

APPASSAMY RANJITH

MACHU ALEXANDRE

PHILIPPE YTHIER AYMERIC

INFO 1 GROUPE D 18

RENDU PARTIE 2 SAE 2.04

I. CREATION DES DIFFERENTES TABLES

A. LES TABLES PREDEFINIES

Lorsque nous avons commencé le travail demandé sur la création et l'exploitation des données, il y avait au préalable certaines tables prédéfinies. On peut notamment citer les tables suivants :

- Aliments
- Mois
- Oiseaux

Il s'agit de tables simple avec un numéro d'identifiant ainsi que un champ de libellé qui donne plus de sens. Exemple dans la table de « Mois » nous avons 1 en numéro du mois et « janvier » en libellé. Il a fallu donc simplement modifier les noms des différents colonnes pour qu'il puisse correspondre avec la base de données génères.

En ce qui concerne la table oiseaux, nous avons pris un peu plus de temps pour effectuer nos recherches et ainsi pouvoir des données précises et réel. En effet il a fallu compléter des champs tels que l'envergure, le poids, le nom scientifique de l'oiseau etc.

Voici des capture d'écran Excel qui peuvent illustrer les tables cités.

	A	B
1	nummois	nommois
2		1 janvier
3		2 fevrier
4		3 mars
5		4 avril
6		5 mai
7		6 juin
8		7 juillet
9		8 août
10		9 septembre
11		10 octobre
12		11 novembre
13		12 décembre

Table Mois

Table Alimentation

	A	B
1	numaliment	nomaliment
2		1 Graines
3		2 Fruits
4		3 Bourgeons
5		4 Invertébrés
6		5 Vers
7		6 Galnds
8		7 Œufs
9		8 Oisillions
10		9 Araignées

Table Espece

	A	B	C	D	E	F	G	H
1	NUMERO_ESPECE	NOM_SCIENTIFIQUE	NOM_ESPECE	POIDS_MIN_ESPECE	POIDS_MAX_ESPECE	ENVERGURE_MIN_ESPECE	ENVERGURE_MAX_ESPECE	LONGUEUR_ESPECE
2	1	Accenteur mouchet	Prunella Modularis	19	21	18	24	14
3	2	Bergeronnette grise	Motacilla alba	19	27	25	30	18
4	3	Bouvreuil Pivoine	Pyrrhula pyrrhula	26	38	22	26	15
5	4	Chardonneret élégant	Carduelis carduelis	14	18	21	25	12
6	5	Corneille noire	Corvus corone	540	600	93	104	47
7	6	Epervier d'Europe	Accipiter nisus	150	320	60	80	40
8	7	Etourneau sansonnet	Sturnus vulgaris	60	96	31	40	21
9	8	Faucon crécerelle	Falco tinnunculus	190	300	65	80	36
10	9	Mésange charbonnière	Parus major	16	21	23	26	14
11	10	Mésange huppée	Lophophanes cristatus	10	13	17	20	12
12	11	Mésange noire	Periparus ater	8	10	17	21	12
13	12	Pinson du Nord	Fringilla montifringilla	23	30	25	27	15

B. TABLES OBTENUES GRACE AUX RECHERCHES

Ensuite nous nous sommes intéressés aux tables pour lesquelles la recherche était indispensable. On peut parler des tables :

- Région
- Département
- Se Nourrit
- Mois de présence

Pour les deux premiers citées, nous avons décidé de utiliser des fichiers existants et d'apporter quelques modifications tels que insérer la ligne d'en tête, modifier le caractère de séparation etc.

Voici à quoi elles ressemblent ouvert dans un éditeur de texte (alternative de Excel pour les .csv) :

```

mois_presence.csv  regions-france.csv x
CSV > regions-france.csv
1 humero_region;nom_region
2 01;Guadeloupe
3 02;Martinique
4 03;Guyane
5 04;La Réunion
6 06;Mayotte
7 11;Île-de-France
8 24;Centre-Val de Loire
9 27;Bourgogne-Franche-Comté
10 28;Normandie
11 32;Hauts-de-France
12 44;Grand Est
13 52;Pays de la Loire
14 53;Bretagne
15 75;Nouvelle-Aquitaine
16 76;Occitanie
17 84;Auvergne-Rhône-Alpes
18 93;Provence-Alpes-Côte d'Azur
19 94;Corse
20

```

Table Region

```

regions-france.csv x  department.csv x
CSV > department.csv
1 humero_departement;numero_region;nom_departement;superficie_departement
2 01;84;Ain;5762
3 02;32;Aisne;7369
4 03;84;Allier;7340
5 04;93;Alpes-de-Haute-Provence;6925
6 05;93;Hautes-Alpes;5549
7 06;93;Alpes-Maritimes;4299
8 07;84;Ardeche;5529
9 08;44;Ardenne;5229
10 09;76;Ariege;4890
11 10;44;Aube;6004
12 11;76;Aude;6139
13 12;76;Aveyron;8735
14 13;93;Bouches-du-Rhone;5088
15 14;28;Calvados;5548
16 15;84;Cantal;5726
17 16;75;Charente;5956
18 17;75;Charente-Maritime;6864
19 18;24;Cher;7235
20 19;75;Correze;5857
21 2A;94;Corse-du-Sud;4014

```

Extrait de la Table Departement

Les deux autres tables étaient destinées à des informations spécifiques pour chaque oiseau. Pour cela nous avons effectué des recherches et nous avons atterri sur le site ci-dessous :

https://www.oiseauxdesjardins.fr/index.php?m_id=20112

Ce site regroupe des fichiers PDF dédié à chaque espèce d'oiseau. Elle regroupait donc plusieurs types d'informations dont l'alimentation ainsi que l'hibernation.

Carte d'identité



- **Poids** : 20 g
- **Longueur** : 14 cm
- **Envergure** : 20 cm
- **Alimentation** : principalement des insectes et quelques graines en hiver
- **Période de présence** :

J	F	M	A	M	J	J	A	S	O	N	D
---	---	---	---	---	---	---	---	---	---	---	---
- [Écoutez le chant de l'Accenteur mouchet](#)

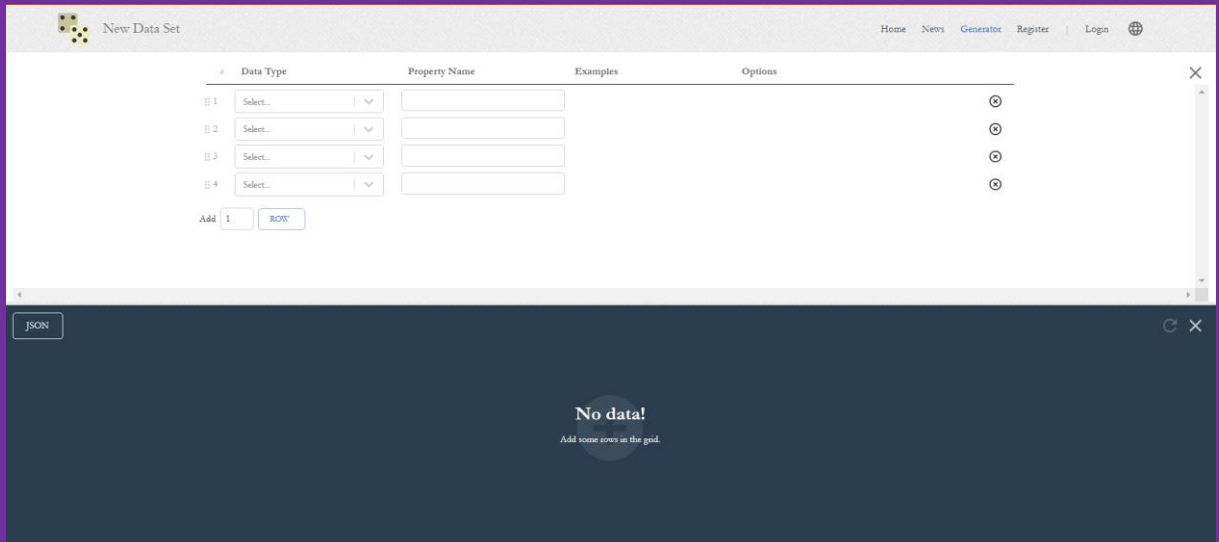
Pour la période de présence, si la case est coloriée, cela traduit la présence de l'espèce.

C. TABLES AUX DONNEES GENEREES

Les tables restantes, nous avons créé des données aléatoires pour avoir un jeu d'essai. Cette manœuvre a été effectuée pour les tables suivants :

- Adhérent
- Fiche
- Concerne

Pour cela nous avons utilisé le site suivant : <https://generatedata.com/generator>



The screenshot shows the 'New Data Set' interface. At the top, there are navigation links: Home, News, Generator, Register, and Login. The main area contains a table with the following structure:

#	Data Type	Property Name	Examples	Options
1	Select...	<input type="text"/>		<input checked="" type="radio"/>
2	Select...	<input type="text"/>		<input checked="" type="radio"/>
3	Select...	<input type="text"/>		<input checked="" type="radio"/>
4	Select...	<input type="text"/>		<input checked="" type="radio"/>

Below the table, there is an 'Add 1 ROW' button. At the bottom of the interface, a dark blue banner displays the message: 'No data! Add some rows in the grid.'

On peut choisir le type de données, le nom du champ, des exemples prédéfinies et des options supplémentaires pour que cela soit obtenu comme désiré.

```

adherent_v2.csv X
CSV > adherent_v2.csv
1 |hum_adherent;numero_departement;nom_adherent;prenom_adherent;rue_adresse;code_postal;commune;email;telephone_fixe;telephone_mobile;telephone_pro
2 |1;75;Van Aalsburg;Leonard;Ap #580-5895 Nullam St.;71520;Brive-la-Gallarde;vanaalsburg.leonard5185@hotmail.fr;367885885;726533053;689112627
3 |2;46;Royer;Courtney;711-1048 Nisi Rd.;35336;Limoges;c-royer2542@google.com;328226782;642884674;725252814
4 |3;88;Koopman;Raven;Ap #797-413 Tincidunt Av.;42152;Montigny-LA-s-Metz;raven.koopman@google.fr;2112131085;672588073;747357625
5 |4;59;Bouquindenhout;Seth;7035 Nuttum Street;74581;Sosissons;bouquindenhout_seth3322@yahoo.com;421327515;622401491;608824865
6 |5;67;Klein;Roary;P.O. Box 408, 9506 Fringilla Rd.;36556;Auxerre;k-roary2333@yahoo.com;425811476;755311556;726284365
7 |6;471;Lachapelle;Amethyst;653-4642 Vitae Avenue;27654;Schiltigheim;amethyst.lachapelle2562@google.fr;333810598;635666778;487568156
8 |7;48;Adrichem;Angelic;525-9375 Et Avenue;47566;Blois;angelica-adrichem@google.fr;551758571;678582888;781863020
9 |8;67;Chaput;Josiah;Ap #760-516 Integer Rd.;29233;Saint-LA;j_chaput6686@google.com;237471565;792675660;128914298
10 |9;41;Haannaads;Amal;4194 Interdum Rd.;2470;Carcassonne;hama17411@yahoo.fr;17323201;747344842;270885567
11 |10;72;Poulin;Madaline;478-6355 Nec Ave;93712;NAmes;p.madaline@hotmail.com;151578606;667663359;524521157
12 |11;54;Cruyssen;Jing;Ap #496-7345 Et Rd.;6583;LiNevin;c-wing@google.fr;496634898;738746836;41519416
13 |12;54;Boumeester;Quyn;571-7779 Faucibus St.;45423;Ajaccio;q.boumeester@yahoo.com;348875623;796247264;222611886
14 |13;33;Brisbis;Tana;Ap #890-3492 Scelerisque St.;38752;CrAoteil;btana2736@hotmail.fr;480268232;712463587;731255254
15 |14;46;Peeters;Neville;P.O. Box 277, 6849 Phasellus Ave;19847;Saint-Atienne-du-Rouvray;neville_peeters@yahoo.com;127867616;770972926;726343570
16 |15;22;Janvier;Adrienne;3676 Purus Ave;82455;Niort;ajjanvier9022@hotmail.com;276072774;6918380237;667455074
17 |16;26;Haak;Macey;P.O. Box 449, 2813 Massa Rd.;49438;Belfort;macey.haak4071@google.com;508555851;690923693;755965144
18 |17;46;Van Aalsburg;Todd;Ap #858-5163 Sed Av.;57334;Blethune;v.todd@hotmail.com;175808626;655376117;744669158
19 |18;12;Travers;Macaulay;Ap #279-4160 Mauris Rd.;45141;Rennes;m-travers266@yahoo.com;515654921;628452907;413154442
20 |19;91;Aaldenberg;Xenos;Ap #342-6959 A Rd.;19541;Biarritz;a.xenos@yahoo.fr;217641341;787842445;749340565
21 |20;53;Taille;Gai;P.O. Box 159, 4451 UT Road;21287;Tours;caintaille1573@hotmail.com;231276795;764173118;509765514
22 |21;60;Roggeveen;Roary;184-5513 Vehicula St.;78242;Blois;roggeveen.roary@google.fr;373677738;652166060;736035614
23 |22;65;Neuville;Aphrodite;P.O. Box 372, 3423 Nec, Road;36172;Pontarlier;n-aphrodite@google.fr;462825535;777164158;574475735
24 |23;31;Beauchene;Hope;P.O. Box 321, 2204 Ultricies Street;77444;Tarbes;hope.beauchene9242@yahoo.com;241655594;754784480;552663029
25 |24;29;Roggeveen;Quamar;Ap #863-2874 Nullam Road;27261;Castres;qquamar3783@google.fr;222647176;697434844;397510209
26 |25;75;Bunshoten;Quail;P.O. Box 210, 5613 Elit Avenue;40287;Saint-Dizier;bunshoten.quail@google.fr;481323403;679513535;325729605

```

Extrait de la Table Adhérent

```

fiche.csv X
CSV > fiche.csv
1 |numero_releve;num_adherent;numero_departement;libelle_lieu_releve;longitude_lieu_releve;latitude_lieu_releve;date_observation
2 |1;45;54;Besancon;-79.1577614336;-44.056780288;Mar 28, 2022
3 |2;63;78;Alès;-0.492552704;-83.6077466624;Feb 10, 2022
4 |3;38;66;Dreux;125.1916138496;25.2287604736;Mar 2, 2022
5 |4;58;26;Champigny-sur-Marne;123.3941938176;20.9003884544;Jan 9, 2022
6 |5;55;55;Bastia;152.5821750272;88.5206329344;Mar 23, 2022
7 |6;12;54;Saint-Lô;64.41397504;-57.0550718464;Jan 31, 2022
8 |7;49;29;Aix-en-Provence;37.4651577344;-46.3175097344;Jan 11, 2022
9 |8;44;95;Narbonne;-145.0754408448;89.0810738688;Feb 22, 2022
10 |9;32;21;Poitiers;66.0831883264;68.2467733504;Feb 18, 2022
11 |10;65;82;Pau;-161.9690287104;81.6377494528;Jan 3, 2022
12 |11;75;54;Châtellerauld;176.8682848256;-61.405658112;Jan 5, 2022
13 |12;63;48;Quimper;152.012076032;23.8864661504;Jan 11, 2022
14 |13;50;16;Tournefeuille;122.3854590976;-53.6392267776;Feb 16, 2022
15 |14;79;22;Vichy;144.217785344;54.1154927616;Mar 5, 2022
16 |15;79;46;Ajaccio;110.1774645248;-43.3418738688;Feb 10, 2022
17 |16;62;39;Herouville-Saint-Clair;-18.8014056448;-16.9619171328;Feb 5, 2022
18 |17;68;34;Saint-Dizier;-13.4854651904;31.4287079424;Feb 6, 2022
19 |18;9;68;Périgueux;-72.2223695872;2.8456402944;Mar 18, 2022
20 |19;49;56;Cherbourg-Océville;7.506447872;-12.062542384;Mar 16, 2022
21 |20;55;72;Vichy;173.1640282112;46.9689197568;Mar 9, 2022
22 |21;32;51;Haguenau;123.224213504;-86.878746112;Mar 9, 2022
23 |22;38;45;Narbonne;142.4259798016;27.4696005632;Feb 10, 2022
24 |23;33;37;Reims;139.2540570624;74.2717155328;Feb 28, 2022
25 |24;28;39;Limoges;-92.8972945408;-15.49811305344;Feb 18, 2022
26 |25;61;29;Rogles;-56.107516928;64.6307064832;Feb 12, 2022

```

Extrait de la Table Fiche

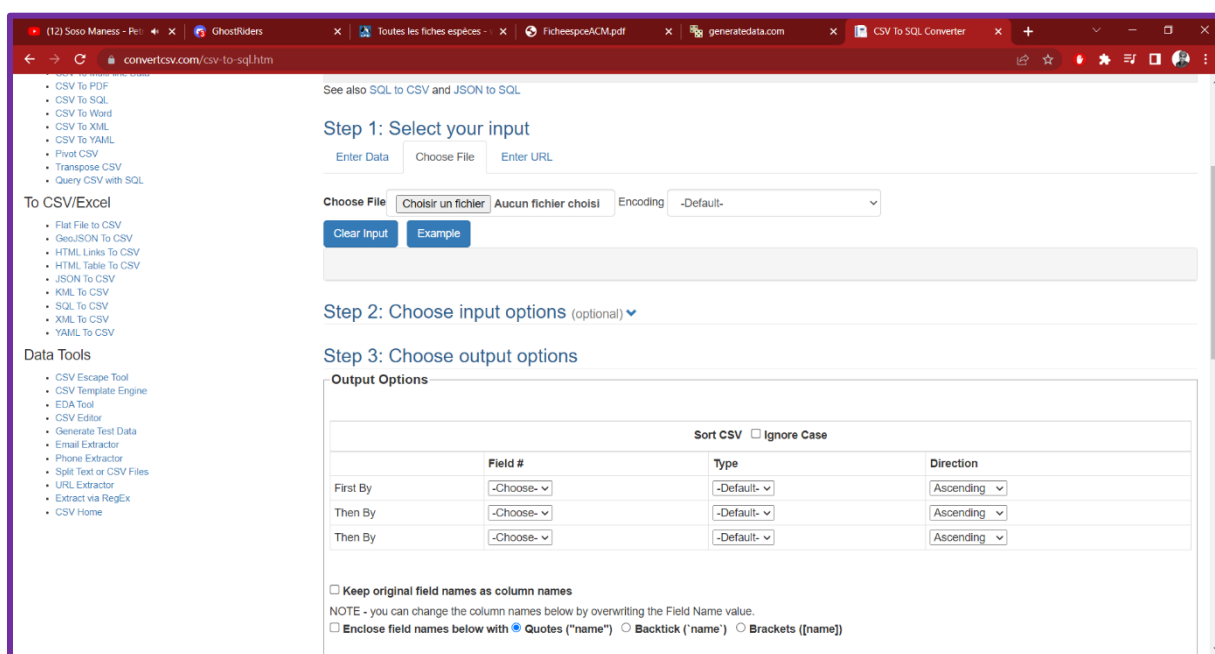
La table concerne est l'association des deux dernières. Elle est donc le résultat d'un produit cartésien. En effet nous avons 12 espèces et pour chaque espèce nous avons 500 fiches soit 6000 données dans la table concerne. Le nombre de spécimen a été cependant généré de manière aléatoire suivant une loi normale.

II. INSERTION DES DONNEES

Lors de la précédente partie nous avons pu générer le script SQL qui assurait la création de la base de données. Cependant cette table est vide. Notre objectif est donc d'insérer les données contenues dans les fichiers csv dans la base de données. Pour cela il existe de manières d'importer dans les logiciels comme PgAdmin. Néanmoins pas dans forcément dans tous, de plus en cas de perte de données il est important d'avoir une sauvegarde et de réeffectuer la manœuvre.

La solution pour ce problème est de créer des requêtes SQL d'insertion et d'en faire un fichier. Pour cela nous avons utilisés un convertisseur en ligne qui permettait de passer un fichier de .csv à .sql.

Voici le lien : <https://www.convertcsv.com/csv-to-sql.htm>



Aperçu du convertisseur en ligne

Une fois tous les fichiers convertis, nous avons créer un fichier ultime. Ce dernier à deux parties :

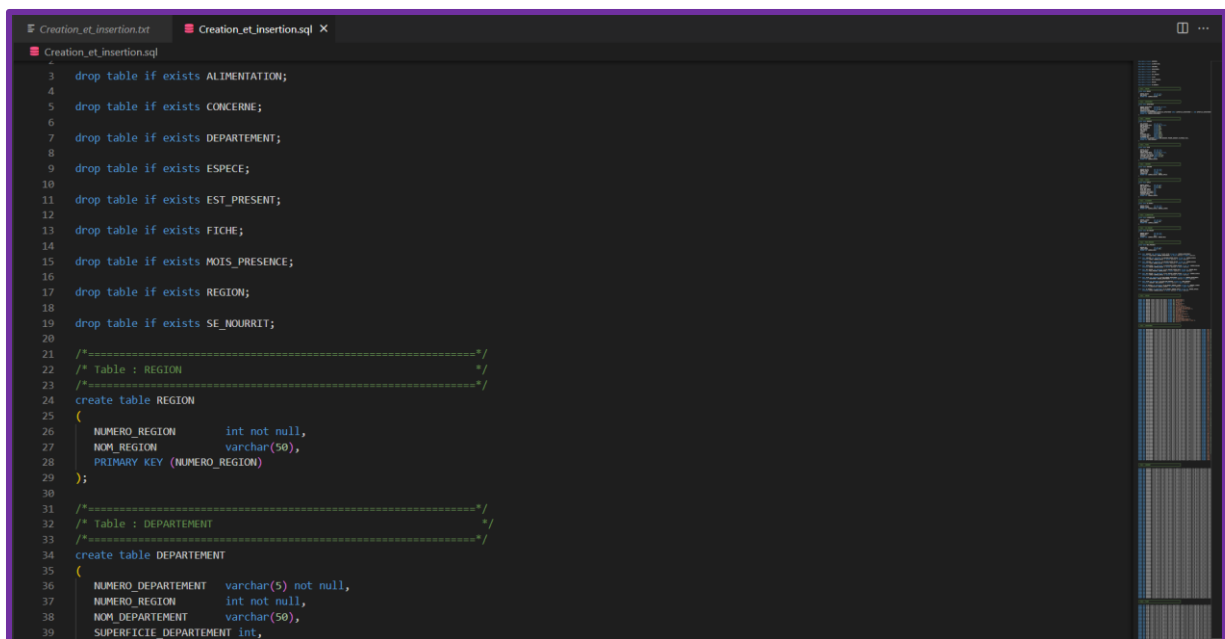
- Création de la base de données
- Insertion des données

Ainsi il suffira d'importer le fichier .sql lors de la création de la base ou de copier-coller lors d'un query tool.

Il est également important de préciser que l'insertion des données dans les tables n'ont pas été effectuée de manière aléatoire. En effet nous avons d'abord inséré les données dans les tables les plus indépendantes qui n'ont pas besoin d'autres tables. On peut notamment citer la table Région qui fût la première table où les données ont été insérés. A contrario on peut voir que la table concerne dépend de deux autres et ne peut pas être contenir des données avant les deux autres (Fiche, Espèce) car elle a besoin des clés primaires des deux autres.

Voici l'ordre des tables dans lesquelles nous avons inséré les données :

1. Région
2. Département
3. Adhérent
4. Fiche
5. Espèce
6. Concerne
7. Alimentation
8. Se Nourrit
9. Mois Présence
10. Est Présent



```
Creation_et_insertion.txt  Creation_et_insertion.sql X
Creation_et_insertion.sql
3 drop table if exists ALIMENTATION;
4
5 drop table if exists CONCERNE;
6
7 drop table if exists DEPARTEMENT;
8
9 drop table if exists ESPECE;
10
11 drop table if exists EST_PRESENT;
12
13 drop table if exists FICHE;
14
15 drop table if exists MOIS_PRESENCE;
16
17 drop table if exists REGION;
18
19 drop table if exists SE_NOURRIT;
20
21 /*-----*/
22 /* Table : REGION */
23 /*-----*/
24 create table REGION
25 (
26     NUMERO_REGION    int not null,
27     NOM_REGION       varchar(50),
28     PRIMARY KEY (NUMERO_REGION)
29 );
30
31 /*-----*/
32 /* Table : DEPARTEMENT */
33 /*-----*/
34 create table DEPARTEMENT
35 (
36     NUMERO_DEPARTEMENT  varchar(5) not null,
37     NUMERO_REGION       int not null,
38     NOM_DEPARTEMENT     varchar(50),
39     SUPERFICIE_DEPARTEMENT int,
```

Extrait de la partie création

III. REQUETES

Pour cette partie nous allons effectuer les requêtes demandées. Nous allons pour chaque requêtes afficher la demande, la requête réalisé ainsi que la sortie obtenu (output).

Question 1 :

Les adhérents qui n'ont pas rempli les nombres d'observations (nombre non rempli) dans les fiches pour une espèce donnée (avec le nom d'adhèrent et le nom de l'espèce).

```
SELECT a.nom_adherent, e.nom_espece
FROM adherent a
  JOIN fiche f ON f.num_adherent = a.num_adherent
  JOIN concerne c ON c.numero_releve = f.numero_releve
  JOIN espece e ON e.numero_espece = c.numero_espece
WHERE nbr_specimen IS NULL OR nbr_specimen = 0
```

	nom_adherent character varying (50)	nom_espece character varying (50)
1	Janvier	Prunella Modularis
2	Van Alphen	Prunella Modularis
3	Tailler	Prunella Modularis
4	Chaput	Prunella Modularis
5	Plourde	Prunella Modularis
6	Beaulieu	Prunella Modularis
7	Peerenboom	Prunella Modularis
8	Lachapelle	Prunella Modularis
9	Peerenboom	Prunella Modularis
10	Lane	Prunella Modularis
11	Brisbois	Prunella Modularis
12	Chastain	Prunella Modularis
13	Monet	Motacilla alba
14	Poulin	Motacilla alba
15	Travers	Motacilla alba
16	Chastain	Motacilla alba
17	Van Assen	Pyrrhula pyrrhula

31 lignes au total

Question 2 :

Afficher pour chaque espèce d'oiseaux les aliments qui ne sont pas consommés par cette espèce. Vous afficherez le nom de l'espèce et le nom de l'aliment.

```
SELECT e.numero_espece, e.nom_espece, a.numero_aliment, a.nom_aliment
FROM espece e
CROSS JOIN alimentation a
WHERE (e.numero_espece, a.numero_aliment) NOT IN
(
  SELECT e1.numero_espece, a1.numero_aliment
  FROM espece e1
  JOIN SE_NOURRIT sn1 ON sn1.numero_espece = e1.numero_espece
  JOIN alimentation a1 ON a1.numero_aliment = sn1.numero_aliment
)
ORDER BY 1,2
```

	numero_espece integer	nom_espece character varying (50)	numero_aliment integer	nom_aliment character varying (20)
1	1	Prunella Modularis	2	Fruits
2	1	Prunella Modularis	3	Bourgeons
3	1	Prunella Modularis	5	Vers
4	1	Prunella Modularis	6	Galnds
5	1	Prunella Modularis	7	Oeufs
6	1	Prunella Modularis	8	Oisillions
7	2	Motacilla alba	2	Fruits
8	2	Motacilla alba	3	Bourgeons
9	2	Motacilla alba	5	Vers
10	2	Motacilla alba	6	Galnds
11	2	Motacilla alba	7	Oeufs
12	2	Motacilla alba	8	Oisillions
13	2	Motacilla alba	9	Araignees
14	3	Pyrrhula pyrrhula	5	Vers
15	3	Pyrrhula pyrrhula	6	Galnds
16	3	Pyrrhula pyrrhula	7	Oeufs
17	3	Pyrrhula pyrrhula	8	Oisillions

71 lignes au total

Question 3 :

Le nombre moyen (moyenne), le nombre minimal, le nombre maximal et le nombre total d'oiseaux observés sur l'ensemble des observations pour une espèce donnée (vous n'afficherez pas l'espèce).

```
SELECT ROUND(AVG(c.nbr_specimen),2) AS "Moyenne", MIN(c.nbr_specimen) AS "MIN",
      MAX (c.nbr_specimen) AS "MAX", COUNT(c.nbr_specimen) AS "TOTAL"
FROM CONCERNE c
      JOIN ESPECE e ON e.numero_espece = c.numero_espece
WHERE e.nom_espece = 'Accipiter nisus'
```

	Moyenne numeric	MIN integer	MAX integer	TOTAL bigint
1	15.20	1	40	498

Valeurs pour l'espèce 'Accipiter nisus'

Question 4 :

Le nombre moyen, le nombre minimal, le nombre maximal et l'écart type des oiseaux observés (nombre) par espèce pour un mois donné. Vous afficherez l'espèce.

```
SELECT e.nom_espece, DATE_PART('month', f.date_observation) AS "Num_Mois" ,
      AVG(c.nbr_specimen) AS "Moyenne", MIN(c.nbr_specimen) AS "MIN",
      MAX (c.nbr_specimen) AS "MAX", SQRT(VARIANCE(nbr_specimen)) AS "Ecart-
type"
FROM CONCERNE c
      JOIN ESPECE e ON e.numero_espece = c.numero_espece
      JOIN FICHE f ON f.numero_releve = c.numero_releve
WHERE e.nom_espece = 'Accipiter nisus'
      AND DATE_PART('month', f.date_observation) = 1
GROUP BY e.nom_espece, DATE_PART('month', f.date_observation)
```

	nom_espece character varying (50)	Num_Mois double precision	Moyenne numeric	MIN integer	MAX integer	Ecart-type numeric
1	Accipiter nisus	1	14.9440993788819876	1	34	6.1808660873749814

Valeurs pour l'espèce 'Accipiter nisus' pour le mois de janvier

Question 5 :

L'adhérent qui a observé le moins d'oiseaux d'une espèce donnée lors d'une observation. Vous afficherez le nom de l'adhérent, le nombre d'oiseaux observés et l'espèce.

```
SELECT ad.nom_adherent, e.nom_espece, COUNT(c.nbr_specimen)
FROM ADHERENT ad
  JOIN FICHE f ON f.num_adherent = ad.num_adherent
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
WHERE e.nom_espece = 'Accipiter nisus'
GROUP BY ad.nom_adherent, e.nom_espece
HAVING COUNT(c.nbr_specimen) <= ALL
(
  SELECT COUNT(nbr_specimen)
  FROM CONCERNE c1
  JOIN ESPECE e1 ON e1.numero_espece = c1.numero_espece
  WHERE e1.nom_espece = e.nom_espece
)
```

	nom_adherent character varying (50)	nom_espece character varying (50)	count bigint
1	Aaldenberg	Accipiter nisus	5
2	Aarle	Accipiter nisus	8
3	Achterberg	Accipiter nisus	6
4	Achthoven	Accipiter nisus	15
5	Adrichem	Accipiter nisus	11
6	Beauchene	Accipiter nisus	24
7	Beaulieu	Accipiter nisus	10
8	Bezuindenhout	Accipiter nisus	15
9	Bouwmeester	Accipiter nisus	4
10	Brisbois	Accipiter nisus	27
11	Bunschoten	Accipiter nisus	5
12	Chaput	Accipiter nisus	10
13	Chastain	Accipiter nisus	5
14	Cloutier	Accipiter nisus	5
15	Cruyssen	Accipiter nisus	6
16	Dubois	Accipiter nisus	8
17	Duval	Accipiter nisus	12

Valeurs pour l'espèce 'Accipiter nisus'

56 valeurs au total

Question 6 :

L'adhérent qui a compté le nombre total d'oiseaux le plus faible pour une espèce donnée. Vous afficherez le nom de l'adhérent et le calcul.

```
DROP VIEW IF EXISTS v_oiseau_faible CASCADE;

CREATE VIEW v_oiseau_faible AS
  SELECT ad.nom_adherent AS NomAdherent, e.nom_espece AS NomEspece,
         COUNT(c.nbr_specimen) AS NbOiseau
  FROM ADHERENT ad
       JOIN FICHE f ON f.num_adherent = ad.num_adherent
       JOIN CONCERNE c ON c.numero_releve = f.numero_releve
       JOIN ESPECE e ON e.numero_espece = c.numero_espece
  WHERE e.nom_espece = 'Accipiter nisus'
  GROUP BY ad.nom_adherent, e.nom_espece
  HAVING COUNT(c.nbr_specimen) <= ALL
    (
      SELECT COUNT(nbr_specimen)
      FROM CONCERNE c1
      JOIN ESPECE e1 ON e1.numero_espece = c1.numero_espece
      WHERE e1.nom_espece = e.nom_espece
    )
```

Puis :

```
SELECT *
FROM v_oiseau_faible
WHERE NbOiseau <= ALL (
  SELECT NbOiseau
  FROM v_oiseau_faible
)
```

	nomadherent character varying (50) 🔒	nomespece character varying (50) 🔒	nboiseau bigint 🔒
1	Garcon	Accipiter nisus	2
2	Langlais	Accipiter nisus	2
3	Lemaire	Accipiter nisus	2

Valeurs pour l'espèce 'Accipiter nisus'

Question 7 :

L'adhérent qui a observé le plus d'oiseaux d'une espèce donnée lors d'une observation. Vous afficherez le nom de l'adhérent, le nombre d'oiseaux observés et l'espèce.

```
SELECT ad.nom_adherent, e.nom_espece, SUM(c.nbr_specimen)
FROM ADHERENT ad
  JOIN FICHE f ON f.num_adherent = ad.num_adherent
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
WHERE e.nom_espece = 'Accipiter nisus'
GROUP BY ad.num_adherent, e.numero_espece
HAVING SUM(c.nbr_specimen) >= ALL
(
  SELECT SUM(nbr_specimen)
  FROM CONCERNE c1
  JOIN ESPECE e1 ON e1.numero_espece = c1.numero_espece
  JOIN FICHE f1 ON f1.numero_releve = c1.numero_releve
  WHERE e1.nom_espece = 'Accipiter nisus'
  GROUP BY f1.num_adherent
  ORDER BY 1
)
```

	nom_adherent character varying (50)	nom_espece character varying (50)	sum bigint
1	Poulin	Accipiter nisus	224

Valeurs pour l'espèce 'Accipiter nisus'

Question 8 :

L'adhérent qui a compté le nombre total d'oiseaux le plus élevé pour une espèce donnée. Vous afficherez le nom de l'adhérent et le calcul.

```
SELECT ad.nom_adherent, e.nom_espece, COUNT(c.nbr_specimen)
FROM ADHERENT ad
  JOIN FICHE f ON f.num_adherent = ad.num_adherent
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
WHERE e.nom_espece = 'Accipiter nisus'
GROUP BY ad.nom_adherent, e.nom_espece
HAVING COUNT(c.nbr_specimen) >= ALL
(
  SELECT COUNT(c1.nbr_specimen)
  FROM ADHERENT ad1
    JOIN FICHE f1 ON f1.num_adherent = ad1.num_adherent
    JOIN CONCERNE c1 ON c1.numero_releve = f1.numero_releve
    JOIN ESPECE e1 ON e1.numero_espece = c1.numero_espece
  WHERE e1.nom_espece = 'Accipiter nisus'
  GROUP BY ad1.nom_adherent, e1.nom_espece
  ORDER BY 1 desc
)
```

	nom_adherent character varying (50) 🔒	nom_espece character varying (50) 🔒	count bigint 🔒
1	Janvier	Accipiter nisus	28

Valeurs pour l'espèce 'Accipiter nisus'

Question 9 :

Nom et prénom des adhérents qui ont saisi des fiches d'observation le... (date) pour une espèce donnée.

```
SELECT ad.nom_adherent, ad.prenom_adherent, e.nom_espece
FROM ADHERENT ad
  JOIN FICHE f ON f.num_adherent = ad.num_adherent
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
WHERE f.date_observation = '2022-01-01'
  AND e.nom_espece = 'Accipiter nisus'
GROUP BY ad.nom_adherent, ad.prenom_adherent, e.nom_espece
```

	nom_adherent character varying (50)	prenom_adherent character varying (50)	nom_espece character varying (50)
1	Aaldenberg	Xenos	Accipiter nisus
2	Peerenboom	Adara	Accipiter nisus
3	Travers	Abel	Accipiter nisus
4	Van Aalsburg	Leonard	Accipiter nisus

Valeurs pour l'espèce 'Accipiter nisus'

Question 10 :

Nombre d'adhérents qui ont saisi des fiches d'observation le... (date) pour une espèce donnée.

```
SELECT COUNT(ad.num_adherent)
FROM ADHERENT ad
  JOIN FICHE f ON f.num_adherent = ad.num_adherent
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
WHERE f.date_observation = '2022-01-01'
  AND e.nom_espece = 'Accipiter nisus'
```

	count bigint
1	4

Valeurs pour l'espèce 'Accipiter nisus'

Question 11 :

Les départements dans lesquels on a compté moins de 10 oiseaux (au total) d'une espèce donnée. Ce seuil est indicatif, vous pourrez le modifier. Vous afficherez le nom du département et le nombre d'oiseaux.

```
SELECT d.nom_departement, COUNT(e.nom_espece)
FROM DEPARTEMENT d
  JOIN FICHE f ON f.numero_departement = d.numero_departement
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
GROUP BY (d.nom_departement)
HAVING COUNT (e.nom_espece) <= 15
```

	nom_departement character varying (50)	count bigint
1	Haute-Saone	12
2	Correze	12
3	Corse-du-Sud	12

Seuil modifié, 15 au lieu de 10 oiseaux

Question 12 :

Les noms des régions dans lesquels on a compté moins de 5000 oiseaux (compris) en affichant une information supplémentaire : 'moins de 3000' ou 'de 3000 à 5000'. Vous pouvez choisir d'autres bornes en fonction de votre jeu de données. Vous afficherez également le calcul.

```
SELECT r.nom_region, COUNT(c.nbr_specimen),
CASE
  WHEN COUNT(c.nbr_specimen) < 400 THEN 'moins de 400'
  WHEN COUNT(c.nbr_specimen) >= 400 THEN 'de 400 à 600'
END as Commentaire
FROM REGION r
  JOIN DEPARTEMENT d ON d.numero_region = r.numero_region
  JOIN FICHE f ON f.numero_departement = d.numero_departement
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
GROUP BY r.nom_region
HAVING COUNT (c.nbr_specimen) <= 600
ORDER BY 2 desc
```

	nom_region character varying (50)	count bigint	commentaire text
1	Auvergne-Rhône-Alpes	563	de 400 à 600
2	Bourgogne-Franche-Comté	559	de 400 à 600
3	Centre-Val de Loire	442	de 400 à 600
4	Bretagne	393	moins de 400
5	Normandie	382	moins de 400
6	Hauts-de-France	263	moins de 400
7	Pays de la Loire	262	moins de 400
8	Provence-Alpes-Côte d'Azur	240	moins de 400
9	Corse	36	moins de 400

Seuil modifié 400 au lieu de 3000, 600 au lieu de 5000

Question 13 :

Le nombre de régions dans lesquelles on a compté moins de 5000 oiseaux (compris) en fonction des bornes 'moins de 3000' et 'de 3000 à 5000'. Vous pouvez choisir d'autres bornes en fonction de votre jeu de données, mais utilisez les mêmes bornes que dans la question précédente.

```
DROP VIEW IF EXISTS v_seuil_oiseau CASCADE;

CREATE VIEW v_seuil_oiseau AS
  SELECT r.nom_region AS nomRegion, COUNT(c.nbr_specimen) AS NbOiseau,
  CASE
    WHEN COUNT(c.nbr_specimen) < 400 THEN 'moins de 400'
    WHEN COUNT(c.nbr_specimen) >= 400 THEN 'de 400 à 600'
  END as Commentaire
  FROM REGION r
    JOIN DEPARTEMENT d ON d.numero_region = r.numero_region
    JOIN FICHE f ON f.numero_departement = d.numero_departement
    JOIN CONCERNE c ON c.numero_releve = f.numero_releve
    JOIN ESPECE e ON e.numero_espece = c.numero_espece
  GROUP BY r.nom_region
  HAVING COUNT (c.nbr_specimen) <= 600
```

Puis :

```
SELECT Commentaire, COUNT(*)
FROM v_seuil_oiseau
GROUP BY Commentaire
```

	commentaire text	count bigint
1	de 400 à 600	3
2	moins de 400	6

Question 14 :

Le classement des départements en fonction du nombre total d'oiseaux observés d'une espèce donnée, trié par ordre décroissant. Vous afficherez le nom du département, le calcul et le classement.

```
SELECT d.nom_departement, COUNT(c.nbr_specimen),
       DENSE_RANK () OVER (ORDER BY COUNT(c.nbr_specimen) desc)
FROM DEPARTEMENT d
     JOIN FICHE f ON f.numero_departement = d.numero_departement
     JOIN CONCERNE c ON c.numero_releve = f.numero_releve
     JOIN ESPECE e ON e.numero_espece = c.numero_espece
GROUP BY d.nom_departement
```

	nom_departement character varying (50)	count bigint	dense_rank bigint
1	Ille-et-Vilaine	167	1
2	Eure	132	2
3	Finistere	130	3
4	Saone-et-Loire	120	4
5	Tarn-et-Garonne;	119	5
6	Gironde	108	6
7	Moselle	108	6
8	Cher	108	6
9	Marne	108	6
10	Meuse	108	6
11	Vaucluse	108	6
12	Charente	107	7
13	Eure-et-Loir	107	7
14	Pas-de-Calais	107	7
15	Hautes-Pyrenees	107	7
16	Seine-Saint-Denis	106	8
17	Bas-Rhin	96	9

86 lignes au total

Question 15 :

Le classement des régions en fonction du nombre total d'oiseaux observés trié par ordre décroissant. Vous afficherez le nom de la région, le calcul et le classement.

```
SELECT r.nom_region, COUNT(c.nbr_specimen),
       DENSE_RANK () OVER (ORDER BY COUNT(c.nbr_specimen) desc)
FROM REGION r
      JOIN DEPARTEMENT d ON d.numero_region = r.numero_region
      JOIN FICHE f ON f.numero_departement = d.numero_departement
      JOIN CONCERNE c ON c.numero_releve = f.numero_releve
      JOIN ESPECE e ON e.numero_espece = c.numero_espece
GROUP BY r.nom_region
```

	nom_region character varying (50)	count bigint	dense_rank bigint
1	Occitanie	752	1
2	Nouvelle-Aquitaine	740	2
3	Grand Est	719	3
4	Île-de-France	618	4
5	Auvergne-Rhône-Alpes	563	5
6	Bourgogne-Franche-Comté	559	6
7	Centre-Val de Loire	442	7
8	Bretagne	393	8
9	Normandie	382	9
10	Hauts-de-France	263	10
11	Pays de la Loire	262	11
12	Provence-Alpes-Côte d'Azur	240	12
13	Corse	36	13

Question 16 :

Le nombre de fiches réalisées par espèces. En fonction de ce nombre, afficher un message qui indique si ce nombre est important ou non :

Si ce nombre est <10, afficher 'Peu de comptages'.

Si compris entre 10 et 40, 'Nombre correct de comptages'.

Sinon afficher 'Beaucoup de comptages'.

Vous pouvez choisir d'autres bornes en fonction de votre jeu de données. Vous afficherez également l'espèce et le calcul.

```
SELECT e.numero_espece, e.nom_espece, COUNT(c.nbr_specimen),
CASE
  WHEN COUNT(c.nbr_specimen) < 10 THEN 'Peu de comptages'
  WHEN COUNT(c.nbr_specimen) < 40 THEN 'Nombre correct de comptages'
  WHEN COUNT(c.nbr_specimen) >= 40 THEN 'Beaucoup de comptages'
END as Commentaire
FROM FICHE f
  JOIN CONCERNE c ON c.numero_releve = f.numero_releve
  JOIN ESPECE e ON e.numero_espece = c.numero_espece
GROUP BY e.numero_espece, e.nom_espece
ORDER BY 1, 2
```

	numero_espece [PK] integer	nom_espece character varying (50)	count bigint	commentaire text
1	1	Prunella Modularis	488	Beaucoup de comptages
2	2	Motacilla alba	496	Beaucoup de comptages
3	3	Pyrrhula pyrrhula	499	Beaucoup de comptages
4	4	Carduelis carduelis	499	Beaucoup de comptages
5	5	Corvus corone	496	Beaucoup de comptages
6	6	Accipiter nisus	498	Beaucoup de comptages
7	7	Sturnus vulgaris	499	Beaucoup de comptages
8	8	Falco tinnunculus	498	Beaucoup de comptages
9	9	Parus major	500	Beaucoup de comptages
10	10	Lophophanes cristatus	500	Beaucoup de comptages
11	11	Periparus ater	499	Beaucoup de comptages
12	12	Fringilla montifringilla	497	Beaucoup de comptages

Question 17 :

Affichez les noms des départements qui pour une espèce donnée ont une densité d'oiseaux supérieure à un seuil. La densité est calculée en totalisant l'ensemble des oiseaux observés sur un département par rapport à sa superficie. Attention : la division d'un entier par un entier renvoie un entier dans PostgreSQL, il faudra donc caster l'un de ces champs ou les 2 en `float`. Afficher le nom du département et la densité.

```
DROP VIEW IF EXISTS v_departement_nbSpecimen CASCADE;

CREATE VIEW v_departement_nbSpecimen AS
  SELECT d.nom_departement as nomDepartement,
         d.superficie_departement AS superficeDepartement,
         e.nom_espece as nomEspece,
         COUNT(c.nbr_specimen) AS nbSpecimen
  FROM departement d
       JOIN FICHE f ON f.numero_departement = d.numero_departement
       JOIN CONCERNE c ON c.numero_releve = f.numero_releve
       JOIN ESPECE e ON e.numero_espece = c.numero_espece
  WHERE e.nom_espece = 'Accipiter nisus'
  GROUP BY d.nom_departement, d.superficie_departement, e.nom_espece
```

Puis :

```
SELECT v.nomDepartement
FROM v_departement_nbSpecimen v
WHERE CAST(v.nbSpecimen as float) / v.superficeDepartement <= 0.0005
```

	nomdepartement character varying (50)
1	Aube
2	Aude
3	Aveyron
4	Correze
5	Corse-du-Sud
6	Dordogne
7	Drôme
8	Haute-Corse
9	Haute-Saone
10	Haute-Vienne
11	Morbihan
12	Nievre
13	Vosges

Question 18 :

Le nombre de départements dont la densité d'oiseaux pour une espèce donnée est supérieure à un seuil (même seuil que dans la question précédente).

P.S : On utilisera donc la même vue créer lors de la question précédente

```
SELECT v.nomDepartement
FROM v_departement_nbSpecimen v
WHERE CAST(v.nbSpecimen as float) / v.superficieDepartement >= 0.0005
```

	nomdepartement character varying (50)
1	Hauts-de-Seine
2	Paris
3	Seine-Saint-Denis
4	Tarn-et-Garonne;
5	Territoire de Belfort
6	Val-de-Marne

Question 19 :

Quels sont les 3 départements ayant la plus grande densité d'oiseaux ? Afficher le nom du département et la densité.

```
DROP VIEW IF EXISTS v_departement_densite CASCADE;

CREATE VIEW v_departement_densite AS
  SELECT v.nomDepartement AS nomDepartement ,
         (v.nbSpecimen / CAST (v.superficieDepartement as float)) AS densite
  FROM v_departement_nbSpecimen v
```

Puis :


```

DROP VIEW IF EXISTS v_top3_densite CASCADE;

CREATE VIEW v_top3_densite ("Nom Departement", "Densite", "Classement") AS
  WITH t AS
    (
      SELECT v.*, RANK () OVER (ORDER BY v.densite desc) AS classement
      FROM v_departement_densite v
    )
  SELECT t.*
  FROM t
  WHERE t.classement <= 3

```

Enfin :

```
SELECT * FROM v_top3_densite;
```

	Nom Departement character varying (50)	Densite double precision	Classement bigint
1	Paris	0.06666666666666667	1
2	Hauts-de-Seine	0.03977272727272727	2
3	Seine-Saint-Denis	0.038135593220338986	3

Question 20 :

Les mois (noms) au cours desquels toutes les espèces ont été présentes.

```

SELECT mp.libelle_mois
FROM MOIS_PRESENCE mp
WHERE NOT EXISTS
  (
    SELECT 'X'
    FROM ESPECE e
    WHERE NOT EXISTS
      (
        SELECT 'X'
        FROM EST_PRESENT ep
        WHERE ep.numero_espece = e.numero_espece
              AND ep.numero_mois = mp.numero_mois
              AND ep.presence = 'True'
      )
  )

```

	libelle_mois character varying (20) 
1	janvier
2	fevrier
3	mars
4	octobre
5	novembre
6	décembre