

Naufrage du Titanic

BARBIERI Hugo, GIBELLO Grégoire

Groupe D1

12 décembre 2023

Table des matières

1. Justification et implémentation du SLR.....	3
1.1. Justification.....	3
1.2. Implémentation.....	4
1.2.1. Contraintes.....	4
2. Test de conformité.....	6
3. Peuplement.....	6
4. Premières requêtes.....	7

1. Justification et implémentation du SLR

1.1. Justification

Dans l'association Passenger :

- PassengerId, Name, Sex, Age et 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.
- PClass provient de l'application de la règle R1 à l'association Registration qui associe une et une seule occurrence de l'entité CLASS à une occurrence de l'entité PASSENGER (cardinalités 1,1)
- PClass est clé étrangère en tant qu'identifiant de CLASS
- PortId provient de l'application de la règle R1 à l'association Boarding qui associe une et une seule occurrence de l'entité PORT à une occurrence de l'entité PASSENGER (cardinalités 1,1)
- PortId est clé étrangère en tant qu'identifiant de PORT

Dans l'association 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 provient de l'application de la règle R1 à l'association classification qui associe une et une seule occurrence de l'entité CATEGORY à une occurrence de l'entité LIFEBOAT (cardinalités 1,1)
- LifeBoatCat est clé étrangère en tant qu'identifiant de CATEGORY
- Launching_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)
- Launching_Time est clé étrangère en tant qu'identifiant de OBSERVED_TIME

Dans l'association Category :

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

Dans l'association 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.

Dans l'association Classe :

- 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.

Dans l'association Cabine :

- 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.

Dans l'association 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'identifiant de cette entité.

Dans l'association Service :

- L'association SERVICE devient une entité dont les attributs sont Role et PassengerId, et dont la clé primaire et étrangère est PassengerId, provenant de l'entité PASSENGER, en application de la règle R2. (Cardinalités 0,1)
- Pour des soucis de compréhension PassengerId se divise en deux attributs : PassengerId_Dom (clé primaire) et PassengerId_Emp.

Dans l'association Rescue :

- L'association RESCUE devient une entité dont la clé primaire et étrangère est PassengerId, provenant de l'entité PASSENGER, et LifeBoatId est clé étrangère en tant qu'identifiant de LIFEBOAT, en application de la règle R2. (Cardinalités 0,1)

Dans l'association Recovery :

- L'association RECOVERY devient une entité dont la clé primaire et étrangère est LifeBoatId, provenant de l'entité LIFEBOAT, et Recovery_Time est clé étrangère en tant qu'identifiant de TIME, en application de la règle R2. (Cardinalités 0,1)

Dans l'association Occupation :

- L'association OCCUPATION devient une entité dont les clés primaires et étrangères sont le couple PassengerId et CabinCode respectivement issues de l'entité PASSENGER et CABIN, ce, en application de la règle R. (cardinalités 0,1)

1.2. Implémentation

1.2.1. Contraintes

Dans la relation Port :

- *PortId* prend pour valeur la première lettre du nom (c'est donc un caractère char(1)) du port de départ, étant seulement Southampton en Angleterre, Cherbourg en Irlande et Queenstown au Etats-Unis, c'est aussi la clé primaire car il ne peut pas avoir deux ports identiques.
- *PortName* doit forcément être renseigné car un port a obligatoirement un nom. C'est une chaîne de texte donc varchar.
- *Country* doit forcément être renseigné car un port est obligatoirement dans un pays. C'est une chaîne de texte donc varchar.

Dans la relation **Passenger** :

- *PassengerId* est la clé primaire de *Passenger* car un identifiant est forcément unique. L'identifiant est un nombre (int).
- *Name* doit être obligatoirement renseigné car un passager à forcément un nom. C'est une chaîne de texte donc varchar.
- *Sex* doit être obligatoirement renseigné car un passager à forcément un sexe. C'est une chaîne de texte donc varchar.
- *Age* est un nombre (int). Il n'est pas forcément renseigné car on peut ne pas le connaître s'il n'a pas été renseigné à l'achat du billet.
- *Survived* est un nombre (int) qui ne peut avoir que deux valeurs possibles soit 0 si la personne n'a pas survécu ou alors 1 si la personne a survécu.
- *PClass* est un nombre compris entre 1 et 3 et qui correspondent au 3 classe de billets/chambre qu'il était possible d'acheter. Celui-ci doit être renseigné car un passager avait forcément un billet qui lui-même avait forcément une classe attribuée.
- *PortId* référence *Port(PortId)* il possède donc les mêmes contraintes que celui-ci.

Dans la relation **Occupation** :

- *PassengerId* référence *Passenger(PassengerId)* de ce fait il possède les mêmes contraintes que celui-ci.
- *CabinCode* est une chaîne de caractère (varchar).
- Le couple *PassengerId* et *CabinCode* forment ensemble la clé primaire de la table ils sont donc uniques et non vides.

Dans la relation **Service** :

- *PassengerId_Dom* référence *Passenger(PassengerId)* et en possède donc toutes les contraintes car un domestique fait partis des passagers du bateau. Il est aussi la clé primaire de la relation qu'un même domestique ne peut pas exister deux fois.
- *PassengerId_Emp* référence *Passenger(PassengerId)* et en possède donc toutes les contraintes car un employeur fait partis des passagers du bateau.
- *Rôle* est une chaîne de caractère (varchar) qui décrit quel est le métier du domestique, il doit donc forcément être renseigné.

Dans la relation **Category** :

- *LifeBoatCat* représente la catégorie du bateau, il n'en existe seulement 3 (standard, secours ou radeau). C'est donc une chaîne de caractère (varchar). C'est la clé primaire car il ne peut y avoir deux fois la même catégorie.
- *Structure* représente la structure du bateau, il n'en existe seulement 2 (bois ou bois et toile). C'est donc une chaîne de caractère (varchar) qui doit être forcément renseigné car un bateau à toujours une structure.
- *Places* représente le nombre (int) de places que le bateau peut contenir, il est forcément renseigné car un bateau à toujours une capacité maximum.

Dans la relation LifeBoat :

- *LifeBoatId* est la clé primaire de la table LifeBoat, il est donc unique et sous forme de chaîne de caractère.
- *LifeBoatCat* référence *Category(LifeBoatCat)* il possède donc les mêmes contraintes que celui-ci.
- *Side* ne peut posséder que deux valeur (bâbord ou tribord) et doit forcément être renseigné car le LifeBoat est obligatoirement à un endroit précis dans le Titanic.
- *Position* ne peut posséder que deux valeur (avant ou arrière) et doit forcément être renseigné car le bateau est obligatoirement à un endroit précis dans le Titanic.
- *Location* précise dans quelle partie du Titanic le LifeBoat est stocké, il est donc forcément renseigné et prend par défaut la valeur « pont ».

Dans la relation Recovery :

- *LifeBoatId* référence *LifeBoat(LifeBoatId)* il hérite de toutes les contraintes de celui-ci. Il en est aussi la clé primaire.
- *Recovery_Time* est une heure (Time) est doit forcément être renseigné car le LifeBoat à forcément été secouru à une heure précise.

Dans la relation Rescue :

- *PassengerId* référence *Passenger(PassengerId)* il en hérite toutes les contraintes. Il est aussi clé primaire.
- *LifeBoatId* référence *LifeBoat(LifeBoatId)* il en hérite toutes les contraintes. Il est aussi clé primaire.

2. Test de conformité

- Voir le fichier testconfo_barbiehu.sql.

3. Peuplement

Requête :

```
SELECT 'PORT' AS Relation, COUNT(*) AS Nombre_de_lignes FROM PORT
UNION ALL
SELECT 'PASSENGER' AS Relation, COUNT(*) AS Nombre_de_lignes FROM PASSENGER
UNION ALL
SELECT 'OCCUPATION' AS Relation, COUNT(*) AS Nombre_de_lignes FROM OCCUPATION
UNION ALL
SELECT 'SERVICE' AS Relation, COUNT(*) AS Nombre_de_lignes FROM SERVICE
UNION ALL
SELECT 'CATEGORY' AS Relation, COUNT(*) AS Nombre_de_lignes FROM CATEGORY
UNION ALL
SELECT 'LIFEBOAT' AS Relation, COUNT(*) AS Nombre_de_lignes FROM LIFEBOAT
UNION ALL
```

```
SELECT 'RECOVERY' AS Relation, COUNT(*) AS Nombre_de_lignes FROM RECOVERY
UNION ALL
SELECT 'RESCUE' AS Relation, COUNT(*) AS Nombre_de_lignes FROM RESCUE;
```

- Utilisation de UNION ALL au lieu de UNION pour éviter que Postgresql élimine les doublons s'il devait en avoir.
- Normalement il ne devrait pas y avoir de doublons mais nous ne pouvons pas connaître ce qui rentre dans la table avant d'avoir exécuter cette commande donc par précaution il vaut mieux utiliser UNION ALL.

Résultat :

relation	nombre_de_lignes
RESCUE	490
LIFEBOAT	20
OCCUPATION	349
PORT	3
CATEGORY	3
SERVICE	40
PASSENGER	1309
RECOVERY	17
(8 rows)	

4. Premières requêtes

- Voir le fichier testinfos_barbiehu.sql.