Jean Paul Uwimana

ST-659 Project Deliverable Part II

Part 1: [Design 2](#ytnb3617cbyn)

[Project Narrative 2](#ytnb3617cbyn)

[Data Dictionary 3](#xcokosku03c8)

[Data Questions 4](#w69jhpq4zm1d)

[Entity Relationship Diagram 5](#74lglc1rbol0)

[Normalized Logical Model 6](#ksng2jg9amrt)

Part 2: [Implementation 7](#phjr8tqoxv9q)

[SQL DDL 7](#anqq998gw31w)

[SQL DDL (Programming objects) 10](#fw56rgqetuse)

[SQL DML 13](#6dp4eufs9kbh)

INSERT Statements

[DQTeam Table 13](#22e98txfs5f)

[Status Table 13](#4wyyr5jjktwg)

[Request Table 13](#t49t5m62184q)

[Requestor Table 14](#yby8eh4undi5)

[InternalTeam Table 14](#pv9il4hhnp1b)

[RequestLine Table 15](#cn688picr0kt)

[StatusLine Table 15](#kxjmgelasb5d)

[SQL DML](#vau9rmj5lvr4)

[SELECT - Programming Objects 15](#vau9rmj5lvr4)

[GUI Prototype - R scripting 16](#1aq0p3zcedj5)

Part 1: Design

Project Narrative

At work, my team is responsible for collecting, analyzing, validating data from a source data which is owned by another team within our organization. This internal team is thus responsible for correcting any data discrepancies identified by my team. After this data has validated clean, my team is then responsible for distributing it to various clients outside our organization.

The data is delivered per request by these clients. Currently, my team receives these data requests through email which is very cumbersome and overwhelming due to the amount of emails my team receives on a hourly/daily basis.

My team would like to have a tool that would help us manage these data requests and track information including: requestor name, request date, email address, phone number. Requestor may be an employee or external client.

We would also like to maintain specific data that allow the processing of these data requests including: item part number (item for which the data is being requested), serial number, cage code, intended destination (country), data request description(short description of the issue needing the data). This information will be submitted by the person submitting the data request to our team.

In addition, we would also like to maintain a generalized list of our internal teams we coordinate with so that they can update the status of these requests whenever they have finished doing their part of the job in order to fulfill the request.

Within the tool where the data request is logged, we (my team) would like to be able to track its status, who submitted it to our team (the client aka the requestor), and whether it’s been claimed by an analyst or not. The request status could take a number of options such as whether our team is still waiting on some other team (kind of prerequisite) that has to happen before my team could complete the request, unclaimed or to completed.

A data request can only be worked on by one analyst (member of my team) at a time, but an analyst can work on multiple requests at the same time. Our clients (requestors) submitting requests are only be able to submit it to my team as a whole, not to an individual analyst as they don’t know the availability of each individual analyst.

**Data Dictionary**

With the described data problem above, I have identified the following list of entities and their attributes.

|  |  |  |
| --- | --- | --- |
| **Entity** | **Attribute** | **Properties** |
| DQTeam | AnalystID | Required and Unique |
|  | AnalystName | Required and composite of first and last name |
|  | DQTeam Email | Required |
| Request | RequestID | Required and Unique |
|  | SerialNumber | Required |
|  | PartNumber | Required |
|  | Description | Required |
|  | RequestDate | Required DEFAULT to GETDATE() |
| Status | StatusName | Required and Unique |
| Requestor | ClientID | Required and unique |
|  | Name | Required composite of first and last name |
|  | Phone |  |
|  | Email | Required |
| InternalTeam | TeamName | Required and Unique |
|  | Email | Required and Unique |
| RequestLine | RequestLineID | Required and Unique |
|  | ClientID | Required |
|  | RequestID | Required |
| StatusLine | StatusLineID | Required and Unique |
|  | AnalystID | Required |
|  | StatusName | Required |
|  | InternalTeamName | Required |
|  | RequestID | Required and Unique |
| InternalTeamStatus | InternalTeamStatusID | Required and Unique |
|  | StatusName | Required |
|  | TeamName | Required |

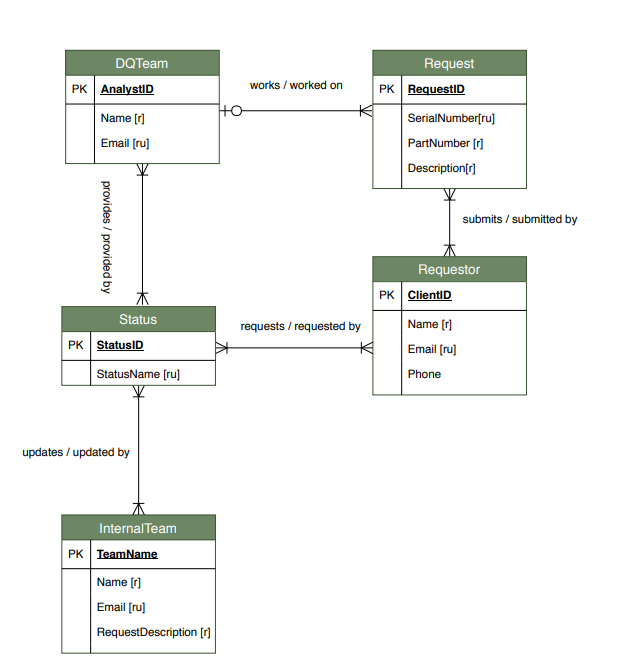
Data Questions

Some questions my Data Quality Team hopes to answer using this database are:

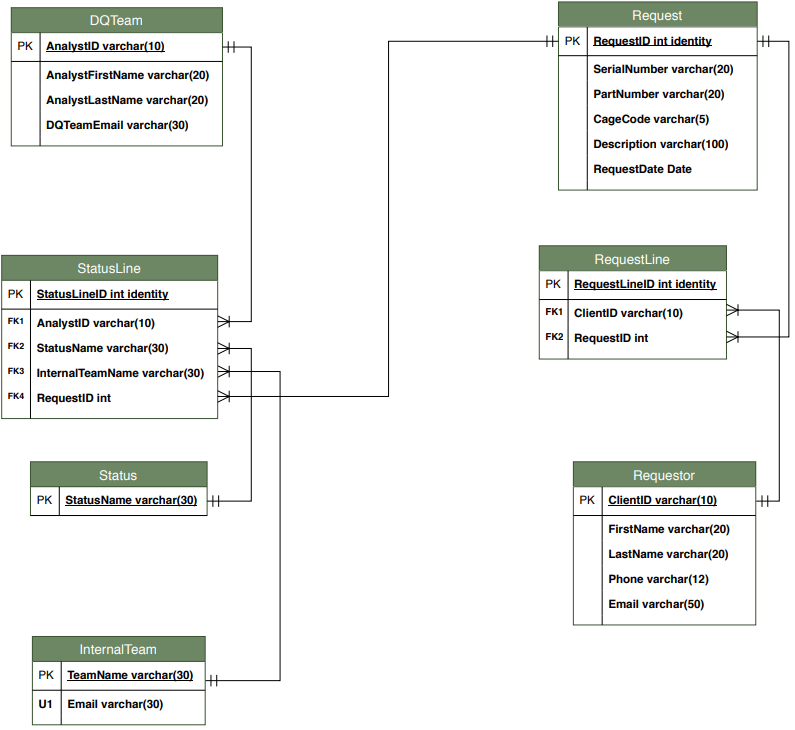
1. Which requests are in pending status and from which client?
2. Which requests have been completed for a particular day?
3. For any given day, which requests are being worked on by a particular analyst?
4. For any given day, which requests are pending action from other teams?
5. Which requests have not yet been claimed by an analyst?

Entity Relationship Diagram

When connected via relationship, the ERD is as follows:



Logical Model Diagram



Part 2: Implementation

**SQL DDL**

-- Course Project Deliverable II

-- Creating DQTeam table

CREATE TABLE DQTeam

(

-- columns for the DQTeam table

AnalystID varchar(10),

DQTeamEmail varchar(30) NOT NULL,

AnalystFirstName varchar(20) NOT NULL,

AnalystLastName varchar(20) NOT NULL

-- Constraints on the DQTeam table

CONSTRAINT PK\_DQTeam PRIMARY KEY (AnalystID),

)

-- Creating Request table

CREATE TABLE Request

(

-- columns for the Request table

RequestID int identity NOT NULL,

SerialNumber varchar(20) NOT NULL,

PartNumber varchar(20) NOT NULL,

CageCode varchar(5) NOT NULL,

Description varchar(100) NOT NULL,

RequestDate Date DEFAULT GETDATE() NOT NULL

-- Constraints on the Request table

CONSTRAINT PK\_Request PRIMARY KEY (RequestID)

)

-- Creating Status table

CREATE TABLE Status

(

-- Columns for the Status table

StatusName varchar(30) NOT NULL

--Constraints on the Status table

CONSTRAINT PK\_Status PRIMARY KEY (StatusName)

)

-- Creating Requestor table

CREATE table Requestor

(

-- Columns for the Requestor table

ClientID varchar(10) NOT NULL,

FirstName varchar(20) NOT NULL,

LastName varchar(20) NOT NULL,

Phone varchar(12) NOT NULL,

Email varchar(50) NOT NULL

-- Constraints on the Requestor table

CONSTRAINT PK\_Requestor PRIMARY KEY (ClientID)

)

-- Creating InternalTeam table

CREATE TABLE InternalTeam

(

-- Columns for the InternalTeam table

TeamName varchar(30) NOT NULL,

Email varchar(30) NOT NULL

-- Constraints on the InternalTeam table

CONSTRAINT PK\_InternalTeam PRIMARY KEY (TeamName),

CONSTRAINT U1\_InternalTeam UNIQUE (Email)

)

-- Creating RequestLine table

CREATE TABLE RequestLine

(

-- Columns for the RequestLine table

RequestLineID int identity NOT NULL,

ClientID varchar(10) NOT NULL,

RequestID int NOT NULL

-- Constraints on the RequestLine table

CONSTRAINT PK\_RequestLine PRIMARY KEY (RequestLineID),

CONSTRAINT FK1\_RequestLine FOREIGN KEY (ClientID) REFERENCES Requestor (ClientID),

CONSTRAINT FK2\_RequestLine FOREIGN KEY (RequestID) REFERENCES Request (RequestID)

)

-- Creating StatusLine table

CREATE TABLE StatusLine

(

-- Columns for the StatusLine table

StatusLineID int identity NOT NULL,

AnalystID varchar(10) NOT NULL,

StatusName varchar(30) NOT NULL,

InternalTeamName varchar(30) NOT NULL,

RequestID int

-- Constraints on the StatusLine table

CONSTRAINT PK\_StatusLine PRIMARY KEY (StatusLineID),

CONSTRAINT U1\_StatusLine UNIQUE (RequestID),

CONSTRAINT FK1\_StatusLine FOREIGN KEY (AnalystID) REFERENCES DQTeam (AnalystID),

CONSTRAINT FK2\_StatusLine FOREIGN KEY (StatusName) REFERENCES Status (StatusName),

CONSTRAINT FK4\_StatusLine FOREIGN KEY (InternalTeamName) REFERENCES InternalTeam (TeamName),

CONSTRAINT FK5\_StatusLine FOREIGN KEY (RequestID) REFERENCES Request (RequestID)

)

DDL - Programming Objects

-- Programming objects --

-- Create a function that returns the count of Requests made by a particular user

GO

CREATE FUNCTION dbo.RequestCount(@ClientID int)

RETURNS int AS

BEGIN

DECLARE @returnValue int

SELECT @returnValue = COUNT(RequestLine.ClientID)

FROM Requestor

JOIN RequestLine ON Requestor.ClientID = RequestLine.ClientID

JOIN Request ON Request.RequestID = RequestLineID

WHERE RequestLine.ClientID = @ClientID

--Return @returnValue to the calling code

RETURN @returnValue

END

-- Testing Function

SELECT TOP 10

\*,

dbo.RequestCount(ClientID) as RequestCount

FROM Requestor

ORDER BY RequestCount DESC

-- Create View to retrieve the top 10 data Requestors and their Request Counts

GO

CREATE VIEW MostActiveRequestors AS

SELECT TOP 10 \* ,

dbo.RequestCount(ClientID) AS RequestCount

FROM Requestor

WHERE dbo.RequestCount(ClientID) > 0

ORDER BY RequestCount DESC

GO

-- Testing the View

SELECT \* FROM MostActiveRequestors

-- Create View to see requests that are NOT in hands of my team (DQTeam - Data Quality Team)

-- Requests that are pending actions from other Teams, not my team

GO

CREATE VIEW RequestByOtherTeams AS

SELECT Requestor.FirstName, Requestor.LastName, StatusLine.StatusName, Request.RequestDate, Request.Description

FROM Requestor

JOIN RequestLine ON Requestor.ClientID = RequestLine.ClientID

JOIN Request ON RequestLine.RequestID = Request.RequestID

JOIN StatusLine ON RequestLine.RequestID = Request.RequestID

WHERE StatusLine.StatusName != 'Claimed by DQTeam'

GO

-- Testing the RequestOtherTeams View

Select \* from RequestByOtherTeams ORDER BY LastName

-- Create procedure to assign Requests with a specific Status to a particular group

-- The first parameter is the StatusName (status of request)

-- The second parameter is the Team to be assigned a request to

GO

CREATE PROCEDURE AssignRequestToTeam(@StatusName varchar(30), @TeamName varchar(30))

AS

BEGIN

UPDATE StatusLine SET StatusName = @StatusName

WHERE InternalTeamName = @TeamName

END

GO

-- Execute the procedure

EXEC AssignRequestToTeam 'Assigned to FDM', 'OBPHM'

-- Create View to query requests that are in pending status (NOT Submitted to LDM in this case)

-- Completed requests will have the Status of either: 'Available in LDM' or 'Submitted to LDM'. All others are pending

GO

CREATE VIEW PendingRequests AS

SELECT Requestor.FirstName + ' ' + Requestor.LastName AS DataRequestor, Status.StatusName, Request.RequestDate, Request.Description

FROM Requestor

JOIN RequestLine ON Requestor.ClientID = RequestLine.ClientID

JOIN Request ON RequestLine.RequestID = Request.RequestID

JOIN StatusLine ON Request.RequestID = StatusLine.RequestID

JOIN Status ON RequestLine.RequestID = StatusLine.RequestID

WHERE Status.StatusName NOT LIKE 'LDM'

GO

-- Testing the view

Select \* From PendingRequests ORDER BY RequestDate

-- Create View that retrieves Claimed requests

-- Requests that are being worked by my team, DQTeam

GO

CREATE VIEW ClaimedRequests AS

SELECT Requestor.FirstName + ' ' + Requestor.LastName AS DataRequestor, Status.StatusName, Request.RequestDate, Request.Description

FROM Requestor

JOIN RequestLine ON Requestor.ClientID = RequestLine.ClientID

JOIN Request ON RequestLine.RequestID = Request.RequestID

JOIN StatusLine ON Request.RequestID = StatusLine.RequestID

JOIN Status ON RequestLine.RequestID = StatusLine.RequestID

WHERE Status.StatusName = 'Claimed by DQTeam'

GO

-- Testing the above View

Select \* From ClaimedRequests

GO

CREATE VIEW WhoIsWorkingWhat AS

SELECT DQTeam.AnalystID, DQTeam.AnalystFirstName, Request.Description, StatusLine.StatusName

FROM DQTeam

JOIN StatusLine ON DQTeam.AnalystID = StatusLine.AnalystID

JOIN Request ON Request.RequestID = StatusLine.RequestID

GO

-- Testing the above View

SELECT \* FROM WhoIsWorkingWhat

DML: INSERT and Update

-- Beginning of INSERT Statements --

-- Inserting into DQTeam table

INSERT INTO DQTeam (AnalystID, AnalystFirstName, AnalystLastName, DQTeamEmail)

VALUES

('m310000', 'Jean Paul', 'Uwimana', 'JeanPaul.Uwimana@amazon.com'),

('m310009', 'Leila', 'Khalaf', 'Leila.Khalaf@amazon.com'),

('m310295', 'Steve', 'Lucas', 'Steve.Lucas@amazon.com'),

('S167512', 'Bruno', 'Mullins', 'Bruno.Mullins@Maecenas.org'),

('N660032', 'Rudyard', 'Olson', 'Rudyard.Olson@eteuismodet.edu')

-- Updating Analyst email adress

UPDATE DQTeam SET DQTeamEmail = 'Steve.Lucas@glorious.com'

WHERE DQTeamEmail = 'Steve.Lucas@amazon.com'

-- Inserting into Request table select \* from request

INSERT INTO Request (SerialNumber, PartNumber, CageCode, Description)

VALUES

('TSGCAA0900', '5100072', '73030', 'IDMS SCU showed up at Fort Worth without electronic data (EEL). Please submit EEL ASAP'),

('TSGCAB0212', '5100049', '77400', 'Jet engine at Eglin without electronic data (EEL)'),

('XVSABA4546', '5132056', '06456', 'Submit EEL for Washington DC retrieval'),

('TSGCAB0976', '4138668', '77263', 'Submit all data related to the power module in Middletown, CT'),

('60063255363', 'PW83673', '03208', 'Gwen request EEL for Fan Module, please send to LDM by COB today'),

('T5TGOG0984', '4451993', '78925', 'Please submit data ASAP to avoid further aircraft grounding'),

('T4TGOG0239', '4695982', '77603', 'An AR for AF data has been submitted but there was no follow-up. Please refer to AR:63562'),

('QAGGOG00376', '4509726', '77350', 'Ferry for BF-90 is coming for Italy. Please data for reconciliation'),

('9500202052', '4289239', '79179', 'Need data for SN: 9500202052 PN: 4289239'),

('T7TGOG0PGY', '4876436', '79091', 'Fort Worth is requesting that you confirm DOI for the above IDMS')

-- Inserting into Status table

INSERT INTO Status (StatusName)

VALUES

('Available in LDM'),

('Submitted to LDM'),

('Claimed by DQTeam'),

('Assigned to FDM'),

('Waiting for PADL file'),

('Assigned to OBPHM'),

('Rejected by DQIM'),

('Assigned to PAIR')

-- Inserting into Requestor table

INSERT INTO Requestor (ClientID, FirstName, LastName, Phone, Email)

VALUES

('1882772', 'Brice', 'Butdorf', '817-555-1212', 'Brice.Butdorf@elf.com'),

('1673880', 'Kygo', 'LaRose', '817-555-1200', 'Kygo.LaRose@elf.com'),

('1875466', 'Peter', 'Andrew', '817-555-1000', 'Peter.Andrew@elf.com'),

('1882958', 'Lamirou', 'Gwen', '800-555-9388', 'Lamirou.Gwen@mode.com'),

('1620092', 'Salvador', 'Waters', '602-147-4725', 'Donec.feugiat.metus@amet.net'),

('1678102', 'Joel', 'Wilder', '911-180-2758', 'magna.et@egestasSed.ca'),

('1625080', 'Hector', 'Strickland', '807-309-1324', 'interdum.enim@eros.net') ,

('1677040', 'Peter', 'Lewis', '800-409-6905', 'mattis.semper.dui@egestasnunc.edu'),

('1626101', 'Thane', 'Baxter', '484-672-4085', 'aliquet.diam.Sed@velitAliquam.net'),

('1686062', 'Quentin', 'Guerra', '635-684-6126', 'inceptos.hymenaeos@ac.ca')

-- Updating a Client contact information on Requestor Table --

UPDATE Requestor SET Phone = '817-456-0000', Email = 'Brice.Butdorf@lion.com'

WHERE FirstName = 'Brice' AND LastName = 'Butdorf'

-- Updating ClientID of Requestor Table

-- First step we need to alter table first due to foreign key constraint on RequestLine Table

ALTER TABLE RequestLine

DROP CONSTRAINT FK1\_RequestLine

-- Second step would be to update record

UPDATE Requestor SET ClientID = '1671000'

WHERE ClientID = '1678102'

-- Finally, adding the Constraint back to the RequestLine Table

ALTER TABLE RequestLine ADD CONSTRAINT FK1\_RequestLine FOREIGN KEY (ClientID) REFERENCES Requestor (ClientID)

-- End of Updating Requestor Table --

-- Inserting into InternalTeam table

INSERT INTO InternalTeam (TeamName, Email)

VALUES

('Fleet Data Mgmt', 'FDM@google.com'), -- FDM

('OBPHM', 'OBPHM@google.com'), -- OBPHM

('PAIR', 'PAIR.Ptr@google.com'), -- PAIR

('Supply Chain', 'SupplyChain@business.biz') -- SCM

-- Inserting into RequestLine table

INSERT INTO RequestLine (ClientID, RequestID)

VALUES

('1620092', 1),

('1625080', 2),

('1625080', 3),

('1677040', 6),

('1677040', 10)

-- Inserting into StatusLine table

INSERT INTO StatusLine (AnalystID, StatusName, InternalTeamName, RequestID)

VALUES

('m310000', 'Assigned to FDM', 'Fleet Data Mgmt', 1),

('m310295', 'Submitted to LDM', 'Supply Chain', 3),

('N660032', 'Assigned to FDM', 'Fleet Data Mgmt', 4),

('m310000', 'Waiting for PADL file', 'OBPHM', 6),

('S167512','Assigned to FDM', 'Fleet Data Mgmt', 10),

('S167512','Claimed by DQTeam', 'Supply Chain', 7),

('N660032','Assigned to OBPHM', 'OBPHM', 9)

**SQL DML - Programming Objects**

-- Programming Object names are highlighted in blue

-- Using a pre-created function that returns the count of Requests made by a particular user

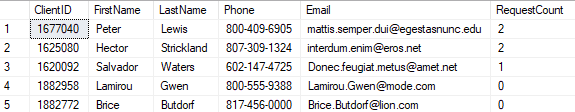
SELECT TOP 10

\*,

dbo.RequestCount(ClientID) as RequestCount

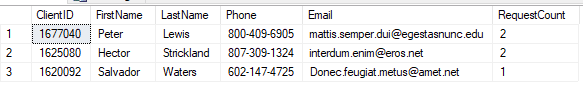
FROM Requestor

ORDER BY RequestCount DESC



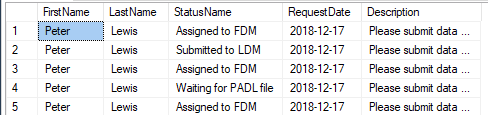
-- Querying View that shows the most active data requestors

SELECT \* FROM MostActiveRequestors



/\*-- Requests that are pending actions from other Teams, not my own team and who the data requestor is \*/

Select \* from RequestByOtherTeams ORDER BY LastName



-- Using a stored procedure that assigns requests from a particular internal team to another

EXEC AssignRequestToTeam 'Assigned to FDM', 'OBPHM'

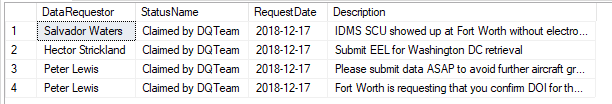
-- Querying View that retrieves requests with a pending status

Select \* From PendingRequests ORDER BY RequestDate



-- Querying View that retrieves requests assigned to my Team (Data Quality Team)

Select \* From ClaimedRequests



**GUI Prototype I chose to use R programming language**

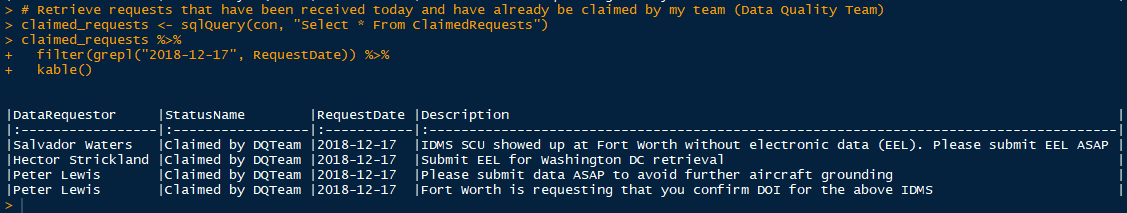
# Retrieve requests that have been received today and have already be claimed by my team (DQ Team)

claimed\_requests <- sqlQuery(con, "Select \* From ClaimedRequests")

claimed\_requests %>%

filter(grepl("2018-12-17", RequestDate)) %>%

kable()

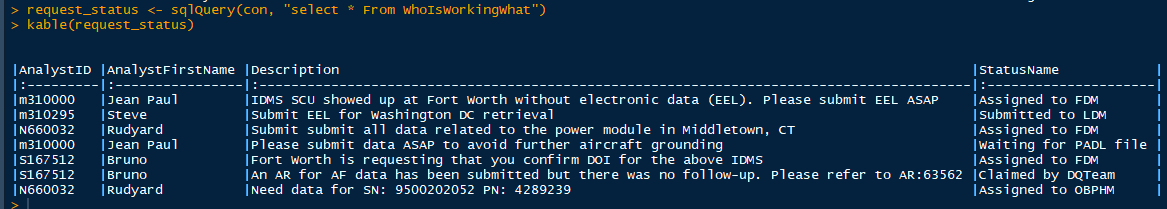
****

# Check data request status and which analyst is working it

sqlQuery(con,"select \* From WhoIsWorkingWhat")

request\_status <- sqlQuery(con, "select \* From WhoIsWorkingWhat")

kable(request\_status)



# My boss wants to see a report of who is working what, but doesn't have access to the database

# Extract data requests, status and who's working them, and export them to external Excel file

request\_status <- sqlQuery(con, "select \* From WhoIsWorkingWhat")

write.xlsx(request\_status, file = "request\_report.xlsx", col.names = T, sheetName = "Request Status")



# The exported file of the above code is the spreadsheet below

