

# CHALLENGER

## EXTRACTEUR DE DONNEES / CQL

### INTRODUCTION

L'extracteur de données donne la possibilité par le biais de commandes simples (dans un langage nommé **CQL** : **Challenge Query Language**), d'obtenir un ensemble de données visuellement utilisable et exportable. Les liens avec les différentes tables se font de manière automatique (ou tout du moins de manière très simplifiée), et les commandes permettent, si cela est désiré de grouper et/ou filtrer les données par le biais de « contraintes ». Les champs à afficher sont par ailleurs paramétrables.

### LEXIQUE

<b>Table</b>	Entité primaire contenant l'ensemble des données identifiées par des champs
<b>Champ</b>	Élément secondaire composant les tables et contenant des données textuelles
<b>Groupe</b>	Table permettant de décomposer les données en sous tableaux ayant chacun la même valeur pour ce groupe
<b>Sous-groupe</b>	Champ permettant de regrouper les mêmes données d'un groupe de données
<b>Case</b>	Expression booléenne renseignée par l'utilisateur et identifiée par un champ
<b>Contrainte</b>	Expression permettant de filtrer les données
<b>Lien fort / Lien faible</b>	Un lien fort lie des données si et seulement si elles existent simultanément, un lien faible permet de récupérer les données parents mêmes si les données liées n'existent pas
<b>Cardinalité</b>	Nombre d'éléments composant une entité

### SYNTAXE DES COMMANDES

Les commandes s'écrivent les unes à la suite des autres sur une ligne, chaque commande est composée de sa lettre de groupe et de différents paramètres. L|param1:param2:param3

On peut regrouper plusieurs commandes du même groupe sur une seule ligne en les séparant par des barres verticales L|param11:param12:param13|param21|param31:param32

## GROUPES DE COMMANDES

<b>T</b>	Ajout d'une table	Lien fort
<b>T_ / TL</b>		Lien faible
<b>G / G0</b>	Ajout d'une table de groupe (lien fort ou faible suivant la syntaxe)	Min cardinalité = 0
<b>G1</b>		Min cardinalité = 1
<b>G2 / GN</b>		Min cardinalité = 2
<b>S / S1</b>	Définition d'un champ de sous-groupe	Min cardinalité = 1
<b>S2 / SN</b>		Min cardinalité = 2
<b>F</b>	Filtrage des champs affichés	
<b>O</b>	Omission d'un champ affiché	
<b>W</b>	Déclaration d'une case	
<b>C</b>	Restriction des données via une contrainte	
<b>B</b>	Déclaration des attributs de tri	

## TABLES DISPONIBLES ET DES TABLES LIEES

Les tables dont le nom est en violet ne peuvent pas être utilisées en tant que table de groupe ou en tant que table principale

Tables	Description
<b>chambres</b>	Liste des chambres avec les infos sur les propriétaires
chambres_participants	
<b>chambres_participants</b>	Table liant les chambres aux participants
chambres	
participants	
<b>concurrents</b>	Liste des concurrents (equipes / sportifs) associés aux tournois (sports)
equipes	<i>Nul si sportif défini</i>
sportifs	<i>Nul si équipe définie</i>
sports	
phases_concurrents	
<b>configurations</b>	Liste des configurations (constantes)
<b>connexions</b>	Liste des connexions des utilisateurs
utilisateurs	
<b>contacts</b>	Liste des utilisateurs affichés dans la page de contact
utilisateurs	
<b>droits_admins</b>	Liste des modules accessibles aux différents utilisateurs
utilisateurs	

<b>droits_ecoles</b>	Liste des écoles accessibles aux différents utilisateurs
ecoles	
utilisateurs	
<b>ecoles</b>	Liste des écoles et des informations sur les responsables
utilisateurs	<i>(Responsable de l'école)</i>
images	
tentes	
tarifs_ecoles	
tarifs	<i>(Tarifs avec sport spécial associé uniquement à cette école)</i>
quotas_ecoles	
points	
participants	
paiements	
ecoles_sports	
droits_ecoles	
<b>ecoles_sports</b>	Table liant les écoles aux sports avec les quotas sportifs associés
ecoles	
sports	
equipes	
<b>envois</b>	Liste des messages (email / sms) envoyés
recus	<i>(Message reçu associé lors d'une réponse)</i>
participants	
messages	
<b>equipes</b>	Liste des équipes
ecoles_sports	
participants	<i>(Capitaine de l'équipe)</i>
sportifs	
concurrents	
<b>erreurs</b>	Liste des erreurs soumises par les participants
participants	
<b>groupes</b>	Liste des groupes associés à une phase (poules) d'un tournoi
phases	
phases_concurrents	
<b>images</b>	Liste des images (logo d'école)
ecoles	

<b>licences</b>	Liste des licences récupérées depuis la FFSU
<b>matches</b>	Liste des matchs dans les tournois
sites	
phases	
(concurrents)	<i>Lien non défini car il y en a deux (id_concurrent_a / id_concurrent_b)</i>
<b>messages</b>	Liste des messages associés à un (ou des) envoi(s)
envois	
<b>modeles</b>	Liste des modèles de message à envoyer (sms / email)
<b>paiements</b>	Liste des paiements des écoles
ecoles	
<b>participants</b>	Liste des participants
ecoles	
tarifs_ecoles	
visites	
sportifs	
signatures	
recus	
erreurs	
equipes	
envois	
chambres_participants	
<b>phases</b>	Liste des phases des différents tournois (sports)
sports	
phases	<i>(Phase suivante)</i>
phases_concurrents	
matches	
groupes	
<b>phases_concurrents</b>	Table liant les concurrents (sports / équipes) aux phases des tournois
phases	
groupes	
concurrents	
<b>podiums</b>	Liste des podiums des différents tournois (sports)
sports	
(concurrents)	<i>Lien non défini car il y en a quatre (id_concurrent1 / id_concurrent2 / id_concurrent3 / id_concurrent3ex)</i>
<b>points</b>	Liste des points (DD / Fairplay / Pompom) associés aux écoles
ecoles	

<b>quotas_ecoles</b>	Liste des divers quotas associés aux écoles
ecoles	
<b>recus</b>	Liste des messages reçus par les participants (ou pas !)
participants	
envois	
<b>remember</b>	Liste des tokens et vecteurs initiaux pour la connexion auto. des utilisateurs
utilisateurs	
<b>signatures</b>	Liste des signatures du RI par les participants
participants	
<b>sites</b>	Liste des sites sportifs sur lesquels se déroulent les matchs
matchs	
<b>sportifs</b>	Liste des sportifs (associés à des participants)
equipes	
participants	
concurrents	
<b>sports</b>	Liste des sports
utilisateurs	<i>(Responsable du sport)</i>
tarifs	<i>(Tarifs avec ce sport comme tarif spécial)</i>
podiums	
phases	
ecoles_sports	
concurrents	
<b>taches</b>	Liste des tâches périodiques (CRON)
<b>tarifs</b>	Liste des tarifs
sports	<i>(Sport spécial)</i>
ecoles	
tarifs_ecoles	
<b>tarifs_ecoles</b>	Table liant les écoles aux tarifs (visibilité)
tarifs	
ecoles	
participants	
<b>tentes</b>	Liste des différentes tentes, associées ou non aux écoles
ecoles	
zones	
<b>tokens</b>	Liste des actions tokens utilisés pour les actions sécurisées de l'API
utilisateurs	

<b>utilisateurs</b>	Liste des différents utilisateurs
tokens	
sports	
remember	
ecoles	
droits_admin	
contacts	
connexions	
<b>visites</b>	Liste des visites (connexions) des participants à leur page personnelle
participants	
<b>zones</b>	Liste des zones regroupant des tentes
tentes	

## CHAMPS ACCESSIBLES POUR CHACUNE DES TABLES

Tables	Champs définis dans CQL ( <u>pour le tri</u> ) / non définis mais accessibles / liens
<b>chambres</b>	<u>numero</u> , nom, prenom, login, surnom, places, telephone, email, etat, etat_clef, commentaire, relance, lit_camp, bracelet
<b>chambres_participants</b>	respo, id_chambre, id_participant
<b>concurrents</b>	id_sport, id_equipe, id_sportif
<b>configurations</b>	flag, nom, valeur
<b>connexions</b>	connexion, dernier, methode, ip, geoip, agent, id_utilisateur
<b>contacts</b>	poste, photo, id_utilisateur
<b>droits_admins</b>	module, id_utilisateur
<b>droits_ecoles</b>	id_utilisateur, id_ecole
<b>ecoles</b>	nom, abbreviation, format_long, ecole_lyonnaise, adresse, code_postal, ville, email_ecole, telephone_ecole, nom_respo, prenom_respo, email_respo, telephone_respo, nom_corespo, prenom_corespo, email_corespo, telephone_corespo, charte_acceptee, caution_recue, caution_logement, etat_inscription, malus, langue, as_code, commentaire, id_image, id_respo
<b>ecoles_sports</b>	quota_max, quota_reserves, quota_equipes, id_ecole, id_sport
<b>envois</b>	<u>date</u> , <u>to</u> , envoi, echec, tentatives, data, id_recu, id_participant, id_message
<b>equipes</b>	label, id_ecole_sport, id_capitaine
<b>erreurs</b>	message, etat, <u>date</u> , id_participant
<b>groupes</b>	nom, id_phase
<b>images</b>	token, height, width, image

<b>licences</b>	<u>nom</u> , <u>prenom</u> , <u>licence</u> , ecole, type, ia, rc, <u>as_code</u> , <u>inscription</u>
<b>matches</b>	numero_elimination, phase_elimination, <u>date</u> , duree, gagne, forfait, <u>sets_a</u> , <u>sets_b</u> , <u>commentaire</u> , <u>id_phase</u> , <u>id_concurrent_a</u> , <u>id_concurrent_b</u> , <u>id_site</u>
<b>messages</b>	type, label, <u>titre</u> , <u>contenu</u> , <u>publipostage</u>
<b>modeles</b>	<u>nom</u> , type, <u>titre</u> , <u>modele</u>
<b>paiements</b>	libelle, montant, etat, type, saisie, <u>date</u> , <u>pointe</u> , <u>commentaire</u> , <u>id_ecole</u>
<b>participants</b>	<u>nom</u> , <u>prenom</u> , sexe, sportif, licence, fanfaron, cameraman, pompom, telephone, email, <u>recharge</u> , <u>logueur</u> , <u>recharge</u> , <u>date_inscription</u> , <u>file_attente</u> , <u>hors_malus</u> , <u>carte</u> , <u>certificat_medical</u> , <u>certificat_assurance</u> , <u>id_ecole</u> , <u>id_tarif_ecole</u>
<b>phases</b>	<u>nom</u> , type, points_victoire, points_defaite, points_nul, points_forfait, cloturee, <u>prevision</u> , <u>commentaire</u> , <u>id_sport</u> , <u>id_phase_suivante</u>
<b>phases_concurrents</b>	qualifie, <u>id_phase</u> , <u>id_groupe</u> , <u>id_concurrent</u>
<b>podiums</b>	coeff, ex_12, ex_23, ex_3, <u>id_sport</u> , <u>id_concurrent1</u> , <u>id_concurrent2</u> , <u>id_concurrent3</u> , <u>id_concurrent3ex</u>
<b>points</b>	dd, pompom, fairplay, <u>id_ecole</u>
<b>quotas_ecoles</b>	<u>quota</u> , valeur, <u>id_ecole</u>
<b>recus</b>	type, from, titre, ouvert, <u>date</u> , <u>message</u> , <u>id_participant</u>
<b>remember</b>	token, hash, iv, expire, <u>id_utilisateur</u>
<b>signatures</b>	<u>id_participant</u>
<b>sites</b>	<u>nom</u> , description, latitude, longitude
<b>sportifs</b>	<u>id_participant</u> , <u>id_equipe</u>
<b>sports</b>	<u>sport</u> , <u>sexe</u> , individuel, tournoi_initie, <u>quota_inscription</u> , <u>groupe_multiple</u> , <u>id_respo</u>
<b>taches</b>	<u>nom</u> , periodicite, script, active, execution
<b>tarifs</b>	<u>nom</u> , tarif, logement, sportif, ecole_lyonnaise, format_long, <u>description</u> , <u>for_pompom</u> , <u>for_cameraman</u> , <u>for_fanfaron</u> , <u>id_sport_special</u> , <u>id_ecole_for_special</u>
<b>tarifs_ecoles</b>	<u>id_tarif</u> , <u>id_ecole</u>
<b>tentes</b>	<u>numero</u> , <u>latitude</u> , <u>longitude</u> , places, etat, <u>commentaire</u> , <u>id_zone</u>
<b>tokens</b>	token, <u>expire</u> , <u>id_utilisateur</u>
<b>utilisateurs</b>	<u>nom</u> , <u>prenom</u> , <u>login</u> , telephone, email, cas, responsable, <u>pass</u> , <u>private_token</u> , <u>public_token</u>
<b>visites</b>	<u>connexion</u> , dernier, ip, <u>agent</u> , <u>geoip</u> , <u>id_participant</u>
<b>zones</b>	<u>zone</u> , color, <u>path</u>

Toutes les tables à quelques exceptions près (connexions, licences, visites, signatures, messages, tokens, envois, recus, remember) possèdent les champs suivants : id, \_date, \_ref, \_auteur, \_message, \_etat. Seuls les enregistrements avec \_etat=active sont renvoyés par le CQL. \_date correspond à la date de modification, \_auteur à l'id de l'auteur de la modification, \_ref est l'id de la précédente modification, enfin \_message correspond à une description de la modification.

## Quelques remarques

Rien ne vous empêche d'afficher plus de champs ou d'enlever quelques uns par le biais des commandes F et O, l'ensemble des champs des tables listés ci-dessus est placé en annexe. Pour autant vous ne pourrez utiliser que les tables listées ci-dessus. Pour travailler sur les autres tables, il vous faudra soit modifier le code-source de l'outil ou alors travailler directement sur la base de données.

Les tables ne pouvant pas grouper ne peuvent pas être utilisées par le biais de la commande G. Cela est notamment du au fait que ces tables permettent principalement de lier les données entre elles et ne contiennent pas d'autres données complémentaires.

Pour autant ces tables sont vitales pour lier correctement les données. Comme le montre le diagramme suivant, il n'est pas possible de lier toutes les tables entre elles, seules certaines le peuvent. Ainsi pour afficher le tarif de chaque participant il faut lier trois tables : participants, tarifs\_ecoles et tarifs.

## LA COMMANDE « T »

La commande T permet de sélectionner les tables utilisées. La première correspond à la table principale. Elle peut avoir entre 1 et 3 arguments. Le lien s'il est précisé doit être un des alias (ou table si alias non précisé) précédemment déclarés. Les alias (ou table si alias non précisé) doivent être uniques.

<b>T   table</b>	Sélectionne/lie la table « table »
<b>T   table : lien</b>	Sélectionne/lie la table « table » et forçant le lien entre « table » et « lien »
<b>T   alias : table : lien</b>	Sélectionne/lie la table, force le lien et déclare l'alias « alias » pour « table »

**Remarque :** Pour forcer l'utilisation d'une table sans aucun lien (ni implicite, ni explicite), il suffit de mettre le caractère dans le champ « lien ». Cela duplique fortement les données mais permet d'avoir plus de souplesse ensuite avec les contraintes.

### Exemples :

- T | ecoles « Liste des écoles »
- T | participants | sportifs « Liste des sportifs »
- T | equipes | participants « Liste des équipes avec leur capitaine »
- T | participants | sp : sportifs : | equipes : sp « Liste des sportifs avec leur équipe »
- T | ecoles | ecoles2 : ecoles : # « Matrice des écoles »

## LA COMMANDE « T\_ » OU « TL »

La grande différence avec la commande « T » est la mise en place d'un lien faible entre deux tables (en terme SQL on parle alors de jointure externe gauche : LEFT JOIN). Ce qui veut dire que l'on la table à lier ne doit pas nécessairement contenir d'élément à lier. Les exemples illustrent la différence.

Si cette commande est utilisée sur la table principale alors le lien sera faible sur toutes les tables liées.

Dans le cas où le lien ne pourrait se faire, les champs de la table concernée seront vides.



<b>T_ table</b>	Sélectionne/lie la table « table »
<b>TL table</b>	
<b>T_ table:lien</b>	Sélectionne/lie la table « table » et forçant le lien entre « table » et « lien »
<b>TL table:lien</b>	
<b>T_ alias:table:lien</b>	Sélectionne/lie la table, force le lien et déclare l'alias « alias » pour « table »
<b>TL alias:table:lien</b>	

**Exemples :**

- T|participants                      « Liste des participants ...  
 TL|sp:sportifs:                      ... et pour les sportifs ...  
 TL|equipes:sp                      ... affichage de l'équipes »
- T\_|participants|equipes              « Liste des participants et pour les capitaines, de leur équipe »

**LA COMMANDE « G », « G0 », « G1 », « G2 », « GN »**

L'utilité principale de cette commande est de grouper les données pour les afficher dans des tableaux distincts. Cela rend les choses plus lisibles mais cela permet aussi de télécharger des fichiers Excel pour chacun des groupes ou bien pour l'ensemble où chaque groupe correspondra à une feuille.

Il ne peut y avoir qu'un seul groupe (le dernier déclaré). Il peut être fort (déclaration de manière similaire aux tables), ou faibles en faisant référence à un alias (ou table si alias non précisé) précédemment déclaré.

La différence entre les deux est que pour le cas du lien fort, on peut avoir des tableaux de cardinalité nulle et de manière plus sûre. La différence est infime mais peut bien dépanner ! Dans le cas d'un lien fort, l'alias (ou la table si alias non précisé) ne doit pas être identique à celui d'une table.

<b>G# groupe</b>	Groupe par la table « groupe »	Lien fort
<b>G# table:lien</b>	Groupe en forçant le lien entre « groupe » et « lien »	Lien fort
<b>G# alias:table:lien</b>	Groupe, force le lien et déclare l'alias « alias » pour « groupe »	Lien fort
<b>G# alias</b>	Groupe en utilisant la table précédemment déclarée d'alias « alias »	Lien faible

# définit la cardinalité des groupes.

<b>G ou G0</b>	Affiche tous les groupes peu importe la cardinalité	0 et +
<b>G1</b>	Affiche tous les groupes non vides	1 et +
<b>G2 ou GN</b>	Affiche tous les groupes de plus de un élément	2 et +

**Exemples :**

- T|participants                      « Liste des participants ...  
 G|ecoles                      ... groupés par école » (les écoles peuvent être vides) / Lien fort

- `T|participants|sp:sportifs|eq:equipes|sp|ecoles_sports|sports`  
`G|eq` « Liste des sportifs avec leur sport par équipes » / Lien faible

## LA COMMANDE « S », « S1 », « S2 », « SN »

La commande « S » et associés s'utilise très simplement pour regrouper des données d'un même groupe en fonction d'un champ spécifié. Elle s'utilise ainsi avec ou sans l'utilisation d'une commande de type « G ». De même que cette dernière, une seule commande « S » est autorisée et c'est donc la dernière qui sera prise en compte. Le champ spécifié peut-être un champ qui n'est pas listé dans les champs affichés plus haut.

<b>S# alias</b>	Groupement en sous-groupes suivant le champ « alias ».id
<b>S# alias.item</b>	Groupement en sous-groupes suivant le champ « alias.item »

# définit la cardinalité des sous-groupes, il ne peut pas y avoir de sous-groupe vide.

<b>S ou S1</b>	Affiche tous les sous-groupes non vides	1 et +
<b>S2 ou SN</b>	Affiche tous les sous-groupes de plus de un élément	2 et +

### Exemples :

- `T|participants` « Liste des participants ...  
`G|ecoles` ... groupés par école » (les écoles peuvent être vides) / Lien fort

## LA COMMANDE « F » / LA COMMANDE « O »

Commande utilisée pour « Filtrer » les champs affichés. Par défaut un certain nombre de champs sont prédéfinis (ceux listés en noir dans le tableau correspondant plus haut) par l'outil, l'utilisation de la commande F, remet ces champs prédéfinis à zéro. Si les champs à afficher sont un sous-ensemble des champs prédéfinis, il est préférable de se pencher sur l'utilisation de la commande O pour « Omettre » certains champs.

La commande F s'utilise de deux/trois manières différentes :

<b>F alias ou F alias.*</b>	Affiche tous les champs de la table « alias » prédéfinis par l'outil
<b>F alias.item</b>	Affiche le champ « alias.item »

La commande O s'utilise similairement à la commande F et permet d'enlever un champ affiché parmi ceux sélectionnés aussi bien de manière automatique par l'outil que par la commande F.

<b>O alias.item</b>	Omet le champ « alias.item »
---------------------	------------------------------

## LA COMMANDE « C »

Dans le but d'utiliser des « Contraintes » et donc limiter le jeu de résultats, la commande C doit être utilisée. Elle permet d'exprimer des relations entre champs ou entre un champ et une valeur étant soit une chaîne soit une valeur numérique. Cette commande possède deux ou trois arguments ; dans le cas de seulement deux arguments, la commande teste l'égalité de la valeur de ces deux expressions.

<b>C a.i:op:value</b>	Contrainte d'opération « op » entre la valeur de champ « a.i » et « value »
<b>C a.i:value</b>	Contrainte d'égalité entre la valeur de champ « ai » et « value »

Les opérateurs possibles sont les suivants et peuvent être écrits en majuscules ou non.

<b>=, ==, ===</b>	<b>=</b>
<b>&lt;&gt;, !=, !==</b>	<b>&lt;&gt;</b>
<b>&lt;</b>	<b>&lt;</b>
<b>&gt;</b>	<b>&gt;</b>
<b>&lt;=</b>	<b>&lt;=</b>
<b>&gt;=</b>	<b>&gt;=</b>
<b>like</b>	<b>LIKE</b>
<b>not like, notlike</b>	<b>NOT LIKE</b>

« value » peut prendre les différentes valeurs suivantes (ce n'est pas forcément un champ) :

<b>b.j</b>	Champ « b.j »
<b>"..."</b>	Une chaîne de caractères entre doubles guillemets
<b>###</b>	Une valeur numérique
<b>null</b>	La valeur NULL
<b>not null, notnull</b>	La valeur NOT NULL (la double négation avec NOT LIKE se supprime)

L'utilisation de multiples contraintes se cumulent, l'opérateur les liant est donc un « AND », il peut cependant être intéressant d'utiliser l'opérateur « OR » et aussi de conjuguer les deux et les priorités comme on le ferait dans une expression logique normale avec des parenthèses. Les expressions spéciales suivantes permettent de faire cela :

<b>AND</b>	Opérateur logique AND (implicite entre les contraintes)
<b>OR</b>	Opérateur OR
<b>UP</b>	Equivalent d'une ouverture de parenthèse
<b>DOWN</b>	Equivalent d'une fermeture de parenthèse

UP et DOWN correspondent à un compteur et l'utilisation erronée de ces expressions spéciales ne fera pas planter l'application, l'utilisation de DOWN alors qu'il n'y a plus de commande UP associée ne fera rien et s'il y a plus de commandes UP que de commandes DOWN alors le programme les complètera automatiquement à la fin des contraintes.

### Exemple :

- T|p:participants:  
C|UP|p.prenom:like:"Thomas"|p.recharge:>:10|DOWN  
C|OR|p.nom:notlike:"Dupont"

« Liste des participants ...

... dont le prénom est Thomas avec une recharge de plus de 10€

... ou ayant un nom de famille différent de Dupont »

## LA COMMANDE « W »

La commande W permet de créer de nouveau attribut en fonction d'une expression logique à la manière de ce qui est possible de faire avec la commande C (une seule expression !) : CASE WHEN ... THEN ... ELSE ... END AS .... Cela crée ainsi un nouvel attribut ayant deux valeurs possibles (1 ou 0) en fonction de la validité ou non de l'expression logique. La commande possède trois ou quatre arguments. Dans le cas de trois arguments c'est un opérateur d'égalité qui teste les valeurs.

<b>W name:a.i:op:value</b>	Attribut booléen « name » issu de l'opération « op » entre « a.i » et « value »
<b>W name:a.i:value</b>	Attribut booléen « name » issu de l'opération d'égalité entre « a.i » et « value »

### Exemple :

- T|p:participants:|sportifs|eq:equipes:  
 W|capitaine:eq.id\_capitaine:=:p.id  
 « Liste des participants étant associés à un sport  
 ... avec un nouvel attribut pour savoir s'ils sont ou non capitaines »

## LA COMMANDE « B »

La commande B permet de trier le jeu de données renvoyé, il s'agit ni plus ni moins de champs utilisés dans une commande ORDER BY. L'utilisation est très simple et possède un ou deux arguments

<b>B alias</b>	Tri suivant le champ « alias ».id
<b>B alias.item</b>	Tri suivant le champ « alias.item »

Le deuxième argument correspond à l'ordre de trie, il peut-être ASC ou DESC

### Exemple :

- T|p:participants:  
 B|p.recharge:DESC|p.nom|p.prenom  
 « Liste des participants triés par ordre décroissant avec la recharge »

T	FROM / JOIN	« <b>T</b> able »
T_ / TL	LEFT JOIN	« <b>T</b> able <b>L</b> eft »
G	GROUP BY	« <b>G</b> roup »
S	ORDER BY	« <b>S</b> ubgroup »
F	SELECT	« <b>F</b> ilter »
C	WHERE	« <b>C</b> onstraint »
W	CASE WHEN	« <b>W</b> hen »
B	ORDER BY	« <b>B</b> y »