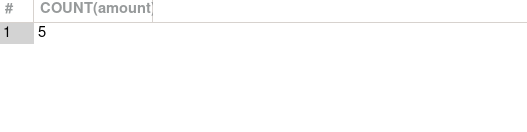
// Functions

// count()

SELECT COUNT(amount)

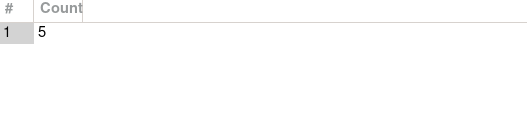
FROM transactions;



// now we named the function with the name count

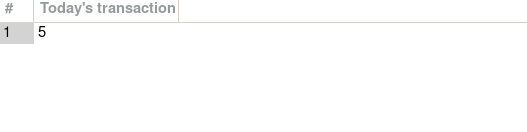
SELECT COUNT(amount) AS Count

FROM transactions;



SELECT COUNT(amount) AS "Today's transaction"

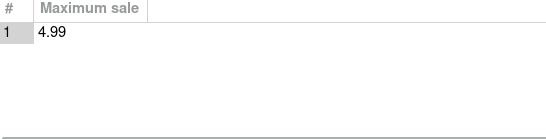
FROM transactions;



// MAX()

SELECT MAX(amount) AS "Maximum sale"

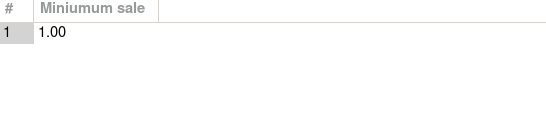
FROM transactions;



// MIN()

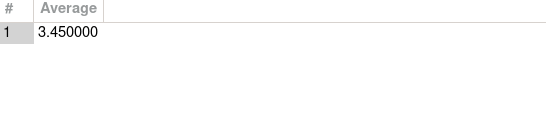
SELECT MIN(amount) AS "Miniumum sale"

FROM transactions;

// Average()

SELECT AVG(amount) AS "Average"

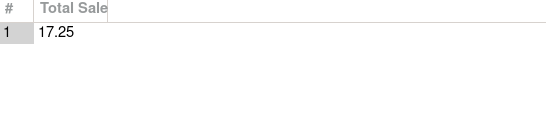
FROM transactions;



// SUM()

SELECT SUM(amount) AS "Total Sale"

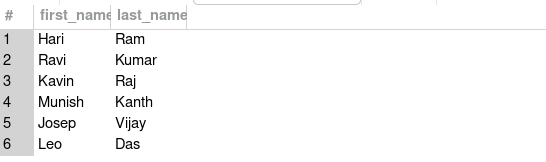
FROM transactions;



// CONCAT()

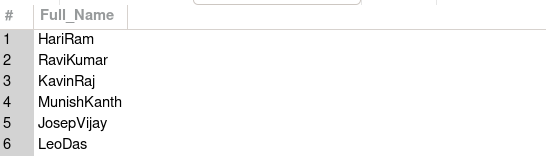
SELECT first\_name, last\_name

FROM employees;



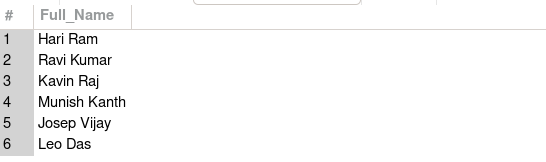
SELECT CONCAT(first\_name, last\_name) as Full\_Name

FROM employees;



SELECT CONCAT(first\_name, " ", last\_name) as Full\_Name

FROM employees;



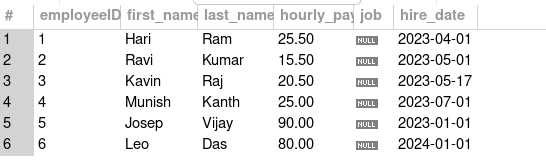
// **OPERATORS**

lets add a column job to employees table

ALTER TABLE employees

ADD COLUMN job VARCHAR(25) AFTER hourly\_pay;

SELECT \* FROM employees;



// UPDATE the job details

UPDATE employees

SET job = "cashier"

WHERE employeeID = 1;

SELECT \* FROM employees;

--> this will throw an error that you are using sql in safe mode so you cant update, to over came this we use this below command

// SET SQL\_SAFE\_UPDATES = 0;

// now update the values

UPDATE employees

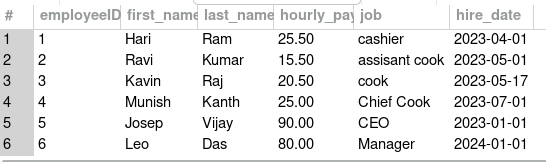
SET job = "cashier"

WHERE employeeID = 1;

SELECT \* FROM employees;

--> value updated successful

// After updating all details



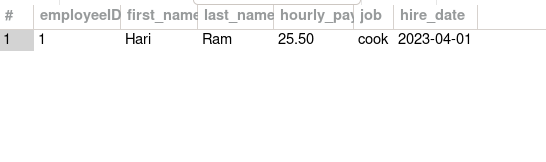
// AND OPERATOR

// select the cooks whose hire date date is befor 2023-05-01

SELECT \*

FROM employees

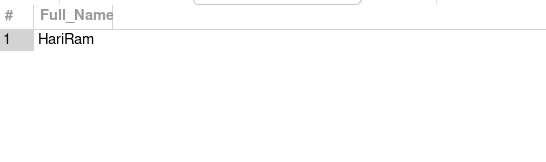
WHERE hire\_date < "2023-05-01" AND job = "cook";



SELECT CONCAT(first\_name, last\_name) AS Full\_Name

FROM employees

WHERE hire\_date < "2023-05-01" AND job = "cook";



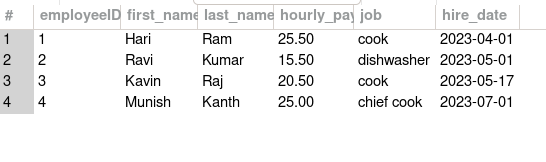
// NOT OPERATOR

// select the persons who are not at the managerial level

SELECT \*

FROM employees

WHERE NOT job = "manager" AND NOT job = "ceo";

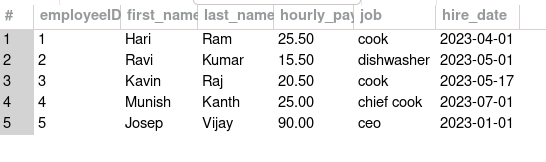


// BETWEEN OPERATOR

SELECT \*

FROM employees

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

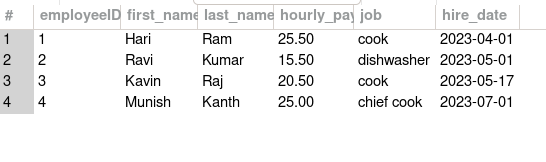


// IN OPERATOR

SELECT \*

FROM employees

WHERE job IN ("cook", "dishwasher", "chief cook");

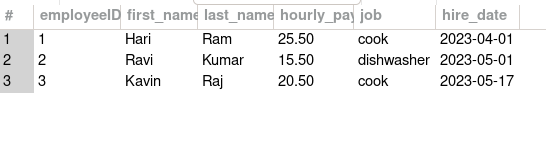


// OR OPERATOR

SELECT \*

FROM employees

WHERE job = "cook" OR job = "dishwasher";



// **WILD CARD CHARACTERS** ( %,\_) using LIKE

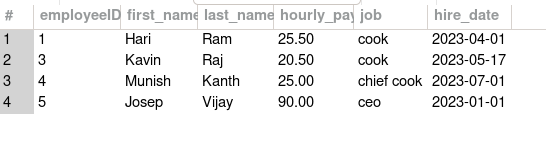
// This is used to match the contents present in the table

// eg., if you want to view all the jobs whose name starts with ‘c’

SELECT \*

FROM employees

WHERE job LIKE "c%";



// % is used to match the contents that starts or ends with

// while ‘\_’ is used to match the exact pattern

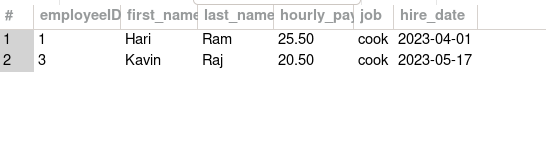
// eg., to find the jobs with has the patter a random character followed by ook

// \_ook --> cook

SELECT \*

FROM employees

WHERE job LIKE "\_ook";



// what if “\_ook\_” this will check for 5 letter words whose 1st and last position of random characters and middle one with “ook”.

// this will perform the task of persons who where hired on the 1st of the month

SELECT \*

FROM employees

WHERE hire\_date LIKE "\_\_\_\_-\_\_-01";

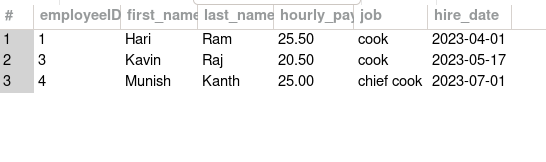


// select all the cooks

SELECT \*

FROM employees

WHERE job LIKE "%\_ook";



// **ORDER BY CLAUSE -->** This will give the table in format of asc r des order of the given column

// **ASC / DES**

// By Default the Order By will give you in asc order

//

SELECT \*

FROM employees

ORDER BY first\_name ASC;



SELECT \*

FROM employees

ORDER BY first\_name DESC;



SELECT \*

FROM employees

ORDER BY hire\_date ASC;



SELECT \*

FROM transactions

ORDER BY amount ASC;



// Incase if you order by based on 2 columns then you can do it

// this will work based on the if 2 or more rows having same amount, it will check for the 2nd columns asc or des order

eg

SELECT \*

FROM transactions

ORDER BY amount, customer\_id ASC;



SELECT \*

FROM transactions

ORDER BY amount ASC, customer\_id DESC;



// **LIMIT CLAUSE**