

Administration du système : gestion des règles de l'assistance (mode "Firewall™")

Le mode Firewall™ (ou *Mur de feu*) de Lesahr^{Web} permet à lui seul de 'déposer' un élève exactement dans le cursus précis, sans échappatoire possible. Ce mode est exclusivement réservé à l'[Assistant d'inscription : sélection du ou des domaines et cours suivis](#) et est débrayé lors de la gestion des curriculums hors assistant.

Le Firewall™ est appuyé sur 2 grands axes :

- la gestion de filtres poussés, appelés **règles de base de Firewall™**
Une **règle de base de FireWall™** est une recherche thématique par table. Les comparaisons, égalités, exclusions sont autorisées. Le résultat est la quantité d'items qui correspondent à la requête.
- la gestion des combinaisons desdites règles, appelés **actions de Firewall™**
Une **action de FireWall™** est le résultat de l'association de une ou plusieurs règles interdisant ou obligeant un accès à un curriculum.

Règles de base du Firewall™

Il s'agit ici d'exploiter le langage poussé de MySQL pour extraire une information de la base de données. La seule opération possible est une sélection d'une population de données. Cette documentation n'ayant pas vocation d'expliquer le langage MySQL, veuillez vous référer à la documentation officielle disponible [ici](#). Cependant, dans les exemples ci-après, la base est disponible.

```
SELECT * FROM table WHERE condition
```

Cette commande générique permet de sélectionner, et donc de compter, le nombre d'éléments qui satisfont à la condition.

Pour exécuter cette commande en tenant compte du contexte dans lequel Lesahr^{Web} se trouve, vous disposez des variables génériques suivantes :

- **\$annee** : année académique en cours
- **\$matricule** : matricule de l'élève en cours
- **\$code_cours** : pointeur vers le code cours
- **\$adulte** : peut valoir 'checked' (coché) ou 'unchecked' (non coché)
- **\$degre** : degré concerné (F1, T3,...)

Vous disposez également de tous les comparateurs disponibles dans MySQL :

```
<, ≤, =, ≥, >, !, %, BETWEEN, ...
```

Exemple 1 : nous allons vérifier que l'élève en cours de traitement par l'[Assistant](#) est âgé de 7 ans. Nous avons besoin de :

- son matricule : **\$matricule**
- son année de naissance : **DN_annee** de la table **eleves**
- de l'année en cours : **\$annee**

pour construire la requête :

```
SELECT * from eleves where matricule = '$matricule' AND DN_annee = '$annee' - 7
```

Si l'élève dont le matricule = **\$matricule** a bien 7 ans au 31 décembre de l'année **\$annee**, alors le résultat sera = 1, sinon 0.

Exemple 2 : nous allons vérifier l'âge de l'élève dans la fourchette 10-14 ans inclus :

```
SELECT * from eleves where matricule = '$matricule' AND (DN_annee <= '$annee' - 10 AND DN_annee >= '$annee' - 14)
```

Autre exemple : recherche si l'élève a suivi le cours de Diction-Orthophonie (0049) les années antérieures (et donc pas cette année) :

```
SELECT * FROM curriculum WHERE matricule = '$matricule' AND code_cours = '0049' AND annee <= '$annee' - 1
```

Ce système de règle permet donc de tester des cas simple, mais aussi bien plus complexe comme :

From:

<https://www.lesahr.be/dokuwiki/> - LesahrDoc

Permanent link:

https://www.lesahr.be/dokuwiki/doku.php/admin_firewall?rev=1502282835

Last update: **20/06/2021 12:40**

