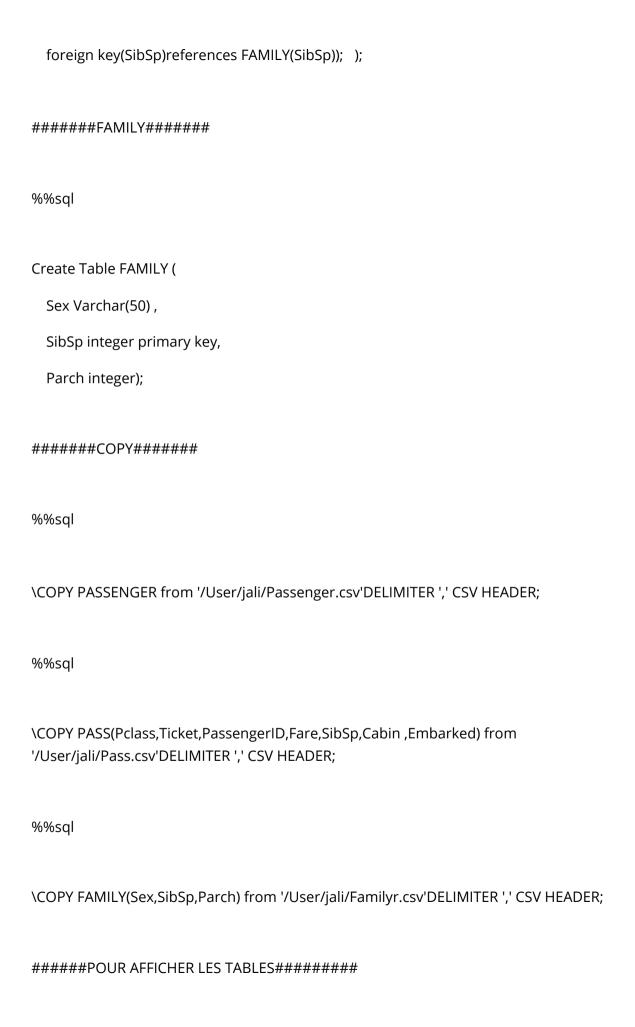
NOMS ET PRENOMS: JAVED ALI HASSAN , ILYES ZEROUAL

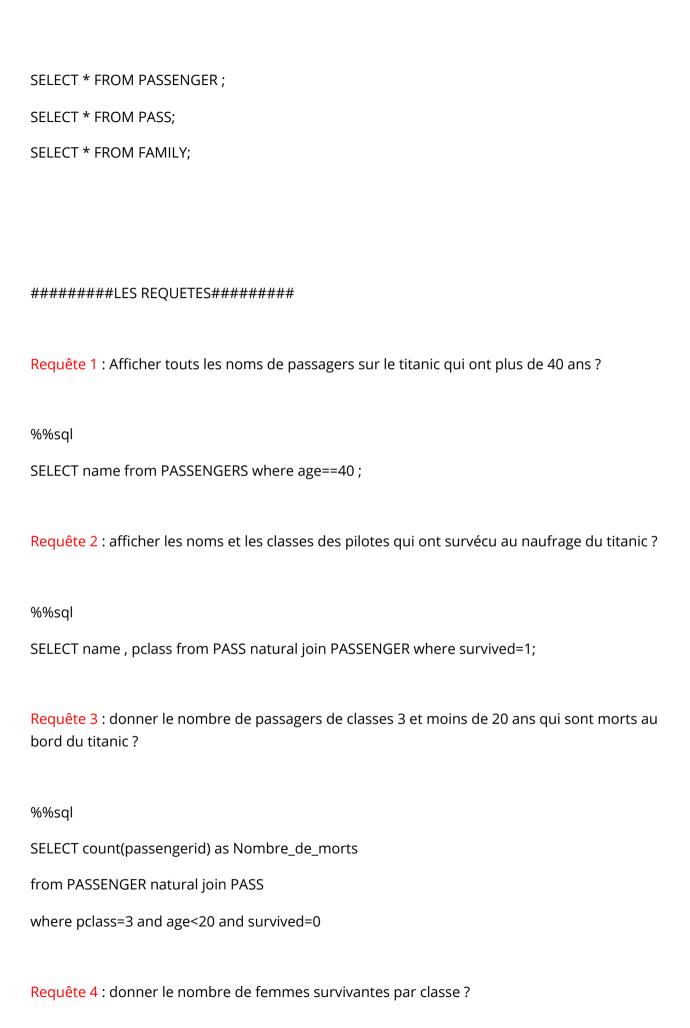
SAE base de données : Phase 3
1) Création des tables :
<u>a) RAPPORT :</u>
On a créé 3 Tables : PASSENGER , PASS , FAMILY .
La 1ére Table:
1) Table Passengers :
- PassengerID
- Survived
- Name
- Sex
- Age
 Passengerld est la clé primaire de la table PASSENGERS car il représente chaque passage avec un identifiant différent.
Name : Nom et prénom avec(Mr/Mrs)
Age : C'est l'age de la personne à bord du navire.

Sex : Permet de savoir si le passager est un homme ou une femme.
 Survived : qui nous donne des informations à propos du passager si il est vivant ou bien mort.
2) La 2éme Table
2) Table PASS:
- Passengerld
- Pclass
- Ticket
- SibSp
- Fare
- Cabin
- Embarked
Pclass : nous informe de la classe de voyage du passager
PassengerID : qui est une clé étrangère car c'est la clé primaire de la table PASSENGER .
• Ticket : le Billet
• Fare : le prix du billet .

Cabine : le numéro de cabine de la personne à bord (passager).
Embarked : les destinations du Titanic .
La 3éme TABLE
3) Table FAMILY:
- Parch
- SibSp
- Sex
• Sibsp :le nombre de frères et sœurs ou de conjoints d'une personne à bord .
• Parch : le nombre de parents et enfants d'une personne à bord .
• Sex : C'est une clé étrangère car c'est la clé primaire de la table PASSENGER .
! pip install ipython-sql
%Load_ext sql
%sql sqlite://
%sql postgresql://postgres:@localhost/postgres

```
*********CRÉATION DES TABLES******
#######PASSENGER#######
%%sql
CREATE TABLE PASSENGER(
 PassengerID integer primary key,
 Survived integer,
 Name varchar(50),
 Sex varchar(50),
 Age integer);
#######PASS#######
%%sql
create Table PASS (
  Pclass integer,
 Ticket integer
 PassengerID integer,
  Fare integer,
  SibSp varchar(200),
  Cabin Varchar(50),
 Embarked varchar(50),
 foreign key(PassengerID)references PASSENGER(PassengerID),
```





```
%%sql
SELECT pclass, count(passengerid) as Survivantes from PASSENGER
natural join PASS where sex='female' and survived=1
group by pclass
order by pclass;
Requête 5 : Donner la somme des prix des billets selon les classes ?
%%sql
SELECT pclass ,SUM(fare) as MONEY
from PASS
group by pclass
order by pclass;
Requête 6 : Donner pour chaque classe le prix du billet le moins chère ?
%%sql
select pclass, MIN(fare) as billet_le_moins_chère
from PASS
group by pclass
order by pclass;
Requête 7 : Quel est l'âge du plus petit passager mort lors du naufrage?
%%sql
```

select MIN (age) as age_plus_petit from PASSENGER

Requête 8 : Citez par ordre croissant l'âge des personnes vivantes au bord du navire ?
%%sql
SELECT age, name from PASSENGER
where survived=1
order by age ASC;
Requête 9 : Donner les numéros et noms des homme âgées de moins de 25 ans qui leurs noms contiennent 'Mi ' ?
%%sql
SELECT passengerid , name
from PASSENGER
where age<25 and sex='male' and name LIKE '%Mi%';
######################################
1/ combien de classes différentes y'avait-il au bord du Titanic ?
%%sql
SELECT count (distinct pclass) as nb_classes
From PASS;

Where survived=0

```
1/combien de classes différentes y'avait il au bord du titanic:

Entrée [19]:

$ELECT count(distinct pclass) as nb_classes
from PASS;

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

Out[19]:

nb_classes

3
```

2/Combien de passagers y'avait-il dans chaque classe?

%%sql

SELECT pclass, count(passengerid)

from PASS

group by pclass;

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

%%sql

SELECT pclass, sex, count(passengerid) nombre_passagers

from PASS natural join PASSENGER

group by pclass, sex;

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

```
Entrée [52]:

**SELECT polass,sex,count(passengerid) nombre_passagers
from PASSENGER natural join PASS
group by polass,sex

* postgresq1://postgres:***@localhost/postgres
6 rows affected.

Out[52]:

polass sex nombre_passagers

1 female 94

1 male 122

2 female 76

2 male 108

3 female 144

3 male 346
```

4/Comptez le nombre et le pourcentage de survivants et de passagers morts?

%%sql

SELECT survived,count(*) as nombre_de_passagers,count(*)*100 / (SELECT count(*) from PASSENGER) as pourcentage

from PASS

group by survived;

5/ Visualiser la répartition des passagers survivants et morts par classe?

%%sql

SELECT pclass, count (*) as nb_survivants

from PASS natural join PASSENGER

```
where survived=1
group by pclass
order by pclass;

%%sql

SELECT pclass, count (*) as nb_morts
from reservations natural join status
where survived=0
```

group by pclass

order by pclass

```
Entrée [110]: **sql
SELECT polass,count(*) as nb_survivants
from PASS natural join PASSENGER
where survived*-1
group by polass
order by polass;

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

Out[110]: polass nb_survivants

1 134
2 87
3 119

Entrée [112]: **sql
SELECT polass,count(*) as nb_morts
from PASS natural join PASSENGER
where survived*-0
group by polass
order by polass

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

Out[112]: polas nb_morts

1 80
2 97
3 372
```

6/ Visualiser la répartition des passagers survivants et des passagers décédés selon le sexe ?

SELECT sex, count (*) as nb_survivants

from PASSENGER

where survived=1

group by sex;

%%sql

SELECT sex, count (*) as nb_morts

from PASSENGER

where survived=0

group by sex;

```
6/Visualiser la répartition des passagers survivants et des passagers décédés selon le sexe

Entrée [23]: %%sql

SELECT sex,count(*) as nb_survivants
from PASSENGER

* sqlite://
(sqlites).operationalError) no such table: passengers
[SQL: SELECT sex,count(*) as nb_survivants
from passengers natural join status
where survived=1
group by sex;]
(Background on this error at: https://sqlalche.me/e/14/e3q8)

Entrée [117]: %%sql

SELECT sex, count(*) as nb_morts
from PASSENGER
where survived=0
group by sex;

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

Out[117]: sex nb_morts
female 64
male 359
```

