- **Design**

The “Restaurant” table has information regarding perspectives that customers are concerned about, such as price, address, restaurant open and close time, restaurant condition at weekends and its WiFi condition

The “City” table contains city information, like city population, major race, average income, and age.

The “Shuttle” table provides route information of UMD Shuttles, such as shuttle ID, shuttle schedule, from which users can find out the shuttle stop closest to each restaurant.

The “User” table records the basic information of users that leave their ratings, such as their cities and the number of their total reviews.

The “Stoplocation” table records the information of the name of shuttle stops, and also the location of the stops.

- Finalize ER schema and diagram.

Entity:

City (**ctyId**, ctyName, ctyPoppulation, ctyRace, ctyIncome, ctyAge)
Restaurant (**rstId**, rstName, rstPrice, rstStreet, rstZipcode, rstCity, rstState, rstPhoneNo, rstRating, rstCategory, rstReviewCount, rstLocation, -rstLatitude, -rstLongitude)
User (**usrId**, usrName)
Review (**rvwId**, rvwStar, rvwComment)
Shuttle (**shtId**, shtFrom, shtTo, shtWeekend, shtInterval)
StopLocation (**stpId**, stpName, stpLocation, -stpLatitude, -stpLongitude)

Relationship:

Locate: binary relationship
    1 Restaurant to 1 City
    1 City to 0 or more Restaurant
Pass: binary relationship
    1 Restaurant to 0 or more Shuttles
    1 Shuttle to 0 or more Restaurants
Comment With: binary relationship
    1 User to 1 or many Reviews
    1 Review to 1 User
Is Commented With: binary relationship

1 Review to 1 Restaurant

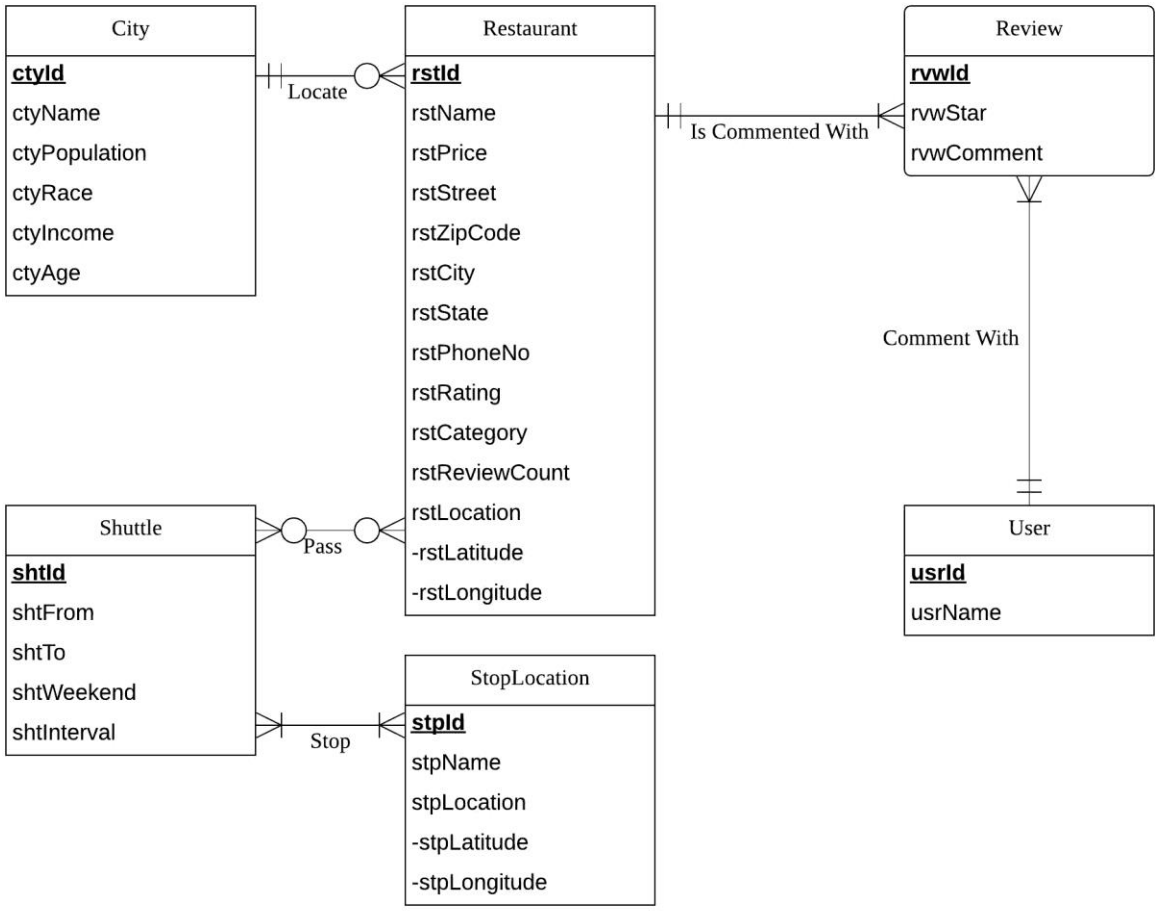
1 Restaurant to 1 or many Reviews

Stop: binary relationship

1 Shuttle to 1 or more StopLocations

1 StopLocations to 1 or more Shuttles

Of the



- Convert ER model into relational schema and identify primary and foreign keys.

Relations:

City (**ctyId**, ctyName, ctyPoppulation, ctyRace, ctyIncome, ctyAge)

Restaurant (**rstId**, rstName, rstPrice, rstStreet, rstZipcode, rstCity, rstState, rstPhoneNo, rstRating, rstCategory, rstReviewCount, rstLatitude, rstLongitude, *ctyId*)

User (**usrId**, usrName)

Review (**rvwId**, rvwStar, rvwComment, *rstId, usrId*)

Shuttle (**shtId**, shtFrom, shtTo, shtWeekend, shtInterval)
StopLocation (**stpId**, stpName, stpLatitude, stpLongitude)
Pass (***rstId***, ***shtId***)
Stop (***stpId***, ***shtId***)

Functional Dependency:

ctyId → ctyName, ctyPoppulation, ctyRace, ctyIncome, ctyAge
rstId → rstName, rstPrice, rstStreet, rstZipcode, rstCity, rstState, rstPhoneNo, rstRating, rstCategory, rstReviewCount, rstLatitude, rstLongitude, ctyId
usrId → usrId, usrName
rvwId → rvwStar, rvwComment, rstId, usrId
shtId → shtFrom, shtTo, shtWeekend, shtInterval
stpId → shtFrom, shtTo, shtWeekend, shtInterval
rstId, shtId →
stpId, shtId →

Normalizaton:

City (**ctyId**, ctyName, ctyPoppulation, ctyRace, ctyIncome, ctyAge) = 3NF
Restaurant (**rstId**, rstName, rstPrice, rstStreet, rstZipcode, rstCity, rstState, rstPhoneNo, rstRating, rstCategory, rstReviewCount, rstLatitude, rstLongitude, *ctyId*) = 3NF
User (**usrId**, usrName) = 3NF
Review (**rvwId**, rvwStar, rvwComment, *rstId, usrId*) = 3NF
Shuttle (**shtId**, shtFrom, shtTo, shtWeekend, shtInterval) = 3NF
StopLocation (**stpId**, stpName, stpLatitude, stpLongitude) = 3NF
Pass (***rstId***, ***shtId***) = 3NF
Stop (***stpId***, ***shtId***) = 3NF

- Generate business rules and determine referential integrity actions.

Business rules:

[R1] When a restaurant is located in a certain city, the information of the city cannot be deleted or changed.

[R2] When a restaurant is no longer in the database, the restaurant information should be deleted from the database.

[R3] When the information of a restaurant is changed in the database, the corresponding restaurant information should be changed accordingly.

[R4] When a shuttle is deleted from the database, the shuttle information should be deleted from the database.

[R5] When the information of a shuttle is changed in the database, the corresponding shuttle information should be changed accordingly.

[R6] When a user is deleted from the database, the user information should be deleted from the database.

[R7] When the information of a user is changed in the database, the corresponding user information should be changed accordingly.

[R8] When the information of a stop location is changed in the database, the corresponding stop information should be changed accordingly.

[R9] When the information of a shuttle is changed in the database, the corresponding shuttle information should be changed accordingly.

[R10] When the information of a shuttle and its stops is stored in the database, the information cannot be deleted.

Referential integrity:

Relation	Foreign Key	Base Relation	Primary Key	Business Rule	Constraint: ON DELETE	Business Rule	Constraint: ON UPDATE
Restaurant	ctyId	City	ctyId	R1	NO ACTION	R1	NO ACTION
Pass	rstId	Restaurant	rstId	R2	CASCADE	R3	CASCADE
Pass	shtId	Shuttle	shtId	R4	CASCADE	R5	CASCADE
Review	rstId	Restaurant	rstId	R2	CASCADE	R3	CASCADE
Review	usrId	User	usrId	R6	CASCADE	R7	CASCADE
Stop	stpId	StopLocation	stpId	R10	NO ACTION	R8	CASCADE

Stop	shtId	Shuttle	shtId	R10	NO ACTION	R9	CASCADE

- Describe sample data for every relation.

City ('C0005','Silver Spring',79483,'White',76608,34.6)

Restaurant ('R0003','Abol',2,'8626 Colesville Rd','20910','Silver Spring','MD','3016500061',4,'ethiopian',189,38.99674,-77.02771,'C0005')

User ('U0003','Hiral D.')

Review ('V0003',4,'Very tasty Ethiopian food and nice people running the restaurant. Its a bit pricier than other Ethiopian restaurants Ive been to in the area, but I...','R0003','U0003')

Shuttle ('B0111','Stamp Student Union Lot HH','Regents Drive Garage','NO','15')

StopLocation ('S0069','Regents Drive Garage',38.989522,-76.940772)

Pass ('R0003','B0111')

Stop ('S0069','B0111')