Visualization: Num of Tankers per Contractor

```
SELECT
  COUNT(contractor),
  contractor
FROM
  dbSeniorDesign.FirePlane
GROUP BY
  contractor
```

Visualization: Hours Spent Grounded vs Airborne per Aircraft

```
SELECT
    tail_no,
    AVG(CASE WHEN year = 2020 THEN diff_over_avg ELSE NULL END) AS '2020',
    AVG(CASE WHEN year = 2021 THEN diff_over_avg ELSE NULL END) AS '2021',
    AVG(CASE WHEN year = 2022 THEN diff_over_avg ELSE NULL END) AS '2022'
FROM
    dbSeniorDesign.view_plane_grounded
WHERE
    year IN (2020, 2021, 2022)
GROUP BY
    tail_no;
```

Visualization: Most Visited Tanker Bases

```
SELECT tb.base_name, COUNT(*) AS visits_count
FROM dbSeniorDesign.TankerBaseVisits tbv
JOIN dbSeniorDesign.TankerBase tb ON tbv.base_code = tb.base_code
GROUP BY tb.base_name
ORDER BY visits_count DESC;
```

Visualization: # of Fires > 500 Acres that Received Drops from Tankers

```
SELECT COUNT(s.has_plane_drop) AS 'Received Drops'
FROM (SELECT f.fire_id, s.has_plane_drop
      FROM dbSeniorDesign.SuppressionStats s JOIN dbSeniorDesign.FirePoint
f ON f.fire_id=s.fire_id
      WHERE f.final_acres > 500 AND s.has_plane_drop = 1
     ) s
GROUP BY s.has_plane_drop
```

```
SELECT COUNT(s.has_plane_drop) AS 'Did Not Receive Drops'
FROM (SELECT f.fire_id, s.has_plane_drop
      FROM dbSeniorDesign.SuppressionStats s JOIN dbSeniorDesign.FirePoint
f ON f.fire_id=s.fire_id
      WHERE f.final_acres > 500 AND s.has_plane_drop = 0
     ) s
GROUP BY s.has_plane_drop
```

Visualization: Fire Heatmap

```
SELECT
  longitude,
  latitude,
  base_name
FROM
  dbSeniorDesign.TankerBase
SELECT
  latitude,
  longitude,
  create_date
FROM
  dbSeniorDesign.FirePoint
WHERE
  create_date BETWEEN '2021-05-01 00:00:00.000'
  AND '2021-08-26 00:00:00.000'
```

Visualization: Acres Burned Gauges / Pie Chart

```
SELECT COUNT(final_acres) AS '< 1 Acre' FROM dbSeniorDesign.FirePoint
WHERE final_acres < 1
SELECT COUNT(final_acres) AS '1 - 25 Acres' FROM dbSeniorDesign.FirePoint
WHERE final_acres BETWEEN 1 AND 25
```

```
SELECT COUNT(final_acres) AS '26 - 100 Acres' FROM
dbSeniorDesign.FirePoint WHERE final_acres BETWEEN 26 AND 100
SELECT COUNT(final_acres) AS '101 - 1,000 Acres' FROM
dbSeniorDesign.FirePoint WHERE final_acres BETWEEN 101 AND 1000
SELECT COUNT(final_acres) AS '1,001 - 50,000 Acres' FROM
dbSeniorDesign.FirePoint WHERE final_acres BETWEEN 1001 AND 50000
SELECT COUNT(final_acres) AS '> 50,000 Acres' FROM
dbSeniorDesign.FirePoint WHERE final_acres > 50000
```

Visualization: Average Monthly Flight Hours by Air Tanker

```
SELECT monthly_avg, month, tail_no
FROM dbSeniorDesign.view_PlaneHours
GROUP BY month, tail_no
ORDER BY
 CASE
   WHEN month = 'January' THEN 1
   WHEN month = 'February' THEN 2
   WHEN month = 'March' THEN 3
   WHEN month = 'April' THEN 4
   WHEN month = 'May' THEN 5
   WHEN month = 'June' THEN 6
   WHEN month = 'July' THEN 7
   WHEN month = 'August' THEN 8
   WHEN month = 'September' THEN 9
   WHEN month = 'October' THEN 10
   WHEN month = 'November' THEN 11
   WHEN month = 'December' THEN 12
 END,
 CASE tail_no
   WHEN 'N131CG' THEN 1
   WHEN 'N137CG' THEN 2
   WHEN 'N17085' THEN 3
   WHEN 'N292EA' THEN 4
   WHEN 'N295EA' THEN 5
```

```
  WHEN 'N297EA' THEN 6
  WHEN 'N325AC' THEN 7
  WHEN 'N385AC' THEN 8
  WHEN 'N416AC' THEN 9
  WHEN 'N417BT' THEN 10
  WHEN 'N470NA' THEN 11
  WHEN 'N471NA' THEN 12
  WHEN 'N474NA' THEN 13
  WHEN 'N478NA' THEN 14
  WHEN 'N522AX' THEN 15
  WHEN 'N603AX' THEN 16
  WHEN 'N612AX' THEN 17
  WHEN 'N635AC' THEN 18
  WHEN 'N839AC' THEN 19
  WHEN 'N291EA' THEN 20
  WHEN 'N392AC' THEN 21
  WHEN 'N475NA' THEN 22
  WHEN 'N477NA' THEN 23
  WHEN 'N355AC' THEN 24
  WHEN 'N374AC' THEN 25
  WHEN 'N418BT' THEN 26
END;
```

Note: Cases are necessary so that the order of the output is predictable for color coding and axis labeling purposes.

Visualization: Average Proximity of Fires Closest to a Given Tanker Base

```
SELECT avg_proximity, base_code
FROM dbSeniorDesign.BaseFireProximity
ORDER BY avg_proximity DESC;
```

Visualization: Suppression Result by Response Time

```
SELECT response_time, suppression_result
FROM dbSeniorDesign.view_ResponseTime
JOIN dbSeniorDesign.SuppressionStats ON
(dbSeniorDesign.view_ResponseTime.fire_id =
dbSeniorDesign.SuppressionStats.fire_id)
WHERE response_time < 500;
```

API Endpoints for Google Maps Rendering:

Fire Points: `SELECT * FROM FirePoint WHERE discovery_date BETWEEN (?) AND (?)`

- Gather all fires that follow between two dates

Tanker Bases: `SELECT base_code, base_name, airport, region, elevation, latitude, longitude FROM TankerBase`

- Get all tanker base information from TankerBase table

Flights:

1. Flight Information: `SELECT DISTINCT flight_id FROM FlightInfo WHERE DATE(landing) >= DATE(?) AND DATE(takeoff) <= DATE(?)`
    a. Gather all Flights that follow the conditions below:
        i. Start Date must be before the landing DateTime
        ii. End Date must be after the takeoff DateTime
2. Flight Paths: `SELECT * FROM Flight JOIN FlightInfo USING(flight_id) WHERE flight_id = ? AND flight_timestamp <= TIMESTAMP(?, ?) ORDER BY flight_timestamp DESC`
    a. Gather all FlightPaths and FlightInfos for all Flights that follow the conditions below:
        i. Flight_id must match for each point of the aircraft.
        ii. The timestamp must be less than the endDate to allow planes be in midair from endDate.
        iii. Order these by descending so that the end of the flight is first in the array.
3. Plane Colors/Tail Numbers: `SELECT DISTINCT tail_no FROM Flight`
    a. Get all the tail_no's found in the Flight table, ensuring that there are no duplicate entry's.

Other Queries:

1. Tanker Data: `SELECT * FROM AirTanker JOIN FirePlane USING (tanker_type)`
    a. Gather all FirePlanes while getter what kind of tanker they are:
        i. Tanker_type: The Type of Aircraft is
            1. B737
            2. BAe-146
            3. C130Q
            4. CL-215

            5. DC-10
            6. MD-87
            7. RJ85
        ii. Tank_size: How large their fire retardant bay is
        iii. Tail_no: The Tail Number associated with this aircraft
        iv. Contractor: Who owns this plane

2. Fire Containment Data: `SELECT * FROM FirePoint WHERE YEAR(create_date) = \'2020\' and MONTH(create_date) = \'1\' and containment_date IS NOT NULL and fireout_date IS NOT NULL and discovery_acres IS NOT NULL and discovery_acres > 0.1 and incident_size_acres IS NOT NULL and incident_size_acres > 0.1`
    a. Find FirePoints in 2020 in January where discovery_acres and incident_size_acres are > 0.1. A fireout_date must exist.
    b. It would be best to not use create_date and instead use discovery_date as it is a non-null attribute.