Display all of the indexes present in the flights table. (2 points)

```
show indexes from flights;
```

```
flights    1    Year      1    year      A    33                  YES    BTREE
flights    1    Date      1    year      A    33                  YES    BTREE
flights    1    Date      2    month     A    33                  YES    BTREE
flights    1    Date      3    day       A    1042                YES    BTREE
flights    1    Origin    1    origin    A    10191                      BTREE
flights    1    Dest      1    dest      A    10191                      BTREE
flights    1    Carrier   1    carrier   A    538                        BTREE
flights    1    tailNum1       tailnumA       153917              YES    BTREE
```

Display all of the indexes present in the planes table. (2 points)

```
show indexes from planes;
```

```
planes   0    PRIMARY        1    tailnum A    3322        BTREE
```

Display the CREATE TABLE statement for the flights table. Why does it not make sense for this table to have a PRIMARY KEY? (5 points)

```
create table myflights like flights;
```

flights table has no unique index column, it doesn't make sense for this table to have a primary keys.

Why is the partitioning scheme based on year useful for the flights table? Suppose that we wanted to compute the average arrival delay time by carrier from 1987–2016. Would this partitioning scheme help us? (5 points)

Partitioning is a way that database splits actual data here is flights table down into seperate tables, but still get treated as a single table by the SQL layer. Partitioning scheme based on year is useful for the large flights table processing. If we wanted to compute the average arrival delay time by carrier from 1987-2016, this partitioning scheme can improve the calculation performance by averaging on smaller partitions on different year ranges.

Suppose that we defined the partitions based on carrier instead of year. For what usage patterns would that partitioning scheme likely work better than the current scheme based on year? (3 points)

if we defined the partitions based on carrier instead of year, it can work better than the current year partition scheme when we like to get query result group by carrier etc.

Run EXPLAIN on the following two queries (solutions to problems 1 and 2 on the previous assignment). Which query is more efficient and why? (8 points)

The second query with subquery is a bit more efficient.

The first query explain shows the tables are table alias a and f, the second query explain shows the original tables are used.