|  |  |
| --- | --- |
| **Database Table** | **Description/Usage** |
| Abstract | Main table for Abstracts storage. Data is loaded into this table by DBA. |
| AbstractStatus | Abstract’s status is constantly changing by application code or database nightly jobs. |
| AbstractStatusChangeHistory | The first record for each Abstract will be inserted into this table by DBA when the Abstract is loaded into the database. AbstractStatusCode will be set to “0”. In addition, CreatedDate value will be added.  Application code will insert new records for the same Abstract each time the Abstract’s Status is changed. In addition, UpdatedDate and UpdatedBy values will be added. UpdatedBy column will contain user’s identity (UserId from dbo. aspnet\_Users table).  When nightly job inserts a new record here,  *UpdatedByDailyJob* column is set to 1.  No update or delete operations for this table. Inserts only! |
| Evaluation | Evaluation is an entity used to model the process. When evaluation process starts, a row is inserted into this table with appropriate EvaluationTypeId. IsCompete column is set to “true” when Consensus data is submitted by Coders (1B) OR Comparison is done by ODP Staff Members (2C). In addition, DateTimeEnded value is inserted when Abstract Status is changed to 1B or 2C). |
| EvaluationType | 2 different Evaluation Types exist: *Coder Evaluation (1 > 1B), ODP Staff Members Evaluation (2 > 2C).* |
|  |  |
| Submission | Main table for submitted evaluation data |
| SubmissionType | Submission Type is used to differentiate between submissions for different evaluation processes stored in the same table. 5 submission types exist: *Coder Evaluation (1 > 1A), Coder Consensus (1A > 1B), ODP Staff Members Evaluation (2 > 2A), ODP Staff Member Consensus (2A > 2B),* and *ODP Staff Member Comparison (2B > 2C).* |
| Status | StatusID column is used in multiple tables as Foreign Key (FK) to mark items *active*, *inactive*, or *deleted* instead of physical row deletion. If StatusID exists in any table as a FK, developers have to check the Status making sure the item is *Active*. |
| A\_StudyFocus | Sort column should be used on Evaluation Form.  AbstractStudyFocusSort column could be used to build a queue of Abstracts available for Coder’s review.  ShowAsAbstractTopic column is for DBA use only. It indicates which Study Focus is served as Abstract Topic.  A1\_IsEnabled, A2\_IsEnables, A3\_IsEnables are used to mark not available columns in Abstract Coding form. |
| Team | At midnight SQL job will set the status of all teams to *InActive*. When supervisor creates Team and makes mistake he set Team’s Status to *Deleted* instead of physical row deletion in Team table.  The field “UpdatedBy” in Team table is used by developers and “UpdatedByDailyJob” is used by database job.  Team Type is used to differentiate between *Coders* and *ODP Staff Members* teams stored in this table. |
| AbstractReviewList | At each moment not more than one Review List can exist. The list could be modified any time through the user interface. Review List is a subset of items from Abstract table with Status “1N”. It’s used for display purposes. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Purging

## Abstract Purge

**Nightly database job**: All Abstracts with Abstract Status Code “1” or “1A” should be reset to Abstract Status Code = “0”

## Coder Team Purge

**Nightly database job**: All Coder Teams with Status *Active* will get Status *InActive*.

# Overriding/Canceling Evaluation Process

Supervisor of each group (Coders or ODP Members) can stop (override) evaluation process. This could be done BEFORE the Abstract under evaluation gets the Status 1B or 2B. The following should be done:

## Evaluation Cleaning

Update row with appropriate EvaluationId in *Evaluation* table: set IsStopped = 1, insert User’s Id into StoppedBy column.

## Abstract Cleaning

AbstractStatusCode of Abstract that the team was working on should be reset to “0” for Coder process or to “1N” for ODP evaluation process - new record is inserted into AbstractStatusChangeHistory table.

## Team Cleaning

Set Status to *Deleted* for the team you want to clean up – update StatusID in Team table. In addition, save UpdateDate and UpdateBy (User’s identity) values.

# Evaluation Process Modelling

(Based on Abstract Workflow Diagram)

|  |  |
| --- | --- |
| **Abstract Status Code** | **Actions** |
| 0 | DBA loads data into *Abstract* table and inserts a row for each Abstract into *AbstractStatusChangeHistory* table with the following column values:  AbstractStatusChangeHistoryID – identity column  AbstractID - FK  AbstractStatusID = 1  CreatedDate = set current date  CreatedBy = NULL  UpdatedDate = NULL  UpdatedBy = NULL  UpdatedByDailyJob = 0 |
| 1 | When Coder Process starts, Abstract’s Status Code is changed from 0 to 1. A new row is inserted into *AbstractStatusChangeHistory* table with the following column values:  AbstractStatusChangeHistoryID – identity column  AbstractID - FK  AbstractStatusID = 2  CreatedDate = NULL  CreatedBy = NULL  UpdatedDate = current time  UpdatedBy = UserId in *aspnet\_Users* table  UpdatedByDailyJob = 0  Also, a new row is inserted into *Evaluation* table (new EvaluationId) with the following values:  EvaluationTypeId = 1 (Coder Evaluation)  ConsensusStartedBy = NULL  TeamID – insert current team ID  AbstractID - insert current abstract ID  IsComplete = 0  DateTimeStarted – insert current time  DateTimeEnded = NULL  IsStopped = 0  StoppedBy = NULL  Individual Coder’s evaluation data will be saved to *Submission* table with SubmissionTypeId = 1 (Coder Evaluation). Additional tables exist in the database for storing answers with *SubmissionID* as a Foreign Key. |
| 1A | After all 3 coders saved data into *Submission* table for the same EvaluationId; Abstract’s Status Code will be changed to 1A. Application code will insert a new row into *AbstractStatusChangeHistory* table with  AbstractStatusID = 3 |
| 1B | On Coders’ Consensus starting application code will update row for the current EvaluationId and set ConsensusStartedBy = UserId.  On Coders’ Consensus submission application code will do the following:  Validate that UserId is the same as ConsensusStartedBy value in *Evaluation* table – only the user who started consensus is allowed to submit consensus data;  Consensus data is saved into *Submission* table for the same EvaluationId and SubmissionTypeId = 2 (Coder Consensus);  Update row for the same EvaluationId in *Evaluation* table, set IsComplete = 1, DateTimeEnded = current time;  Insert a new row into *AbstractStatusChangeHistory* table with AbstractStatusID = 4 |
| 1N | When Coder Supervisor uploads Scanned evaluation forms, application code will do the following:   1. Insert row into *AbstractScan* table   AbstractScanID – identity column  EvaluationId – FK  FileName – Naming convention: concatenate AbstractID, current date, and add suffix “\_coder”  UpdatedBy = UserId  UpdatedDateTime = current time  The same scan could be uploaded again (overwritten on hard drive);   1. Insert a new row into *AbstractStatusChangeHistory* table with AbstractStatusID = 6 |
| 2 | When ODP Staff Process starts, Abstract’s Status Code is changed from 1N to 2. A new row is inserted into *AbstractStatusChangeHistory* table with the following column values:  AbstractStatusChangeHistoryID – identity column  AbstractID - FK  AbstractStatusID = 7  CreatedDate = NULL  CreatedBy = NULL  UpdatedDate = current time  UpdatedBy = UserId in *aspnet\_Users* table  UpdatedByDailyJob = 0  Also, a new row is inserted into *Evaluation* table (new EvaluationId) with the following values:  EvaluationTypeId = 2 (ODP Staff Members Evaluation)  ConsensusStartedBy = NULL  TeamID – insert current team ID  AbstractID - insert current abstract ID  IsComplete = 0  DateTimeStarted – insert current time  DateTimeEnded = NULL  IsStopped = 0  StoppedBy = NULL  Individual ODP Staff Member’s evaluation data will be saved to *Submission* table with SubmissionTypeId = 3 (ODP Staff Members Evaluation). Additional tables exist in the database for storing answers with *SubmissionID* as a Foreign Key. |
| 2A | After all 3 ODP Staff Members saved data into *Submission* table for the same EvaluationId; Abstract’s Status Code will be changed to 2A. Application code will insert a new row into *AbstractStatusChangeHistory* table with  AbstractStatusID = 8 |
| 2B | On ODP Staff Member’s Consensus starting application code will update row for the current EvaluationId and set ConsensusStartedBy = UserId.  On ODP Staff Member’s Consensus submission application code will do the following:  Validate that UserId is the same as ConsensusStartedBy value in *Evaluation* table – only the user who started consensus is allowed to submit consensus data;  Consensus data is saved into *Submission* table for the same EvaluationId and SubmissionTypeId = 4 (ODP Staff Member Consensus);  Insert a new row into *AbstractStatusChangeHistory* table with AbstractStatusID = 9 |
| 2C | On ODP Staff Member Comparison starting application code will validate that the Abstract Status Code is 2B and that UserId is the same as the value in ConsensusStartedBy column in evaluation table for the current process.  On ODP Staff Member Comparison submission application code will do the following:  Validate that UserId is the same as ConsensusStartedBy value in *Evaluation* table – only the user who started consensus is allowed to submit consensus data;  Consensus data is saved into *Submission* table for the same EvaluationId and SubmissionTypeId = 5 (ODP Staff Member Comparison);  Update row for the same EvaluationId in *Evaluation* table, set IsComplete = 1, DateTimeEnded = current time;  Insert a new row into *AbstractStatusChangeHistory* table with AbstractStatusID = 10 |
| 2N | When ODP Staff Supervisor uploads Scanned evaluation forms, application code will do the following:   1. Insert row into *AbstractScan* table   AbstractScanID – identity column  EvaluationId – FK  FileName – Naming convention: concatenate AbstractID, current date, and add suffix “\_odp”  UpdatedBy = UserId  UpdatedDateTime = current time  The same scan could be uploaded again (overwritten on hard drive);   1. Insert a new row into *AbstractStatusChangeHistory* table with AbstractStatusID = 12 |
| 3 |  |
| 4 |  |
|  |  |