```
GET kibana_sample_data_flights/_search

## Simple Query
GET _sql
{
  "query": "SELECT * FROM kibana_sample_data_flights LIMIT 10"
}

## Query with a filter
GET _sql
{
  "query": """
  SELECT  FlightNum as FlightNumber,
        OriginCountry,
        Origin,
        DestCountry as DestinationCountry,
        Dest as Destination
  FROM    kibana_sample_data_flights
  WHERE   FlightNum = 'MEWA6Z2'
  """
}


## Text format for interactive querying in Kibana
GET _sql?format=txt
{
  "query": """
  SELECT  FlightNum as FlightNumber,
        OriginCountry,
        Origin,
        Dest as DestinationAirport
  FROM    kibana_sample_data_flights
  WHERE   FlightNum = 'MEWA6Z2'
  """
}

## The SQL interpreter will extract object names in order to create the Abstract Syntax Tree
GET _sql?format=txt
{
  "query": """
  SELECT  FlightNum as FlightNumber,
        OriginCountry as Country,
        "Origin",
        Dest as DestinationAirport
  FROM    "*flights*"
  WHERE   FlightNum = 'MEWA6Z2'
  """
```

```
}

## Translate endpoint will show the optimized Search API for the SQL query
GET _sql/translate
{
  "query": """
  SELECT  FlightNum as FlightNumber,
          OriginCountry,
          Origin,
          DestCountry as DestinationCountry,
          Dest as Destination
  FROM    kibana_sample_data_flights
  WHERE   FlightNum = 'MEWA6Z2'
  """
}


## Another optimizer example - aggregations merging
GET _sql/translate
{
  "query": """
  SELECT    MIN(DistanceMiles) AS MinDistance,
            MAX(DistanceMiles) AS MaxDistance,
            AVG(DistanceMiles) AS AverageDistance
  FROM      kibana_sample_data_flights
  WHERE     OriginCountry = 'US'
  LIMIT     11
  """
}

## Optimizer example - translating into Painless
GET _sql?format=txt
{
  "query": """
  SELECT    Day_OF_WEEK(timestamp) AS WeekDayNum,
            DAY_NAME(timestamp) AS WeekDay,
            COUNT(*) AS Flights
  FROM      kibana_sample_data_flights
  GROUP BY  WeekDayNum,WeekDay
  ORDER BY  WeekDayNum
  """
}

## Optimizer example - translating into Painless #2
GET _sql?format=txt
{
  "query": """
```

```
  SELECT    CASE
            WHEN DAY_OF_WEEK(timestamp) = 1 THEN 7
            ELSE DAY_OF_WEEK(timestamp) - 1
           END AS WeekDayNum,
           DAY_NAME(timestamp) AS WeekDay,
           COUNT(*) AS Flights
  FROM      kibana_sample_data_flights
  GROUP BY  WeekDayNum,WeekDay
  ORDER BY  WeekDayNum
  """
}

################## Fulltext ##################

## Looking at the tables in our cluster
GET _sql?format=txt
{
  "query": "SHOW TABLES"
}

## Show the different columns in the shakespeare table
GET _sql?format=txt
{
  "query": "DESCRIBE shakespeare"
}

## Filtering, usage of text / keyword fields automatically
GET _sql?format=txt
{
  "query": """
  SELECT    play_name,
           text_entry
  FROM      shakespeare
  WHERE     play_name = 'Romeo and juliet'
  LIMIT     50
  """
}

## Matching, using SCORE()
GET _sql?format=txt
{
  "query": """
  SELECT    play_name,
           text_entry
  FROM      shakespeare
  WHERE     MATCH(play_name,'romeo and juliet')
  """
```

```
}


## Finding a play by a famous quote
GET _sql?format=txt
{
  "query": """
  SELECT    play_name,
            speaker,
            text_entry,
            SCORE()
  FROM      shakespeare
  WHERE     MATCH(text_entry,'to be or not to be')
  ORDER BY SCORE() DESC
  LIMIT 10
  """
}

## Matching in different fields, boosting
GET _sql?format=txt
{
  "query": """
  SELECT    play_name,
            speaker,
            text_entry,
            SCORE()
  FROM      shakespeare
  WHERE     MATCH('text_entry,speaker','henry')
  ORDER BY SCORE() DESC
  LIMIT     20
  """
}


## Additional parameters
GET _sql?format=txt
{
  "query": """
  SELECT    play_name, speaker, text_entry, SCORE()
  FROM      shakespeare
  WHERE     MATCH(text_entry,'to be or not to be that is the question',
'minimum_should_match=7')
  ORDER BY SCORE() DESC
  LIMIT     50
  """
}
```

```
## Combining different match queries
GET _sql?format=txt
{
 "query": """
 SELECT    play_name, speaker, text_entry, SCORE()
 FROM      shakespeare
 WHERE     MATCH(text_entry,'to be or not to be')
        AND MATCH(text_entry, 'that is the question', 'operator=and')
 ORDER BY SCORE() DESC
 LIMIT     50
 """
}

################### Aggregations ###################

## Find the 3 longest plays by shakespeare
GET _sql?format=txt
{
 "query": """
 SELECT    play_name as PlayName,
        COUNT(*) as Entries
 FROM      shakespeare
 GROUP BY  play_name
 ORDER BY  Entries DESC
 LIMIT     3
 """
}


## Using FIRST / LAST
GET _sql?format=txt
{
 "query": """
 SELECT    play_name,
        FIRST(text_entry, line_id) as FirstText,
        LAST(text_entry, line_id) as LastText
 FROM      shakespeare
 WHERE     text_entry NOT IN ('ACT I', 'PROLOGUE', 'Exeunt')
        AND play_name IN ('Hamlet', 'Romeo and Juliet')
        AND text_entry NOT LIKE 'SCENE I%'
 GROUP BY  play_name
 """
}


## Filtering aggregation results
GET _sql?format=txt
```

```
{
  "query": """
  SELECT    Origin AS OriginAirport,
            MIN(DistanceMiles) AS MinDistance,
            MAX(DistanceMiles) AS MaxDistance,
            AVG(DistanceMiles) AS AverageDistance
  FROM      kibana_sample_data_flights
  WHERE     OriginCountry = 'US'
  GROUP BY  Origin
  HAVING.   MinDistance > 0
  ORDER BY  MIN(DistanceMiles)
  LIMIT     20
  """
}


## Filtering aggregation results - using Geo
GET _sql?format=txt
{
  "query": """
  SELECT    Origin AS OriginAirport,
            ST_Distance(OriginLocation, DestLocation) AS FlightDistance
  FROM      kibana_sample_data_flights
  WHERE     OriginCountry = 'US'
  LIMIT     20
  """
}
```