

SAE Phase 3 - Rapport (B12)

Sommaire

- 1) Initialisation des tables
 - a) Suppression des tables pré-existantes
 - b) Création des tables
 - c) Copie des données
 - 2) Rapport sur les requêtes imposées (6 requêtes)
 - 3) Rapport sur les requêtes inventées (8 requêtes)
 - 4) Script sous format texte
 - 5) Capture d'écrans du Notebook
-

1) Initialisation des tables

a) Suppression des tables pré-existantes

```
drop table if exists survivant;  
drop table if exists passager;  
drop table if exists ticket;
```

Permet la suppression des tables si elles existent déjà auparavant et permet la recréation des tables si on effectue des modifications sur le script et que l'on le relance.

b) Création des tables

```
create table ticket(Name varchar(100) primary key, Pclass int, Ticket varchar(20),  
Fare float, Cabin varchar(20), Embarked varchar(1));  
  
create table passager(PassengerId int primary key, Name varchar(100), Sex varchar(6), Age float, SibSp int, Parch int,  
foreign key (Name) references ticket(Name));  
  
create table survivant(PassengerId int, Name varchar(100), Survived int,  
foreign key (PassengerId) references passager(passengerid),  
foreign key (Name) references ticket(name));
```

Dans cette phase 3, nous réutilisons les 3 tables que nous avons créées en phase 2: les tables **ticket**, **passager** et **survivant**.

(suite à la page suivante)

- **Ticket** répertorie les informations susceptible d'apparaître sur un ticket pour un séjour sur le Titanic, avec les attributs suivant: name (clé primaire), pclass, ticket, fare, cabin, embarked

name	pclass	ticket	fare	cabin	embarked
Braund, Mr. Owen Harris	3	A/5 21171	7.25		S
Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1	PC 17599	71.2833	C85	C
Heikkinen, Miss. Laina	3	STON/O2. 3101282	7.925		S
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	113803	53.1	C123	S
Allen, Mr. William Henry	3	373450	8.05		S
Moran, Mr. James	3	330877	8.4583		Q
McCarthy, Mr. Timothy J	1	17463	51.8625	E46	S
Palsson, Master. Gosta Leonard	3	349909	21.075		S
Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	3	347742	11.1333		S
Nasser, Mrs. Nicholas (Adele Achem)	2	237736	30.0708		C
Sandstrom, Miss. Marguerite Rut	3	PP 9549	16.7	G6	S
Bonnell, Miss. Elizabeth	1	113783	26.55	C103	S
Saunderscock, Mr. William Henry	3	A/5. 2151	8.05		S
Andersson, Mr. Anders Johan	3	347082	31.275		S
Vestrom, Miss. Hulda Amanda Adolfina	3	350406	7.8542		S
Hewlett, Mrs. (Mary D Kingcome)	2	248706	16		S
Rice, Master. Eugene	3	382652	29.125		Q
Williams, Mr. Charles Eugene	2	244373	13		S
Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	3	345763	18		S
Maselmani, Mrs. Fatima	3	2649	7.225		C
Fynney, Mr. Joseph J	2	239865	26		S
Beesley, Mr. Lawrence	2	248698	13	D56	S

- **Passager** répertorie les informations sur l'identité des passagers et leur famille, avec les attributs passengerid (clé primaire), name (clé étrangère de **ticket**), sex, age, sibsp, parch

passengerid	name	sex	age	sibsp	parch
1	Braund, Mr. Owen Harris	male	22	1	0
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0
3	Heikkinen, Miss. Laina	female	26	0	0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0
5	Allen, Mr. William Henry	male	35	0	0
6	Moran, Mr. James	male	0	0	0
7	McCarthy, Mr. Timothy J	male	54	0	0
8	Palsson, Master. Gosta Leonard	male	2	3	1
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2
10	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0
11	Sandstrom, Miss. Marguerite Rut	female	4	1	1
12	Bonnell, Miss. Elizabeth	female	58	0	0

- **Survivant** répertorie les données concernant la survie des passagers, avec les attributs passengerid et name (clés étrangères de **passager** et **ticket** respectivement), ainsi que de survived qui est l'indicateur de survie

passengerid	name	survived
1	Braund, Mr. Owen Harris	0
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1
3	Heikkinen, Miss. Laina	1
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1
5	Allen, Mr. William Henry	0
6	Moran, Mr. James	0
7	McCarthy, Mr. Timothy J	0
8	Palsson, Master. Gosta Leonard	0
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	1
10	Nasser, Mrs. Nicholas (Adele Achem)	1
11	Sandstrom, Miss. Marguerite Rut	1
12	Bonnell, Miss. Elizabeth	1
13	Saunderscock, Mr. William Henry	0

c) Copie des données

```
\copy ticket from '/Users/Asus/Documents/ticket.txt'
\copy passager from '/Users/Asus/Documents/passager.txt'
\copy survivant from '/Users/Asus/Documents/survivant.txt'
```

Les données du tableau sont copiées depuis des fichiers .txt que j'ai créés à partir des données du fichier *titanic_train.csv*. Les données de chaque tuples sont séparées par une tabulation pour correspondre à leur attribut respectif. Ci-dessous, des extraits

ticket.txt - Bloc-notes

Fichier	Edition	Format	Affichage	Aide
Braund, Mr. Owen Harris	3	A/5	21171	7.25
Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1	PC	17599	71.2833 C85
Heikkinen, Miss. Laina	3	STON/02.	3101282	7.925
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	113803	53.1	C123 S
Allen, Mr. William Henry	3	373450	8.05	S
Moran, Mr. James	3	330877	8.4583	Q
McCarthy, Mr. Timothy J	1	17463	51.8625	E46 S
Palsson, Master. Gosta Leonard	3	349909	21.075	S
Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	3	347742	11.1333	S
Nasser, Mrs. Nicholas (Adele Achem)	2	237736	30.0708	C
Sandstrom, Miss. Marguerite Rut	3	PP 9549	16.7	G6 S
Bonnell, Miss. Elizabeth	1	113783	26.55	C103 S
Saunderscock, Mr. William Henry	3	A/5.	2151	8.05 S

passager.txt - Bloc-notes

Fichier	Edition	Format	Affichage	Aide
1	Braund, Mr. Owen Harris	male	22	1 0
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1 0
3	Heikkinen, Miss. Laina	female	26	0 0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1 0
5	Allen, Mr. William Henry	male	35	0 0
6	Moran, Mr. James	male	0	0 0
7	McCarthy, Mr. Timothy J	male	54	0 0
8	Palsson, Master. Gosta Leonard	male	2	3 1
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0 2
10	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1 0
11	Sandstrom, Miss. Marguerite Rut	female	4	1 1
12	Bonnell, Miss. Elizabeth	female	58	0 0
13	Saunderscock, Mr. William Henry	male	20	0 0

survivant.txt - Bloc-notes

Fichier	Edition	Format	Affichage	Aide
1	Braund, Mr. Owen Harris	0		
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1		
3	Heikkinen, Miss. Laina	1		
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1		
5	Allen, Mr. William Henry	0		
6	Moran, Mr. James	0		
7	McCarthy, Mr. Timothy J	0		
8	Palsson, Master. Gosta Leonard	0		
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	1		
10	Nasser, Mrs. Nicholas (Adele Achem)	1		
11	Sandstrom, Miss. Marguerite Rut	1		
12	Bonnell, Miss. Elizabeth	1		
13	Saunderscock, Mr. William Henry	0		

2) Rapport sur les requêtes imposées (6 requêtes)

- Combien de classes de passagers différentes y avait-il à bord du Titanic ? (n°1/6)

- Code SQL

```
select count(*) as classes_de_passagers
  from (select distinct pclass from ticket) as liste_classes;
```

- Résultat sur terminal

```
classes_de_passagers
-----
                        3
(1 ligne)
```

---> Il y avait 3 classes de passagers différentes.

Pour trouver le nombre de classes de passagers, on cherche d'abord les différentes classes en listant les **pclass** (classes de passagers) distinctes de la table **ticket**, ce qui nous donne 3 tuples: 1, 2 et 3.

Ensuite, il suffit simplement de compter le nombre de tuples à partir du résultat précédent avec la fonction d'agrégation **count(*)**: on trouve donc au total 3 classes de passagers différentes.

-
- Combien de passagers y avait-il dans chaque classe? (n°2/6)

- Code SQL

```
select pclass, count(*) as passagers
  from ticket
 group by pclass
 order by pclass;
```

- Résultat sur terminal

```
pclass | passagers
-----+-----
      1 |      216
      2 |      184
      3 |      491
(3 lignes)
```

---> Il y avait 216 passagers en 1ère classe, 184 en 2ème classe et 491 en 3ème classe, ce qui équivaut bien à 891 passagers.

Afin de parvenir au nombre de passagers dans chaque classe, il faut regrouper les passagers en filtrant par leur classe respective **pclass** avec la commande **group by**, et compter les passagers dans chaque groupe avec la fonction d'agrégation **count(*)**, tout cela à partir de la table **ticket**.

Ainsi, on regroupe par exemple les passagers appartenant à la 1ère classe donc **pclass=1**, puis on compte les passagers dans ce groupe: on dénombre 216 passagers en 1ère classe. Le comptage du nombre de passagers pour le restant des classes fonctionne sur le même principe.

Pour être plus précis sur l'affichage, on trie les données selon l'ordre des classes avec la commande **order by**, ici order by pclass.

- Combien de femmes et d'hommes y avait-il dans chaque classe? (n°3/6)

- Code SQL

```
select ticket.pclass, passenger.sex, count(*) as passagers
  from ticket natural join passenger
 group by ticket.pclass,passenger.sex
 order by ticket.pclass,passenger.sex;
```

- Résultat sur terminal

pclass	sex	passagers
1	female	94
1	male	122
2	female	76
2	male	108
3	female	144
3	male	347

(6 lignes)

---> En 1ère classe, il y avait 94 femmes et 122 hommes, en 2ème classe 76 femmes et 108 hommes, en 3ème classe 144 femmes et 347 hommes.

Le principe de cette requête est similaire à la requête précédente, mais il faut désormais en plus filtrer le sexe des passagers parmi chaque groupe. L'attribut **pclass** étant dans la table **ticket** et **sex** dans la table **passager**, il faut réaliser une **jointure naturelle** entre ces deux tables avec comme attribut commun **name**.

On regroupe donc les passagers:

- dans un premier temps par leur classe (**group by ticket.pclass,...**) et

- dans un second temps par leur sexe (**group by ...,passager.sex**),

ce qui nous donne donc le dénombrement des passagers à la fois appartenant à une classe spécifique et à la fois étant un homme ou une femme. Il reste à compter le nombre de tuples respectant les filtres de ces groupes avec **count(*)**.

Il est donc possible de compter par exemple 94 passagers parmi ceux en **pclass=1** et de **sex='female'**.

Pour un meilleur affichage, on trie les données principalement selon l'ordre des classes et puis par le sexe, avec donc la commande **order by ticket.pclass,passager.sex**.

-
- Comptez le nombre et le pourcentage de survivants et de passagers morts. (n°4/6)

- Code SQL

```
select survivant.survived, count(*) as passagers, (count(*) * 100 / (select count(*) from survivant)) as pourcentage
  from survivant
 group by survived;
```

- Résultat sur terminal

survived	passagers	pourcentage
0	549	61
1	342	38

(2 lignes)

---> Après le naufrage du Titanic, il y avait 549 passagers morts et 342 survivants, soit environ 62% de décédés et 38% survivants.

Pour cette requête, il faut regrouper les passagers par leur indicateur de survie constitué de l'attribut **survived** de la table **survivant**, qui prend 0 si l'individu est décédé et 1 s'il a survécu.

Il faut donc filtrer et regrouper les passagers qui ont survécu et ceux qui n'ont pas survécu avec **group by survived**, puis les compter avec **count(*)**.

Les commandes $(count(*) * 100 / (select count(*) from survivant))$ nous permet de calculer le pourcentage de chaque groupe.

-
- Visualiser la répartition des passagers survivants et morts par classe. (n°5/6)

- Code SQL

```
select ticket.pclass, survivant.survived, count(*) as passagers
  from ticket natural join survivant
 group by pclass, survived
 order by pclass, survived;
```

- Résultat sur terminal

pclass	survived	passagers
1	0	80
1	1	136
2	0	97
2	1	87
3	0	372
3	1	119

(6 lignes)

---> A partir de la requête, on peut voir qu'il y avait:

- 80 morts et 136 survivants en 1ère classe,
- 97 morts et 87 survivants en 2ème classe,
- 372 morts et 119 survivants en 3ème classe.

On remarque donc qu'il y a le plus de décès en 3ème classe (372 morts), où les passagers sont en théorie plutôt de classe populaire, et qu'il y a le plus de survivants en 1ère classe (136 survivants), où la population est plus fortunée.

Cela peut-être en partie expliqué par le fait que les passagers de 1ère classe sont situés en haut du Titanic et ceux de la 3ème classe en bas, ce peut déterminer leurs chances de survie.

Il faut néanmoins tenir compte de la différence en termes de nombre de passagers dans chaque classe.

D'un point de vue SQL, on visualise ici la répartition des passagers morts et des survivants selon leur classe de passager, on doit donc filtrer les tuples par leur **pclass** et ensuite par leur état **survived** en utilisant **group by**, sachant que ces attributs sont respectivement de la table **ticket** et **survivant**, donc il faut une **jointure naturelle** sur **name**. On a donc **group by pclass, survived**.

Il faut ensuite compter les tuples de chaque groupes avec **count(*)** pour avoir le nombre de passagers respectant chaque critères/conditions.

-
- Visualiser la répartition des passagers survivants et des passagers décédés selon le sexe. (n°6/6)

- Code SQL

```
select survivant.survived, passager.sex, count(*) as passagers
  from passager natural join survivant
 group by survived, sex
 order by survived;
```

- Résultat sur terminal

survived	sex	passagers
0	female	81
0	male	468
1	female	233
1	male	109

(4 lignes)

---> On remarque qu'il y a un très grand nombre d'hommes passagers qui ont perdu la vie lors du naufrage: 468 hommes sont morts, comparé à 81 femmes. Les femmes représentent une partie importante des survivants: 233 femmes ont survécu comparé à 109 hommes.

Pour cette requête, il faut filtrer et regrouper par l'attribut **survived** et **sex** qui appartiennent aux tables **survivant** et **passager** respectivement dont on fait la **jointure naturelle** sur l'attribut **passengerid**.

On regroupe donc par exemple ceux qui n'ont pas survécu et sont de sexe féminin, donc **survived=0** et **sex='female'**, et on compte avec **count(*)** les tuples respectant cela, pour ainsi obtenir 81 femmes passagères décédées. Ensuite, on répète cela avec **survived=0** et **sex='male'**, **survived=1** et **sex='female'** et ainsi de suite.

Page suivante: Rapport sur les requêtes inventées

3) Rapport sur les requêtes inventées (8 requêtes)

- Donnez l'identifiant et le nom de tous les passagers survivants. (n°1/8)

- Code SQL

```
select passengerid,name
  from survivant
 where survived=1;
```

- Résultat sur terminal

passengerid	name
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3	Heikkinen, Miss. Laina
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
10	Nasser, Mrs. Nicholas (Adele Achem)
11	Sandstrom, Miss. Marguerite Rut
12	Bonnell, Miss. Elizabeth
16	Hewlett, Mrs. (Mary D Kingcome)
18	Williams, Mr. Charles Eugene
20	Masselmani, Mrs. Fatima
22	Beesley, Mr. Lawrence
23	McGowan, Miss. Anna "Annie"
24	Sloper, Mr. William Thompson
26	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)
29	O'Dwyer, Miss. Ellen "Nellie"
32	Spencer, Mrs. William Augustus (Marie Eugenie)
33	Glynn, Miss. Mary Agatha
37	Mamee, Mr. Hanna
40	Nicola-Yarred, Miss. Jamila
44	Laroche, Miss. Simonne Marie Anne Andree
45	Devaney, Miss. Margaret Delia
48	O'Driscoll, Miss. Bridget
53	Harper, Mrs. Henry Sleeper (Myna Haxtun)
54	Faunthorpe, Mrs. Lizzie (Elizabeth Anne Wilkinson)
56	Woolner, Mr. Hugh
57	Rugg, Miss. Emily
59	West, Miss. Constance Mirium
62	Icard, Miss. Amelie
66	Moubarek, Master. Gerios
67	Nye, Mrs. (Elizabeth Ramell)
69	Andersson, Miss. Erna Alexandra
75	Bing, Mr. Lee
79	Caldwell, Master. Alden Gates
80	Dowdell, Miss. Elizabeth
82	Sheerlinck, Mr. Jan Baptist
83	McDermott, Miss. Brigdet Delia
85	Ilett, Miss. Bertha
86	Backstrom, Mrs. Karl Alfred (Maria Mathilda Gustafsson)
89	Fortune, Miss. Mabel Helen
98	Greenfield, Mr. William Bertram
99	Doling, Mrs. John T (Ada Julia Bone)
107	Salkjelsvik, Miss. Anna Kristine
108	Moss, Mr. Albert Johan
110	Moran, Miss. Bertha
124	Webber, Miss. Susan
126	Nicola-Yarred, Master. Elias
128	Madsen, Mr. Fridtjof Arne
129	Peter, Miss. Anna
134	Weisz, Mrs. Leopold (Mathilde Françoise Pede)
137	Newsom, Miss. Helen Monypeny
142	Nysten, Miss. Anna Sofia
143	Hakkarainen, Mrs. Pekka Pietari (Elin Matilda Dolck)
147	Andersson, Mr. August Edvard ("Wennerstrom")
152	Pears, Mrs. Thomas (Edith Wearne)
157	Gilnagh, Miss. Katherine "Katie"
162	Watt, Mrs. James (Elizabeth "Bessie" Inglis Milne)
166	Goldsmith, Master. Frank John William "Frankie"
167	Chibnall, Mrs. (Edith Martha Bowerman)

Pour répondre à cette requête, il faut sélectionner les identifiants (**passengerid**) et les noms (**name**) appartenant aux tuples de la table **survivant** (**from survivant**) où la condition d'indicateur de survie **survived** est égale à 1 (**where survived=1**).

- Donnez le nom, le ticket et l'âge des hommes âgés de plus ou égal 50 ans qui ont fait embarcation depuis le port de Cherbourg (C). (n°2/8)

- Code SQL

```
select ticket.name,ticket.ticket, passager.age
  from ticket natural join passager
 where ticket.embarked='C'and passager.age>=50 and passager.sex='male';
```

- Résultat sur terminal

name	ticket	age
Ostby, Mr. Engelhart Cornelius	113509	65
Goldschmidt, Mr. George B	PC 17754	71
Williams, Mr. Charles Duane	PC 17597	51
Smith, Mr. James Clinch	17764	56
Kent, Mr. Edward Austin	11771	58
Artagaveytia, Mr. Ramon	PC 17609	71
Douglas, Mr. Walter Donald	PC 17761	50
Frolicher-Stehli, Mr. Maxmillian	13567	60
Simonius-Blumer, Col. Oberst Alfons	13213	56
Newell, Mr. Arthur Webster	35273	58
(10 lignes)		

Cette requête demande à ce que l'on sélectionne à afficher le nom (**name**), le ticket (**ticket.ticket**) et l'âge (**age**) dont **name** et **ticket.ticket** appartenant à la table **ticket** et âge appartenant à la table **passager**, il faut donc réaliser une **jointure naturelle** sur l'attribut commun **name**. De plus, on ne sélectionne que les tuples respectant 3 conditions: **where embarked='C'** et **age>=50** et **sex='male'** , donc les passagers qui ont embarqué depuis Cherbourg, qui ont plus ou égal 50 ans et qui sont de sexe masculin.

-
- Donnez l'âge du passager le plus âgé. (n°3/8)

- Code SQL

```
select max(age) as age
  from passager;
```

- Résultat sur terminal

```
age
----
 80
(1 ligne)
```

---> Le passager le plus âgé a 80 ans.

Cette requête exige simple de sélectionner l'âge (**age**) du tuple parmi les passager avec l'âge le plus élevé, avec la fonction d'agrégation **max()** dans la table **passager**.

-
- Combien y'avait-il de mineurs parmi les passagers (strictement moins de 18 ans) ? (n°4/8)

- Code SQL

```
--Les passagers dont l'age est inconnu ont un age de 0.0
select count(*) as mineurs
  from passager
 where age>0.0 and age<18;
```

- Résultat sur terminal

```
mineurs
-----
      113
(1 ligne)
```

---> Parmi les passagers, il y a 113 mineurs ayant donc moins de 18 ans.

Pour répondre à cette requête, il faut compter avec **count(*)** les tuples de la table **passager** respectant les conditions **where age>0.0** et **age<18**, c'est-à-dire qui sont âgés de plus de 0 et strictement moins que 18 ans.

- Combien y'avait-il de femmes parmi les passagers ? (n°5/8)

- Code SQL

```
select sex, count(*) as passagers
  from passager
 where sex='female'
 group by sex;
```

- Résultat sur terminal

```
sex | passagers
-----+-----
female |      314
(1 ligne)
```

---> Il y avait 314 passagères à bord du Titanic.

Afin de trouver le nombre de femmes parmi les passagers, on regroupe les passagers respectant la condition **where sex='female'** et on compte les tuples avec la fonction d'agrégation **count(*)** dans la table **passager**.

- Combien y'avait-il de survivants au naufrage du Titanic ? (n°6/8)

- Code SQL

```
select survived, count(*) as passagers
  from survivant
 where survived=1
 group by survived;
```

- Résultat sur terminal

```
survived | passagers
-----+-----
      1 |      342
(1 ligne)
```

---> Il y avait 342 survivants au naufrage du Titanic.

Afin de trouver le nombre de survivants parmi les passagers, on regroupe les passagers respectant la condition **where survived=1** et on compte les tuples avec la fonction d'agrégation **count(*)** dans la table **survivant**.

-
- **Combien y'avait-il de femmes mineures qui ont survécu au naufrage ? (n°7/8)**

- Code SQL

```
select count(*) as mineures_survivantes
  from passager natural join survivant
 where survived=1 and sex='female' and (age>0.0 and age<18.0);
```

- Résultat sur terminal

```
mineures_survivantes
-----
                38
(1 ligne)
```

---> Il y avait 38 femmes mineures qui ont survécu.

Pour cette requête, on compte avec la fonction d'agrégation **count(*)** les tuples des tables **passager** et **survivant** dont on a fait la **jointure naturelle** qui respectent les conditions **where survived=1** et **sex='female'** et **(age>0.0 et age<18.0)**.

-
- **A combien s'élève les profits réalisés à partir de la vente des tickets pour le voyage à bord du Titanic ? (n°8/8)**

- Code SQL

```
select sum(fare) as benefices
  from ticket;
```

- Résultat sur terminal

```
benefices
-----
28693.9493
(1 ligne)
```

---> Au total, les bénéfices de la vente des tickets s'élève à environ 28693.95 £.

Ici, on additionne les tarifs (**fare**) de tous les tuples de la table **ticket** avec la fonction d'agrégation **sum()** pour trouver ce résultat.

4) Script sous format texte

```
-- Initialisation

drop table if exists survivant;
drop table if exists passager;
drop table if exists ticket;

create table ticket(Name varchar(100) primary key, Pclass int, Ticket varchar(20),
    Fare float, Cabin varchar(20), Embarked varchar(1));

create table passager(PassengerId int primary key, Name varchar(100), Sex varchar(6), Age float, SibSp int, Parch int,
    foreign key (Name) references ticket(Name));

create table survivant(PassengerId int, Name varchar(100), Survived int,
    foreign key (PassengerId) references passager(PassengerId),
    foreign key (Name) references ticket(Name));

\copy ticket from '/Users/Asus/Documents/ticket.txt'
\copy passager from '/Users/Asus/Documents/passager.txt'
\copy survivant from '/Users/Asus/Documents/survivant.txt'

select * from ticket;
select * from passager;
select * from survivant;

-- Requetes imposées

-- Combien de classes de passagers différentes y avait-il à bord du Titanic ?

select count(*) as classes_de_passagers
    from (select distinct pclass from ticket) as liste_classes;

-- Combien de passagers y avait-il dans chaque classe?

select pclass, count(*) as passagers
    from ticket
    group by pclass
    order by pclass;

-- Combien de femmes et d'hommes y avait-il dans chaque classe?

select ticket.pclass, passager.sex, count(*) as passagers
    from ticket natural join passager
    group by ticket.pclass, passager.sex
    order by ticket.pclass, passager.sex;

-- Comptez le nombre et le pourcentage de survivants et de passagers morts.

select survivant.survived, count(*) as passagers, (count(*) * 100 / (select count(*) from survivant)) as pourcentage
    from survivant
    group by survived;

-- Visualiser la répartition des passagers survivants et morts par classe

select ticket.pclass, survivant.survived, count(*) as passagers
    from ticket natural join survivant
    group by pclass, survived
    order by pclass, survived;

-- Visualiser la répartition des passagers survivants et des passagers décédés selon le sexe

select survivant.survived, passager.sex, count(*) as passagers
    from passager natural join survivant
    group by survived, sex
    order by survived;

-- Requetes inventées

-- Donnez l'identifiant et le nom de tous les passagers survivants.

select PassengerId, Name
```

```
from survivor
where survived=1;

-- Donnez le nom, le ticket et l'âge des hommes âgés de plus ou égal 50 ans qui ont fait embarcation depuis le port de
Cherbourg (C).

select ticket.name,ticket.ticket, passenger.age
  from ticket natural join passenger
  where ticket.embarked='C'and passenger.age>=50 and passenger.sex='male';

-- Donnez l'âge du passager le plus âgé.

select max(age) as age
  from passenger;

-- Combien y'avait-il de mineurs parmi les passagers (strictement moins de 18 ans) ?

--Les passagers dont l'age est inconnu ont un age de 0.0
select count(*) as mineurs
  from passenger
  where age>0.0 and age<18;

-- Combien y'avait-il de femmes parmi les passagers ?

select sex, count(*) as passagers
  from passenger
  where sex='female'
 group by sex;

-- Combien y'avait-il de survivants au naufrage du Titanic ?

select survived, count(*) as passagers
  from survivor
  where survived=1
 group by survived;

-- Combien y'avait-il de femmes mineures qui ont survécus au naufrage ?

select count(*) as mineures_survivantes
  from passenger natural join survivor
  where survived=1 and sex='female' and (age>0.0 and age<18.0);

-- A combien s'élève les profits réalisés à partir de la vente des tickets pour le voyage à bord du Titanic ?

select sum(fare) as benefices
  from ticket;

--Commandes de réinitialisations

--delete from survivor;
--delete from passenger;
--delete from ticket;
```

Page suivante: Capture d'écrans du Notebook

5) Capture d'écrans du Notebook

```
1 %load_ext sql

1 %sql postgresql://user1:pw2021@localhost/postgres
```

1) Initialisation des tables

```
1 %%sql
2 CREATE TABLE ticket(Name varchar(100) primary key, Pclass int, Ticket varchar(20),
3 Fare float, Cabin varchar(20), Embarked varchar(1));
4
5 CREATE TABLE passager(PassengerId int primary key, Name varchar(100), Sex varchar(6), Age float, Sibsp int, Parch int,
6 foreign key (Name) references ticket(Name));
7
8 CREATE TABLE survivant(PassengerId int, Name varchar(100), Survived int,
9 foreign key (PassengerId) references passager(PassengerId),
10 foreign key (Name) references ticket(Name));

* postgresql://user1:***@localhost/postgres
Done.
Done.
Done.
```

```
1 %%sql
2 \copy ticket from '/Users/Asus/Documents/ticket.txt'

* postgresql://user1:***@localhost/postgres
0 rows affected.

1 %%sql
2 \copy passager from '/Users/Asus/Documents/passager.txt'

* postgresql://user1:***@localhost/postgres
0 rows affected.

1 %%sql
2 \copy survivant from '/Users/Asus/Documents/survivant.txt'

* postgresql://user1:***@localhost/postgres
0 rows affected.
```

```

1 %%sql
2 SELECT * from ticket;

```

* postgresql://user1:***@localhost/postgres
891 rows affected.

	name	pclass	ticket	fare	cabin	embarked
	Braund, Mr. Owen Harris	3	A/5 21171	7.25		S
	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1	PC 17599	71.2833	C85	C
	Heikkinen, Miss. Laina	3	STON/O2. 3101282	7.925		S
	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	113803	53.1	C123	S
	Allen, Mr. William Henry	3	373450	8.05		S
	Moran, Mr. James	3	330877	8.4583		Q
	McCarthy, Mr. Timothy J	1	17463	51.8625	E46	S
	Palsson, Master. Gosta Leonard	3	349909	21.075		S
	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	3	347742	11.1333		S
	Nasser, Mrs. Nicholas (Adele Achem)	2	237736	30.0708		C

```

1 %%sql
2 SELECT * from passenger;

```

* postgresql://user1:***@localhost/postgres
891 rows affected.

passengerid	name	sex	age	sibsp	parch
1	Braund, Mr. Owen Harris	male	22.0	1	0
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0
3	Heikkinen, Miss. Laina	female	26.0	0	0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0
5	Allen, Mr. William Henry	male	35.0	0	0
6	Moran, Mr. James	male	0.0	0	0
7	McCarthy, Mr. Timothy J	male	54.0	0	0
8	Palsson, Master. Gosta Leonard	male	2.0	3	1
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2
10	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0

```

1 %%sql
2 SELECT * from survivant;

```

* postgresql://user1:***@localhost/postgres
891 rows affected.

passengerid	name	survived
1	Braund, Mr. Owen Harris	0
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1
3	Heikkinen, Miss. Laina	1
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1
5	Allen, Mr. William Henry	0
6	Moran, Mr. James	0
7	McCarthy, Mr. Timothy J	0
8	Palsson, Master. Gosta Leonard	0
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	1
10	Nasser, Mrs. Nicholas (Adele Achem)	1

2) Requêtes imposées

- Combien de classes de passagers différentes y avait-il à bord du Titanic ?

```
1 %%sql
2 select count(*) as classes_de_passagers
3     from (select distinct pclass from ticket) as liste_classes;
```

* postgresql://user1:***@localhost/postgres
1 rows affected.

classes_de_passagers
3

- Combien de passagers y avait-il dans chaque classe?

```
1 %%sql
2 select pclass, count(*) as passagers
3     from ticket
4     group by pclass
5     order by pclass;
```

* postgresql://user1:***@localhost/postgres
3 rows affected.

pclass	passagers
1	216
2	184
3	491

- Combien de femmes et d'hommes y avait-il dans chaque classe?

```
1 %%sql
2 select ticket.pclass, passenger.sex, count(*) as passagers
3     from ticket natural join passenger
4     group by ticket.pclass, passenger.sex
5     order by ticket.pclass, passenger.sex;
```

* postgresql://user1:***@localhost/postgres
6 rows affected.

pclass	sex	passagers
1	female	94
1	male	122
2	female	76
2	male	108
3	female	144
3	male	347

- Comptez le nombre et le pourcentage de survivants et de passagers morts.

```
1 %%sql
2 select survivant.survived, count(*) as passagers, (count(*) * 100 / (select count(*) from survivant)) as pourcentage
3     from survivant
4     group by survived;
```

* postgresql://user1:***@localhost/postgres
2 rows affected.

survived	passagers	pourcentage
0	549	61
1	342	38

- Visualiser la répartition des passagers survivants et morts par classe

```
1 %%sql
2 select ticket.pclass, survivant.survived, count(*) as passagers
3     from ticket natural join survivant
4     group by pclass, survived
5     order by pclass, survived;
```

* postgresql://user1:***@localhost/postgres
6 rows affected.

pclass	survived	passagers
1	0	80
1	1	136
2	0	97
2	1	87
3	0	372
3	1	119

- Visualiser la répartition des passagers survivants et des passagers décédés selon le sexe

```
1 %%sql
2 select survivant.survived, passenger.sex, count(*) as passagers
3     from passenger natural join survivant
4     group by survived, sex
5     order by survived;
```

* postgresql://user1:***@localhost/postgres
4 rows affected.

survived	sex	passagers
0	female	81
0	male	468
1	female	233
1	male	109

3) Requêtes inventés

- Donnez l'identifiant et le nom de tous les passagers survivants.

```
1 %%sql
2 select passengerid,name
3   from survivant
4  where survived=1;
```

* postgresql://user1:***@localhost/postgres
342 rows affected.

passengerid	name
2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3	Heikkinen, Miss. Laina
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
10	Nasser, Mrs. Nicholas (Adele Achem)
11	Sandstrom, Miss. Marguerite Rut
12	Bonnell, Miss. Elizabeth
16	Hewlett, Mrs. (Mary D Kingcome)
18	Williams, Mr. Charles Eugene
20	Masselmani, Mrs. Fatima

- Donnez le nom, le ticket et l'âge des hommes âgés de plus ou égal 50 ans qui ont fait embarcation depuis le port de Cherbourg (C).

```
1 %%sql
2 select ticket.name,ticket.ticket, passenger.age
3   from ticket natural join passenger
4  where ticket.embarked='C' and passenger.age>=50 and passenger.sex='male';
```

* postgresql://user1:***@localhost/postgres
10 rows affected.

name	ticket	age
Ostby, Mr. Engelhart Cornelius	113509	65.0
Goldschmidt, Mr. George B	PC 17754	71.0
Williams, Mr. Charles Duane	PC 17597	51.0
Smith, Mr. James Clinch	17764	56.0
Kent, Mr. Edward Austin	11771	58.0
Artagaveytia, Mr. Ramon	PC 17609	71.0
Douglas, Mr. Walter Donald	PC 17761	50.0
Frolicher-Stehli, Mr. Maxmillian	13567	60.0
Simonius-Blumer, Col. Oberst Alfons	13213	56.0
Newell, Mr. Arthur Webster	35273	58.0

- Donnez l'âge du passager le plus âgé.

```
1 %%sql
2 select max(age) as age
3     from passager;
```

* postgresql://user1:***@localhost/postgres
1 rows affected.

age

80.0

- Combien y'avait-il de mineurs parmi les passagers (strictement moins de 18 ans) ?

```
1 %%sql
2 --Les passagers dont l'age est inconnu ont un age de 0.0
3 select count(*) as mineurs
4     from passager
5     where age>0.0 and age<18;
```

* postgresql://user1:***@localhost/postgres
1 rows affected.

mineurs

113

- Combien y'avait-il de femmes parmi les passagers ?

```
1 %%sql
2 select sex, count(*) as passagers
3     from passager
4     where sex='female'
5     group by sex;
```

* postgresql://user1:***@localhost/postgres
1 rows affected.

sex	passagers
-----	-----------

female	314
--------	-----

- Combien y'avait-il de survivants au naufrage du Titanic ?

```
1 %%sql
2 select survived, count(*) as passagers
3     from survivant
4     where survived=1
5     group by survived;
```

```
* postgresql://user1:***@localhost/postgres
1 rows affected.
```

survived	passagers
1	342

- Combien y'avait-il de femmes mineures qui ont survécus au naufrage ?

```
1 %%sql
2 select count(*) as mineures_survivantes
3     from passager natural join ticket natural join survivant
4     where survived=1 and sex='female' and age<18;
```

```
* postgresql://user1:***@localhost/postgres
1 rows affected.
```

mineures_survivantes
74

- A combien s'élève les profits réalisés à partir de la vente des tickets pour le voyage à bord du Titanic ?

```
1 %%sql
2 select sum(fare) as benefices
3     from ticket;
```

```
* postgresql://user1:***@localhost/postgres
1 rows affected.
```

benefices
28693.9493

4) Commandes de réinitialisations

```
1 %%sql
2 delete from survivant;
3 delete from passager;
4 delete from ticket;
```

```
1 %%sql
2 drop table if exists survivant;
3 drop table if exists passager;
4 drop table if exists ticket;
```

```
* postgresql://user1:***@localhost/postgres
Done.
Done.
Done.
```