# Project Outline

### **Course: COEN-380 Advanced Database Systems, Spring 2017**

### **Topic: SQL on Yelp dataset**

#### **Group No: 2**

#### **Group Members:**

Ashvita Hadge (W1270748)

Chetan Pangam (W1282207)

Sai Lalitha Sree Vuddamarry (W1270081)

Snehal Patil (W1159432)

### **Yelp dataset:**

The project will be completed using Yelp academic dataset. It is available online for student projects. The size of the dataset is small(around 1 GB). It has three tables

1. User(user\_id, name, review\_count, yelping\_since, friends\_count, fans, average\_stars)
2. Review( review\_id, user\_id, business\_id, stars, date\_of\_review, votes)
3. Business( business\_id, name, city, state, stars, review\_count, open, longitude, latitude, neighborhoods)

### **Queries:**

**(Simple query on Single table having predicates with special operator LIKE)**

1. Find every business in WA that has the word “Pizza‘ in its name.

**Ans:**

SELECT B.BUSINESS\_ID, B.NAME

FROM BUSINESS B

WHERE B.STATE=’WI’

AND B.NAME LIKE ‘%PIZZA%’;

**(Single table query with predicates and group by clause)**

2. Get the businesses who have a 5 star rating and are still in business. The results should be sorted by review counts in descending order. Return the top 10 businesses, the business ID, name, review count and the address.

**Ans:**

SELECT \* FROM

(SELECT COUNT(\*) AS REVIEW\_COUNT, B.BUSINESS\_ID, B.NAME

FROM BUSINESS B

WHERE B.STARS=5 AND B.OPEN=1

GROUP BY B.BUSINESS\_ID, B.NAME

ORDER BY 1 DESC)

WHERE ROWNUM <=10;

**(Single table query having two way join SUBQUERY with NOT IN operator)**

3. Find the user who never reviewed any business in Santa Clara, CA.

**Ans:**

SELECT U.USER\_ID, U.NAME

FROM USERS U

WHERE U.USER\_ID NOT IN( SELECT R.USER\_ID FROM REVIEW R, BUSINESS B

WHERE R.BUSINESS\_ID = B.BUSINESS\_ID AND B.CITY =’Madison’ AND B.STATE=’WI’) ;

**(Two way join having predicates)**

4. Find the businesses who have been given a review by elite users between the dates “1-JAN-2006” & “31-DEC-2010” and the business is 3 star. Elite users are the one’s who has review count more than 10.

**Ans**:

SELECT BUSINESS\_ID, NAME

FROM BUSINESS B, review r

WHERE r.BUSINESS\_ID=B.BUSINESS\_ID

AND b.review\_count >=30

AND r.date\_of\_review BETWEEN ‘1-JAN-2006’ AND ‘31-DEC-2016’

AND B.STARS =3;

**(Three way join with predicates on each table and order by clause)**

5. Give the user name & ID of top 10 users order by review count who have found the businesses in WI, friends count is greater than 100 and votes more than 10.

**Ans**:

select \* from(

select distinct u.user\_id,u.NAME,u.review\_count

from users u, review r,business b

where r.user\_id=u.user\_id

and b.business\_id=r.business\_id

and b.state='WI'

and u.FRIENDS\_COUNT >100

and r.VOTES>10

order by u.review\_count desc) where rownum <=10;

**(Three way join with oracle’s in built function EXTRACT)**

6. Get the businesses in Madison having 5 starts, that have been reviewed by more than 10 ANCIENT users. ANCIENT users are those who have been yelping till 2010. Return business ID, name, review count, Average rating for particular business.

Ans:

select r.business\_id,b.NAME from business b, users u, review r where b.city='Madison' and r.stars=5 and extract(year from u.yelping\_since) <= 2010 and b.business\_id=r.business\_id and r.user\_id=u.user\_id group by r.business\_id,b.name having count(r.user\_id)>10;

**(Three way join having INLINE VIEW with group by and having clauses)**

7. List all “5 stars” business that have been reviewed by any user who has been yelping for 10 to 15 years. 5 star businesses are the one’s who have average rating of 5.

**Ans**:

SELECT B.BUSINESS\_ID,ROUND(AVG(R.STAR)) AVG\_RATING FROM BUSINESS B, REVIEW R, (SELECT USERID FROM USERS WHERE ((SYSDATE-YELPING\_SINCE)/365) BETWEEN 10 AND 15 ) U WHERE R.BUSINESS\_ID=B.BUSINESS\_ID AND R.USERID=U.USERID GROUP BY R.BUSINESS\_ID HAVING ROUND(AVG(R.STAR)) =5;

8 Find the average rating across all reviews written by a particular user.

Ans:

SELECT AVG(R1.stars) AS AVG\_RATING,u.user\_id AS REVIEW\_USER FROM USERs u, REVIEWS R1 WHERE R1.user\_id=u.user\_id GROUP BY u.user\_ID ORDER BY AVG(R1.stars) DESC;