

CLIENTS			
<u>CODE_CLIENT</u>	CHAR(5)	<u><pk></u>	not null
SOCIETE	NVARCHAR2(40)		not null
TELEPHONE	VARCHAR2(24)		not null

CODE_CLIENT = CODE_CLIENT

ADRESSES									
NO_ADRESSE	NUMBER(8)	<u><pk></u>		not null					
TYPE_ADRESSE	NUMBER(1)		<u><fk></u>	not null					
CODE_CLIENT	CHAR(5)			not null					
ADRESSE	NVARCHAR2(80)			not null					
VILLE	NVARCHAR2(80)			not null					
CODE_POSTAL	NVARCHAR2(20)			not null					
PROVINCE	NVARCHAR2(40)			not null					
PAYS	NVARCHAR2(60)			not null					

NO_ADRESSE = NO_ADRESSE

ACHETEURS					
NO_ACHETEUR	NUMBER(6)	<u><pk></u>		not null	
NO_ADRESSE	NUMBER(8)		<u><fk></u>	not null	
NOM	NVARCHAR2(60)			not null	
PRENOM	NVARCHAR2(60)			not null	
TITRE	VARCHAR2(5)			not null	
MAIL	NVARCHAR2(90)			not null	

NO_ACHETEUR = NO_ACHETEUR

COMMANDES									
NO_COMMANDE	NUMBER(6)			not null					
NO_VENDEUR	NUMBER(6)		<u><fk1></u>	not null					
NO_ACHETEUR	NUMBER(6)		<u><fk2></u>	not null					
DATE_COMMANDE	DATE			not null					
DATE_ENVOI	DATE			not null					
LIVREE	NUMBER(1)			not null					
ACQUITEE	NUMBER(1)			not null					
ANNULEE	NUMBER(1)			not null					
ANNEE	NUMBER(1)			not null					
TRIMESTRE	NUMBER(1)			not null					
MOIS	NUMBER(1)			not null					
EXTRACT(YEAR FROM DATE_COMMANDE)									
TO_NUMBER(TO_CHAR(DATE_COMMANDE, 'Q'))									
EXTRACT(MONTH FROM DATE_COMMANDE)									

NO_COMMANDE = NO_COMMANDE

DETAILS_COMMANDES							
NO_COMMANDE	NUMBER(6)	<u><pk fk1></u>		not null			
REF_PRODUIT	NUMBER(6)		<u><pk fk2></u>	not null			
PRIX_UNITAIRE	NUMBER(8,2)			not null			
QUANTITE	NUMBER(5)			not null			
PORT	NUMBER(8,2)			not null			
REMISE	NUMBER(8,2)			not null			
RETOURNE	NUMBER(1)			not null			
ECHANGE	NUMBER(1)			not null			

REF_PRODUIT = REF_PRODUIT

CATEGORIES			
<u>CODE_CATEGORIE</u>	NUMBER(6)	<u><pk></u>	not null
NOM_CATEGORIE	VARCHAR2(25)		not null
DESCRIPTION	VARCHAR2(100)		not null

CODE_CATEGORIE = CODE_CATEGORIE

PRODUITS						
REF_PRODUIT	NUMBER(6)	<u><pk></u>		not null		
NOM_PRODUIT	NVARCHAR2(50)			not null		
NO_FOURNISSEUR	NUMBER(6)		<u><fk2></u>	not null		
CODE_CATEGORIE	NUMBER(6)		<u><fk1></u>	not null		
QUANTITE	VARCHAR2(30)			not null		
PRIX_UNITAIRE	NUMBER(8,2)			not null		
UNTES_STOCK	NUMBER(5)			not null		
INDISPONIBLE	NUMBER(1)			not null		

NO_FOURNISSEUR = NO_FOURNISSEUR

FOURNISSEURS						
NO_FOURNISSEUR	NUMBER(6)	<u><pk></u>		not null		
SOCIETE	NVARCHAR2(40)			not null		
ADRESSE	NVARCHAR2(60)			not null		
VILLE	VARCHAR2(30)			not null		
CODE_POSTAL	VARCHAR2(10)			not null		
PAYS	VARCHAR2(15)			not null		
TELEPHONE	VARCHAR2(24)			not null		
FAX	VARCHAR2(24)			not null		

EMPLOYES									
NO_EMPLOYE	NUMBER(6)	<u><pk></u>		not null					
REND_COMPTTE	NUMBER(6)		<u><fk></u>	not null					
NOM	NVARCHAR2(40)			not null					
PRENOM	NVARCHAR2(30)			not null					
FUNCTION	VARCHAR2(30)			not null					
TITRE	VARCHAR2(5)			not null					
DATE_NAISSANCE	DATE			not null					
DATE_EMBAUACHE	DATE			not null					
SALAIRE	NUMBER(8,2)			not null					

NO_EMPLOYE = REND_COMPTTE

NO_EMPLOYE = NO_VENDEUR

VENDEURS				
NO_VENDEUR	NUMBER(6)	<u><pk fk></u>		not null
COMMISSION	NUMBER(8,2)			not null
PAYS	VARCHAR(20)			not null
REGION	VARCHAR(50)			not null

NO_VENDEUR = NO_VENDEUR

Créer les requêtes suivantes :

Exercice n° 01

Le nom du produit, le fournisseur, la catégorie et les quantités des produits pour les fournisseurs français ou des catégories 'Boissons' ou 'Desserts'.

Affichez les produits pour lesquelles les quantités sont données en « boîtes » ou « carton ».

Résultat de la requête :

NOM_PRODUIT	FOURNISSEUR	CATEGORIE	UNITES_STOCK	QUANTITE
¹ Chai	Exotic Liquids	Boissons	56243910	boîtes x 20 sacs
² Brownie Mix	Nouvelle-Orléans Caiun Delights	Desserts	536933	boîtes
³ Pavlova	Pavlova, Ltd.	Desserts	22321932	boîtes (500 g)
⁴ Chocolade Biscuits Mix	Pavlova, Ltd.	Desserts	1043310	boîtes de 12 pièces
⁵ Sir Rodney's Marmalade	Specialty Biscuits, Ltd.	Desserts	35910930	boîtes
⁶ Teatime Chocolate Biscuits	Specialty Biscuits, Ltd.	Desserts	29377210	boîtes x 12 pièces
⁷ Sir Rodney's Scones	Specialty Biscuits, Ltd.	Desserts	974824	cartons x 4 pièces
⁸ Coffee	Bigfoot Breweries	Boissons	3406316	boîtes de 500 g
⁹ Ipoh Coffee	Leka Trading	Boissons	1043316	boîtes (500 g)
¹⁰ Cake Mix	Leka Trading	Desserts	95634	boîtes
¹¹ Chocolade	Zaanse Snoepfabriek	Desserts	6135310	cartons
¹² Zaanse koeken	Zaanse Snoepfabriek	Desserts	14781510	boîtes (4 onces)
¹³ Maxilaku	Karkki Oy	Desserts	25266924	cartons (50 g)

```

SELECT
    NOM_PRODUIT,
    SOCIETE FOURNISSEUR,
    NOM_CATEGORIE CATEGORIE,
    UNITES_STOCK,
    QUANTITE
FROM PRODUITS
JOIN FOURNISSEURS USING (NO_FOURNISSEUR)
JOIN CATEGORIES USING (CODE_CATEGORIE)
WHERE (PAYS = 'FRANCE'
       OR NOM_CATEGORIE IN ('Boissons','Desserts'))
AND (QUANTITE LIKE '%boîtes%'
     OR QUANTITE LIKE '%carton%')
ORDER BY NO_FOURNISSEUR, CODE_CATEGORIE;

```

Exercice n° 02

Le pays, la société cliente, le nom, le prénom du acheteur, le nombre de commandes par année qui ont effectué plus de 23 commandes en 2019.

Résultat de la requête :

📌 PAYS	📌 SOCIETE	📌 VENDEUR	📌 ANNEE	📌 NB_COMMANDES
1 Brésil	Laughing Bacchus Wine Cellars	Mazza OTIS	2019	27
2 Brésil	Tortuga Restaurante	Resnick RICHMOND	2019	26
3 Canada	B's Beverages	Fepoi WORTH	2019	24
4 Canada	B's Beverages	Mcevov KAKIT	2019	25
5 Canada	Lehmanns Marktstand	Schoolev DAMIEN	2019	24
6 États-Unis	Centro comercial Moctezuma	Stiller COLMAN	2019	26
7 États-Unis	Laughing Bacchus Wine Cellars	East RACHEL	2019	28
8 France	Blondel père et fils	Tucker HEATHER	2019	26
9 France	Bólido Comidas preparadas	Tavener NELSON	2019	24
10 Italie	Around the Horn	Carr ARNOLD	2019	29
11 Italie	LILA-Supermercado	Oppv LINETTE	2019	24
12 Italie	Ottillies Käseladen	Calahan PENELOPE	2019	24
13 Italie	Ottillies Käseladen	Lvon LYLE	2019	27
14 Italie	The Big Cheese	Wolsev THALIA	2019	24
15 Pologne	B's Beverages	Mallonev ERROL	2019	25
16 Royaume-Uni	Reggiani Caseifici	Yarborough RAE	2019	26
17 Singapour	Laughing Bacchus Wine Cellars	Gerber CAREY	2019	24
18 Singapour	Laughing Bacchus Wine Cellars	Rivera DUKE	2019	24

```

SELECT AD.PAYS
      ,CL.SOCIETE
      ,AC.NOM||' '||UPPER(AC.PRENOM) VENDEUR
      ,EXTRACT(YEAR FROM CO.DATE_COMMANDE) ANNEE
      ,COUNT(DISTINCT CO.NO_COMMANDE) NB_COMMANDES
FROM ADRESSES AD
      JOIN ACHETEURS AC ON AC.NO_ADRESSE = AD.NO_ADRESSE
      JOIN COMMANDES CO ON CO.NO_ACHETEUR = AC.NO_ACHETEUR
      JOIN CLIENTS CL ON AD.CODE_CLIENT = CL.CODE_CLIENT
WHERE EXTRACT(YEAR FROM CO.DATE_COMMANDE) = 2019
GROUP BY AD.PAYS
      ,CL.SOCIETE
      ,AC.NOM
      , AC.PRENOM
      , EXTRACT(YEAR FROM CO.DATE_COMMANDE)
HAVING COUNT(DISTINCT CO.NO_COMMANDE) > 23
ORDER BY AD.PAYS
      ,CL.SOCIETE
      ,AC.NOM;

```

Exercice n° 03

La somme des frais de port saisis par vendeur, et par pays du vendeur.

Affichez uniquement les enregistrements du mois de mai de l'année 2019 qui ont une somme de frais de port supérieure à 80000€.

Résultat de la requête :

	PAYS	VENDEUR	SUM_PORT
1	Brésil	Jenny Michel	89522,3
2	Brésil	Poidatz Benoît	153635,4
3	Brésil	Virv Yvan	101582,7
4	États-Unis	Coutou Myriam	138483,6
5	États-Unis	Silberreiss Albert	115763,6
6	États-Unis	Urbanjak Isabelle	104387,4
7	France	Frederic Jean Marie	98648,1
8	France	Regner Charles	86949,6
9	Royaume-Uni	Perret Laurence	85541,9

```

SELECT VN.pays
      ,EM.NOM||' '||EM.PRENOM VENDEUR
      ,SUM(DC.PORT) SUM_PORT
FROM EMPLOYES EM
      JOIN VENDEURS VN ON VN.NO_VENDEUR = EM.NO_EMPLOYE
      JOIN COMMANDES CO ON CO.NO_VENDEUR = VN.NO_VENDEUR
      JOIN DETAILS_COMMANDES DC ON DC.NO_COMMANDE = CO.NO_COMMANDE
WHERE extract(year from CO.DATE_COMMANDE) = 2019
      and extract(month from CO.DATE_COMMANDE) = 5
GROUP BY VN.pays
      ,EM.NOM
      ,EM.PRENOM
HAVING SUM(DC.PORT) > 80000
ORDER BY VN.pays, EM.NOM;

```

Exercice n° 04

Le nom, le prénom, la fonction des **tous** les employés, le nom et prénom des employés gérés par eux, si tel est le cas, ainsi que le nom et prénom des employés gérés par les précédents si tel est le cas.

Résultat de la requête :

Employé A	FONCTION	Employé B Gérés par Employé A	Gérés par Employé B
28 Besse José	Représentant(e)		
29 Bettan Henri-Michel	Représentant(e)		
30 Blard Jean-Benoît	Représentant(e)		
31 Bodard René	Représentant(e)		
32 Brasseur Hervé	Vice-Président	Belin Chantal	Aubert Maria
33 Brasseur Hervé	Vice-Président	Belin Chantal	Blard Jean-Benoît
34 Brasseur Hervé	Vice-Président	Belin Chantal	Cazade Anne-Claire
35 Brasseur Hervé	Vice-Président	Belin Chantal	Charles Philippe
36 Brasseur Hervé	Vice-Président	Belin Chantal	Cremel Brigitte
37 Brasseur Hervé	Vice-Président	Belin Chantal	Guillosoou Nadine
38 Brasseur Hervé	Vice-Président	Belin Chantal	Idesheim Annick
39 Brasseur Hervé	Vice-Président	Belin Chantal	King Robert
40 Brasseur Hervé	Vice-Président	Belin Chantal	Kremser Arnaud
41 Brasseur Hervé	Vice-Président	Belin Chantal	Lefebvre Michel
42 Brasseur Hervé	Vice-Président	Belin Chantal	Maillard Corinne
43 Brasseur Hervé	Vice-Président	Belin Chantal	Marielle Michel
44 Brasseur Hervé	Vice-Président	Belin Chantal	Mennetrier Alexandra
45 Brasseur Hervé	Vice-Président	Belin Chantal	Messelier Philippe
46 Brasseur Hervé	Vice-Président	Belin Chantal	Mure Guv
47 Brasseur Hervé	Vice-Président	Belin Chantal	Pequignot Guv
48 Brasseur Hervé	Vice-Président	Belin Chantal	Seigne Marie-Pierre

```

SELECT A.NOM||' '||A.PRENOM "Employé A",
       A.FONCTION,
       B.NOM||' '||B.PRENOM "Employé B Gérés par Employé A",
       C.NOM||' '||C.PRENOM "Gérés par Employé B"
FROM EMPLOYES A LEFT OUTER JOIN EMPLOYES B
      ON ( A.NO_EMPLOYE = B.REND_COMPTE )
LEFT OUTER JOIN EMPLOYES C
      ON ( B.NO_EMPLOYE = C.REND_COMPTE )
ORDER BY "Employé A","Employé B Gérés par Employé A","Gérés par Employé B" ;

```

Exercice n° 05

Le pays et les fournisseurs pour les vendeurs du même pays que les fournisseurs.

Affichez le nombre de commandes et la somme des quantités vendues par année pour les enregistrements qui comportent plus de 600 commandes.

Résultat de la requête :

	PAYS	SOCIETE	VENDEUR	ANNEE	QUANTITES	NB_COMMANDES
1	États-Unis	Bigfoot Breweries	Coutou MYRIAM	2017	145341	646
2	États-Unis	Bigfoot Breweries	Silberreiss ALBERT	2017	148589	668
3	États-Unis	Nouvelle-Orléans Cajun Delights	Silberreiss ALBERT	2017	170732	670
4	États-Unis	Nouvelle-Orléans Cajun Delights	Coutou MYRIAM	2017	174007	674
5	États-Unis	Bigfoot Breweries	Urbaniak ISABELLE	2017	153798	678
6	États-Unis	Nouvelle-Orléans Cajun Delights	Urbaniak ISABELLE	2017	163604	708
7	États-Unis	Bigfoot Breweries	Coutou MYRIAM	2018	143016	646
8	États-Unis	Nouvelle-Orléans Cajun Delights	Coutou MYRIAM	2018	167894	685
9	États-Unis	Bigfoot Breweries	Urbaniak ISABELLE	2018	162056	714
10	États-Unis	Bigfoot Breweries	Silberreiss ALBERT	2018	165055	723
11	États-Unis	Nouvelle-Orléans Cajun Delights	Urbaniak ISABELLE	2018	179581	734
12	États-Unis	Nouvelle-Orléans Cajun Delights	Silberreiss ALBERT	2018	177937	740

```

SELECT VN.PAYS
      ,FO.SOCIETE
      ,EM.NOM||' '||UPPER(EM.PRENOM) VENDEUR
      ,EXTRACT(YEAR FROM CO.DATE_COMMANDE)
      ,SUM(DC.QUANTITE)
      ,COUNT(DISTINCT CO.NO_COMMANDE) NB_COMMANDES
FROM EMPLOYES EM
      JOIN VENDEURS VN ON VN.NO_VENDEUR = EM.NO_EMPLOYE
      JOIN COMMANDES CO ON CO.NO_VENDEUR = VN.NO_VENDEUR
      JOIN DETAILS_COMMANDES DC ON DC.NO_COMMANDE = CO.NO_COMMANDE
      JOIN PRODUITS PR ON PR.REF_PRODUIT = DC.REF_PRODUIT
      JOIN FOURNISSEURS FO ON FO.NO_FOURNISSEUR = PR.NO_FOURNISSEUR
      AND FO.PAYS = VN.PAYS
GROUP BY FO.SOCIETE
      ,VN.PAYS
      ,EM.NOM
      ,EM.PRENOM, EXTRACT(YEAR FROM CO.DATE_COMMANDE)
HAVING COUNT(DISTINCT CO.NO_COMMANDE) > 600
ORDER BY EXTRACT(YEAR FROM CO.DATE_COMMANDE) , NB_COMMANDES;

```

Exercice n° 06

Le fournisseur, le produit et les unités en stock pour les produits qui ont un stock inférieur à la moyenne des unités en stock pour les produits de la catégorie « Desserts » du même fournisseur. Triez les enregistrements suivant les fournisseurs par ordre croissante et les unités en stock par ordre décroissante.

Résultat de la requête :

NOM_CATEGORIE	FOURNISSEUR	NOM_PRODUIT	UNITES_STOCK
¹ Desserts	Forêts d'érables	Tarte au sucre	10076
² Desserts	Karkki Oy	Maxilaku	252669
³ Desserts	Pavlova, Ltd.	Pavlova	223219
⁴ Desserts	Specialty Biscuits, Ltd.	Sir Rodney's Marmalade	359109
⁵ Desserts	Specialty Biscuits, Ltd.	Scottish Longbreads	325213
⁶ Desserts	Specialty Biscuits, Ltd.	Teatime Chocolate Biscuits	293772
⁷ Desserts	Tokvo Traders	Scones	336923
⁸ Desserts	Zaanse Snoepfabriek	Zaanse koeken	147815

```

SELECT NOM_CATEGORIE, SOCIETE FOURNISSEUR, NOM_PRODUIT, UNITES_STOCK
FROM PRODUITS PR1
  JOIN CATEGORIES CA ON PR1.CODE_CATEGORIE = CA.CODE_CATEGORIE
  JOIN FOURNISSEURS FR ON FR.NO_FOURNISSEUR = PR1.NO_FOURNISSEUR
WHERE UNITES_STOCK > (SELECT AVG (PR2.UNITES_STOCK)
                      FROM PRODUITS PR2
                      WHERE PR1.NO_FOURNISSEUR = PR2.NO_FOURNISSEUR)
  AND NOM_CATEGORIE = 'Desserts'
ORDER BY SOCIETE, UNITES_STOCK DESC;

```

Exercice n° 07

Le nombre de commandes, les quantités vendues par année et par mois pour les fournisseurs français. Calculez également le pourcentage des frais de port dans la somme annuelle des frais de port et la somme cumulative des quantités dans l'année.

Résultat de la requête :

	ANNEE	MOIS	QUANTITES	%	Somme cumulative
1	2018	1	575863	8,98	575863
2	2018	2	488396	7,62	1064259
3	2018	3	520608	8,12	1584867
4	2018	4	562569	8,78	2147436
5	2018	5	562108	8,77	2709544
6	2018	6	588022	9,17	3297566
7	2018	7	498332	7,78	3795898
8	2018	8	577841	9,02	4373739
9	2018	9	487977	7,61	4861716
10	2018	10	568390	8,87	5430106
11	2018	11	510694	7,97	5940800
12	2018	12	468392	7,31	6409192
13	2019	1	398376	9	398376
14	2019	2	361374	8,16	759750
15	2019	3	351871	7,95	1111621
16	2019	4	356397	8,05	1468018
17	2019	5	342915	7,75	1810933
18	2019	6	343697	7,76	2154630
19	2019	7	369969	8,36	2524599
20	2019	8	403030	9,1	2927629
21	2019	9	373971	8,45	3301600
22	2019	10	360939	8,15	3662539
23	2019	11	399002	9,01	4061541
24	2019	12	365006	8,25	4426547

```

SELECT FO.PAYS
,CO.ANNEE
,CO.MOIS
,SUM(DC.QUANTITE) QUANTITES
,ROUND(SUM(DC.QUANTITE)*100/
SUM(SUM(DC.QUANTITE)) OVER
(PARTITION BY ANNEE),2) "%"
,SUM(SUM(DC.QUANTITE)) OVER
(PARTITION BY ANNEE ORDER BY MOIS) "Somme cumulative"
FROM COMMANDES CO
JOIN DETAILS_COMMANDES DC ON DC.NO_COMMANDE = CO.NO_COMMANDE
JOIN PRODUITS PR ON PR.REF_PRODUIT = DC.REF_PRODUIT
JOIN FOURNISSEURS FO ON FO.NO_FOURNISSEUR = PR.NO_FOURNISSEUR
AND FO.PAYS = 'France'
JOIN CATEGORIES CA ON PR.CODE_CATEGORIE = CA.CODE_CATEGORIE
GROUP BY FO.PAYS,ANNEE,MOIS;

```


Python -- DataFrames :

Exercice n° 08

Ecrivez le code python pour la transformation du DataFrame initiale en résultat :

```
1 commandes.head()
```

executed in 26ms, finished 19:19:00 2021-12-25

	no_commande	code_client	no_employe	date_commande	date_envoi	port	livree	acquitee	annulee	annee	trimestre	mois
0	215414	CACTU	90	2010-01-21	2010-03-05	96.0	1	1	0	2010	1	1
1	215427	LONEP	79	2010-01-21	2010-02-08	55.8	1	1	0	2010	1	1
2	215420	EASTC	62	2010-01-21	2010-03-06	64.8	1	1	0	2010	1	1
3	215416	GREAL	84	2010-01-21	2010-02-15	57.6	1	1	0	2010	1	1
4	215422	WARTH	54	2010-01-21	2010-02-18	63.9	1	1	0	2010	1	1

Résultat de la transformation :

			port			
			sum	mean	min	max
annee	trimestre	mois				
2010	1	1	49332.8	75.547933	50.0	100.0
		2	46707.1	74.731360	50.1	100.0
		3	49768.4	74.392227	50.3	99.9
	2	4	46904.0	75.046400	50.0	100.0
		5	50921.5	74.446637	50.1	100.0
		6	49397.9	74.619184	50.1	99.9
	3	7	49321.8	75.531087	50.1	99.9

```
commandes.groupby(
    ['annee', 'trimestre', 'mois']
).agg({'port': ['sum', 'mean', 'min', 'max']})
```

Exercice n° 09

Ecrivez le code python pour la transformation du DataFrame initiale en résultat :

	Annee	Mois	Zone	Temperature
0	1997	1	NE	-1.491549
1	1997	1	NO	1.963876
2	1997	1	SE	5.708186
3	1997	1	SO	5.028942
4	1997	2	NE	5.810785
5	1997	2	NO	7.705910
6	1997	2	SE	8.345754
7	1997	2	SO	8.940475
8	1997	3	NE	8.512270
9	1997	3	NO	9.491090
10	1997	3	SE	11.195458
11	1997	3	SO	11.057508
12	1997	4	NE	8.967430
13	1997	4	NO	10.093495

Résultat de la transformation :

	Temperature			
Zone	NE	NO	SE	SO
Annee				
1997	10.652706	11.842338	13.977437	13.563081
1998	10.491352	11.422466	13.245572	12.520386
1999	10.916535	12.035814	13.535628	12.830786
2000	11.376685	11.784401	13.793652	12.984124
2001	10.710783	11.577554	13.625802	12.542115
2002	11.276275	12.058265	13.828998	12.934789
2003	11.365645	12.151587	14.278178	13.635487
2004	10.576076	11.576703	13.402740	12.543694
2005	10.694333	11.754246	13.025132	12.338676
2006	11.080125	11.962027	14.027730	13.308124

```
temperatures = meteo01[['Zone', 'Annee', 'Temperature']]
                    .pivot_table(index=['Annee'],
                                columns='Zone')
temperatures.head(10)
```

Exercice n° 10

Ecrivez le code python pour la transformation du DataFrame initiale en résultat :

```
1 meteo01.head(14)
```

executed in 21ms, finished 20:31:55 2021-12-25

	Annee	Mois	Zone	Temperature
0	1997	1	NE	-1.491549
1	1997	1	NO	1.963876
2	1997	1	SE	5.708186
3	1997	1	SO	5.028942
4	1997	2	NE	5.810785
5	1997	2	NO	7.705910

Résultat de la transformation :

```
1 meteo01[(meteo01.Mois < 3)&
2 (meteo01.Annee < 1999)].head(16)
```

executed in 20ms, finished 20:37:18 2021-12-25

	Annee	Mois	Zone	Temperature	avg_AnneeMois	avg_ZoneMois
0	1997	1	NE	-1.491549	2.802364	2.698996
1	1997	1	NO	1.963876	2.802364	5.608975
2	1997	1	SE	5.708186	2.802364	5.853549
3	1997	1	SO	5.028942	2.802364	5.630594
4	1997	2	NE	5.810785	7.700731	3.678959
5	1997	2	NO	7.705910	7.700731	6.040014
6	1997	2	SE	8.345754	7.700731	6.554702
7	1997	2	SO	8.940475	7.700731	6.432651
48	1998	1	NE	3.611864	5.570234	2.698996
49	1998	1	NO	5.708332	5.570234	5.608975
50	1998	1	SE	6.380107	5.570234	5.853549
51	1998	1	SO	6.580631	5.570234	5.630594
52	1998	2	NE	4.169858	6.516558	3.678959
53	1998	2	NO	6.477876	6.516558	6.040014
54	1998	2	SE	7.661633	6.516558	6.554702
55	1998	2	SO	7.756867	6.516558	6.432651

```
meteo01['avg_AnneeMois'] =
meteo01.groupby(['Annee', 'Mois']).Temperature.transform('mean')
meteo01['avg_ZoneMois'] =
meteo01.groupby(['Zone', 'Mois']).Temperature.transform('mean')
```

Exercice n° 11

Ecrivez le code python pour la transformation du DataFrame initiale en résultat :

	Annee	Mois	Zone	Precipitation
0	1997	1	NE	149.500
1	1997	1	NO	257.375
2	1997	1	SE	1177.300
3	1997	1	SO	386.750
4	1997	2	NE	448.850
5	1997	2	NO	1193.625
6	1997	2	SE	175.575
7	1997	2	SO	410.900
8	1997	3	NE	112.300
9	1997	3	NO	175.850

Résultat de la transformation :

	Annee	Mois	Zone	Precipitation	sum_AnneeMois	sum_CumulAnnee
0	1997	1	NE	149.500	1970.925	149.500
1	1997	1	NO	257.375	1970.925	406.875
2	1997	1	SE	1177.300	1970.925	1584.175
3	1997	1	SO	386.750	1970.925	1970.925
4	1997	2	NE	448.850	2228.950	2419.775
5	1997	2	NO	1193.625	2228.950	3613.400
6	1997	2	SE	175.575	2228.950	3788.975
7	1997	2	SO	410.900	2228.950	4199.875
8	1997	3	NE	112.300	421.700	4312.175
9	1997	3	NO	175.850	421.700	4488.025
10	1997	3	SE	49.250	421.700	4537.275
11	1997	3	SO	84.300	421.700	4621.575
12	1997	4	NE	148.250	905.425	4769.825

```
meteo03['sum_AnneeMois'] =
meteo03.groupby(['Annee', 'Mois']).Precipitation.transform('sum')
meteo03['sum_CumulAnnee'] =
meteo03.groupby(['Annee']).Precipitation.transform('cumsum')
```

Exercice n° 12

Ecrivez le code python pour la transformation du DataFrame initiale en résultat :

1	patients.head()					
executed in 14ms, finished 19:08:58 2021-12-26						
	annee	gender	alzheimer	heartfailure	cancer	stroke
0	1928	homme	1	0	1	1
1	1923	homme	0	1	1	1
2	1915	femme	0	1	1	1
3	1929	homme	1	1	0	1
4	1930	femme	1	1	1	1

Résultat de la transformation :

		alzheimer			heartfailure			cancer			stroke		
		sum	min	max	sum	min	max	sum	min	max	sum	min	max
annee	gender												
1983	homme	59	0	1	43	0	1	74	0	1	78	0	1
	femme	62	0	1	47	0	1	84	0	1	81	0	1
1982	homme	66	0	1	58	0	1	94	0	1	94	0	1
	femme	54	0	1	52	0	1	80	0	1	82	0	1
1981	homme	83	0	1	65	0	1	106	0	1	109	0	1
	femme	57	0	1	55	0	1	85	0	1	83	0	1
1980	homme	79	0	1	67	0	1	96	0	1	99	0	1
	femme	62	0	1	52	0	1	91	0	1	86	0	1
1979	homme	69	0	1	49	0	1	81	0	1	83	0	1
	femme	48	0	1	44	0	1	70	0	1	71	0	1

```
patients.groupby(['annee',
                  'gender'])
            .agg(['sum', 'min', 'max']).sort_index(ascending=False)
```

Exercice n° 13

Ecrivez le code python pour la transformation des DataFrame initiales en résultat :

```
1 patients.head()
```

executed in 19ms, finished 19:22:05 2021-12-26

	patient_id	date_naissance	date_deces	gender	alzheimer	heartfailure	kidneydisease	cancer	obstrpulmonary	depression	diabetes	ischemicheart	osteo
0	BENE11207	1928-06-01	NaT	homme	oui	non	non	oui	oui	non	non	non	
1	BENE11208	1923-09-01	NaT	homme	non	oui	non	oui	oui	non	non	non	
2	BENE11209	1915-07-01	NaT	femme	non	oui	oui	oui	oui	non	non	non	
3	BENE11210	1929-06-01	NaT	homme	oui	oui	oui	non	oui	oui	non	non	
4	BENE11211	1930-07-01	NaT	femme	oui	oui	oui	oui	oui	oui	non	oui	

```
1 patients_hospitalises.head()
```

executed in 16ms, finished 19:22:10 2021-12-26

	patient_id	reclamation_id	debut_reclamation	fin_reclamation	provider	remboursement	date_admission	franchise	date_sortie
0	BENE11769	CLM65259	2009-08-25	2009-09-04	PRV56421	19000	2009-08-25	1068.0	2009-09-04
1	BENE11772	CLM73448	2009-10-26	2009-10-29	PRV54350	9000	2009-10-26	1068.0	2009-10-29
2	BENE11778	CLM31851	2009-01-01	2009-01-08	PRV54692	20000	2009-01-01	1068.0	2009-01-08
3	BENE11791	CLM68361	2009-09-17	2009-09-18	PRV54692	2000	2009-09-17	1068.0	2009-09-18
4	BENE11793	CLM56484	2009-06-21	2009-06-26	PRV55209	11000	2009-06-21	1068.0	2009-06-26

Résultat de la transformation :

```
1 resultat[['date_naissance','gender','date_admission',
2          'date_sortie','remboursement','franchise']].head()
```

executed in 40ms, finished 19:29:15 2021-12-26

	patient_id	date_naissance	gender	date_admission	date_sortie	remboursement	franchise
	BENE100002	1938-09-01	femme	2009-04-21	2009-05-16	12000	1068.0
	BENE100004	1943-06-01	femme	2009-08-02	2009-08-04	3000	1068.0
	BENE100006	1952-01-01	femme	2009-11-22	2009-11-28	17000	1068.0
	BENE100007	1939-03-01	homme	2009-01-02	2009-01-06	4000	1068.0
	BENE100010	1927-10-01	homme	2009-04-15	2009-05-01	8000	1068.0

```
resultat = patients.set_index('patient_id'
                             ).join(patients_hospitalises.set_index('patient_id')
                             ,how='inner')
resultat[['date_naissance','gender','date_admission',
          'date_sortie','remboursement','franchise']].head()
```