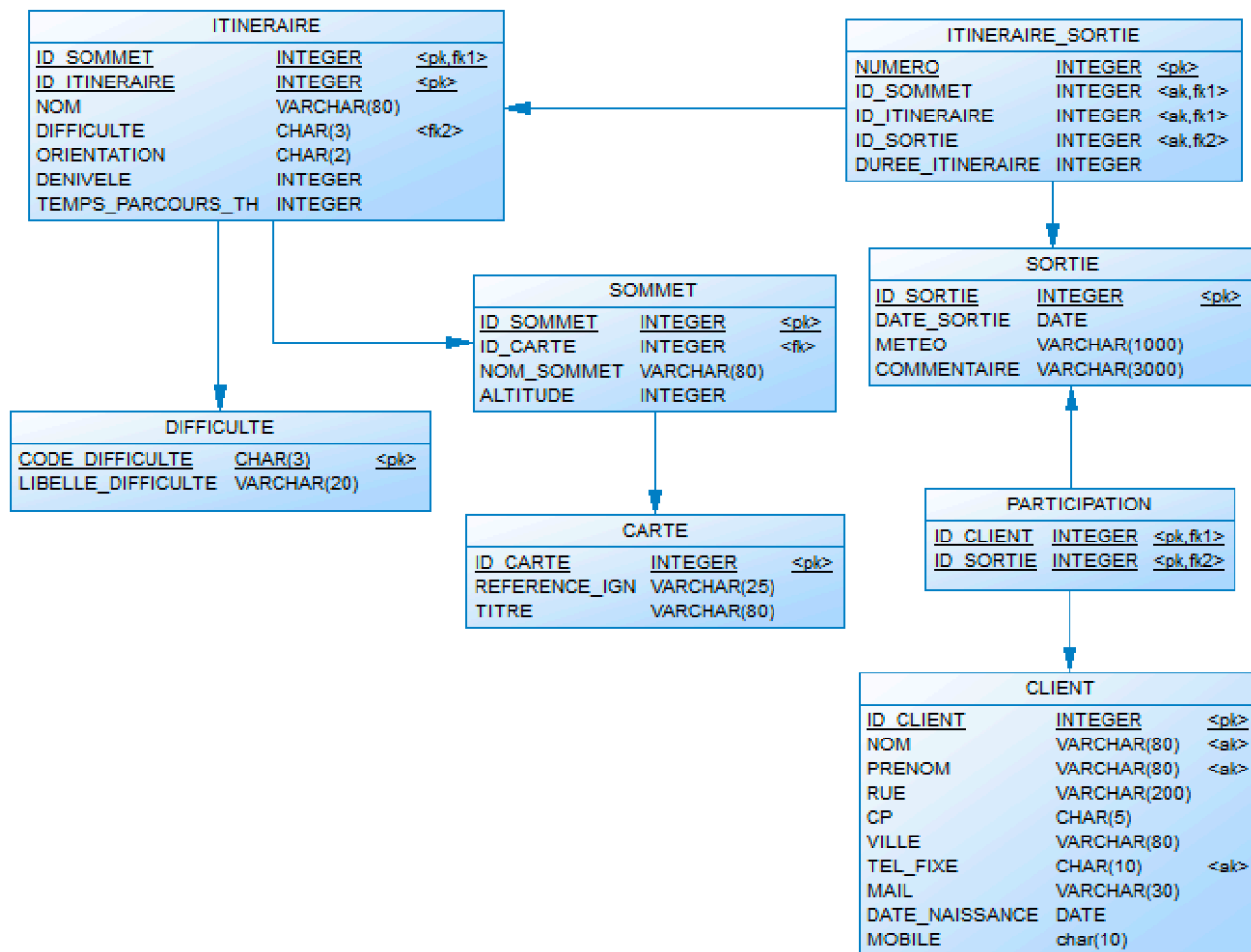


**Ce TD fait suite au TD2.**

**Vous devez au préalable exécuter le script « Correction complete TD2.sql » correspondant à la correction du TD2.**

**Enregistrez au fur et à mesure vos réponses aux questions dans un fichier texte**

**MPD :**



**Question 1 :** Utilisation d'une séquence

Ajoutez les données suivantes dans la table **CLIENT**. **Vous utiliserez la séquence créée sur la table CLIENT lors du TD2. Votre séquence doit s'appeler CLIENT\_SEQ.**

N°	Nom	Prénom	Rue	CP	Ville	Tel fixe	Mail	Date naissance	Mobile
1	GILLOT	André	rue des martyrs	74000	Annecy	0450487932	andre.guillot@myspace.fr	12/06/1954	0615487932
2	MARTIN	Laurent	allee de la renaissance	69100	Villeurbanne	0450030478	lmartin@noos.com	04/11/1980	0656030478

*Remarque : pour l'utilisation d'une séquence, cf. Annexe 2 TD2 (située en fin de sujet).*

**Vous vérifierez que les nouveaux clients ont bien pour numéro les valeurs 1 et 2. Sinon, supprimez et recréez la séquence et relancez les deux INSERT précédents.**

```
Delete from CLIENT;
DROP SEQUENCE CLIENT_SEQ;
CREATE SEQUENCE CLIENT_SEQ ... ;
INSERT...
```

### Question 2 :

Regardez puis exécutez le script `insertionAMM.sql` permettant d'insérer des données dans les tables CLIENT (9 tuples supplémentaires), CARTE, SOMMET, DIFFICULTE, ITINERAIRE, SORTIE, PARTICIPATION et ITINERAIRE\_SORTIE.

**ATTENTION, votre séquence devra être nommée CLIENT\_SEQ pour que les insertions dans la table CLIENT puissent être réalisées.**

**A la fin des insertions vous devez avoir le nombre suivant de tuples par table : CLIENT (11 enregistrements numérotés de 1 à 11), CARTE (18 tuples), SOMMET (66), DIFFICULTE (8), ITINERAIRE (76), SORTIE (112), PARTICIPATION (342) et ITINERAIRE\_SORTIE (72). Sinon, recommencez tout depuis la question 1.**

*Remarque : afin que le jeu d'essai soit toujours d'actualité, les données de type date de la table SORTIE sont exprimées en relatif par rapport à la date du jour du serveur (`current_date`).*

### Question 3 :

Pour faciliter la gestion de l'état des clients et déterminer plus rapidement les clients actifs de ceux qui ne le sont pas, ajoutez une colonne `etat_client` (dans la table CLIENT) qui peut prendre les valeurs 'A' ('actif', valeur par défaut) et 'NA' ('non actif').

Écrivez l'instruction SQL qui permet d'effectuer la modification de structure souhaitée. Mettez ensuite à jour l'état de chaque client en le faisant passer à 'non actif' si le client n'a pas participé à une sortie depuis plus de 2 ans.

*Indications : la sous-requête SELECT doit trouver les clients actifs (i.e. ayant fait une sortie depuis moins de 2 ans) et la requête principale doit mettre à jour les clients ne se trouvant pas dans ce cas (NOT IN). L'inverse ne fonctionne pas car il suffira qu'un client ait effectué une sortie il y a plus de 2 ans malgré qu'il ait fait une sortie il y a moins de 2 ans pour qu'il soit considéré comme non actif.*

Résultat :

	id_client integer	nom character varying (80)	prenom character varying (80)	rue character varying (200)	cp character (5)	ville character varying (80)	tel_fixe character (10)	mail character varying (30)	date_naissance date	mobile character (10)	etat_client character (2)
1	1	GILLOT	André	rue des martyrs	74000	Annecy	0450487932	andre.guillot@myspace.fr	1954-06-12	0615487932	A
2	2	MARTIN	Laurent	allee de la renaissance	69003	Villeurbanne	0450030478	lmartin@noos.com	1980-11-04	0656030478	A
3	3	FACILE	Isabelle	montee de la ruche	74940	Annecy le Vieux	0450325641	facisa@yahoo.fr	1981-05-27	0620325641	A
4	4	ALFONSO	Gérard	impasse des Lylas	38000	Grenoble	0450751511	gegea@gmail.com	1964-01-05	0689751511	A
5	5	CARRE	Lucile	rue des cygnes	74600	Seynod	0450102389	lulukaribou@yahoo.fr	1983-10-10	0689754517	A
6	6	MIFRIDE	Sophie	allee du stade	73000	Chambery	0450275361	smifride@gmail.com	1978-08-27	0690753410	A
7	7	MARTIN	Stéphane	allee de la renaissance	69003	Villeurbanne	0450841020	smartin@noos.com	1977-03-10	0622841000	A
8	8	MICHON	Antoine	impasse Grenette	75003	Paris	0450361877	seb_michaut@gmail.com	1979-07-14	0625321475	A
9	9	JOSSE	Francis	montee des Airelles	74940	Annecy le Vieux	0450455550	utul@campcamp.org	1980-02-14	0610355350	A
10	10	FLICK	Marcel	boulevard de la Mandallaz	74960	Cran Gevrier	0450661245	m.flick@caramail.fr	1962-04-23	0610355757	NA
11	11	MERGUES	Clément	boulevard de Teigne	74000	Annecy	0450184569	clemerg@laposte.net	1984-09-22	0610503040	NA

### Question 4 :

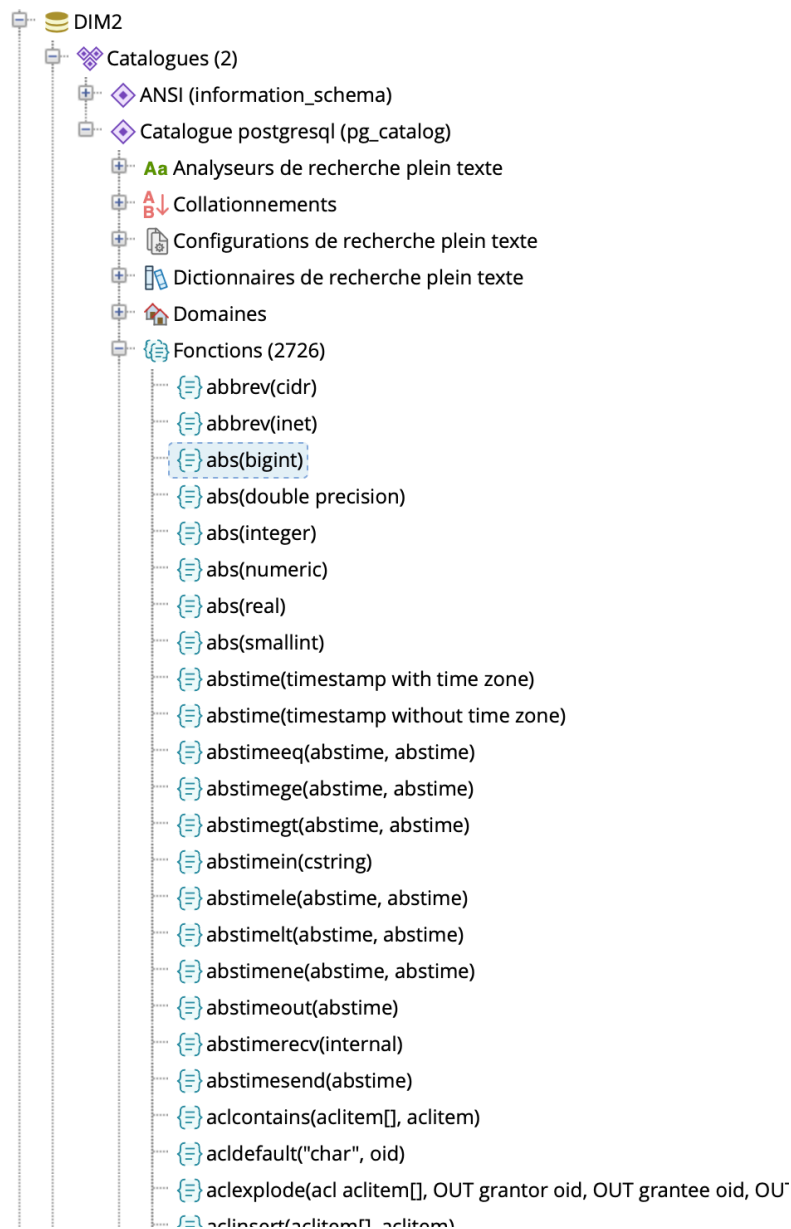
À partir des niveaux de difficulté, affichez le public de chaque itinéraire en vous appuyant sur la table des correspondances suivante. *Indications : utilisez l'instruction CASE*

Difficulté	Public
F, PD-, PD	Tout public
PD+, AD-, AD	Activité sportive occasionnelle
AD+, D	Sportif confirmé

	sommet text	itineraire text	Public text
1	1 - Aiguille des Calvaires	1 - l'Aiguille	Tout public
2	1 - Aiguille des Calvaires	2 - sommet sud-est	Tout public
3	1 - Aiguille des Calvaires	3 - sommet sud-ouest	Tout public
4	1 - Aiguille des Calvaires	4 - sommet nord-ouest	Activité sportive occasionnelle
5	1 - Aiguille des Calvaires	5 - sommet nord-est	Tout public
6	2 - Col de Balme	1 - voie normale du versant nord-ouest	Tout public
7	2 - Col de Balme	2 - voie normale du versant sud-est	Sportif confirmé
8	3 - Tête Pelouse	1 - par la combe du Grand Crêt	Tout public
9	4 - Passage du Père	1 - par la combe de Paccaly	Tout public
10	5 - Trou de la Mouche	1 - par la combe de Paccaly	Activité sportive occasionnelle
11	5 - Trou de la Mouche	2 - par la combe du Grand Crêt	Activité sportive occasionnelle
12	6 - Ambrevetta	1 - par la combe de Tardevant	Tout public
13	7 - Col des Verts	1 - par le Val du Bouchet	Activité sportive occasionnelle
14	8 - Montagnes de Sulens	1 - voie normale par Plan Bois	Tout public
15	8 - Montagnes de Sulens	2 - circuit des Montagnes de Sulens par Plan Bois	Tout public
16	9 - Pointe de Mandallaz	1 - versant sud-ouest	Tout public
17	10 - Col des Porthets	1 - par le vallon des Fours	Tout public
18	11 - Pointes Sud de la Blonnière	1 - combe sud ouest	Tout public
19	11 - Pointes Sud de la Blonnière	2 - combe à Marion - couloir Nord des Aravis	Activité sportive occasionnelle
20	13 - Tour du Jallouvre	1 - par le lac de Lessy	Tout public
21	14 - Col de Balafrasse	1 - col sud-ouest, par la combe est de la Pointe Blan...	Activité sportive occasionnelle
22	14 - Col de Balafrasse	2 - col est, versant sud	Tout public
23	14 - Col de Balafrasse	3 - col ouest, versant sud	Tout public
24	16 - Tournette	1 - voie normale du versant nord-est	Tout public

(76 lignes)

*Remarque : la liste des fonctions PostgreSQL est visualisable sous pg\_catalog / fonctions. Pour plus de détail sur ces fonctions, utilisez la documentation.*



**Question 5 :** création d'une table en utilisant l'instruction `CREATE TABLE <nom_table> AS SELECT...`  
 Créez une nouvelle table nommée `SOMMET_CARTE` à partir des tables `CARTE` et `SOMMET` (vous renommerez la colonne `TITRE`).

Données de la table `SOMMET_CARTE` après création :

	id_sommet integer	nom_sommet character varying (80)	altitude integer	reference_ign character varying (25)	titre_carte character varying (80)
1	1	Aiguille des Calvaires	2322	3431 Ouest	Lac d'Annecy
2	2	Col de Balme	2481	3431 Ouest	Lac d'Annecy
3	3	Tête Pelouse	2537	3431 Ouest	Lac d'Annecy
4	4	Passage du Père	2377	3431 Ouest	Lac d'Annecy
5	5	Trou de la Mouche	2453	3431 Ouest	Lac d'Annecy
6	6	Ambrevetta	2463	3431 Ouest	Lac d'Annecy
7	7	Col des Verts	2595	3431 Ouest	Lac d'Annecy
8	8	Montagnes de Sulens	1839	3534 Ouest	Les 3 Vallées - Modane - PN de la Vanoise
9	9	Pointe de Mandallaz	2277	3534 Ouest	Les 3 Vallées - Modane - PN de la Vanoise
10	10	Col des Porthets	2175	3534 Ouest	Les 3 Vallées - Modane - PN de la Vanoise
11	11	Pointes Sud de la Blonnière	2369	3534 Ouest	Les 3 Vallées - Modane - PN de la Vanoise
12	13	Tour du Jallouvre	2250	3431 Ouest	Lac d'Annecy
13	14	Col de Balafrasse	2253	3431 Ouest	Lac d'Annecy
14	16	Tournette	2351	3432 Est	Albertville
15	17	Pointe de Chaurionde	2173	3432 Ouest	Massif des Bauges
16	19	Les 3 cols	3323	3335 Est	Bourg d'Oisans - L'Alpe d'Huez
17	20	Chamechaude	2082	284S	Mischabel
18	22	Croix de Chamrousse	2248	IGMCH-3600-7015	Laguna Del Rio
19	23	Grand Sorbier	2536	IGMCH-3600-7015	Laguna Del Rio
20	24	Porte des Aravis	2390	3431 Ouest	Lac d'Annecy
21	25	Cîme de la Jasse	2478	IGMCH-3600-7015	Laguna Del Rio
22	27	Dent du Pra	2623	IGMCH-3600-7015	Laguna Del Rio
23	28	Grand Ar	2484	3432 Ouest	Massif des Bauges

(66 lignes)

La nouvelle table a-t-elle des contraintes (PK, FK,...). Si oui, de quels types ?

#### Question 6 : création d'une table temporaire globale (cf. annexe 1)

On souhaite obtenir le nombre total de participants (différents) pour chaque itinéraire (autrement dit, combien de personnes ont emprunté chaque itinéraire) et le nombre total de participants (différents) pour chaque sommet (cf. écran suivant). Une personne ayant pris plusieurs fois un même itinéraire ne doit être comptabilisée qu'une seule fois.

Indications :

- 1<sup>ère</sup> étape : créez une table temporaire globale, dans laquelle les informations vont être ajoutées au fur et à mesure de leur calcul.  
Utilisez la clause `ON COMMIT PRESERVE ROWS` lors de la création de la table temporaire globale.
- 2<sup>ème</sup> étape : Ajoutez les informations relatives à chaque itinéraire (insert)
- 3<sup>ème</sup> étape : Ajoutez les informations relatives à chaque sommet (update)

Visualisez les données de la table temporaire (Cf. tableau suivant).

Puis, supprimez les données de la table et supprimez la table.

	id_sommet integer	id_itineraire integer	nombreparticipantsitineraire numeric (10)	nombreparticipantssommet numeric (10)
1	1	1	3	5
2	1	2	5	5
3	1	3	0	5
4	1	4	4	5
5	1	5	0	5
6	2	1	0	0
7	2	2	0	0
8	3	1	2	2
9	4	1	0	0
10	5	1	2	2
11	5	2	2	2
12	6	1	2	2
13	7	1	2	2
14	8	1	2	2
15	8	2	1	2
16	9	1	2	2
17	10	1	2	2
18	11	1	2	2
19	11	2	0	2
20	13	1	2	2
21	14	1	2	3
22	14	2	2	3
23	14	3	2	3
24	16	1	5	5
25	17	1	5	5
26	19	1	5	5
27	20	1	1	1
28	22	1	2	2

(76 lignes)

### Question 7 : création d'une vue (cf. annexe 2)

Créer une vue qui permet de connaître, pour chaque client, le nombre de participations.

	id_client integer	nom character varying (80)	prenom character varying (80)	Nombre participations bigint
1	1	GILLOT	André	38
2	2	MARTIN	Laurent	63
3	3	FACILE	Isabelle	22
4	4	ALFONSO	Gérard	17
5	5	CARRE	Lucile	12
6	6	MIFRIDE	Sophie	35
7	7	MARTIN	Stéphane	30
8	8	MICHON	Antoine	24
9	9	JOSSE	Francis	100
10	10	FLICK	Marcel	0
11	11	MERGUES	Clément	1

Remarque : pour visualiser les vues, utilisez la vue système `views` : `select * from information_schema.views;`

### Question 8 :

Créer une vue affichant pour chaque itinéraire, la date de la dernière sortie (afficher l'itinéraire même si celui-ci n'a jamais été emprunté). Utilisez une jointure externe.

	id_sommet integer	id_itineraire integer	DER SORTIE date
1	1	1	2015-09-26
2	1	2	2015-09-26
3	1	3	[null]
4	1	4	2012-10-19
5	1	5	[null]
6	2	1	[null]
7	2	2	[null]
8	3	1	2013-01-18
9	4	1	[null]
10	5	1	2017-08-10
11	5	2	2017-08-10
12	6	1	2014-03-02
13	7	1	2014-03-01
14	8	1	2013-11-23
15	8	2	2014-02-09
16	9	1	2014-12-13
17	10	1	2013-01-26
18	11	1	2013-12-13
19	11	2	[null]
20	13	1	2014-01-04
21	14	1	2013-12-07
22	14	2	2013-12-07
23	14	3	2019-01-19
24	16	1	2014-02-22
25	17	1	2015-03-21
26	19	1	2015-02-14
27	20	1	2015-03-10
28	22	1	2015-11-27

(76 lignes)

### Question 9 :

Ecrire une requête qui permette d'afficher un message en fonction du nombre de participants à une sortie.

Nombre de participants / sortie	Message
Moins de 3 (non compris)	Peu
Entre 3 et 6	Normal
7 et +	Beaucoup

Indications : utilisez l'instruction CASE

	id_sortie integer	date_sortie date	Nombre participants text
1	146	2019-10-19	Normal
2	139	2019-10-09	Normal
3	138	2019-03-16	Normal
4	137	2019-03-11	Normal
5	136	2019-03-10	Normal
6	135	2019-03-09	Normal
7	134	2019-03-03	Peu
8	133	2019-03-02	Normal
9	132	2019-02-22	Normal
10	131	2019-02-21	Normal
11	130	2019-02-20	Normal
12	129	2019-02-19	Normal
13	128	2019-02-18	Normal
14	127	2019-02-17	Normal
15	126	2019-02-09	Peu
16	125	2019-01-26	Normal
17	124	2019-01-19	Peu
18	123	2019-01-12	Normal
19	122	2019-01-05	Normal
20	121	2018-12-29	Normal
21	120	2018-02-09	Peu
22	119	2018-12-04	Peu
23	118	2018-11-11	Peu
24	101	2018-02-10	Beaucoup
25	100	2018-01-28	Normal
26	99	2018-01-27	Normal
27	86	2018-01-14	Normal
28	85	2018-01-13	Peu

(112 lignes)

## ANNEXES

### 1. Les tables temporaires

PostgreSQL offre la possibilité de créer des tables temporaires pour stocker des informations le temps d'une session ou d'une transaction. Les données stockées dans une table temporaire créée avec l'ordre `CREATE TEMPORARY TABLE` sont accessibles uniquement depuis la session qui a créé les données. En fait chaque session ne voit que ses propres données.

La spécificité d'une table temporaire réside dans le fait que les données insérées dans cette table ne restent présentes que le temps de la transaction. La table, quant à elle, reste bien sûr en place et il est possible de l'utiliser dans les autres transactions. Si l'on souhaite que les données insérées dans cette table soient stockées de façon plus persistante, il faut utiliser la clause `ON COMMIT PRESERVE ROWS` lors de la création de la table temporaire globale. Ainsi, les informations insérées dans cette table seront visibles à travers toutes les transactions pendant la durée de la session de l'utilisateur Oracle qui a inséré les informations dans la table.

### Syntaxe



```
CREATE TEMPORARY TABLE table (nom colonne type, ...) [ON COMMIT PRESERVE ROWS];
```

## **2. View (vue)**

Les vues sont des tables virtuelles "contenant" le résultat d'un SELECT.

### **Création : Syntaxe**

```
CREATE [OR REPLACE] VIEW nom  
[(colonnes,...)] AS SELECT ... ;
```

OR REPLACE

permet le remplacement de la description par la nouvelle requête si la vue existe déjà. En effet, la définition d'une vue ne peut être modifiée partiellement.

### **Exemple**

Vue retournant les clients du département 44.

```
SQL> create or replace view V_CLIENT44 as  
      select NOCLI, NOMCLI, CODE POSTAL, VILLE  
      from CLIENTS  
      where CODE_POSTAL between 44000 and 44999;
```

```
SQL> select * from V_CLIENT44;
```

### **Suppression : Syntaxe**

```
DROP VIEW nom;
```