Structure de la base de données pour l'évaluation avancée

erDiagram grading_definitions { id bigint(10) PK "Accès interface à la page de 'modification de la méthode d'évaluation avancée' : /grade/grading/form/'XXX(method)'/edit.php?areaid='XXX'" areaid bigint(10) FK "jointure avec grading_area.id" method varchar(100) FK "jointure avec grading_area.activemethod si non vide ; values = guide, rubric ou checklist" name varchar(255) description longtext descriptionformat tinyint(2) "value = 1 (affichage obligatoire)" status bigint(10) "values = 10 (brouillon) ou 20 (prête à l'usage)" copiedfromid bigint(10) FK "jointure avec grading_definition.id" timecreated bigint(10) usercreated bigint(10) FK "jointure avec user.id" timemodified bigint(10) usermodified bigint(10) FK "jointure avec user.id" timecopied bigint(10) options longtext "Voir note ci-dessous" } grading_areas { id bigint(10) PK "Accès interface à la page 'paramétrage de l'évaluation avancée' : /grade/grading/manage.php?contextid=XXX&component=XXX&area=XXX" contextid bigint(10) FK "jointure avec context.id sous contrainte que contextlevel=70 (niveau module)" component varchar(100) FK "values = core_grading (???), mod_assign (activité devoir) ou mod_forum (activité forum)" areaname varchar(100) FK "values = forum (activité forum), submissions (activité devoir) ou rubrics_XXX (grille partagées ???)" activemethod varchar(100) FK "values = guide, rubric, checklist ou vide (méthode d’évaluation simple)" } user { id bigint(10) PK username varchar(100) firstname varchar(100) lastname varchar(100) } grading_instances { id bigint(10) PK definitionid bigint(10) FK "jointure avec grading_definition.id" raterid bigint(10) FK "jointure avec user.id ; correcteur assigné et/ou correcteur qui réalise l'évaluation" itemid bigint(10) FK "jointure avec assign_grades (activité devoir) ou forum_grades (activité forum)" rawgrade decimal "???" status bigint(10) "values = 0 à 3 ; statuts du flux d'évaluation" feedback longtext "???" feedbackformat tinyint(2) "value = 0 (affichage obligatoire)" timemodified bigint(10) } gradingform_rubric_criteria { id bigint(10) PK definitionid bigint(10) FK "jointure avec grading_definition.id" sortorder bigint(10) "values = 1 to n ; n étant le nombre de critères définis pour un grading_definitions.id spécifique" description longtext descriptionformat tinyint(2) "value = 0 (affichage obligatoire)" } gradingform_rubric_levels { id bigint(10) PK criterionid bigint(10) FK "jointure avec gradingform_criterion.id" score decimal "value = nombre de points max 100 avec jusqu'à 5 chiffres après la virgule" definition longtext definitionformat tinyint(2) "value = 0 (affichage obligatoire)" } gradingform_rubric_fillings { id bigint(10) PK instanceid bigint(10) FK "jointure avec grading_instance.id" criterionid bigint(10) FK "jointure avec gradingform_criterion.id" levelid bigint(10) FK "value = niveau sélectionné par le correcteur ; jointure avec gradingform_levels.id" remark longtext "value = feedback écrit par le correcteur pour ce critère" remarkformat tinyint(2) "value = 0 (affichage obligatoire)" } assign_grades { id bigint(10) PK assignment bigint(10) FK "jointure avec assign.id ('assignement' est l'ancien nom du module 'assign')" userid bigint(10) FK "jointure avec user.id" timecreated bigint(10) timemodified bigint(10) grader bigint(10) FK "jointure avec user.id ; dernier correcteur" grade decimal "value = nombre de points max 100 avec jusqu'à 5 chiffres après la virgule" attemptnumber bigint(10) FK } assign { id bigint(10) PK course bigint(10) FK "jointure avec course.id" name varchar(255) grade bigint(10) "value = note généralement de 0 à 100" } context { id bigint(10) PK } course { id bigint(10) PK } grade_item { id bigint(10) PK courseid bigint(10) FK "jointure avec course.id" categoryid bigint(10) FK "jointure avec grade_category.id" } grade_category { id bigint(10) PK courseid bigint(10) FK "jointure avec course.id" } grading_areas ||--o{ grading_definitions : "si 'activemethod' non vide > recherche de la 'grading_définition' correspondante" user ||--o{ grading_definitions : "traçabilité créateur initial + dernière personne qui a modifié" grading_definitions ||--o{ grading_instances : "Traçabilité des correcteurs (assigné ou non)" grading_definitions ||--o{ gradingform_rubric_criteria : "si 'method' =

'rubric' > recherche des critères ordonnés par champs 'sortorder'" gradingform_rubric_criteria ||--o{ gradingform_rubric_levels : "si critère définit > recherche de ses niveaux" gradingform_rubric_criteria ||--o{ gradingform_rubric_fillings : "Le correcteur évalue chaque critère" gradingform_rubric_levels ||--o{ gradingform_rubric_fillings : "Le correcteur choisi le niveau atteint pour le critère" grading_instances ||--o{ gradingform_rubric_fillings : "Consigne le statut de l'évaluation (flux)" assign ||--o{ assign_grades : "Chaque copie va pouvoir être notée" context ||--o{ assign : "Le devoir possède la référence contexte module (level=70)" context ||--o{ grading_areas : "L'activité et son grading_areas sont liées par le contexte" assign_grades ||--o{ grading_instances : "Chaque copie peut être évalué par un ensemble de correcteurs" user ||--o{ assign_grades : "Chaque participant au devoir va pouvoir recevoir une note" user ||--o{ grading_instances : "Chaque correcteur potentiel est identifié" course ||--o{ assign : "Un devoir a été créé dans le cours" course ||--o{ grade_item : "Un devoir possède un item (ou élément d'évaluation) dans la carnet de notes" course ||--o{ grade_category : "Une catégorie de notes est définie à l'échelle d'un cours" grade_category ||--o{ grade_item : "Un item (ou élément d'évaluation) relève d'une catégorie dans la carnet de notes"

Légende :

- PK = Primary Key
- FK = Foreign Key
- UK = Unique Key

Attention : checklist est un plugin additionnel !

Principes :

- Plusieurs gradings peuvent être définis pour un même grading_areas (par exemple : une grille et un guide) mais il ne peut y en avoir qu'un seul actif à la fois (activemethod).
- Une même personne (user) peut tout à fait créer/définir (created) plusieurs gradings (grading_definitions).
- Pour un critère, je peux avoir plusieurs niveaux.

Options de grading_definitions

- Valeurs possibles = '1' oui, 'null' non
- Par défaut, toutes les valeurs sont à 1.

Pour une grille

```
{"sortlevelsasc":"1","lockzeropoints":"1","alwaysshowdefinition":"1","showdescriptionteacher":"1","showdescriptionstudent":"1","showscoreteacher":"1","showscorestudent":"1","enableremarks":"1","showremarksstudent":"1"}
```

- sortlevelsasc = Ordre de tri pour les niveaux ('1' ascendant, 'null' descendant)
- lockzeropoints = Calculer la note sur la base de la note minimale possible de la grille d'évaluation
- alwaysshowdefinition = Permet aux utilisateurs de prévisualiser la grille d'évaluation (sinon, la grille ne sera visible qu'après l'évaluation)
- etc.

Pour le guide

```
{"alwaysshowdefinition":"1","showmarkspercriterionstudents":"1"}
```

From:
https://webcemu.unicaen.fr/ - **CEMU**

Permanent link:
https://webcemu.unicaen.fr/doku.php?id=moodle4:devoir:evaluation_avancee:bdd

Last update: **12/03/2026 18:51**