**MIS 531 Enterprise Database Management**

**Project Report**

December 9th 2016



Team: **DataMate**

Members:

Brianda Armenta

Karan Dhamija

Menglu Pei

Abhinav Sharma

Meng-Hua Tsai

Table of Contents

[Chapter 1: Requirements analysis](#_30j0zll)

[Chapter 2: Conceptual Design](#_1fob9te)

[Chapter 3: Relational Design](#_3znysh7)

[Chapter 4: Data population and Queries](#_2et92p0)

[Chapter 5: Triggers and Procedures](#_tyjcwt)

[5.1 Triggers](#_3dy6vkm)

[5.2 Procedures](#_1t3h5sf)

[Chapter 6: Interface and Reports](#_4d34og8)

[Chapter 7: Conclusions and implementation plan](#_2s8eyo1)

[Appendix A Table Creation Script](#_17dp8vu)

# 

# **Chapter 1**: Requirements analysis

The organization we are working for is the Arizona-Sonora Desert Museum (ASDM). The ASDM is in Tucson, Arizona, which has a large collection of minerals. Some of these minerals are precious while some are samples. For each mineral, ASDM would like to track of various things like when was it collected. In addition, some roles, such as appraiser, collector, owner, guest, etc., act on minerals on different aspects, and the ASDM would like to record the information of them as well as the actions they perform on minerals. However, the existing database is too old and messy causing confusion. Hence, the volunteer Rocco Santangelo wanted us to upgrade their mineral database with a clearer modern structure and a more visual user interface.

The main body of the ASDM database is mineral. A unique mineral ID is assigned to each mineral for identification. For each mineral, the ASDM stores its name, category, date of registration in system, and source to acquire it. They also would like to record the type to acquire it, which can be origin, purchase, donate, trade or collect. The size of mineral and its crystal size are also recorded. Further, additional measure details including length, height, and depth along with measurement unit about the mineral may or may not be stored. The mineral weight and weight unit are also included. The ASDM also would like to keep track of the cut shape of each mineral and the cutter of it. A short description can be recorded for each mineral. The mineral can be classified by its current usage type, which contain display, loan, borrowed, deaccess, and vault. The displayed mineral is currently used for display, and they would like to record the display ID, display name, display location, and its start date and end date. One displayed mineral can only be in one display currently, and one display can show many displayed minerals. The loaned minerals are currently used for loan, and the ASDM would like to keep track of the details of loan, including loan ID, loan company, start date and end date. Similarly, one mineral can be only used for one loan currently, and one loan can loan several minerals at the same time. When the current usage end and the minerals are used for another usage, the current usage details will be updated, and the previous usage will be recorded as the history of the mineral. For the history of minerals, we record the unique history ID, history name, the type ID (display ID or loan ID), usage type, start date, and end date. Relevant comments are also recorded.

For each mineral, we are required to record its location including city, county, province, and country along with a location name. Each location is assigned with a unique. Furthermore, each location has latitude, longitude, location type (e.g. a mine, place). A location details (such as mountain name) may or may not be stored in location.

One mineral may consist of many species, but only one species (major one or the most precious one) will be recorded. Many minerals may have same species. For each species with a unique ID, we would like to know its name, chemistry formula, group, system, variety, modifier, synonym, and whether it is euhedral. For each species, if it always appear together with other species, it is cross referenced, and the species existing together with it is called And-minerals. If the species always attaches or is attached with other species, we call it is in a slave cross reference and the relevant minerals are recorded as With-mineral. We record this property for each species.

In addition to the digital information, each mineral may have one or many physical documents. Each document will be used to describe only one mineral, including its physical image, graphics, and comments. To identify documents, the ASDM assign a unique ID for each document. The documents also record the reference slide of each mineral. The documents are stored in folders, and each has a folder ID, its description and its storage location. We also count the number of documents in each folder.

The ASDM want us to track the information of persons who has relationship with minerals, which include collectors, owners, appraisers, and analyst. For these roles, we would like to know their personal details such as unique ID, name, phones and emails. One person may have more than one phone or email. We would also like to store their operation details on mineral. We would like to know the collect field (story), collection\_date and relevant association for each mineral by each collector. We would like to know the acqusion\_date, purchased price and owner comments for each mineral by its owner. As for appraisal details, we would like to track the appraisal date, appraiser comments and extimated value for each mineral. For analysis details, we track the analysis date, analysis method, analysis code, analysis results, analysis category and analyst comments.

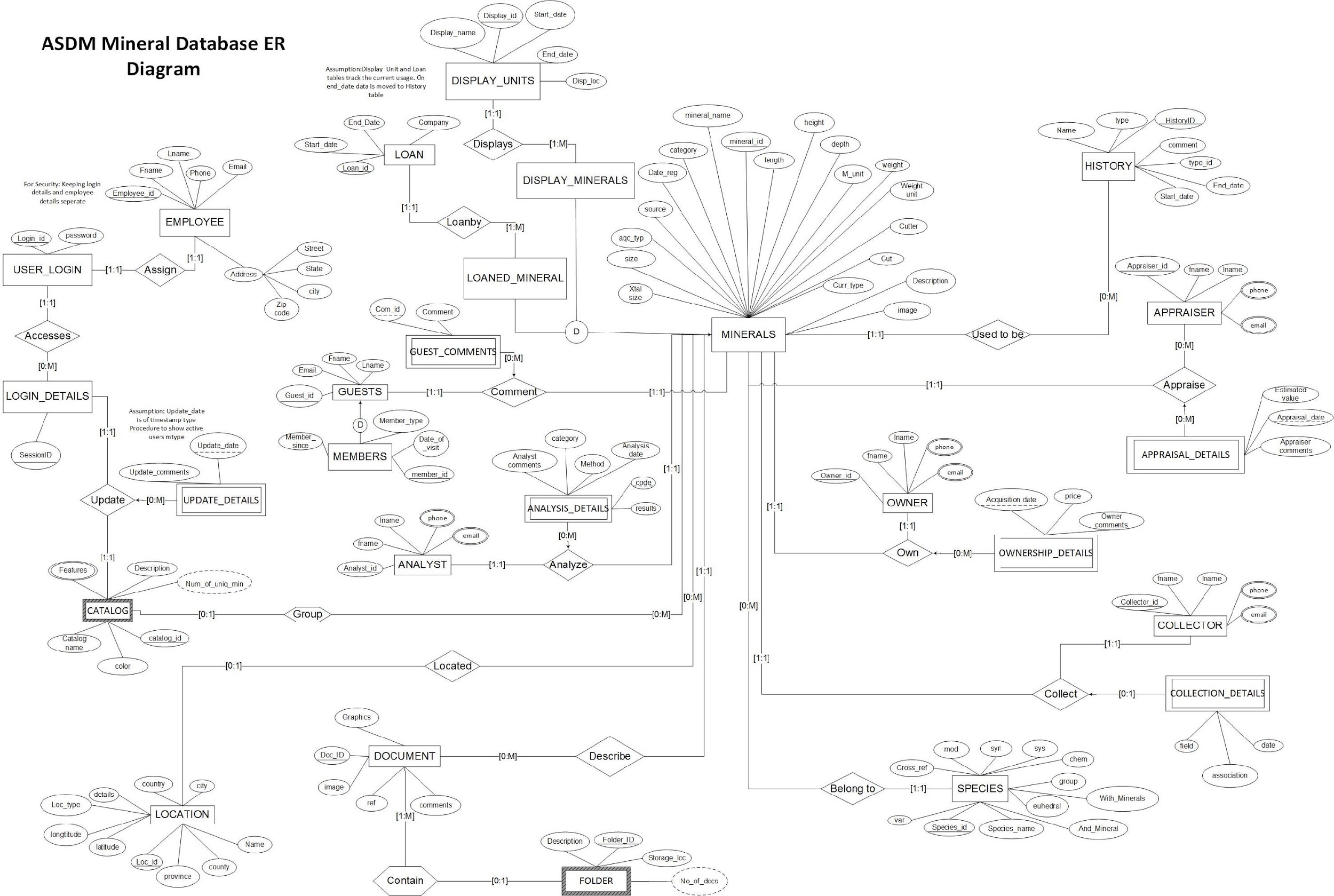
Guests are allowed to visit and comment on minerals. For each guest, the museum record their name and one available email. Their comments on minerals are also stored. If the guest is a member of museum, we would also like to know their member ID, member type, the day when they becomemember, and the date they visit the dessert museum.

The minerals (specimen) can be grouped into catalog according to their similarity. For each catalog of minerals, we are required to record its catalog name, description and color of cataloged minerals along with a unique catalog ID. A catalog of minerals may have one or more features. We also count the number of unique minerals (specimen) in each catalog.

The information of catalog can be updated by employees. The ASDM track the employee ID, employee name, home address, and their work contact (email and phone). For security consideration, each employee has one log\_in information, including log\_in ID and password. If they log in the mineral database, their log\_in details will be recorded and a unique session ID will be generated. Only when employee log in, the information of catalog can be changed and the update date and comments are recorded.

**Chapter 2:** Conceptual Design

Based on the requirement analysis, the ER Diagram of the ASDM Project is shown as following:



The data dictionary (conceptual) is shown as following:

*Table 1 Data Dictionary (Conceptual)*

|  |  |  |
| --- | --- | --- |
| Schema Construct | Construct Description | Other Information |
| ACCESSES | Relationship to model the user log\_in information and login details |  |
| ANALYSIS\_DETAILS | Weak Entity Class, to model details about analyst and mineral |  |
| * analysis\_comments | The comments made by analyst for each analysis |  |
| * analysis\_date | The date of analysis |  |
| * category | The category of each mineral analyzed by analysts |  |
| * code | The unique number of each analysis | *Partial Identifier* |
| * method | Method used for analysis |  |
| * results | Result of the analysis |  |
| ANALYST | Entity Class, to model analyst |  |
| * analyst\_id | Identifying number of the analyst | *Identifier* |
| * email | Email of the analyst | Can have multiple values |
| * fname | First name of the analyst | Should not contain NULL values |
| * lname | Last address of the analyst | Should not contain NULL values |
| * phone | Phone number of the analyst | Can have multiple values |
| ANALYZE | Relationship to model the analysts analyze the minerals | One analyst can analyze many minerals; One minerals can be analyzed by many analysts. |
| APPRAISAL\_DETAILS | Weak Entity Class, to model appraisal details |  |
| * appraisal\_date | The date of appraisal | *Partial Identifier* |
| * appraiser\_comments | The comments of each mineral made by appraisers |  |
| * estimated\_value | The value (in $) of each mineral estimated by appraisers | Should not contain negative values |
| APPRAISE | Relationship to model the appraisers appraise the minerals with appraisal details |  |
| APPRAISER | Entity Class, to model appraisers |  |
| * appraiser\_id | Identifying number of the analyst | *Identifier* |
| * email | Email of the analyst | Can have multiple values |
| * fname | First name of the analyst | Should not contain NULL values |
| * lname | Last address of the analyst | Should not contain NULL values |
| * phone | Phone number of the analyst | Can have multiple values |
| ASSIGN | Relationship to model the employee and user log\_in | One to one relationship |
| BELONG\_TO | Relationship to model the minerals belong to species | One mineral can belong to only one species, but on species may have many mineral specimen |
| CATALOG | Entity Class, to model catalog |  |
| * catalog \_id | Identifying number of the catalog | *Identifier* |
| * catalog\_name | Catalog name | Should not contain NULL values |
| * color | The color of minerals in this catalog |  |
| * description | Description about the catalog |  |
| * features | The features of the minerals in this catalog | Can have multiple values |
| * num\_of\_uni\_min | Number of unique minerals present in the catalog | Should be integer number |
| COLLECT | Relationship of collectors, minerals and collection details |  |
| COLLECTION\_DETAILS | Weak Entity Class, to model details about collector and mineral |  |
| * association | The association where to collect the minerals from |  |
| * collect\_date | The date when collect the minerals | Should not contain NULL values |
| * collect\_field | The field where to collect the minerals |  |
| COLLECTOR | Entity Class, to model collector |  |
| * collector\_id | Identifying number of the collector | *Identifier* |
| * email | Email of the collector | Can have multiple values |
| * fname | First name of the collector | Should not contain NULL values |
| * lname | Last address of the collector | Should not contain NULL values |
| * phone | Phone number of the collector | Can have multiple values |
| COMMENT | Relationship to model the guests comment on the minerals with comment details. | One guest can comment on many minerals with various comment details. |
| CONTAIN | Relationship to model the documents are contained in folders | One folder can contain many documents; One document can only be contained by one folder. |
| DESCRIBE | Relationship to model the documents describe details of minerals. | One document only describes one mineral; One mineral can have many documents |
| DESPLAYS | Relationship to model the displayed minerals are currently displayed for display units | One display can show many minerals at the same time, but one mineral can only be on one display unit currently |
| DISPLAY\_MINERALS | Sub-class of minerals, to model the minerals which currently used for displays | This subclass is identified by current usage type of minerals, all attributes inherit its super-class’s attributes |
| DISPLAY\_UNITS | Entity class, to model minerals displays hold by Museum |  |
| * display \_id | Identifying number of the displays | *Identifier* |
| * display\_location | The location of displays |  |
| * display\_name | The name of displays | Should not contain NULL values |
| * end\_date | The end date of the display | Should be later than system date |
| * start\_date | The start date of the display | Should be earlier than system date |
| DOCUMENT | Entity Class, to model physical documents for minerals |  |
| * comments | Comments of the mineral in the document |  |
| * doc\_id | Identifying number of the physical documents | *Identifier* |
| * graphics | Graphic of the minerals in the document |  |
| * image | Image of the minerals in the document |  |
| * ref | The reference slide number of the mineral in that documents |  |
| EMPLOYEE | Entity Class, to model employees work in the dessert museum |  |
| * address | Address of the employees | Composite attribute |
| * city | The city of employees’ address |  |
| * emails | The work email of the employee | Should be single value |
| * employee\_id | Identifying number of the employees | *Identifier* |
| * fname | The first name of employee | Should not contain NULL values |
| * lname | The last name of employee | Should not contain NULL values |
| * phone | The work phone number of employees | Should be single value |
| * state | The state of employees’ address |  |
| * status | The status of employee |  |
| * street | The street of employees’ address |  |
| * zip code | The zip code of employees’ address | Should be the zip code in US |
| FOLDER | Entity Class, to model folders which are used to save mineral physical documents |  |
| * description | General description about folder |  |
| * fol der\_id | Unique identifier for folder | *Identifier* |
| * No\_of\_docs | How many documents are stored in this folder | The minimal number can be 0 |
| * storage\_loc | The location to store the folder |  |
| GROUP | Relationship to model the minerals are grouped into catalogs with similarity | One mineral can only belong to one catalog |
| GUEST\_COMMENTS | Weak Entity Class, to model the comments made by guests |  |
| * com \_id | Identifying number of the guest comments | *Partial Identifier* |
| * comment | The comment made by guest |  |
| GUESTS | Entity Class, to model guest who visit minerals |  |
| * emails | The email of the guest |  |
| * guest\_id | Identifying number of the guest | *Identifier* |
| * fname | The first name of guest |  |
| * lname | The last name of guest |  |
| HISTORY | Entity Class, to model location of minerals |  |
| * comment | Comment of the history event |  |
| * end\_date | The end date of the history event |  |
| * history\_id | Identifying number of history event | *Identifier* |
| * name | Details about the location |  |
| * type | The type of the history event | The value should be in: 'DISPLAY', 'VAULT', 'DEACCESS', 'LOAN', 'BORROWED' |
| * type\_id | Identifying type id such as loan and display |  |
| * start\_date | Start date of the |  |
| LOAN | Entity Class, to model loan of minerals |  |
| * company | Company who rent minerals |  |
| * end\_date | End date of loan | Should be later than system date |
| * loan\_id | Identifying number of the guest | *Identifier* |
| * start\_date | Start of loan | Should be earlier than system date |
| LOANBY | Relationship to model the loaned mineral is currently loaned by the loan events | One loan event can loan several minerals, but one mineral can only be loaned to one event currently. |
| LOANED\_MINERALS | Sub-class of minerals, to model the minerals which currently used for loans | This subclass is identified by current usage type of minerals, all attributes inherit its super-class’s attributes |
| LOCATED | Relationship to modify that minerals are in locations | One mineral has only one location; One location may locate many minerals |
| LOCATION | Entity Class, to model location of minerals |  |
| * city | City detail about the location |  |
| * country | Country detail about the location |  |
| * county | County detail about the location |  |
| * details | Details about the location |  |
| * latitude | Latitude of the location | Should be reserved with 4 decimal |
| * loc\_id | Identifying number of the location | *Identifier* |
| * loc\_type | Location\_type  about the location | Should be one of: 'MINE', 'QUARRY', 'CLAIM', 'PLACE' |
| * longitude | Longitude of the location | Should be reserved with 4 decimal |
| * name | Name of the location | Should not contain NULL values |
| * province | Province detail about the location |  |
| LOGIN\_DETAILS | Entity Class, to store session for everyone |  |
| * session\_id | Identifying number of the session log in | *Identifier* |
| MEMBERS | Sub-class of the guest |  |
| * date\_of\_visit | The date when the member visit the museum |  |
| * member\_id | Identifying number of the member | *Identifier of subclass* |
| * member\_since | The date when the guest become member |  |
| * member\_type | The type of member |  |
| MINERAL | Entity Class, to model mineral |  |
| * acquired\_type | describing the type of how the mineral was acquired | The value should be in: 'ORIG', 'PURCH', 'DON', 'TRADE', 'COLLECT' |
| * category | category the mineral belongs to | The value should be in: 'MIN', 'GEM', 'MET', 'SYN', 'ORG','GEOL' |
| * curr\_type | The type of the current usage | The value should be in: 'DISPLAY', 'VAULT', 'DEACCESS', 'LOAN', 'BORROWED' |
| * cut | Cut of the mineral | Should be the shape of the cut |
| * cutter | cutter of the mineral |  |
| * date\_registered | Date of registration in the database |  |
| * depth | depth of the mineral | With 2 decimals |
| * description | Description of the mineral |  |
| * height | height of the mineral | With 2 decimals |
| * image | Image of the mineral |  |
| * length | length of the mineral | With 2 decimals |
| * mineral \_id | Identifying number of the mineral | *Identifier* |
| * mineral\_name | The name of each mineral |  |
| * m\_unit | measurement unit of the mineral |  |
| * source | source of the mineral |  |
| * size | size of the mineral |  |
| * weight | weight of the mineral | With 2 decimals |
| * weight\_unit | unit of weight of the mineral | The value should be in: 'GM', 'CT', 'OZ' |
| * xtal\_size | Crystal size of the mineral |  |
| OWN | Relationship of minerals, owners and their ownership details | One mineral has only one location; One location may locate many minerals |
| OWNER | Entity class, to model owners |  |
| * owner\_id | Identifying number of the owner | *Identifier* |
| * email | Email of the owner | Can have multiple values |
| * fname | First name of the owner | Should not contain NULL values |
| * lname | Last address of the owner | Should not contain NULL values |
| * phone | Phone number of the owner | Can have multiple values |
| OWNER\_DETAILS | Weak Entity Class, to model details about owners and mineral |  |
| * comments | comments from the owner |  |
| * date\_of\_acq | date of acquisition of the mineral |  |
| * purchase\_price | purchasing price (in $) of the mineral | Should not be negative value |
| SPECIES | Entity Class, to model species of minerals |  |
| * and\_mineral | The mineral species which appear together with this species | Should be NULL value if cross-ref is not ‘YES’ |
| * chem | The chemistry formula of the species |  |
| * cross-ref | Whether the species are crossly referenced with other species | The value should be in: 'YES', 'NO', 'SLAVE' |
| * euhedral | The crystal property of the mineral (with crystal faces or not) | The value should be in: 'YES', 'NO' |
| * group | The group of the species |  |
| * mod | The modifier of the species |  |
| * species\_id | Identifying number of the species | *Identifier* |
| * species\_name | The name of the species |  |
| * syn | The synonym of the species |  |
| * sys | The crystal systems of the species |  |
| * var | The variety of the species |  |
| * with\_mineral | The mineral species which are attached to this species. | Should be NULL value if cross-ref is not ‘SLAVE’ |
| UPDATE | Relationship to modify that information of catalog information can be updated by logged employees |  |
| UPDATE\_DETAILS | Weak Entity Class, to model the update details of catalog by login session |  |
| * update\_comment | The comment when update |  |
| * update\_date | Identifying date of update | *Partial Identifier* |
| USED\_TO\_BE | Relationship of minerals and its history usage records |  |
| USER LOGIN | Entity Class, to login details for everyone |  |
| * Login\_id | Identifies the login\_id  for the user | *Identifier* |
| * Password | identifies the password for each user |  |

# **Chapter 3**: Relational Design

The final translated relations after normalization include 35 tables and are shown as follows:

ANALYSIS\_DETAILS (analysis\_date, analyst\_id, analysis\_date, aCode, aMethod, aCategory, aResult, aComent)

ANALYST (analyst\_id, fname, lname)

ANALYST\_EMAIL (analyst\_id, analyst\_email)

ANALYST\_PHONE (analyst\_id, analyst\_phone)

APPRAISAL\_DETAILS (mAppraiserID, mMineralID, appraisal\_date, appraiser\_comments, estimated\_value)

APPRAISER (appraiser\_id, a\_fname, a\_lname)

APPRAISER\_EMAIL (appraiser\_id, appraiser\_email)

APPRAISER\_PHONE (appraiser\_id, appraiser\_phone)

CATALOG (catalog \_id, catalog\_name, color, description, num\_of\_uni\_min)

CATALOG (catalog \_id, feature)

COLLECTION\_DETAILS (collector\_id, mineral\_id, collect\_date, collect\_field, association)

COLLECTOR (collector \_id, collector\_fname, collector\_lname)

COLLECTOR\_EMAIL (collector \_id, collector\_email)

COLLECTOR\_PHONE (collector \_id, collector\_phone)

DISPLAY\_MINERAL (mineral\_id, display\_id)

DISPLAY\_UNITS (displayID, displayLocation, displayName, dEndDate, dStartDate)

DOCUMENT (doc\_id, folder\_id, mineral\_id, graphics, image, comments, reference)

EMPLOYEE (employee\_id, efname, elname, eemail, ephone, estatus, estate, ecity, estreet, ezipcode)

FOLDER (folder\_id, description, num\_of\_docs, fStorageLocation)

GUEST\_COMMENTS (commentID, guestID, mineralID, mcomment)

GUEST (guestID, guestfname, guestlname, guestemailID)

HISTORY (history\_id, mineral\_id, hname, htype\_id, htype, hstartdate, henddate, hcomment)

LOAN (loan\_id, loan\_startdate, loan\_enddate, loan\_partyname)

LOANED\_MINERAL (mineral\_id, loan\_id)

LOCATION (location\_id, lname, location\_type, country, province, county, city, details, latitude, longitude)

LOGIN\_DETAILS (login\_session, login\_id)

MEMBERS (member\_id, member\_type, member\_since, date\_of\_visit)

MINERALS (mineralid, date\_reg, m\_unit, maquis\_type, mcatalogid, mcategory, mcurtype, mcut, mcutter, mdepth, mdescription, mheight, mimage, mineral\_name, mlength, mlocalid, msize, msource, mspeciesid, mweight, mweight\_unit, mxtal\_size)

OWN\_DETAILS (owner\_id, mineral\_id, acquisition\_date, owner\_comments, price)

OWNER (owner\_id, owner\_fname, owner\_lname)

OWNER\_EMAIL (owner \_id, owner\_email)

OWNER\_PHONE (owner \_id, owner\_phone)

SPECIES (species\_id, species\_name, variety, mgroup, chemistry, crystal\_system, synonymy, modifier, cross\_ref, euhedral, and\_mineral, with\_mineral)

UPDATE\_DETAILS (catalog\_id, login\_id, update\_date, ucomment)

USER\_LOGIN (login\_id, mpassword)

The instruction about foreign key will be shown in the data dictionary as following:

*Table 2 Data Dictionary (Relational)*

|  |  |  |
| --- | --- | --- |
| Schema Construct | Data Type | Constraint |
| ANALYSIS\_DETAILS | Relation representing the weak entity class PRODUCTS | |
| * aComment | Varchar2 (256) |  |
| * analysis\_date | Date |  |
| * analyst\_id | Varchar2 (16) | Foreign Key, references ANALYST |
| * aCategory | Varchar2 (32) |  |
| * aCode | Varchar2 (16) |  |
| * aMethod | Varchar2 (32) |  |
| * mineral\_id | Varchar2 (16) | Foreign Key, references MINERALS |
| * aResult | Varchar2 (32) |  |
| Primary Key Constraint: analysis\_date, analyst\_id, analysis\_date  *FD: aCode → aCode, analysis\_date, analyst\_id, analysis\_date, aMethod, aCategory, aResult, aComent*  *FD: analysis\_date, analyst\_id, analysis\_date → aCode, analysis\_date, analyst\_id, analysis\_date, aMethod, aCategory, aResult, aComent* | | |
| ANALYST | Relation representing the entity class ANALYST | |
| * analyst\_id | Varchar2 (16) | Primary Key |
| * fname | Varchar2 (32) |  |
| * lname | Varchar2 (32) |  |
| *FD: analyst\_id → analyst\_id, fname, lname* | | |
| ANALYST\_EMAIL | Relation representing the multi-value attribute email in ANALYST | |
| * analyst\_id | Varchar2 (16) | Foreign Key, references ANALYST |
| * analyst\_email | Varchar2 (64) |  |
| Primary Key Constraint: analyst\_id, analyst\_email  *FD: analyst\_id, analyst\_email → analyst\_id, analyst\_email* | | |
| ANALYST\_PHONE | Relation representing the multi-value attribute phone in ANALYST | |
| * analyst\_id | Varchar2 (16) | Foreign Key, references ANALYST |
| * analyst\_phone | Number (10,0) |  |
| Primary Key Constraint: analyst\_id, analyst\_phone  *FD: analyst\_id, analyst\_phone → analyst\_id, analyst\_phone* | | |
| APPRAISAL\_DETAILS | Relation representing the weak entity class APPRAISAL\_DETAILS | |
| * appraisal\_date | Date |  |
| * appraiser\_comments | Varchar2 (256) |  |
| * estimated\_value | Number (10,0) |  |
| * mAppraiserID | Varchar2 (16) | Foreign Key, references APPRAISER |
| * mMineralID | Varchar2 (16) | Foreign Key, references MINERAL |
| Primary Key Constraint: mAppraiserID, mMineralID, appraisal\_date  *FD:* mAppraiserID, mMineralID, appraisal\_date *→* mAppraiserID, mMineralID, appraisal\_date, appraiser\_comments, estimated\_value | | |
| APPRAISER | Relation representing the entity class APPRAISER | |
| * appraiser\_id | Varchar2 (16) | Primary Key |
| * a\_fname | Varchar2 (32) |  |
| * a\_lname | Varchar2 (32) |  |
| *FD: appraiser\_id → appraiser\_id, a\_fname, a\_lname* | | |
| APPRAISER\_EMAIL | Relation representing the multi-value attribute email in APPRAISER | |
| * appraiser\_id | Varchar2 (16) | Foreign Key, references APPRAISER |
| * appraiser\_email | Varchar2 (64) |  |
| Primary Key Constraint: appraiser\_id, appraiser\_email  *FD: appraiser\_id, appraiser\_email → appraiser\_id, appraiser\_email* | | |
| APPRAISER\_PHONE | Relation representing the multi-value attribute phone in APPRAISER | |
| * appraiser\_id | Varchar2 (16) | Foreign Key, references APPRAISER |
| * appraiser\_phone | Number (10,0) |  |
| Primary Key Constraint: appraiser\_id, appraiser\_phone  *FD: appraiser\_id, appraiser\_phone → appraiser\_id, appraiser\_phone* | | |
| CATALOG | Relation representing the entity class CATALOG | |
| * catalog \_id | Varchar2 (16) | Primary Key |
| * catalog\_name | Varchar2 (16) |  |
| * color | Varchar2 (16) |  |
| * description | Varchar2 (128) |  |
| * num\_of\_uni\_min | Number (10,0) | Can be derived by counting the unique mineral ID in the same catalog ID |
| *FD:* catalog \_id *→* catalog \_id, catalog\_name, color, description, num\_of\_uni\_min | | |
| CATALOG\_FEATURE | Relation representing the multi-value attribute feature in CATALOG | |
| * catalog \_id | Varchar2 (16) | Foreign Key, references CATALOG |
| * feature | Varchar2 (128) |  |
| Primary Key Constraint: catalog \_id, feature  *FD: catalog \_id, feature → catalog \_id, feature* | | |
| COLLECTION\_DETAILS | Relation representing the weak entity class COLLECTION\_DETAILS | |
| * association | Varchar2 (128) |  |
| * collector\_id | Varchar2 (16) | Foreign Key, references COLLECTOR |
| * collect\_date | Date |  |
| * collect\_field | Varchar2 (32) |  |
| * mineral\_id | Varchar2 (16) | Foreign Key, references MINERALS |
| Primary Key Constraint: collector\_id, mineral\_id  *FD: collector\_id, mineral\_id → collector\_id, mineral\_id, collect\_date, collect\_field, association* | | |
| COLLECTOR | Relation representing the entity class COLLECTOR | |
| * collector\_id | Varchar2 (16) | Primary Key |
| * collector\_fname | Varchar2 (32) |  |
| * collector\_lname | Varchar2 (32) |  |
| *FD: collector\_id → collector\_id, collector\_fname, collector\_lname* | | |
| COLLECTOR\_EMAIL | Relation representing the multi-value attribute email in COLLECTOR | |
| * collector\_id | Varchar2 (16) | Foreign Key, references COLLECTOR |
| * collector\_email | Varchar2 (64) |  |
| Primary Key Constraint: collector\_id, collector\_email  *FD: collector\_id, collector\_email → collector\_id, collector\_email* | | |
| COLLECTOR\_PHONE | Relation representing the multi-value attribute phone in COLLECTOR | |
| * collector\_id | Varchar2 (16) | Foreign Key, references COLLECTOR |
| * collector\_phone | Number (10,0) |  |
| Primary Key Constraint: collector\_id, collector\_ phone  *FD: collector\_id, collector\_ phone → collector\_id, collector\_ phone* | | |
| DISPLAY\_MINERAL | Relation representing the sub-class DISPLAY\_MINERAL | |
| * display\_id | Varchar2 (16) | Foreign Key, references DISPLAY\_UNITS |
| * mineral\_id | Varchar2 (16) | Primary Key, also Foreign Key, references MINERALS |
| *FD: mineral\_id → mineral\_id, display\_id* | | |
| DISPLAY\_UNITS | Relation representing the entity class DISPLAY\_UNITS | |
| * displayID | Varchar2 (16) | Primary Key |
| * displayLocation | Varchar2 (32) |  |
| * displayName | Varchar2 (64) |  |
| * dEndDate | Date |  |
| * dStartDate | Date |  |
| *FD: displayID → displayID, displayLocation, displayName, dEndDate, dStartDate* | | |
| DOCUMENT | Relation representing the entity class DOCUMENT | |
| * comments | Varchar2 (128) |  |
| * doc\_id | Varchar2 (16) | Primary Key |
| * folder\_id | Varchar2 (16) | Foreign Key, references FOLDER |
| * graphics | Varchar2 (32) |  |
| * image | Varchar2 (32) |  |
| * mineral\_id | Varchar2 (16) | Foreign Key, references MINERALS |
| * reference | Varchar2 (32) |  |
| *FD: doc\_id → doc\_id, folder\_id, mineral\_id, graphics, image, comments, reference* | | |
| EMPLOYEE | Relation representing the entity class EMPLOYEE | |
| * ecity | Varchar2 (32) |  |
| * eemail | Varchar2 (64) |  |
| * employee\_id | Varchar2 (16) | Primary Key |
| * efname | Varchar2 (32) |  |
| * elname | Varchar2 (32) |  |
| * ephone | Number (16,0) |  |
| * estate | Varchar2 (32) |  |
| * estatus | Varchar2 (16) | Check value IN ('ONROLL','SUSPENDED', 'RESIGNED') |
| * estreet | Varchar2 (64) |  |
| * ezipcode | Number (5,0) |  |
| *FD:* employee\_id *→* employee\_id*,* efname*,* elname*,* eemail*,* ephone*,* estatus,estate*,* ecity, estreet, ezipcode | | |
| FOLDER | Relation representing the entity class FOLDER | |
| * description | Varchar2 (128) |  |
| * folder\_id | Varchar2 (16) | Primary Key |
| * num\_of\_docs | Number (8,0) | Can be derived by counting the unique document ID in the same folder ID |
| * fStorageLocation | Varchar2 (128) |  |
| *FD: folder\_id → folder\_id, description, num\_of\_docs, fStorageLocation* | | |
| GUEST\_COMMENTS | Relation representing the weak entity class GUEST\_COMMENTS | |
| * commentID | Varchar2 (16) | Primary Key |
| * guestID | Varchar2 (16) | Foreign Key, references GUEST |
| * mcomment | Varchar2 (128) |  |
| * mineralID | Varchar2 (16) | Foreign Key, references MINERALS |
| *FD: commentID → commentID, guestID, mineralID, mcomment* | | |
| GUESTS | Relation representing the entity class GUEST | |
| * guestemailID | Varchar2 (64) |  |
| * guestID | Varchar2 (16) | Primary Key |
| * guestfname | Varchar2 (32) |  |
| * guestlname | Varchar2 (32) |  |
| *FD: guestID → guestID, guestfname, guestlname, guestemailID* | | |
| HISTORY | Relation representing the entity class HISTORY | |
| * history\_id | Varchar2 (16) |  |
| * hname | Varchar2 (16) | Primary Key |
| * hstartdate | Date |  |
| * henddate | Date |  |
| * hcomment | Varchar2 (128) |  |
| * htype | Varchar2 (16) | CHECK value in: 'DISPLAY', 'VAULT', 'DEACCESS', 'LOAN', 'BORROWED' |
| * htype\_id | Varchar2 (16) |  |
| * mineral\_id | Varchar2 (16) | Foreign Key, references MINERALS |
| *FD: history\_id → history\_id, mineral\_id, hname, htype\_id, htype, hstartdate, henddate, hcomment*  *FD: htype\_id → htype* | | |
| LOAN | Relation representing the entity class LOAN | |
| * loan\_id | Varchar2 (16) | Primary Key |
| * loan\_startdate | Date |  |
| * loan\_enddate | Date |  |
| * loan\_partyname | Varchar2 (64) |  |
| *FD: loan\_id → loan\_id, loan\_startdate, loan\_enddate, loan\_partyname* | | |
| LOANED\_MINERAL | Relation representing the sub-class LOANED\_MINERAL | |
| * loan\_id | Varchar2 (16) | Foreign Key, references LOAN |
| * mineral\_id | Varchar2 (16) | Primary Key, also Foreign Key, references MINERALS |
| *FD: mineral\_id → mineral\_id, loan\_id* | | |
| LOCATION | Relation representing the entity class LOCATION | |
| * city | Varchar2 (32) |  |
| * country | Varchar2 (32) |  |
| * county | Varchar2 (32) |  |
| * details | Varchar2 (32) |  |
| * latitude | Number (8,4) |  |
| * location\_id | Varchar2 (16) | Primary Key |
| * location\_type | Varchar2 (16) | CHECK value in: 'MINE', 'QUARRY', 'CLAIM', 'PLACE' |
| * longitude | Number (8,4) |  |
| * lname | Varchar2 (32) |  |
| * province | Varchar2 (32) |  |
| *FD: location\_id → location\_id, lname, location\_type, country, province, county, city, details, latitude, longitude* | | |
| LOGIN\_DETAILS | Relation representing the entity class LOGIN\_DETAILS | |
| * login\_id | Varchar2 (16) | Foreign Key, references USER\_LOGIN |
| * login\_session | Varchar2 (16) | Primary Key |
| *FD: login\_session → login\_session, login\_id* | | |
| MEMBERS | Relation representing the sub-class MEMBERS | |
| * date\_of\_visit | Date |  |
| * member\_id | Varchar2 (16) | Primary Key, also Foreign Key, references GUESTS |
| * member\_since | Date |  |
| * member\_type | Varchar2 (16) | Check value in ('BRONZE','SILVER', 'GOLD', 'PLATINUM') |
| *FD: member\_id → member\_id, member\_type, member\_since, date\_of\_visit* | | |
| MINERALS | Relation representing the entity class MINERALS | |
| * date\_reg | Date |  |
| * m\_unit | Varchar2 (8) | CHECK VALUE IN ('CM', 'INCH') |
| * maquis\_type | Varchar2 (8) | CHECK VALUE IN ('ORIG', 'PURCH', 'DON', 'TRADE', 'COLLECT') |
| * mcatalogid | Varchar2 (16) | Foreign Key, references CATALOG |
| * mcategory | Varchar2 (8) | CHECK VALUE IN ('MIN', 'GEM', 'MET', 'SYN', 'ORG','GEOL') |
| * mcurtype | Varchar2 (16) | CHECK VALUE IN ('DISPLAY', 'VAULT', 'LOAN', 'BORROWED', 'DEACCESS') |
| * mcut | Varchar2 (32) |  |
| * mcutter | Varchar2 (32) |  |
| * mdepth | Number (8, 3) |  |
| * mdescription | Varchar2 (128) |  |
| * mheight | Number (8, 3) |  |
| * mimage | Varchar2 (32) |  |
| * mineral\_name | Varchar2 (32) |  |
| * mineralid | Varchar2 (16) | Primary Key |
| * mlength | Varchar2 (16) |  |
| * mlocalid | Varchar2 (16) | Foreign Key, references LOCATION |
| * msize | Varchar2 (8) | CHECK VALUE IN ('MIN', 'CAB', 'OVERSIZE', 'MICRO', 'TN', 'UNLIM') |
| * msource | Varchar2 (32) |  |
| * mspeciesid | Varchar2 (16) | Foreign Key, references SPECIES |
| * mweight | Number (8, 3) |  |
| * mweight\_unit | Varchar2 (4) | CHECK VALUE IN ('GM', 'CT', 'OZ') |
| * mxtal\_size | Varchar2 (32) |  |
| *FD: mineralid → mineralid, date\_reg, m\_unit, maquis\_type, mcatalogid, mcategory, mcurtype, mcut, mcutter, mdepth, mdescription, mheight, mimage, mineral\_name, mlength, mlocalid, msize, msource, mspeciesid, mweight, mweight\_unit, mxtal\_size* | | |
| OWN\_DETAILS | Relation representing the weak entity class OWN\_DETAILS | |
| * Acquisition\_date | Date |  |
| * mineral\_id | Varchar2 (16) | Foreign Key, references MINERALS |
| * owner\_id | Varchar2 (16) | Foreign Key, references OWNER |
| * owner\_comments | Varchar2 (128) |  |
| * price | Number (10, 0) |  |
| Primary Key Constraint: owner\_id, mineral\_id, acquisition\_date  *FD: owner\_id, mineral\_id, acquisition\_date → owner\_id, mineral\_id, acquisition\_date, owner\_comments, price* | | |
| OWNER | Relation representing the entity class OWNER | |
| * owner\_id | Varchar2 (16) | Primary Key |
| * owner\_fname | Varchar2 (32) |  |
| * owner\_lname | Varchar2 (32) |  |
| *FD: owner\_id → owner\_id, owner\_fname, owner\_lname* | | |
| OWNER\_EMAIL | Relation representing the multi-value attribute email in OWNER | |
| * owner\_id | Varchar2 (16) | Foreign Key, references OWNER |
| * owner\_email | Varchar2 (64) |  |
| Primary Key Constraint: owner\_id, owner\_email  *FD: owner\_id, owner\_email → owner\_id, owner\_email* | | |
| OWNER\_PHONE | Relation representing the multi-value attribute phone in OWNER | |
| * collector\_id | Varchar2 (16) | Foreign Key, references OWNER |
| * collector\_phone | Number (10,0) |  |
| Primary Key Constraint: owner\_id, owner\_phone  *FD: owner\_id, owner\_ phone → owner\_id, owner\_ phone* | | |
| SPECIES | Relation representing the entity class SPECIES | |
| * and\_mineral | Varchar2 (16) |  |
| * chemistry | Varchar2 (32) |  |
| * cross-ref | Varchar2 (16) | CHECK VALUE IN ('YES', 'NO', 'SLAVE') |
| * crystal\_system | Varchar2 (32) |  |
| * euhedral | Varchar2 (16) | CHECK VALUE IN ('YES', 'NO') |
| * mgroup | Varchar2 (32) |  |
| * modifier | Varchar2 (32) |  |
| * species\_id | Varchar2 (16) | Primary Key |
| * species\_name | Varchar2 (32) |  |
| * synonymy | Varchar2 (32) |  |
| * variety | Varchar2 (32) |  |
| * with\_mineral | Varchar2 (16) |  |
| *FD: species\_id → species\_id, species\_name, variety, mgroup, chemistry, crystal\_system, synonymy, modifier, cross\_ref, euhedral, and\_mineral, with\_mineral* | | |
| UPDATE\_DETAILS | Relation representing the weak entity class UPDATE\_DETAILS | |
| * catalog\_id | Varchar2 (16) | Foreign Key, references CATALOG |
| * login\_id | Varchar2 (16) | Foreign Key, references LOGIN\_DETAILS |
| * ucomment | Varchar2 (128) |  |
| * update\_date | Varchar2 (16) |  |
| Primary Key Constraint: catalog\_id, login\_id, update\_date  *FD:* catalog\_id, login\_id, update\_date *→* catalog\_id, login\_id, update\_date, ucomment | | |
| USER LOGIN | Relation representing the entity class USER\_LOGIN | |
| * login\_id | Varchar2 (16) | Primary Key, also Foreign Key, references EMPLOYEE |
| * mpassword | Varchar2 (16) |  |
| *FD: login\_id → login\_id, mpassword* | | |

The scripts of table creation are shown in the Appendix A.

# **Chapter 4:** Data population and Queries

You should design 10+ queries for your project. In addition to the SQL for the query, please explain in English what your query does. Your queries should be reasonably complex, and include illustrations (non-trivial) of using sub-queries and aggregation. You will need to input sufficient data in the relevant tables so that your queries return meaningful results. Typically, this involves 10-20 rows in (each of) 10 or more tables. The data does not have to be actual client data (“test data” is often preferred due to privacy concerns).

# **Chapter 5:** Triggers and Procedures

## 5.1 Triggers

**1. The mineralID trigger**

In this trigger, a mineral ID number is automatically created for a mineral each time there is an insert of new mineral data. The mineral ID starts with a prefix of ‘M’ followed by a serial number.

**create** **SEQUENCE** "DATAMATES"**.**"MINERALS\_SEQ" MINVALUE 1 MAXVALUE 9999999 **INCREMENT** **BY** 1 **START** **WITH** 61 CACHE 20 NOORDER NOCYCLE NOPARTITION **;**

**create** **or** **replace** **TRIGGER** mineralID **BEFORE**

**INSERT** **ON** MINERALS **FOR** **EACH** **ROW**

**DECLARE** mineralID MINERALS**.**mineralID**%type;**

**BEGIN**

**SELECT** 'M'

**||** minerals\_seq**.nextval**

**INTO** mineralID

**FROM** dual**;**

**:new.**mineralID **:=** mineralID**;**

**END;**

**2. The display\_mineral\_type\_update trigger**

This trigger is automatically fired each time when there’s an insert or delete on the DISPLAY\_MINERAL table. If it is a delete, the mineral current type in the MINERALS table will be updated to ‘VAULT’. If it is an insert, the mineral current type in the MINERALS table will be updated to ‘DISPLAY’.

**create** **or** **replace** **TRIGGER** display\_mineral\_type\_update **AFTER**

**INSERT** **OR** **DELETE** **ON** DISPLAY\_MINERAL **FOR** **EACH** **ROW**

**BEGIN**

**If** **(:old.**display\_id **is** **not** **null** **and** **:new.**display\_id **is** **null)** **then**

**UPDATE** MINERALS **set** mCurType **=** 'VAULT' **where** mineralID **=** **:old.**mineral\_id**;**

**ELSIF** **(:old.**display\_id **is** **null** **and** **:new.**display\_id **is** **not** **null)** **then**

**UPDATE** MINERALS **set** mCurType **=** 'DISPLAY' **where** mineralID **=** **:new.**mineral\_id**;**

**END** **IF;**

**END;**

**3. The loaned\_mineral\_type\_update\_2 trigger**

The loaned\_mineral\_type\_update\_2 trigger is fired each time after an insert or delete of data row on the LOANED\_MINERAL table. If it is a delete, the mineral current type in the MINERALS table will be updated to ‘VAULT’. If it is an insert, the mineral current type in the MINERALS table will be updated to ‘LOAN’.

**create** **or** **replace** **TRIGGER** loaned\_mineral\_type\_update\_2 **FOR**

**INSERT** **OR** **DELETE** **ON** LOANED\_MINERAL

compound **trigger**

**after** **each** **row**

**is**

**BEGIN**

**if** deleting **then**

**UPDATE** MINERALS **set** mCurType **=** 'VAULT' **where** mineralID **=** **:old.**mineral\_id**;**

**ELSIF** inserting **then**

**UPDATE** MINERALS **set** mCurType **=** 'LOAN' **where** mineralID **=** **:new.**mineral\_id**;**

**END** **IF;**

**END** **after** **each** **row;**

**end;**

**4. The document\_folder\_count\_2 trigger**

This trigger will be fired each time after an insert, delete, or update of the data row on the DOCUMENT table. If it is an update, the number of documents in the FOLDER table will stay the same. If it is a data row insert on the DOCUMENT table, the number of documents in the FOLDER table will be increased by 1. If it is a delete of data row on the DOCUMENT table, the number of documents in the FOLDER table will be decreased by 1.

**create** **or** **replace** **TRIGGER** document\_folder\_count\_2 **for**

**INSERT** **OR** **DELETE** **or** **UPDATE** **of** folder\_ID **ON** DOCUMENT

compound **trigger**

**after** **each** **row**

**is**

**BEGIN**

**if** updating **then**

**UPDATE** FOLDER **set** num\_of\_docs **=** num\_of\_docs-1 **where** folder\_ID **=** **:old.**folder\_ID**;**

**UPDATE** FOLDER **set** num\_of\_docs **=** num\_of\_docs+1 **where** folder\_ID **=** **:new.**folder\_ID**;**

**ELSIF** inserting **then**

**UPDATE** FOLDER **set** num\_of\_docs **=** num\_of\_docs +1 **where** folder\_ID **=** **:new.**folder\_ID**;**

**elsif** deleting **then**

**UPDATE** FOLDER **set** num\_of\_docs **=** num\_of\_docs-1 **where** folder\_ID **=** **:old.**folder\_ID**;**

**END** **IF;**

**END** **after** **each** **row;**

**end;**

**5. The login\_sesion trigger**

**create** **or** **replace** **TRIGGER** login\_sesion **BEFORE**

**INSERT** **ON** LOGIN\_DETAILS **FOR** **EACH** **ROW** **DECLARE** login\_sesion LOGIN\_DETAILS**.**Login\_session**%type;**

cur\_status EMPLOYEE**.**eStatus**%type;**

**BEGIN**

**SELECT** eStatus **into** cur\_status **from** EMPLOYEE **where** Employee\_id **=** **:new.**Login\_id**;**

**If** **(**cur\_status **=** 'ONROLL'**)** **then**

**SELECT** 'LD' **||** **TO\_CHAR(sysdate,** 'MMDDYYHHMMSS'**)** **||** login\_session\_seq**.nextval**

**INTO** login\_sesion

**FROM** dual**;**

**:new.**Login\_session **:=** login\_sesion**;**

**ELSE**

RAISE\_APPLICATION\_ERROR**(**'-20100'**,** 'Employee ID not valid. Please check!'**);**

**END** **IF;**

**END;**

## 5.2 Procedures

**1 The loan\_type\_report\_procedure procedure**

This procedure is used for the purpose of storing the previous usage of minerals. Once the procedure is executed, it will check the loan end date for each data row in the LOAN table. If the loan end date is already past due, the corresponding data rows in the LOAN and LOANED\_MINERAL tables will be inserted into the HISTORY table. Also if the display end date is already past due, the corresponding data rows in the DISPLAY\_UNIT and DISPLAY\_MINERAL tables will be inserted into the HISTORY table. We are able to complete these processes by using two cursors, get\_loan\_minerals and get\_display\_minerals, in this procedure. After the inserting is done in the HISTORY table, the corresponding data rows will be deleted from the LOAN and/or DISPLAY\_MINERAL table(s).

**create** **or** **replace** **PROCEDURE** loan\_type\_report\_procedure

**AS**

**CURSOR** get\_loan\_minerals

**IS**

**select** LOAN**.**loan\_id**,**LOAN**.**loan\_startDate**,**LOAN**.**loan\_endDate**,**LOAN**.**loan\_partyName**,**LOANED\_MINERAL**.**MINERAL\_ID **from** LOAN**,** LOANED\_MINERAL **where** LOAN**.**loan\_id **=** LOANED\_MINERAL**.**loan\_id **and** loan\_endDate **<** **sysdate;**

**CURSOR** get\_display\_minerals

**IS**

**select** DISPLAY\_UNITS**.**DisplayID**,**DISPLAY\_UNITS**.**DisplayName**,**DISPLAY\_UNITS**.**dStartDate**,**DISPLAY\_UNITS**.**dEndDate**,**DISPLAY\_MINERAL**.**mineral\_id **from** DISPLAY\_UNITS**,** DISPLAY\_MINERAL **where** DISPLAY\_UNITS**.**DisplayID **=** DISPLAY\_MINERAL**.**display\_id **and** dEndDate **<** **sysdate;**

**BEGIN**

**FOR** loan\_rec **in** get\_loan\_minerals **LOOP**

**INSERT**

**INTO** HISTORY

**(**

History\_id**,**

Mineral\_id**,**

hName**,**

hStartDate**,**

hEndDate**,**

hType**,**

hTypeID**,**

hComment

**)**

**VALUES**

**(**

'H1'**,**

loan\_rec**.**mineral\_id**,**

loan\_rec**.**loan\_partyName**,**

loan\_rec**.**loan\_startDate**,**

loan\_rec**.**loan\_endDate**,**

'LOAN'**,**

loan\_rec**.**loan\_id**,**

''

**);**

**END** **LOOP;**

**FOR** display\_rec **in** get\_display\_minerals **LOOP**

**INSERT**

**INTO** HISTORY

**(**

History\_id**,**

Mineral\_id**,**

hName**,**

hStartDate**,**

hEndDate**,**

hType**,**

hTypeID**,**

hComment

**)**

**VALUES**

**(**

'H1'**,**

display\_rec**.**mineral\_id**,**

display\_rec**.**DisplayName**,**

display\_rec**.**dStartDate**,**

display\_rec**.**dEndDate**,**

'DISPLAY'**,**

display\_rec**.**DisplayID**,**

''

**);**

**END** **LOOP;**

**delete** **from** LOAN **where** loan\_endDate **<** **sysdate;**

**delete** **from** DISPLAY\_MINERAL **where** display\_id **in** **(select** displayid **from** display\_units **where** denddate**<** **sysdate);**

**END;**

**2 The user\_login\_procedure\_2 procedure**

**Not sure if we need it or not???**

**create** **or** **replace** **PROCEDURE** user\_login\_procedure\_2 **(**userID LOGIN\_DETAILS**.**Login\_id**%type,** loginType **number)**

**AS**

existdata login\_details**.**login\_id**%type;**

**BEGIN**

**if(**loginType **=** 0**)** **then**

**DELETE** **FROM** LOGIN\_DETAILS **where** Login\_id **=** userID**;**

**ELSIF(**loginType **=** 1**)** **then**

dbms\_output**.**put\_line**(**'I m starting'**);**

**select** login\_id **into** existdata **from** LOGIN\_DETAILS **where** LOGIN\_ID**=**userID**;**

raise\_application\_error**(**'-20020'**,** 'User already logged in! Please end last unknown session or contact IT Department!'**);**

**END** **IF;**

**Exception**

**when** no\_data\_found **then**

**INSERT**

**INTO** LOGIN\_DETAILS

**(**

Login\_id**,**

Login\_session

**)**

**VALUES**

**(**

userID**,**

'SESSION'

**);**

**when** too\_many\_rows **then**

**DELETE** **FROM** LOGIN\_DETAILS **where** Login\_id **=** userID**;**

raise\_application\_error**(**'-20021'**,** 'Please Login again!'**);**

**END;**

# **Chapter 6**: Interface and Reports

For this chapter, you will create a cloud-based web interface to display the results of your queries that you’ve written (i.e., those in Chapter 4; additional queries are always welcome). The user should be able to easily find and execute your queries, should be able to pick any one, and upon execution, the results of the queries should display online. The report will contain a narration for a “user walk-through” including screen-captures. You should include facilities to add, edit, delete rows from at least 5 of your tables (including inserts/updates involving superclasses & subclasses or weak entity classes). Your report should list a URL (and any user/password combination that is needed to access the system). Please highlight any features you want users to observe.

# **Chapter 7**: Conclusions and implementation plan

Describe the lessons learnt from your project experience (including possibly based on presentations of other groups). Describe an implementation plan that will describe the steps needed to implement the full project on a real-world database management system (presuming the implementing consultant has your report and design easily available). Include approximate estimates of person-hours and hardware / software costs. Break down your estimates logically and with reasonable level of detail (e.g., instead of simply saying "hardware costs are XYZ ", explain how you reached that total). A layout that summarizing your estimates visually is usually easier to read. You may pick any implementation platform of your choice.

*The Client Letter*: Each group undertaking an external project should also turn in / have the client directly send in to the instructor a letter of evaluation discussing the client experience with your project team. The letter is due at the same time as the Project Final Report. Clients may email in a letter to Dr. Currim.

# **Appendix A Table Creation Script**

**CREATE** **TABLE** LOCATION

**(**

location\_ID **VARCHAR2(**16**),**

City **VARCHAR2(**32**),**

County **VARCHAR2(**32**),**

Province **VARCHAR2(**32**),**

Country **VARCHAR2(**32**),**

Latitude **NUMBER(**8**,**4**),**

Longitude **NUMBER(**8**,**4**),**

lName **VARCHAR2(**32**),**

Location\_type **VARCHAR2(**16**),**

Details **VARCHAR2(**32**),**

**CONSTRAINT** location\_id\_pk **PRIMARY** **KEY** **(**location\_ID**),**

**CONSTRAINT** local\_type\_check **CHECK** **(** **upper(**Location\_type**)** **IN** **(**'MINE'**,** 'QUARRY'**,** 'CLAIM'**,** 'PLACE'**)** **)**

**);**

**CREATE** **TABLE** **CATALOG**

**(**

catalog\_id **VARCHAR2(**16**)** **NOT** **NULL** **PRIMARY** **KEY,**

catalog\_name **VARCHAR2(**16**),**

description **VARCHAR2(**128**),**

color **VARCHAR2(**16**),**

num\_of\_uni\_min **NUMBER(**10**)**

**);**

**CREATE** **TABLE** CATALOG\_FEATURE

**(**

catalog\_id **VARCHAR2(**16**)** **NOT** **NULL,**

feature **VARCHAR2(**128**),**

**CONSTRAINT** cat\_feat\_id\_pk **PRIMARY** **KEY** **(**catalog\_id**,**feature**),**

**CONSTRAINT** cat\_feat\_fk **FOREIGN** **KEY** **(**catalog\_id**)** **REFERENCES** **CATALOG** **(**catalog\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** SPECIES

**(**

Species\_id **VARCHAR2(**16**),**

Species\_name **VARCHAR2(**32**),**

Variety **VARCHAR2(**32**),**

mGroup **VARCHAR2(**32**),**

Chemistry **VARCHAR2(**32**),**

Crystal\_system **VARCHAR2(**32**),**

Synonymy **VARCHAR2(**32**),**

Modifier **VARCHAR2(**32**),**

Cross\_ref **VARCHAR2(**16**),**

Euhedral **VARCHAR2(**16**),**

AND\_MINERAL **VARCHAR2(**16**),**

**CONSTRAINT** species\_id\_pk **PRIMARY** **KEY** **(**species\_id**),**

**CONSTRAINT** Cross\_ref\_check **CHECK** **(** **upper(**Cross\_ref**)** **IN** **(**'YES'**,** 'NO'**,** 'SLAVE'**)** **),**

**CONSTRAINT** Euhedral\_check **CHECK** **(** **upper(**Euhedral**)** **IN** **(**'YES'**,** 'NO'**)** **)**

**);**

**CREATE** **TABLE** OWNER

**(**

Owner\_id **VARCHAR2(**16**),**

Owner\_fname **VARCHAR(**32**),**

Owner\_lname **VARCHAR(**32**),**

**CONSTRAINT** owner\_id\_\_pk **PRIMARY** **KEY** **(**owner\_id**)**

**);**

**CREATE** **TABLE** OWNER\_EMAIL

**(**

Owner\_id **VARCHAR2(**16**),**

Owner\_email **VARCHAR2(**64**),**

**CONSTRAINT** owner\_email\_pk **PRIMARY** **KEY** **(**Owner\_id**,** Owner\_email**),**

**CONSTRAINT** owner\_id\_email\_fk **FOREIGN** **KEY** **(**Owner\_id**)** **REFERENCES** OWNER**(**Owner\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** OWNER\_PHONE

**(**

Owner\_id **VARCHAR2(**16**),**

Owner\_phone **NUMBER(**10**),**

**CONSTRAINT** owner\_phone\_pk **PRIMARY** **KEY** **(**Owner\_id**,** Owner\_phone**),**

**CONSTRAINT** owner\_id\_phone\_fk **FOREIGN** **KEY** **(**Owner\_id**)** **REFERENCES** OWNER**(**Owner\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** DISPLAY\_UNITS

**(**

DisplayID **VARCHAR2(**16**),**

DisplayName **VARCHAR(**32**),**

DisplayLocation **VARCHAR2(**64**),**

dStartDate **DATE,**

dEndDate **DATE,**

**CONSTRAINT** diplayu\_pk **PRIMARY** **KEY** **(**DisplayID**)**

**);**

**CREATE** **TABLE** LOAN

**(**

loan\_id **VARCHAR2(**16**),**

loan\_startDate **DATE,**

loan\_endDate **DATE,**

loan\_partyName **VARCHAR(**64**),**

**CONSTRAINT** loan\_id\_\_pk **PRIMARY** **KEY** **(**loan\_id**)**

**);**

**CREATE** **TABLE** FOLDER

**(**

folder\_ID **VARCHAR2(**16**),**

Description **VARCHAR2(**128**),**

fStorageLocation **VARCHAR2(**128**),**

num\_of\_docs **NUMBER(**8**),**

**CONSTRAINT** folder\_pk **PRIMARY** **KEY** **(**folder\_ID**)**

**);**

**CREATE** **TABLE** MINERALS

**(**

mineralID **VARCHAR2(**16**),**

mineral\_name **VARCHAR2(**32**),**

mCategory **VARCHAR2(**32**),**

date\_reg **DATE,**

mSource **VARCHAR2(**32**),**

mAquis\_type **VARCHAR2(**32**),**

mSize **VARCHAR2(**32**),**

mXtal\_Size **VARCHAR2(**32**),**

mLength **NUMBER(**8**,**3**),**

mHeight **NUMBER(**8**,**3**),**

mDepth **NUMBER(**8**,**3**),**

m\_unit **VARCHAR2(**32**),**

mWeight **NUMBER(**8**,**3**),**

mWeight\_unit **VARCHAR2(**32**),**

mCutter **VARCHAR2(**32**),**

mCut **VARCHAR2(**32**),**

mDescription **VARCHAR2(**32**),**

mRef\_Folder\_Num **VARCHAR2(**32**),**

mLocalID **VARCHAR2(**16**),**

mCatalogID **VARCHAR2(**16**),**

mSpeciesID **VARCHAR2(**16**),**

mPrevOwner **VARCHAR2(**16**),**

mPrice **NUMBER(**8**,**3**),**

mCurPrice **NUMBER(**8**,**3**),**

mCurOwner **VARCHAR2(**16**),**

mCurType **VARCHAR2(**16**),**

mImage **VARCHAR2(**128**),**

**CONSTRAINT** MINERAL\_PK **PRIMARY** **KEY** **(**mineralID**),**

**CONSTRAINT** Location\_FK **FOREIGN** **KEY** **(**mLocalID**)** **REFERENCES** LOCATION **(**location\_ID**),**

**CONSTRAINT** Catalog\_FK **FOREIGN** **KEY** **(**mCatalogID**)** **REFERENCES** **CATALOG** **(**catalog\_id**),**

**CONSTRAINT** Species\_FK **FOREIGN** **KEY** **(**mSpeciesID**)** **REFERENCES** SPECIES **(**Species\_id**),**

**CONSTRAINT** Owner\_FK **FOREIGN** **KEY** **(**mCurOwner**)** **REFERENCES** OWNER **(**Owner\_id**),**

**CONSTRAINT** category\_check **CHECK** **(** **upper(**mCategory**)** **IN** **(**'MIN'**,** 'GEM'**,** 'MET'**,** 'SYN'**,** 'ORG'**,**'GEOL'**)** **),**

**CONSTRAINT** acquisition\_check **CHECK** **(** **upper(**mAquis\_type**)** **IN** **(**'ORIG'**,** 'PURCH'**,** 'DON'**,** 'TRADE'**,** 'COLLECT'**)** **),**

**CONSTRAINT** weight\_unit\_check **CHECK** **(** **upper(**mWeight\_unit**)** **IN** **(**'GM'**,** 'CT'**,** 'OZ'**)** **),**

**CONSTRAINT** mineral\_unit\_check **CHECK** **(** **upper(**m\_unit**)** **IN** **(**'CM'**,** 'INCH'**)** **),**

**CONSTRAINT** mineral\_size\_check **CHECK** **(** **upper(**mSize**)** **IN** **(**'MIN'**,** 'CAB'**,** 'OVERSIZE'**,** 'MICRO'**,** 'TN'**,** 'UNLIM'**)** **)**

**);**

**CREATE** **TABLE** LOANED\_MINERAL

**(**

loan\_id **VARCHAR2(**16**),**

mineral\_id **VARCHAR2(**16**),**

**CONSTRAINT** loaned\_mineral\_\_pk **PRIMARY** **KEY** **(**loan\_id**,** mineral\_id**),**

**CONSTRAINT** loan\_id\_fk **FOREIGN** **KEY** **(**loan\_id**)** **REFERENCES** LOAN**(**loan\_id**),**

**CONSTRAINT** loan\_mineralid\_fk **FOREIGN** **KEY** **(**mineral\_ID**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** DISPLAY\_MINERAL

**(**

display\_id **VARCHAR2(**16**),**

mineral\_id **VARCHAR2(**16**),**

**CONSTRAINT** display\_mineral\_\_pk **PRIMARY** **KEY** **(**display\_id**,** mineral\_id**),**

**CONSTRAINT** display\_unit\_id\_fk **FOREIGN** **KEY** **(**display\_id**)** **REFERENCES** DISPLAY\_UNITS**(**DisplayID**),**

**CONSTRAINT** display\_mineralid\_fk **FOREIGN** **KEY** **(**mineral\_ID**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** DOCUMENT

**(**

document\_ID **VARCHAR2(**16**)** **NOT** **NULL** **PRIMARY** **KEY,**

folder\_ID **VARCHAR2(**16**),**

Image **VARCHAR2(**32**),**

fReference **VARCHAR2(**32**),**

Graphics **VARCHAR2(**32**),**

Comments **VARCHAR2(**128**),**

mineral\_id **VARCHAR2(**16**),**

**CONSTRAINT** doc\_mineralid\_fk **FOREIGN** **KEY** **(**mineral\_ID**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE,**

**CONSTRAINT** doc\_folderid\_fk **FOREIGN** **KEY** **(**folder\_ID**)** **REFERENCES** FOLDER**(**folder\_ID**)**

**);**

**CREATE** **TABLE** ANALYST

**(**

analyst\_ID **VARCHAR2(**16**)** **NOT** **NULL** **PRIMARY** **KEY,**

fname **VARCHAR2(**32**),**

lname **VARCHAR2(**32**)**

**);**

**CREATE** **TABLE** ANALYST\_EMAIL

**(**

analyst\_ID **VARCHAR2(**16**),**

analyst\_email **VARCHAR2(**64**),**

**CONSTRAINT** analyst\_email\_pk **PRIMARY** **KEY** **(**analyst\_ID**,** analyst\_email**),**

**CONSTRAINT** ana\_id\_email\_fk **FOREIGN** **KEY** **(**analyst\_ID**)** **REFERENCES** ANALYST**(**analyst\_ID**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** ANALYST\_PHONE

**(**

analyst\_ID **VARCHAR2(**16**),**

analyst\_phone **NUMBER(**10**),**

**CONSTRAINT** analyst\_phone\_pk **PRIMARY** **KEY** **(**analyst\_ID**,** analyst\_phone**),**

**CONSTRAINT** ana\_id\_phone\_fk **FOREIGN** **KEY** **(**analyst\_ID**)** **REFERENCES** ANALYST**(**analyst\_ID**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** ANALYSIS\_DETAILS

**(**

analyst\_ID **VARCHAR2(**16**),**

mineral\_ID **VARCHAR2(**16**),**

aCode **VARCHAR2(**16**),**

aCategory **VARCHAR2(**32**),**

aMethod **VARCHAR2(**32**),**

aResult **VARCHAR2(**32**),**

analysis\_date **DATE,**

aComment **VARCHAR2(**256**),**

**CONSTRAINT** a\_details **PRIMARY** **KEY** **(**analyst\_ID**,** mineral\_ID**,**analysis\_date**),**

**CONSTRAINT** a\_details\_analyst\_fk **FOREIGN** **KEY** **(**analyst\_ID**)** **REFERENCES** ANALYST**(**analyst\_ID**),**

**CONSTRAINT** a\_details\_mineralid\_fk **FOREIGN** **KEY** **(**mineral\_ID**)** **REFERENCES** MINERALS**(**mineralID**)**

**);**

**CREATE** **TABLE** APPRAISER

**(**

appraiser\_id **VARCHAR2(**16**)** **NOT** **NULL** **PRIMARY** **KEY,**

a\_fname **VARCHAR2(**32**),**

a\_lname **VARCHAR2(**32**)**

**);**

**CREATE** **TABLE** APPRAISER\_EMAIL

**(**

appraiser\_id **VARCHAR2(**16**),**

appraiser\_email **VARCHAR2(**64**),**

**CONSTRAINT** appraiser\_email\_pk **PRIMARY** **KEY** **(**appraiser\_id**,** appraiser\_email**),**

**CONSTRAINT** app\_id\_email\_fk **FOREIGN** **KEY** **(**appraiser\_id**)** **REFERENCES** APPRAISER**(**appraiser\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** APPRAISER\_PHONE

**(**

appraiser\_id **VARCHAR2(**16**),**

appraiser\_phone **NUMBER(**10**),**

**CONSTRAINT** appraiser\_phone\_pk **PRIMARY** **KEY** **(**appraiser\_id**,** appraiser\_phone**),**

**CONSTRAINT** app\_id\_phone\_fk **FOREIGN** **KEY** **(**appraiser\_id**)** **REFERENCES** APPRAISER**(**appraiser\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** APPRAISAL\_DETAILS

**(**

mAppraiserID **VARCHAR2(**16**),**

mMineralID **VARCHAR2(**16**),**

appraisal\_date **DATE,**

estimated\_value **NUMBER(**10**),**

appraiser\_comments **VARCHAR2(**128**),**

**CONSTRAINT** appraisal\_id\_pk **PRIMARY** **KEY** **(**mAppraiserID**,**mMineralID**,**appraisal\_date**),**

**CONSTRAINT** apr\_det\_min\_fk **FOREIGN** **KEY** **(**mMineralID**)** **REFERENCES** MINERALS**(**mineralID**),**

**CONSTRAINT** apr\_det\_apr\_fk **FOREIGN** **KEY** **(**mAppraiserID**)** **REFERENCES** APPRAISER**(**appraiser\_id**)**

**);**

**CREATE** **TABLE** COLLECTOR

**(**

collector\_id **VARCHAR2(**16**)** **NOT** **NULL** **PRIMARY** **KEY,**

collector\_fname **VARCHAR2(**32**),**

collector\_lname **VARCHAR2(**32**)**

**);**

**CREATE** **TABLE** COLLECTOR\_EMAIL

**(**

collector\_id **VARCHAR2(**16**),**

collector\_email **VARCHAR2(**64**),**

**CONSTRAINT** collector\_email\_pk **PRIMARY** **KEY** **(**collector\_id**,** collector\_email**),**

**CONSTRAINT** col\_id\_email\_fk **FOREIGN** **KEY** **(**collector\_id**)** **REFERENCES** COLLECTOR**(**collector\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** COLLECTOR\_PHONE

**(**

collector\_id **VARCHAR2(**16**),**

collector\_phone **NUMBER(**10**),**

**CONSTRAINT** collector\_phone\_pk **PRIMARY** **KEY** **(**collector\_id**,** collector\_phone**),**

**CONSTRAINT** col\_id\_phone\_fk **FOREIGN** **KEY** **(**collector\_id**)** **REFERENCES** COLLECTOR**(**collector\_id**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** COLLECTION\_DETAILS

**(**

mineral\_ID **VARCHAR2(**16**),**

collector\_id **VARCHAR2(**16**),**

collect\_date **DATE,**

collect\_field **VARCHAR2(**32**),**

association **VARCHAR2(**128**),**

**CONSTRAINT** col\_det **PRIMARY** **KEY** **(**mineral\_ID**,** collector\_id**),**

**CONSTRAINT** col\_det\_min\_fk **FOREIGN** **KEY** **(**mineral\_ID**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE,**

**CONSTRAINT** col\_det\_colid\_fk **FOREIGN** **KEY** **(**collector\_id**)** **REFERENCES** COLLECTOR**(**collector\_id**)**

**);**

**CREATE** **TABLE** EMPLOYEE

**(**

Employee\_id **VARCHAR2(**16**),**

E\_fname **VARCHAR(**32**),**

E\_lname **VARCHAR(**32**),**

ePhone **NUMBER(**16**),**

eEmail **VARCHAR(**64**),**

eStreet **VARCHAR(**64**),**

eState **VARCHAR(**32**),**

eCity **VARCHAR(**32**),**

eZipcode **NUMBER(**5**),**

eStatus **VARCHAR(**16**),**

**CONSTRAINT** employee\_pk **PRIMARY** **KEY** **(**Employee\_id**),**

**CONSTRAINT** status\_check **CHECK** **(upper(**eStatus**)** **IN** **(**'ONROLL'**,**'SUSPENDED'**,** 'RESIGNED'**)** **)**

**);**

**CREATE** **TABLE** GUESTS

**(**

GuestID **VARCHAR2(**16**),**

GuestFName **VARCHAR2(**32**),**

GuestLName **VARCHAR2(**32**),**

GuestEmailID **VARCHAR2(**64**),**

**CONSTRAINT** guest\_pk **PRIMARY** **KEY** **(**GuestID**)**

**);**

**CREATE** **TABLE** MEMBERS

**(**

memberID **VARCHAR2(**16**),**

memberType **VARCHAR2(**16**),**

date\_of\_visit **DATE,**

**CONSTRAINT** member\_pk **PRIMARY** **KEY** **(**memberID**),**

**CONSTRAINT** mem\_guest\_fk **FOREIGN** **KEY** **(**memberID**)** **REFERENCES** GUESTS**(**GuestID**)** **ON**

**DELETE** **CASCADE,**

**CONSTRAINT** member\_type\_check **CHECK** **(upper(**memberType**)** **IN** **(**'BRONZE'**,**'SILVER'**,** 'GOLD'**,** 'PLATINUM'**)** **)**

**);**

**CREATE** **TABLE** GUEST\_COMMENTS

**(**

CommentID **VARCHAR2(**16**),**

GuestID **VARCHAR2(**16**),**

MineralID **VARCHAR2(**16**),**

mComment **VARCHAR2(**128**),**

**CONSTRAINT** guestcomment\_pk **PRIMARY** **KEY** **(**CommentID**,**GuestID**,**MineralID**),**

**CONSTRAINT** guest\_comment\_fk **FOREIGN** **KEY(**GuestID**)** **REFERENCES** GUESTS**(**GuestID**),**

**CONSTRAINT** comment\_mineral\_fk **FOREIGN** **KEY(**MineralID**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** HISTORY

**(**

History\_id **VARCHAR2(**16**),**

Mineral\_id **VARCHAR2(**16**),**

hName **VARCHAR2(**16**),**

hStartDate **DATE,**

hEndDate **DATE,**

hType **VARCHAR2(**16**),**

hTypeID **VARCHAR2(**16**),**

hStatus **VARCHAR2(**16**),**

hComment **VARCHAR2(**128**),**

**CONSTRAINT** history\_pk **PRIMARY** **KEY** **(**History\_id**),**

**CONSTRAINT** history\_mineral\_fk **FOREIGN** **KEY(**Mineral\_id**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE,**

**CONSTRAINT** history\_type\_check **CHECK** **(upper(**hType**)** **IN** **(**'DISPLAY'**,** 'VAULT'**,** 'LOAN'**,** 'BORROWED'**,** 'DEACCESS'**)** **)**

**);**

**CREATE** **TABLE** USER\_LOGIN

**(**

Login\_id **VARCHAR2(**16**),**

mPassword **VARCHAR(**32**),**

**CONSTRAINT** login\_id\_pk **PRIMARY** **KEY** **(**login\_id**),**

**CONSTRAINT** emp\_id\_fk **FOREIGN** **KEY** **(**Login\_id**)** **REFERENCES** EMPLOYEE**(**Employee\_id**)**

**);**

**CREATE** **TABLE** LOGIN\_DETAILS

**(**

Catalog\_id **VARCHAR2(**16**),**

Login\_id **VARCHAR2(**16**),**

Login\_sesion **VARCHAR2(**16**),**

**CONSTRAINT** log\_date\_pk **PRIMARY** **KEY** **(**Login\_sesion**),**

**CONSTRAINT** cat\_id\_fk **FOREIGN** **KEY** **(**catalog\_id**)** **REFERENCES** **CATALOG(**catalog\_id**),**

**CONSTRAINT** log\_cid\_fk **FOREIGN** **KEY** **(**login\_id**)** **REFERENCES** USER\_LOGIN**(**Login\_id**)**

**);**

**CREATE** **TABLE** OWN\_DETAILS

**(**

Mineral\_id **VARCHAR2(**16**),**

Owner\_id **VARCHAR2(**16**),**

Acquisition\_date **DATE,**

Price **NUMBER(**10**),**

Owner\_comments **VARCHAR(**128**),**

**CONSTRAINT** min\_id\_pk **PRIMARY** **KEY** **(**mineral\_id**,**owner\_id**,**Acquisition\_date**),**

**CONSTRAINT** owner\_id\_fk **FOREIGN** **KEY(**Owner\_id**)** **REFERENCES** OWNER**(**Owner\_id**),**

**CONSTRAINT** own\_min\_fk **FOREIGN** **KEY(**Mineral\_id**)** **REFERENCES** MINERALS**(**mineralID**)** **ON**

**DELETE** **CASCADE**

**);**

**CREATE** **TABLE** UPDATE\_DETAILS

**(**

Update\_date **VARCHAR2(**16**),**

Catalog\_id **VARCHAR2(**16**),**

Employee\_id **VARCHAR2(**16**),**

uComment **VARCHAR2(**256**),**

**CONSTRAINT** update\_id\_pk **PRIMARY** **KEY** **(**Update\_date**,**Employee\_id**),**

**CONSTRAINT** update\_date\_pk **FOREIGN** **KEY(**Catalog\_id**)** **REFERENCES** **CATALOG(**catalog\_id**),**

**CONSTRAINT** emp\_id\_pk **FOREIGN** **KEY** **(**Employee\_id**)** **REFERENCES** EMPLOYEE**(**Employee\_id**)**

**);**

**COMMIT;**