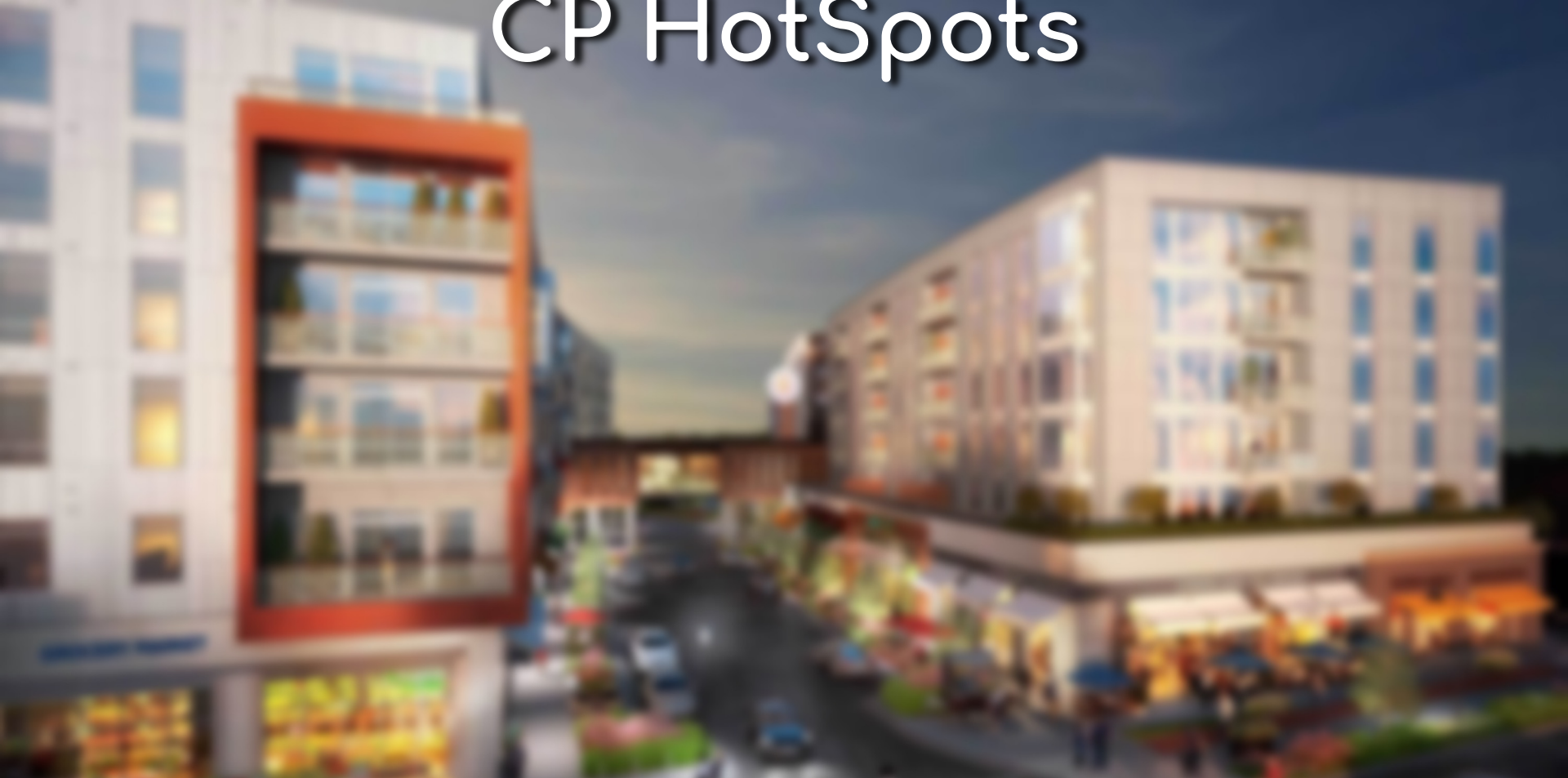


CP HotSpots



Group 6: Nisarg Patel, Kai-Hsiang Lin, Parth Nanwani, and Snigdha Madiraju
December 6th, 2021

Background

- Target users:
 - Potential/future restaurant business owners in College Park
 - Newcomers in College Park
- Database includes:
 - Restaurant
 - Customers/reviewers
 - Review information
- Database also includes:
 - Operation hours
 - Restaurant categories
- Across 3 of the largest online review sites:
 - Yelp
 - Google/Google Reviews
 - TripAdvisor



Introduction

Mission Statements

1. To offer insights to potential new business owners and help them understand target customers and determine pricing strategy to maximize growth potential and profit margin.
2. To provide useful information about restaurants around College Park and to help newcomers explore local restaurants and find a good place to dine at.

Mission Objectives

1. To identify which city the most reviewers come from so restaurants can recognize their target customers.
2. To identify the most popular price level so that restaurants can make better pricing decisions that attract more customers.
3. To find the highest rated restaurant across all price level so that consumers have a wide variety of choices of high quality restaurants.
4. To find the top 3 highest rated restaurants, so newcomers can find the best places to dine at.
5. To find the restaurant that operates for the longest time so newcomers know the restaurant with the highest possibility to be opened.

Business Processes / Transactions

1. Which city has the most number of customers that posted a review?
2. What PriceLevel is the most popular among all restaurants?
3. Which restaurant, across all PriceLevel, has the highest rating?
4. Which are the names of the top 3 restaurants without considering their PriceLevel?
5. Which restaurant has the highest opening hours?

Conceptual Database Design: ER Schema

Restaurant (rstId, rstName, rstLocation, -rstStreet, -rstCity, -rstState, -rstZip, rstPhone, rstStar, rstPriceLevel, rstReviewCount, rstCategory [1..5])

Customer (cusId, cusName, -cusFName, -cusLName, cusOrigin, -cusCity, -cusState)

Source (srcId, srcName)

OperationTime (oprDay, oprHours, -oprStart, -oprEnd, =totalHours)

ER Schema: Relationships, Degrees and Constraints

Review (rvwId, rvwStar, rvwDate, rvwPhotoCount): ternary relationship

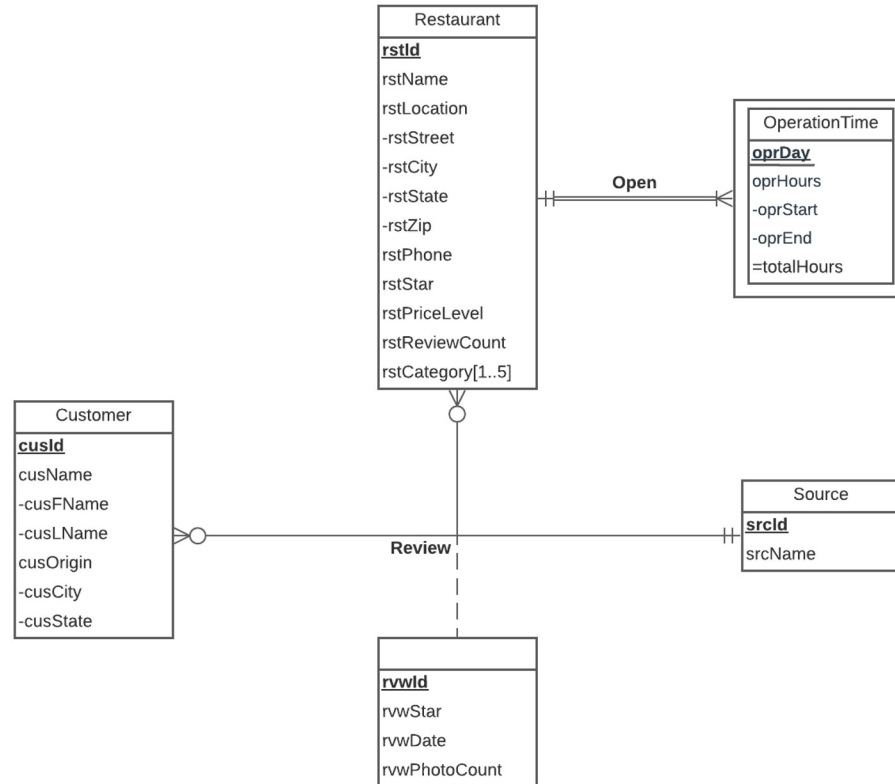
- 1 Customer and 1 Restaurant to 1 Source
- 1 Customer and 1 Source to 0 or many Restaurants
- 1 Restaurant and 1 Source to 0 or many Customers

Open: binary relationship

- 1 Restaurant to 1 or many OperationTime
- 1 OperationTime to 1 Restaurant

ER Diagram

Project_0501_06_ERD



Logical Database Design: Relational Schema

Restaurant (rstId, rstName, rstStreet, rstCity, rstState, rstZip, rstPhone, rstStar, rstPriceLevel, rstReviewCount)

Customer (cusId, cusFName, cusLName, cusCity, cusState)

Source (srcId, srcName)

RestaurantCategory (rstId, rstCat)

OperationTime (rstId, oprDay, oprStart, oprEnd)

Review (rvwId, *rstId*, *cusId*, *srcId*, rvwStar, rvwDate, rvwPhotoCount)

Relational Schema: Functional Dependencies

$\text{rstId} \rightarrow \text{rstName}, \text{rstStreet}, \text{rstCity}, \text{rstState}, \text{rstZip}, \text{rstPhone}, \text{rstStar}, \text{rstPriceLevel}, \text{rstReviewCount}$

$\text{cusId} \rightarrow \text{cusFName}, \text{cusLName}, \text{cusCity}, \text{cusState}$

$\text{srcId} \rightarrow \text{srcName}$

$\text{rstId}, \text{rstCat} \rightarrow$

$\text{oprDay} \rightarrow \text{oprStart}, \text{oprEnd}$

$\text{rvwId} \rightarrow \text{rstId}, \text{cusId}, \text{srcId}, \text{rvwStar}, \text{rvwDate}, \text{rvwPhotoCount}$

Business Rules (1)

[R1] When a restaurant is deleted from the database, the corresponding restaurant category information should be deleted from the database.

[R2] When a restaurant changes information in the database, the corresponding restaurant category information should be changed accordingly.

[R3] When a restaurant is deleted from the database, the operation time information of that restaurant shall be removed from the database.

[R4] When the information on a restaurant is changed in the database, the corresponding operation time information should be changed accordingly.

Business Rules (2)

[R5] When a restaurant is removed from the database, all reviews posted for that restaurant should be removed.

[R6] When information about a restaurant is updated or changed in the database, the corresponding review information for that restaurant should be changed accordingly.

[R7] When a customer is removed from the database, all reviews posted by the customer should be removed.

[R8] When a customer updates his or her information, the corresponding review information posted by that customer should be changed accordingly.

Business Rules (3)

[R9] When a source is removed from the database, all reviews posted on that source should be removed.

[R10] When information about a source is updated or changed in the database, the corresponding review information posted on that source should be updated accordingly.

Referential Integrity Actions

| Relation | Foreign Key | Base Relation | Primary Key | Business Rule | Constraint : ON DELETE | Business Rule | Constraint: ON UPDATE |
|---------------------|-------------|---------------|-------------|---------------|------------------------|---------------|-----------------------|
| Restaurant Category | rstld | Restaurant | rstld | R1 | CASCADE | R2 | CASCADE |
| Operation Time | rstld | Restaurant | rstld | R3 | CASCADE | R4 | CASCADE |
| Review | rstld | Restaurant | rstld | R5 | CASCADE | R6 | CASCADE |
| Review | cusld | Customer | cusld | R7 | CASCADE | R8 | CASCADE |
| Review | srclid | Source | srclid | R9 | CASCADE | R10 | CASCADE |

Sample Data

Restaurant ('R08', 'Ledo Pizza', '4509 Knox Rd', 'College Park', 'MD', '20740', '3014228122', 4.2, '\$\$', 804)

Customer ('C13', 'Ronnie', 'T', 'Hyattsville', 'MD')

Source ('S01', 'Yelp')

RestaurantCategory ('R08', 'Pizza')

OperationTime ('R08', 'Mon', 1100, 2200)

Review ('V40', 'R08', 'C40', 'S02', 3.0, '2019-11-15', 3)



Physical Database Design - CREATE TABLE



```
CREATE TABLE [Hotspots.Restaurant] (  
    rstId CHAR (3) NOT NULL,  
    rstName VARCHAR (30) NOT NULL ,  
    rstStreet VARCHAR (40) ,  
    rstCity VARCHAR (15) ,  
    rstState CHAR (2) ,  
    rstZip CHAR (10) ,  
    rstPhone CHAR (10) ,  
    rstStar FLOAT,  
    rstPriceLevel VARCHAR (4),  
    rstReviewCount INTEGER,  
    CONSTRAINT pk_Restaurant_rstId PRIMARY KEY  
(rstId))
```

```
CREATE TABLE [Hotspots.Customer] (  
    cusId CHAR (3) NOT NULL,  
    cusFName VARCHAR (20) ,  
    cusLName VARCHAR(20) ,  
    cusCity VARCHAR(20) ,  
    cusState CHAR(2) ,  
    CONSTRAINT pk_Customer_cusId PRIMARY KEY  
(cusId))
```



Physical Database Design - CREATE TABLE



```
CREATE TABLE [Hotspots.Source] (  
    srcId CHAR (3) NOT NULL,  
    srcName VARCHAR (15),  
    CONSTRAINT pk_Source_srcId PRIMARY KEY  
(srcId))
```

```
CREATE TABLE [Hotspots.RestaurantCategory] (  
    rstId CHAR (3) NOT NULL,  
    rstCat VARCHAR (20) NOT NULL,  
    CONSTRAINT  
pk_RestaurantCategory_rstId_rstCat PRIMARY KEY  
(rstId,rstCat),  
    CONSTRAINT fk_RestaurantCategory_rstId FOREIGN  
KEY (rstId)  
    REFERENCES [HotSpots.Restaurant] (rstId)  
    ON DELETE CASCADE ON UPDATE CASCADE)
```


```
CREATE TABLE [Hotspots.OperationTime] (  
    rstId CHAR (3) NOT NULL,  
    oprDay CHAR (3) NOT NULL,  
    oprStart INTEGER,  
    oprEnd INTEGER,  
    CONSTRAINT pk_OperationTime_rstId_oprDay  
PRIMARY KEY (rstId, oprDay),  
    CONSTRAINT fk_OperationTime_rstId FOREIGN KEY  
(rstId)  
    REFERENCES [Hotspots.Restaurant] (rstId)  
    ON DELETE CASCADE ON UPDATE CASCADE)
```



Physical Database Design - CREATE TABLE



```
CREATE TABLE [Hotspots.Review] (  
    rvwId CHAR (3) NOT NULL,  
    rstId CHAR (3),  
    cusId CHAR (3),  
    srcId CHAR (3),  
    rvwStar FLOAT,  
    rvwDate DATE,  
    rvwPhotoCount INTEGER,  
    CONSTRAINT pk_Review_rvwId PRIMARY KEY (rvwId),  
    CONSTRAINT fk_Review_rstId FOREIGN KEY (rstId)  
        REFERENCES [Hotspots.Restaurant] (rstId)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT fk_Review_cusId FOREIGN KEY (cusId)  
        REFERENCES [Hotspots.Customer] (cusId)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT fk_Review_srcId FOREIGN KEY (srcId)  
        REFERENCES [Hotspots.Source] (srcId)  
        ON DELETE CASCADE ON UPDATE CASCADE)
```




Which city has the most number of customers that posted a review?

```
CREATE VIEW NumOfCustomerPostedReview_V
AS
SELECT TOP(1) c.cusCity AS 'City Name', c.cusState AS 'State Name', COUNT(r.rvwId) AS 'Number of Customers that Posted a Review'
FROM [Hotspots.Review] r, [Hotspots.Customer] c
WHERE r.cusId = c.cusId AND c.cusCity IS NOT NULL
GROUP BY c.cusCity, c.cusState
ORDER BY 'Number of Customers that Posted a Review' DESC;
```


Results

| | City Name | State Name | Number of Customers that Posted a Review |
|---|------------|------------|--|
| 1 | Washington | DC | 7 |



What PriceLevel is the most popular
among all restaurants?

```
CREATE VIEW NumOfRestaurantbyPriceLevel_V  
AS  
SELECT TOP(1) r.rstPriceLevel, COUNT(r.rstId) as 'Number of Restaurant by PriceLevel'  
FROM [Hotspots.Restaurant] r  
GROUP BY r.rstPriceLevel  
ORDER BY 'Number of Restaurant by PriceLevel' DESC;
```


Results

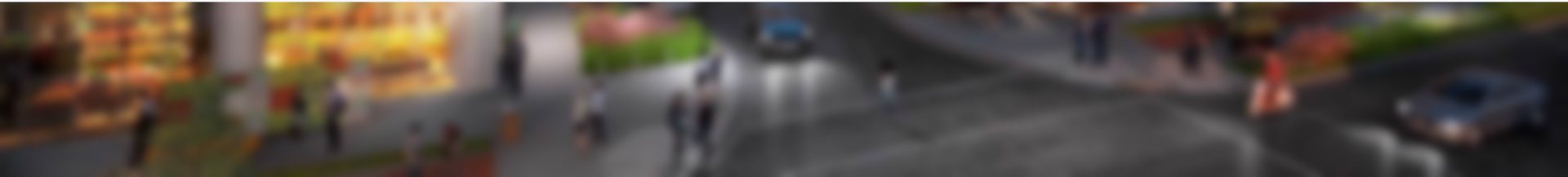
| | rstPriceLevel | Number of Restaurant by PriceLevel |
|---|---------------|------------------------------------|
| 1 | \$\$ | 10 |



Which restaurant, across all price levels, has the highest rating?

```
CREATE VIEW TopRestaurantBasedOnPrice_V  
AS
```

```
  Select r.rstId AS 'Restaurant ID', r.rstName AS 'Restaurant Name' , AVG(v.rvwStar) AS 'Average Rating' , r.rstPriceLevel AS 'Price Level'  
FROM [Hotspots.Restaurant] r  
    JOIN [Hotspots.Review] v  
      ON r.rstId = v.rstId  
WHERE r.rstId=  
      (SELECT TOP(1) re.rstId  
FROM [Hotspots.Restaurant] re JOIN [Hotspots.Review] rv  
  ON re.rstId = rv.rstId  
 WHERE re.rstPriceLevel=r.rstPriceLevel  
  GROUP BY re.rstId  
      )  
GROUP BY r.rstPriceLevel,r.rstId,r.rstName
```



Results

| | Restaurant ID | Restaurant Name | Average Rating | Price Level |
|---|---------------|--------------------|----------------|-------------|
| 1 | R01 | The Board and Brew | 3.6 | \$\$ |
| 2 | R06 | Panda Express | 3.4 | \$ |

What are the names of the top 3 restaurants, without considering their price level?

```
CREATE VIEW Top_3_Restaurants_V  
AS
```

```
SELECT TOP (3) r.rstName AS 'Restaurant Name' , AVG(v.rvwStar) AS 'Average Rating'  
FROM [Hotspots.Restaurant] r INNER JOIN [Hotspots.Review] v ON  
r.rstId = v.rstId  
GROUP BY r.rstName  
ORDER BY AVG(v.rvwStar) DESC;
```


Results

| | Restaurant Name | Average Rating |
|---|-------------------------------|----------------|
| 1 | Blaze Pizza | 4.6 |
| 2 | Silver Diner | 4.6 |
| 3 | Azteca Restaurant and Cantina | 4.4 |

Which restaurant has the highest opening hours?

```
CREATE VIEW LongestOperatingRestaurants_V
AS
SELECT TOP(1) r.rstName AS 'Restaurant Name', SUM(CASE
    WHEN (oprEnd - oprStart) % 100 != 0
    THEN FLOOR(oprEnd - oprStart) / 100 + 0.5
    WHEN oprEnd < oprStart
    THEN (oprEnd + 2400 - oprStart) / 100
    ELSE (oprEnd - oprStart) / 100 END ) AS 'Total Operating Hours in a Week'
FROM [Hotspots.OperationTime] o, [Hotspots.Restaurant] r
WHERE r.rstId = o.rstId
GROUP BY o.rstId, r.rstName
ORDER BY 'Total Operating Hours in a Week' DESC;
```


Results

| | Restaurant Name | Total Operating Hours in a Week |
|---|-----------------|---------------------------------|
| 1 | Silver Diner | 123.0 |