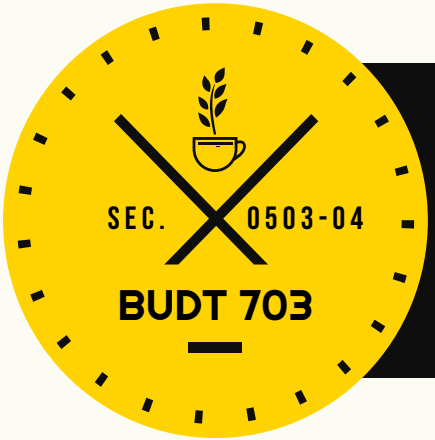
AARTI YELNE, BRENDAN COHEN, DARREN SEQUEIRA, SANIKA PATIL



CAFFEINE

RAIDERS

10 DECEMBER 2021

Background Information

Caffeine Raiders

Consultant firm who is providing consultancy about the top cafés in College Park, MD

Goal

To understand what the student's preferences are and on what attributes the preferences are based on

Users

Students of the University of Maryland who want to find the best Cafés in the area

Data Sources

Location Data

Foursquare API in python. Filtering out by College Park location and “Cafe|Coffee”

Reviews

Data is gathered from Yelp, TripAdvisor, and Foursquare.

Features and Menu Items

Features and Menu Items are gathered from Cafe websites and Google Maps Data

Location Data

```
radius = 5000
LIMIT = 1000
urlumd =
'https://api.foursquare.com/v2/venues/
explore?&client_id={}&client_secret={}
&v={}&ll={},{}&radius={}&limit={}'.
format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    latitude,
    longitude,
    radius,
    LIMIT)
```


```
{'meta': {'code': 200, 'requestId':
'61a52a2348a0040c921849a7'},
 'response': {'groups': [{'items':
[{'reasons': {'count': 0,
       'items': [{'reasonName':
'globalInteractionReason',
         'summary': 'This spot is
popular',
         'type': 'general'}]},
      'referralId': 'e-0-
4b09f512f964a520612023e3-0',
      'venue': {'categories':
[{'icon': {'prefix':
'
https://ss3.4sqi.net/img/categories_v2
/arts_entertainment/performingarts_',
          'suffix': '.png'},
         'id':
'4bf58dd8d48988d1f2931735',
```

Location Data

	City	Name	Categories	Latitude	Longitude	Distance
0	College Park	NuVegan Café	Vegetarian / Vegan Restaurant	38.991512	-76.933671	0.671081
1	College Park	Milk & Honey Cafe	Café	39.023296	-76.924244	2.460629
2	College Park	Vigilante Coffee	Coffee Shop	38.992042	-76.933623	0.671081
3	College Park	Starbucks	Coffee Shop	38.981861	-76.947797	0.708363
4	College Park	IKEA Restaurant & Café	Café	39.020724	-76.931024	2.143730
5	College Park	7-Eleven	Convenience Store	38.998382	-76.924738	1.230315
6	College Park	Board & Brew	Gaming Cafe	38.991618	-76.933648	0.671081
7	College Park	Casey's Coffee & Sandwiches	Café	38.984595	-76.951476	0.590302
8	College Park	Dunkin'	Coffee Shop	38.984632	-76.966796	1.224101
9	College Park	Rudy's Cafe	Café	38.984776	-77.019327	3.964347

Mission Statement

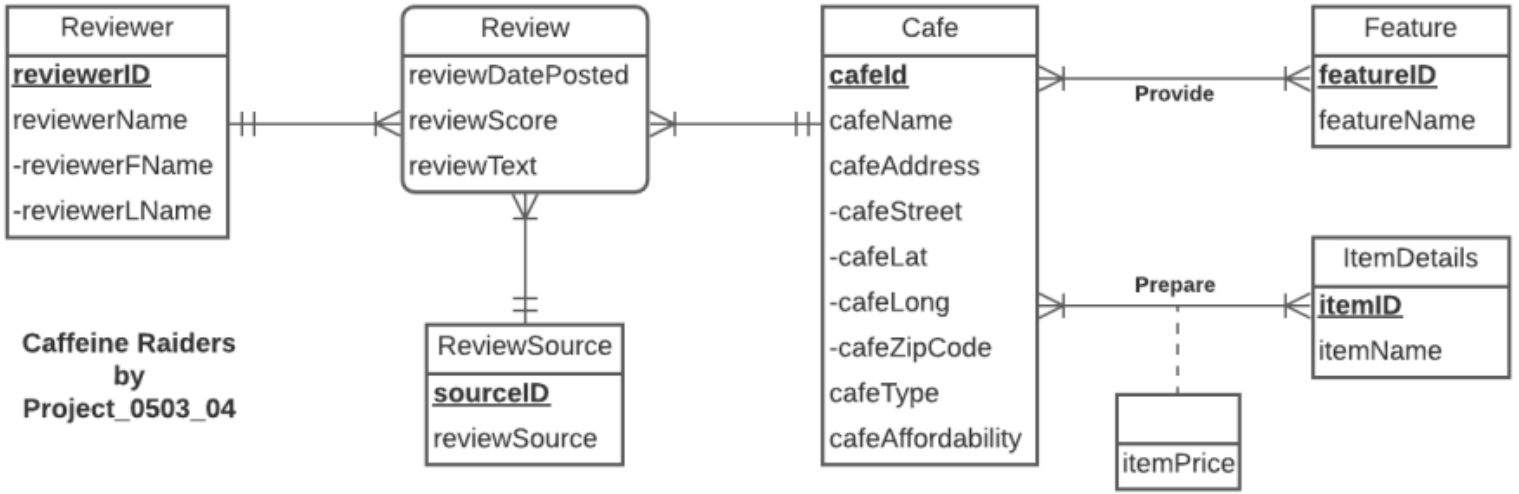
Caffeine Raiders is a consultant firm who is providing consultancy about the top cafés in College Park, MD. By analysing reviews and ratings of the top cafes across multiple review sites, the aim is to understand what the student's preferences are and on what attributes the preferences are based on.

Mission Objectives

The objectives are:

- Determine the full details of all reviewers that review each cafe
- Number of unique streets the cafes are on
- Average price of the menu items for each cafe
- Cafes that offer Wifi, chai tea is available in which cafes on Baltimore Ave
- Cafes that have at least one review on Yelp
- Cafes that have highest review score in 2021
- Cafes that have the highest affordability and their address
- Reviewers that have reviewed coffee and toasts for College Park Cafes
- Most popular features amongst all cafes

ER Diagram


Reviewer
reviewerID
reviewerName
-reviewerFName
-reviewerLName
Review
reviewDatePosted
reviewScore
reviewText
ReviewSource
sourceID
reviewSource
Cafe
cafeId
cafeName
cafeAddress
-cafeStreet
-cafeLat
-cafeLong
-cafeZipCode
cafeType
cafeAffordability
Provide
Feature
featureID
featureName
Prepare
ItemDetails
itemID
itemName
itemPrice
Caffeine Raiders
by
Project_0503_04

Relational Schema

- Café (**cafeID,** cafeName, cafeType, cafeAffordability, cafeStreet, cafeLat, cafeLong, cafeZipCode)
- Feature (**featureID**, featureName)
- ProvidedFeatures (***cafeID, featureID***)
- ItemDetails (**itemID,** itemName)
- Prepares (***cafeID, itemID,*** itemPrice)
- Reviewer (**reviewerID,** reviewerFName, reviewerLName)
- ReviewSource (**sourceID**, reviewSource)
- Review (***sourceID***, ***reviewerID***, ***cafeID***, reviewScore, reviewDatePosted, reviewText)

Physical Database Design

```
DROP TABLE IF EXISTS [CF.Review]
DROP TABLE IF EXISTS [CF.ReviewSource]
DROP TABLE IF EXISTS [CF.Reviewer]
DROP TABLE IF EXISTS [CF.Prepares]
DROP TABLE IF EXISTS [CF.ItemDetails]
DROP TABLE IF EXISTS [CF.ProvidedFeatures]
DROP TABLE IF EXISTS [CF.Feature]
DROP TABLE IF EXISTS [CF.Cafe]
```

Physical Database Design

```
CREATE TABLE [CF.Cafe] (
    cafeID CHAR(8) NOT NULL,
    cafeName VARCHAR(50),
    cafeType VARCHAR(50),
    cafeAffordability DECIMAL (2, 1),
    cafeStreet VARCHAR(20),
    cafeLat DECIMAL(8,6),
    cafeLong DECIMAL(8,6),
    cafeZipCode CHAR(5),
    CONSTRAINT pk_cafe_cafeID PRIMARY KEY (cafeID)
)
```

Physical Database Design

```
CREATE TABLE [CF.Feature] (
    featureID CHAR(8) NOT NULL,
    featureName VARCHAR(20),
    CONSTRAINT pk_feature_featureID PRIMARY KEY (featureID)
)
```

Physical Database Design

```
CREATE TABLE [CF.ProvidedFeatures]
(
    cafeID CHAR(8) NOT NULL,
    featureID CHAR(8) NOT NULL,
    CONSTRAINT pk_providedfeatures PRIMARY KEY (cafeID, featureID),
    CONSTRAINT fk_providedfeatures_cafeID FOREIGN KEY (cafeID)
        REFERENCES [CF.Cafe] (cafeID)
            ON UPDATE CASCADE
            ON DELETE NO ACTION,
    CONSTRAINT fk_providedfeatures_featureID FOREIGN KEY (featureID)
        REFERENCES [CF.Feature] (featureID)
            ON UPDATE CASCADE
            ON DELETE NO ACTION
)
```

Physical Database Design

```
CREATE TABLE [CF.ItemDetails] (
    itemID CHAR(8) NOT NULL,
    itemName VARCHAR(20),
    CONSTRAINT pk_itemdetails_itemID PRIMARY KEY (itemID)
)
```

Physical Database Design

```
CREATE TABLE [CF.Prepares] (
    cafeID CHAR(8) NOT NULL,
    itemID CHAR(8) NOT NULL,
    itemPrice DECIMAL (5, 2),
    CONSTRAINT pk_prepares PRIMARY KEY (cafeID, itemID),
    CONSTRAINT fk_prepares_cafeID FOREIGN KEY (cafeID)
        REFERENCES [CF.Cafe] (cafeID) ON UPDATE CASCADE ON DELETE NO ACTION,
    CONSTRAINT fk_prepares_itemID FOREIGN KEY (itemID)
        REFERENCES [CF.ItemDetails] (itemID) ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

Physical Database Design

```
CREATE TABLE [CF.Reviewer] (
    reviewerID CHAR(8) NOT NULL,
    reviewerFName VARCHAR(20),
    reviewerLName VARCHAR(20),
    CONSTRAINT pk_reviewer_reviewerID PRIMARY KEY (reviewerID)
)
```

Physical Database Design

```
CREATE TABLE [CF.ReviewSource] (
    sourceID CHAR(8) NOT NULL,
    reviewSource VARCHAR(20),
    CONSTRAINT pk_reviewsource_sourceID PRIMARY KEY (sourceID)
)
```

Physical Database Design

```
CREATE TABLE [CF.Review] (
    cafeID CHAR(8) NOT NULL,
    sourceID CHAR(8) NOT NULL,
    reviewerID CHAR(8) NOT NULL,
    reviewDatePosted DATE,
    reviewScore DECIMAL(2,1),
    reviewText VARCHAR(1000),
    CONSTRAINT pk_reviewdetails PRIMARY KEY (cafeID, sourceID, reviewerID),
    CONSTRAINT fk_reviewdetails_sourceID FOREIGN KEY (sourceID)
        REFERENCES [CF.ReviewSource] (sourceID) ON UPDATE CASCADE ON DELETE NO ACTION,
    CONSTRAINT fk_reviewdetails_reviewerID FOREIGN KEY (reviewerID)
        REFERENCES [CF.Reviewer] (reviewerID) ON UPDATE NO ACTION ON DELETE CASCADE,
    CONSTRAINT fk_reviewdetails_cafeID FOREIGN KEY (cafeID)
        REFERENCES [CF.Cafe] (cafeID) ON UPDATE CASCADE ON DELETE CASCADE
)
```

Use Cases

Use Case #1:

What are the full details of all reviewers that review each cafe?

```
CREATE VIEW [Reviewer Details] AS
    SELECT rr.*, c.cafeName
    FROM [CF.Cafe] c, [CF.Review] r, [CF.Reviewer] rr
    WHERE c.cafeID = r.cafeID
    AND r.reviewerID = rr.reviewerID
GO
```

Application

Use Case #1:

What are the full details of all reviewers that review each cafe?

	reviewerID	reviewerFName	reviewerLName	cafeName
1	revr_001	Imran	S	NuVegan Cafe
2	revr_002	bcsanders	NULL	NuVegan Cafe
3	revr_003	Tea	D	Milk and Honey Café
4	revr_004	Italiano58	NULL	Milk and Honey Café
5	revr_005	Kyle	L	Vigilante Coffee
6	revr_006	laurib471	NULL	Vigilante Coffee
7	revr_007	Kosha	L	Starbucks
8	revr_008	Julstumpff	NULL	Starbucks
9	revr_009	Jeena	L	Ikea Restaurant and Café
10	revr_010	TerryElvers	NULL	Ikea Restaurant and Café

Use Cases

Use Case #2:

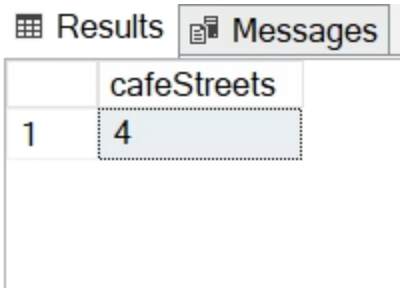
How many unique streets are the cafes on?

```
CREATE VIEW [Unique Streets] AS
    SELECT COUNT(DISTINCT c.cafeStreet) AS 'cafeStreets'
    FROM [CF.Cafe] c
GO
```

Application

Use Case #2:

How many unique streets are the cafes on?



Use Cases

Use Case #3:

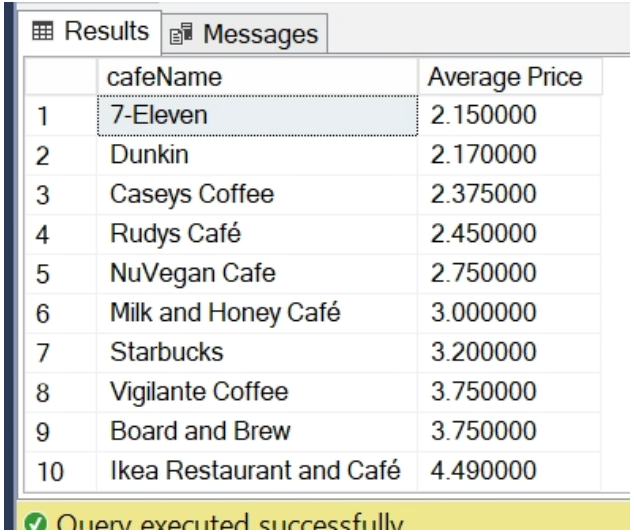
For each cafe, what is the average price of the menu items?

```
CREATE VIEW [Average Price] AS
    SELECT TOP 10 c.cafeName, AVG(p.itemPrice) AS 'Average Price'
    FROM [CF.Prepares] p, [CF.Cafe] c
    WHERE c.cafeID = p.cafeID
    GROUP BY c.cafeName
    ORDER BY 'Average Price'
GO
```

Application

Use Case #3:

For each cafe, what is the average price of the menu items?



Results | Messages

	cafeName	Average Price
1	7-Eleven	2.150000
2	Dunkin	2.170000
3	Caseys Coffee	2.375000
4	Rudys Café	2.450000
5	NuVegan Cafe	2.750000
6	Milk and Honey Café	3.000000
7	Starbucks	3.200000
8	Vigilante Coffee	3.750000
9	Board and Brew	3.750000
10	Ikea Restaurant and Café	4.490000

Query executed successfully

Use Cases

Use Case #4:

What are the cafes that offer Wifi?

```
CREATE VIEW [Cafes With Wifi] AS
    SELECT c.cafeName, f.featureName
    FROM [CF.Cafe] c, [CF.Feature] f, [CF.ProvidedFeatures] pf
    WHERE c.cafeID = pf.cafeID
    AND pf.featureID = f.featureID
    AND f.featureName = 'Offers Wifi'
GO
```

Application

Use Case #4:

What are the cafes that offer Wifi?

	cafeName	featureName
1	NuVegan Cafe	Offers Wifi
2	Milk and Honey Café	Offers Wifi
3	Starbucks	Offers Wifi
4	Board and Brew	Offers Wifi
5	Caseys Coffee	Offers Wifi
6	Rudys Café	Offers Wifi

Use Cases

Use Case #5:

Where on Baltimore Ave can I get a chai tea?

```
CREATE VIEW [Chai Tea Baltimore Ave] AS
    SELECT c.cafeName, c.cafeStreet, i.itemName
    FROM [CF.Cafe] c, [CF.ItemDetails] i, [CF.Prepares] p
    WHERE c.cafeID = p.cafeID
    AND p.itemID = i.itemID
    AND c.cafeStreet = 'Baltimore Ave'
    AND i.itemName = 'Chai Tea'
GO
```

Application

Use Case #5:

Where on Baltimore Ave can I get a chai tea?

Results | Messages

	cafeName	cafeStreet	itemName
1	NuVegan Cafe	Baltimore Ave	Chai Tea
2	Milk and Honey Café	Baltimore Ave	Chai Tea
3	Vigilante Coffee	Baltimore Ave	Chai Tea
4	Starbucks	Baltimore Ave	Chai Tea
5	Ikea Restaurant and Café	Baltimore Ave	Chai Tea
6	7-Eleven	Baltimore Ave	Chai Tea
7	Board and Brew	Baltimore Ave	Chai Tea

Use Cases

Use Case #6:

Which cafes have at least one review on Yelp?

```
CREATE VIEW [Review Source as Yelp] AS
    SELECT c.cafeName
    FROM [CF.Cafe] c
    WHERE c.cafeID IN (
        SELECT rd.cafeID
        FROM [CF.Review] rd, [CF.ReviewSource] rs
        WHERE rd.sourceID = rs.sourceID AND rs.reviewSource = 'Yelp'
    )
GO
```

Application

Use Case #6:

Which cafes have at least one review on Yelp?

Results | Messages

	cafeName
1	NuVegan Cafe
2	Milk and Honey Café
3	Vigilante Coffee
4	Starbucks
5	Ikea Restaurant and Café
6	7-Eleven
7	Board and Brew
8	Caseys Coffee
9	Dunkin

Use Cases

Use Case #7:

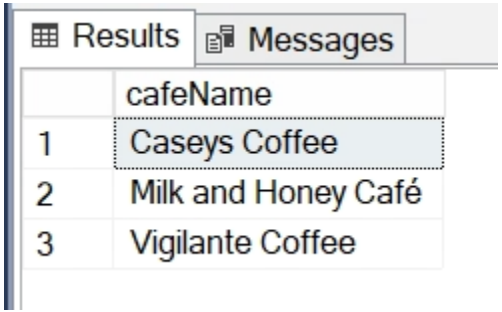
Which are the cafes reviewed in 2021 which have the highest score?

```
CREATE VIEW [Cafes reviewed in 2021 with Score 5] AS
    SELECT DISTINCT c.cafeName
    FROM [CF.Cafe] as c, [CF.Review] as r
    WHERE c.cafeID = r.cafeID
    AND (r.reviewDatePosted > '20210101' AND r.reviewScore>4)
GO
```

Application

Use Case #7:

Which are the cafes reviewed in 2021 which have the highest score?



	cafeName
1	Caseys Coffee
2	Milk and Honey Café
3	Vigilante Coffee

Use Cases

Use Case #8:

Which Cafes have the highest Affordability and what is their address?

```
CREATE VIEW [Affordable Cafe] AS
    SELECT c.cafeID, c.cafeName, c.cafeStreet, c.cafeZipCode
    FROM [CF.Cafe] c
    WHERE c.cafeAffordability IN (
        SELECT MAX(cf.cafeAffordability)
        FROM [CF.Cafe] cf
    )
GO
```

Application

Use Case #8:

Which Cafes have the highest Affordability and what is their address?

Results | Messages

	cafeID	cafeName	cafeStreet	cafeZipCode
1	cafe_001	NuVegan Cafe	Baltimore Ave	20740
2	cafe_002	Milk and Honey Café	Baltimore Ave	20740
3	cafe_004	Starbucks	Baltimore Ave	20740
4	cafe_007	Board and Brew	Baltimore Ave	20740

Use Cases

Use Case #9:

Which reviewers have reviewed coffee and toasts for College Park Cafes? Order the result by their first name.

```
CREATE VIEW [Reviewers for Coffee Toast] AS
    SELECT rw.*
    FROM [CF.Review] rd, [CF.Reviewer] rw
    WHERE (rd.reviewText LIKE '%coffee%' OR rd.reviewText LIKE '%toast%' )
        AND rw.reviewerID = rd.reviewerID
GO
```

Application

Use Case #9:

Which reviewers have reviewed coffee and toasts for College Park Cafes? Order the result by their first name.

Results | Messages

	reviewerID	reviewerFName	reviewerLName
1	revr_003	Tea	D
2	revr_005	Kyle	L
3	revr_006	laurib471	NULL
4	revr_008	Julstumpff	NULL
5	revr_009	Jeena	L
6	revr_014	rejeneration	NULL
7	revr_015	Jonathan	P
8	revr_016	JAMtraveler95	NULL

Use Cases

Use Case #10:

What are the most popular features amongst all cafes and with their counts?

```
CREATE VIEW [Popular Cafe Feature] AS
    SELECT TOP 5 ft.featureName AS 'Feature Name', count(ft.featureID) AS 'Number of Cafes'
    FROM  [CF.Cafe] cf, [CF.Feature] ft, [CF.ProvidedFeatures] pf
    WHERE cf.cafeID = pf.cafeID
        AND ft.featureID = pf.featureID
    GROUP BY ft.featureName
    ORDER BY [Number of Cafes] DESC
GO
```

Application

Use Case #10:

What are the most popular features amongst all cafes and with their counts?

Results | Messages

	Feature Name	Number of Cafes
1	Offers Wifi	6
2	Open All Days	5
3	Afternoon	5
4	Evening	5
5	Morning	4

Conclusion

Thanks!

Any questions ?