Absolutely! Here's your requested overview formatted consistently:

MySQL Aggregate Functions

Aggregate functions operate on multiple rows of data and return a single value.

Function	Description	Example Usage
`SUM()`	Returns the total sum of a numeric column.	`SELECT SUM(salary) FROM emp;`
`MAX()`	Returns the maximum value in a column.	`SELECT MAX(age) FROM emp;`
`MIN()`	Returns the minimum value in a column.	`SELECT MIN(salary) FROM emp;`
`AVG()`	Returns the average value of a numeric column.	`SELECT AVG(salary) FROM emp;`
`COUNT()`	Returns the number of rows or non-null values in a column.	`SELECT COUNT(*) FROM emp;`

Date and Time Functions

These functions help manipulate and retrieve date and time values.

Function	Description	Example Usage
`CURDATE()`	Returns the current date.	`SELECT CURDATE();`
`CURTIME()`	Returns the current time.	`SELECT CURTIME();`
`NOW()`	Returns the current date and time.	`SELECT NOW();`
`DATEDIFF()`	Returns the difference in days between two dates.	`SELECT DATEDIFF('2024-01-01', '2023-01-01');`

MySQL String Handling Functions

These functions manipulate string values.

Function	Description	Example Usage
`UPPER()`	Converts a string to uppercase.	`SELECT UPPER(name) FROM emp;`
`LOWER()`	Converts a string to lowercase.	`SELECT LOWER(name) FROM emp;`
`MID()`	Returns a substring from a string.	`SELECT MID(name, 1, 3) FROM emp;`
`REVERSE()`	Reverses a string.	`SELECT REVERSE(name) FROM emp;`
`CONCAT()`	Concatenates two or more strings.	`SELECT CONCAT(first_name, ' ', last_name) FROM emp;`
`LENGTH()`	Returns the length of a string.	`SELECT LENGTH(name) FROM emp;`

Window Functions

These functions allow performing calculations across a set of table rows related to the current row.

Function	Description	Example Usage
`DENSE_RANK()`	Assigns a rank to rows within a partition, with no gaps.	`SELECT DENSE_RANK() OVER (ORDER BY salary) AS rank FROM emp;`
`NTILE(n)`	Divides ordered rows into `n` buckets.	`SELECT NTILE(4) OVER (ORDER BY salary) AS quartile FROM emp;`
`RANK()`	Assigns a rank to rows, with gaps for ties.	`SELECT RANK() OVER (ORDER BY salary) AS rank FROM emp;`
`CUME_DIST()`	Calculates the cumulative distribution of a value in a group of values.	`SELECT CUME_DIST() OVER (ORDER BY salary) AS cume_dist FROM emp;`
`ROW_NUMBER()`	Assigns a unique sequential integer to rows within a partition.	`SELECT ROW_NUMBER() OVER (ORDER BY salary) AS row_num FROM emp;`

Additional Window Functions

These functions are often used for accessing data from the current row relative to other rows.

Function	Description	Example Usage
`FIRST_VALUE()`	Returns the first value in an ordered set.	`SELECT FIRST_VALUE(salary) OVER (ORDER BY id) AS first_salary FROM emp;`
`LAST_VALUE()`	Returns the last value in an ordered set.	`SELECT LAST_VALUE(salary) OVER (ORDER BY id ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS last_salary FROM emp;`
`LAG(value)`	Accesses data from a previous row in the same result set.	`SELECT salary, LAG(salary) OVER (ORDER BY id) AS previous_salary FROM emp;`
`LEAD(value)`	Accesses data from a subsequent row in the same result set.	`SELECT salary, LEAD(salary) OVER (ORDER BY id) AS next_salary FROM emp;`
`NTH_VALUE(value, n)`	Returns the value of a specific row in the ordered set.	`SELECT NTH_VALUE(salary, 2) OVER (ORDER BY id) AS second_salary FROM emp;`

Summary

These functions are essential for data analysis and manipulation within MySQL. Aggregate functions help summarize data, while window functions allow for advanced analytics without collapsing rows into a single output. Date and string functions facilitate working with various data types.

Let me know if you need any more details or examples on any specific functions!

In [ ]: