CSCI 4370 Term Project

Initial Steps

Title

Lunar Reconnaissance Orbiter (LRO) Image Storage and Interfacing

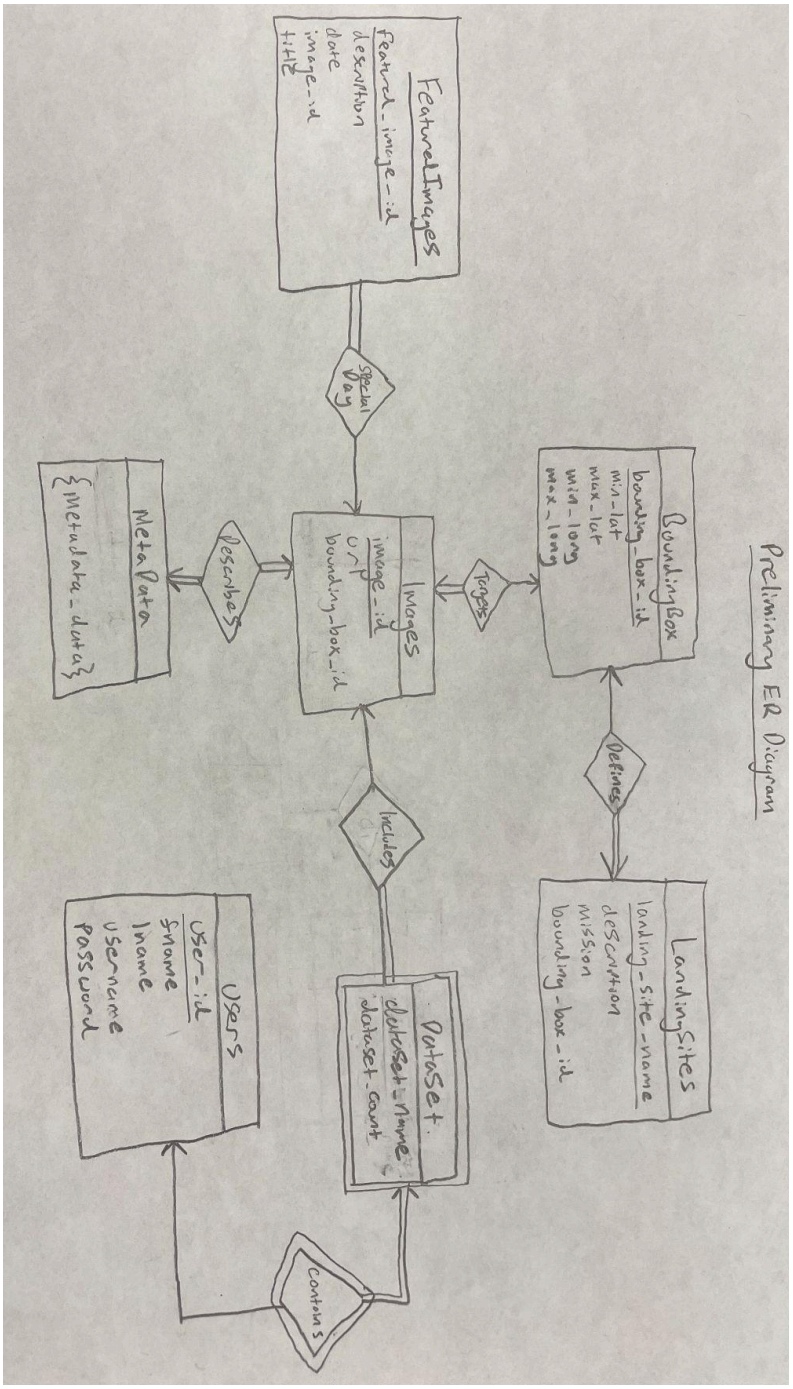
Problem Description

The LRO satellite is a NASA spacecraft that has been orbiting the Moon since its launch in 2009. The mission of LRO was to capture images of the surface and reconstruct them into 3-D maps at 100-meter resolution. This mission was accomplished, as LRO has constructed these maps for upwards of 98% of the Moon's surface. An issue arises when thinking about the storage of these images and their associated metadata. Users should be able to access data regarding images, targets (craters), metadata, and other information associated with the data products produced by the mapping mission.

Solution Description and User Interfaces

Our solution to the issue is to create a web application that will allow users to view all images and craters, as well as individual images and craters. In this individual view more in depth information will be available, and users will have the ability to create their own datasets. Datasets will include images, craters, their bounding boxes, and metadata associated with images. There will also be a functionality that shows featured images, which are on a daily basis.

Preliminary ER Diagram



Technologies Used

Java, Spring Boot, JDBC connection, Docker (MySQL), Maven

Database Design

ER to Table Conversion

The initial ER model was created based on our conceptual understanding and desire for how the project would function in support of the listed requirements. The conversion process was completed using the rules outlined in the course lecture notes, resulting in the tables below.

FeaturedImages(featured_image_id, description, date, title, image_id <fk, not null>)
DataSet(user_id <fk>, dataset_name, dataset_size, image_id <fk, not null>)
Users(user_id, fname, lname, username, password)
Images(image_id, url, metadata_data, bounding_box_id <fk>)
LandingSites(landing_site_name, description, mission, bounding_box_id<fk>)
BoundingBox(bounding_box_id, min_lat, max_lat, min_long, max_long)

Notes on ER -> Table conversion rules used

1. FeaturedImages and Images share many/one relationship set, with the many side being total.
    a. Calls for PK of Images to be added to FeaturedImages as foreign key with non null constraint.
2. DataSet is weak entity identified by Users.
    a. Relation with attributed of DataSet and relationship set created, and Pk set to discriminator attributed (none) combined with PK of Users.
    b. DataSet and Users also share one/one relationship with one side being total.
        i. Calls for PK of Users to be added to DataSet as foreign key.
3. DataSet and Images share one/one relationship set, with one side being total.
    a. Calls for PK of images to be added to DataSet as foreign key.
4. BoundingBox and Images share one/one relationship with one side being total.
    a. Calls for PK of BoundingBox to be added to Images as FK.
5. LandingSites and BoundingBox share one/one relationship with one side being total.
    a. Calls for PK of BoundingBox to be added to LandingSites as FK.
6. Image and MetaData share one/one relationship set with both sides being total.
    a. Calls for relations to be combined.

The functional dependencies listed below were also generated from the ER diagram and were used for normalizing the database schema to 3NF form.

F =

{
    Featured_image_id -> FI_description, date, title, image_id
    User_id, dataset_name, dataset_size -> image_id
    User_id -> fname, lname, username, password
    Image_id -> url, metadata_data, bounding_box_id

Landing_site_name -> LS_description, mission, bounding_box_id
Bounding_box_id -> min_lat, max_lat, min_long, max_long
}

This can be rewritten as

F = {

A -> B, C, D
E, T, U -> D
E -> F, G, H, I
D -> J, K, L
M -> N, O, L
L -> P, Q, R, S

}

Where
A = featured_image_id
B = FI_description
C = date
D = image_id
E = user_id
F = fname
G = lname
H = username
I = password
J = url
K = metadata_data
L = bounding_box_id
M = landing_site_name
N = LS_description
O = mission
P = min_lat
Q = max_lat
R = min_long
S = max_long
T = dataset_name
U = dataset_size

Normalizing to 3NF Form

3NF Synthesis

The 3NF synthesis was performed using the procedures outlined in the course lecture notes. It consists of five total steps:

1. Find minimal basis
2. Merge FDs in minimal basis with same LHS
3. Form table for each FD
4. Remove tables that are subsets of another
5. Ensure at least one table contains the global key

Minimal Basis

Finding the minimal basis of the FDs acquired from the ER model was done following the procedures outlined in the course lecture notes. It consists of three parts:

1. Split RHS of each FD
2. Removing attributes from LHS of each FD, as possible
3. Remove FDs, as possible

Splitting FD RHS

F = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S

}

Removing attributes from LHS

The only LHS with more than one attribute in the alpha set is ETU -> D.

Trying to remove attribute E,

TU+: TU

Trying to remove attribute T,
EU+: EUFGHI

Trying to remove attribute U,
ET+: ETFGHI

None of the closed attribute sets from the removed attribute LHS contain D, therefore E, T, and U are all not redundant in ETU -> D and cannot be removed.

Removing FDs

Trying A -> B

F1 = {

A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S

}

A -> B cannot be inferred from F1 = F - {A -> B}, so A -> B cannot be removed.

Trying A -> C

F2 = {

A -> B
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J

D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

A -> C cannot be inferred from F2 = F - {A -> C}, so A -> C cannot be removed.

Trying A -> D

F3 = {

A -> B
A -> C
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

A -> D cannot be inferred from F3 = F - {A -> D}, so A -> D cannot be removed.

Trying E -> F

F4 = {

A -> B
A -> C
A -> D
ETU -> D
E -> G
E -> H

E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S

}

E -> F cannot be inferred from F4 = F - {E -> F}, so E -> F cannot be removed.

Trying E -> G

F5 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S

}

E -> G cannot be inferred from F5 = F - {E -> G}, so E -> G cannot be removed.

Trying E -> H

F6 = {

A -> B
A -> C

A -> D
ETU -> D
E -> F
E -> G
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

E -> H cannot be inferred from F6 = F - {E -> H}, so E -> H cannot be removed.

Trying E -> I

F7 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

E -> I cannot be inferred from F7 = F - {E -> I}, so E -> I cannot be removed.

Trying D -> J

F8 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

D -> J cannot be inferred from F8 = F - {D -> J}, so D -> J cannot be removed.

Trying D -> K

F9 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

D -> K cannot be inferred from F9 = F - {D -> K}, so D -> K cannot be removed.

Trying D -> L

F10 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S

}

D -> L cannot be inferred from F10 = F - {D -> L}, so D -> L cannot be removed.

Trying M -> N

F11 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S

}

M -> N cannot be inferred from F11 = F - {M -> N}, so M -> N cannot be removed.

Trying M -> O

F12 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> L
L -> P
L -> Q
L -> R
L -> S

}

M -> O cannot be inferred from F12 = F - {M -> O}, so M -> O cannot be removed.

Trying M -> L

F13 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
L -> P
L -> Q
L -> R
L -> S

}

M -> L cannot be inferred from F13 = F - {M -> L}, so M -> L cannot be removed.

Trying L -> P

F14 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> Q
L -> R
L -> S

}

L -> P cannot be inferred from F14 = F - {L -> P}, so L -> P cannot be removed.

Trying L -> Q

F15 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> R
L -> S

}

L -> Q cannot be inferred from F15 = F - {L -> Q}, so L -> Q cannot be removed.

Trying L -> R

F16 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> S

}

L -> R cannot be inferred from F16 = F - {L -> R}, so L -> R cannot be removed.

Trying L -> S

F17 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P

L -> Q
L -> R
}

L -> S cannot be inferred from F17 = F - {L -> S}, so L -> S cannot be removed.

Trying ETU -> D

F18 = {

A -> B
A -> C
A -> D
ETU -> D
E -> F
E -> G
E -> H
E -> I
D -> J
D -> K
D -> L
M -> N
M -> O
M -> L
L -> P
L -> Q
L -> R
L -> S
}

ETU -> D cannot be inferred from F18 = F - {ETU -> D}, so ETU -> D cannot be removed.

Merging FDs with same LHS

Merging,

F = {
A -> B, C, D
ETU -> D
E -> F, G, H, I
D -> J, K, L
M -> N, O, L
L -> P, Q, R, S
}

Forming Table for each FD

Tables are as follows, rewritten in full name form.

FeaturedImages(featured_image_id, FI_description, date, image_id <fk, NN>)
Users(user_id, fname, lname, username, password)
Images(image_id, url, metadata_data, bounding_box_id <fk>)
LandingSites(landing_site_name, LS_description, mission, bounding_box_id <fk>)
BoundingBoxes(bounding_box_id, min_lat, max_lat, min_long, max_long)

Underline indicated primary key, fk stands for foreign key, NN stands for not null constraint.

Removing subset tables

No subset tables, so no tables removed.

Global Key

R = (ABCDEFGHIJKLMNOPQRSTU)

AEMTU+: ABCDEFGHIJKLMNOPQRSTU

The global key is AEMTU = (featured_image_id, user_id, landing_site_name, database_name, database_size), so will need to make additional table that contains AEM.

Final Normalized Relations

GlobalTable(featured_image_id <fk>, user_id <fk>, landing_site_name <fk>, database_name, database_size)
DataSet(user_id <fk>, dataset_name, dataset_size, image_id <fk, NN>)
FeaturedImages(featured_image_id, FI_description, date, image_id <fk, NN>)
Users(user_id, fname, lname, username, password)
Images(image_id, url, metadata_data, bounding_box_id <fk>)
LandingSites(landing_site_name, LS_description, mission, bounding_box_id <fk>)
BoundingBoxes(bounding_box_id, min_lat, max_lat, min_long, max_long)