**SQL Functions**

1. Introduction

select \* from recent\_grads limit 10;

1. Conditional Statements

select Major, ShareWomen from recent\_grads

where ShareWomen < 0.5;

select Major, major\_category, Median, ShareWomen from recent\_grads

where ShareWomen > 0.5 and Median > 50000;

select major, median, Unemployed from recent\_grads

where Median >= 10000 or Unemployed <= 1000

limit 20;

select Major, Major\_category, ShareWomen, Unemployment\_rate from recent\_grads

where (Major\_category = 'Engineering') and (ShareWomen > 0.5 or Unemployment\_rate < 0.051);

1. Ordering the data

select Major, ShareWomen, Unemployment\_rate from recent\_grads

where ShareWomen > 0.3 and Unemployment\_rate < 0.1

order by ShareWomen desc;

select Major\_category, Major, Unemployment\_rate from recent\_grads

where Major\_category = 'Engineering' or Major\_category = 'Physical Sciences'

order by Unemployment\_rate;

1. Count statistics

select COUNT(Major) from recent\_grads where ShareWomen < 0.5;

select Major, Major\_category, MIN(Median) from recent\_grads where Major\_category = 'Engineering';

1. Sum statistics

select SUM(Total) from recent\_grads;

select AVG(Total), MIN(Men), MAX(Women) from recent\_grads;

1. Naming resulting columns

select COUNT(Major) as 'Number of Students', MAX(Unemployment\_rate) as 'Highest Unemployment Rate' from recent\_grads;

select COUNT(DISTINCT(Major)) as 'unique\_majors', COUNT(DISTINCT(Major\_category)) as 'unique\_major\_categories', COUNT(DISTINCT(Major\_code)) as 'unique\_major\_codes' from recent\_grads;

1. Quartile spread

SELECT Major, Major\_category, P75th - P25th as 'quartile\_spread' FROM recent\_grads order by quartile\_spread LIMIT 20;

1. Group by statement

select Major\_category, AVG(ShareWomen) from recent\_grads

Group by Major\_category;

select Major\_category, AVG(Employed)/ AVG(Total) as share\_employed

from recent\_grads Group by Major\_category;

select Major\_category, AVG(Low\_wage\_jobs)/ AVG(Total) as share\_low\_wage

from recent\_grads Group by Major\_category

having share\_low\_wage > .1;

1. Round function

select Round(ShareWomen, 4), Major\_category from recent\_grads limit 10;

select Major\_category, Round(AVG(College\_jobs)/ AVG(Total),3) as share\_degree\_jobs from recent\_grads group by Major\_category having share\_degree\_jobs < .3;

SELECT Major\_category, Cast(SUM(Women) as Float)/Cast(SUM(Total) as Float) SW FROM recent\_grads GROUP BY Major\_category ORDER BY SW

1. Subqueries

select Major, Unemployment\_rate from recent\_grads

where Unemployment\_rate < (select AVG(Unemployment\_rate) from recent\_grads)

order by Unemployment\_rate;

# calculation in subqueries

select CAST(COUNT(\*) as float)/CAST((select COUNT(\*) from recent\_grads)as float) as proportion\_abv\_avg from recent\_grads where ShareWomen > (SELECT AVG(ShareWomen) FROM recent\_grads)

1. Ordering by sum

SELECT Major\_category FROM recent\_grads

GROUP BY Major\_category

ORDER BY SUM(Total)

LIMIT 5)

1. Subqueries for top columns

select Major, Major\_category from recent\_grads

where Major\_category in (select Major\_Category from recent\_grads group by Major\_category order by SUM(Total) limit 5);

Ratio column with category

select AVG(CAST(Sample\_size as float)/CAST(Total as float)) as avg\_ratio from recent\_grads;

select Major, Major\_category,CAST(Sample\_size as float)/CAST(Total as float) as ratio from recent\_grads where ratio > (select AVG(CAST(Sample\_size as float)/CAST(Total as float)) as avg\_ratio from recent\_grads)

1