

Groupe : andromeda	Lilia Mahrez Laetitia Idier Tinhinane Salemkour	IUT De Villeteuse
--------------------	---	-------------------

Phase 1 :

- 1) Intégration de Postgresql sur jupyter notebook en python c'est bien fait en utilisant une BDD Postgresql en ligne pour pouvoir l'utiliser dans les PC de l'IUT afin d'avoir les mêmes données chez moi sur mon PC personnel.
- 2) Degré =12
- 3) Cardinalité=892
- 4) Attributs=12
- 5) Le fichier ***titanic_train.csv*** contient des informations des personnes qui étaient à bord du titanic, la première ligne du fichier contient des attributs qui sont les suivant :

- *PassengerId* : Un nombre qui permet d'identifier le passager
- *Survived* : 0 = n'a pas survécu, 1 = survécu
- *Pclass* : La classe du ticket : 1 = 1ère classe, 2 = 2ème classe, 3 = 3ème classe
- *Name* : Le nom du passager
- *Sex* : Le genre du passager (male, female ou autre)
- *Age* : L'âge du passager
- *SibSp* : Le nombre des frères et sœurs à bord si présent
- *Parch* : Le nombre des parents et fils à bord si présent
- *Ticket* : Un numéro qui permet d'identifier un ticket
- *Fare* : Le prix du ticket pour la personne
- *Cabin* : Le numéro de la cabine où le passager est hébergé
- *Embarked* : Le lieu où le passager a embarqué

Les autres lignes représentent les valeurs de ces attributs, ceci ressemble à un tableau de passager qui est rempli de plusieurs personnes.

L'objectif de cette analyse est de réfléchir et essayer de constituer un modèle équivalent en base de données où on va constituer une relation passager où aura tous les attributs précédents comme des colonnes.

Phase 2

Description des tables ;

La base de données que nous avons modélisé au premier temps contient tous les attributs précédents, mais nous avons remarqué qu'on pouvait améliorer le schéma de bdd en créant une autre relation ticket qui va contenir que les attributs nécessaires. Donc au final on aura deux tables(relations) passager et ticket, et nous constatons aussi après l'analyse du fichier qu'un ticket est attribué à une seule personne ou passager et un passager peut avoir un ticket qui justifie la cardinalité que l'ont a mise sur le schéma. Mais comme les deux relations sont lié, il faut forcément qu'il ait un lien entre eux afin de bien gérer les données, ceci nous a conduit à créer une relation(tableau) supplémentaire (hasTicket) qui contient les deux attribus passengerId et ticket qui sont les clefs primaires dans les relations passanger et ticket afin de les identifier (ils sont aussi clefs étrangères dans la relation hasTicket car ils font références au clefs primaires des autres relations)

RAPPORT

Nous avons commencé par créer 3 tables qui regroupe tous les attributs qui sont présents dans le fichier titanic_train.csv. On a nommé la première table Passager qui a 7 attributs. La deuxième table surnommé Ticket qui a 5 attributs puis nous avons fait un lien entre la première table et la deuxième table en créant la 3eme table Has ticket qui a elle 2 attributs.

Description des tables :

- Table Passenger : les attributs sont : PassengersId integer (identification du passager Nom varchar(100),Sex varchar(10),Age float , survived integer (1ou 0) , SibSp integer nb de frere et sœur , parch integer(nb de parents et fils a bord). Cette table sert donc a recupéré les informations sur le passager du bateau,nous avons choisi de mettre en clé primaire PassengersId.
- Table Ticket : Les attributs sont : ticket varchar (clé primaire), Pclass integer, Fare float (prix du ticket), Cabin varchar(20), Embarked varchar(1). Cette table permet de récupéré les informations par rapport au Le numéro de la cabine ou le passager est hébergé au ticket donc le numéro de ticket,la classe,le prix et Le lieu où le passer à embarqué
- Table has ticket Les attributs sont : PassengersId integer, ticket varchar(20) et ticket varchar 20 . on a defini comme cle primaire (passengerid et ticket) et en clé etrangere passengerid et ticket car il font references a des cles dans la table passanger et la table ticket

