

EF2 - Bases de données

TP - Base de données SNCM

Le but de ce travail est de créer une base de données pour une partie du système d'information de la SNCM, compagnie maritime de transport vers la Corse (cf. <http://www.sncm.fr/>).

Pour naviguer, la SNCM utilise des navires de type Ferry dont elle n'est pas forcément propriétaire. Ainsi, dans les périodes de pointe, elle peut utiliser des navires fournis par d'autres compagnies.

Un navire peut proposer des commodités (« Cabine avec Sanitaires Privés - 4 lits - avec Hublot », « Cabine avec Sanitaires Privés - 4 lits – intérieure », « Fauteuil confort en salon ou en cabine », etc.) en un certain nombre (par ex, le navire dispose de 100 cabines avec Sanitaires Privés – 4 lits – avec Hublot).

De plus, un navire peut accepter des véhicules à bord, de différentes catégories.

Pour proposer un trajet, la SNCM a également besoin de connaître les informations sur chacun des ports dans lesquels ses navires font escale. Toutes ces informations sont données par les capitaineries des ports concernés.

Dès que les informations sur les ports sont enregistrées, la SNCM peut commencer à saisir les informations sur les trajets. Toute information sur un trajet doit être fixée (elle n'est plus modifiable) trois mois avant le départ du trajet. La SNCM ayant des obligations de services public à respecter, elle doit proposer un nombre minimal de trajets au départ de certains ports et à destination de la Corse (4 départs par semaine au départ de Marseille en haute saison, 1 départ par semaine au départ de Nice en basse saison, ...)

Le schéma relationnel permettant d'enregistrer les informations est donné à la page suivante.

NAVIRE (*nomNavire, modele, nbPlacesMax, nomCompagnie*)

Les informations sur chaque navire sont : un nom permettant de l'identifier, le modèle, le nombre maximal de places pour les passagers et le nom de la compagnie propriétaire.

COMMODITE (*idCom, nbPlaces, description*)

Chaque commodité est identifiée par un code, (« A4E », « A4I », « CONF », etc.), et possède un nombre de places maximum, et une description.

PROPOSE(*nomNavire, idCom, nbCom, prixCom*)

Cette relation indique quelles sont les commodités proposées par chaque navire, en quelle quantité et à quel prix. Une même commodité peut avoir des prix différents sur des navires différents. Le nombre de places maximal d'un bateau est bien évidemment supérieur ou égal au nombre de places totales de toutes ses commodités.

CAT_VEHICULE (*idCat, Libelle*)

Une catégorie de véhicule possède un identifiant (« V », « A », ...), et est caractérisée par un libellé. Le libellé d'une catégorie de véhicules est « Voiture », « Autocar », « Camping-car », « Fourgon », « Moto », « Side-Car et Quad ».

ACCEPTE(*nomNavire, idCat, ndCat, prixCat*)

On enregistre pour chaque navire les catégories de véhicules acceptés, et pour chaque catégorie, le nombre maximal de véhicules acceptés de chaque catégorie ainsi que le prix à payer pour un véhicule de cette catégorie. Comme pour les commodités, une catégorie de véhicules peut ne pas avoir le même prix sur des navires différents.

PORT(*nomPort, Taxe, zone, nbMaxPass*)

Chaque port est identifié par son nom. Pour chacun des ports, on enregistre le montant des taxes portuaires, la zone géographique (par exemple, Corse, Continent, Sardaigne, etc.) et le nombre maximum de passagers que peuvent transporter les ferrys qui y accostent (en fonction des infrastructures présentes dans le port).

TRAJET**(idTrajet, dateHeureDep, dateHeureArr, prixBase, nomNavire, nomPortDep, nomPortArr)**

Chaque trajet est identifié par un numéro, a une date et une heure de départ et une date et une heure d'arrivée. Un trajet part d'un port de départ et arrive à un port de destination. Un trajet a un prix de base correspondant à un prix par passager adulte. Un trajet est effectué par un et un seul navire. Le navire attribué à un trajet doit bien sûr être disponible aux dates et heures du trajet, et une durée minimale de 2 heures doit être respectée entre l'arrivée d'un navire et son départ pour un autre trajet (afin de permettre le débarquement des passagers et l'embarquement des passagers suivants).

Etape 1 – schéma et contraintes d'intégrité
--

- 1) Dans les relations où ce n'est pas fait, soulignez les clés primaires.
- 2) Déterminez les clés étrangères (contraintes d'intégrité référentielles). Vous pouvez les indiquer avec un # sur le schéma mais vous devez aussi les lister de manière explicite sous la forme :
l'attribut A de la relation R référence l'attribut B de la relation S
- 3) Listez les autres contraintes d'intégrité. Indiquez, pour chacune de ces contraintes si elles sont :
 - structurelles (à préciser dès les `create table`)
 - à gérer à l'aide de trigger
 - autre

Etape 2 – mise en place de la base de données sous Postgres

Le but de ce TP est de manipuler le SGBD Postgres que vous avez installé sur vos machines afin d'implémenter la base de données SNCM.

Méthode de travail recommandée pour ce TP et les suivants :

→ Lancez un terminal (*Terminal* ou *Konsole*).

→ Dans votre répertoire racine, créez un répertoire BD (en utilisant par exemple la commande ***mkdir BD***), et placez-vous dans ce répertoire (commande ***cd BD***) et dans ce répertoire, créez un sous-répertoire SNCM.

→ Depuis la page d'accueil de la plateforme Chamilo des cours du département informatique : <http://chamilo3.grenet.fr/upmf/> accédez au module « EF2 – Bases de données »

→ Téléchargez les documents ci-dessous :

- ***create.sql*** : fichier de création des tables, à compléter
- ***drop.sql*** : fichier de suppression des tables,
- ***insert.sql*** : fichier d'insertion des données, à compléter si besoin

→ Veuillez noter que :

- Toutes les requêtes sont terminées par ; .
- ***/* ceci est un commentaire */***
- ***-- ceci également***

Pour plus d'informations sur PostgreSQL et la syntaxe SQL, vous pouvez consulter le fichier **PolyPostgres**, disponible sur la plateforme Chamilo du département ou l'aide en ligne de PostgreSQL : <http://docs.postgresql.fr/>

→ Dans le terminal, connectez-vous à la base ***sncm*** avec le compte ***admin*** par la commande ***psql -U admin sncm***

Mise en place de la base

- 1) Dans un éditeur de texte, ouvrez le fichier ***create.sql*** et complétez les directives de création des tables en ajoutant dans chaque table les clés primaires, les clés étrangères et les contraintes structurelles.

Attention : l'ordre de création et de suppression des relations n'est pas neutre puisque votre schéma comporte des contraintes d'intégrité référentielle.

Sous PostgreSQL, exécutez la commande ***\i create.sql*** pour exécuter l'intégralité du contenu du fichier. Si besoin (par exemple en cas d'erreur ou d'oubli d'une contrainte dans une table), exécutez la commande ***\i drop.sql*** pour supprimer les tables qui ont été créées... et recommencez...

- 2) Sous PostgreSQL, exécutez la commande ***\i insert.sql*** pour insérer les données dans les tables. Vous ne devez pas constater d'erreurs lors de l'insertion de vos données.

Etape 3

Interrogation des données : SELECT

Dans un fichier *select.sql*, écrivez les requêtes SELECT permettant de répondre aux questions ci-dessous. Testez-les au fur et à mesure : sous PostgreSQL, exécutez chacune de ces requêtes par un copier/coller, visualisez les résultats et vérifiez leur validité par rapport au fichier *insert.sql*. Vous pouvez également utiliser la commande `\i select.sql` pour exécuter toutes les requêtes et examiner uniquement le résultat de la dernière. Vous serez peut-être amenés à insérer de nouveaux n-uplets ou à modifier des n-uplets existants pour tester si vos requêtes sont correctes.

- 1) Liste des navires de la compagnie *SNCM* possédant plus de 1000 places.
- 2) Dates et heures de départ des ferrys du modèle *NGV* au départ de *Marseille*.
- 3) Nombre de véhicules acceptés sur chaque navire (quelle que soit la catégorie).
- 4) Ports des zones *Corse* ou *Continent* au départ desquels plus de 2 trajets sont proposés.
- 5) Trajets de *Nice* à *Bastia* sur des navires qui proposent à la fois des commodités de type *C2E* et *C4E*.
- 6) Nombre de trajets au départ de chaque port.
- 7) Ports qui ne sont le point de départ d'aucun trajet.
- 8) Nom des navires qui naviguent au moins 2 fois.

Droits et vues

Plusieurs utilisateurs seront amenés à manipuler la base : le directeur (login *sncm*), la secrétaire (login *tapevite*) et des vacanciers (login *invite*).

Les droits de ces utilisateurs sont gérés de la façon suivante :

- La secrétaire saisit (consulte, insère, modifie ou supprime) les informations sur les navires, les commodités proposées par les navires et les catégories de véhicules acceptées dans les navires. De plus elle saisit (consulte, insère, modifie ou supprime) les informations sur les ports.
- Le directeur décide et enregistre les trajets proposés par la compagnie. Pour ce faire, il a le droit de tout consulter, et peut également enregistrer les trajets, leurs ports de départ et d'arrivée, ainsi que les bateaux utilisés par ces trajets. Il peut également décider de modifier le prix d'une commodité ou d'une catégorie de véhicules proposé sur un navire (en le diminuant s'il considère que le prix proposé par la compagnie est trop élevé pour sa clientèle (à sa charge alors de régler cela avec la compagnie ☺), ou en l'augmentant s'il souhaite prendre une marge supplémentaire). Evidemment, en cas d'erreur, le directeur peut modifier ou supprimer tout ce qu'il a saisi.
- Les vacanciers peuvent consulter tous les trajets qui ne sont pas encore partis et pour les navires concernés les prix des commodités proposées et des véhicules acceptés. Ils peuvent consulter le descriptif de toutes les commodités et toutes les catégories de véhicules.

- 1) Connectez-vous sur Postgres en tant que super-user (*postgres*), et tapez les commandes suivantes afin de créer les logins des différents utilisateurs de la base de données : le directeur (login *sncm*), la secrétaire (login *tapevite*) et les vacanciers (login *invite*).

```
create user tapevite with password 'tapevite' nocreatedb ;
create user sncm with password 'sncm' nocreatedb ;
create user invite with password 'invite' nocreatedb ;
```

Dans un fichier *grant.sql*, vous allez écrire les directives permettant de gérer les droits des utilisateurs.

- 2) Proposez une gestion de droits adéquate pour le directeur.
Puis sous PostgreSQL, depuis le compte *admin*, exécutez chacune de ces commandes par un copier/coller.
Connectez vous ensuite sur le compte du directeur et testez les droits que vous avez donnés à ces utilisateurs.
- 3) Ecrivez et testez les droits pour la secrétaire puis les vacanciers.
- 4) On souhaite gérer de nouveaux utilisateurs : les directeurs de ports, dont le login est le nom du port concerné. Par exemple, le directeur du port de Marseille aura le login *marseille*.
 - Créez l'utilisateur directeur du port de Marseille (login *marseille*) et l'utilisateur directeur du port de Nice (login *nice*).
 - Donnez les droits à chaque directeur de port de consulter les ports et de connaître le nombre de passagers qui passent chaque jour dans son port (à l'arrivée ou au départ) (indication : la fonction *getpgusername()* renvoie le login de l'utilisateur connecté).
 - Testez ces droits pour le directeur du port de Marseille ainsi que pour le directeur du port de Nice.

Procédures

Ecrivez puis testez les fonctions suivantes dans un fichier *procedure.sql* :

- 1) Une fonction *nbTrajetSem* qui renvoie le nombre de trajets qui partent d'un port donné durant une semaine à partir d'une date donnée.
- 2) Une procédure *ObligationSP* qui affiche un message d'erreur si les obligations de service public ne sont pas respectées durant la semaine qui suit une date donnée. Pour simplifier, on supposera que les obligations sont les suivantes : 3 départs de Marseille et 2 départs de Nice par semaine.

A réaliser avec curseurs :

- 3) Une procédure *Transfert* qui, pour un navire donné, affiche les dates où un transfert à vide doit être effectué par le navire car il arrive dans un port et repart d'un autre.
- 4) Une procédure *Promo*, qui étant données une date et une compagnie, effectue une remise de 20% sur tous les trajets ayant lieu à cette date sur des navires de cette compagnie. La procédure affichera également, pour chacun des navires concernés, un message indiquant la promo sous la forme : « Le 15/08/2015, sur le navire ..., trajets à partir de ... euros ».

Indications : pensez à utiliser la fonction *date_trunc* et sachez que la variable *found* peut-être utilisée après n'importe quelle requête. Elle indique si des n-uplets ont été trouvés ou modifiés.

Triggers

Ecrivez dans un fichier *trigger.sql* les directives permettant de gérer les contraintes d'intégrité suivantes. Pour chacune d'entre elles, exécutez ensuite des requêtes d'insertion de nouvelles données, de modification ou de suppression des données existantes afin de contrôler que les triggers permettent bien de vérifier les contraintes d'intégrité souhaitées.

- 1) Pour chaque trajet arrivant ou repartant d'un port, le nombre de places sur le navire concerné ne doit pas être supérieur au nombre de passagers que les infrastructures du port peuvent supporter.
- 2) Le nombre de places d'un navire est supérieur au nombre total de places de toutes ses commodités.
- 3) Le modèle d'un navire ne peut pas être modifié. Par contre, on peut changer le nom de la compagnie propriétaire de ce navire. En cas de changement du nom de la compagnie d'un navire, on affichera le nombre total de bateaux dont la nouvelle compagnie est propriétaire.
- 4) On ne peut pas modifier les informations sur un trajet moins de 3 mois avant le départ de la compagnie.
- 5) Pour pouvoir planifier un trajet, le navire choisi doit être disponible et on doit respecter un intervalle de 2h avant et après chaque trajet.