**SIMILAR LIKE ~ & REGEXP**

SIMILAR is used to search for simple string matches. Match any customers whose name begins with M

SELECT first\_name, last\_name

FROM customer

WHERE first\_name SIMILAR TO 'M%';

**% matches for zero or more characters**

**\_ Matches any single character**

We will check if there is an Ashley with 5 \_

SELECT first\_name, last\_name

FROM customer

WHERE first\_name LIKE 'A\_\_\_\_\_';

Return all customers whose 1st name begins with D, or whose last name ends with an n

SELECT first\_name, last\_name

FROM customer

WHERE first\_name SIMILAR TO 'D%' OR last\_name SIMILAR TO '%n';

**REGEXP SLIDE**

REGEXP is used to search for complex patterns using regular expressions. Match 1st name that starts with Ma using the match operator

SELECT first\_name, last\_name

FROM customer

WHERE first\_name ~ '^Ma';

Match names that end with ez

SELECT first\_name, last\_name

FROM customer

WHERE last\_name ~ 'ez$';

Match last names that end with ez or son

SELECT first\_name, last\_name

FROM customer

WHERE last\_name ~ 'ez|son';

Last names that contain w, x, y, or z

SELECT first\_name, last\_name

FROM customer

WHERE last\_name ~ '[w-z]';

**SUMMARIZING RESULTS**

**GROUP BY** defines how the results are grouped. COUNT returns the total number of records that match.

We'll use GROUP BY to return a single row for each unique value. How many customers have birthdays in certain months

SELECT EXTRACT(MONTH FROM birth\_date) AS Month, COUNT(\*) AS Amount

FROM customer

GROUP BY Month

ORDER BY Month;

**HAVING** narrows the results based on a condition. Let's only get months if more than 1 person has a birthday that month

SELECT EXTRACT(MONTH FROM birth\_date) AS Month, COUNT(\*)

FROM customer

GROUP BY Month

HAVING COUNT(\*) > 1

ORDER BY Month;

**AGGREGATE FUNCTIONS**

Aggregate functions return a single value from multiple parameters. For example sum all our inventory

SELECT SUM(price)

FROM item;

Get count, sum, min, max and average value of our items

SELECT COUNT(\*) AS Items,

SUM(price) AS Value,

ROUND(AVG(price), 2) AS Avg,

MIN(price) AS Min,

MAX(price) AS Max

FROM item;