databricksAssignment

Queries in Spark SQL

Module 1 Assignment

In this assignment you:

- Create a table
- Write SQL queries

For each **bold** question, input its answer in Coursera.

```
%run ../Includes/Classroom-Setup
```

```
Data mounted to /mnt/davis ...

OK
```

Working with Incident Data

For this assignment, we'll be using a new dataset: the SF Fire Incident (https://data.sfgov.org/Public-Safety/Fire-Incidents/wr8u-xric) dataset. It has been mounted for you using the script above. The path to this dataset is as follows:

`/mnt/davis/fire-incidents/fire-incidents-2016.csv`

In this assignment, you will read the dataset and perform a number of different queries.

Create a Table

Create a new table called `fireIncidents` for this dataset. Be sure to use options to properly parse the data.

```
CREATE TABLE IF NOT EXISTS fireIncidents USING csv
OPTIONS (path '/mnt/davis/fire-incidents/fire-incidents-2016.csv')
```

OK

Question 1

Return the first 10 lines of the data. On the Coursera platform, input the result to the following question:

What is the first value for "Incident Number"?

```
SELECT * FROM fireIncidents
LIMIT 10
-- 16000003
```

	Incident Number	Exposure Number	Address	Incide
1	16000003	0	Precita Av/florida Street	01/01/2
2	16000004	0	1620 Eucalyptus Drive	01/01/2
3	16000023	0	171 2nd Street	01/01/2
4	16000034	0	535 Wisconsin Street	01/01/2
5	16000051	0	El Camino Del Mar/seal Rock Drive	01/01/2
6	16000053	0	443 Texas Street	01/01/2
7	16000064	0	1217 Ralston	01/01/2
8	16000071	0	Fell St/laguna Street	01/01/2

Showing all 10 rows.

`WHERE` Clauses

A `WHERE` clause is used to filter data that meets certain criteria, returning all values that evaluate to be true.

Question 2

Return all incidents that occurred on Conor's birthday in 2016. For those of you who forgot his birthday, it's April 4th. On the Coursera platform, input the result to the following question:

What is the first value for "Incident Number" on April 4th, 2016?

Remember to use backticks (``) instead of single quotes ('') for columns that have spaces in the name.

```
SELECT * FROM fireIncidents
WHERE `Incident Date` = '04/04/2016'
--16037478
```

	Incident Number	Exposure Number	Address	Incident Da
1	16037478	0	Utah St/15th Street	04/04/2016
2	16037483	0	1755 Ofarrell Street	04/04/2016
3	16037484	0	Leavenworth St/eddy Street	04/04/2016
4	16037492	0	California St/davis Street	04/04/2016
5	16037503	0	765 Burnett Avenue	04/04/2016
6	16037505	0	90 Saturn Street	04/04/2016
7	16037508	0	24th St/folsom Street	04/04/2016
8	16037530	0	100 Pine Street	04/04/2016

Showing all 80 rows.

Question 3

Return all incidents that occurred on Conor's *or* Brooke's birthday. For those of you who forgot her birthday too, it's `9/27`.

Is the first fire call in this table on Brooke or Conor's birthday?

```
SELECT * FROM fireIncidents
WHERE `Incident Date` = '04/04/2016' OR
`Incident Date` = '27/09/2016'
--Conor
```

	Incident Number	Exposure Number	Address	Incident Da

1	16037478	0	Utah St/15th Street	04/04/2016
2	16037483	0	1755 Ofarrell Street	04/04/2016
3	16037484	0	Leavenworth St/eddy Street	04/04/2016
4	16037492	0	California St/davis Street	04/04/2016
5	16037503	0	765 Burnett Avenue	04/04/2016
6	16037505	0	90 Saturn Street	04/04/2016
7	16037508	0	24th St/folsom Street	04/04/2016
8	16037530	0	100 Pine Street	04/04/2016

Showing all 80 rows.

Question 4

Return all incidents on either Conor or Brooke's birthday where the `Station Area` is greater than 20.

What is the "Station Area" for the first fire call in this table?

Aggregate Functions

Aggregate functions compute a single result value from a set of input values. Use the aggregate function `COUNT` to count the total records in the dataset.

```
SELECT * FROM fireIncidents
WHERE `Station Area` > 20 AND
`Incident Date` = '04/04/2016' OR
`Incident Date` = '27/09/2016'
```

	Incident Number	Exposure Number	Address	Incident Da
1	16037478	0	Utah St/15th Street	04/04/2016
2	16037503	0	765 Burnett Avenue	04/04/2016
3	16037543	0	41 Castle Street	04/04/2016
4	16037566	0	4101 Noriega St 6 Street	04/04/2016
5	16037573	0	20 12th Street	04/04/2016
6	16037582	0	768 18th Avenue	04/04/2016
7	16037589	0	560 Central Avenue	04/04/2016
8	16037591	0	747 Ellsworth Street	04/04/2016

Showing all 29 rows.

Question 5

Count the incidents on Conor's birthday.

How many incidents were on Conor's birthday in 2016?

```
SELECT COUNT(*) FROM fireIncidents
WHERE `Incident Date` = '04/04/2016'
```

	count(1)
1	80

Showing all 1 rows.

Question 6

Return the total counts by `Ignition Cause`. Be sure to return the field `Ignition Cause` as well.

Hint: You'll have to use `GROUP BY` for this

```
SELECT `Ignition Cause`, COUNT(*)
FROM fireIncidents
GROUP BY `Ignition Cause`
--5
```

	Ignition Cause	count(1)
1	null	30690
2	3 failure of equipment or heat source	100
3	u cause undetermined after investigation	156
4	4 act of nature	5
5	5 cause under investigation	35
6	2 unintentional	561
7	0 cause, other (only used for additional exposures)	1
8	1 intentional	223

Showing all 8 rows.

Sorting

Question 7

Return the total counts by `Ignition Cause` sorted in ascending order.

Hint: You'll have to use `ORDER BY` for this.

What is the most common "Ignition Cause"? (Put the entire string)

```
SELECT `Ignition Cause`, COUNT(*)
FROM fireIncidents
GROUP BY `Ignition Cause`
ORDER BY COUNT(*) ASC
--2 unintentional
```

	Ignition Cause	count(1)
1	0 cause, other (only used for additional exposures)	1
2	4 act of nature	5
3	5 cause under investigation	35
4	3 failure of equipment or heat source	100
5	u cause undetermined after investigation	156
6	1 intentional	223
7	2 unintentional	561
8	null	30690

Showing all 8 rows.

Return the total counts by `Ignition Cause` sorted in descending order.

```
SELECT `Ignition Cause`, COUNT(*)
FROM fireIncidents
GROUP BY `Ignition Cause`
ORDER BY COUNT(*) DESC
```

	Ignition Cause	count(1)
1		

1	null	30690
2	2 unintentional	561
3	1 intentional	223
4	u cause undetermined after investigation	156
5	3 failure of equipment or heat source	100
6	5 cause under investigation	35
7	4 act of nature	5
8	0 cause, other (only used for additional exposures)	1

Showing all 8 rows.

Joins

Create the table `fireCalls` if it doesn't already exist. The path is as follows: `/mnt/davis/fire-calls/fire-calls-truncated-comma.csv`

```
CREATE TABLE IF NOT EXISTS fireCalls USING csv
OPTIONS ( path '/mnt/davis/fire-calls/fire-calls-truncated-comma.csv')
```

OK

Join the two tables on `Battalion` by performing an inner join.

```
SELECT * FROM fireIncidents
INNER JOIN fireCalls
ON fireIncidents.Battalion = fireCalls.Battalion
```

	Incident Number	Exposure Number	Address	Incident Date
1	16000199	0	17th St/sanchez Street	01/01/2016
2	16000199	0	17th St/sanchez Street	01/01/2016
3	16000199	0	17th St/sanchez Street	01/01/2016
4	16000199	0	17th St/sanchez Street	01/01/2016
5	16000199	0	17th St/sanchez Street	01/01/2016
6	16000199	0	17th St/sanchez Street	01/01/2016
7	16000199	0	17th St/sanchez Street	01/01/2016
8	16000199	0	17th St/sanchez Street	01/01/2016

Showing the first 1000 rows.

Question 8

Count the total incidents from the two tables joined on `Battalion`.

What is the total incidents from the two joined tables?

```
SELECT COUNT(*) FROM fireIncidents
INNER JOIN fireCalls
ON fireIncidents.Battalion = fireCalls.Battalion
```

	count(1)
1	847094402

Showing all 1 rows.

Congratulations! You made it to the end of the assignment!