**FUNCTIONS** (stored program that you can pass parameters into)

(to return a value)

SELECT **COUNT**(amount) AS count -- (count all amount in an entire column)

SELECT COUNT (amount) **AS** "today's transactions" -- (AS or an alias into a column)

SELECT **MAX**(amount) AS maximum -- (largest amount)

SELECT **MIN**(amount) AS minimum --(ex. smallest order)

SELECT **AVG**(amount) AS average

SELECT **SUM**(amount) AS sum -- (sum of an entire column)

FROM transactions;

**CONCAT** function -- (concatenate values together or joining 2 values as one)

SELECT CONCAT(first\_name, " " ,last\_name) AS full\_name

FROM employees;

**LOGICAL OPERATORS**

SELECT \*

FROM employees

WHERE hire\_date < "2023-01-5" AND job = "cook";

WHERE job = "cook" OR job = "cashier";

WHERE NOT job = "manager"; (opposite)

WHERE NOT job = "manager" AND NOT job = "asst. manager";

WHERE hire\_date BETWEEN "2023-01-04" AND "2023-01-07";

WHERE job IN ("cook", "cashier", "janitor");

**WILD CARD CHARACTERS %** \_ (-- comment)

SELECT \* FROM employees (% represents any num. of random characters)

WHERE first\_name LIKE "s%"; (s% begins with letter S)

WHERE hire\_date LIKE "2023%";

WHERE last\_name LIKE "%r"; (ends with letter R)

WHERE job LIKE "\_ook"; (\_ represents one random character)

WHERE hire\_date LIKE "\_\_\_\_-01-\_\_"; (finding only the month)

WHERE job LIKE "\_a%"; (finding values with second letter A)

**ORDER BY CLAUSE** (ascending or descending order) (default ascending)

SELECT \* FROM employees

ORDER BY last\_name **DESC**; (descending) (**ASC** asending)

SELECT \* FROM transactions

**ORDER BY** amount, customer id; (if one has same values)

**LIMIT** clause - is used to limit the number of records.

SELECT \* FROM customers

LIMIT 4; (limit rows to 4)

ORDER BY last name DESC LIMIT 4; (combined limiting order)

LIMIT 3, 1; -- (3 is the offset(laktaw?))

**UNION** combines the results of two or more SELECT statements

SELECT \* FROM income

UNION (union doesn't allow duplicates)

SELECT \* FROM expenses;

SELECT first\_name, last\_name FROM employees

UNION (if the one table has not enough columns)

SELECT first name, last name FROM customers;

**UNION ALL** (includes duplicates)

**SELF JOIN**

-join another copy of a table to itself

-used to compare rows of the same table

-helps to display a heirarchy of data

ALTER TABLE customers

ADD referral id INT;

UPDATE customers

SET referral\_id =1

WHERE customer\_id =2;

SELECT \* FROM customers;

>

SELECT a.customer id, a. first name, a.last\_name,

CONCAT(b. first name," ", b.last name) AS "reffered\_by"

FROM customers AS a

LEFT JOIN customers AS b

ON a.referral\_id = b.customer\_id; (refferal by name)

>

UPDATE employees

SET supervisor\_id = 5

WHERE employee\_id = 6; (by ID)

SELECT \* FROM employees;

>

SELECT a.first\_name, a.last\_name

CONCAT (b.first\_name, " ", b.last\_name) AS "reports to

FROM employees AS a

INNER JOIN employees AS b

ON a.supervisor id = b. employee\_id;

**Views**

a virtual table based on the result-set of an SQL statement

The fields in a view are fields from one or more real tables in the database

They're not real tables, but can be interacted with as if they were

>

**CREATE VIEW** employee\_attendance AS -- (creating a view)

SELECT first name, last name

FROM employees;

SELECT \* FROM employee\_attendance;

ORDER BY last\_name ASC;