

<p align="center"><b>Série 1 Révisions PL/SQL</b> <b>Tableau PL/SQL – Exception - Package</b></p>
---

## Sorties en minibus

### Base de données :

#### Tables relationnelles

```
create table minibus (  
    nominibus      number(2),  
    capacite       number(2) not null,  
    constraint pk_minibus primary key (nominibus)  
);  
  
create table sortie (  
    nosortie       number(4),  
    libelle        varchar2(50) not null,  
    constraint pk_sortie primary key (nosortie)  
);  
  
create table adherent (  
    noadherent     number(4),  
    nomadh         varchar2(30) not null,  
    prenomadh      varchar2(30) not null,  
    constraint pk_adherent primary key (noadherent)  
);  
  
create table inscription_sortie (  
    noadherent     number(4),  
    nosortie       number(4),  
    date_inscription date not null,  
    nominibus      number(2),  
    constraint pk_inscription_sortie primary key (noadherent, nosortie)  
);  
  
create table ERREURS (  
    message        varchar2(255)  
);  
  
Contraintes d'intégrité  
alter table inscription_sortie  
    add constraint fk_inscription_adh foreign key (noadherent) references adherent  
    add constraint fk_inscription_sortie foreign key (nosortie) references sortie  
    add constraint fk_inscription_minibus foreign key (nominibus) references minibus;
```

```
insert into minibus values (1, 6);  
insert into minibus values (2, 8);
```

```

--
insert into sortie values (1, 'City tour de Paris');
insert into sortie values (2, 'City tour de Lyon');
insert into sortie values (3, 'Excursion Monte Carlo');
--
insert into adherent values (1, 'Assie','Alexandre');
insert into adherent values (2, 'Bourges','Amelie');
insert into adherent values (3, 'Brioude','Damien');
insert into adherent values (4, 'Bellaton','Maxime');
insert into adherent values (5, 'Eouzan','Ophelie');
insert into adherent values (6, 'Colombain','Charly');
insert into adherent values (7, 'Belotti','Tom');
insert into adherent values (8, 'Zeichner','Kevin');
insert into adherent values (9, 'Descroix','Thomas');
insert into adherent values (10, 'Razy','Victor');
insert into adherent values (11, 'Montelimard','Virginie');
insert into adherent values (12, 'Mancini','Corentin');
insert into adherent values (13, 'ElAzhar','Adil');
insert into adherent values (14, 'LeDonge','Romain');
insert into adherent values (15, 'Rebeca','Julien');
insert into adherent values (16, 'Varagnat','Kevin');
--
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,1,'10/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,3,'9/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,5,'15/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,7,'1/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,8,'11/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,10,'16/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,11,'4/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,12,'10/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,13,'10/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,14,'5/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,15,'20/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,16,'20/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,4,'21/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,6,'1/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (1,9,'14/08/14');

insert into inscription_sortie(nosortie,noadherent,date_inscription) values (2,2,'10/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (2,4,'15/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (2,10,'18/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (2,6,'16/08/14');
insert into inscription_sortie(nosortie,noadherent,date_inscription) values (2,8,'14/08/14');

commit ;

```

## Écrire le code PL/SQL pour répondre aux questions suivantes :

### 1. Créer un package nommé **pksorties**.

Dans ce package, déclarer un type tableau PL/SQL contenant des enregistrements dont les champs sont les attributs de la table **minibus**.

Déclarer une variable de ce type nommée **v\_tab\_minibus**.

Exemple d'utilisation ultérieure :

`v_tab_minibus(i).nominibus`

`v_tab_minibus(i).capacite`

### 2. Développer la fonction **les\_minibus** sans argument, qui retourne le nombre de minibus et initialise le tableau **v\_tab\_minibus** où les minibus sont **triés** par ordre décroissant de capacité (nombre de places).

Vous gérerez les exceptions (gestionnaire OTHERS) en plaçant le message d'erreur ainsi que la date de l'erreur dans la table **ERREURS**. La fonction renverra alors -1.

**NB** : la fonction appartiendra au package **pksorties**.

### 3. Développer la fonction **affectations\_minibus**, avec un argument : le numéro de sortie, qui affecte un minibus aux adhérents inscrits à la sortie dont le numéro est fourni en argument.

La fonction retourne le nombre de minibus affectés à la sortie.

Les adhérents inscrits à la sortie seront placés dans les minibus dans l'ordre de leur inscription (date d'inscription croissante).

On supposera que tous les minibus sont disponibles pour la sortie.

Il peut y avoir plus d'inscriptions à la sortie que de places dans les minibus (ne pas traiter ce cas).

Vous gérerez les exceptions (gestionnaire OTHERS) en plaçant le message d'erreur ainsi que la date de l'erreur dans la table **ERREURS**. La fonction annulera alors les mises à jour effectuées et renverra -1

**NB** : la fonction appartiendra au package **pksorties**. Vous utiliserez la fonction **les\_minibus** et le tableau de minibus **v\_tab\_minibus**.

### Aide :

L'algorithme général pour la fonction **affectations\_minibus** est le suivant :

Tant qu'il y a des adhérents inscrits à la sortie (date d'inscription croissante) et qu'il reste des minibus :

    S'il ne reste plus de place dans le minibus en cours

        Changer de minibus

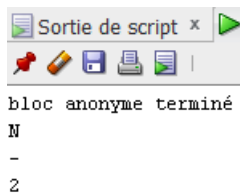
    S'il reste des minibus

        Mettre à jour l'inscription à la sortie (avec le numéro de minibus)

## Exemple de tests (avec SQL Developer) et leurs résultats :

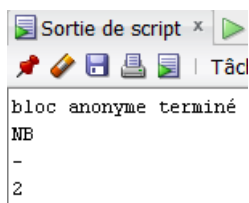
### 1. La fonction **les\_minibus** :

```
variable n number
exec :n := pksorties.les_minibus;
print n
```



### 2. La fonction **affectations\_minibus**

```
var nb number
begin
:nb := pksorties.affectations_minibus(1);
end;
/
print nb
```



La table inscription à ce stade :

NOADHERENT	NOSORTIE	DATE_INSCRIPTION	NOMINIBUS
1	1	10/08/14	2
3	1	09/08/14	2
5	1	15/08/14	1
7	1	01/08/14	2
8	1	11/08/14	1
10	1	16/08/14	1
11	1	04/08/14	2
12	1	10/08/14	2
13	1	10/08/14	2
14	1	05/08/14	2
15	1	20/08/14	1
16	1	20/08/14	1
4	1	21/08/14	(null)
6	1	01/08/14	2
9	1	14/08/14	1
2	2	10/08/14	(null)
4	2	15/08/14	(null)
10	2	18/08/14	(null)
6	2	16/08/14	(null)
8	2	14/08/14	(null)

```

var nb number
begin
:nb := pksorties.affectations_minibus(11);
end;
/
print nb

```

Sortie de script x

bloc anonyme terminé  
NB  
--  
-1

```
select * from erreurs;
```

Sortie de script x Résultat de requête x

Toutes les lignes extraites : 1 en 0,029 secondes

MESSAGE

1 Affectations : numéro de sortie inconnu le 14/10/2015 14:18:33