Ecole Polytechnique Universitaire de Marseille
Département Génie Industriel
5A – Option Systèmes d'information de production

EP BDA
Bases de données approfondies

Bernard ESPINASSE
Professeur à Aix-Marseille Université

2017

Sommaire

1 Objectif de l'enseignement pratique (EP):

Cet EP a pour objectif de permettre à l'étudiant d'assimiler une partie des connaissances développées dans le cours Bases de Données Approfondies de 5 ième année option SIP, principalement la maîtrise du développement d'applications autour d'une base de données relationnelle dans un contexte d'architecture client-serveur. De façon plus précise, on s'intéressera au développement de telles applications d'une part en langage PL/SQL (plus précisément PLpgSQL), un langage procédural, et d'autre part en Java en utilisant JDBC.

L'étude de cas qui est utilisée comme support dans cet EP est celle utilisée dans l'EP de 3ième année, il s'agit de l'organisation CYCLOPROVENCE et le SGBD utilisé est encore PostgreSQL, logiciel libre.

2 Ressources informatiques nécessaires

Matériel disponible :

- PC sous Linux (debian) avec imprimante (le TP peut aussi se faire sous Windows)
- Client PostgreSQL console ou interface graphique avec les programme d'administration pgAdmin3 et phppgadmin) avec PL/pgsql,
- Connexion internet,
- Java et JDBC (pour la connection à la base de données).
- Environnement ECLIPSE (ou NETBEANS),

Configuration requise :

- Serveur **postgreSQL : 147.94.190.228**
- 1 Login BD administrateur équipe : **equipeXX** où XX est le numéro d'équipe
- 3 Login BD utilisateur équipe : **eqXX_u1**, **eqXX_u2** et **eqXX_u3** où XX est le numéro d'équipe
- Mot de passe : password à changer, par défaut : "**pass.tmp**"

3 Rendu de l'EP

Le rendu de l'EP consistera pour chaque équipe en :

- **un fichier .pdf élaboré à partir d'un fichier de rendu VIDE accessible à l'adresse : http://www.lsis.org/espinasseb/Syllabus/syllabus_bda-GII.html**
- **un fichier archive du projet développé en Java (.zip ou .rar).**

Envoyé à l'adresse des encadrants.

Date limite de remise : le lundi xx à 12h.

4 Rappel sur l'étude de cas CYCLOPROVENCE

3.1. Présentation de CYCLOPROVENCE

Il y a maintenant plus de trois ans que vous avez, avec une amie, votre associée, Carine LAFLEUR, fondé la société CYCLOPROVENCE. Chaque été CYCLOPROVENCE permet à quelques centaines de touristes venant de tous les coins du monde de visiter à bicyclette, cette belle région qu'est la Provence.

La gestion de la compagnie CYCLOPROVENCE est assurée par les deux associés. Deux autres personnes vous aident pendant la période estivale. De plus la société CYCLOPROVENCE fait appel à une vingtaine de personnes, principalement des étudiants, pour assurer l'accompagnement des groupes de touristes.

4.1.1 Les circuits

La saison des promenades en bicyclette commence au mois de juin pour finir fin septembre. Pour préparer cette saison, CYCLOPROVENCE réalise dés le mois de mars, un catalogue de circuits. Pour cela, elle établit des circuits de durées différentes et contacte des hôtels ou des particuliers (hébergement à la ferme), se trouvant sur ces itinéraires et négocie avec ces derniers les prix des pensions par personne (repas du soir+nuit+petit déjeuner). Notons qu'à chaque jour du circuit est associé un hébergement différent, et un même hébergement peut se retrouver dans plusieurs circuits.

4.1.2 Les voyages

Pour chaque circuit, CYCLOPROVENCE propose un calendrier de dates. Pour un circuit donné, elle propose en moyenne 10 dates dans la saison, ce qui correspond à 10 voyages (un voyage correspondant à un circuit fixé dans le temps).

C'est au début avril que la compagnie CYCLOPROVENCE envoie un catalogue de ces voyages à ses principaux clients, qui sont principalement des agences de voyages d'Europe, des Etats-Unis et du Canada. Ce catalogue contient pour chaque voyage les informations suivantes:

- numéro du voyage,
- date de début et de fin
- désignation du circuit,
- durée du circuit en nombre de jours,
- distance totale à parcourir,
- prix par personne, …
- le nom de l'hébergement pour chaque jour du voyage,

Exemple:

- *Voyage numéro 2:*
- *Du 8 au 10 juin 2007*
- *"Tour en 3 jours des baux de Provence",*
- *3 jours*
- *Distance totale parcourue: 150 Kms,*
- *Coût 300 € par personne,*
- *Le nom des hébergements,*

1°jour: Le cheval blanc
2°jour Le moulin de Daudet
3°jour: Mas du marais

Ces dates pourront, dans une certaine mesure être discutées avec les agences intéressées.

4.1.3 Réservation par les clients

A partir de ces propositions, les clients (agences), font leurs réservations pour un certain nombre de places pour un voyage donné, ceci en général deux mois à l'avance. Par exemple:

L'agence ALPHA-Voyage de Montréal (Canada) a réservé le 10 mai (réservation RC03):

a) 7 personnes pour le voyage # 20 (qui correspond au circuit #38, effectué entre le 9 et le 13 juillet), et

b) 5 personnes pour le voyage #17 (qui correspond au circuit #20, effectué entre le 1 et le 6 août).

Notons qu'une réservation peut porter sur plusieurs voyages.

A la réception d'une telle réservation, CYCLOPROVENCE envoie à l'agence une première facture de 30 % du montant des billets.

4.1.4 Confirmation des réservations

Les agences ont la possibilité de modifier ou d'annuler leurs réservations jusqu'à un mois avant la date de début du ou des voyages réservés. A cette date (1 mois avant le voyage), les agences doivent avoir confirmé leurs réservations par le règlement de la première facture des 30 %. Confirmées, les réservations clients passent du statut N au statut C. Si à cette date, cette confirmation n'est pas faite, la réservation est annulée. La confirmation étant faite, aucun remboursement pour cause d'annulation ne pourra être fait. A chaque voyage est affecté un accompagnateur.

4.1.5 Réservations des hébergements

Chaque fin de semaine CYCLOPROVENCE traite les réservations clients qui ont été confirmées durant la semaine (statut C), en générant les réservations d'hébergement correspondantes qui seront envoyées aux hébergements concernés (ces réservations clients passent alors en statut T). Par exemple:

La réservation RH03, faite le 20 juin 2007 à hôtel "Le moulin de Daudet" comprend :
- pour le 10 juillet 2007, 8 pensions pour le voyage #22
et
- pour le 11 juillet 2007, 5 pensions pour le voyage #23
La réservation RH05 faite le 22 juin 2007 au Mas de Mme Honorine Olivier comprend :
- pour le 11 juillet 2007, 8 pensions pour le voyage #22

On notera qu'une réservation hébergement ne concerne qu'un et un seul lieu d'hébergement et peut être liée à plusieurs voyages différents pour des dates de pension différentes. De plus on supposera que les hôtels et les fermes ont toujours des places disponibles.

Autres remarques :

- Le solde du montant des billets devra être réglé par l'agence un (1) mois après la réception de la deuxième facture qu'enverra CYCLOPROVENCE, une fois le voyage effectué, les réservations clients passent alors en statut R,
- Le règlement des hôtels et particuliers se fait un mois après réception de leurs factures,
- Les accompagnateurs sont payés après chaque voyage à un tarif fixe par jour d'accompagnement.

4.1.6 Les groupes

Dans la mesure des places disponibles, des touristes particuliers peuvent réserver une ou plusieurs places pour un voyage et de cette façon compléter une équipe déjà constituée. Ils sont alors considérés comme client à part entière. Un règlement de 30 % du montant des billets leur est demandé lors de la réservation et le solde est récolté le jour du départ du voyage. Les équipes ne peuvent pas dépasser plus de dix personnes, ceci pour des raisons évidentes de sécurité.

3.2. Schéma de la base de données considérée

Le schéma de la base de données utilisée pour cette EP est le suivant :

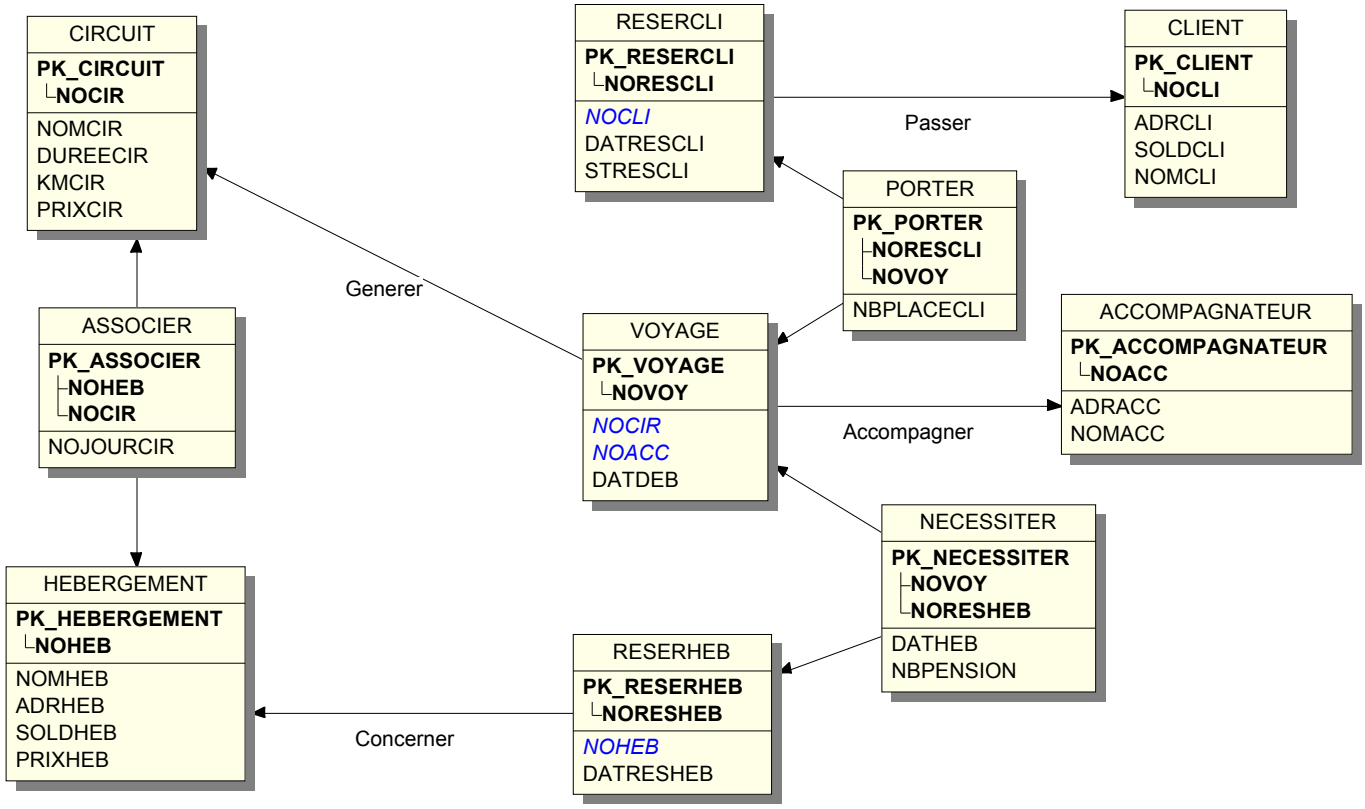
- **CIRCUIT** (**NOCIR**, NOMCIR, DUREECIR, KMCIR, PRIXCIR)
- **CLIENT** (**NOCLI**, NOMCLI, ADRCLI, SOLDCLI)
- **HEBERGEMENT** (**NOHEB**, ADRHEB, SOLDHEB, PRIXHEB)
- **ACCOMPAGNATEUR** (**ACC**, NOMACC, ADRACC)
- **VOYAGE** (**NOVOY**, ***NOCIR***, ***NOACC***, DATDEBVOY)
- **RESERCLI** (**NORESCLI**, ***NOCLI***, DATRESCLI, STRESCLI)
- **RESERHEB** (**NORESHEB**, ***NOHEB***, DATRESHEB)
- **PORTER** (***NOVOY , NORESCLI*,** NBPLACECLI)
- **NECESSITER** (***NOVOY , NORESHEB***, DATHEB, NBPENSION)
- **ASSOCIER** (***NOCIR, NOHEB***, NOJOURCIR)

avec : **CIR** : clé primaire et ***CLI*** : clé étrangère

Nom attribut	Signification
PRIXHEB	- prix d'une pension pour un hébergement donné
SOLDCLI	- solde du client
ADRACC	- adresse accompagnateur
NONCIR	- nom circuit
NOMHEB	- nom hébergement
ADRCLI	- adresse client
ADRHEB	- adresse hébergement
DATRESHEB	- date à laquelle est faite la réservation d'hébergement
DATDEBVOY	- date de début du voyage
DATHEB	- date réservation de l'hébergement
DATRESCLI	- date réservation du client
DUREECIR	- durée du circuit (jours)
NOACC	- ident. accompagnateur
NOHEB	- ident. hébergement (hôtel, restaurant, particulier)

NORESCLI	- ident. réservation du client
NORESHEB	- ident. réservation hébergement (pensions)
NOVOY	- ident. voyage
NOCIR	- ident. circuit
NOCLI	- ident. client
NOMACC	- nom accompagnateur
NOMCLI	- nom du client (agence ou particulier)
KMCIR	- nombre de km du circuit
NBPENSION	- nombre de lits réservés (un voyage et un hébergement donné)
NBPLACECLI	- nombre places réservées par l'agence pour un voyage donné
NOJOURCIR	- numéro du jour d'un hébergement dans un circuit
PRIXCIR	- prix du circuit pour une personne
SOLDHEB	- solde hébergement (hôtel et ferme)
STRESCLI	- statut de la réservation du client: (N: non confirmée; C: confirmée; T:traitée; R:réglée)

Le modèle relationnel associé est le suivant :



Les PK_ en caractères gras précisent les attributs clés primaires et les attributs en italiques sont les clés étrangères.

5 Travail demandé

Le travail demandé dans cet EP concerne principalement le développement d'applications exploitant la base de données. Néanmoins il vous est d'abord demande d'élaborer un jeu d'essai.

Schéma de la base de données

A partir du schéma de création de la base de données INCOMPLET proposé en annexe, on vous demande de le compléter afin de prendre en compte les contraintes d'intégrité suivantes :

- déclaration des clés étrangères,
- déclaration des options de suppression sur ces clés étrangères (ON DELETE ….),
- toutes autres déclarations de contraintes d'intégrité vues en cours que vous jugerez utile.

Elaboration d'un jeu d'essai

A partir des informations fournies en annexes, vous constituerez votre propre jeu de données d'essai complet. Vous saisirez ces données dans chacune des tables de votre base de données précédemment créée (pas plus de 10 enregistrements par table).

Applications à développer sur la base de données Cycloprovence

Comme nous l'avons déjà évoqué, l'objectif de cet EP est de permettre à l'étudiant le développement d'applications autour d'une base de données relationnelle dans un contexte d'architecture client-serveur.

Deux types applications seront développées, le premier type d'application seront développées dans un langage de procédure (PL/pgsql), et le second type d'application en langage Java en utilisant JDBC.

La prise en compte d'une réservation faite par un client sera réalisée en java (Application 2 java) avec utilisation de trigger, et d'autre part la génération de réservations auprès des hébergements suite aux réservations faites par les clients sera réalisé en pl/sql (Application 4).

6 Développement en PL/pgsql

Application 1 : Edition du catalogue

Il s'agit dans cette application en PL/pgsql, langage de procédure disponible dans le SGBD PostgreSQL, permettant d'afficher à l'écran le catalogue de CYCLOPROVENCE, sous la forme d'un tableau du type :

No Voy	Date début	Date fin	Nb jours	Distance	Places dispo	Désignation	Prix	Hébergements
2	8/06/07	10/06/07	3	150	5	*Tour en 3 jours des baux de Provence*	300	*Le cheval blanc*
								Le moulin de Daudet
								Mas du marais
3	10/06/07	12/06/07	4	200	3	*Tour de la Sainte Victoire*	400	…
								…
								…

Correspondant à :

- *Voyage numéro 2:*
- *Du 8 au 10 juin 2007*
- *"Tour en 3 jours des baux de Provence",*
- *3 jours*
- *Distance totale parcourue: 150 Kms,*
- *Coût 300 € par personne,*
- *Le nom des hébergements,*

1°jour: Le cheval blanc
2°jour Le moulin de Daudet
3°jour: Mas du marais

Application 2 : Récapitulatif des réservations clients

Il s'agit dans cette application d'afficher à l'écran sous la forme d'un tableau l'ensemble des réservations faites par les clients, par exemple :

No Reser.	Client	Date	No. Voy	Nb places
…	…	…	…	…
RC03	ALPHA-Voyage	10/05/07	20	7
RC03	10/06/07	10/06/07	17	5

Correspondant à :

L'agence ALPHA-Voyage de Montréal (Canada) a réservé le 10 mai (réservation RC03):

a) 7 personnes pour le voyage # 20 (qui correspond au circuit #38, effectué entre le 9 et le 13 juillet), et

b) 5 personnes pour le voyage #17 (qui correspond au circuit #20, effectué entre le 1 et le 6 août).

Application 3 : Récapitulatif des réservations hébergement

Il s'agit dans cette application d'afficher à l'écran sous la forme d'un tableau l'ensemble des réservations faites aux hébergements.

Par exemple ont été faites les réservations suivantes :

La réservation RH03, faite le 20 juin 2007 à hôtel "Le moulin de Daudet" comprend :
- pour le 10 juillet 2007, 8 pensions pour le voyage #22
et
- pour le 11 juillet 2007, 5 pensions pour le voyage #23
La réservation RH05 faite le 22 juin 2007 au Mas de Mme Honorine Olivier comprend :
- *pour le 11 juillet 2007, 8 pensions pour le voyage #22*

Soit :

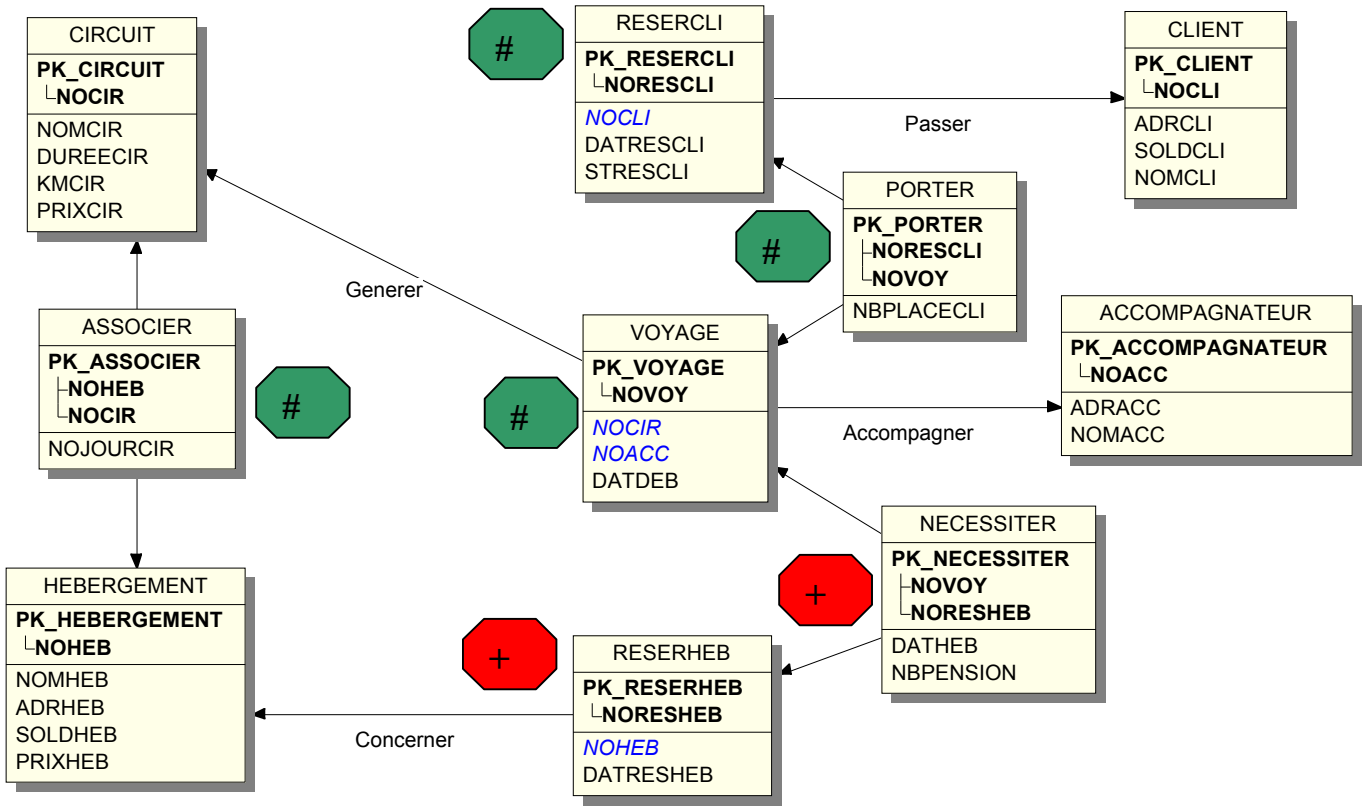
No Reser.	**Date de la réservation**	**Hébergement**	**No Voy**	**Nb places**	**Date reservée**
…	…	…	…	…	…
RH03	20/06/07	*Le moulin de Daudet*	22	8	10/07/07
			23	5	11/07/07
RH05	22/06/07	*Mas de Mme Honorine Olivier*	22	8	11/07/07
…					

Application 4 : Génération de réservations hébergements

Les clients, des agences de voyages rappelons-le, font des réservations de places sur des voyages du catalogue de la société CYCLOPROVENCE, jusqu'à un mois avant le début du voyage, il est encore possible au client d'annuler sa réservation. Passée ce mois, le client doit avoir confirmé sa réservation par un règlement de 30% de sa réservation. Sa réservation passe alors dans le statut *confirmé* « C ».

Chaque fin de semaine, les réservations clients confirmées (statut C) sont traitées pour générer les réservations d'hébergements auprès des hébergements (auberges, hôtels et gîtes). Les réservations client en statut C traitées passent alors en statut *traitée* « T ». On rappelle qu'une réservation hébergement ne concerne qu'un seul hébergement et peut concerner plusieurs voyages différents.

Il vous est demandé de développer une application en PL/pgsql permettant de générer ces réservations aux hébergements. Cette application étant essentiellement utilisée par les membres de Cycloprovence. Cette application exploitera la base de données ainsi :



avec :

#	Consultation d'un enregistrement (tuple)
+	Création d'un enregistrement (tuple)

Remarque : usage de TRIGGER :

Pour la génération automatique des clés primaires des nouveaux enregistrements des tables RESERHEB et NECESSITER, on utilisera un trigger.

7 Développement en Java avec JDBC

Application 1: Affectation d'un accompagnateur à un voyage.

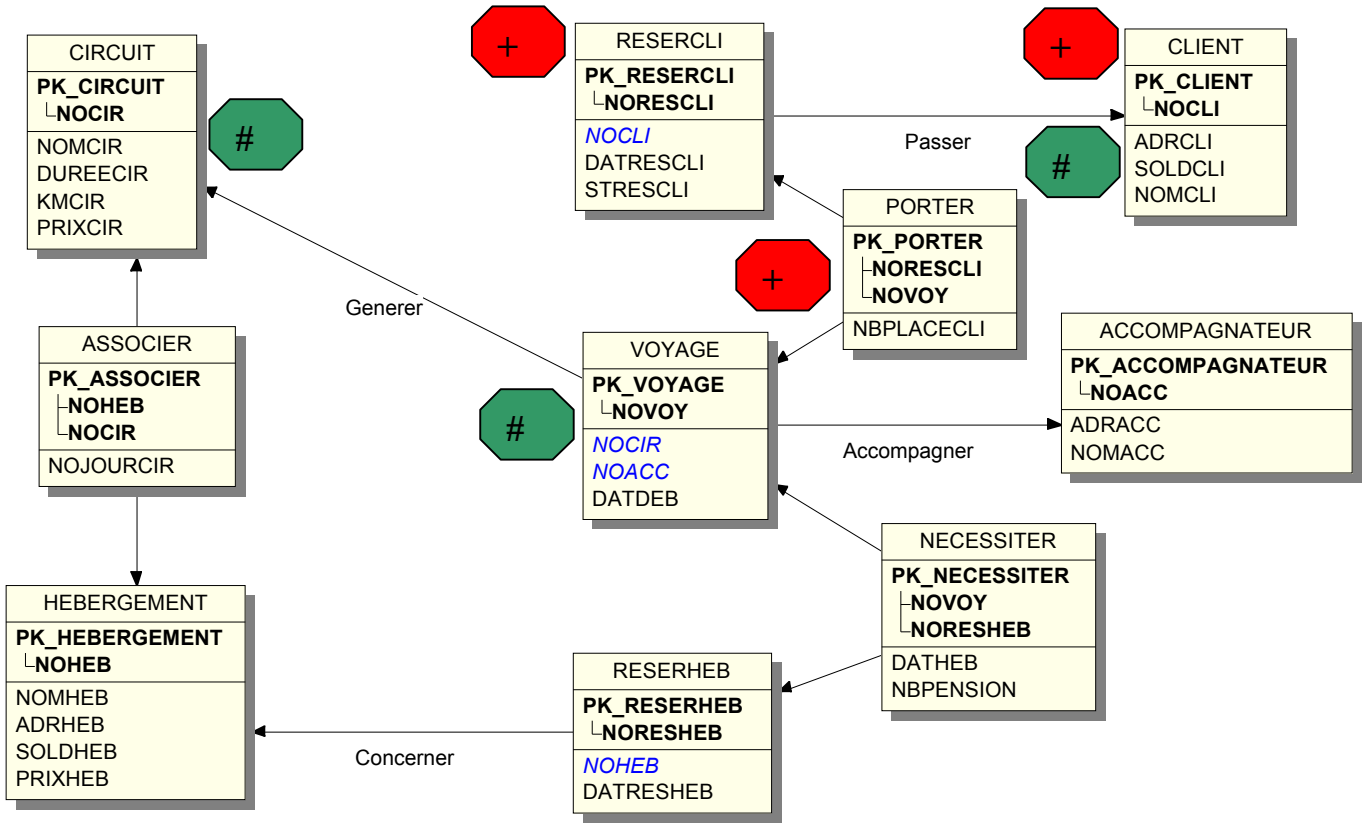
Il s'agit dans cette application de développer un programme Java avec son interface homme-machine associé permettant à un membre de CYCLOPROVENCE d'affecter un accompagnateur à un voyage. Ce programme lui permettra d'affecter soit un accompagnateur déjà dans la base de données à un voyage, soit de créer un nouvel accompagnateur avant de l'affecter à un voyage.

Remarque : usage de TRIGGER :

Pour la génération automatique des clés primaires des nouveaux enregistrements de la table ACCOMPAGNATEUR dans le cas de le création d'un nouvel enregistrement, on utilisera un trigger.

Application 2 : Prise de réservation client

Les clients (agences de voyages) doivent pouvoir réserver des places sur des voyages du catalogue de la société Cycloprovence. Il s'agit de développer une application qui sera utilisée par les clients permettant l'affichage du catalogue et la prise de réservation sur les voyages de ce catalogue. Le développement de cette application sera réalisé en Java en utilisant JDBC.



avec :

#	Consultation d'un enregistrement (tuple)
+	Création d'un enregistrement (tuple)

Cette application nécessitera le développement d'une interface conviviale en Java permettant d'afficher au client le catalogue de voyages proposé par CLYCLOPROVENCE (avec toutes les informations relatives au circuit associé au voyage) et permettant au client de réserver une ou plusieurs places sur un ou plusieurs voyages. Les places disponibles devront être affichées et mises à jour (un voyage ne peut excéder 10 personnes).

Avant de prendre en compte une réservation, l'application demandera au client de s'identifier. S'il n'est pas encore enregistré dans la table CLIENT, il faudra créer un nouvel enregistrement dans celle-ci.

Pour le développement de cette interface utilisateur conviviale, on utilisera en Java par exemple des Jlist avec ascenseur permettant de parcourir le catalogue, afficher les détails de chaque circuit, les dates auxquelles le voyage est proposé et le nombre de places encore disponibles.

Une validation finale de la réservation sera demandée par le client avant d'impacter la base de données.

Remarque : usage de TRIGGER :

Pour la génération automatique des clés primaires des nouveaux enregistrements des tables PORTER et RESCLI et CLIENT, on utilisera un trigger.

8 Annexes

Annexe 1: éléments pour l'élaboration d'un jeu de données

Informations relatives aux circuits :

Nom du circuit	Durée en jour	Distance en km	Prix en € par personne
TOUR DES BAUX	3	170	350
TOUR DU LUBERON	3	90	400
TOUR DE CAMARGUE	3	80	500
LES PAYSAGES DE CEZANE	2	90	350
LA SAINTE VICTOIRE	2	80	350

…

Informations relatives aux clients :

Numéro	Nom du client	Adresse du client
C01	VOYAGE ALPHA	PARIS
C02	QUEBEC TOUR	MONTREAL
C03	BIG APPLE TRAVEL	NEWYORK
C04	SOLEIL TOUR	OSLO
C05	TOUR ITALIA	ROME

…

Informations relatives aux hébergements :

Numéro	Nom hébergement	Prix pension	Adresse
H01	LE CHEVAL BLANC	15	LES BAUX
H02	LE MOULIN DE DAUDET	20	LES BAUX
H03	MAS DU MARAIS	26	LES BAUX
H04	L'AUBERGE DE CAMARGUE	28	CAMARGUE
H05	MAS CHEZ FONFON	25	CAMARGUE
H06	LE BRIN DE LAVANDE	40	CAMARGUE
H07	MAS FARIGOULO	30	LUBERON
H08	FERME DES TORROS	35	LUBERON
H09	LE GITE DES ALPILLES	20	LUBERON

…

Informations relatives aux accompagnateurs :

N°accompagnateur	Nom accompagnateur	Adresse
A01	DURAND	VELAUX
A02	OLIVE	MARSEILLE
A03	GROSJEAN	CASSIS

A04	RAYMOND	AUBAGNE
A05	SEGUIN	MARSEILLE
…		

Informations relatives aux itinéraires des circuits :

Numéro du circuit	Numéro de l'hébergement	Numéro de jour dans le circuit
001	H01	1
001	H02	2
001	H03	3
002	…	1
…		

Annexe 2: Création des tables sous POSTGRESQL

Vous disposez de deux programmes d'administration du SGBD PostgreSQL. Le premier programme est une application nommée « **pgadmin3** » qu'il vous faut lancer de la console. Le second « **phppgadmin** » est une application Web que vous lancerez à partir d'un **navigateur** en tapant, dans la fenêtre d'url : **147.94.190.147/phppgadmin.** Ces deux programmes sont équivalents, cependant pgadmin3 possède quelques fonctionnalités supplémentaires.

ATTENTION : Seul l'utilisateur administrateur de l'équipe (equipeXX) possède les droits pour créer la base de données (BD) et toutes ses tables.

Une fois que l'administrateur à créé la BD, pour chacune des tables il doit transmettre les droits nécessaires (SELECT, UPDATE, …) au utilisateurs non administrateur de son équipe (eqXX_u1, eqxx_u2 et eqXX_u3) en utilisant un des deux programmes d'administration (phppgadmin ou pgadmin3).

Nous vous conseillons de créer vos tables à partir d'un fichier de commande, fichier texte, lancé au niveau de psql par la commande \i nom_de_fichier.

Script INCOMPLET de création de la BD CycloProvence :

```
-- ----------------------------------------------------------------------------
drop database CycloEquipe1;
CREATE DATABASE CycloEquipe1;
-- ----------------------------------------------------------------------------
CREATE TABLE HEBERGEMENT
  (
   NOHEB char(32) NOT NULL  ,
   NOMHEB char(32) NULL  ,
   ADRHEB char(32) NULL  ,
   SOLDHEB money NULL  ,
   PRIXHEB money NULL
,  CONSTRAINT PK_HEBERGEMENT PRIMARY KEY (NOHEB)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE RESERCLI
  (
   NORESCLI char(32) NOT NULL  ,
   NOCLI char(32) NOT NULL  ,
   DATRESCLI date NULL  ,
```

```
   STRESCLI char(32) NULL
,  CONSTRAINT PK_RESERCLI PRIMARY KEY (NORESCLI)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE CLIENT
  (
   NOCLI char(32) NOT NULL  ,
   ADRCLI char(32) NULL  ,
   SOLDCLI money NULL  ,
   NOMCLI char(32) NULL
,  CONSTRAINT PK_CLIENT PRIMARY KEY (NOCLI)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE ACCOMPAGNATEUR
  (
   NOACC char(32) NOT NULL  ,
   ADRACC char(32) NULL  ,
   NOMACC char(32) NULL
,  CONSTRAINT PK_ACCOMPAGNATEUR PRIMARY KEY (NOACC)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE RESERHEB
  (
   NORESHEB char(32) NOT NULL  ,
   NOHEB char(32) NOT NULL  ,
   DATRESHEB date NULL
,  CONSTRAINT PK_RESERHEB PRIMARY KEY (NORESHEB)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE CIRCUIT
  (
   NOCIR char(32) NOT NULL  ,
   NOMCIR char(32) NULL  ,
   DUREECIR int4 NULL  ,
   KMCIR int4 NULL  ,
   PRIXCIR money NULL
,  CONSTRAINT PK_CIRCUIT PRIMARY KEY (NOCIR)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE VOYAGE
  (
   NOVOY char(32) NOT NULL  ,
   NOCIR char(32) NOT NULL  ,
   NOACC char(32) NOT NULL  ,
   DATDEB date NULL
,  CONSTRAINT PK_VOYAGE PRIMARY KEY (NOVOY)
```

```
  );
-- ----------------------------------------------------------------------------
CREATE TABLE NECESSITER
  (
   NOVOY char(32) NOT NULL  ,
   NORESHEB char(32) NOT NULL  ,
   DATHEB char(32) NULL  ,
   NBPENSION char(32) NULL
,  CONSTRAINT PK_NECESSITER PRIMARY KEY (NOVOY, NORESHEB)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE PORTER
  (
   NORESCLI char(32) NOT NULL  ,
   NOVOY char(32) NOT NULL  ,
   NBPLACECLI char(32) NULL
,  CONSTRAINT PK_PORTER PRIMARY KEY (NORESCLI, NOVOY)
  );
-- ----------------------------------------------------------------------------
CREATE TABLE ASSOCIER
  (
   NOHEB char(32) NOT NULL  ,
   NOCIR char(32) NOT NULL  ,
   NOJOURCIR char(32) NULL
,  CONSTRAINT PK_ASSOCIER PRIMARY KEY (NOHEB, NOCIR)
  );
-- ----------------------------------------------------------------------------
```

Script à l'adresse : **http://www.lsis.org/espinasseb/Syllabus/syllabus_bda-GII.html**

Insertion de tuples dans les tables

Pour insérer les enregistrements dans les tables selon votre jeu de données nous vous conseillons la même procédure. Pour insérer des données dans les 2 tables précédentes, voici le fichier utilisé :

```
-- Insertion dans CLIENT et COMMANDE

INSERT INTO CLIENT VALUES ('C01', 'DURAND', 45, 120.50);
INSERT INTO CLIENT VALUES ('C02', 'DUVAL', 51, 200.50);
INSERT INTO CLIENT VALUES ('C03', 'DUPOND', 35, 80.50);
INSERT INTO COMMANDE VALUES ('D01', 'C01', '2004-07-08');
INSERT INTO COMMANDE VALUES ('D02', 'C02', '2004-07-15');
INSERT INTO COMMANDE VALUES ('D03', 'C03', '2004-07-20');
INSERT INTO COMMANDE VALUES ('D04', 'C01', '2004-07-25');
```

Annexe 3: Mise en œuvre de PL/pgSQL

Avant de développer vos applications en **PL/pgSQL**, l'administrateur doit **rattacher le langage « plpgsql » à la base de données de l'equipe.**

Rattachement de PL/pgSQL sous linux en mode commande

Pour se connecter à la base de données executer la commande suivante:

```
$ psql   nom_base_de_donnees
```

Vous obtiendrez l'interface de PostgreSQL en ligne de commande avec comme début quelques commandes clés

```
==> Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
nom_base_de_donnees=#
```

Le langage PL/pgSQL est un langage procédural chargeable pour le système de bases de données PostgreSQL, inspiré du célèbre langage PL/SQL proposé avec la solution ORACLE leadeur des SGBD. Il vient compléter les lacunes du langage d'interrogation SQL, comme la déclaration des variables, et le traitement des opérations complexes.

```
$ createlang plpgsql nom_base_de_donnees
CREATE LANGUAGE
$
```

Pour lancer un script SQL à la console, comme pour exécuter des requêtes SQL vous pouvez utiliser directement la ligne de commande, ou bien des fichiers d'extension « .sql » dans lequel sont stockées les requêtes à travers la commande « \i ».

```
nom_base_de_donnees=#\i script.sql
```

Comme décrit ci-dessus, les fonctions PL/pgSQL peuvent être lancées directement dans l'interface ligne de commande ou à travers des fichiers « .pgs ». Elles sont lancées comme suit :

```
nom_base_de_donnees=#\i script.pgs
```

Rattachement de PL/pgSQL à partir du programme d'administration « pgadmin3 »

Dans le programme « pgadmin3 », aller dans l'onglet « préférences », puis « navigateur », cocher la case « langage » , cliquer sur « langage », ajouter objet, puis ajouter langage, choisir « plpgsql ».

Syntaxe de PL/pgSQL de PostgreSQL

Ci-dessous la syntaxe de la déclaration d'une procédure ou fonction PL/PGSQL.

```
 [ <<label>>]
[ DECLARE
```

```
déclarations ]
BEGIN
instructions
END [label];
```

En PL/PGSQL il n'y a **pas de bloc anonyme et de procédure**. Il n'y a que des fonctions.
Exemple de fonction :

```
CREATE FUNCTION toto() RETURNS integer AS $$
DECLARE
    quantité integer := 30;
BEGIN
   RAISE NOTICE 'quantité vaut ici %',
     quantité; -- quantité vaut ici 30 quantité := 50;
   --
   -- Crée un sous-bloc
   --
   DECLARE
        quantité integer := 80;
   BEGIN
      RAISE NOTICE 'quantité vaut ici %', quantité;
      -- quantité vaut ici 80
   END;
   RAISE NOTICE 'quantité vaut ici %', quantité;
   -- quantité vaut ici 50
   RETURN quantité;
END;
$$ LANGUAGE plpgsql;
```

Cette fonction sera par exemple sauvegardée dans le fichier : **test.sql**

Le chargement de ce fichier se fra à la console ainsi :

> \i test.sql

La fonction toto sera exécutée à la console ainsi :

> SELECT * FROM toto();

Une fonction PL/pgSQL peut être déclarée comme renvoyant le type ***void*** si elle n'a pas de valeur de retour utile, elle est alors assimilable à une **procédure**, par exemple :

```
CREATE FUNCTION titi (cle INT, donnee TEXT) RETURNS VOID AS $$
DECLARE
…
BEGIN
  …
END;
$$ LANGUAGE plpgsql;
```

Pour plus d'information se conformer au poly présentant le langage, notamment celui mis sur la page du cours.

Annexe 4: Usage de JAVA, ECLIPSE et JDBC

JAVA et ECLIPSE

La mise en œuvre de JAVA se fera au travers de l'environnement ECLIPSE. ECLIPSE est une communauté ope source dont le but est de fournir un environnement de développement open source qui soit extensible, et favorise la construction, le déploiement et la gestion des applications informatiques durant leurs cycles de vie. Nous nous intéressons dans cet EP exclusivement à la partie d'implémentation utilisant le langage JAVA, qui est l'une des possibilités qu'offre cet environnement pour le développement.

Connexion à la base de données

Pour exploiter les données de la base de données PostgreSQL, vous pouvez procéder aux étapes suivantes dans l'ordre.

1. Importer JDBC

```
import java.sql.*;
```

Cette instruction permet à l'utilisateur d'utiliser les classes et les fonctions relatives à la connexion et à l'interrogation des bases de données (Connection).

2. Charger le pilote avec la méthode:

```
Class.forName("org.postgresql.Driver");
```

Cette instruction permet de charger le pilote en tant qu'interface avec PostgreSQL.

3. Se connecter à la base de données

Avec JDBC, une base de données est représentée par une URL. Avec PostgreSQL, elle peut prendre l'une des formes suivantes :

```
jdbc:postgresql: cycloequipeXX
jdbc:postgresql://hôte/ cycloequipeXX
jdbc:postgresql://hôte:port/cycloequipeXX
```

4. Pour vous connecter à la base de données, vous avez besoin d'une instance de la classe Connection provenant de JDBC. Pour cela, vous utilisez la méthode DriverManager.getConnection() :

```
Connection db = DriverManager.getConnection(url, username, password);
```

5. Fermer la connexion

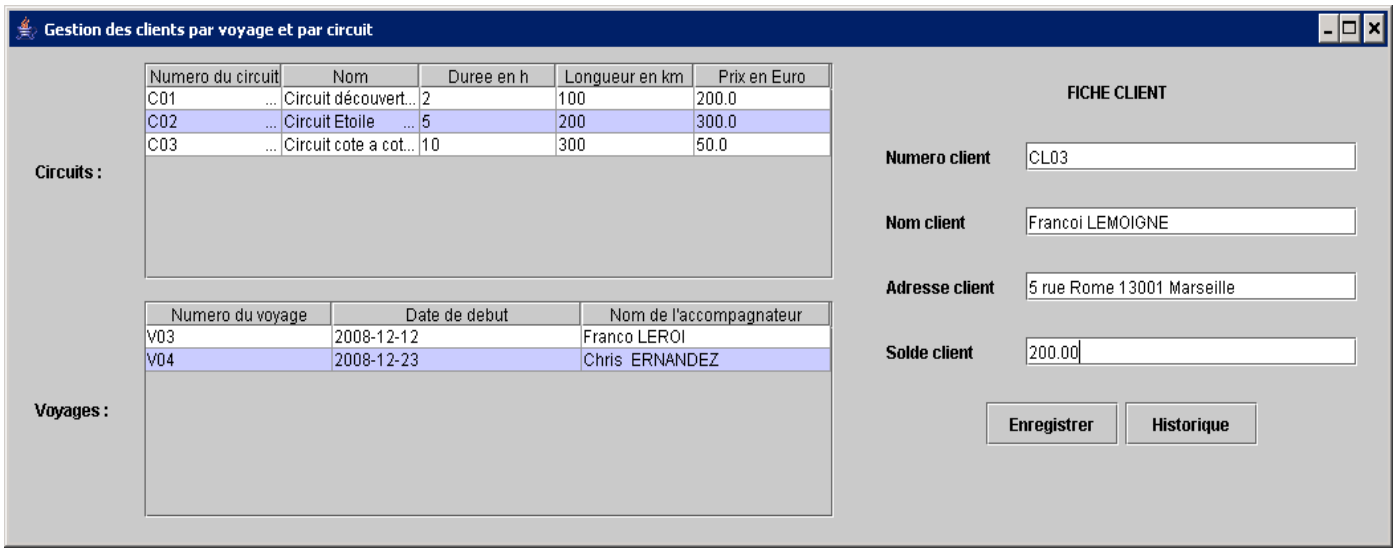
Pour fermer la connexion à la base de données, appelez simplement la méthode close() pour la Connexion déjà établie :

```
db.close();
```

Exemple de maquette

Voici un exemple d'une maquette Maitre/détail qui représente une première partie décrivant la liste des circuits proposés par l'agence pour ses clients, une deuxième partie

décrivant la liste des voyages relatifs à un circuit sélectionné, et une troisième partie présentant les informations client relatives à un voyage dans un circuit donné. Cette dernière partie permet de modifier les informations relatives à un client.



JDBC

JDBC est une API relative au langage Java. Elle fournit un ensemble standard d'interfaces vers les bases de données compatibles SQL. La mise en œuvre de JDBC permettant au programme développé en JAVA d'accéder à la base de données se fera ainsi.

Interroger PostgreSQL à travers le JDBC

Voici un exemple de récupération du résultat d'une requête SQL simple

```
Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE);
String requete = "SELECT * FROM circuit";
ResultSet rs = stmt.executeQuery(requete);
```

En premier lieu il faut déclarer un objet de type « Statement » pour pouvoir lancer la requête correspondante, ensuite définir la requête dans une variable chaine de caractère, Enfin créer un objet de type « ResultSet », dans lequel sera stocké le résultat de la requête lancée à laide de la fonction « executeQuery() ».

L'objet rs contiendra le résultat de la requête lancée.