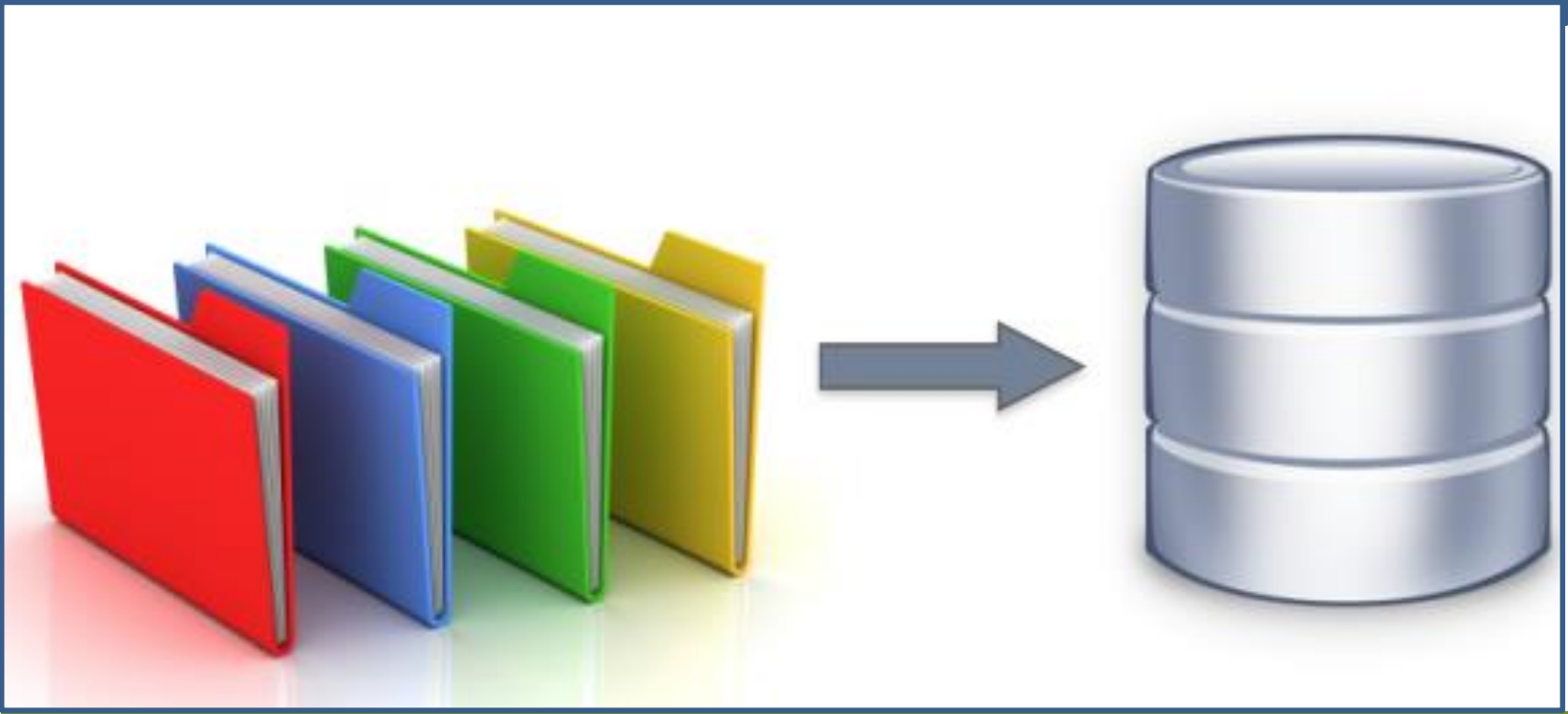
Syracuse University

Office of the Chief Information Officer

USPTO Systems Development Life Cycle

System Design Document
for
Supplemental Complex Repository for Examiners
(SCORE)



Project Design: Debasis Chatterjee (Data Analyst)
IST 659 Data Admin Concepts & Database Management
Student ID: 233176962
Date: 03/29/2019

Table of Contents:

1. Business Description (PBO)

SCORE is a Web-based system developed under the direction of the United States Patent and Trademark Office (USPTO) as specified by the Chief Information Officer (CIO) for Automated Information Systems (AIS)

The purpose of the Supplemental Complex Repository for Examiners (SCORE) 3.2 is to provide users with access to unpublished non-image application data and files that cannot be scanned into the Image File Wrapper (IFW) repository in the tagged image file format (TIFF) because of their file size or type. These files contain sequence listings with millions of pages, tables, or biotechnology information that requires specific file types and has specific viewing requirements. USPTO patent examiners, other USPTO staff, and applicants can then use SCORE to access these application files. USPTO benefits from SCORE include consolidating data management and increasing efficiency in data flow and quality control.

This document provides details for the design of the Supplemental Complex Repository for Examiners (SCORE) 3.2 based on the system requirements approved by the United States Patent and Trademark Office (USPTO).

2. Problem Description (PD)

The intention behind kicking of this project is to overcome the current challenges users and business are facing with the system. Slag Performance and limited functionalities which are identified after the application is developed and went on live. Challenges or problem areas are described as below -

This application is using Cassandra database at back end to store metadata and file-store to store files. Cassandra is a non-relational database but it gives maximum up time because of its server architecture. It's through put is max and downtime is almost tends to zero. Even though it has good features, drawback is revealed a year later when new requirements started coming in.

The main challenge with Cassandra is to decide first what kind of queries can be asked against database. Once the database is designed and start using it, there no much changes can be done and no one can make any different kind of query against the database as it was not captured before datamodel was designed. Cassandra needs fields to be part of culture key to make query. Cassandra can't do any kind of joining or in query.

So ultimately business landed up in a bottleneck situation as they can't enhance functionalities. USPTO generates several reports and does heavy analysis on business data on Patents regular basis and they have different needs on time. Based on international treaty and users' need application functionalities gets changed frequently which need to be accommodate at earliest.

3. Solution Overview:

Chief Information Officer (CIO) for Automated Information Systems (AIS) decided to hire data analyst to find optimal database solution for the business needs with limited cost as they have already spent huge money on Cassandra.

USPTO was need a data solution would take minimum license cost, Data independence, efficient data access, Data integrity and security, Easy Data administration, Concurrent access and crash recovery and reduced application development time.

So as first step it was required to know how business performs and to understand their need daily basis. CIO and other stake holders like Scientific and Technology Information Center (STIC), Examiner (bio-sequence), end users were involved to explained their needs and describes minute attributes of the business process.

After lots of analysis Microsoft SQL server was proposed for optimal database solution. The best part of the proposed database is **cost (pretty pricey).** It provides multiple level of security (which was lack in Cassandra). It provides excellent data recovery at Enterprise-grade management software. It supports all relational functionalities, no limitation on queries and any kind of

analysis can be done on data. MS SQL data can be pulled into MS Access for report generation and MS Access form wizard can be for designing user form and query designer can be used to make any complex query on relations with limited time and with limited knowledge on database query.

Requirement of the business process was provided by the stake holders at minute level. Some top few business problem are captured as below for which CIO is looking for new solution –

- Business want a dashboard which will give entire data in one glance which will display the relation and information on image or bio-sequence application for major version, minor version, their international WIPO standard, file format, decision of the application and publication status.
- How many applications are certified or rejected over time.
- How many Specific standard of bio-sequence have been submitted over time.
- How many bio-sequence patent has been approved over time.
- How many bio-sequence was edited over time.
- Projection of how many bio-sequence has been submitted after certain period.

Above analysis will help CIO to decide how much STIC or examiners are required time to time to support patent application and it will help them on resource optimization at the same time it helps PTO to respond to users query on their filing status for a patent.

To support above business needs, it was interpreted in database query term to project how it will solve the needs. The interpretation of business need in terms of database query is converted as below

- Complete overview report of Bio-Sequence from submission to export/publication
- How many documents are certified or rejected in last one month By application number
- How many ST26 was loaded into system in last one year by projected year
- How many certified sequencelisting have been exported to ABSS after march 1st 2019.
- How many sequencelisting header have been edited in last one yea
- Provide the list of the sequencelisting or total count of sequencelisting are in the system have not been reviewed yet where sequencelisting is loaded after Jan 1, 2019

4. DATA-Modeling:

All the data CIO expects to store and display in dashboard are metadata of the application. But it was the responsibility of data analyst to understand the data, its use and its patterns. So gathered all the data and started removing multivalued dependencies separate tuples. Then all functional dependencies are identified and group of functions are separated out in different tuples. All transitional dependencies (where non key fields are depends on another non key attributes) and separate out in different tuple without dropping it out to avoid data loss. So finally business needs are captured in conceptual data model → logical data model.

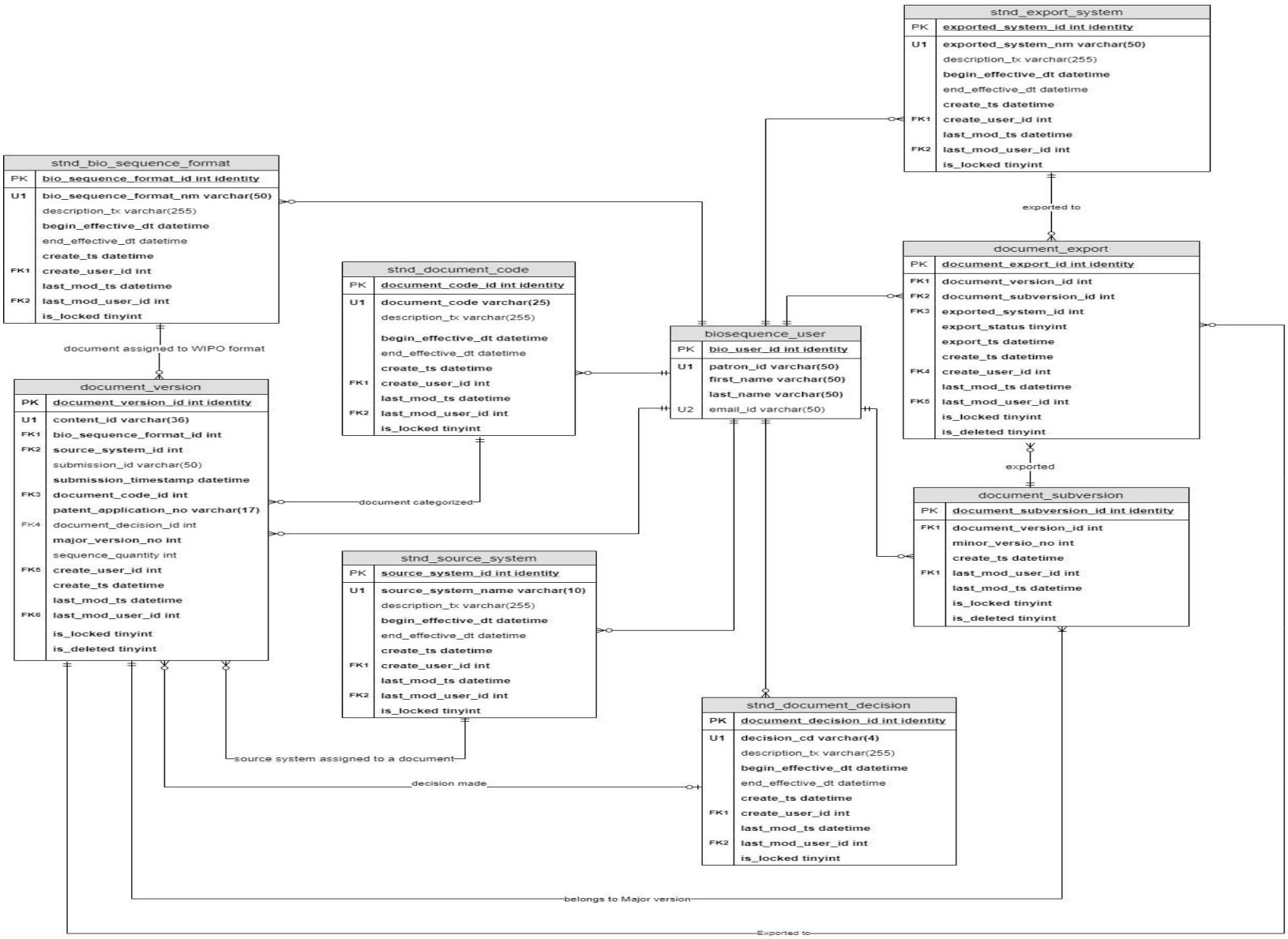
5. Business Rules:

In this Bio-sequence business process when application is filed USPTO does not change/modify anything in the original file. System stores this as 1.0 version. System created another copy of it and split the file into .header, .body and .index files and it created version 1.1. Every submitted bio-sequence file should follow some WIPO standard ST23/25/26. Every format is having specific category of file like ST23/25 are text files ST26 are XML files. USPTO assigns SEQ.TXT.SUPP and SEQ.XML.SUPP to identify ST23/25 and ST26 respectively. After application is submitted for patent PTO assigns an application number for applicant. The application can user multiple source to submit the file LIKE EFSWeb (Electronic filing system), OIPE (Office of Initial Patent Examination). IB (International bureau) e.t.c. Each upstream system assigns system generated submission number to the application so that filling process can be tracked down through the process. Before a patent is issues by USPTO to an application files are reviewed by examiners/STIC. They may approve or reject the application. If they approve/certify it is called CRFE and if they reject it is called CRFD. If there is any issue in the application header, examiner is allowed to just change the application header but not the body part of the application because the body explains the details for which application is filed. Once the header is edited by examiner new document is stored in the system as next minor version 1.2. If there is application is rejected USPTO send mail to application on with show cause. If the application is certified by STIC/examiner then patent is approved and it is published (in technical term application is exported to publication site)

Based on the above business process/rules below are the identified tuples/tables following 1-2-3 normal formal to avoid data redundancy and better storage optimization so that database query could be faster.

stnd_bio_sequence_format (Where the format of the ST23/25/26 is defined)
stnd_document_code (Category of file is defined with doc code like SEQ.TXT.SUPP and SEQ.XML.SUPP to identify ST23/25 and ST26 respectively)
stnd_source_system (Which system is sued to submit application like EFSWeb, IB, OIPE)
stnd_document_decision (Standard Decision code is on filed application)
stnd_export_system (Publication site where application is published after patent is approved)
document_version
document_subversion } These are the three main table where application information, minor version of the application
publication captured
document_export
biosequence_user (USPTO stake holders who are involved throughout the process)

6. ER-DIAGRAM:



7. Explanation of ER with cardinality:

stnd_bio_sequence_format – It captures standard format of the bio-sequence by USPTO users. It has effective period. One user can create multiple format in the system. One standard format can be assign to multiple document_version. But one document can have only one format.

stnd_document_code - It captures standard document code of the bio-sequence by USPTO users. It has effective period. One user can create multiple doc code in the system. One standard doc code can be assign to multiple document_version. But one document can have only one doc code.
stnd_source_system - It captures standard source system name of the bio-sequence by USPTO users. It has effective period. One user can create multiple source in the system. Application can be filed only through one system. But multiple application can be filed from same source system.
stnd_document_decision - It has standard decision code of the bio-sequence by USPTO users. It has effective period. One user can create multiple decision code in the system. One decision code can be assign to multiple document_version. But one document can have only one decision code.
stnd_export_system - It has standard publication site information of the bio-sequence by USPTO users. It has effective period. One user can create multiple publication site in the system.

document_version – Main tuple. Captures content reference information to locate the applicant file. It captures bio-sequence format, sourcesystem, submission id, submission time stamp, document code, application number, document decision, total sequence in the application, major version. One user can create multiple document_version but one document_version is created by one user.
document_subversion - Mainly it captures the minor version of the application for the same major version. One user can create multiple document_subversion but one document_subversion is created by one user.
document_export – It captures which major, minor/sub-version, when and where (publication site) and by whom it is done, when the export package is created. One user can do multiple exports but one specific is done by one user. **One publication site can be assign to multiple application/exports. But one application-export can be made to a specific publication site but same application can be published to multiple site.**

Note: It was a challenge for USPTO to capture this relationship. Which has been resolved through this solution. It will be available in dashboard later part of the document.

biosequence_user – This tuple is having Bio-Sequence stakeholder information (patron id, first name, last name email id) who are involved throughout the process.

Keys -
PK – Primary Key (auto generated by SQL server increment by 1)
FK[*i*] – Foreign Key, [*i*] is the index of the foreign key.
U[*i*] – Unique Key, [*i*] is the index of the Unique key.

Cardinalities -
- 0 or Many cardinality
- One cardinality

8. Physical DataModel: (described major attributes with data type and constraints)

biosequence_user:

```
bio_user_id   int identity, //Auto incremental primary key maintained by DB
patron_id     varchar(50) not null, //Patron ID unique for each person
first_name    varchar(50) not null, //First name of the user
last_name     varchar(50) not null, //Last name of the user
email_id      varchar(50),          //email ID

-- Constraints on the User Table
CONSTRAINT pk_biosequence_user PRIMARY KEY (bio_user_id),
CONSTRAINT u1_biosequence_user UNIQUE(patron_id),
CONSTRAINT u2_biosequence_user UNIQUE(email_id)
```

stnd_document_code:

```
    document_code_id      int identity,            //Auto incremental primary key maintained by DB
    document_code         varchar(25)      NOT NULL,//doc code SEQ.TXT.SUPP etc.
    description_tx        varchar(255),            //Description of the doc code
    begin_effective_dt    datetime         NOT NULL, //doc code effective date
    end_effective_dt      datetime,                 //doc code expire date
    create_ts             datetime         DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                      NOT NULL,
    last_mod_ts           datetime         DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id      int                      NOT NULL,
    is_locked              tinyint          DEFAULT 0 NOT NULL,

-- Constraints on the document code Table
       CONSTRAINT pk_stnd_document_code  PRIMARY KEY (document_code_id),
       CONSTRAINT u1_stnd_document_code  UNIQUE (document_code),
       CONSTRAINT fk1_stnd_document_code FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_document_code FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
```

stnd_bio_sequence_format:

```
    bio_sequence_format_id    int identity,           // Auto incremental primary key maintained by DB
    bio_sequence_format_nm    varchar(50)     NOT NULL,// format name like ST23
    description_tx            varchar(255),           // format description
    begin_effective_dt        datetime        NOT NULL, // format effective date
    end_effective_dt          datetime,                 // format expire date
    create_ts                 datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id            int              NOT NULL,
    last_mod_ts               datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id          int              NOT NULL,
    is_locked                  tinyint         DEFAULT 0 NOT NULL,
```

```
-- Constraints on the sequence format Table
       CONSTRAINT pk_stnd_bio_sequence_format  PRIMARY KEY (bio_sequence_format_id),
       CONSTRAINT u1_stnd_bio_sequence_format  UNIQUE (bio_sequence_format_nm),
       CONSTRAINT fk1_stnd_bio_sequence_format FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_bio_sequence_format FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
```

stnd_document_decision:

```
    document_decision_id  int identity,                    // Auto incremental primary key maintained by DB
    decision_cd           varchar(4)       NOT NULL,// decision code like CRFE
    description_tx        varchar(255),                    // Decision description
    begin_effective_dt    datetime        NOT NULL,        // decision code effective date
    end_effective_dt      datetime,                        // decision code expire date
    create_ts             datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                  NOT NULL,
    last_mod_ts           datetime        DEFAULT CURRENT_TIMESTAMP  NOT NULL,
    last_mod_user_id      int                  NOT NULL,
    is_locked             tinyint         DEFAULT 0 NOT NULL,

-- Constraints on the document decision Table
       CONSTRAINT pk_stnd_document_decision  PRIMARY KEY (document_decision_id),
       CONSTRAINT u1_stnd_document_decision  UNIQUE (decision_cd),
       CONSTRAINT fk1_stnd_document_decision FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_document_decision FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
```

stnd_source_system:

```
    source_system_id      int identity,                // Auto incremental primary key maintained by DB
    source_system_name    varchar(10)      NOT NULL,   // Source system name
    description_tx        varchar(255),                // Source system description
    begin_effective_dt    datetime        NOT NULL,    // Source system effective date
    end_effective_dt      datetime,                    // Source system effective date
    create_ts             datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                  NOT NULL,
    last_mod_ts           datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id      int                  NOT NULL,
    is_locked             tinyint         DEFAULT 0 NOT NULL,

-- Constraints on the Source system
       CONSTRAINT pk_stnd_source_system  PRIMARY KEY (source_system_id),
       CONSTRAINT u1_stnd_source_system  UNIQUE (source_system_name),
       CONSTRAINT fk1_stnd_source_system FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_source_system FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
```

stnd_export_system:

```
    exported_system_id    int identity,         // Auto incremental primary key maintained by DB
```

```
    exported_system_nm    varchar(50)      NOT NULL,          //Exported system name/publication site
    description_tx        varchar(255),                       // description of the publication sire
    begin_effective_dt    datetime        NOT NULL,           // publication site effective date
    end_effective_dt      datetime,                           // publication site expiry date
    create_ts             datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                  NOT NULL,
    last_mod_ts           datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id      int                  NOT NULL,
    is_locked             tinyint         DEFAULT 0 NOT NULL,

-- Constraints on the export system
       CONSTRAINT pk_stnd_export_system  PRIMARY KEY (exported_system_id),
       CONSTRAINT u1_stnd_export_system  UNIQUE (exported_system_nm),
       CONSTRAINT fk1_stnd_export_system FOREIGN KEY (create_user_id)       References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_export_system FOREIGN KEY (last_mod_user_id)     References
biosequence_user(bio_user_id)
```

document_version:

```
    document_version_id          int identity,         // Auto incremental primary key maintained by DB
    content_id                   varchar(36)    NOT NULL, // Unique to all content file
    bio_sequence_format_id       int                   NOT NULL, //refer to format id
    source_system_id             int                   NOT NULL, //refer to system id
    submission_id                varchar(50),                    // submission id assigned by upstream system.
    submission_timestamp         datetime       NOT NULL,     // submission time stamp/mailroom date
    document_code_id             int                   NOT NULL,// refer to standard document code.
    patent_application_no        varchar(17)    NOT NULL,       // application number assigned to file.
    document_decision_id         int,                           // refer to decision code like CRFE/CRFD
    major_version_no             INT            DEFAULT -1 NOT NULL,//Major version for each file like 1 or 2
    sequence_qtuantity           INT,                          //Number of sequence for the bio-sequence
    create_user_id               int                NOT NULL,
    create_ts                    datetime       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_ts                  datetime       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id             int            NOT NULL,
    is_locked                    tinyint        DEFAULT 0 NOT NULL,
    is_deleted                   tinyint        DEFAULT 0 NOT NULL,

-- Constraints on the document version
       CONSTRAINT pk_document_version  PRIMARY KEY (document_version_id),
       CONSTRAINT u1_document_version  UNIQUE (content_id),
       CONSTRAINT fk1_document_version FOREIGN KEY (bio_sequence_format_id)        References
stnd_bio_sequence_format(bio_sequence_format_id),
       CONSTRAINT fk2_document_version FOREIGN KEY (source_system_id)              References
stnd_source_system(source_system_id),
       CONSTRAINT fk3_document_version FOREIGN KEY (document_code_id)              References
stnd_document_code(document_code_id),
       CONSTRAINT fk4_document_version FOREIGN KEY (document_decision_id)          References
stnd_document_decision(document_decision_id),
       CONSTRAINT fk5_document_version FOREIGN KEY (create_user_id)                References
biosequence_user(bio_user_id),
```

```
      CONSTRAINT fk6_document_version FOREIGN KEY (last_mod_user_id)             References
biosequence_user(bio_user_id)
```

document_subversion:

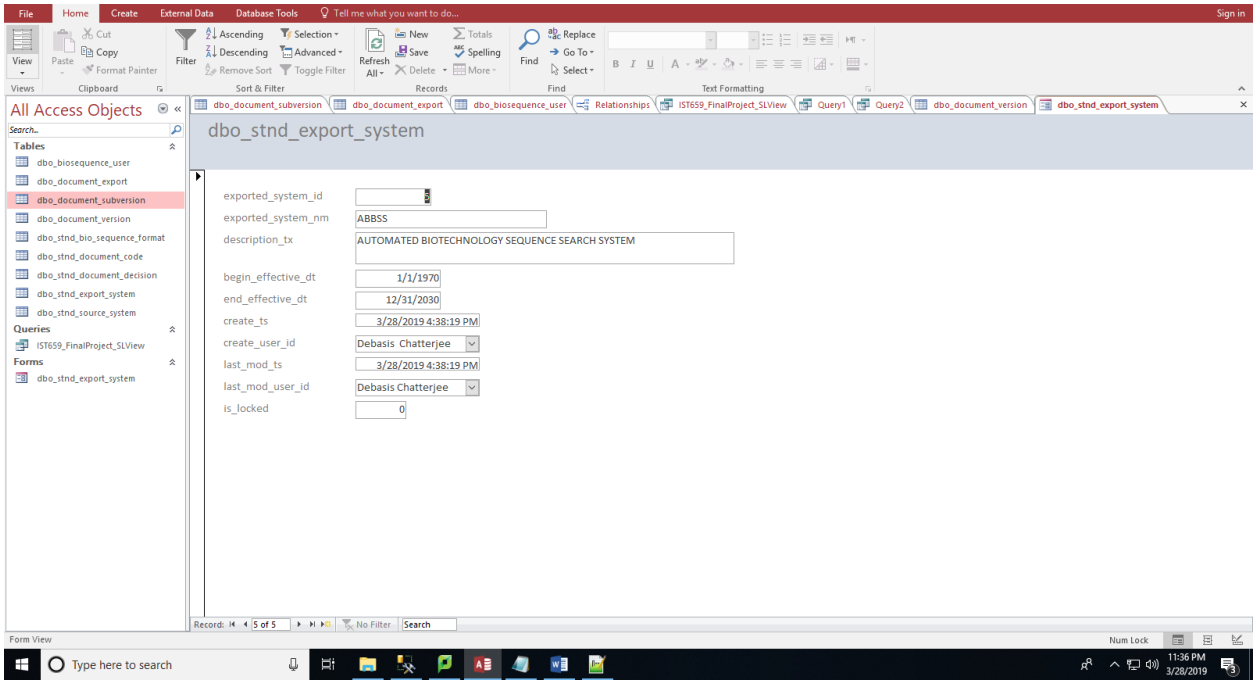
```
    document_subversion_id             int identity,  // Auto incremental primary key maintained by DB
    document_version_id                int            NOT NULL,//Refer each minor version under major
    minor_version_no                   int            NOT NULL,//minor version number
    create_ts                          DATETIME       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_ts                        DATETIME       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id                   int            NOT NULL,
    is_locked                           tinyint        DEFAULT 0 NOT NULL,
    is_deleted                          tinyint        DEFAULT 0 NOT NULL,

-- Constraints on the document subversion Table
      CONSTRAINT pk_document_subversion  PRIMARY KEY (document_subversion_id),
      CONSTRAINT fk1_document_subversion FOREIGN KEY (document_version_id) References
document_version(document_version_id),
      CONSTRAINT fk2_document_subversion FOREIGN KEY (last_mod_user_id)    References
biosequence_user(bio_user_id)
```

document_export:

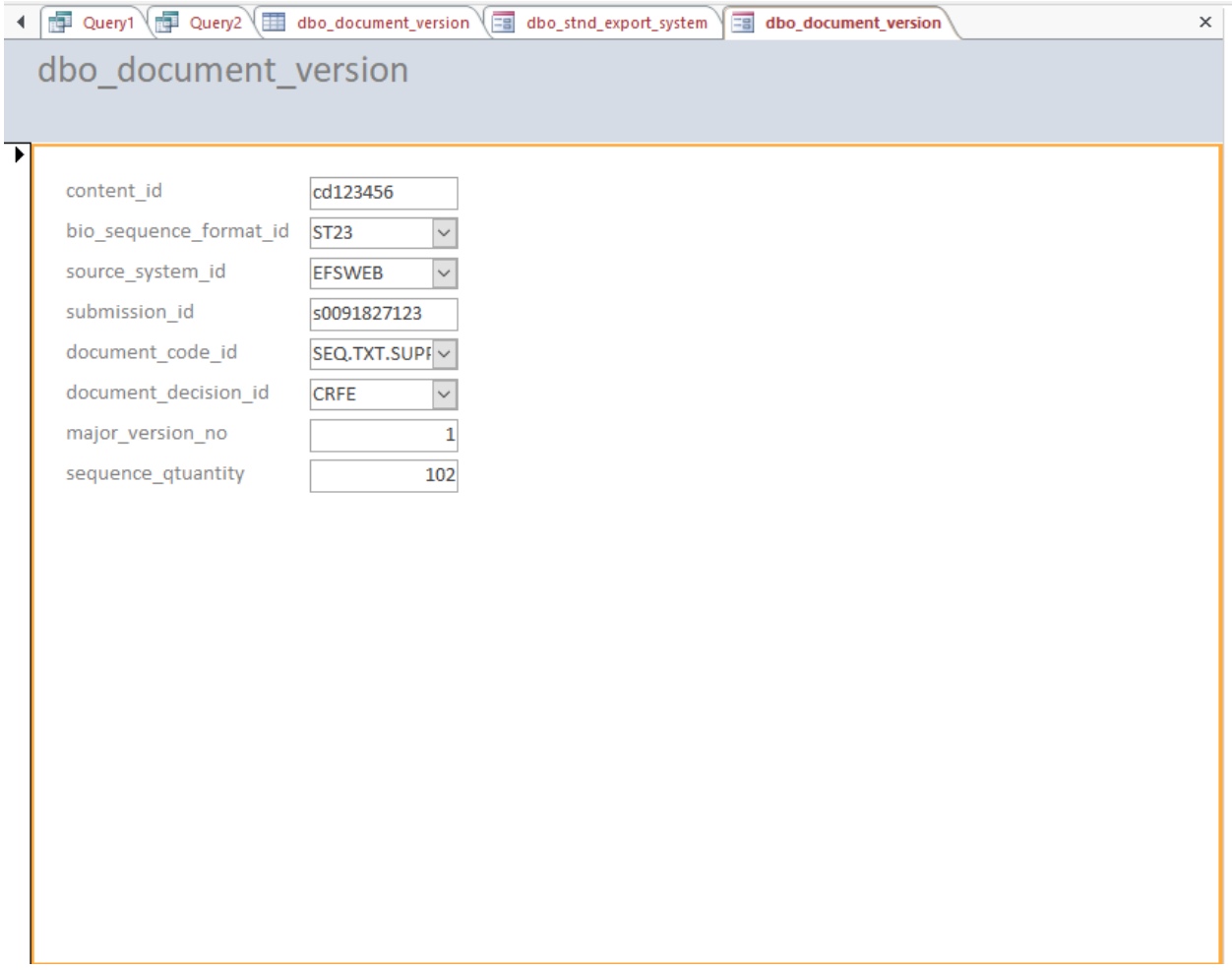
```
    document_export_id            int identity, // Auto incremental primary key maintained by DB
    document_version_id           int                 NOT NULL,// refer to document version
    document_subversion_id        int                 NOT NULL,// refer to document subversion
    exported_system_id            int                 NOT NULL,// refer where export is done
    export_status                 tinyint             NOT NULL,// export status success(1)/failure(0)
    export_ts                     datetime      NOT NULL,
    create_ts                     datetime    DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id                int               NOT NULL,
    last_mod_ts                   datetime    DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id              int               NOT NULL,
    is_locked                     tinyint     DEFAULT 0 NOT NULL,
    is_deleted                    tinyint     DEFAULT 0 NOT NULL,

-- Constraints on the document export
      CONSTRAINT pk_document_export  PRIMARY KEY (document_export_id),
      CONSTRAINT fk1_document_export FOREIGN KEY (document_version_id)     References
document_version(document_version_id),
      CONSTRAINT fk2_document_export FOREIGN KEY (document_subversion_id)  References
document_subversion(document_subversion_id),
      CONSTRAINT fk3_document_export FOREIGN KEY (exported_system_id)      References
stnd_export_system(exported_system_id),
      CONSTRAINT fk4_document_export FOREIGN KEY (create_user_id)          References
biosequence_user(bio_user_id),
      CONSTRAINT fk5_document_export FOREIGN KEY (last_mod_user_id)        References
biosequence_user(bio_user_id),
```

9. Data Questions (codes and additional screen is in appendix)

Form is designed to enter data in the standard look up table (stnd_export_system).



For the above business data questions data entry form is generated in MS Access. The same can be used for data modification. This form is designed to enter data in the document_version table. Where format, system, doc code, decision is coming from standard look up tables in a drop down.



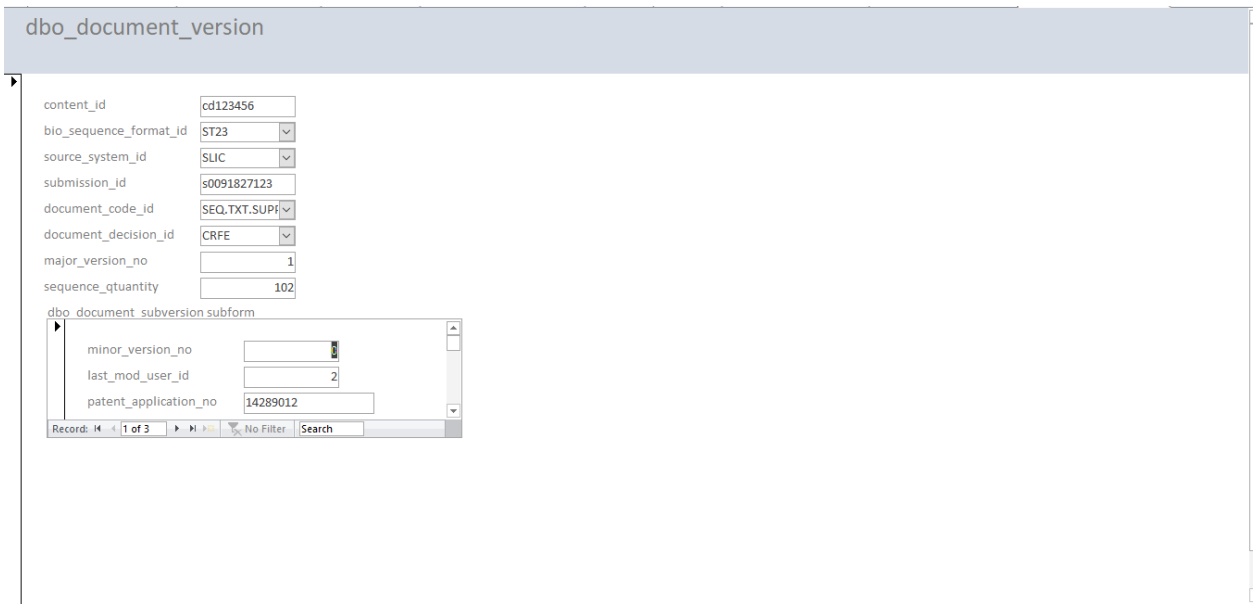
Record is entered where source system 1 means EFSWEB.

document_v	content_id	bio_sequen	source_syste	submission_
1	cd123456	1	1	s0091827123
2	cd12345a	2	2	s0091827124
3	cd12345b	3	2	s0091827125
4	cd12345c	3	3	s0091827127
(New)				

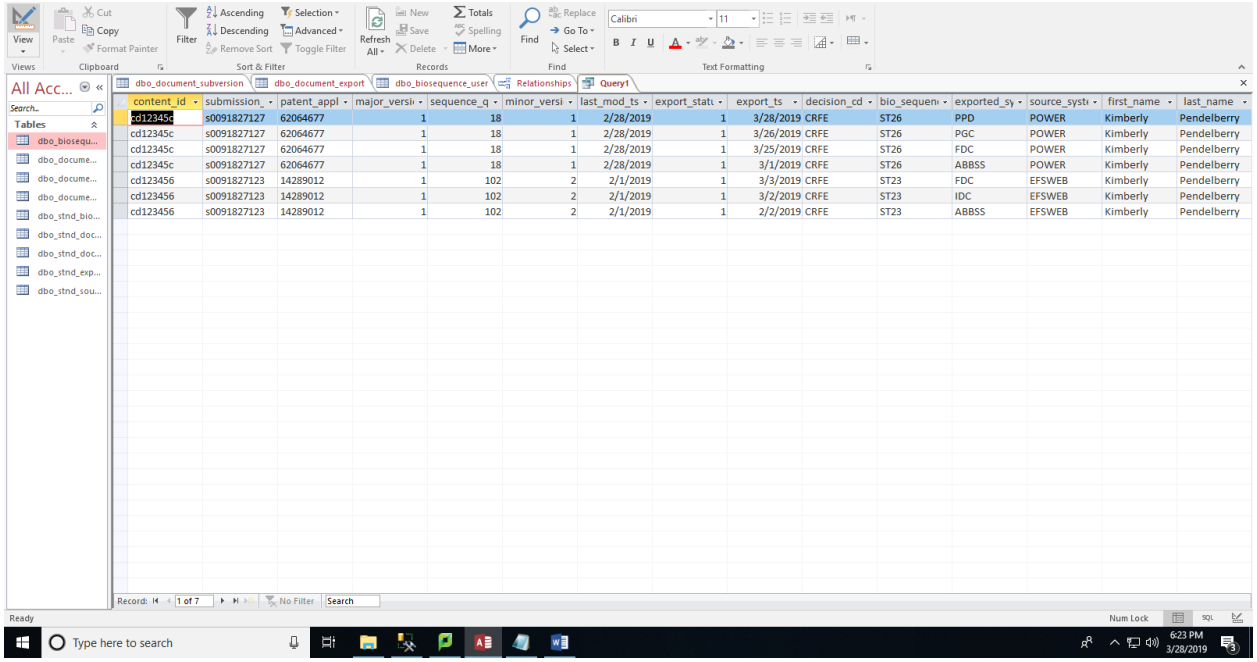
Source system is updated to OIPE (2) from the form and it updated the back end database

document_v	content_id	bio_sequen	source_syste	submission_
1	cd123456	1	2	s0091827123
2	cd12345a	2	2	s0091827124
3	cd12345b	3	2	s0091827125
4	cd12345c	3	3	s0091827127
(New)				

Complete form with Sub-form is designed in MS access where data can be entered document_version and document_subversion table together in a single form.

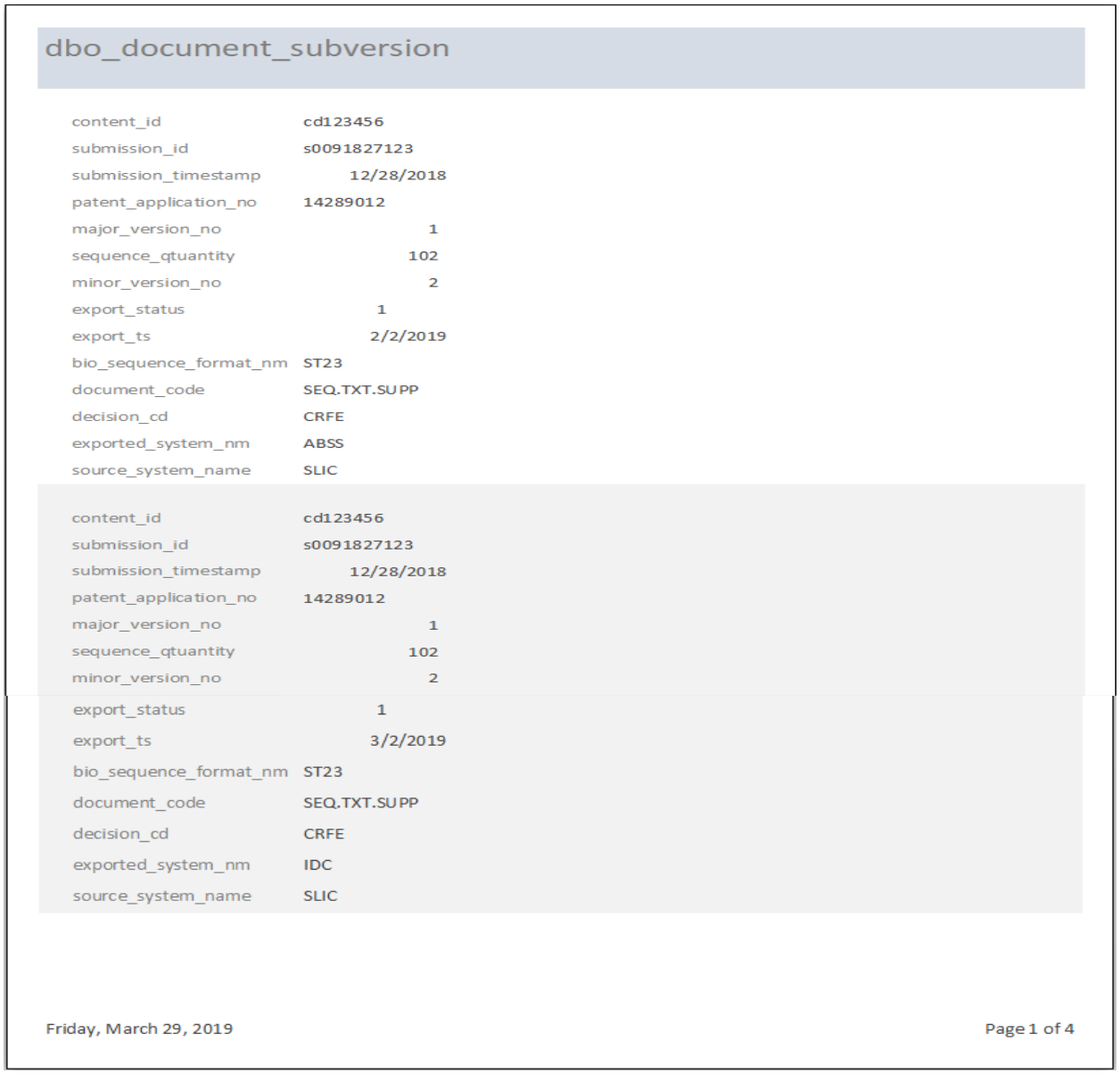


- Complete overview report of Bio-Sequence from submission to export/publication (This DASHBOARD is generated in MS Access)



This gives the complete information about the application when, what application from which system, when patent is approved and who published. This is a token dashboard to show that Kind of relational business data can be projected through dash board. Basically at back end it joins all the relational data and display the common set which is finally published or got patent.

Complete report on a particular Application designed in MS Access to show how report can be generate on this data model.

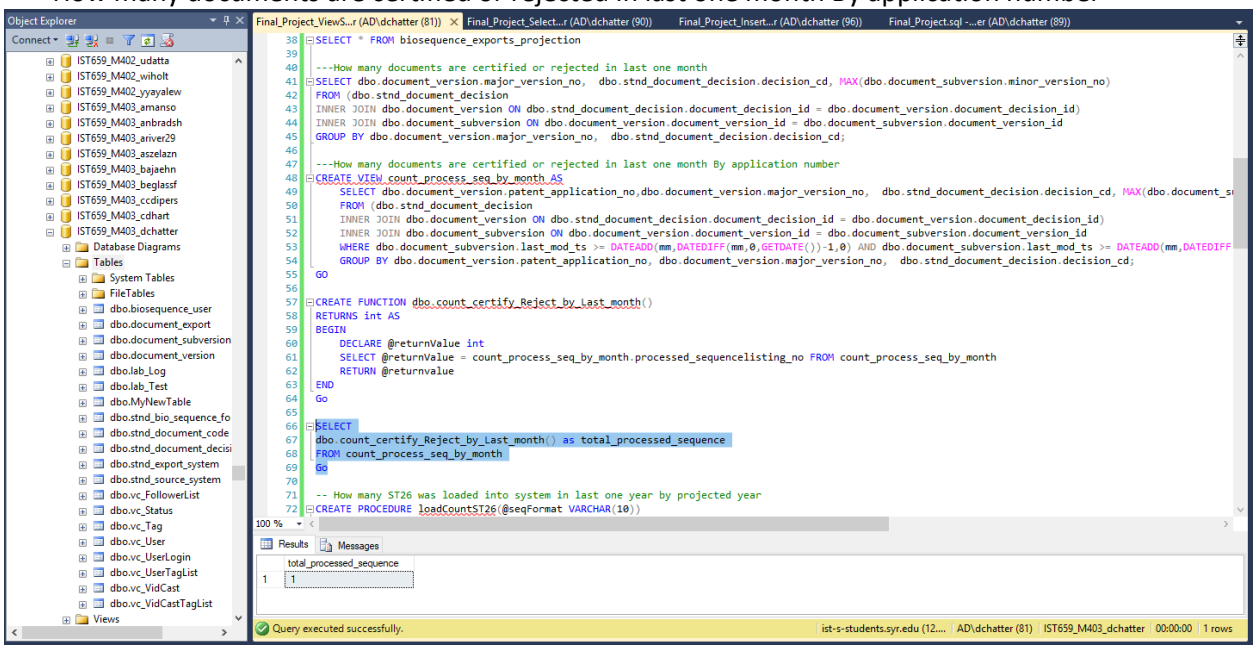


dbo_document_subversion

content_id	cd123456
submission_id	s0091827123
submission_timestamp	12/28/2018
patent_application_no	14289012
major_version_no	1
sequence_qtuantity	102
minor_version_no	2
export_status	1
export_ts	2/2/2019
bio_sequence_format_nm	ST23
document_code	SEQ.TXT.SUPP
decision_cd	CRFE
exported_system_nm	ABSS
source_system_name	SLIC

content_id	cd123456
submission_id	s0091827123
submission_timestamp	12/28/2018
patent_application_no	14289012
major_version_no	1
sequence_qtuantity	102
minor_version_no	2
export_status	1
export_ts	3/2/2019
bio_sequence_format_nm	ST23
document_code	SEQ.TXT.SUPP
decision_cd	CRFE
exported_system_nm	IDC
source_system_name	SLIC

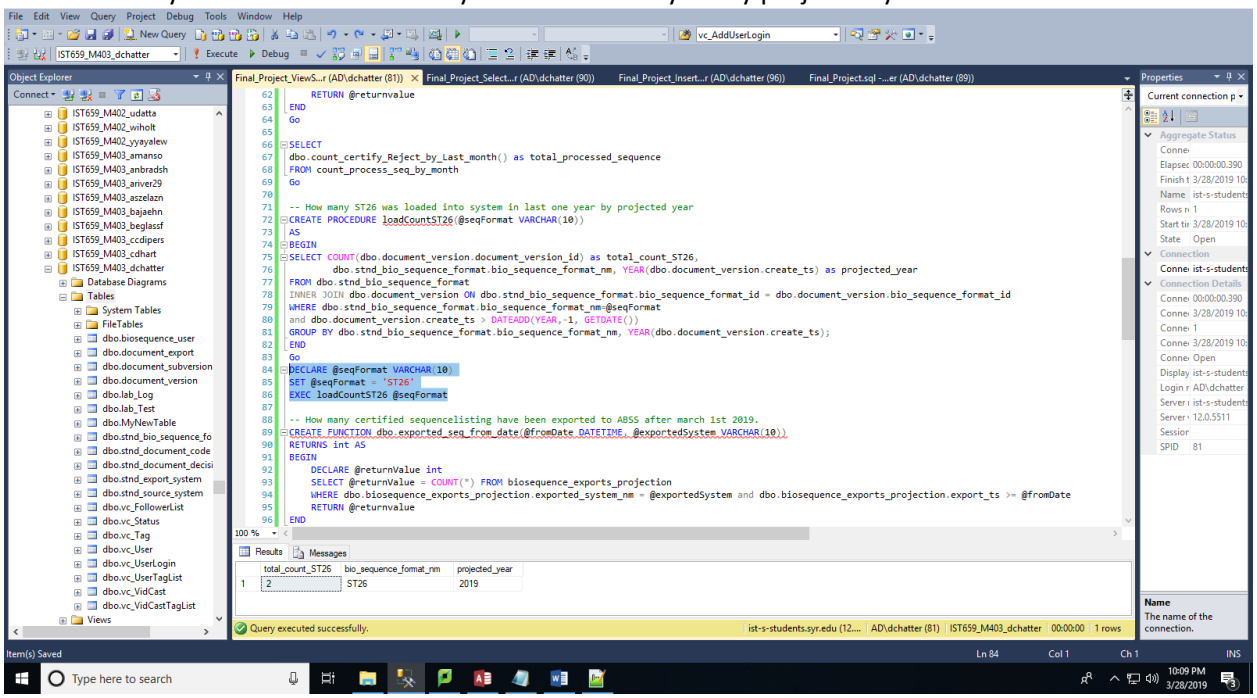
- How many documents are certified or rejected in last one month By application number



```
SELECT * FROM biosequence_exports_projection

---How many documents are certified or rejected in last one month
SELECT dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd, MAX(dbo.document_subversion.minor_version_no)
FROM (dbo.stnd_document_decision
INNER JOIN dbo.document_version ON dbo.stnd_document_decision.document_decision_id = dbo.document_version.document_decision_id)
INNER JOIN dbo.document_subversion ON dbo.document_version.document_version_id = dbo.document_subversion.document_version_id
GROUP BY dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd;

---How many documents are certified or rejected in last one month By application number
CREATE VIEW count_process_seq_by_month AS
    SELECT dbo.document_version.patent_application_no,dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd, MAX(dbo.document_s
    FROM (dbo.stnd_document_decision
    INNER JOIN dbo.document_version ON dbo.stnd_document_decision.document_decision_id = dbo.document_version.document_decision_id)
    INNER JOIN dbo.document_subversion ON dbo.document_version.document_version_id = dbo.document_subversion.document_version_id
    WHERE dbo.document_subversion.last_mod_ts >= DATEADD(mm,DATEDIFF(mm,0,GETDATE())-1,0) AND dbo.document_subversion.last_mod_ts >= DATEADD(mm,DATEDIFF
    GROUP BY dbo.document_version.patent_application_no, dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd;
GO

CREATE FUNCTION dbo.count_certify_Reject_by_Last_month()
RETURNS int AS
BEGIN
    DECLARE @returnValue int
    SELECT @returnValue = count_process_seq_by_month.processed_sequencelisting_no FROM count_process_seq_by_month
    RETURN @returnvalue
END
Go

SELECT
dbo.count_certify_Reject_by_Last_month() as total_processed_sequence
FROM count_process_seq_by_month
Go

-- How many ST26 was loaded into system in last one year by projected year
CREATE PROCEDURE loadCountST26(@seqFormat VARCHAR(10))
```

```
total_processed_sequence
1
```

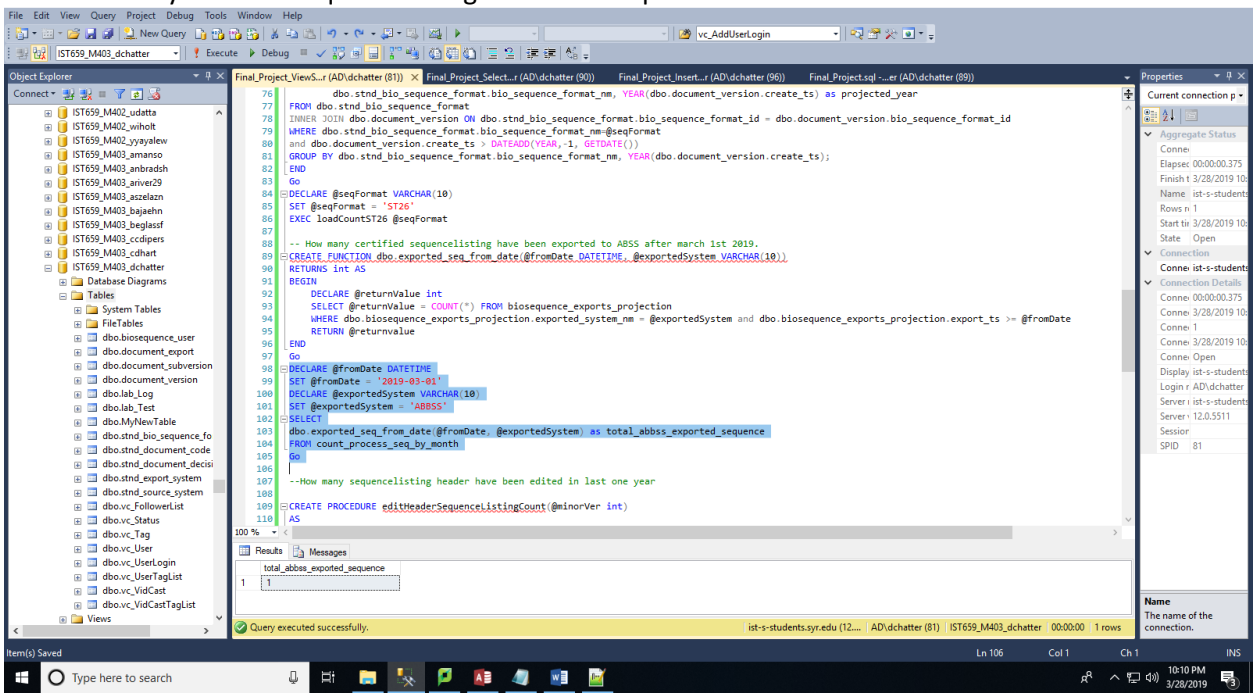
- How many ST26 was loaded into system in last one year by projected year



```
    RETURN @returnvalue
END
Go

SELECT
dbo.count_certify_Reject_by_Last_month() as total_processed_sequence
FROM count_process_seq_by_month
Go

-- How many ST26 was loaded into system in last one year by projected year
CREATE PROCEDURE loadCountST26(@seqFormat VARCHAR(10))
AS
BEGIN
SELECT COUNT(dbo.document_version.document_version_id) as total_count_ST26,
        dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts) as projected_year
FROM dbo.stnd_bio_sequence_format
INNER JOIN dbo.document_version ON dbo.stnd_bio_sequence_format.bio_sequence_format_id = dbo.document_version.bio_sequence_format_id
WHERE dbo.stnd_bio_sequence_format.bio_sequence_format_nm=@seqFormat
and dbo.document_version.create_ts > DATEADD(YEAR,-1, GETDATE())
GROUP BY dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts);
END
Go
DECLARE @seqFormat VARCHAR(10)
SET @seqFormat = 'ST26'
EXEC loadCountST26 @seqFormat

-- How many certified sequencelisting have been exported to ABSS after march 1st 2019.
CREATE FUNCTION dbo.exported_seq_from_date(@fromDate DATETIME, @exportedSystem VARCHAR(10))
RETURNS int AS
BEGIN
    DECLARE @returnValue int
    SELECT @returnValue = COUNT(*) FROM biosequence_exports_projection
    WHERE dbo.biosequence_exports_projection.exported_system_nm = @exportedSystem and dbo.biosequence_exports_projection.export_ts >= @fromDate
    RETURN @returnvalue
END
```

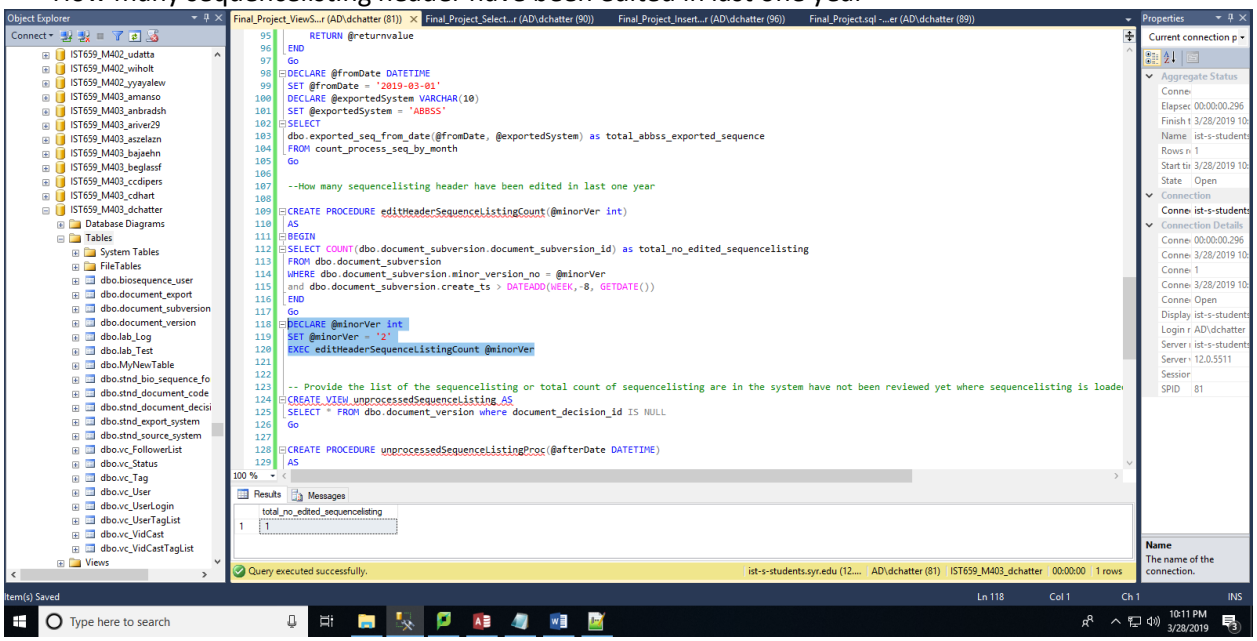
	total_count_ST26	bio_sequence_format_nm	projected_year
1	2	ST26	2019

- How many certified sequencelisting have been exported to ABSS after march 1st 2019.



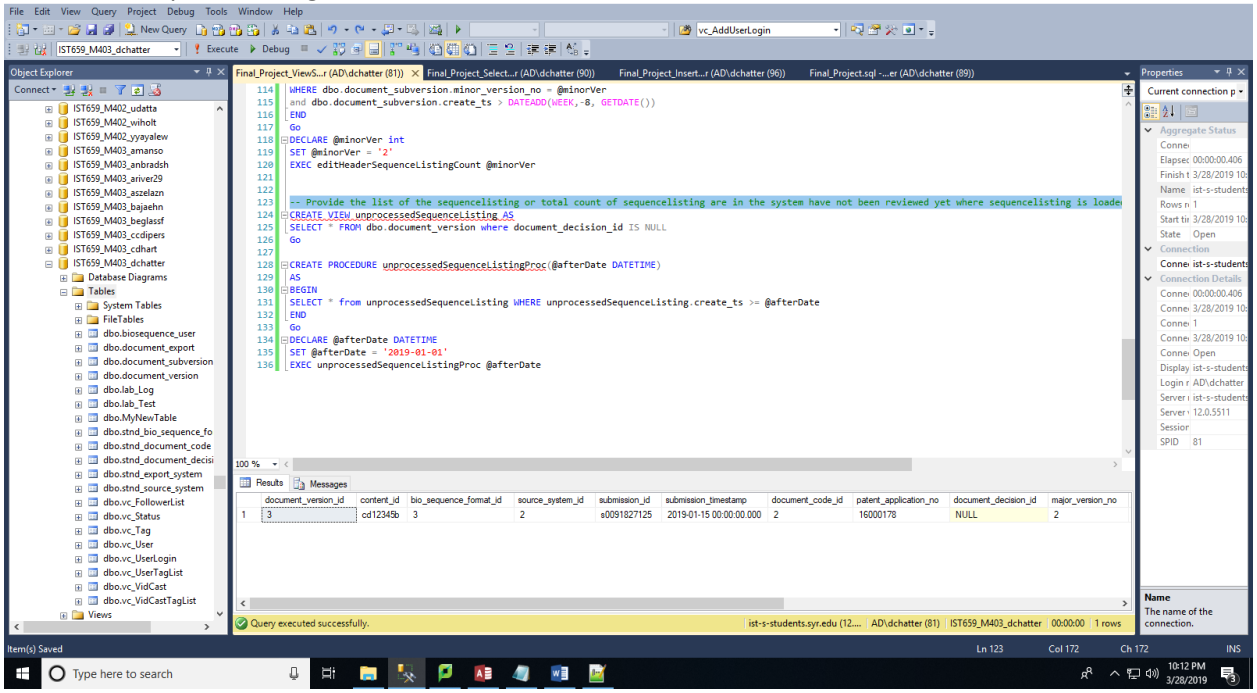
```
        dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts) as projected_year
FROM dbo.stnd_bio_sequence_format
INNER JOIN dbo.document_version ON dbo.stnd_bio_sequence_format.bio_sequence_format_id = dbo.document_version.bio_sequence_format_id
WHERE dbo.stnd_bio_sequence_format.bio_sequence_format_nm=@seqFormat
and dbo.document_version.create_ts > DATEADD(YEAR,-1, GETDATE())
GROUP BY dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts);
END
Go
DECLARE @seqFormat VARCHAR(10)
SET @seqFormat = 'ST26'
EXEC loadCountST26 @seqFormat

-- How many certified sequencelisting have been exported to ABSS after march 1st 2019.
CREATE FUNCTION dbo.exported_seq_from_date(@fromDate DATETIME, @exportedSystem VARCHAR(10))
RETURNS int AS
BEGIN
    DECLARE @returnValue int
    SELECT @returnValue = COUNT(*) FROM biosequence_exports_projection
    WHERE dbo.biosequence_exports_projection.exported_system_nm = @exportedSystem and dbo.biosequence_exports_projection.export_ts >= @fromDate
    RETURN @returnvalue
END
Go
DECLARE @fromDate DATETIME
SET @fromDate = '2019-03-01'
DECLARE @exportedSystem VARCHAR(10)
SET @exportedSystem = 'ABBSS'
SELECT
dbo.exported_seq_from_date(@fromDate, @exportedSystem) as total_abbss_exported_sequence
FROM count_process_seq_by_month
Go

--How many sequencelisting header have been edited in last one year

CREATE PROCEDURE editHeaderSequenceListingCount(@minorVer int)
AS
```

- How many sequencelisting header have been edited in last one year



```
    RETURN @returnvalue
END
Go
DECLARE @fromDate DATETIME
SET @fromDate = '2019-03-01'
DECLARE @exportedSystem VARCHAR(10)
SET @exportedSystem = 'ABBSS'
SELECT
dbo.exported_seq_from_date(@fromDate, @exportedSystem) as total_abbss_exported_sequence
FROM count_process_seq_by_month
Go

--How many sequencelisting header have been edited in last one year

CREATE PROCEDURE editHeaderSequenceListingCount(@minorVer int)
AS
BEGIN
SELECT COUNT(dbo.document_subversion.document_subversion_id) as total_no_edited_sequencelisting
FROM dbo.document_subversion
WHERE dbo.document_subversion.minor_version_no = @minorVer
and dbo.document_subversion.create_ts > DATEADD(WEEK,-8, GETDATE())
END
Go
DECLARE @minorVer int
SET @minorVer = '2'
EXEC editHeaderSequenceListingCount @minorVer

-- Provide the list of the sequencelisting or total count of sequencelisting are in the system have not been reviewed yet where sequencelisting is loade
CREATE VIEW unprocessedSequenceListing AS
SELECT * FROM dbo.document_version where document_decision_id IS NULL
Go

CREATE PROCEDURE unprocessedSequenceListingProc(@afterDate DATETIME)
AS
```

- Provide the list of the sequencelisting or total count of sequencelisting are in the system have not been reviewed yet where sequencelisting is loaded after Jan 1, 2019



10. Project Summary:

This report comprises of the system design for the proposed system and starting from their functions, dependencies and relationships between the various entities and attributes. The analysis of the prototype is included in this report. The prototype has been completed successfully giving with robust impression that the proposed data model can solve USPTO SCORE business problem and it also can be example for other AIS to start thinking about moving to SQL server with proper data model design which will provide data independency and flexibility to make a query and free to design application based on data.

11. Appendix:

DML Script:

```
/*
Author : Debasis Chatterjee
Course : IST659 M403
Term   : March 20, 2019
*/

--CREATE DATABASE [BIOSEQ]
--GO

IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='biosequence_user')
BEGIN
	DROP TABLE biosequence_user
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='stnd_document_code')
BEGIN
	DROP TABLE stnd_document_code
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='stnd_bio_sequence_format')
BEGIN
	DROP TABLE stnd_bio_sequence_format
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='stnd_document_decision')
BEGIN
	DROP TABLE stnd_document_decision
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='stnd_source_system')
BEGIN
	DROP TABLE stnd_source_system
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='stnd_export_system')
BEGIN
	DROP TABLE stnd_export_system
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='document_version')
BEGIN
	DROP TABLE document_version
END
GO
```

```
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='document_subversion')
BEGIN
	DROP TABLE document_subversion
END
GO
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='document_export')
BEGIN
	DROP TABLE document_export
END
GO

-- Creating the BioSequence User Table
CREATE TABLE biosequence_user(
	-- Columns for the user table
	bio_user_id int identity,
	patron_id	varchar(50) not null,
	first_name	varchar(50) not null,
	last_name	varchar(50) not null,
	email_id	varchar(50),
	-- Constraints on the User Table
	CONSTRAINT pk_biosequence_user PRIMARY KEY (bio_user_id),
	CONSTRAINT u1_biosequence_user UNIQUE(patron_id),
	CONSTRAINT u2_biosequence_user UNIQUE(email_id)
)
GO

CREATE TABLE stnd_document_code(
    document_code_id      int identity,
    document_code         varchar(25)      NOT NULL,
    description_tx        varchar(255),
    begin_effective_dt    datetime         NOT NULL,
    end_effective_dt      datetime,
    create_ts             datetime         DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                      NOT NULL,
    last_mod_ts           datetime         DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id      int                      NOT NULL,
    is_locked              tinyint          DEFAULT 0 NOT NULL,
	CONSTRAINT pk_stnd_document_code  PRIMARY KEY (document_code_id),
	CONSTRAINT u1_stnd_document_code  UNIQUE (document_code),
	CONSTRAINT fk1_stnd_document_code FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
	CONSTRAINT fk2_stnd_document_code FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE stnd_bio_sequence_format(
    bio_sequence_format_id    int identity,
    bio_sequence_format_nm    varchar(50)     NOT NULL,
    description_tx            varchar(255),
    begin_effective_dt        datetime        NOT NULL,
```

```
    end_effective_dt          datetime,
    create_ts                 datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id            int              NOT NULL,
    last_mod_ts               datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id          int              NOT NULL,
    is_locked                       tinyint         DEFAULT 0 NOT NULL,
       CONSTRAINT pk_stnd_bio_sequence_format  PRIMARY KEY (bio_sequence_format_id),
       CONSTRAINT u1_stnd_bio_sequence_format  UNIQUE (bio_sequence_format_nm),
       CONSTRAINT fk1_stnd_bio_sequence_format FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_bio_sequence_format FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE stnd_document_decision(
    document_decision_id  int identity,
       decision_cd                varchar(4)    NOT NULL,
    description_tx        varchar(255),
    begin_effective_dt    datetime        NOT NULL,
    end_effective_dt      datetime,
    create_ts             datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                  NOT NULL,
    last_mod_ts           datetime        DEFAULT CURRENT_TIMESTAMP  NOT NULL,
    last_mod_user_id      int                  NOT NULL,
    is_locked                tinyint         DEFAULT 0 NOT NULL,
       CONSTRAINT pk_stnd_document_decision  PRIMARY KEY (document_decision_id),
       CONSTRAINT u1_stnd_document_decision  UNIQUE (decision_cd),
       CONSTRAINT fk1_stnd_document_decision FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_document_decision FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE stnd_source_system(
    source_system_id   int identity,
       source_system_name     varchar(10)   NOT NULL,
    description_tx        varchar(255),
    begin_effective_dt    datetime        NOT NULL,
    end_effective_dt      datetime,
    create_ts             datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id        int                  NOT NULL,
    last_mod_ts           datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id      int                  NOT NULL,
    is_locked                tinyint         DEFAULT 0 NOT NULL,
       CONSTRAINT pk_stnd_source_system  PRIMARY KEY (source_system_id),
       CONSTRAINT u1_stnd_source_system  UNIQUE (source_system_name),
       CONSTRAINT fk1_stnd_source_system FOREIGN KEY (create_user_id)   References
biosequence_user(bio_user_id),
```

```
       CONSTRAINT fk2_stnd_source_system FOREIGN KEY (last_mod_user_id) References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE stnd_export_system(
    exported_system_id        int identity,
       exported_system_nm     varchar(50)   NOT NULL,
    description_tx         varchar(255),
    begin_effective_dt     datetime        NOT NULL,
    end_effective_dt       datetime,
    create_ts              datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id         int                  NOT NULL,
    last_mod_ts            datetime        DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id       int                  NOT NULL,
    is_locked                 tinyint         DEFAULT 0 NOT NULL,
       CONSTRAINT pk_stnd_export_system  PRIMARY KEY (exported_system_id),
       CONSTRAINT u1_stnd_export_system  UNIQUE (exported_system_nm),
       CONSTRAINT fk1_stnd_export_system FOREIGN KEY (create_user_id)       References
biosequence_user(bio_user_id),
       CONSTRAINT fk2_stnd_export_system FOREIGN KEY (last_mod_user_id)     References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE document_version(
    document_version_id          int identity,
    content_id                   varchar(36)    NOT NULL,
    bio_sequence_format_id         int                NOT NULL,
    source_system_id               int                NOT NULL,
    submission_id                varchar(50),
    submission_timestamp           datetime       NOT NULL,
    document_code_id               int                NOT NULL,
    patent_application_no        varchar(17)    NOT NULL,
    document_decision_id         int,
    major_version_no             INT            DEFAULT -1 NOT NULL,
    sequence_qtuantity           INT,
    create_user_id                int              NOT NULL,
    create_ts                    datetime       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_ts                  datetime       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id             int            NOT NULL,
    is_locked                       tinyint        DEFAULT 0 NOT NULL,
       is_deleted                   tinyint        DEFAULT 0 NOT NULL,
       CONSTRAINT pk_document_version  PRIMARY KEY (document_version_id),
       CONSTRAINT u1_document_version  UNIQUE (content_id),
       CONSTRAINT fk1_document_version FOREIGN KEY (bio_sequence_format_id) References
stnd_bio_sequence_format(bio_sequence_format_id),
       CONSTRAINT fk2_document_version FOREIGN KEY (source_system_id)             References
stnd_source_system(source_system_id),
       CONSTRAINT fk3_document_version FOREIGN KEY (document_code_id)             References
stnd_document_code(document_code_id),
```

```
       CONSTRAINT fk4_document_version FOREIGN KEY (document_decision_id)          References
stnd_document_decision(document_decision_id),
       CONSTRAINT fk5_document_version FOREIGN KEY (create_user_id)                References
biosequence_user(bio_user_id),
       CONSTRAINT fk6_document_version FOREIGN KEY (last_mod_user_id)              References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE document_subversion(
    document_subversion_id              int identity,
    document_version_id                 int                       NOT NULL,
    minor_version_no                    int            NOT NULL,
    create_ts                           DATETIME       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_ts                         DATETIME       DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id                    int            NOT NULL,
    is_locked                                          tinyint        DEFAULT 0 NOT NULL,
       is_deleted                                             tinyint        DEFAULT 0 NOT NULL,
       CONSTRAINT pk_document_subversion  PRIMARY KEY (document_subversion_id),
       CONSTRAINT fk1_document_subversion FOREIGN KEY (document_version_id) References
document_version(document_version_id),
       CONSTRAINT fk2_document_subversion FOREIGN KEY (last_mod_user_id)           References
biosequence_user(bio_user_id)
)
GO

CREATE TABLE document_export(
    document_export_id             int identity,
       document_version_id         int                  NOT NULL,
       document_subversion_id      int                  NOT NULL,
       exported_system_id          int                  NOT NULL,
       export_status               tinyint              NOT NULL,
       export_ts                          datetime       NOT NULL,
    create_ts                      datetime    DEFAULT CURRENT_TIMESTAMP NOT NULL,
    create_user_id                 int                NOT NULL,
    last_mod_ts                          datetime    DEFAULT CURRENT_TIMESTAMP NOT NULL,
    last_mod_user_id       int                 NOT NULL,
    is_locked                      tinyint     DEFAULT 0 NOT NULL,
       is_deleted                        tinyint     DEFAULT 0 NOT NULL,
       CONSTRAINT pk_document_export  PRIMARY KEY (document_export_id),
       CONSTRAINT fk1_document_export FOREIGN KEY (document_version_id)     References
document_version(document_version_id),
       CONSTRAINT fk2_document_export FOREIGN KEY (document_subversion_id)  References
document_subversion(document_subversion_id),
       CONSTRAINT fk3_document_export FOREIGN KEY (exported_system_id)             References
stnd_export_system(exported_system_id),
       CONSTRAINT fk4_document_export FOREIGN KEY (create_user_id)                 References
biosequence_user(bio_user_id),
       CONSTRAINT fk5_document_export FOREIGN KEY (last_mod_user_id)        References
biosequence_user(bio_user_id),
)
```

```
GO
```

DQL

Insertion of the data script:

```
/*
Author : Debasis Chatterjee
Course : IST659 M403
Term   : March 28, 2019
*/

--CREATE DATABASE [BIOSEQ]
--GO

-- Insert into biosequence_user
INSERT INTO biosequence_user
       (patron_id, first_name, last_name, email_id) VALUES
       ('312312', 'Debasis', 'Chatterjee', 'dchatter@syr.edu')
GO
INSERT INTO biosequence_user
       (patron_id, first_name, last_name, email_id) VALUES
       ('312313', 'Dale',      'Thompson', 'edthomps@syr.edu')
GO
INSERT INTO biosequence_user
       (patron_id, first_name, last_name, email_id) VALUES
       ('312314', 'Kimberly', 'Pendelberry', 'kpendelb@syr.edu')
GO
INSERT INTO biosequence_user
       (patron_id, first_name, last_name, email_id) VALUES
       ('312315', 'Abdullah', 'Mamdouh', 'amawaysh@syr.edu')
GO
INSERT INTO biosequence_user
       (patron_id, first_name, last_name, email_id) VALUES
       ('312316', 'James',     'Robertson', 'jrober12@syr.edu')
GO
-- Insert into stnd_document_code
INSERT INTO stnd_document_code(document_code,    description_tx,                        begin_effective_dt,
create_user_id, last_mod_user_id) values
                            ('SEQ.TXT.SUPP',   'Sequencelisting txt file for ST23/ST25','1970-01-01',
'1',   '1');
GO
INSERT INTO stnd_document_code(document_code,    description_tx,                        begin_effective_dt,
create_user_id, last_mod_user_id) values
                                                 ('SEQ.XML.SUPP',   'Sequencelisting tXML file for ST26',
'1970-01-01',      '1',   '1');
GO
```

```
-- Insert into stnd_bio_sequence_format
INSERT INTO stnd_bio_sequence_format (bio_sequence_format_nm, description_tx,        begin_effective_dt,
create_user_id, last_mod_user_id) values
                                                          ('ST23',                 'Sequence Listing
23', '1970-01-01',       '1',            '1');
GO
INSERT INTO stnd_bio_sequence_format (bio_sequence_format_nm, description_tx,        begin_effective_dt,
create_user_id, last_mod_user_id) values
                                                          ('ST25',                 'Sequence Listing
25', '1970-01-01',       '1',            '1');
GO
INSERT INTO stnd_bio_sequence_format (bio_sequence_format_nm, description_tx,        begin_effective_dt,
create_user_id, last_mod_user_id) values
                                                          ('ST26',                 'Sequence Listing
26', '1970-01-01',       '1',            '1');
GO

-- Insert into stnd_document_decision
INSERT INTO stnd_document_decision (decision_cd, description_tx,                     begin_effective_dt,
create_user_id, last_mod_user_id) values
                                  ('CRFE',      'COMPUTER READABLE FORMAT EFFECTIVE','1970-01-01',
'1',            '1');
GO
INSERT INTO stnd_document_decision (decision_cd, description_tx,                     begin_effective_dt,
create_user_id, last_mod_user_id) values
                                  ('CRFD',      'COMPUTER READABLE FORMAT DEFFERED','1970-01-01',       '1',
'1');
GO
-- Insert into stnd_source_system
INSERT INTO stnd_source_system (source_system_name, description_tx,                  begin_effective_dt,
create_user_id, last_mod_user_id) VALUES
                               ('EFSWEB',           'ELECTRONIC FILING SYSTEM',         '1970-01-01',
'1',            '1')
GO
INSERT INTO stnd_source_system (source_system_name, description_tx,                  begin_effective_dt,
create_user_id, last_mod_user_id) VALUES
                               ('SLIC',             'SEQUENCELISTING CONTROL',          '1970-01-01',
'1',            '1')
GO
INSERT INTO stnd_source_system (source_system_name, description_tx,                  begin_effective_dt,
end_effective_dt, create_user_id, last_mod_user_id) VALUES
                               ('POWER',            'PATENT POWER',                     '1970-01-01',
'2030-12-31',     '1',            '1')
GO
INSERT INTO stnd_source_system (source_system_name, description_tx,                  begin_effective_dt,
end_effective_dt, create_user_id, last_mod_user_id) VALUES
                               ('IB',               'INTERNATIONAL BUREAU',             '1970-01-01',
'2030-12-31',     '1',            '1')
GO
-- Insert into stnd_export_system
```

```
INSERT INTO stnd_export_system (exported_system_nm,  description_tx,                    begin_effective_dt,
end_effective_dt, create_user_id, last_mod_user_id) VALUES
                            ('IDC',               'INITIAL DATA CAPTURE',            '1970-01-01',
'2030-12-31',     '1',            '1')
GO
INSERT INTO stnd_export_system (exported_system_nm,  description_tx,                    begin_effective_dt,
end_effective_dt, create_user_id, last_mod_user_id) VALUES
                            ('FDC',               'FINAL DATA CAPTURE',            '1970-01-01',
'2030-12-31',     '1',            '1')
GO
INSERT INTO stnd_export_system (exported_system_nm,  description_tx,                    begin_effective_dt,
end_effective_dt, create_user_id, last_mod_user_id) VALUES
                            ('PGC',               'INTERNATIONAL BUREAU',            '1970-01-01',
'2030-12-31',     '1',            '1')
GO
INSERT INTO stnd_export_system (exported_system_nm,  description_tx,                    begin_effective_dt,
end_effective_dt, create_user_id, last_mod_user_id) VALUES
                            ('PPD',               'PROJECTED PUBLICATION DATE',      '1970-01-01',
'2030-12-31',     '1',            '1')
GO
INSERT INTO stnd_export_system (exported_system_nm,  description_tx,
begin_effective_dt, end_effective_dt, create_user_id, last_mod_user_id) VALUES
                            ('ABBSS',              'AUTOMATED BIOTECHNOLOGY SEQUENCE SEARCH SYSTEM',
'1970-01-01',       '2030-12-31',     '1',            '1')
GO
-- Insert into document_version----------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------
INSERT INTO document_version (content_id, bio_sequence_format_id, source_system_id, submission_id,
submission_timestamp, document_code_id, patent_application_no, document_decision_id, major_version_no,
sequence_qtuantity, create_user_id, create_ts,   last_mod_ts, last_mod_user_id) VALUES
                          ('cd123456','1',                    '1',             's0091827123', '2018-12-
28',         '1',             '14289012',           '1',                 '1',             '102',
'2',           '2019-01-31','2019-01-31','2'           )
GO
--need to correct the document_code_id to 1
INSERT INTO document_version (content_id, bio_sequence_format_id, source_system_id, submission_id,
submission_timestamp, document_code_id, patent_application_no, document_decision_id, major_version_no,
sequence_qtuantity, create_user_id, create_ts,   last_mod_ts, last_mod_user_id) VALUES
                          ('cd12345a','2',                    '2',             's0091827124', '2019-01-
15',        '2',             '16000178',           '2',                 '1',             '32189',
'5',           '2019-03-21','2019-03-21','5'           )
GO
INSERT INTO document_version (content_id, bio_sequence_format_id, source_system_id, submission_id,
submission_timestamp, document_code_id, patent_application_no, major_version_no, sequence_qtuantity,
create_user_id, create_ts,   last_mod_ts, last_mod_user_id) VALUES
                          ('cd12345b','3',                    '2',             's0091827125','2019-01-
15',         '2',             '16000178',           '2',             '7675655',         '5',
'2019-03-25','2019-03-25','5'           )
GO
```

```
INSERT INTO document_version (content_id, bio_sequence_format_id, source_system_id, submission_id,
submission_timestamp, document_code_id, patent_application_no, document_decision_id, major_version_no,
sequence_qtuantity, create_user_id, create_ts,   last_mod_ts, last_mod_user_id) VALUES
                            ('cd12345c','3',                    '3',             's0091827127', '2017-05-
23',        '2',             '62064677',           '1',                  '1',             '18',
'1',          '2019-02-28', '2019-02-28','1'           )
GO
-- Insert into document_subversion-----------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('1',                '0',             '2019-01-31', '2019-01-31', '2')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('1',                '1',             '2019-02-01', '2019-02-01', '3')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('1',                '2',             '2019-02-01', '2019-02-01', '3')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('2',                '0',             '2019-03-21', '2019-03-21', '5')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('2',                '1',             '2019-03-22', '2019-03-21', '5')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('3',                '0',             '2019-03-25', '2019-03-25', '4')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('4',                '0',             '2019-02-28', '2019-02-28', '1')
GO
INSERT INTO document_subversion (document_version_id, minor_version_no, create_ts,    last_mod_ts,
last_mod_user_id) VALUES
                               ('4',                '1',             '2019-02-28', '2019-02-28', '2')
GO
-- Insert into document_export---------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                            ('1' ,               '11',                '5',               '1',
'2019-02-02',  '2019-02-02', '3',           '2019-02-02','3'            )
GO
```

```
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                           ('1' ,               '11',                    '1',               '1',
'2019-03-02',  '2019-03-02', '3',           '2019-03-02','3'            )
GO
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                           ('1' ,               '11',                    '2',               '1',
'2019-03-03',  '2019-03-03', '3',           '2019-03-03','3'            )
GO
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                           ('4' ,               '16',                    '5',               '1',
'2019-03-01',  '2019-03-01', '3',           '2019-03-01','3'            )
GO
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                           ('4' ,               '16',                    '2',               '1',
'2019-03-25',  '2019-03-25', '3',           '2019-03-25','3'            )
GO
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                           ('4' ,               '16',                    '3',               '1',
'2019-03-26',  '2019-03-26', '3',           '2019-03-26','3'            )
GO
INSERT INTO document_export (document_version_id, document_subversion_id, exported_system_id, export_status,
export_ts,     create_ts,    create_user_id, last_mod_ts, last_mod_user_id) VALUES
                           ('4' ,               '16',                    '4',               '1',
'2019-03-28',  '2019-03-28', '3',           '2019-03-28','3'            )
GO
```

Select queries to find data:

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [bio_user_id]
      ,[patron_id]
      ,[first_name]
      ,[last_name]
      ,[email_id]
  FROM [BIOSEQ].[dbo].[biosequence_user]


  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [document_code_id]
      ,[document_code]
      ,[description_tx]
      ,[begin_effective_dt]
      ,[end_effective_dt]
      ,[create_ts]
```

```
      ,[create_user_id]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
  FROM [BIOSEQ].[dbo].[stnd_document_code]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [bio_sequence_format_id]
      ,[bio_sequence_format_nm]
      ,[description_tx]
      ,[begin_effective_dt]
      ,[end_effective_dt]
      ,[create_ts]
      ,[create_user_id]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
  FROM [BIOSEQ].[dbo].[stnd_bio_sequence_format]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [document_decision_id]
      ,[decision_cd]
      ,[description_tx]
      ,[begin_effective_dt]
      ,[end_effective_dt]
      ,[create_ts]
      ,[create_user_id]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
  FROM [BIOSEQ].[dbo].[stnd_document_decision]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [source_system_id]
      ,[source_system_name]
      ,[description_tx]
      ,[begin_effective_dt]
      ,[end_effective_dt]
      ,[create_ts]
      ,[create_user_id]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
  FROM [BIOSEQ].[dbo].[stnd_source_system]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [exported_system_id]
      ,[exported_system_nm]
      ,[description_tx]
      ,[begin_effective_dt]
      ,[end_effective_dt]
```

```
      ,[create_ts]
      ,[create_user_id]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
  FROM [BIOSEQ].[dbo].[stnd_export_system]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [document_version_id]
      ,[content_id]
      ,[bio_sequence_format_id]
      ,[source_system_id]
      ,[submission_id]
      ,[submission_timestamp]
      ,[document_code_id]
      ,[patent_application_no]
      ,[document_decision_id]
      ,[major_version_no]
      ,[sequence_qtuantity]
      ,[create_user_id]
      ,[create_ts]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
      ,[is_deleted]
  FROM [BIOSEQ].[dbo].[document_version]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [document_subversion_id]
      ,[document_version_id]
      ,[minor_version_no]
      ,[create_ts]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
      ,[is_deleted]
  FROM [BIOSEQ].[dbo].[document_subversion]

  /****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [document_export_id]
      ,[document_version_id]
      ,[document_subversion_id]
      ,[exported_system_id]
      ,[export_status]
      ,[export_ts]
      ,[create_ts]
      ,[create_user_id]
      ,[last_mod_ts]
      ,[last_mod_user_id]
      ,[is_locked]
      ,[is_deleted]
```

```
  FROM [BIOSEQ].[dbo].[document_export]
```

DB queries to support Business questions behind the forms/reports –

```
--Complete overview report of Bio-Sequence Exports

CREATE VIEW biosequence_exports_projection AS
       SELECT dbo.document_version.content_id,
              dbo.document_version.submission_id,
                 dbo.document_version.patent_application_no,
                 dbo.document_version.major_version_no,
                 dbo.document_version.sequence_qtuantity,
                 dbo.document_subversion.minor_version_no,
                 dbo.document_subversion.last_mod_ts,
                 dbo.document_export.export_status,
                 dbo.document_export.export_ts,
                 dbo.stnd_document_decision.decision_cd,
                 dbo.stnd_bio_sequence_format.bio_sequence_format_nm,
                 dbo.stnd_export_system.exported_system_nm,
                 dbo.stnd_source_system.source_system_name,
                 dbo.biosequence_user.first_name,
                 dbo.biosequence_user.last_name
       FROM dbo.biosequence_user
       INNER JOIN (dbo.stnd_source_system
                           INNER JOIN (dbo.stnd_export_system
                                              INNER JOIN (dbo.stnd_bio_sequence_format
                                                                 INNER JOIN (dbo.stnd_document_decision
                                                                                    INNER JOIN
(dbo.document_version

       INNER JOIN (dbo.document_subversion

                           INNER JOIN dbo.document_export

                           ON dbo.document_subversion.document_subversion_id =
dbo.document_export.document_subversion_id)

       ON (dbo.document_version.document_version_id = dbo.document_export.document_version_id)

       AND (dbo.document_version.document_version_id = dbo.document_subversion.document_version_id))
                                                                                                  ON
dbo.stnd_document_decision.document_decision_id = dbo.document_version.document_decision_id)
                                                                 ON
dbo.stnd_bio_sequence_format.bio_sequence_format_id = dbo.document_version.bio_sequence_format_id)
                                              ON dbo.stnd_export_system.exported_system_id =
dbo.document_export.exported_system_id)
                           ON dbo.stnd_source_system.source_system_id =
dbo.document_version.source_system_id)
       ON dbo.biosequence_user.bio_user_id = dbo.document_export.create_user_id;
```

```
GO

SELECT * FROM biosequence_exports_projection

---How many documents are certified or rejected in last one month
SELECT dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd,
MAX(dbo.document_subversion.minor_version_no)
FROM (dbo.stnd_document_decision
INNER JOIN dbo.document_version ON dbo.stnd_document_decision.document_decision_id =
dbo.document_version.document_decision_id)
INNER JOIN dbo.document_subversion ON dbo.document_version.document_version_id =
dbo.document_subversion.document_version_id
GROUP BY dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd;

---How many documents are certified or rejected in last one month By application number
CREATE VIEW count_process_seq_by_month AS
       SELECT dbo.document_version.patent_application_no,dbo.document_version.major_version_no,
dbo.stnd_document_decision.decision_cd, MAX(dbo.document_subversion.minor_version_no) as
processed_sequencelisting_no
       FROM (dbo.stnd_document_decision
       INNER JOIN dbo.document_version ON dbo.stnd_document_decision.document_decision_id =
dbo.document_version.document_decision_id)
       INNER JOIN dbo.document_subversion ON dbo.document_version.document_version_id =
dbo.document_subversion.document_version_id
       WHERE dbo.document_subversion.last_mod_ts >= DATEADD(mm,DATEDIFF(mm,0,GETDATE())-1,0) AND
dbo.document_subversion.last_mod_ts >= DATEADD(mm,DATEDIFF(mm,0,GETDATE()),0)
       GROUP BY dbo.document_version.patent_application_no, dbo.document_version.major_version_no,
dbo.stnd_document_decision.decision_cd;
GO

CREATE FUNCTION dbo.count_certify_Reject_by_Last_month()
RETURNS int AS
BEGIN
       DECLARE @returnValue int
       SELECT @returnValue = count_process_seq_by_month.processed_sequencelisting_no FROM
count_process_seq_by_month
       RETURN @returnvalue
END
Go

SELECT
dbo.count_certify_Reject_by_Last_month() as total_processed_sequence
FROM count_process_seq_by_month
Go

-- How many ST26 was loaded into system in last one year by projected year
CREATE PROCEDURE loadCountST26(@seqFormat VARCHAR(10))
AS
BEGIN
SELECT COUNT(dbo.document_version.document_version_id) as total_count_ST26,
```

```
            dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts) as
projected_year
FROM dbo.stnd_bio_sequence_format
INNER JOIN dbo.document_version ON dbo.stnd_bio_sequence_format.bio_sequence_format_id =
dbo.document_version.bio_sequence_format_id
WHERE dbo.stnd_bio_sequence_format.bio_sequence_format_nm=@seqFormat
and dbo.document_version.create_ts > DATEADD(YEAR,-1, GETDATE())
GROUP BY dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts);
END
Go
DECLARE @seqFormat VARCHAR(10)
SET @seqFormat = 'ST26'
EXEC loadCountST26 @seqFormat

-- How many certified sequencelisting have been exported to ABSS after march 1st 2019.
CREATE FUNCTION dbo.exported_seq_from_date(@fromDate DATETIME, @exportedSystem VARCHAR(10))
RETURNS int AS
BEGIN
       DECLARE @returnValue int
       SELECT @returnValue = COUNT(*) FROM biosequence_exports_projection
       WHERE dbo.biosequence_exports_projection.exported_system_nm = @exportedSystem and
dbo.biosequence_exports_projection.export_ts >= @fromDate
       RETURN @returnvalue
END--Complete overview report of Bio-Sequence Exports

CREATE VIEW biosequence_exports_projection AS
       SELECT dbo.document_version.content_id,
              dbo.document_version.submission_id,
                 dbo.document_version.patent_application_no,
                 dbo.document_version.major_version_no,
                 dbo.document_version.sequence_qtuantity,
                 dbo.document_subversion.minor_version_no,
                 dbo.document_subversion.last_mod_ts,
                 dbo.document_export.export_status,
                 dbo.document_export.export_ts,
                 dbo.stnd_document_decision.decision_cd,
                 dbo.stnd_bio_sequence_format.bio_sequence_format_nm,
                 dbo.stnd_export_system.exported_system_nm,
                 dbo.stnd_source_system.source_system_name,
                 dbo.biosequence_user.first_name,
                 dbo.biosequence_user.last_name
       FROM dbo.biosequence_user
       INNER JOIN (dbo.stnd_source_system
                           INNER JOIN (dbo.stnd_export_system
                                              INNER JOIN (dbo.stnd_bio_sequence_format
                                                                 INNER JOIN (dbo.stnd_document_decision
                                                                                    INNER JOIN
(dbo.document_version

       INNER JOIN (dbo.document_subversion
```

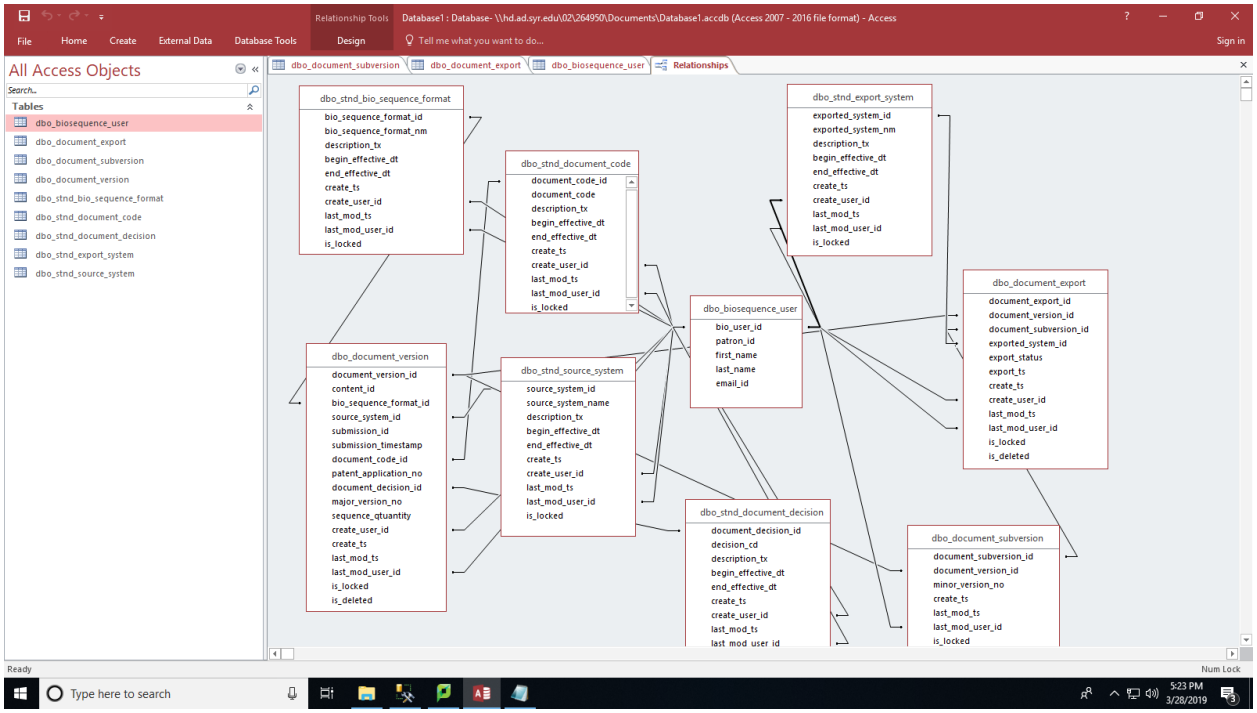
```
                            INNER JOIN dbo.document_export

                            ON dbo.document_subversion.document_subversion_id =
dbo.document_export.document_subversion_id)

       ON (dbo.document_version.document_version_id = dbo.document_export.document_version_id)

       AND (dbo.document_version.document_version_id = dbo.document_subversion.document_version_id))
                                                                                    ON
dbo.stnd_document_decision.document_decision_id = dbo.document_version.document_decision_id)
                                                                          ON
dbo.stnd_bio_sequence_format.bio_sequence_format_id = dbo.document_version.bio_sequence_format_id)
                                                  ON dbo.stnd_export_system.exported_system_id =
dbo.document_export.exported_system_id)
                            ON dbo.stnd_source_system.source_system_id =
dbo.document_version.source_system_id)
       ON dbo.biosequence_user.bio_user_id = dbo.document_export.create_user_id;

GO

SELECT * FROM biosequence_exports_projection

---How many documents are certified or rejected in last one month
SELECT dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd,
MAX(dbo.document_subversion.minor_version_no)
FROM (dbo.stnd_document_decision
INNER JOIN dbo.document_version ON dbo.stnd_document_decision.document_decision_id =
dbo.document_version.document_decision_id)
INNER JOIN dbo.document_subversion ON dbo.document_version.document_version_id =
dbo.document_subversion.document_version_id
GROUP BY dbo.document_version.major_version_no,  dbo.stnd_document_decision.decision_cd;

---How many documents are certified or rejected in last one month By application number
CREATE VIEW count_process_seq_by_month AS
       SELECT dbo.document_version.patent_application_no,dbo.document_version.major_version_no,
dbo.stnd_document_decision.decision_cd, MAX(dbo.document_subversion.minor_version_no) as
processed_sequencelisting_no
       FROM (dbo.stnd_document_decision
       INNER JOIN dbo.document_version ON dbo.stnd_document_decision.document_decision_id =
dbo.document_version.document_decision_id)
       INNER JOIN dbo.document_subversion ON dbo.document_version.document_version_id =
dbo.document_subversion.document_version_id
       WHERE dbo.document_subversion.last_mod_ts >= DATEADD(mm,DATEDIFF(mm,0,GETDATE())-1,0) AND
dbo.document_subversion.last_mod_ts >= DATEADD(mm,DATEDIFF(mm,0,GETDATE()),0)
       GROUP BY dbo.document_version.patent_application_no, dbo.document_version.major_version_no,
dbo.stnd_document_decision.decision_cd;
GO

CREATE FUNCTION dbo.count_certify_Reject_by_Last_month()
RETURNS int AS
```

```
BEGIN
	DECLARE @returnValue int
	SELECT @returnValue = count_process_seq_by_month.processed_sequencelisting_no FROM
count_process_seq_by_month
	RETURN @returnvalue
END
Go

SELECT
dbo.count_certify_Reject_by_Last_month() as total_processed_sequence
FROM count_process_seq_by_month
Go

-- How many ST26 was loaded into system in last one year by projected year
CREATE PROCEDURE loadCountST26(@seqFormat VARCHAR(10))
AS
BEGIN
SELECT COUNT(dbo.document_version.document_version_id) as total_count_ST26,
		dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts) as
projected_year
FROM dbo.stnd_bio_sequence_format
INNER JOIN dbo.document_version ON dbo.stnd_bio_sequence_format.bio_sequence_format_id =
dbo.document_version.bio_sequence_format_id
WHERE dbo.stnd_bio_sequence_format.bio_sequence_format_nm=@seqFormat
and dbo.document_version.create_ts > DATEADD(YEAR,-1, GETDATE())
GROUP BY dbo.stnd_bio_sequence_format.bio_sequence_format_nm, YEAR(dbo.document_version.create_ts);
END
Go
DECLARE @seqFormat VARCHAR(10)
SET @seqFormat = 'ST26'
EXEC loadCountST26 @seqFormat

-- How many certified sequencelisting have been exported to ABSS after march 1st 2019.
CREATE FUNCTION dbo.exported_seq_from_date(@fromDate DATETIME, @exportedSystem VARCHAR(10))
RETURNS int AS
BEGIN
	DECLARE @returnValue int
	SELECT @returnValue = COUNT(*) FROM biosequence_exports_projection
	WHERE dbo.biosequence_exports_projection.exported_system_nm = @exportedSystem and
dbo.biosequence_exports_projection.export_ts >= @fromDate
	RETURN @returnvalue
END
Go
DECLARE @fromDate DATETIME
SET @fromDate = '2019-03-01'
DECLARE @exportedSystem VARCHAR(10)
SET @exportedSystem = 'ABBSS'
SELECT
dbo.exported_seq_from_date(@fromDate, @exportedSystem) as total_abbss_exported_sequence
FROM count_process_seq_by_month
Go
```

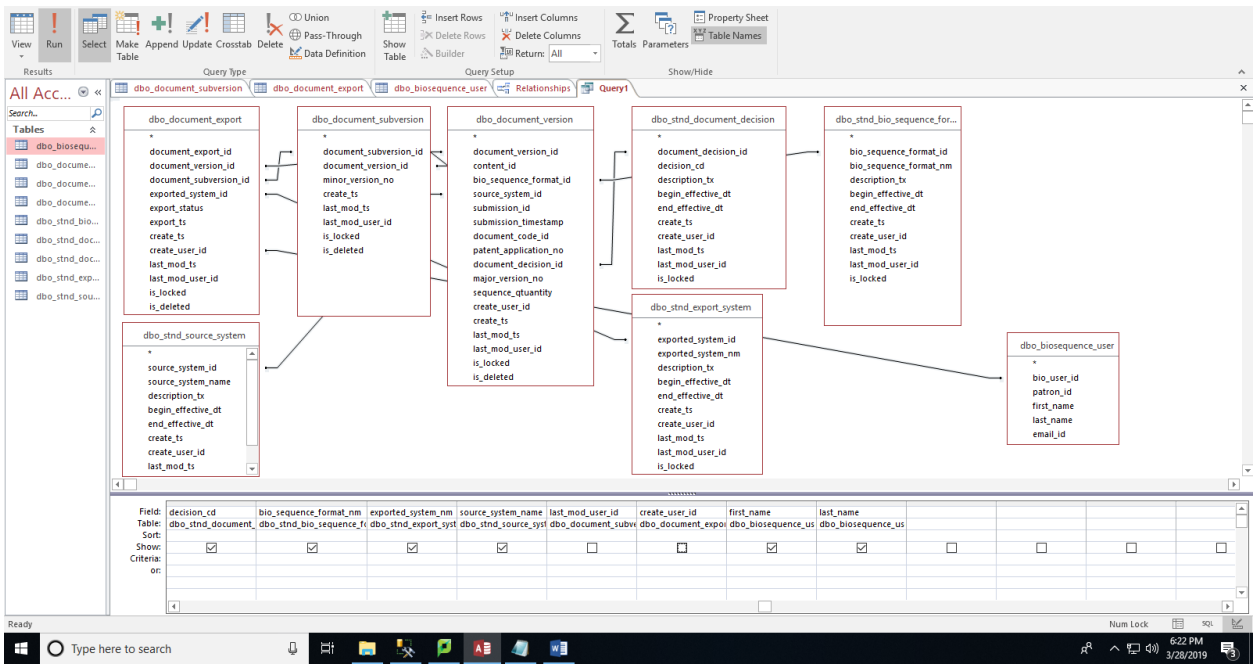
```
--How many sequencelisting header have been edited in last one year

CREATE PROCEDURE editHeaderSequenceListingCount(@minorVer int)
AS
BEGIN
SELECT COUNT(dbo.document_subversion.document_subversion_id) as total_no_edited_sequencelisting
FROM dbo.document_subversion
WHERE dbo.document_subversion.minor_version_no = @minorVer
and dbo.document_subversion.create_ts > DATEADD(WEEK,-8, GETDATE())
END
Go
DECLARE @minorVer int
SET @minorVer = '2'
EXEC editHeaderSequenceListingCount @minorVer


-- Provide the list of the sequencelisting or total count of sequencelisting are in the system have not been
reviewed yet where sequencelisting is loaded after Jan 1, 2019
CREATE VIEW unprocessedSequenceListing AS
SELECT * FROM dbo.document_version where document_decision_id IS NULL
Go

CREATE PROCEDURE unprocessedSequenceListingProc(@afterDate DATETIME)
AS
BEGIN
SELECT * from unprocessedSequenceListing WHERE unprocessedSequenceListing.create_ts >= @afterDate
END
Go
DECLARE @afterDate DATETIME
SET @afterDate = '2019-01-01'
EXEC unprocessedSequenceListingProc @afterDate

Go
DECLARE @fromDate DATETIME
SET @fromDate = '2019-03-01'
DECLARE @exportedSystem VARCHAR(10)
SET @exportedSystem = 'ABBSS'
SELECT
dbo.exported_seq_from_date(@fromDate, @exportedSystem) as total_abbss_exported_sequence
FROM count_process_seq_by_month
Go

--How many sequencelisting header have been edited in last one year

CREATE PROCEDURE editHeaderSequenceListingCount(@minorVer int)
AS
BEGIN
SELECT COUNT(dbo.document_subversion.document_subversion_id) as total_no_edited_sequencelisting
FROM dbo.document_subversion
WHERE dbo.document_subversion.minor_version_no = @minorVer
```

```
and dbo.document_subversion.create_ts > DATEADD(WEEK,-8, GETDATE())
END
Go
DECLARE @minorVer int
SET @minorVer = '2'
EXEC editHeaderSequenceListingCount @minorVer


-- Provide the list of the sequencelisting or total count of sequencelisting are in the system have not been
reviewed yet where sequencelisting is loaded after Jan 1, 2019
CREATE VIEW unprocessedSequenceListing AS
SELECT * FROM dbo.document_version where document_decision_id IS NULL
Go

CREATE PROCEDURE unprocessedSequenceListingProc(@afterDate DATETIME)
AS
BEGIN
SELECT * from unprocessedSequenceListing WHERE unprocessedSequenceListing.create_ts >= @afterDate
END
Go
DECLARE @afterDate DATETIME
SET @afterDate = '2019-01-01'
EXEC unprocessedSequenceListingProc @afterDate
```

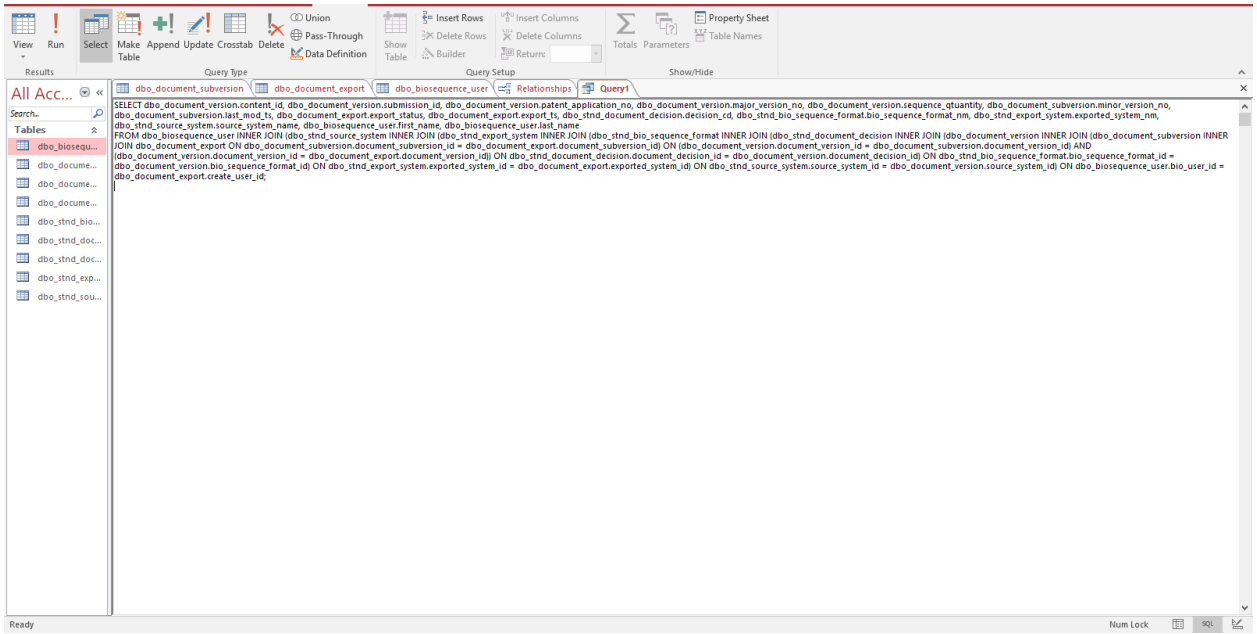
Relationship made in MS Access for Query design and Form



Query Design



Sql Query View



12.

Glossary	
USPTO	UNITED STATES AND PATENT TRADEMARK OFFICE
WIPO	World Intellectual Property Organization
ST.23	Standard 23
ST.25	Standard 25
ST.26	Standard 26
ERD	Entity relationship diagram
DBMS	Database management system
EFS	Electronic filing system
OIPE	Office of Initial Patent Examination
STIC	Scientific and Technical Information Center
ABSS	Automated Biotechnology Sequence Search
CRFE	Computer readable format effective
CRFD	Computer readable format differed