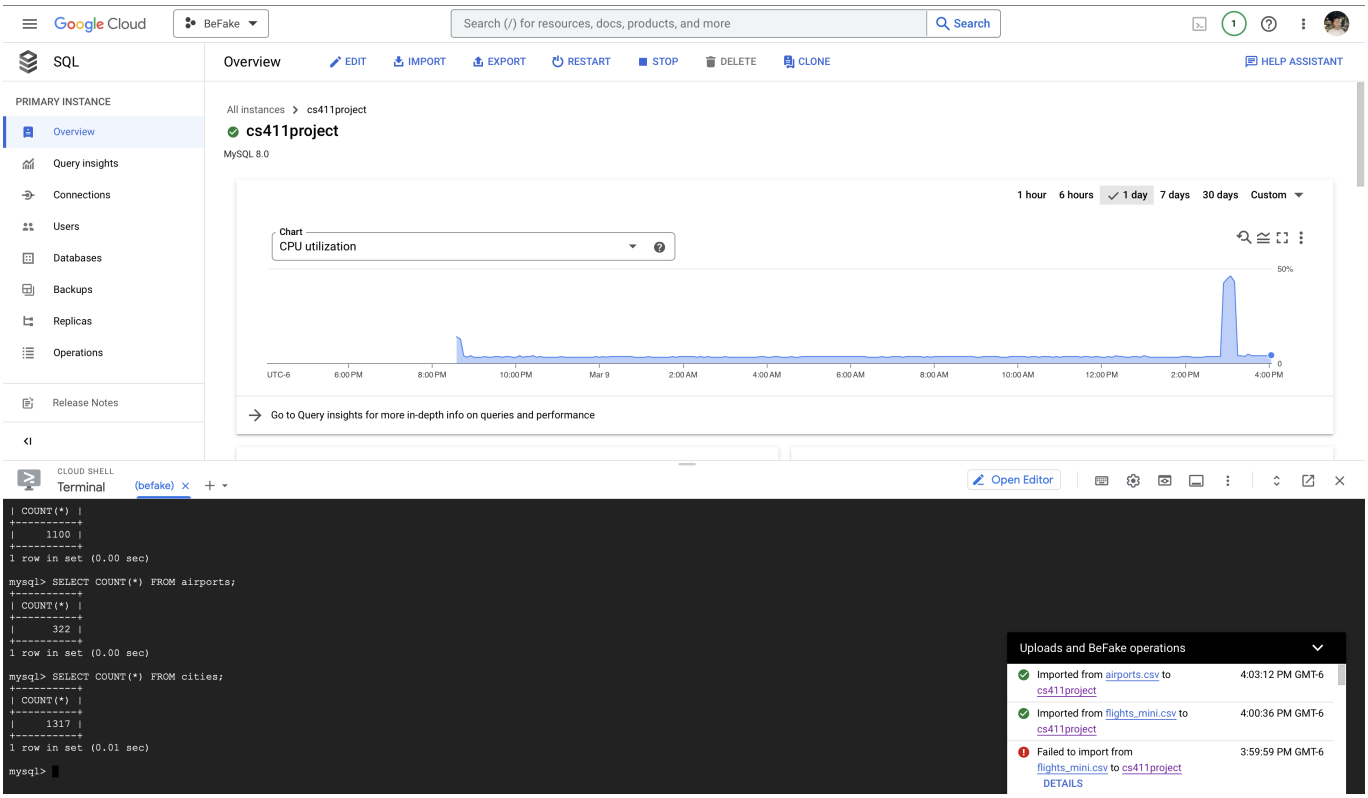
Team YAAG Database Design

GCP Set Up



DDL Commands

Airlines Table

```
CREATE TABLE airlines
(
    iata_code VARCHAR(255) NOT NULL,
    airline VARCHAR(255) NOT NULL,
    PRIMARY KEY (iata_code),
    UNIQUE (airline)
);
```

Cities Table

```
CREATE TABLE cities
(
    city VARCHAR(255),
    state VARCHAR(255),
    PRIMARY KEY (city, state)
);
```

Airports Table

```
CREATE TABLE airports
(
    iata_code VARCHAR(255) NOT NULL,
    city VARCHAR(255) NOT NULL,
    state VARCHAR(255) NOT NULL,
    airport VARCHAR(255) NOT NULL,
    latitude DECIMAL(40, 20),
    longitude DECIMAL(40, 20),
    PRIMARY KEY (iata_code)
);
```

Flights Table

```
/*
delay reason is an int where:
1: air_system_delay
2: security_delay
3: airline_delay
4: late_aircraft_delay
5: weather_delay
/*

CREATE TABLE flights
(
    flightid INT unsigned NOT NULL,
    airline VARCHAR(255) NOT NULL,
    tailnumber VARCHAR(255),
    origin VARCHAR(255) NOT NULL,
    destination VARCHAR(255) NOT NULL,
    month INT NOT NULL,
    day INT NOT NULL,
    year INT NOT NULL,
    flightnumber INT NOT NULL,
    departdelay INT,
    arrivaldelay INT,
    cancelled INT,
    cancel_reason VARCHAR(1),
    delay_reason INT,
    PRIMARY KEY (flightid)
);
```

Data Insertion

Airlines Table

```
mysql> SELECT COUNT(*) FROM airlines;
+----------+
| COUNT(*) |
+----------+
|     1013 |
+----------+
1 row in set (0.01 sec)
```

Flights Table

```
mysql> SELECT COUNT(*) FROM flights;
+----------+
| COUNT(*) |
+----------+
|     1100 |
+----------+
1 row in set (0.00 sec)
```

Airports Table

```
mysql> SELECT COUNT(*) FROM airports;
+----------+
| COUNT(*) |
+----------+
|      322 |
+----------+
1 row in set (0.00 sec)
```

Cities Table

```
mysql> SELECT COUNT(*) FROM cities;
+-----------+
| COUNT(*)  |
+-----------+
|      1317 |
+-----------+
1 row in set (0.01 sec)
```

Queries

First Advanced Query

```
SELECT a.iata_code, a.airport, AVG(arrivaldelay) as arrivaldelaymins
FROM airports a JOIN flights f ON (a.iata_code = f.origin)
WHERE a.iata_code IN (SELECT iata_code FROM airports a1 JOIN flights f1 ON
(a1.iata_code = f1.origin) GROUP BY a1.iata_code HAVING AVG(arrivaldelay)
> AVG(departdelay))
GROUP BY a.iata_code
ORDER BY AVG(arrivaldelay) DESC
LIMIT 15;
```

```
mysql> SELECT a.iata_code, a.airport, AVG(arrivaldelay) as arrivaldelaymins
    -> FROM airports a JOIN flights f ON (a.iata_code = f.origin)
    -> WHERE a.iata_code IN (SELECT iata_code FROM airports a1 JOIN flights f1 ON (a1.iata_code = f1.origin) GROUP BY a1.iata_code HAVING AVG(arrivaldelay) > AVG(departdelay))
    -> GROUP BY a.iata_code
    -> ORDER BY AVG(arrivaldelay) DESC
    -> LIMIT 15;
+-----------+-----------------------------------------+------------------+
| iata_code | airport                                 | arrivaldelaymins |
+-----------+-----------------------------------------+------------------+
| LAW       | Lawton-Fort Sill Regional Airport       |         376.0000 |
| CLD       | McClellan-Palomar Airport               |          88.0000 |
| BUR       | Bob Hope Airport (Hollywood Burbank Airport) |          59.5000 |
| GUM       | Guam International Airport              |          56.0000 |
| DRO       | Durango-La Plata County Airport         |          56.0000 |
| LIH       | Lihue Airport                           |          51.0000 |
| SJU       | Luis Muñoz Marín International Airport  |          36.0000 |
| ABQ       | Albuquerque International Sunport       |          20.0000 |
| ERI       | Erie International Airport              |          19.0000 |
| SMF       | Sacramento International Airport        |          18.0000 |
| BLI       | Bellingham International Airport        |          14.0000 |
| RDU       | Raleigh-Durham International Airport    |          13.7143 |
| FAT       | Fresno Yosemite International Airport   |          12.8750 |
| GUC       | Gunnison-Crested Butte Regional Airport |          12.0000 |
| OKC       | Will Rogers World Airport               |          12.0000 |
+-----------+-----------------------------------------+------------------+
15 rows in set (0.01 sec)
```

Second Advanced Query

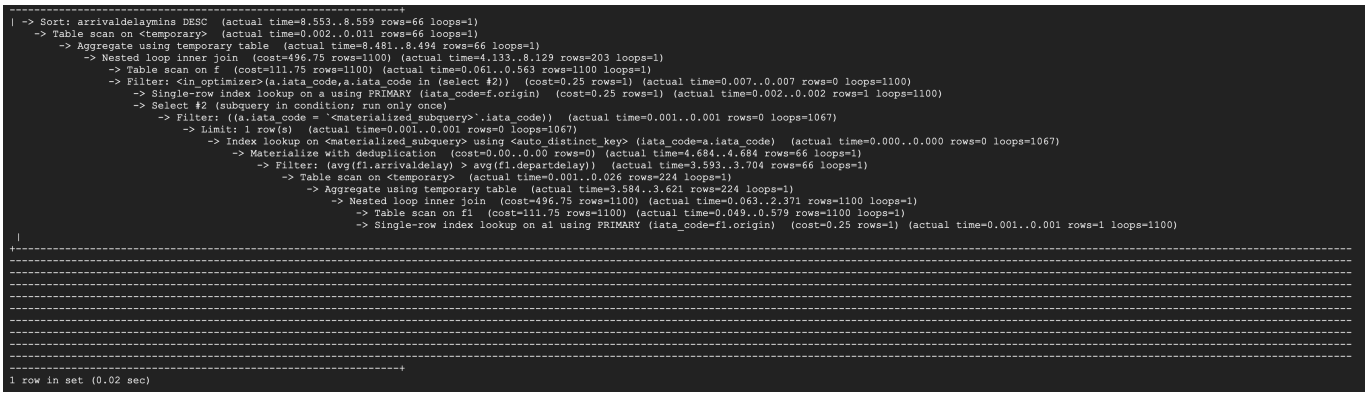
```
SELECT a.iata_code, a.airport, COUNT(f.destination) as countdest
FROM airports a JOIN flights f ON (a.iata_code = f.origin)
GROUP BY a.iata_code
ORDER BY countdest DESC
LIMIT 15;
```

```
mysql> SELECT a.iata_code, a.airport, COUNT(f.destination) as countdest
    -> FROM airports a JOIN flights f ON (a.iata_code = f.origin)
    -> GROUP BY a.iata_code
    -> ORDER BY countdest DESC
    -> LIMIT 15;
+-----------+-----------------------------------------------------------------------+-----------+
| iata_code | airport                                                               | countdest |
+-----------+-----------------------------------------------------------------------+-----------+
| BOS       | Gen. Edward Lawrence Logan International Airport                      |        38 |
| SEA       | Seattle-Tacoma International Airport                                  |        37 |
| LAX       | Los Angeles International Airport                                     |        35 |
| SFO       | San Francisco International Airport                                   |        32 |
| DFW       | Dallas/Fort Worth International Airport                               |        30 |
| LAS       | McCarran International Airport                                        |        29 |
| JFK       | John F. Kennedy International Airport (New York International Airport) |        27 |
| PHX       | Phoenix Sky Harbor International Airport                              |        25 |
| MCO       | Orlando International Airport                                         |        25 |
| MIA       | Miami International Airport                                           |        22 |
| PDX       | Portland International Airport                                        |        22 |
| SAN       | San Diego International Airport (Lindbergh Field)                     |        21 |
| FLL       | Fort Lauderdale-Hollywood International Airport                       |        19 |
| EWR       | Newark Liberty International Airport                                  |        18 |
| ORD       | Chicago O'Hare International Airport                                  |        18 |
+-----------+-----------------------------------------------------------------------+-----------+
15 rows in set (0.01 sec)
```

Indexing

Query 1

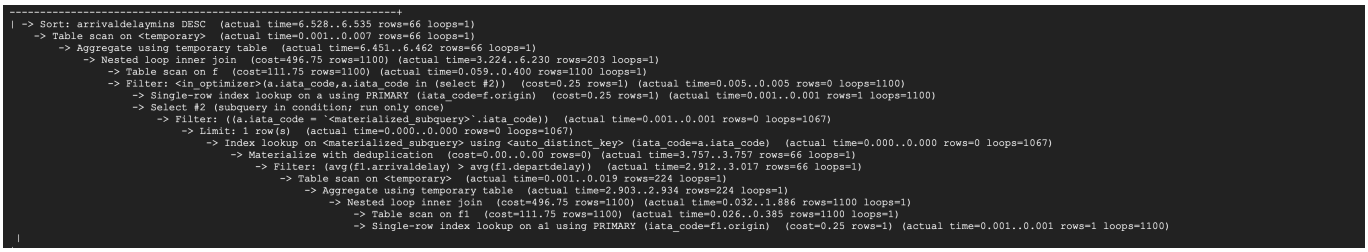
```
SELECT a.iata_code, a.airport, AVG(arrivaldelay) as arrivaldelaymins
FROM airports a JOIN flights f ON (a.iata_code = f.origin)
WHERE a.iata_code IN (SELECT iata_code FROM airports a1 JOIN flights f1 ON
(a1.iata_code = f1.origin) GROUP BY a1.iata_code HAVING AVG(arrivaldelay)
> AVG(departdelay))
GROUP BY a.iata_code
ORDER BY AVG(arrivaldelay) DESC;
```



Indexed Arrival Delay

Created index for arrival delay because the sort by arrivaldelay took the longest time at 8.553 units of time.

```
mysql> CREATE INDEX idx_arrivaldelay ON flights(arrivaldelay);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



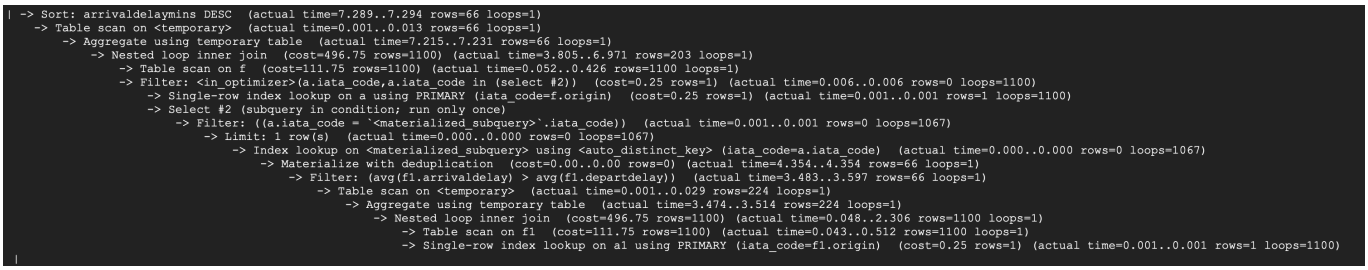
```
-> Sort: arrivaldelaymins DESC  (actual time=6.528..6.535 rows=66 loops=1)
    -> Table scan on <temporary>  (actual time=0.001..0.007 rows=66 loops=1)
        -> Aggregate using temporary table  (actual time=6.451..6.462 rows=66 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=3.224..6.230 rows=203 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.059..0.400 rows=1100 loops=1)
                -> Filter: <in_optimizer>(a.iata_code,a.iata_code in (select #2))  (cost=0.25 rows=1) (actual time=0.005..0.005 rows=0 loops=1100)
                    -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
                    -> Select #2 (subquery in condition; run only once)
                        -> Filter: ((a.iata_code = `<materialized_subquery>`.iata_code))  (actual time=0.001..0.001 rows=0 loops=1067)
                            -> Limit: 1 row(s)  (actual time=0.000..0.000 rows=0 loops=1067)
                                -> Index lookup on <materialized_subquery> using <auto_distinct_key> (iata_code=a.iata_code)  (actual time=0.000..0.000 rows=0 loops=1067)
                                    -> Materialize with deduplication  (cost=0.00..0.00 rows=0) (actual time=3.757..3.757 rows=66 loops=1)
                                        -> Filter: (avg(f1.arrivaldelay) > avg(f1.departdelay))  (actual time=2.912..3.017 rows=66 loops=1)
                                            -> Table scan on <temporary>  (actual time=0.001..0.019 rows=224 loops=1)
                                                -> Aggregate using temporary table  (actual time=2.903..2.934 rows=224 loops=1)
                                                    -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.032..1.886 rows=1100 loops=1)
                                                        -> Table scan on f1  (cost=111.75 rows=1100) (actual time=0.026..0.385 rows=1100 loops=1)
                                                        -> Single-row index lookup on a1 using PRIMARY (iata_code=f1.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
```

After index created, the sort for arrival delay went down to 6.528 units of time.

Indexed Departure Delay

Created index for departure delay because it is an attribute being averaged.

```
mysql> CREATE INDEX idx_departdelay ON flights(departdelay);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

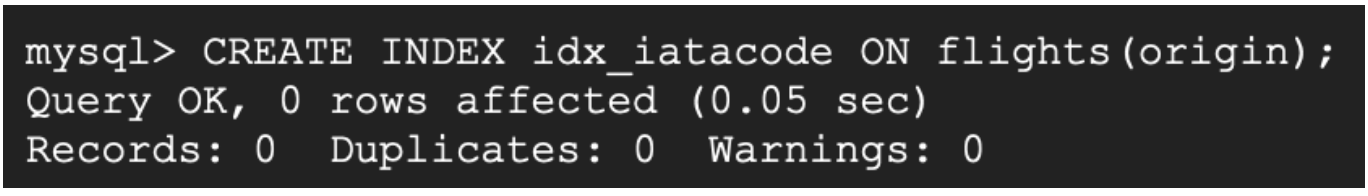


```
-> Sort: arrivaldelaymins DESC  (actual time=7.289..7.294 rows=66 loops=1)
    -> Table scan on <temporary>  (actual time=0.001..0.013 rows=66 loops=1)
        -> Aggregate using temporary table  (actual time=7.215..7.231 rows=66 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=3.805..6.971 rows=203 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.052..0.426 rows=1100 loops=1)
                -> Filter: <in_optimizer>(a.iata_code,a.iata_code in (select #2))  (cost=0.25 rows=1) (actual time=0.006..0.006 rows=0 loops=1100)
                    -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
                    -> Select #2 (subquery in condition; run only once)
                        -> Filter: ((a.iata_code = `<materialized_subquery>`.iata_code))  (actual time=0.001..0.001 rows=0 loops=1067)
                            -> Limit: 1 row(s)  (actual time=0.000..0.000 rows=0 loops=1067)
                                -> Index lookup on <materialized_subquery> using <auto_distinct_key> (iata_code=a.iata_code)  (actual time=0.000..0.000 rows=0 loops=1067)
                                    -> Materialize with deduplication  (cost=0.00..0.00 rows=0) (actual time=4.354..4.354 rows=66 loops=1)
                                        -> Filter: (avg(f1.arrivaldelay) > avg(f1.departdelay))  (actual time=3.483..3.597 rows=66 loops=1)
                                            -> Table scan on <temporary>  (actual time=0.001..0.029 rows=224 loops=1)
                                                -> Aggregate using temporary table  (actual time=3.474..3.514 rows=224 loops=1)
                                                    -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.048..2.306 rows=1100 loops=1)
                                                        -> Table scan on f1  (cost=111.75 rows=1100) (actual time=0.043..0.512 rows=1100 loops=1)
                                                        -> Single-row index lookup on a1 using PRIMARY (iata_code=f1.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
```

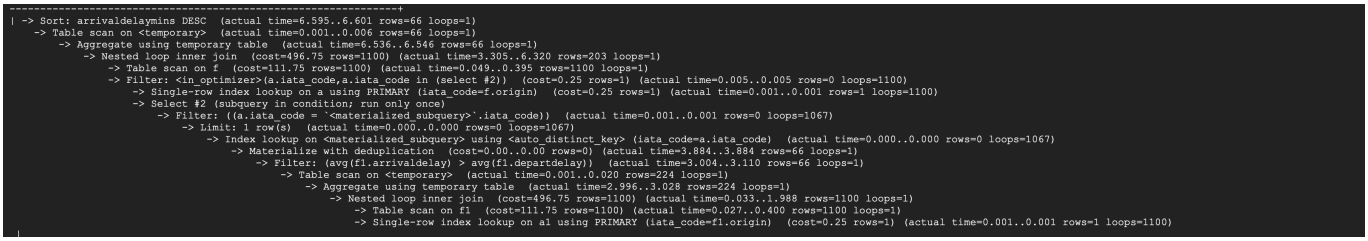
After index created, the subquery comparison for departure delay went down to 3.483 from 3.593.

Indexed IATA Code

Created index for iata code it was the attribute being joined on with 3.805 units of time.



```
mysql> CREATE INDEX idx_iatacode ON flights(origin);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



```
-> Sort: arrivaldelaymins DESC  (actual time=6.595..6.601 rows=66 loops=1)
    -> Table scan on <temporary>  (actual time=0.001..0.006 rows=66 loops=1)
        -> Aggregate using temporary table  (actual time=6.536..6.546 rows=66 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=3.305..6.320 rows=203 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.049..0.395 rows=1100 loops=1)
                -> Filter: <in_optimizer>(a.iata_code,a.iata_code in (select #2))  (cost=0.25 rows=1) (actual time=0.005..0.005 rows=0 loops=1100)
                    -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
                    -> Select #2 (subquery in condition; run only once)
                        -> Filter: ((a.iata_code = `<materialized_subquery>`.iata_code))  (actual time=0.001..0.001 rows=0 loops=1067)
                            -> Limit: 1 row(s)  (actual time=0.000..0.000 rows=0 loops=1067)
                                -> Index lookup on <materialized_subquery> using <auto_distinct_key> (iata_code=a.iata_code)  (actual time=0.000..0.000 rows=0 loops=1067)
                                    -> Materialize with deduplication  (cost=0.00..0.00 rows=0) (actual time=3.884..3.884 rows=66 loops=1)
                                        -> Filter: (avg(f1.arrivaldelay) > avg(f1.departdelay))  (actual time=3.004..3.110 rows=66 loops=1)
                                            -> Table scan on <temporary>  (actual time=0.001..0.020 rows=224 loops=1)
                                                -> Aggregate using temporary table  (actual time=2.996..3.028 rows=224 loops=1)
                                                    -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.033..1.988 rows=1100 loops=1)
                                                        -> Table scan on f1  (cost=111.75 rows=1100) (actual time=0.027..0.400 rows=1100 loops=1)
                                                        -> Single-row index lookup on a1 using PRIMARY (iata_code=f1.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
```

After index created, the nested loop inner join based on the iata code attribute went down to 3.305.

Query 2

```
SELECT a.iata_code, a.airport, COUNT(f.destination) as countdest
FROM airports a JOIN flights f ON (a.iata_code = f.origin)
GROUP BY a.iata_code
ORDER BY countdest DESC;
```

```
mysql> EXPLAIN ANALYZE SELECT a.iata_code, a.airport, COUNT(f.destination) as countdest
    -> FROM airports a JOIN flights f ON (a.iata_code = f.origin)
    -> GROUP BY a.iata_code
    -> ORDER BY countdest DESC;
+--------------------------------------------------------------------------------
| EXPLAIN
                                                                                 |
+--------------------------------------------------------------------------------
| -> Sort: countdest DESC  (actual time=3.946..3.964 rows=224 loops=1)
    -> Table scan on <temporary>  (actual time=0.002..0.029 rows=224 loops=1)
        -> Aggregate using temporary table  (actual time=3.813..3.856 rows=224 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.112..2.617 rows=1100 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.079..0.553 rows=1100 loops=1)
                -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1100)
 |
+--------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

Indexed Destination Airport

Created index for destination airports because the sort by countdest took the longest time at 3.143 units of time.

```
mysql> CREATE INDEX idx_arrivaldelay ON flights(arrivaldelay);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
--------------------------------------------------------------------------------
| -> Sort: countdest DESC  (actual time=2.984..3.002 rows=224 loops=1)
    -> Table scan on <temporary>  (actual time=0.001..0.019 rows=224 loops=1)
        -> Aggregate using temporary table  (actual time=2.887..2.919 rows=224 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.072..1.985 rows=1100 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.051..0.394 rows=1100 loops=1)
                -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
 |
+--------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

After index created, the sort for destination airports went down to 2.984 units of time.

Indexed Airports Iata Codes

Created index for destination airports because the aggregation took a long time at 2.887 units of time.

```
mysql> CREATE INDEX idx_iatacode ON airports(iata_code);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
--------------------------------------------------------------------------------
| -> Sort: countdest DESC  (actual time=2.971..2.987 rows=224 loops=1)
    -> Table scan on <temporary>  (actual time=0.001..0.019 rows=224 loops=1)
        -> Aggregate using temporary table  (actual time=2.869..2.902 rows=224 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.078..1.976 rows=1100 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.056..0.415 rows=1100 loops=1)
                -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1100)
 |
```

After index created, aggregation went down to 2.869 units of time, which is a very minimal decrease but will likely be more influential given a greater amount of data.

Indexed Flights Origin

Created index for origin airports for flight because it is one of the attributes being joined on.

```
mysql> CREATE INDEX idx_origin ON flights(origin);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
| -> Sort: countdest DESC  (actual time=3.327..3.357 rows=224 loops=1)
    -> Table scan on <temporary>  (actual time=0.001..0.021 rows=224 loops=1)
        -> Aggregate using temporary table  (actual time=3.223..3.256 rows=224 loops=1)
            -> Nested loop inner join  (cost=496.75 rows=1100) (actual time=0.074..2.325 rows=1100 loops=1)
                -> Table scan on f  (cost=111.75 rows=1100) (actual time=0.054..0.415 rows=1100 loops=1)
                -> Single-row index lookup on a using PRIMARY (iata_code=f.origin)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1100)
|
```

After index created, inner join time taken went down from 0.112 to 0.074, which seems insignificant but will likely matter more given a larger amount of data.

Conclusion

The six indicies shown above are the ones that we added in order to enhance performance, each of which improves the respective query by some measure as dictated in the explanations. This indexing design was chosen because the attributes indexed are relevant to the query and are used to either join two tables or aggregate data within the column. While the changes seem insignificant, that is due to the small amount of data. When these tables are entirely populated, the performance changes from the indexing are likely to be more prevalent.