRE** (numcham, numhot, catcham, pxcham)

**DISPONIBILITE\_CHAMBRE** (numcham, numhot, datedebdispo, datefindispo)

**CATEGORIE\_CHAMBRE** (catcham, pxmin, pxmax)

**RESERVATION** (numres, datedebres, datefinres, dateres, etatres, numdem, numcli, numhot)

**RESERVATION\_CHAMBRE** (numres, numhot, numcham)

**CLIENT** (numcli, nomcli, adrcli, telcli)

### **Questions:**

1) Déclarer la structure de la table RESERVATION\_CHAMBRE

2) Exprimer en SQL les requêtes suivantes :

- i) Afficher par hôtel, la liste des numéros et noms des clients devant arriver demain.
- ii) Afficher par hôtel et par catégorie de chambre le nombre de chambres proposées par ce système de réservation
- iii) Quel est le nom du client qui a la réservation 27 ?
- iv) Quelles sont les chambres qui n'ont fait l'objet d'aucune réservation ?
- v) Donner les réservations qui concernent plus de 5 chambres

3) Définir une vue qui contient par réservation le nombre de chambres réservées.

4) Utiliser la vue définie dans la question 3) pour exprimer la requête v)

5) On souhaite modifier la structure de la base de données pour prendre en compte la nouvelle règle de gestion suivante : « Le prix des chambres varie en fonction des saisons et les saisons peuvent être différentes d'une station à une autre. Par exemple la haute saison peut commencer le 1<sup>ier</sup> mai dans une station et le 1<sup>ier</sup> juin dans une autre ». Proposer une nouvelle structure pour mémoriser dans la base de données les différents tarifs des chambres. La solution proposée doit être normalisée.

## **REPONSES**

1) Structure de la table RESERVATION\_CHAMBRE

```
CREATE TABLE RESERVATION_CHAMBRE
(numres : <type1> REFERENCES RESERVATION,
numhot : <type2>,
numcham : <type3>,
PRIMARY KEY (numres, numhot, numcham)
FOREIGN KEY (numhot, numcham) REFERENCES CHAMBRE)
```

## 2) Requêtes SQL

i) Select numcli, nomcli  
From RESERVATION, CLIENT  
Where RESERVATION.numcli = CLIENT.numcli  
And datedebres = datecourante+1 (ou datedebres = demain)  
Order by numhot

ii) Select numhot, catcham, count(\*)  
From CHAMBRE  
Group by numhot, catcham

iii) Select nomcli  
From RESERVATION, CLIENT  
Where RESERVATION.numcli = CLIENT.numcli  
And numres = 27

iv) Select numhot, numcham  
From CHAMBRE  
MINUS (ou EXCEPT)  
Select numhot, numcham  
From RESERVATION\_CHAMBRE

v) Select numres  
From RESERVATION\_CHAMBRE  
Group by numres  
Having count (\*) > 5

3) CREATE VIEW NOMBRE\_CHAMBRE\_RESERVEE (numres, nombre) AS  
Select numres, count(\*)  
From RESERVATION\_CHAMBRE  
Group By numres

4) Select \*  
From NOMBRE\_CHAMBRE\_RESERVEE  
Where nombre > 5

5) Il faut rajouter les tables suivantes :

PRIX-CHAMBRE (numhot, numcham, numsaizon, prix) Cette table fournit les prix des chambres pour chaque saison.

SAISON-STATION (numsaizon, numstat, datedebutsaizon, datefinsaizon) Cette table fournit les saisons pour chaque station.

## SUJET N° 4

On s'intéresse à l'activité du département commercial d'une importante entreprise dans le secteur de la fabrication et de la commercialisation de cartons et papiers. Cette entreprise vend sur catalogue. Son catalogue commercial comporte environ 25.000 articles qui sont regroupés en 60 familles et 90 sous-familles. Un article appartient à une seule sous-famille qui n'appartient qu'à une famille. Une famille regroupe un ensemble d'articles de même sorte (papier machine, cartons emballage...), chaque famille est caractérisée par un numéro et une désignation. Les articles d'une sous-famille ont le même format, ils se différencient sur le grammage et la couleur. Par exemple la sous-famille correspondant au papier machine de format A4 comporte plusieurs articles en fonction du grammage (le papier 80g de couleur verte, le papier 90g de couleur jaune...).

Dans ce catalogue, chaque article est décrit par un numéro d'article qui est unique (NART), une désignation commerciale qui est aussi unique (DESART), un prix unitaire de vente (PVART) et un délai de livraison (LIVART).

Les clients sont des entreprises industrielles. Un client a un numéro (NCLI) et une raison sociale (RSCLI) qui sont uniques. Il a une seule adresse de facturation (ADRFACCLI), il peut avoir jusqu'à dix adresses de livraison (ADRLIVCLI). Ces adresses de livraison correspondent aux différents établissements du client.

L'action commerciale est faite par des commerciaux qui agissent auprès des établissements des clients. Un seul commercial intervient auprès de chaque établissement. Plusieurs commerciaux peuvent intervenir auprès d'un même client.

A chaque commande passée par un client, on affecte un numéro d'accusé de réception (NAR) qui est unique. Une commande se réfère à un seul client et ne comporte qu'un lieu de livraison. Elle peut comporter plusieurs postes (ou lignes). Chaque poste est relatif à un seul article. Cependant, un poste peut donner lieu à plusieurs livraisons successives. Chaque quantité (qteliv) correspondant à une date de livraison distincte (dtliv) est appelée cadence. Le système d'information doit renseigner sur le détail des commandes: la date d'enregistrement, l'adresse de livraison, l'adresse de facturation, les postes, les dates et quantités de livraison de toutes les cadences d'un poste.

La base de données relationnelle suivante a été mise en place pour aider le département commercial dans son activité :

ARTICLE (nart, desart, pvar, livart, grammage, couleur, nossfam)

FAMILLE (nofam, designationfam)

SOUS-FAMILLE (nossfam, format, nofam)

CLIENT (ncli, adrfaccli, rscli)

ETABLISSEMENT (ncli, nordre, adrlivcli, nocomm)

COMMERCIAUX (nocomm, nomcom, telcom)

COMMANDE (nar, datecommande, ncli, nordre)

POSTE (nar, noordre, nart, qtecom)

CADENCE (nar, noordre, datliv, qteliv)

- 1) Déclarer les structures des tables POSTE et CADENCE
- 2) Ecrire en SQL les requêtes suivantes :

- a. Donner l'adresse de facturation du client qui a passé la commande 123
  - b. Donner par famille et sous famille la liste des articles
  - c. Donner par famille et sous famille le prix moyen des articles
  - d. Donner les numéros des commandes qui ont plus de cinq postes
  - e. Donner les numéros des commandes qui ont plus de cinq postes dont la quantité commandée est inférieure à 100
  - f. Donner les numéros des commandes qui n'ont fait l'objet d'aucune livraison.
  - g. Donner les articles qui ont été commandés à la fois par le client 15 et le client 19.
  - h. Donner les articles qui n'ont jamais été commandés
  - i. Donner l'article (son numéro et son prix) le plus cher
- 3) Ecrire une vue permettant d'obtenir une synthèse des quantités vendues par famille et sous famille d'articles

## REPONSES

- 1) Déclaration des tables POSTE et CADENCE

**Create table** POSTE (  
 nar Numeric **references** COMMANDE,  
 noordre integer,  
 nart numeric **references** ARTICLE,  
 qtecom entier,  
**Primary key** (nar, noordre))

**Create table** CADENCE (  
 nar Numeric,  
 noordre integer,  
 datliv date,  
 qtativ entier,  
**Primary key** (nar, noordre, datliv))  
**Foreign key** (nar, noordre) **references** POSTE

- 2) Ecriture de requêtes en SQL

a – Select adrfaccli  
 From CLIENT, COMMANDE  
 Where nar = 123 and CLIENT.ncli = COMMANDE.ncli

b- Select ARTICLE.nossfam, nofam, nart  
 From ARTICLE, SOUS-FAMILLE  
 Where ARTICLE.nossfam = SOUS-FAMILLE.nossfam  
 Order by ARTICLE.nossfam, nofam

c- Select ARTICLE.nossfam, nofam, AVG(pvart)  
 From ARTICLE, SOUS-FAMILLE  
 Where ARTICLE.nossfam = SOUS-FAMILLE.nossfam

GROUP BY ARTICLE.nossfam, nofam

d- Select nar  
From POSTE  
Group by nar  
Having count(\*) >5

e- Select nar  
From POSTE  
Where qtecom < 100  
Group by nar  
Having count(\*) >5

f- Select nar  
From COMMANDE  
Where not exists (select \* From CADENCE Where CADENCE.nar = COMMANDE.nar)

g- Select nart  
From COMMANDE, POSTE  
Where COMMANDE.nar = POSTE.nar and nocli = 15  
INTERSECT  
Select nart  
From COMMANDE, POSTE  
Where COMMANDE.nar = POSTE.nar and nocli = 19

h- Select nart  
From ARTICLE  
Where NOT EXISTS (Select \* From POSTE Where ARTICLE.nart = POSTE.nart)

i- Select nart, pvar  
From ARTICLE  
Where pvar = (Select MAX(pvar)  
From Article)

3) Déclaration de vue

**Create view VENTES** (sous-famille, famille, total-vente) **AS**  
select ARTICLE.nossfam, nofam, SUM(qtecom)  
from POSTE, ARTICLE, SOUS-FAMILLE  
where POSTE.nart = ARTICLE.nart and ARTICLE.nossfam = SOUS-FAMILLE.nossfam  
group by ARTICLE.nossfam, nofam

## SUJET N°5

On considère la base de données définie ci-dessous. L'annexe 2 présente le contexte applicatif de cette base de données.

**SESSIONNAIRE** (numsess, nomorg, nomsess, adresse)

**ORGANISME** (nomorg, specialité, adrorg)

**DISPONIBILITE** (numsess, nom-mois)

**CHOIX** (numsess, numsession, priorité)

**SESSION-CATALOGUE** (numsession, thème-session, duree-session, niveau-session, descriptif-session)

**SESSION-PROGRAMMEE** (numsession, date-debut, nom-centre, date-fin, date-limite-inscription, nombre-place, etat-session) avec etat-session = (en cours d'inscription, annulée, réalisée)

**COURS-SESSION** (numsession, date-debut, numero-cours, nointervenant, titre-cours, salle-cours, date-debut-cours, date-fin-cours, heure-debut-cours, heure-fin-cours)

**CENTRE-FORMATION** (nom-centre, ville-centre, adresse-centre)

**DEMANDE-INSCRIPTION** (no-demande, numsess, numsession, date-debut, date-demande, etat-demande) avec etat-demande = (acceptée, en attente, refusée)

**INSCRIPTION** (numinsc, no-demande, date-insc, etat-insc) avec etat-insc = (en cours, annulée)

**INTERVENANT** (nointervenant, nom-int, adr-int, type-int, nom-employeur)

**INTERVENTION** (nointervention, nointervenant, numsession, date-debut, duree, taux-horaire)

Nom logique	Libellé
<b>CENTRE_FORMATION</b>	Lieux où se déroulent les sessions programmées
<b>CHOIX</b>	Choix de sessions effectués par les sessionnaires avec pour chaque choix priorité
<b>COURS_SESSION</b>	Une session est découpée en un ensemble de cours
<b>DEMANDE_INSCRIPTION</b>	Demande d'inscription formulée par un sessionnaire. Elle peut être acceptée, en attente ou refusée
<b>DISPONIBILITE</b>	Disponibilités données par les sessionnaires dans les questionnaires. Les disponibilités sont exprimées en mois. Par exemple un sessionnaire a indiqué qu'il était disponible au mois de mars et de septembre
<b>INSCRIPTION</b>	Inscription à une session programmée. Elle peut être en cours ou annulée
<b>INTERVENANT</b>	Personne assurant des cours dans cet organisme de formation
<b>INTERVENTION</b>	Intervention assurée par un intervenant à une session réalisée. Cette intervention est quantifiée avec un nombre d'heures et un taux horaire
<b>ORGANISME</b>	Organisme d'appartenance des sessionnaires
<b>SESSIONNAIRE</b>	Personne appartenant à un organisme pour lequel l'ENGREF assure la formation des élèves
<b>SESSION_CATALOGUE</b>	Formation apparaissant dans le catalogue envoyé aux sessionnaires
<b>SESSION_PROGRAMMEE</b>	Formation mise au programme. Les sessionnaires ne peuvent demander d'inscriptions que pour ces formations. Une session du catalogue peut être programmée plusieurs fois à des moments différents.

- 1) Déclarer la structure de la table COURS-SESSION
- 2) Exprimer en SQL les requêtes suivantes :

- a. Combien y a-t-il eu de sessions programmées dans l'année qui ont été annulées
  - b. Quel est le nom du centre de formation dans lequel devra se rendre le sessionnaire qui a obtenu l'inscription 15
  - c. Donner les noms des sessionnaires qui ont donné dans leurs réponses aux questionnaires plus de deux mois de disponibilité
  - d. Donner les noms des sessionnaires qui ont donné dans leur réponses aux questionnaires au moins deux sessions avec la priorité = 1
  - e. Quelles sont les sessions du catalogue qui n'ont pas été programmées
  - f. Pour la session 28 du catalogue, donner toutes les sessions qui sont programmées dans le centre de Marseille
  - g. Quel est par session, le nombre de sessions programmées
- 3) Définir une vue permettant de considérer une session du catalogue avec toutes ses sessions programmées ainsi que leur centre
  - 4) Exprimer la requête f sur cette vue

## REPONSES

Déclaration de la table SESSION\_PROGRAMMEE

```
Create table SESSION_PROGRAMMEE (
Numsession, Numeric references SESSION_CATALOGUE,
Date_debut date,
Nom-centre chaine references CENTRE_FORMATION,
Dat_fin date,
Date_limite_inscription date ,
Nombre_place entier,
Primary key (numsession, date_debut))
```

2)

```
A – Select count(*)
From SESSION_ANNULEE
Where dateannul between 1/01/05 and 31/12/05
```

```
B – Select nom_centre
From INSCRIPTION I, DEMANDE_INSCRIPTION D, SESSION_PROGRAMMEE S
Where I.no_demande = D.no_demande and D.numsession = S.numsession and D.date_debut =
S.date_debut and num_insc = 15
```

```
C – Select nomsess
From SESSIONNAIRE S
Where nomsess IN (select .numsess
                  From DISPONIBILITE
                  Group by D.numsess
                  Having count(*) > 2)
```

```
D – Select nomsess
From SESSIONNAIRE S
```

```
Where numsess IN (select .numsess
                  From CHOIX
                  Where priorite = 1
```

```
Group by D.numsess
Having count(*) > 2)
```

```
E - Select numsession
From SESSION_CATALOGUE
Where numsession not in (select numsession
                        From SESSION_PROGRAMMEE)
```

```
ou
Select numsession
From SESSION_CATALOGUE SC
Where not exists (select *
                 From SESSION_PROGRAMMEE SP
                 Where SC.numsession = SP.numsession)
```

```
F - Select numsession, date_debut
From SESSION_PROGRAMMEE SP, CENTRE_FORMATION CF
Where numsession = 28 and SP_nom_centre = CF_nom_centre and ville_centre = 'Marseille'
```

```
G - Select numsession, count(*)
From SESSION_PROGRAMMEE
```

```
3)
Create view SESSION_OUVERTE AS
(select numsession, date_debut, ville_centre
 From SESSION_PROGRAMMEE SP, CENTRE_FORMATION CF
 Where SP_nom_centre = CF_nom_centre
```

```
4) select numsession, date_debut
From SESSION_OUVERTE
Where ville_centre = 'Marseille'
```

## **ANNEXE 2 : DESCRIPTION DU DEPARTEMENT "FORMATION CONTINUE"**

L'ENGREF est issue de la fusion en 1965 de deux écoles:

- l'école Nationale du Génie Rural,
- l'école Nationale des Eaux et des Forêts.

C'est une école d'application d'écoles d'ingénieurs. A ce titre, elle assure la formation de ses élèves (fonctionnaires, civils et étrangers). Depuis 1968, elle a en charge la formation continue de ses anciens élèves.

La formation continue de l'école se présente sous la forme de sessions de courte durée (3 à 5 jours) à raison d'une centaine par an.

On peut essentiellement distinguer des formations générales (sciences de base, sciences humaines, informatique et Bureautique) et des formations techniques (gestion des milieux naturels et environnement, agro-alimentaire, aménagement du territoire).

Au fil du temps, la formation continue de l'école a diversifié ses sessions et s'est ouverte à un public de plus en plus large. Elle n'accueille pas uniquement ses anciens élèves mais aussi d'autres ingénieurs, techniciens et divers personnels administratifs; il viennent tous essentiellement des organismes cités ci-après:

- le Ministère de l'Agriculture,
- l'Office National des Forêts,
- le Ministère de l'Environnement,
- divers instituts et offices
- des établissements privés.

Géographiquement, les sessions sont organisées pour la plupart dans les trois centres de l'école: Paris, Nancy et Montpellier.

Le fonctionnement du département "Formation Continue" est organisé autour de quatre activités complémentaires.

## I. L'ACTIVITE D'ETABLISSEMENT DES PROGRAMMES

Au mois de juin de l'année N, la direction de la formation continue établit un catalogue des sessions qu'elle envoie à l'ensemble des sessionnaires potentiels. Un questionnaire est joint au catalogue pour permettre à chaque destinataire (s'il le désire) d'exprimer les sessions souhaitées ainsi que ses mois de disponibilité. Les réponses aux questionnaires sont des informations très utiles qui permettent d'établir des statistiques sur les sessions les plus demandées, sur les mois de disponibilité apparaissant le plus souvent dans les réponses....

Le dépouillement des questionnaires conduit à programmer un certain nombre de sessions. La liste des sessions programmées constitue le programme de la Formation Continue. Ce programme est envoyé à chaque sessionnaire potentiel à la mi-novembre de l'année N.

## II. L'ACTIVITE DE GESTION DES INSCRIPTIONS

Les personnes souhaitant s'inscrire à des sessions envoient dans le courant de l'année N+1 des demandes d'inscriptions pour les sessions programmées. La priorité des inscriptions est établie selon l'ordre chronologique de leur arrivée.

Des demandes d'inscription peuvent être refusées si elles parviennent au département formation continue au delà de la date limite d'inscription ou si le nombre de places disponibles est atteint ou encore si l'ONF refuse la demande d'inscription (l'organisme national des forêts contrôle chaque demande et peut la refuser si par exemple le sessionnaire a suivi de nombreuses sessions durant les trois dernières années).

Un sessionnaire peut demander l'annulation d'une inscription. Une demande d'inscription en attente peut alors être acceptée si les convocations n'ont pas encore été envoyées.

### III. L'ACTIVITE DE GESTION DES SESSIONS

Un mois avant le début de la session, une lettre de convocation est envoyée aux sessionnaires accompagnée du programme détaillé de la session.

Une session peut être annulée si le nombre d'inscriptions est inférieur au seuil d'annulation. Dans ce cas une lettre d'annulation est envoyée aux sessionnaires et aux intervenants.

### IV. L'ACTIVITE DE GESTION DES INTERVENTIONS

Les sessions sont réalisées par des intervenants extérieurs. Pour chaque intervenant participant à une session, le département formation continue établit une fiche d'intervention mentionnant: le nom, le prénom, l'adresse, le type de l'intervenant (fonctionnaire, secteur privé,...), le nom de son employeur et le nombre d'heures effectuées pendant la session. Envoyée au département comptabilité, cette fiche permet d'effectuer le paiement des intervenants.