Problem 1 (2.5 Points):

Write the appropriate relational algebra that maps to the following functionality:

a) Find all users whose name is Bob

$\sigma_{\text{Name = 'Bob'}}^{\text{(Users)}}$

b) Find all business categorized as "Mexican"

$\sigma_{\text{Categories = 'Mexican'}}^{\text{(Businesses)}}$

c) Find the UserID of all users that reviewed Taco bell.

$\pi_{\text{UserID}}\ ((\sigma_{\text{Name = 'Taco Bell'}}^{\text{(Businesses)}})\text{Reviews} \bowtie_{\text{BusinessID = BusinessID}}\ )$

d) Find the UserID of all users that rated Mexican restaurants

$\pi_{\text{UserID}}\ ((\sigma_{\text{Categories = 'Mexican'}}^{\text{(Businesses)}})\text{Reviews} \bowtie_{\text{BusinessID = BusinessID}}\ )$

e) Find all users who joined Yelp in 2018 and rated the Pita Jungle restaurant.

$\pi_{\text{UserID,Name}}\ (\sigma_{\text{Year(Yelping Since)}} = \text{'2018'(Users)(Reviews} \bowtie \sigma_{\text{Name = 'Pita Jungle'}}\text{(Businesses))}$

f) Find two different relational algebra ways to: Find all restaurants rated by users that joined Yelp in either Year 2016 or 2018.

$\pi_{\text{BusinessID,Name}}\ \ (\sigma_{\text{Year(Yelping Since)}} = \text{'2018'(Users)(Reviews} \bowtie \text{Businesses)) U (}\sigma_{\text{Year(Yelping Since)}} = \text{'2016'(Users)(Reviews} \bowtie \text{Businesses))}$

g) Find users who have rated every single business except Chinese restaurants

$\pi_{\text{UserID,Name}}\ (\text{Reviews} \bowtie_{\text{BusinessID = BusinessID}}\ (\sigma_{\text{Categories} \neq \text{'Chinese'}}^{\text{(Businesses)}})$

Problem 2 (2.5 Points): Write the appropriate SQL that maps to the following functionality

a) For (parts (e), (f), and (g)) in Problem 1, find the suitable SQL query that perform the same functionality

Part e:

```
SELECT a.Name, a.UserID FROM Users a

INNER JOIN Reviews r ON a.UserID = r.UserID

INNER JOIN Businesses k ON k.BusinessID = r.BusinessID

WHERE EXTRACT (year FROM Yelping_since) = 2018 AND k.Name = 'Pita Jungle';
```

Part f:

```
SELECT a.Name, a.UserID FROM Users a

INNER JOIN Reviews r ON a.UserID = r.UserID

WHERE EXTRACT (year FROM Yelping_since) = 2018 or 2016;
```

Part g:

```
SELECT a.Name, a.UserID FROM Users a

INNER JOIN Reviews r ON a.UserID = r.UserID

INNER JOIN Businesses k ON k.BusinessID = r.BusinessID

WHERE EXTRACT k.Categories NOT LIKE %CHINESE%;
```

b) Write a SQL query that performs the following

a. Find all users who joined Yelp between year 2016 and 2018

```
SELECT a.Name, a.UserID FROM Users a

WHERE EXTRACT (year FROM a.Yelping_since)

BETWEEN 2016 AND 2018;
```

b. Find UserID’s of users who have rate Mexican or Chinese restaurants.

```
SELECT r.Name FROM Reviews r

INNER JOIN Businesses b

ON b.BusinessID = r.BusinessID

WHERE b.Categories LIKE %CHINESE% OR b.Categories LIKE %MEXICAN%;
```

c. Find two different ways to write a SQL query that: Retrieves IDs of all BurgerKing rated by users who joined Yelp in 2018.

Method A

```
SELECT k.BusinessID FROM Reviews k

INNER JOIN Businesses a ON k.BusinessID = a.UserID

INNER JOIN Users u ON k.UserID = u.UserID

WHERE a.Name = ‘Burger King’

AND EXTRACT (year FROM u.Yelping_Since) = 2018;
```

Method B

```
SELECT k.BusinessID FROM Reviews k, b.BusinessID FROM Businesses b, Users u
WHERE ((k.BusinessID = b.BusinessID)
AND(b.Name = ‘Burger King’)
AND( EXTRACT ( year FROM u.Yelping_Since) = 2018));
```

d. Find taco restaurants whose average review is greater than 4.0.

```
SELECT b.BusinessID, b.NAME from Businesses b
INNER JOIN Reviews r ON b.BusinessID = r.BusinessID
WHERE avg(r.Reviews) > 4.0);
```