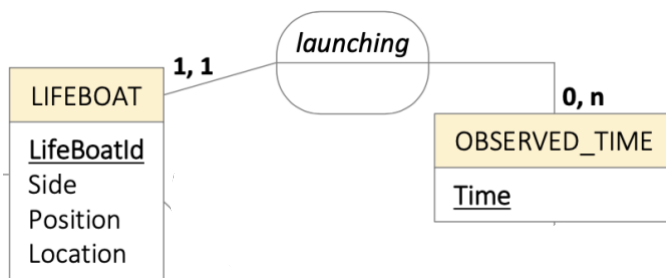


Étape 2 - SAE 1.04

Table des matières

1	<i>Justification du SLR</i>	3
2	<i>Implémentation sous PostgreSQL:</i>	8
2.1	Expression des contraintes :	8
3	<i>Peuplement</i>	10
4	<i>Requêtes</i>	11

1 Justification du SLR



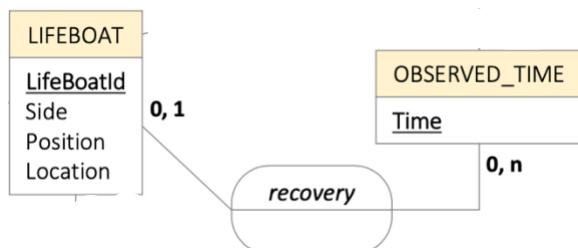
Dans *Observed_time* :

- Time provient de l'application de la règle **R0** à l'entité Observed_Time et Time est clé primaire parce qu'il représente l'unique identifiant de cette entité.

Dans *LifeBoat* :

- LifeBoatId, Side, Position et Location proviennent de l'application de la règle **R0** à l'entité LifeBoat et LifeBoatId est clé primaire parce qu'il représente l'identifiant de cette entité
- Time provient de l'application de la règle **R1** à l'association launching qui associe une et une seule occurrence de l'entité Observed_Time à une occurrence de l'entité LifeBoat (cardinalités 1,1).
- Time est ajouté en tant que clé étrangère

LIFEBOAT (LifeBoatId, Side, Position, Location, #Launching_Time)
OBSERVED_TIME (TIME)



Dans *Observed_time* :

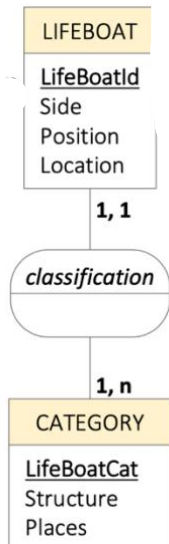
- Time provient de l'application de la règle **R0** à l'entité Observed_Time et Time est clé primaire parce qu'il représente l'unique identifiant de cette entité.

Dans *LifeBoat* :

- LifeBoatId, Side, Position et Location proviennent de l'application de la règle **R0** à l'entité LifeBoat et LifeBoatId est clé primaire parce qu'il représente l'identifiant de cette entité
- Recovery devient une nouvelle relation par application de la règle R2 car une de ces pattes a pour cardinalités 0,1.
- Time et LifeBoatId provient de l'application de la règle **R2** à l'association recovery.

- LifeBoatId est clé primaire, il est identifiant de l'entité associée à cette patte 0,1.
Time (renommé Recovery_Time) deviant clé étrangère car les identifiants des autres entités participant à cette association, sont clés étrangères selon la **R2**.

RECOVERY (#LifeBoatId, #Recovery_Time)



Dans *Category* :

- LifeBoatCat, Structure et Places provient de l'application de la règle **R0** à l'entité Observed_Time et LifeBoatCat est clé primaire parce qu'il représente l'identifiant de cette entité.

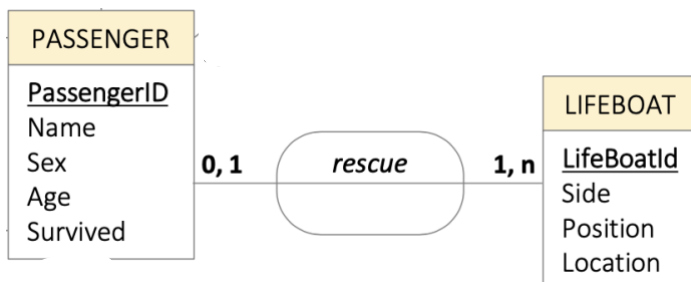
Dans *LifeBoat* :

- LifeBoatId, Side, Position et Location proviennent de l'application de la règle **R0** à l'entité LifeBoat et LifeBoatId est clé primaire parce qu'il représente l'identifiant de cette entité

- LifeBoatCat devient clé étrangère dans l'entité LifeBoat après l'application de la **R1** car sa patte a pour cardinalités 1,1.

LIFEBOAT (LifeBoatId, #LifeBoatCat, Side, Position, Location, #Lauching_Time)

CATEGORY (LifeBoatCat, Structure, Places)



Dans *LifeBoat* :

- LifeBoatId, Side et Position et Location proviennent de l'application de la règle **R0** à l'entité LifeBoat et LifeBoatId est clé primaire parce qu'il représente l'identifiant de cette entité.

Dans *Passenger* :

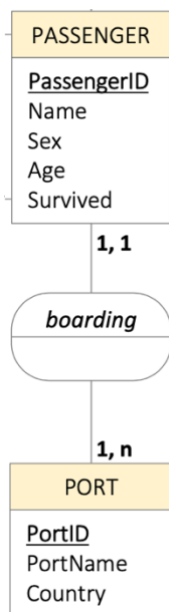
- PassengerId, Name, Sex, Age, Survived proviennent de l'application de la règle **R0** à l'entité Passenger et PassengerId est clé primaire parce qu'il représente l'identifiant de cette entité.

- Rescue devient une nouvelle relation car on sait qu'une association dont une des pattes a pour cardinalités 0,1 devient une nouvelle relation par application de la règle **R2**.

- La clé primaire est l'identifiant de l'entité associée à cette patte, donc PassengerId devient clé primaire et étrangère dans la relation Rescue. (**R2**)

- Les identifiants des autres entités participant à cette association, en tant que clés étrangères (**R2**). LifeBoatId est ajouté à Rescue en tant que clé étrangère.

RESCUE (#PassengerId, #LifeBoatId)



Dans Passenger :

- PassengerId, Name, Sex, Age, Survived proviennent de l'application de la règle **R0** à l'entité Passenger et PassengerId est clé primaire parce qu'il représente l'identifiant de cette entité.

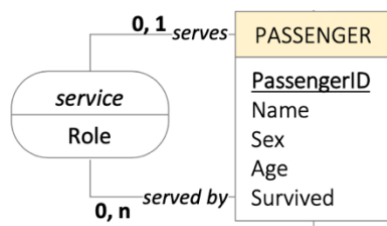
Dans Port :

- PortId, PortName et Country proviennent de l'application de la règle **R0** à l'entité Port et PortId est clé primaire parce qu'il représente l'identifiant de cette entité.

- L'une des pattes de l'association boarding à une cardinalité de 1,1 donc on applique la règle **R1**. Donc, les identifiants des autres entités participant à l'association deviennent clés étrangères dans la relation complétée. On a donc PortId qui devient clé étrangère dans Passenger.

PASSENGER (PassengerId, Name, Sex, Age, Survived, #PortId)

PORT (PortId, PortName, Country)



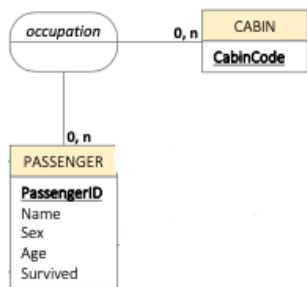
Dans Passenger :

- PassengerId, Name, Sex, Age, Survived proviennent de l'application de la règle **R0** à l'entité Passenger et PassengerId est clé primaire parce qu'il représente l'identifiant de cette entité.

Puisqu'une des pattes de l'association porte la cardinalité 0,1, Service devient une nouvelle relation d'après l'application de la **R2**.

- PassengerId rentre donc 2 fois dans la relation, par souci de compréhension, on le nommera une fois PassengerId_Dom et l'autre PassengerId_Emp. Ces deux attributs viendront compléter Role et ils seront clés étrangères.
- En application la règle **R2**, PassengerId_Dom sera clé primaire car il porte la cardinalité 0,1.

SERVICE (#PassengerId_Dom, #PassengerId_Emp, Role)



Dans Passenger :

- PassengerId, Name, Sex, Age, Survived proviennent de l'application de la règle **R0** à l'entité Passenger et PassengerId est clé primaire parce qu'il représente l'identifiant de cette entité.

Dans Cabin ;

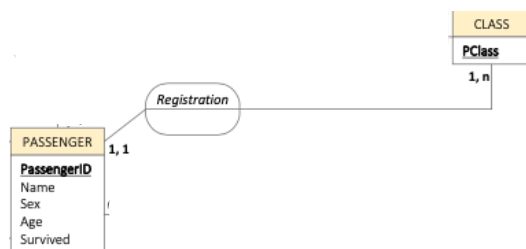
- CabinCode provient de l'application de la règle **R0** à l'entité

Cabin et CabinCode est clé primaire parce qu'il représente l'identifiant de cette entité.

- Les deux pattes de l'association Occupation ont des cardinalités de 0, n. On applique donc la règle **R3**. L'association Occupation devient donc une nouvelle relation où PassengerId ainsi que CabinCode s'ajoute en tant que clé primaire et étrangère.

OCCUPATION (#PassengerId, #CabinCode)

CABIN (CabinCode)



Dans Passenger :

- PassengerId, Name, Sex, Age, Survived proviennent de l'application de la règle **R0** à l'entité Passenger et PassengerId est clé primaire parce qu'il représente l'identifiant de cette entité.

Dans Class ;

- PClass provient de l'application de la règle **R0** à l'entité Class et PClass est clé primaire parce qu'il représente l'identifiant de cette entité.

- Une des pattes de l'association Registration a pour cardinalité de 1,1. On applique donc la règle **R1**. On ajoute donc la clé primaire de l'entité Class en tant que clé étrangère dans l'entité Passenger.

PASSENGER(PassengerId, Name, Sex, Age, Survived, #PClass, #PortId)

D'après le cours :

Cas d'une relation réduite à un seul attribut

Si cet attribut est déjà clé étrangère dans une autre relation, suite à l'application d'une des règles R1, R2 ou R3 :

Il peut être décidé de ne pas conserver cette relation

Donc on a:

PORT (PortId, PortName, Country)

PASSENGER (PassengerId, Name, Sex, Age, Survived, #PortId, #PClass)

OCCUPATION (#PassengerId, #CabinCode)

SERVICE (#PassengerId Dom, #PassengerId_Emp, Role)

CATEGORY (LifeBoatCat, Structure, Places)

LIFEBOAT (LifeBoatId, #LifeBoatCat, Side, Position, Location, #Lauching_Time)

RECOVERY (#LifeBoatId, #Recovery_Time)

RESCUE (#PassengerId, #LifeBoatId)

~~CABIN (CabinCode)~~

~~OBSERVED_TIME (TIME)~~

~~CLASS (PClass)~~

Ce qui donne finalement :

PORT (PortId, PortName, Country)

PASSENGER (PassengerId, Name, Sex, Age, Survived, #PortId, PClass)

OCCUPATION (#PassengerId, #CabinCode)

SERVICE (#PassengerId Dom, #PassengerId_Emp, Role)

CATEGORY (LifeBoatCat, Structure, Places)

LIFEBOAT (LifeBoatId, #LifeBoatCat, Side, Position, Location, Lauching_Time)

RECOVERY (#LifeBoatId, Recovery_Time)

RESCUE (#PassengerId, #LifeBoatId)

2 Implémentation sous PostgreSQL:

2.1 Expression des contraintes :

Port ->

- PortId char(1) **Contrainte d'attribut** primary key **Contrainte de CHECK** (PortId in ('C', 'Q', 'S'))
- PortName varchar not null,
- Country varchar not null

Passenger ->

- PassengerId int **Contrainte d'attribut** primary key,
- Name varchar not null,
- Sex varchar,
- Age int,
- Survived int **Contrainte de CHECK** (Survived in (0, 1)),
- PClass int not null **Contrainte de CHECK** (PClass in (1, 2, 3)),
- PortId char(1) **Contrainte de CHECK** (PortId in ('C', 'Q', 'S')) **Contrainte de relation** REFERENCES PORT(PortId)

Occupation ->

- PassengerId int **Contrainte de relation** REFERENCES PASSENGER(PassengerId),
- CabinCode varchar,
- primary key (PassengerId, CabinCode)

Service ->

- PassengerId_Dom int **Contrainte d'attribut** primary key **Contrainte de relation** REFERENCE PASSENGER(PassengerId),
- PassengerId_Emp int not null **Contrainte de relation** REFERENCES PASSENGER(PassengerId),
- Role varchar not null

Category ->

- LifeBoatCat varchar **Contrainte d'attribut** primary key **Contrainte de CHECK** (LifeBoatCat in ('standard', 'secours', 'radeau')),
- Structure varchar not null **Contrainte de CHECK** (Structure in ('bois', 'bois et toile')),
- Places int not null

LifeBoat ->

- LifeBoatId varchar **Contrainte d'attribut** primary key,
- LifeBoatCat varchar **Contrainte de relation** REFERENCES CATEGORY(LifeBoatCat),
- Side varchar not null **Contrainte de CHECK** (Side in ('babord', 'tribord')),
- Position varchar not null **Contrainte de CHECK** (Position in ('avant', 'arriere')),
- Location varchar not null DEFAULT 'pont',
- Lauching_Time Time not null

Recovery ->

- LifeBoatId varchar **Contrainte de relation** REFERENCES LIFEBOAT(LifeBoatId),
- Recovery_Time Time not null

Rescue ->

- PassengerId int **Contrainte de relation** REFERENCES PASSENGER(PassengerId),
- LifeBoatId varchar not null **Contrainte de relation** REFERENCES LIFEBOAT(LifeBoatId)

3 Peuplement

Aucune erreur

```
guilbava=> \i peuplement.sql
COPY 3
COPY 1309
COPY 349
COPY 40
COPY 3
COPY 20
COPY 17
COPY 490
guilbava=> □
```

4 Requêtes

Dans le fichier **testinfos_guilbaua.sql**, nous avons écrit trois requêtes répondant à :

- Les informations relatives au passager n°916 : son nom, son âge, sa classe, sa ou ses cabines, le nom du port où il a embarqué, numéro et catégorie de l'embarcation de sauvetage qui l'a éventuellement secouru
- Si le nom et le rôle des domestiques du passager n°1264 est exact
- Si la liste des passagers ayant été secourus par le canot n°7 est exacte