Core Standards Stored Procedures

1. **GetGradeLevel**

   accepts gradeKey and publicationKey as input parameters. When publicationKey is null, returns key, name and description for all records from GradeLevel table that match given gradeKey. Otherwise returns key, name and description for all grade records that are associated with given publicationKey and given gradeKey if it is not null.

2. **GetPublication**

   accepts publicationKey, publisherKey, subjectKey as the input parameters. Returns publication key, publisher key, publication date, publication version, publication isCanonical, publication description, subject key, subject name, publication url, publication status, publisher name for all publications that match given publicationKey and/or are associated with given publisherKey and/or subjectKey. Parameters publicationKey, publisherKey, subjectKey may be null. If null, publications are not filtered by a given attribute.

3. **GetPublicationBySubjectByPublisher**

   accepts publisherKey, subjectKey as the input parameters. Returns publication key, publication description, publication version for all publications that are associated with given publisherKey and subjectKey. Parameters cannot be null.

4. **GetPublisher**

   accepts PublisherKey as input parameter. Returns publisher key, publisher name and publisher url from table Publisher that match given publisherKey. When publisherKey is null, publishers are not filtered by publisherKey.

5. **GetStandard**

   accepts publicationKey, gradeKey, gradeAgnostic as input parameters. Returns standard key, standard name, standard parent key, publication key, standard description, standard tree level, grade key, standard pubkey, standard short name for all standards that match given publicationKey and/or are associated with given subjectKey and/or are not associated with any grades. Parameters publicationKey, gradeKey, gradeAgnostic may be null. If null, standards are not filtered by a given attribute.

6. **GetSubject**

   accepts subjectKey as input parameter. Returns subject key, subject name, subject code for subjects that match given subjectKey. Parameter subjectKey may be null. If null, subjects are not filtered by the subjectKey.

7. **GetSubjectByPublisher**

   accepts publisherKey as input parameter. Returns subject key, subject name for all subjects associated with a given publisher.

8. **ExportPublication**

   accepts publicationKey as input parameter. Returns all publication data (publisher, subject, version, description, standards, categories, SOCKs, grades) associated with a given key.

9. **GenerateSessionKey**

   generates unique session key.

All of the following stored procedures take in a sessionKey parameter and check whether the user associated with this session has appropriate permissions to modify data in the database.

10. **AddPublisher**
    accepts sessionKey, name, publisherKey, url as input parameters. If neither publisherKey nor name are empty nor do they already exist in Publisher table, it inserts publisherKey, name and url into the Publisher table.
11. **AddSubject**
    accepts sessionKey, name and code as the input parameters. If neither name nor code are empty, nor do they already exist in Subject table, it inserts them into Subject table setting subject key equal to code.
12. **AddGrade**
    accepts sessionKey, gradeKey, name and description as input parameters. If given gradeKey and name are not empty and gradeKey does not yet exist in the Grade table, inserts gradeKey, name and description into the Grade table.
13. **CreatePublication**
    accepts sessionKey, publicationKey, publisherKey, datePublished, version, isCannonical, description, subjectKey, subjectLabel, url as input parameters and inserts all the values into the Publication table.
14. **ModifyPublicationStatus**
    accepts sessionKey, publicationKey and publicationStatus as input parameters. Updates the status of the Publication.
15. **AddSession**
    accepts sessionKey, user ID and role and inserts them in Session table. Sets start date to current date and end date to null.
16. **Logout**
    accepts sessionKey as an input argument and sets session end date and calls Loader_Clear to delete data from Loader tables for a given session.
17. **CopyPublication**
    accepts sessionKey and publicationKey as input parameters and copies existing publication and associated categories, standards, SOCKs, benchmark grades from main tables into Loader tables and associates it with a given sessionKey.
18. **EditPublication**
    accepts sessionKey and publicationKey as input parameters and loads existing publication and associated categories, standards, SOCKs, benchmark grades into Loader tables and associates it with a given sessionKey.
19. **ImportPublication**
    accepts sessionKey as input parameter and associates publication in Loader tables with a given sessionKey.
20. **PartialImportPublication**
    accepts sessionKey as input parameter and associates publication in Loader tables with a given sessionKey.
21. **Loader_Parent**
    Accepts sessionKey as input parameter and loads all records in Loader tables associated with a given sessionKey into main tables.
22. **Loader_Main**

accepts sessionKey as input parameter. Loads records associated with a given sessionKey from Loader tables into main tables. Do not use this stored procedure directly, use Loader_Parent instead.

23. **Loader_Main_Partial**

    accepts sessionKey as input parameter. Loads records associated with a given sessionKey from Loader tables into main tables. Do not use this stored procedure directly, use Loader_Parent instead.

24. **Loader_Validate**

    accepts sessionKey as input parameter and validates data in all loader tables associated with a given session key. Verifies that there is only one publication and that publisher and subject exist in main tables, publication description is not empty, category tree levels are continuous numbers, categories are not empty, socks knowledge category and description are not empty, socks knowledge categories are not duplicated, standard levels exist in categories tree levels, standard keys are not empty, standard names are not empty, standard descriptions are not empty, standard keys are not duplicated, for each standard verifies that its key corresponds to its level and that standard parent exists.

25. **Loader_Clear**

    accepts sessionKey as input parameter and clears all records in Loader tables associated with a given sessionKey.

26. **Loader_AddPublication**

    accepts sessionKey, publisherName, publicationDescription, subject, version as input parameters and inserts a record into Loader_Publication table.

27. **Loader_AddCategories**

    accepts sessionKey, category and treeLevel as input parameters and inserts a record into Loader_Categories table. Neither of input parameters can be null.

28. **Loader_AddStandards**

    accepts sessionKey, level, key, name, description, shortName as input parameters and inserts a record into Loader_Standards table.

29. **Loader_AddBenchmarkGrades**

    accepts sessionKey, benchmark and grade as input parameters and inserts a record into Loader_BenchmarkGrades table. Neither of input parameters can be null.

30. **Loader_AddSOCKs**

    accepts sessionKey, knowledgeCategory, description as input parameters and inserts a record into Loader_SOCKs table.

31. **Loader_GetPublication**

    accepts sessionKey as input parameter and returns PublisherName, PublicationDescription, Subject and Version from Loader_Publication table for this session.

32. **Loader_GetCategories**

    accepts sessionKey as input parameter and returns Category and TreeLevel pairs from Loader_Categories table for this session.

33. **Loader_GetStandards**

    accepts sessionKey as input parameter and returns level, key, name, description, shortName for all standards in Loader_Standards table associated with a given session.

34. **Loader_GetSOCKs**

accepts sessionKey as input parameter and returns KnowledgeCategory and Description for all SOCKs in Loader_SOCKs table associated with this.

35. **Loader_GetBenchmarkGrades**
    accepts sessionKey as input parameter and returns benchmark grade pairs from loader tables for this session.
36. **Loader_EditVersion**
    accepts sessionKey, publisherName, subject, newVersion changes a version in Loader_Publication table associated with a given session, publisher, and subject. Publisher and subject may be null.
37. **Loader_EditCategory**
    accepts sessionKey, category, treeLevel and newCategory and changes category associated with a given session, category and tree level in a Loader_Categories table. Parameter category may be null.
38. **Loader_EditStandard**
    accepts sessionKey, level, standardKey, shortName, newDescription, newShortName and changes description and short name in Loader_Standards table associated with a given session, level, standard key and standard name.
39. **Loader_EditSOCK**
    accepts sessionKey, knowledgeCategory, description, newKnowledgeCategory, and newDescription changes knowledgeCategory and description in a Loader_SOCKs table associated with a given session, knowledge category and description.
40. **Loader_EditBenchmarkGrade**
    accepts sessionKey, benchmarkKey and a comma-separated list of grades and updates Loader tables to associate given benchmark key with given grades.
41. **Loader_DeleteBenchmarkGrade**
    accepts sessionKey, benchmarkKey and grade as input parameters and deletes benchmark grade association in Loader table for a given session. Delete is not allowed if it results in no grade alignment for a particular benchmark.
42. **Loader_ValidatePublicationVersion**
    accepts sessionKey and verifies that version of the publication in Loader_Publication associated with given session does not match existing version in the Publication table.
43. **Loader_GetNextPublicationVersion**
    accepts sessionKey and returns next available version number for a publication in Loader_Publication associated with given session.