



SQL-Mongo Project – Spatial Data of US Wildfires

BUAN 6320

Anu Jose
Shylaja Vijayaraghavan
Geetika Yalamanchili
Anne Emmanuel

Group #: 3

Activity	Member 1	Member 2	Member 3	Member 4
Prepared Data Model and Created Physical DB	x	x	x	x
Loaded Data into Database	x	x	x	x
Wrote SQL Queries	x	x	x	x
Prepared Mongo Database	x	x	x	x
Loaded data into Mongo DB	x	x	x	x
Wrote Mongo Queries	x	x	x	x
Prepared Report	x	x	x	x
Reviewed Report	x	x	x	x

Contents

Data Model	5
Assumptions/Notes About Data Entities and Relationships	5
Entity-Relationship Diagram	7
Physical Database	8
Assumptions/Notes About Data Set	6
Screen shot of Physical Database objects	8
Data in the Database	6
SQL Queries.....	18
Query 1	18
Question.....	18
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	18
Translation	18
Screen Shot of SQL Query and Results	18
Query 2	20
Question.....	20
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	20
Translation	20
Screen Shot of SQL Query and Results	20
Query 3	21
Question.....	21
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	21
Translation	21
Screen Shot of SQL Query and Results	21
Query 4	22
Question.....	22
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	22
Translation	22
Screen Shot of SQL Query and Results	22
Query 5	23
Question.....	23
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	23
Translation	23

Screen Shot of SQL Query and Results	23
Query 6	24
Question.....	24
Notes/Comments About SQL Query and Results (Include # of Rows in Result)	24
Translation	24
Screen Shot of SQL Query and Results	24
Data Review for MongoDB	25
Assumptions/Notes About Data Collections, Attributes and Relationships between Collections	13
Physical Mongo Database.....	25
Assumptions/Notes About Data Set.....	14
Screen shot of Physical Database objects (Database, Collections and Attributes)	25
Data in the Database	25
MongoDB Queries/Code.....	27
Query 1	27
Question.....	15
Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)...	27
Translation	27
Screen Shot of MongoDB Query/Code and Results	27
Query 2	28
Question.....	28
Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)...	28
Translation	28
Screen Shot of MongoDB Query/Code and Results	28
Query 3	29
Question.....	29
Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)...	29
Translation	29
Screen Shot of MongoDB Query/Code and Results	29
Query 4	30
Question.....	30
Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)...	30
Translation	30
Screen Shot of MongoDB Query/Code and Results	30

Query 5	31
Question.....	31
Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)...	31
Translation	31
Screen Shot of MongoDB Query/Code and Results	31
Query 6	32
Question.....	32
Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)...	32
Translation	32
Screen Shot of MongoDB Query/Code and Results	32

Data Model

Assumptions

No assumptions are made while making the model. All the records on the data set are kept intact. Some data has been mis-recorded which has affected the normalization.

Following fields were excluded since they presented Functional dependency in the tables:

- Cont_DOY (Calculated Field – Can be obtained from Contained Date)
- Discovery_DOY (Calculated Field – Can be obtained from Discovery Date)
- Fire_Year (Calculated Field – Can be obtained from Discovery Date)

Notes About Data Entities and Relationships

Following are the Table descriptions and relationships:

1. **Fires** – This is the main data table that includes the fire details. The Fire location, Fire Discovery details, Fire Contained details, Fire Size details, Fire Name, Statistical Cause of the fire, Local incident and report details, NWCG_Source details, ICS_209 and MTBS report details can be joined to this table to get all information about the fire
2. **Owner** – Describes the owner details of the location where fire occurred
3. **Location** – Describes the details of the location of fire including the latitude, longitude. State, Country and Owner details can be joined from the respective linked tables.
4. **State_Country** – The details of the state and country where the fire occurred can be obtained from this table
5. **Location_Complex** – The details of the location of fire including the complex name can be obtained here
6. **Fire_Size** – The size of the fire along with the classification of the fire size is included
7. **Stat_Cause** – The statistical cause of the fire is described in this table
8. **Source_System** – Includes the Source system from which the fire record is drawn and also identifies the type of source system (Fed, Non Fed and Interagency)
9. **Source_System_Reporting** – Details of the Source Reporting unit
10. **Source_NWCG_Link** – Linking table of the Source database of fire with NWCG reporting database of fire
11. **ICS_209** – Details of the incident from ICS_209 report
12. **MTBS** – Details of the incident from the MTBS perimeter dataset
13. **NWCG_Active_Units** – Details of the active NWCG units present in the database
14. **NWCG_Unit** – Details of the NWCG unit including the WildLand role of the unit in the fire community along with the details of the unit mapped from the NWCG_Unit_Detail table

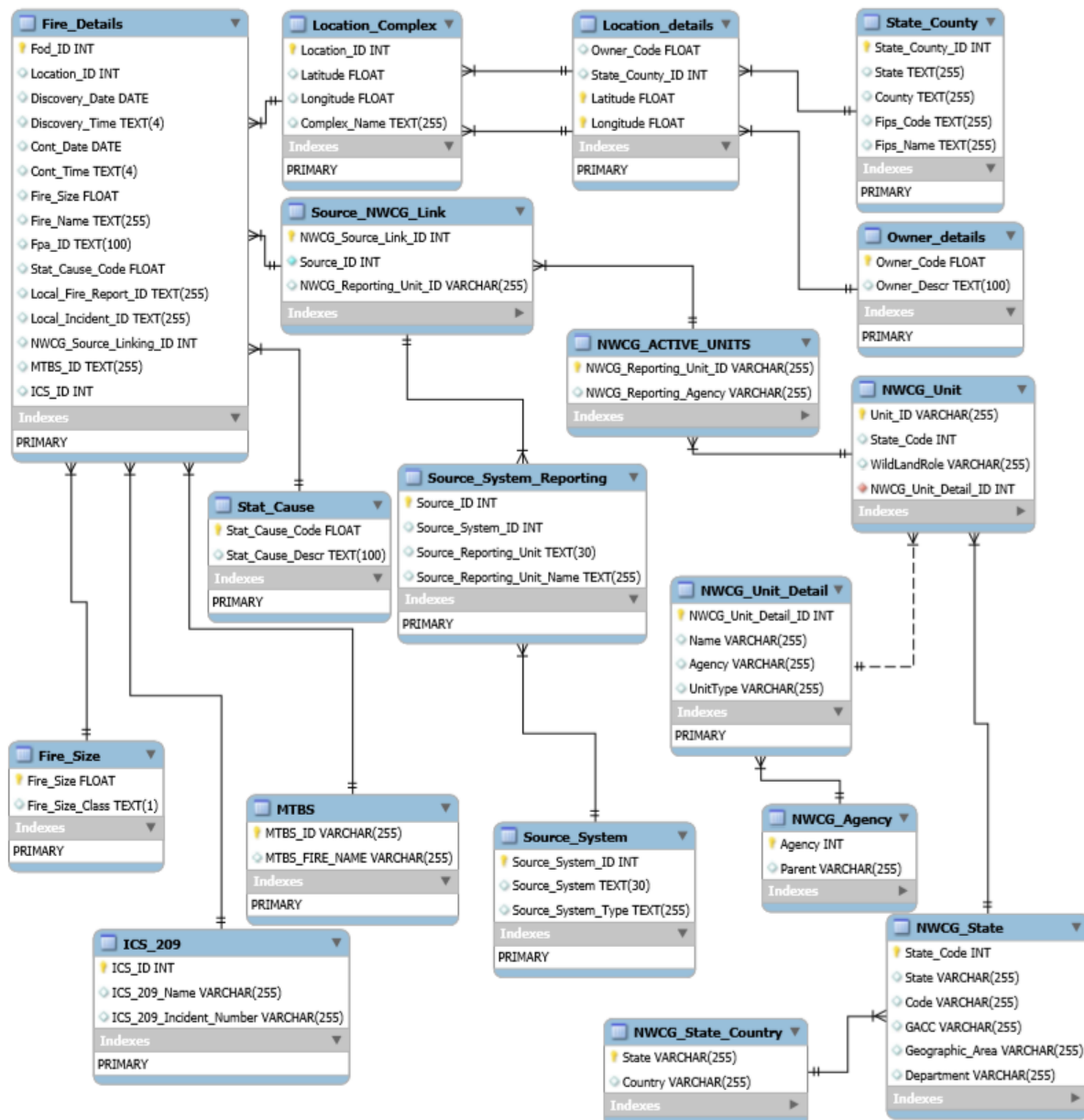
15. **NWCG_Unit_Detail** – Describes the name, agency and unit type of the NWCG_Unit
16. **NWCG_Agency** – Agency details and the Agency Sub group details are included in the table
17. **NWCG_State_Country** – State, Geographic are, GACC and department details are included
18. **NWCG_State** – NWCG unit location – State and country included in the table

Reasons for the Model to be in 3NF

The model presented in the 3NF form. It is said to be in 3NF since it satisfies the following conditions:

- Each table contains only a single value
- Each record is unique
- Primary key is defined for each table that is Unique and Not Null
- No transitive and functional dependencies

Entity-Relationship Diagram



Physical Database

Screen shot of Physical Database objects

1. Owner

```
1
2 • CREATE TABLE IF NOT EXISTS `Owner_details` (
3   `Owner_Code` FLOAT NOT NULL,
4   `Owner_Descr` TEXT(100) NULL,
5   PRIMARY KEY (`Owner_Code`))
6   ENGINE = InnoDB;
7
8 • insert into Owner_details
9   select distinct owner_code, owner_descr from fires_raw;
10
11 • select count(*) from Owner_details;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(*)
16

2. Fire_Size

```
1
2 • CREATE TABLE IF NOT EXISTS `Fire_Size` (
3   `Fire_Size` DOUBLE NOT NULL,
4   `Fire_Size_Class` TEXT(1) NULL,
5   PRIMARY KEY (`Fire_Size`))
6   ENGINE = InnoDB;
7
8 • insert into fire_size
9   select distinct fire_size, fire_size_class from fires_raw;
10
11 • select count(*) from fire_size;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(*)
13605

3. Stat_Cause

```
1
2 • CREATE TABLE IF NOT EXISTS `Stat_Cause` (
3   `Stat_Cause_Code` FLOAT NOT NULL,
4   `Stat_Cause_Descr` TEXT(100) NULL,
5   PRIMARY KEY (`Stat_Cause_Code`))
6   ENGINE = InnoDB;
7
8 • insert into stat_cause
9   select distinct stat_cause_code, stat_cause_descr from fires_raw;
10
11 • select count(*) from stat_cause;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

count(*)
13

4. State_County

```
1
2 • CREATE TABLE IF NOT EXISTS `State_County` (
3   `State_County_ID` INT AUTO_INCREMENT,
4   `State` TEXT(255) NULL,
5   `County` TEXT(255) NULL,
6   `Fips_Code` TEXT(255) NULL,
7   `Fips_Name` TEXT(255) NULL,
8   PRIMARY KEY (`State_County_ID`))
9   ENGINE = InnoDB;
10
11 • insert into State_County(State, County, Fips_Code, Fips_Name)
12   select distinct state, county, fips_code, fips_name from fires_raw;
13
14 • select count(*) from State_County;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

count(*)
4943

5. Source_System

```

1
2 • CREATE TABLE IF NOT EXISTS `Source_System` (
3     `Source_System_ID` INT Auto_increment,
4     `Source_System` TEXT(30) NULL,
5     `Source_System_Type` TEXT(255) NULL,
6     PRIMARY KEY (`Source_System_ID`))
7     ENGINE = InnoDB;
8
9 • insert into Source_System (Source_System, Source_System_Type)
10    select distinct source_system, source_system_type from fires_raw;
11
12 • select count(*) from Source_System;

```

Result Grid

count(*)
39

6. Source_System_Reporting

```

1
2 • CREATE TABLE IF NOT EXISTS `Source_System_Reporting` (
3     `Source_ID` INT Auto_increment,
4     `Source_System_ID` INT NULL,
5     `Source_Reporting_Unit` TEXT(30) NULL,
6     `Source_Reporting_Unit_Name` TEXT(255) NULL,
7     PRIMARY KEY (`Source_ID`),
8     CONSTRAINT `Source_System_ID`
9     FOREIGN KEY (`Source_System_ID`)
10    REFERENCES `Source_System` (`Source_System_ID`)
11    ON DELETE NO ACTION
12    ON UPDATE NO ACTION)
13    ENGINE = InnoDB;
14
15 • insert into source_system_reporting (Source_System_ID, Source_Reporting_Unit, Source_Reporting_Unit_Name )
16    select distinct source_system_id, source_reporting_unit, source_reporting_unit_name from fires_raw F
17    left join source_system S ON F.SOURCE_SYSTEM=S.SOURCE_SYSTEM AND F.SOURCE_SYSTEM_TYPE = S.SOURCE_SYSTEM_TYPE;
18
19 • select count(*) from source_system_reporting;

```

Result Grid

count(*)
6898

7. Source_NWCG_Link

```

1 CREATE TABLE IF NOT EXISTS `Source_NWCG_Link` (
2   `NWCG_Source_Link_ID` int Auto_Increment,
3   `Source_ID` INT NOT NULL,
4   `NU_ID` int,
5   PRIMARY KEY (`NWCG_Source_Link_ID`),
6   CONSTRAINT `NU_ID`
7     FOREIGN KEY (`NU_ID`)
8     REFERENCES `NWCG_Active_Units` (`NU_ID`)
9     ON DELETE NO ACTION
10    ON UPDATE NO ACTION);
11
12 insert into Source_NWCG_Link(Source_ID, NU_ID)
13 select distinct SOURCE_ID, nwcg.NU_ID from
14 (
15   SELECT DISTINCT SOURCE_ID, NWCG_REPORTING_UNIT_ID
16   FROM (select distinct SOURCE_REPORTING_UNIT, SOURCE_REPORTING_UNIT_NAME, NWCG_REPORTING_UNIT_ID from fires_raw ) F
17   LEFT JOIN SOURCE_SYSTEM_REPORTING S ON F.SOURCE_REPORTING_UNIT = S.SOURCE_REPORTING_UNIT
18   AND F.SOURCE_REPORTING_UNIT_NAME = S.SOURCE_REPORTING_UNIT_NAME
19 )a
20 left join NWCG_Active_Units nwcg on nwcg.NWCG_REPORTING_UNIT_ID=a.NWCG_REPORTING_UNIT_ID ;
21
22 select count(*) from Source_NWCG_Link;

```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content:

count(*)
1645

8. Location

```

1 CREATE TABLE IF NOT EXISTS `Location` (`Owner_Code` FLOAT NULL, `State_County_ID` INT NULL, `Latitude` DOUBLE NOT NULL,
2   `Longitude` DOUBLE NOT NULL,
3   PRIMARY KEY (`Latitude`, `Longitude`), CONSTRAINT `Owner_Code`
4     FOREIGN KEY (`Owner_Code`)
5     REFERENCES `Owner_details` (`Owner_Code`)
6     ON DELETE NO ACTION
7     ON UPDATE NO ACTION,
8   CONSTRAINT `State_County_ID`
9     FOREIGN KEY (`State_County_ID`)
10    REFERENCES `State_County` (`State_County_ID`)
11    ON DELETE NO ACTION
12    ON UPDATE NO ACTION)
13 ENGINE = InnoDB; INSERT INTO location SELECT DISTINCT od.OWNER_CODE, STATE_COUNTY_ID, F.latitude, F.longitude FROM T F
14 LEFT JOIN STATE_COUNTY S ON F.STATE=S.STATE AND F.COUNTY = S.COUNTY
15 LEFT JOIN owner_details od ON od.owner code = f.owner code AND od.owner descr=f.owner descr: SELECT count(*) from location

```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content:

count(*)
1594603

9. Location_Complex

```

1 ● CREATE TABLE IF NOT EXISTS `Location_Complex` (
2   `Location_ID` INT AUTO_INCREMENT, `Latitude` DOUBLE, `Longitude` DOUBLE,
3   `Complex_Name` TEXT(255) NULL,
4   PRIMARY KEY (`Location_ID`),
5   CONSTRAINT `Latitude`
6   FOREIGN KEY (`Latitude`)
7   REFERENCES `Location` (`Latitude`)
8   ON DELETE NO ACTION
9   ON UPDATE NO ACTION);
10 ● INSERT INTO location_complex
11   SELECT DISTINCT latitude, longitude, complex_name from
12   (SELECT DISTINCT LATITUDE, LONGITUDE, COMPLEX_NAME FROM fires_raw) F
13   LEFT JOIN location L ON F.LATITUDE=L.LATITUDE AND F.LONGITUDE = L.LONGITUDE;
14 ● SELECT count(*) from location_complex

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

count(*)
1569862

10. ICS_209

```

1
2 ● CREATE TABLE IF NOT EXISTS `ICS_209` (
3   `ICS_ID` INT auto_increment,
4   `ICS_209_Name` VARCHAR(255) NULL,
5   `ICS_209_Incident_Number` VARCHAR(255) NULL,
6   PRIMARY KEY (`ICS_ID`));
7
8 ● insert into ICS_209 (ICS_209_Name, ICS_209_Incident_Number)
9   SELECT DISTINCT ICS_209_INCIDENT_NUMBER, ICS_209_NAME FROM fires_raw;
10
11 ● select count(*) from ICS_209;

```

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

count(*)
23314

11. MTBS

The screenshot shows a SQL IDE window with a toolbar at the top. The SQL script is as follows:

```
1
2 • CREATE TABLE IF NOT EXISTS `MTBS` (
3   `MTBS_ID` varchar(255),
4   `MTBS_FIRE_NAME` VARCHAR(255) NULL,
5   PRIMARY KEY (`MTBS_ID`));
6
7 • insert into MTBS
8   SELECT DISTINCT MTBS_ID, MTBS_FIRE_NAME FROM fires_raw
9   where MTBS_ID is not null;
10
11 • select count(*) from MTBS;
```

Below the script, the 'Result Grid' is visible, showing the output of the final query:

count(*)
10481

12. NWCG_State_Country

The screenshot shows a SQL IDE window with a toolbar at the top. The SQL script is as follows:

```
1
2 • CREATE TABLE IF NOT EXISTS `NWCG_State_Country` (
3   `State` VARCHAR(255) NOT NULL,
4   `Country` VARCHAR(255) NULL,
5   PRIMARY KEY (`State`));
6
7
8 • insert into NWCG_State_Country (State, Country)
9   SELECT DISTINCT STATE, COUNTRY FROM project_wildfires.nwcg;
10
11 • select count(*) from NWCG_State_Country;
```

Below the script, the 'Result Grid' is visible, showing the output of the final query:

count(*)
68

13. NWCG_State

```

1
2 • CREATE TABLE IF NOT EXISTS `NWCG_State` (
3   `State_Code` Int Auto_Increment,
4   `State` VARCHAR(255) NULL,
5   `Code` VARCHAR(255) NULL,
6   `GACC` VARCHAR(255) NULL,
7   `Geographic_Area` VARCHAR(255) NULL,
8   `Department` VARCHAR(255) NULL,
9   PRIMARY KEY (`State_Code`),
10  CONSTRAINT `State`
11  FOREIGN KEY (`State`)
12  REFERENCES `NWCG_State_Country` (`State`)
13  ON DELETE NO ACTION
14  ON UPDATE NO ACTION);
15
16 • insert into NWCG_State (State, Code, GACC, Geographic_Area, Department )
17   select distinct b.State, Code, GACC, GeographicArea, Department from
18   ( select distinct State, Code, GACC, GeographicArea, Department, COUNTRY from project_wildfires.nwcg ) a
19   join NWCG_State_Country b on (a.state=b.state and a.country=b.country);
20
21 • select count(*) from NWCG_State;

```

Result Grid: count(*) 5867

14. NWCG_Agency

```

1
2 • CREATE TABLE IF NOT EXISTS `NWCG_Agency` (
3   `Agency` VARCHAR(255) NOT NULL,
4   `Parent` VARCHAR(255) NULL,
5   PRIMARY KEY (`Agency`));
6
7 • insert into NWCG_Agency
8   SELECT DISTINCT AGENCY, PARENT FROM project_wildfires.nwcg
9   where AGENCY is not null;
10
11 • select count(*) from NWCG_Agency;

```

Result Grid: count(*) 51

15. NWCG_Unit_Detail

```

2 • CREATE TABLE IF NOT EXISTS 'NWCG_Unit_Detail' (
3   'NWCG_Unit_Detail_ID' INT Auto_Increment,
4   'Name' VARCHAR(255) NULL,
5   'Agency' VARCHAR(255) NULL,
6   'UnitType' VARCHAR(255) NULL,
7   PRIMARY KEY ('NWCG_Unit_Detail_ID'),
8   CONSTRAINT 'Agency'
9     FOREIGN KEY ('Agency')
10    REFERENCES 'NWCG_Agency' ('Agency')
11    ON DELETE NO ACTION
12    ON UPDATE NO ACTION);
13
14 • CREATE TABLE A AS SELECT DISTINCT UNITID, NAME FROM project_wildfires.nwcfg;
15 • CREATE TABLE B AS SELECT DISTINCT NWCG_REPORTING_UNIT_ID , NWCG_REPORTING_UNIT_NAME FROM fires_raw;
16 • CREATE TABLE C AS SELECT * FROM A UNION SELECT * FROM B;
17
18 • insert into NWCG_Unit_Detail (Name, Agency, UnitType)
19   SELECT DISTINCT C.NAME, AGENCY, UNITTYPE FROM C
20   LEFT JOIN
21   project_wildfires.nwcfg N ON C.Name=N.NAME;
22
23 • select count(*) from NWCG_Unit_Detail;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ☐

count(*)
5802

16. NWCG_Unit

```

1 • CREATE TABLE IF NOT EXISTS 'NWCG_Unit' (
2   'NU_ID' int auto_increment,
3   'Unit_ID' varchar(255), 'State_Code' INT NULL,
4   'WildLandRole' VARCHAR(255) NULL,
5   'NWCG_Unit_Detail_ID' int,
6   PRIMARY KEY ('NU_ID'),
7   CONSTRAINT 'State_Code'
8     FOREIGN KEY ('State_Code')
9     REFERENCES 'NWCG_State' ('State_Code')
10    ON DELETE NO ACTION
11    ON UPDATE NO ACTION,
12   CONSTRAINT 'NWCG_Unit_Detail_ID'
13     FOREIGN KEY ('NWCG_Unit_Detail_ID')
14     REFERENCES 'NWCG_Unit_Detail' ('NWCG_Unit_Detail_ID')
15    ON DELETE NO ACTION
16    ON UPDATE NO ACTION);
17 • CREATE TABLE A AS SELECT DISTINCT UNITID, NAME FROM project_wildfires.nwcfg;
18 • CREATE TABLE B AS SELECT DISTINCT NWCG_REPORTING_UNIT_ID , NWCG_REPORTING_UNIT_NAME FROM fires_raw;
19 • CREATE TABLE C AS SELECT * FROM A UNION SELECT * FROM B;
20 • CREATE TABLE D AS SELECT DISTINCT C.UNITID, NWCG_UNIT_DETAIL_ID, WILDLANDROLE, STATE, CODE FROM C
21   LEFT JOIN project_wildfires.nwcfg N ON C.UNITID = N.UNITID
22   LEFT JOIN NWCG_UNIT_DETAIL NN ON C.NAME = NN.NAME;
23
24 • insert into NWCG_Unit (Unit_ID, State_Code, WildLandRole, NWCG_Unit_Detail_ID)
25   SELECT DISTINCT D.UNITID, STATE_CODE, WILDLANDROLE, NWCG_UNIT_DETAIL_ID FROM D
26   LEFT JOIN NWCG_STATE N ON D.STATE = N.STATE AND D.CODE = N.CODE;
27 • select count(*) from NWCG_Unit;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ☐

count(*)
6041

17. NWCG_Active_Unit


```

1 CREATE TABLE IF NOT EXISTS `NWCG_Active_Units` (
2   `NU_ID` int,
3   `NWCG_Reporting_Unit_ID` varchar(255),
4   `NWCG_Reporting_Agency` VARCHAR(255) NULL,
5   PRIMARY KEY (`NU_ID`),
6   CONSTRAINT `Added_NU_ID`
7     FOREIGN KEY (`NU_ID`)
8     REFERENCES `NWCG_Unit` (`NU_ID`)
9     ON DELETE NO ACTION
10    ON UPDATE NO ACTION);
1
2 insert into NWCG_Active_Units
3 select NU_ID, NWCG_REPORTING_UNIT_ID, b.NWCG_REPORTING_AGENCY from NWCG_Unit a
4 join (select distinct NWCG_REPORTING_UNIT_ID, NWCG_REPORTING_AGENCY FROM fires_raw)b
5 on a.unit_id=b.NWCG_REPORTING_UNIT_ID;
6
7 select count(*) from NWCG_Active_Units;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

count(*)
1645

18. Fires

```

1 Create table Fire as select distinct FOD_ID, FPA_ID, LOCATION_ID, DISCOVERY_DATE, DISCOVERY_TIME, CONT_DATE, CONT_TIME, FIRE_SIZE, FIRE_NAME,
2   STAT_CAUSE_CODE, MTBS_ID, ICS_ID, LOCAL_FIRE_REPORT_ID, LOCAL_INCIDENT_ID, NWCG_SOURCE_LINK_ID FROM FIRES_RAW f
3   LEFT JOIN LOCATION_COMPLEX LC ON f.LATITUDE=LC.LATITUDE AND f.LONGITUDE = LC.LONGITUDE AND f.COMPLEX_NAME = LC.COMPLEX_NAME
4   LEFT JOIN ICS_209 I ON f.ICS_209_NAME=I.ICS_209_NAME AND f.ICS_209_INCIDENT_NUMBER=I.ICS_209_INCIDENT_NUMBER
5   LEFT JOIN (select distinct nwcg_source_link_id, nwcg_reporting_unit_id, source_reporting_unit, source_reporting_unit_name, source_system,
6     source_system_type from source_nwcg_link a
7   LEFT JOIN (select distinct source_id, source_reporting_unit, source_reporting_unit_name, source_system, source_system_type from
8     source_system_reporting a LEFT JOIN source_sys_details b on a.source_system_ID = b.source_system_id) b
9     on a.source_id=b.source_id) b
10   on f.nwcg_reporting_unit_id = b.nwcg_reporting_unit_id and
11   f.source_reporting_unit = b.source_reporting_unit and
12   f.source_reporting_unit_name = b.source_reporting_unit_name and
13   f.source_system = b.source_system and
14   f.source_system_type = b.source_system_type;
15 select count(*) from Fire

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

count(*)
1880465

Data in Database

Table Name	Primary Key	Foreign Key	# of Rows in Table(mysql)
Fire Details	FOD_ID	Location_ID; Fire Size; Stat_Cause_Code; NWCG_Source_Linking_ID; MTBS_ID; ICS_ID	1880645
Owner	Owner Code	NA	16
Location	Latitude; Longitude	Owner_Code ; State_County_ID	15,94,382
State_County	State_County_ID	NA	4,943
Location_Complex	Location_ID	Latitude & Longitude	15,69,862
Fire_Size	Fire_Size	NA	13,605
Stat_Cause	Stat_Cause_ID	NA	13
Source_System	Source_System_ID	NA	39
Source_System_Reporting	Source_ID	Source_System_ID	6646
Source_NWCG_Link	NWCG_Source_Link_ID	Source_ID; NWCG_Reporting_Unit_ID	7135
ICS_209	ICS_ID	NA	23,314
MTBS	MTBS_ID	NA	10,482
NWCG_Unit	NWCG_Unit_ID	NWCG_Unit_Detail_ID; State_Code	6,039
NWCG_Unit_Detail	NWCG_Unit_Detail_ID	Agency	5,802
NWCG_State	State_Code	State	5,867
NWCG_Agency	Agency	NA	51
NWCG_State_Country	State	NA	68
NWCG_Active_Units	NWCG_Reporting_Unit_ID		1,640

SQL Queries

Assumptions:

1. "Source_Reporting_Unit_Name" is considered as forests with the assumption that each forest has one and only one agency to report the fires occurring the respective forest
2. "FOD_ID" is considered as the fires under the assumption that each fire is reported only by one reporting unit

Query 1

Question 1

A leading beverage company has announced a billion-dollar fund for removing debris from forests, rivers and mountains in the US. All states are interested. Which state has the best chance to win a share of the fund?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

The state with maximum number of fires is expected to have maximum debris and thereby have higher chances of receiving the funds for enabling them to ensure debris removal. The state with more debris must have more funds to clean up, hence better chance of the state to win the share. # of Rows in Result: 1

Translation

-- select state which has max no. of fires from fire table left joined with location complex table on location id of both fire and location complex tables and

left join on latitude and longitude from location and location complex table and left join on state county id from statecounty and location tables that is

grouped by state.

Clean-Up

-- cleanup: select state from max no. of fires from fire left join location_complex table on location id of fire and location complex,

left join on latitude and longitude from location and location complex and left join on state county id from state county and location, grouped by

state

(Screenshot next page)

Screen Shot of SQL Query and Results

```
1  #####Q1: A leading beverage company has announced a billion-dollar fund for removing debris from
2  #####forests, rivers and mountains in the US. All states are interested. Which state has the best
3  #####chance to win a share of the fund?
4  • select state from(
5    select max(a), state from
6    (select state, count(fod_id) as a from
7     fire f
8     left join location_complex lc on f.location_id =lc.location_id
9     left join location l on lc.latitude = l.latitude and lc.longitude = l.longitude
10    left join State_county s on l.state_county_id =s.state_county_id
11    group by state))
12
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

state
CA

Query 2

Question 2

One of the reporting agencies has suggested that children be banned from its forests unless there is one adult for every 3 children in a group visiting a forest. Name 3 forests where this would be the most appropriate.

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Georgia Forestry Commission, Red Lake Agency and New Jersey Forest Fire Service Division C are the three forests where the mentioned suggestion seems appropriate. These three forests have the maximum fires caused by the children. Since children have been the statistical cause for the fire, it is ideal to implement regulations on the visit of children into the forests under the supervision of the adults.

of Rows in Result: 3

Translation

-- select source reporting unit name as forest from fire table left joined with source nwcg link table on NWCG source link id from both the tables, left

join on source id from source nwcg link and source system reporting tables and left join on stat cause code from fire and stat cause details tables grouped by

source reporting unit name, ordered by count of fires(count of fod_id) and limit by 3.

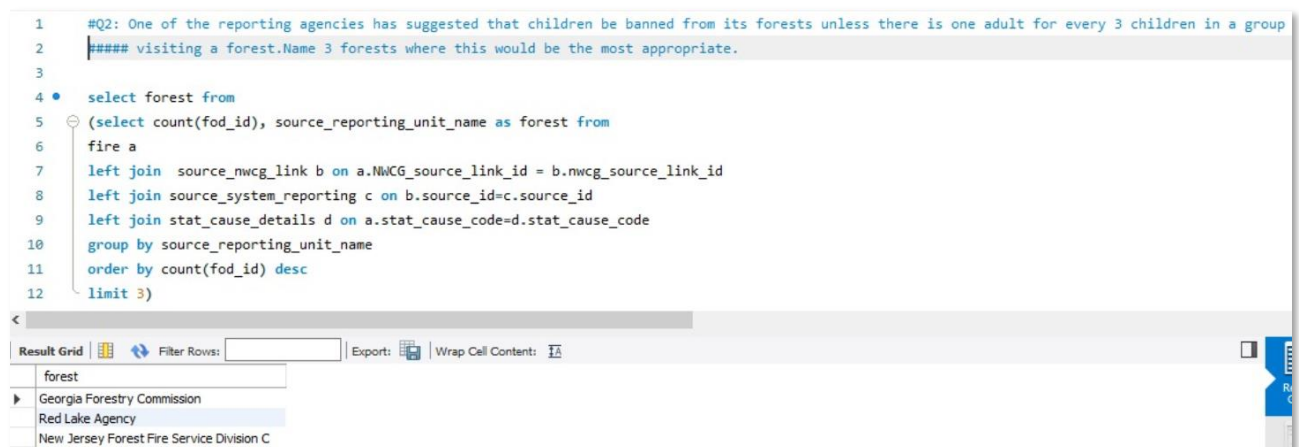
Clean-Up

-- cleanup: select source reporting unit name as forest from fire left join with source nwcg link on NWCG_source_link_id from fire and source nwcg link, left

join on source_id from source nwcg link and source system reporting, left join on stat_cause_code from fire and stat_cause_details, grouped by

source_reporting_unit_name, order by count of fod_id, limit by 3.

Screen Shot of SQL Query and Results



```
1  #Q2: One of the reporting agencies has suggested that children be banned from its forests unless there is one adult for every 3 children in a group
2  ##### visiting a forest.Name 3 forests where this would be the most appropriate.
3
4  • select forest from
5  (select count(fod_id), source_reporting_unit_name as forest from
6  fire a
7  left join source_nwcg_link b on a.NWCG_source_link_id = b.nwcg_source_link_id
8  left join source_system_reporting c on b.source_id=c.source_id
9  left join stat_cause_details d on a.stat_cause_code=d.stat_cause_code
10 group by source_reporting_unit_name
11 order by count(fod_id) desc
12 limit 3)
```

Result Grid

forest
Georgia Forestry Commission
Red Lake Agency
New Jersey Forest Fire Service Division C

Query 3

Question 3

One advocacy group says Nature and not human actions is to blame for most wildfires. Write a query that supports this statement.

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Fires Due to Natural Cause: 768996; Fires Due to Manmade Causes: 1111469. We do not have enough evidence to prove that nature and not human actions is to blame for most wildfires. The statistical cause of Nature caused fires were considered as Lightning, Miscellaneous and Missing/Unidentified while the rest causes were categorized as Human caused fires. Data clearly depicts that the statement cannot be supported unless any other Human caused fires is erroneously bucketed under Nature caused fires.

of Rows in Result: 1

Translation

-- select the count of fires caused by natural causes as count_natural from fire table left joined with stat cause details table on stat_cause_code

where stat_cause_code is either 1,9 or 13, select the count of fires caused by mannmade causes as count_manmade from fire table left joined with stat cause detals table on stat_cause_code where

stat_cause_code is neither 1,9 or 13 and display count_natural and count_manmade.

Clean-Up

-- cleanup: select count(FOD_ID) as count_natural from fire left join with stat_cause_details on stat_cause_code

where stat_cause_code is (1,9 or 13), select count(FOD_ID) as count_manmade from fire left join with stat_cause_details on stat_cause_code where

stat_cause_code is not(1,9 or 13), display count_natural and count_manmade.

Screen Shot of SQL Query and Results

```
1  #Q3: One advocacy group says Nature and not human actions is to blame for most wildfires. Write a query that supports this statement.
2  • select count_natural, count_manmade from
3  (
4  (select count(FOD_id) as count_natural from
5   fire f
6   left join stat_cause s on f.stat_cause_code = s.stat_cause_code
7   where f.stat_cause_code = 1 or f.stat_cause_code =9 or f.stat_cause_code = 13) ,
8  (select count(fod_id) as count_manmade from
9   fire f
10  left join stat_cause s on f.stat_cause_code = s.stat_cause_code
11  where f.stat_cause_code not in (1,9,13))
12 )
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

count_natural	count_manmade
768996	1111469

Query 4

Question 5

How many wildfires were reported by more than one unit/agency?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

0 records returned. Since FOD ID is assumed to be the wildfire, the FOD ID is unique across the database and hence there is no chance for the one fire to be reported by multiple unit/agency. Hence result is 0.

of Rows in Result: 0

Translation

-- select FOD_ID and count of distinct source_reporting_unit_name as no_of_units from fire table left joined with source nwcg link table on nwcg_sourcr_link_id

from both tables, left join on source_id from source nwcg link and source system reporting tables grouped by FOD_ID that have count of

distinct source_reporting_unit_name greater than 1

Clean-Up

-- cleanup: select FOD_ID and count(distinct source_reporting_unit_name) as no_of_units from fire left joined with source_nwcg_link on nwcg_sourcr_link_id

from fire and source_nwcg_link, left join on source_id from source nwcg link and source_system_reporting, group by FOD_ID, having count(distinct source_reporting_unit_name) >1

Screen Shot of SQL Query and Results

```
1 • select count(distinct source_reporting_unit_name) as no_of_units, fod_id as fires from
2 fire a
3 left join source_nwcg_link b on a.NWCG_source_link_id = b.nwcg_source_link_id
4 left join source_system_reporting c on b.source_id=c.source_id
5 group by fod_id
6 having count(distinct source_reporting_unit_name)>1
7
8
9
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

no_of_units	fires

Query 5

Question 6

What were the forests that had more than one fire that lasted more than two days?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

778 Forests had more than one fire that lasted for more than 2 days

Translation

-- select the no of rows from the result set obtained by selecting count of FOD_ID as no_of_fires and source_reporting_unit_name as forest

from another result set obtained by selecting FOD_ID, source_reporting_unit_name, difference between cont_date and discovery_date from fire table

left joined with source nwcg link on NWCG_source_link_id from both tables, left joined on source_id from source system reporting and

source nwcg link tables where difference between cont_date and discovery_date is greater than 2, grouped by source_reporting_unit_name

and having count of FOD_ID greater than 1

Clean-Up

-- cleanup: select the no of rows from the result set obtained by selecting count of FOD_ID as no_of_fires and source_reporting_unit_name as forest

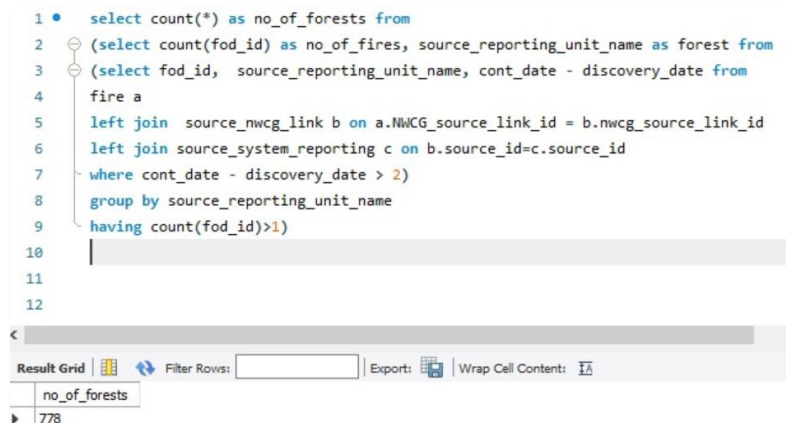
from another result set obtained by selecting FOD_ID, source_reporting_unit_name, cont_date - discovery_date from fire

left joined with source nwcg link on NWCG_source_link_id, left joined on source_id from source system reporting and

source_nwcg_link where (cont_date - discovery_date)> 2, group by source_reporting_unit_name

having count(FOD_ID)> 1

Screen Shot of SQL Query and Results



```
1 • select count(*) as no_of_forests from
2   (select count(fod_id) as no_of_fires, source_reporting_unit_name as forest from
3   (select fod_id, source_reporting_unit_name, cont_date - discovery_date from
4   fire a
5   left join source_nwcg_link b on a.NWCG_source_link_id = b.nwcg_source_link_id
6   left join source_system_reporting c on b.source_id=c.source_id
7   where cont_date - discovery_date > 2)
8   group by source_reporting_unit_name
9   having count(fod_id)>1)
10
11
12
```

The screenshot shows a SQL query editor with a query that counts the number of forests with more than one fire lasting more than two days. The query is executed, and the results are displayed in a table with one row: 778.

no_of_forests
778

Query 6

Question 8

Which forest had the most number of fires?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Georgia Forestry Commission has the maximum number of fires

Translation

--select the source_reporting_unit_name as forest and the corresponding count of fod_id as no_of_fires from fire table left joined with source nwcg link table on NWCG_source_link_id

left joined with source system reporting table on source_id, grouped by source reporting unit_name and ordered by the count of fod_id and limit by 1

Clean-Up

-- cleanup: select source_reporting_unit_name as forest and count(fod_id) as no_of_fires from fire left joined with source_nwcg_link on NWCG_source_link_id

left joined with source_system_reporting table on source_id, group by source_reporting_unit_name, order count(fod_id), limit by 1

Screen Shot of SQL Query and Results

The screenshot shows a SQL query editor with the following query:

```
1 • select count(fod_id) as no_of_fires, source_reporting_unit_name as forest from
2 fire a
3 left join source_nwcg_link b on a.NWCG_source_link_id = b.nwcg_source_link_id
4 left join source_system_reporting c on b.source_id=c.source_id
5 group by source_reporting_unit_name
6 order by count(fod_id) desc limit 1
7
8
9
```

Below the query editor is a results grid showing the following data:

no_of_fires	forest
97844	Georgia Forestry Commission

Data Review for MongoDB

Assumptions for Querying:

1. "Source_Reporting_Unit_Name" is considered as forests with the assumption that each forest has one and only one agency to report the fires occurring the respective forest
2. "FOD_ID" is considered as the fires under the assumption that each fire is reported only by one reporting unit

Notes About Data Collections

The un-normalized data of the Fires database and NWCG database is loaded.

Physical Mongo Database

Screen shot of Physical Database objects (Database, Collections and Attributes)

1. Fires Collection

DBProject.fires_updated

DOCUMENTS 1.9m TOTAL SIZE 1.7GB AVG. SIZE 957B INDEXES 1 TOTAL SIZE 17.9MB AVG. SIZE 17.9MB

Documents Aggregations Explain Plan Indexes

FILTER OPTIONS FIND RESET ...

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 20 of N/A

```
{
  "_id": ObjectId("5ded6d410edcb949c4a8ee14"),
  "OBJECTID": 1,
  "FOD_ID": 1,
  "FPA_ID": "FS-1418826",
  "SOURCE_SYSTEM_TYPE": "FED",
  "SOURCE_SYSTEM": "FS-FIRESTAT",
  "NWCG_REPORTING_AGENCY": "FS",
  "NWCG_REPORTING_UNIT_ID": "USCAPNF",
  "NWCG_REPORTING_UNIT_NAME": "Plumas National Forest",
  "SOURCE_REPORTING_UNIT": 511,
  "SOURCE_REPORTING_UNIT_NAME": "Plumas National Forest",
  "LOCAL_FIRE_REPORT_ID": 1,
  "LOCAL_INCIDENT_ID": "PNF-47",
  "FIRE_CODE": "BJ8K",
  "FIRE_NAME": "FOUNTAIN",
  "ICS_209_INCIDENT_NUMBER": "",
  "ICS_209_NAME": "",
  "MTBS_ID": "",
  "MTBS_FIRE_NAME": "",
  "COMPLEX_NAME": "",
  "FIRE_YEAR": 2005,
  "DISCOVERY_DATE": "2005-02-01",
  "DISCOVERY_DOY": 33,
  "DISCOVERY_TIME": 1300
}
```

```
> db.fires_updated.count()
1880466
>
```

Fires Database is uploaded with 1.88million documents

2. NWCG Collection

The screenshot shows the MongoDB DBProject.nwcg interface. At the top, it displays 'DOCUMENTS 5.9k' with a 'TOTAL SIZE' of '1.7MB' and 'AVG. SIZE' of '302B'. To the right, it shows 'INDEXES 1' with a 'TOTAL SIZE' of '80.0KB' and 'AVG. SIZE' of '80.0KB'. Below this, there are tabs for 'Documents', 'Aggregations', 'Explain Plan', and 'Indexes'. The 'Documents' tab is selected. A 'FILTER' button is on the left, and 'OPTIONS', 'FIND', 'RESET', and a menu icon are on the right. Below the tabs, there is an 'INSERT DOCUMENT' button and a 'VIEW' section with 'LIST' and 'TABLE' options. A status bar at the bottom right says 'Displaying documents 1 - 20 of 5868'. The main content area shows a JSON document with fields like '_id', 'OBJECTID', 'UnitId', 'GeographicArea', 'Gacc', 'WildlandRole', 'UnitType', 'Department', 'Agency', 'Parent', 'Country', 'State', 'Code', and 'Name'.

```
> db.nwcg.count()
5868
>
```

NWCG database is uploaded with 5868 documents

Data in the Database

Collection Name	Relationships With Other Collections (if any)	# of Documents in Collection
Fires_Updated	NA	1880466
NWCG	NA	5868

Comment: While uploading the databases, a null record was automatically uploaded for each of the databases. Hence the increment of the total documents by 1 for each.

MongoDB Queries/Code

Query 1

Question 1:

A leading beverage company has announced a billion-dollar fund for removing debris from forests, rivers and mountains in the US. All states are interested. Which state has the best chance to win a share of the fund?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Result: California is the state that has the best chance to win a share of the fund.

Result Translation: The result depicts the state with maximum fires which would ideally have the maximum debris and hence they should stand the best chance to win the fund (to have the positive impact on the state through cleaning up utilizing these funds)

Translation - Query

- DBProject Database selected to Db variable
- Fires Collection used
- Aggregate Pipeline
- Group the fires using "State" using \$group
- Order the result using sort in descending order using \$sort
- Display the State with maximum fires using \$project and limit the result to 1 using \$limit

Screen Shot of MongoDB Query/Code and Results

```
> db.Fires.aggregate([
...   {$group: {_id: "$STATE", count: {$sum: 1}}},
...   {$sort: {count: -1}},
...   {$limit: 1},
...   {$project: {state: 1}}
... ]);
{ "_id" : "CA" }
```

Query 2

Question 2:

One of the reporting agencies has suggested that children be banned from its forests unless there is one adult for every 3 children in a group visiting a forest. Name 3 forests where this would be the most appropriate.

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Result: Georgia Forestry Commission, Red Lake Agency and New Jersey Forest Fire Service Division C are the three forests where the mentioned suggestion seems appropriate.

Result Translation: These three forests have the maximum fires caused by the children. Since children have been the statistical cause for the fire, it is ideal to implement regulations on the visit of children into the forests under the supervision of the adults.

Translation - Query

- DBProject Database selected to Db variable
- Fires Collection used
- Aggeregate Pipeline
- Filter Statistical Cause as Children
- Group the Source Reporting Unit Name of the resultant data using \$group
- Order the result using sort in descending order using \$sort
- Display the Source Reporting Unit Name using \$project and limit the result to 1 using \$limit

Screen Shot of MongoDB Query/Code and Results

```
> db.Fires.aggregate([
...   {$match:{ "STAT_CAUSE_DESCR":"Children"}},
...   {$group:{_id:"$SOURCE_REPORTING_UNIT_NAME", count:{$sum:1}}},
...   {$sort:{count:-1}},
...   {$limit:3},
...   {$project:{SOURCE_REPORTING_UNIT_NAME:1}}
... ]));
{ "_id" : "Georgia Forestry Commission" }
{ "_id" : "Red Lake Agency" }
{ "_id" : "New Jersey Forest Fire Service Division C" }
>
```

Query 3

Question 3:

One advocacy group says Nature and not human actions is to blame for most wildfires. Write a query that supports this statement.

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Result: Fires Due to Natural Cause: 768996; Percent of Nature caused Fires: 40.89%. We do not have enough evidence to prove that nature and not human actions is to blame for most wildfires.

Result Translation: The statistical cause of Nature caused fires were considered as Lightening, Miscellaneous and Missing/Unidentified while the rest causes were categorized as Human caused fires. Data clearly depicts that the statement cannot be supported unless any other Human caused fires is erroneously bucketed under Nature caused fires.

Translation

- DBProject Database selected to Db variable
- Fires Collection used
- Aggeregated Pipeline
- Statistical Code – 1,9,13 which correspond to “Lightening”, “Miscellaneous” & “Missing/Unidentified” are filtered using \$match
- Group the resultant data using \$group
- Display the resultant data using \$project along with the calculated field - Percentage of the Nature caused fires

Screen Shot of MongoDB Query/Code and Results

```
> db.Fires.aggregate([
...   {$match : {"STAT_CAUSE_CODE" : {$in: [1,9,13]}}},
...   {$group: {_id:"Count of Fires Due to Natural Cause", count:{$sum:1}}},
...   {$project: {_id:1, count:1, PercentNatural : { $divide: ["$count",1880465]}}}
... ]);
{ "_id" : "Count of Fires Due to Natural Cause", "count" : 768996, "PercentNatural" : 0.4089392783167993 }
>
```

Query 4

Question 5

How many wildfires were reported by more than one unit/agency?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Result: 0

Result Translation: Since FOD ID is assumed to be the wildfire, the FOD ID is unique across the database and hence there is no chance for the one fire to be reported by multiple unit/agency. Hence result is 0.

Translation

- DBProject Database selected to Db variable
- Fires Collection used
- Aggeregate Pipeline
- Group the data by FOD_ID and sum the Source Reporting Unit Name using \$group
- Filter the Source Reporting unit greater than 1 using \$match
- The length of the resultant data is returned

Screen Shot of MongoDB Query/Code and Results

```
> db.Fires.aggregate([
... {$group:{_id:"$FOD_ID",No_of_units_reported: {$sum:"$SOURCE_REPORTING_UNIT_NAME"}}},
... {$match:{No_of_units_reported:{$gt:1}}},
... ],{allowDiskUse: true}).toArray().length
0
```

Query 5

Question 6

What were the forests that had more than one fire that lasted more than two days?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

778 Forests had more than one fire that lasted for more than 2 days

Translation

- DBProject Database selected to Db variable
- Fires Collection used
- Aggeregate Pipeline
- Calculated Field FireDays created using \$addFields (Subtract Discovery Date from Controlled Date)
- Filter the data set with FireDays > 2 using \$match
- Group the data by fires on Source Reporting Unit Name using \$group
- The length of the resultant data is returned

Screen Shot of MongoDB Query/Code and Results

```
> db.Fires.aggregate([
...   {$addFields: {FireDays : {$subtract: ["$CONT_DATE", "$DISCOVERY_DATE"]}}},
...   {$match : {"FireDays" : {$gt:2}}},
...   {$group: {_id:"SOURCE_REPORTING_UNIT_NAME", count:{$sum:1}}},
...   {$project: {_id:1, count:1}}
... ]);
```


Query 6

Question 8:

Which forest had the most number of fires?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Result: Georgia Forestry Commission has the maximum number of fires

Translation

- DBProject Database selected to Db variable
- Fires Collection used
- Aggeregate Pipeline
- Group the resultant data on Source Reporting Unit Name using \$group
- Order the result data using \$sort in the descending order
- Display the resultant data – “Source Reporting Unit Name” field using \$project limiting the result table to 1 record using \$limit

Screen Shot of MongoDB Query/Code and Results

```
> db.Fires.aggregate([
...   {$group: {_id: "$SOURCE_REPORTING_UNIT_NAME", count: {$sum: 1}}},
...   {$sort: {count: -1}},
...   {$limit: 1},
...   {$project: {SOURCE_REPORTING_UNIT_NAME: 1, count: 1}}
... ]);
{ "_id" : "Georgia Forestry Commission", "count" : 97844 }
>
```