Exploratory Data Analysis

-- 1. Basic Queries

-- SELECT: The `SELECT` statement is used to retrieve data from one or more tables.

-- SELECT all columns

SELECT \* FROM customers LIMIT 5;

-- SELECT specific columns with ALIAS

SELECT

product\_name AS 'Product Name',

purchase\_price AS 'Cost Price'

FROM

products

LIMIT 10;



#Average Product price:

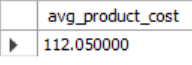
-- AVG

SELECT

AVG(purchase\_price) AS avg\_product\_cost

FROM

products;



-- Find customers who placed orders after 2023-06-01 AND have spent more than $500 in total:

SELECT

customer\_id,

SUM(total\_amount) AS total\_spent,

COUNT(order\_id) AS num\_orders

FROM

sales\_orders

WHERE

order\_date > '2023-06-01'

GROUP BY customer\_id

HAVING SUM(total\_amount) > 500.00; -- Filters groups AFTER aggregation



-- 4. Built-in Functions for Data Manipulation & Analysis

-- Date/Time Functions (MySQL Equivalents)

-- EXTRACT / DATE\_PART -> YEAR(), MONTH(), DAY(), HOUR(), EXTRACT()

SELECT

order\_date,

YEAR(order\_date) AS order\_year,

MONTH(order\_date) AS order\_month

FROM

sales\_orders

LIMIT 5;

SELECT

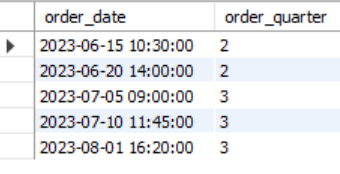
order\_date,

EXTRACT(QUARTER FROM order\_date) AS order\_quarter

FROM

sales\_orders

LIMIT 5;

  
  
-- Example: Calculate discount percentage, avoid division by zero if selling\_price is 0:

SELECT

order\_item\_id,

selling\_price,

discount\_amount,

(discount\_amount \* 100 / NULLIF(selling\_price, 0)) AS discount\_percentage

FROM

order\_items

LIMIT 10;

– -- TO\_CHAR -> DATE\_FORMAT()

SELECT

order\_date,

DATE\_FORMAT(order\_date, '%Y/%m/%d %H:%i') AS formatted\_date

FROM

sales\_orders

LIMIT 5;

-- TO\_DATE -> STR\_TO\_DATE()

SELECT STR\_TO\_DATE("2024-May-24", "%Y-%b-%d") as Date;

–SELECT

order\_id,

CONCAT('Order #', CAST(order\_id AS CHAR)) AS order\_label

FROM

sales\_orders

LIMIT 5;

-- POSITION / LOCATE

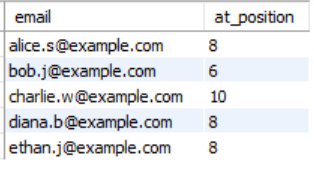
SELECT

email, LOCATE('@', email) AS at\_position

FROM

customers

LIMIT 5;



SELECT

order\_id,

CONCAT('Order #', CAST(order\_id AS CHAR)) AS order\_label

FROM

sales\_orders

LIMIT 5;

-- SUBSTRING

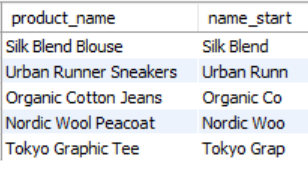
SELECT

product\_name, SUBSTRING(product\_name, 1, 10) AS name\_start

FROM

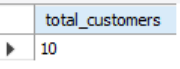
products

LIMIT 5;



-- Number of customers

SELECT COUNT(\*) AS total\_customers FROM customers;



-- Number of orders by status

SELECT status, COUNT(\*) FROM sales\_orders GROUP BY status;

-- Most popular product categories

SELECT c.category\_name, COUNT(\*) AS total\_orders

FROM order\_items oi

JOIN products p ON oi.product\_id = p.product\_id

JOIN categories c ON p.category\_id = c.category\_id

GROUP BY c.category\_name

ORDER BY total\_orders DESC;