



Master 1 Informatique

Projet Bases de données Avancées

Thème : gestion d'évènements

Cas : Gestion de matchs de Football

Réalisé par :

- Salim ABDELFFETTAH

Année Universitaire

2014/2015

Introduction

Le projet a pour but la gestion de compétitions de Football. Il est inspiré et modélisé pour le football britannique.

- Un client achète un billet (réserve une place d'un stade).
- Un billet correspond à une place dans le stade de l'équipe (club) qui joue à domicile et le concerne deux équipes (l'équipe qui reçoit et l'équipe reçu).
- Le match se joue pour une phase (journée ou phase d'élimination) d'une compétition donnée pour une saison donnée.
- Une compétition est soit, une suite de journées (qui elles même sont des suites de matchs entre deux équipes ; les deux équipes sont distinctes entre elles dans le match et toutes les équipes sont distinctes dans la journée), soit une suite de matchs à élimination direct (pour deux équipes jouant lors d'un match, une seule accède à la phase suivante).
- Une phase se joue lors d'une saison donnée dans une compétition donnée. La même phase est jouée dans la même compétition chaque saison.
- Un club (équipe) joue dans plusieurs compétitions différentes dans une même saison et joue (généralement) chaque saison.

Modèle entités-associations

A partir de la brève présentation du sujet, le modèle entités-associations a été conçu comme suit :

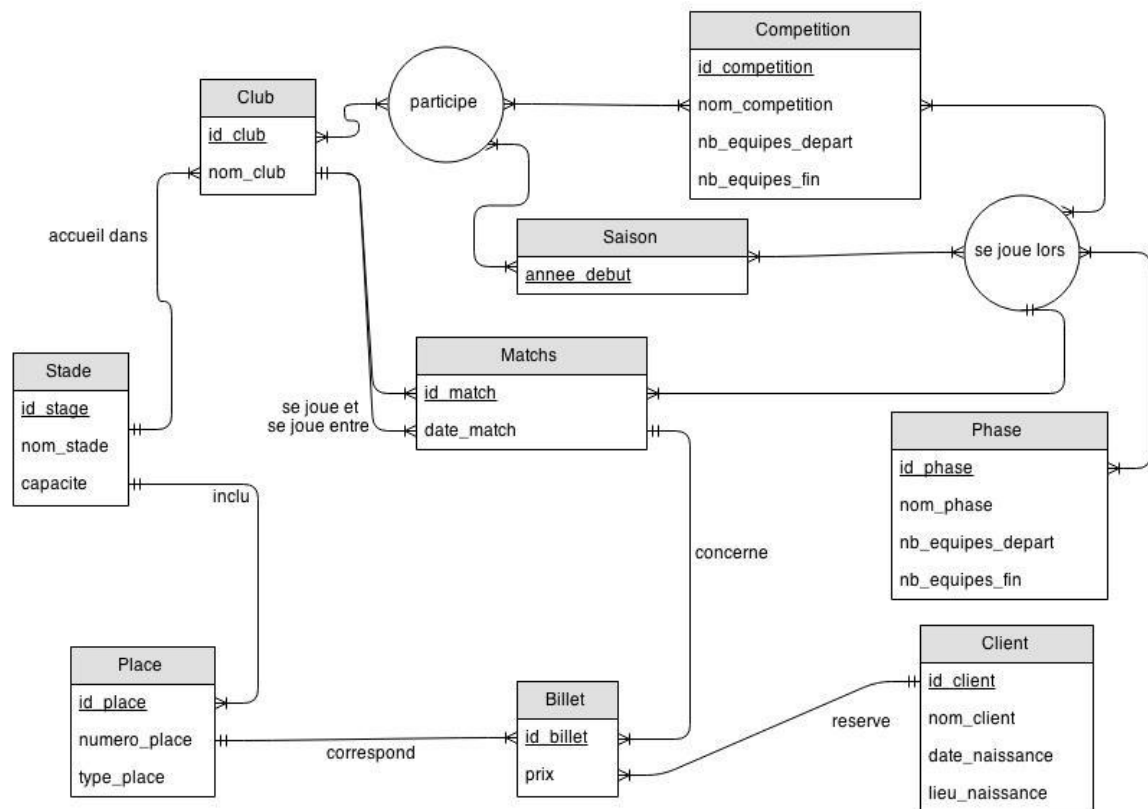


Figure 1: modèle entités-associations

Lors de la modélisation, un accent a été mis sur la cohérence du contenu des tables. En effet, pour ajouter un club (une équipe) dans un match d'une phase d'une compétition pour une saison donnée, il doit y être autorisé. Cela se traduit par le fait de déclarer (à l'aide de la relation participe) qu'un club participe à une compétition donnée lors d'une saison spécifique. Un autre exemple de cohérence est le fait de faire jouer une équipe au plus un match pour une même phase d'une même compétition d'une même saison. D'autres exemples sont donnés dans les triggers.

Le reste du modèle a été conçu en se basant sur le concret. Un billet contient les informations suivantes : le client qui l'a réservé, le match qu'il concerne et la place du stade où se joue le match. Un stade peut accueillir un ou plusieurs clubs, et par accueillir l'on sous-entend le fait que le match à domicile d'une équipe se joue dans ce stade.

Schéma relationnel

Le schéma relationnel obtenu à partir du modèle entités-associations est comme suit :

Client(**id_client**, nom_client, date_naissance, lieu_naissance)

Competition (**id_competition**, nom_competition, nb_equipes_depart, nb_equipes_fin)

Phase(**id_phase**, nom_phase, nb_equipes_depart, nb_equipes_fin)

Saison (**annee_debut**)

Stade(**id_stade**, nom_stade, capacite)

Place(**id_place**, numero_place, type_place, id_stade)

Club (**id_club**, nom_club, id_stade)

Participe(**id_club**, **id_competition**, **annee_debut**)

Matches(**id_match**, date_match, id_club_domicile, id_club_exterieur, id_competition, id_phase, annee_debut)

Billet (**id_billet**, prix, id_place, id_match, id_client)

Messagerie (**id_message**, id_client, date_message, message)

Il est à noter que les relations « se joue entre » et « se joue et » ont été respectivement renommé dans le schéma relationnel par « id_club_domicile » et « id_club_exterieur ». Il est aussi à noter l'ajout de la table 'Messagerie' qui ne fait pas partie de modèle entités-associations et qui a été rajouté pour des besoins de communication avec le client.

Les Fonctions

- type_competition : *paramètre : identifiant d'une compétition (id_competition)* - [TEXT](#)

Renvoie le type de la compétition (identifiant donné en paramètre) :

- 1- 'Championnat' : si le nombre d'équipes de départ et de fin sont égal et supérieur à 1
- 2- 'Coupe' : si le nombre d'équipes de départ et de fin sont différent, le nombre d'équipes de départ est supérieur au nombre d'équipes de la fin et ce dernier est égal à 1
- 3- 'Non reconnu' dans les autres cas
- 4- 'Nombre faux' si les nombres sont inférieurs à 1

- est_championnat : *paramètre : identifiant d'une compétition (id_competition)* - [BOOLEAN](#)

Renvoie TRUE si la fonction type_compétition (avec le même paramètre et sa valeur) est 'Championnat' et FALSE sinon.

- est_coupe : *paramètre : identifiant d'une compétition (id_competition)* - [BOOLEAN](#)

Renvoie TRUE si la fonction type_compétition (avec le même paramètre et sa valeur) est 'Coupe' et FALSE sinon.

- nombre_places_libres_total : *paramètre : identifiant d'un match (id_match)* - [INTEGER](#)

Retourne le nombre de places libres (billets non vendus) du match correspondant à l'identifiant donné.

- `nombre_places_libres_cat` : *paramètres : identifiant d'un match (`id_match`) et une catégorie de places (`type_place`)* - **INTEGER**

Retourne le nombre de places libres (billets non vendus) du match correspondant à l'identifiant donné et qui sont de la catégorie donnée en paramètre.

- `participe_a_la_competition` : *paramètres : identifiant d'un club (`id_club`), celui d'une compétition (`id_competition`) et l'année de début d'une saison (`annee_debut`)* - **BOOLEAN**

Retourne TRUE si un club donné participe à une compétition donnée lors d'une saison donnée (le tous donnés en paramètres) et FALSE sinon.

- `deja_present` : *paramètres : identifiant d'un club (`id_club`), celui d'une compétition (`id_competition`), celui d'une phase (`id_phase`) et l'année de début d'une saison (`annee_debut`)* - **BOOLEAN**

Retourne TRUE si un club donné participe déjà à une phase donnée d'une compétition donnée lors d'une saison donnée (le tous donnés en paramètres) et FALSE sinon.

- `surcharge` : *paramètres : identifiant d'un club (`id_club`) et une date* - **BOOLEAN**

Retourne TRUE si un club donné (en paramètre) joue trois matchs aux alentours d'une semaine (+/- 7 jours de la date) de la date donnée (en paramètre) et FALSE sinon.

- `a_deja_un_match` : *paramètres : identifiant d'un club (`id_club`) et une date* - **BOOLEAN**

Retourne TRUE si un club donné (en paramètre) joue déjà un match aux alentours (+/- 2 jours de la date) de la date donnée (en paramètre) et FALSE sinon.

- `max_match` : *paramètres : identifiant d'une compétition (`id_competition`), celui d'une phase (`id_phase`) et l'année de début d'une saison (`annee_debut`)* - **BOOLEAN**

Retourne TRUE si une phase donnée d'une compétition donnée lors d'une saison donnée (le tous donnés en paramètres) a atteint le nombre maximal de match qu'elle doit (normalement) contenir et FALSE sinon.

- `modifier_prix` : *paramètres : identifiant d'un match (`id_match`), une catégorie de places (`type_place`) et un prix (nouveau)* - **VOID**

Modifie le prix des billets d'une catégorie donnée d'un match donné en lui attribuant un prix donné (le tous donnés en paramètres).

- `recherche_matches` : *paramètres : identifiant d'un club (`id_club`), celui d'une compétition (`id_competition`), celui d'une phase (`id_phase`) et l'année de début d'une saison (`annee_debut`) où l'identifiant du club est en dernier* - **VOID**

Permet la recherche d'un match selon plusieurs critères qui sont : la compétition, la phase de la compétition, la saison et le club. Les critères peuvent être utilisés en même temps (ils le tous sont à la base sauf si la valeur donnée en paramètre est null).

- `reserver_match` : *paramètres : identifiant d'un match (`id_match`), celui d'un client (`id_client`) et une catégorie de places (`type_place`)* - VOID

Permet de réserver un billet d'une catégorie donnée d'un match donné pour un client donné (le tous donnés en paramètres). Cela est effectué seulement dans le cas où il reste encore des billets de la catégorie donnée.

- `reserver_matches_competition` : *paramètres : identifiant d'une compétition (`id_competition`) l'année de début d'une saison (`annee_debut`), l'identifiant d'un club (`id_club`) et celui d'un client (`id_client`)* - VOID

Permet de réserver des billets de matchs d'une compétition donnée lors d'une saison donnée d'un club donné pour un client donné (le tous donnés en paramètres). Cela est effectué seulement dans le cas où il reste encore des billets pour tous les matchs de la compétition de la saison donnée.

- `annuler_reservation` : *paramètres : identifiant d'un match (`id_match`) et celui d'un client (`id_client`)* - VOID

Permet d'annuler la réservation d'un billet match donné pour un client donné (le tous donnés en paramètres). Dans le cas où le client n'a pas de réservation pour le match, rien ne se produit.

- `reporter_match` : *paramètres : identifiant d'un match (`id_match`) et une date (nouvelle date)* - VOID

Permet de reporter un match donné pour une date donnée (le tous donnés en paramètres). La modification se produit si les deux clubs concernés sont libres à la date donnée (la fonction `a_deja_un_match` pour les deux clubs avec la date donnée renvoie FALSE).

- `stats_match` : *paramètres : identifiant d'un match (`id_match`)* - VOID

Donne des statistiques d'un match donnée (en paramètre). Les informations retournées sont :

- 1- Le nombre de places prises (billets vendus)
 - 2- Les nombres de place totale
 - 3- La somme totalisé de la vente des billets
- `stats_club` : *paramètres : identifiant d'un club (`id_club`) et l'année de début d'une saison (`annee_debut`)* - VOID

Donne des statistiques d'un club donnée pour une saison donnée (le tous donnés en paramètres). Les informations retournées sont :

- 1- Le nombre de places prises (billets vendus)
 - 2- Les nombres de place totale
 - 3- Le nombre de matchs joués
 - 4- La somme totalisé de la vente des billets
- stats_competition : *identifiant d'une compétition (id_competition) et l'année de début d'une saison (annee_debut) - VOID*

Donne des statistiques d'un club donnée pour une saison donnée (le tous donnés en paramètres). Les informations retournées sont les mêmes que celles des statistiques d'un club.

Les Triggers

Les triggers sont comme suit :

- ajout_places : *après l'ajout d'un stade*

Ajoute dans la table 'place' les places du stade ajouté (du même nombre que la capacité du stade).

- rajout_places : *après la modification de la capacité d'un stade (nombre augmenté)*

Rajoute dans la table 'place' les places du stade modifié (du même nombre que la différence entre l'ancienne capacité et la nouvelle capacité du stade).

- annulation_modification_stade : *avant la modification de la capacité d'un stade (nombre réduit)*

Empêche la modification (par choix). Pour réduit la capacité d'un stade, il faut supprimer les places une à une.

- verification_place : *avant l'insertion ou la mise à jour d'une place*

Si l'opération est l'insertion alors il vérifie que le nombre de places n'a pas atteint la capacité du stade auxquelles elles correspondent. Si le nombre est atteint alors on ne l'ajoute pas. De façon générale (insertion ou mise à jour), il vérifie que le numéro de la place (du stade) n'est pas déjà pris.

- annulation_modification_place : *avant la modification du stade de la place*

Empêche la modification du stade d'une place (pour une question de bonne gestion). Il est préférable de supprimer la place et d'en créer une autre.

- maj_stade : *après la suppression d'une place d'un stade*

Décrémente la capacité du stade.

- ajout_stade : *avant l'insertion d'un club*

Si l'identifiant du stade donnée lors de l'insertion n'existe pas alors il crée un nouveau stade avec ce même identifiant (sauf si sa valeur est null, dans ce cas une autre valeur lui est affectée). Le nouveau stade a pour nom 'Stade_de' concaténé au nom du club et comme capacité 1 (pour pouvoir être augmenté sans difficulté jusqu'à la valeur souhaitée).

- verification_ajout_match : *avant l'insertion d'un match*

Vérifie que le match sur le point d'être inséré est cohérent. Le trigger vérifie que :

- ➔ Les deux clubs sont distincts
- ➔ Les deux clubs participent à la compétition pour laquelle le match se joue pour la saison donnée (grâce à la table 'Participe')
- ➔ Les deux clubs ne jouent pas déjà lors de la phase (de la compétition) spécifiée
- ➔ Les deux clubs ne jouent pas un autre match aux alentours de la date spécifiée (+/- 2 jours)
- ➔ Le nombre maximum de match de la phase (pour un championnat c'est le nombre d'équipes/2 et pour une coupe c'est le nombre d'équipes qu'il reste à la fin de la phase) spécifiée n'est pas atteint

Si tous les tests sont passés alors l'ajout s'effectue, sinon un message apparaît.

- ajout_billets : *après l'insertion d'un match*

Ajoute les billets correspondant au match. Le nombre de billets générés est égale à la capacité du stade de l'équipe qui accueille. Chaque billet correspond à une place du stade.

- verification_billet : *avant l'insertion ou la mise à jour du billet*

Comme pour place, le trigger vérifie que le nombre maximal de billets du match n'est pas atteint et que la place du billet ne correspond pas à un billet déjà existant.

- Annulation_modification_billet : *avant la mise à jour autre que le client du billet*

Empêche la modification des valeurs de la place et du match du billet.

- notifier_reservation : *après la mise à jour de l'identifiant du client dans la table 'billet'*

Notifie au client (par la voie d'un message – table messagerie) que la réservation ou l'annulation de la réservation (et qu'il va être remboursé dans ce cas) du match s'est fait avec succès.

- annulation_modification_match : *avant la mise à jour d'un match*

Annule la modification si un autre champ que la date du match est modifié. Si seul la date est modifiée, le trigger vérifie que les deux équipes sont disponibles aux alentours de la nouvelle date.

- `notifier_report` : *après la mise à jour d'un match*

Notifie au client (par la voie d'un message – table messagerie) que le match a été reporté à une date ultérieure.

- `annulation_match` : *avant la suppression d'un match*

Supprimer les billets correspondant au match et notifie aux clients que le match a été annulé (se produit très rarement dans ma réalité, du moins dans le monde du football).

Jeux de tests

Pour pouvoir tester les fonctions et triggers implémenté, deux fichiers contiennent des requêtes d'insertion ('5_tables_filling.sql') et de mise à jour/exécution de fonctions ('6_reqs.sql').

Voici les résultats obtenus :

```
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
-----+-----+-----+-----
Remplissage de la table client
...
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
-----+-----+-----+-----
Remplissage de la table matchs
...
psql:5_tables_filling.sql:176: NOTICE:  Erreur lors de l'ajout du match ; les deux clubs saisis sont identiques.
INSERT 0 0
INSERT 0 1
psql:5_tables_filling.sql:178: NOTICE:  Erreur lors de l'ajout du match ; l'un des deux clubs participe deja a la phase de la competition.
INSERT 0 0
INSERT 0 1
INSERT 0 1
psql:5_tables_filling.sql:181: NOTICE:  Erreur lors de l'ajout du match ; l'un des deux clubs participe deja a la phase de la competition.
INSERT 0 0
psql:5_tables_filling.sql:182: NOTICE:  Erreur lors de l'ajout du match ; l'un des deux clubs ne peut participer a la competition.
INSERT 0 0
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

Figure 2 : Quelques erreurs d'insertion de matchs

```

salimo=> select count(*) from billet;
count
-----
5796
(1 row)

salimo=> INSERT INTO matchs (date_match, id_competition, id_phase, annee_debut, id_club_domicile, id_club_exterieur) VALUES ('2015-05-15', 1, 4, 2014, 1, 6);
INSERT 0 1
salimo=> select count(*) from billet;
count
-----
6212
(1 row)

```

Figure 3 : Ajout automatique de billets

```

-----
24
(1 row)

salimo=> select count(*) from club;
count
-----
24
(1 row)

salimo=> select count(*) from place;
count
-----
9502
(1 row)

salimo=> INSERT INTO club(nom_club, id_stade) VALUES ('CLUB C', 67);
INSERT 0 1
salimo=> select count(*) from stade;
count
-----
25
(1 row)

salimo=> select count(*) from club;
count
-----
25
(1 row)

salimo=> select count(*) from place;
count
-----
9503
(1 row)

salimo=> █

```

Figure 4.1 : Ajout automatique d'un stade et d'une place

id_stade	nom_stade	capacite
1	Stamford Bridge	416
2	Old Trafford	757
3	Etihad Stadium	600
4	Emirates Stadium	602
5	Anfield	453
6	White Hart Lane	363
7	St. Mary's Stadium	326
8	The Liberty Stadium	208
9	Britannia Stadium	283
10	Goodison Park	401
11	Upton Park	355
12	Selhurst Park	264
13	The Hawthorns	278
14	St James' Park	523
15	Villa Park	427
16	KC Stadium	254
17	King Power Stadium	325
18	Stadium of Light	482
19	Loftus Road	210
20	Turf Moor	226
21	Allianz-Arena	750
22	Camp Nou	997
23	Stade_de_Club A	1
65	Stade_de_CLUB B	1
67	Stade_de_CLUB C	1

(25 rows)

Figure 4.2 : Liste des stades après l'ajout du club

```

salimo=> select * from stade where id_stade = 23;
  id_stade |          nom_stade          | capacite
-----+-----+-----
      23 | Stade_de_Club A            |         1
(1 row)

salimo=> UPDATE stade SET capacite = 10 WHERE id_stade = 23;
UPDATE 1
salimo=> UPDATE stade SET capacite = 1 WHERE id_stade = 23;
NOTICE:  Modification de la capacite du stade interdite.
NOTICE:  Veuillez supprimer les places une a une.
UPDATE 0
salimo=> select count(*) from place;
 count
-----
   9512
(1 row)

salimo=> UPDATE stade SET capacite = 11 WHERE id_stade = 23;
UPDATE 1
salimo=> select count(*) from place;
 count
-----
   9513
(1 row)

```

Figure 5 : Annulation de modification de la capacité (réduction) d'un stade ainsi que l'ajout automatique d'une place après l'augmentation de la capacité du stade

```

salimo=> UPDATE place SET type_place = 'cat2' WHERE id_stade = 1 AND numero_place BETWEEN 201 AND 300;
UPDATE 100
salimo=> UPDATE place SET type_place = 'cat3' WHERE id_stade = 1 AND numero_place BETWEEN 301 AND 400;
UPDATE 100
salimo=> UPDATE place SET type_place = 'cat4' WHERE id_stade = 1 AND numero_place BETWEEN 401 AND 416;
UPDATE 16
salimo=> SELECT modifier_prix(2, 'cat1', 80);
NOTICE: Prix mis a jour.
modifier_prix
-----
(1 row)

salimo=> SELECT modifier_prix(2, 'cat2', 120);
NOTICE: Prix mis a jour.
modifier_prix
-----
(1 row)

salimo=> SELECT modifier_prix(2, 'cat3', 250);
NOTICE: Prix mis a jour.
modifier_prix
-----
(1 row)

salimo=> SELECT modifier_prix(2, 'cat4', 400);
NOTICE: Prix mis a jour.
modifier_prix
-----
(1 row)

```

Figure 6.1 : Modification de catégorie des places/ prix des billets selon les catégories des places

id_billet	prix	id_client	id_place	id_match
201	120	-1	201	2
202	120	-1	202	2
203	120	-1	203	2
204	120	-1	204	2
205	120	-1	205	2
206	120	-1	206	2

235	120	-1	235	2
236	120	-1	236	2
237	120	-1	237	2
238	120	-1	238	2
239	120	-1	239	2
301	250	-1	301	2
302	250	-1	302	2
303	250	-1	303	2
304	250	-1	304	2
305	250	-1	305	2
306	250	-1	306	2
307	250	-1	307	2
308	250	-1	308	2
309	250	-1	309	2
310	250	-1	310	2
311	250	-1	311	2
312	250	-1	312	2
313	250	-1	313	2
314	250	-1	314	2
401	400	-1	401	2
402	400	-1	402	2
403	400	-1	403	2
404	400	-1	404	2
405	400	-1	405	2
406	400	-1	406	2
407	400	-1	407	2
408	400	-1	408	2
409	400	-1	409	2
410	400	-1	410	2
411	400	-1	411	2
412	400	-1	412	2
413	400	-1	413	2
414	400	-1	414	2
415	400	-1	415	2
416	400	-1	416	2
1	80	-1	1	2

Figure 6.2 : Affichage des nouveaux prix des billets du match numéro 2

```

Requetes de reservation
...
psql:6_reqs.sql:47: NOTICE:  Reservation effectuee.
    reserver_match
-----

(1 row)

psql:6_reqs.sql:48: NOTICE:  Reservation annulee.
    annuler_reservation
-----

(1 row)

psql:6_reqs.sql:49: NOTICE:  Reservation effectuee.
    reserver_match
-----

(1 row)

psql:6_reqs.sql:50: NOTICE:  Match reporte.
    reporter_match

```

Figure 7.1 : Requêtes de réservation/annulation/report

id_message	id_client	date_message	message
1	1	2015-05-04	Votre reservation (cat1) pour le match entre Chelsea et Manchester United a ete reservee avec success.
2	1	2015-05-04	Votre reservation du match entre Chelsea et Manchester United a ete annule avec success.
3	1	2015-05-04	Vous allez etre rembourse de la somme de 80 euros.
4	1	2015-05-04	Votre reservation (cat3) pour le match entre Chelsea et Manchester United a ete reservee avec success.
5	1	2015-05-04	Le match entre Chelsea et Manchester United a ete reporte au 2015-05-29
6	2	2015-05-04	Votre reservation (cat1) pour le match entre Chelsea et Manchester City a ete reservee avec success.
7	2	2015-05-04	Votre reservation (cat1) pour le match entre Chelsea et Tottenham a ete reservee avec success.
8	2	2015-05-04	Votre reservation (cat1) pour le match entre Chelsea et Manchester United a ete reservee avec success.
9	2	2015-05-04	Le match entre Chelsea et Manchester City a ete annule.

Figure 7.2 : Messagerie des clients après les réservations / annulations / reports


```
salimo=> SELECT recherche_matches(NULL, NULL, NULL, NULL) AS RESULTATS_TOUS_LES_MATCHS;
          resultats_tous_les_matches
-----
(4,"Manchester City",Arsenal,"Premier League",Championnat,"1ere journee",2015-04-22,600)
(8,Liverpool,Tottenham,"Premier League",Championnat,"1ere journee",2015-04-22,453)
(9,"Manchester United",Liverpool,"Premier League",Championnat,"2eme journee",2015-04-28,757)
(10,Tottenham,Arsenal,"Premier League",Championnat,"2eme journee",2015-04-28,363)
(11,Chelsea,Tottenham,"Premier League",Championnat,"3eme journee",2015-05-04,415)
(12,Arsenal,"Manchester United", "Premier League",Championnat,"3eme journee",2015-05-04,602)
(13,"Manchester City",Liverpool,"Premier League",Championnat,"3eme journee",2015-05-05,600)
(14,Chelsea,"F.C. Barcelone", "Champions League",Coupe,"1/2 finale",2015-05-12,416)
(15,"Manchester United", "Bayern Munich", "Champions League",Coupe,"1/2 finale",2015-05-13,757)
(2,Chelsea,"Manchester United", "Premier League",Championnat,"1ere journee",2015-05-29,414)
(10 rows)

salimo=> SELECT recherche_matches(1, 1, 2014::smallint, NULL) AS RESULTATS_MATCHS_PL_J1_2014;
          resultats_matches_pl_j1_2014
-----
(4,"Manchester City",Arsenal,"Premier League",Championnat,"1ere journee",2015-04-22,600)
(8,Liverpool,Tottenham,"Premier League",Championnat,"1ere journee",2015-04-22,453)
(2,"Manchester United", "Premier League",Championnat,"1ere journee",2015-05-29,414)
(3 rows)

salimo=> SELECT recherche_matches(NULL, NULL, 2014::smallint, 1) AS RESULTATS_MATCHS_CHELSEA_2014;
          resultats_matches_chelsea_2014
-----
(11,Chelsea,Tottenham,"Premier League",Championnat,"3eme journee",2015-05-04,415)
(14,Chelsea,"F.C. Barcelone", "Champions League",Coupe,"1/2 finale",2015-05-12,416)
(2,Chelsea,"Manchester United", "Premier League",Championnat,"1ere journee",2015-05-29,414)
(3 rows)
```

Figure 8 : Recherche sur plusieurs critères (compétition, phase, année, club)

```
salimo=> SELECT (stats_match(2)).nombre_places_prises, (stats_match(2)).nombre
_places_total, (stats_match(2)).somme;
nombre_places_prises | nombre_places_total | somme
-----
2 | 416 | 330
(1 row)
```

Figure 9.1 : statistiques sur un match

```
statistiques du club numero 1 (Chealsea) pour la saison 2014
...
nombre_places_prises | nombre_places_totale | nombre_matches_joues | somme
-----
7 | 1248 | 3 | 640
(1 row)
```

Figure 9.2 : statistiques sur un club

```
statistiques de la competition numero 1 (Premier League) pour la saison 2014
...
nombre_places_prises | nombre_places_totale | nombre_matches_joues | somme
-----
7 | 4207 | 8 | 640
(1 row)
```

Figure 5.3 : statistiques sur une compétition