Subqueries & DATE Manipulation



6 Session Objectives

- Understand the DATE data type in SQL (MySQL).
- Use SQL date functions to retrieve, analyze, and clean data.
- Apply date arithmetic and format transformations.
- Clean messy date columns for analysis.
- Solve real-world queries using date manipulation.

Correlated Subquery

Products that are returned more than the average return qty in their subcategory

```
-- CORRELATED SUBQUERIES.....
USE bike_analysis;
SELECT
   p.ProductKey,
   p.ProductName,
   r.returnQuantity
FROM Products p
JOIN returns r
ON p.ProductKey = r.ProductKey
WHERE r.returnQuantity > (
   SELECT
       AVG(r2.returnQuantity)
   FROM returns r2
   JOIN Products p2
    ON r2.ProductKey = p2.ProductKey
   WHERE p2.ProductSubcategoryKey = p.ProductSubcategoryKey
ORDER BY r.returnQuantity DESC;
```

ProductKey	ProductName	returnQuantity	
223	AWC Logo Cap	2	
352	Mountain-200 Silver, 38	2	
477	Water Bottle - 30 oz.	2	
477	Water Bottle - 30 oz.	2	productKey - AvgReturn
477	Water Bottle - 30 oz.	2	A CONTRACTOR OF THE PARTY OF TH
477	Water Bottle - 30 oz.	2	
477	Water Bottle - 30 oz.	2	
477	Water Bottle - 30 oz.	2	
478	Mountain Bottle Cage	2	
478	Mountain Bottle Cage	2	
478	Mountain Bottle Cage	2	
479	Road Bottle Cage	2	
480	Patch Kit/8 Patches	2	
480	Patch Kit/8 Patches	2	
480	Patch Kit/8 Patches	2	
481	Racing Socks, M	2	
528	Mountain Tire Tube	2	19 [summary table]
528	Mountain Tire Tube	2	The same of the sa
605	Road-750 Black, 48	2	

```
Product p
Return r
is producing the result
built on Dimension Table
then we apply filter with
the subquery where Avg is
being calculated

Products p2
Returns r2 -> AVG

Group By ProductSubcategory
```

Nested Subquery Inside Subquery

Return the region with maximum return value

→ Australia - 404

Region	total_return_quantity	
Australia	404	
Southwest	362	
Northwest	270	
Canada	238	
United Kingdom	204	
France	186	
Germany	163	
Southeast	1	

```
-- Return the region with maximum return value

SELECT

t.Region,
sub.total_return_quantity

FROM territories t

JOIN (
SELECT
r.TerritoryKey,
SUM(r.ReturnQuantity) AS total_return_quantity

FROM returns r
GROUP BY r.TerritoryKey
) sub

ON t.SalesTerritoryKey = sub.TerritoryKey;
```

```
Region total_return_quantity

Australia 404
```

```
-- Return the region with maximum return value
SELECT
   t.Region,
   sub.total_return_quantity
FROM territories t
JOIN .(.....
   SELECT
       r.TerritoryKey,
       SUM(r.ReturnQuantity) AS total_return_quantity
   FROM returns r
   GROUP BY r. TerritoryKey
ON t.SalesTerritoryKey = sub.TerritoryKey
WHERE sub.total_return_quantity =
(
   SELECT
       MAX(total_return_quantity)
   FROM.(....
       SELECT
           SUM(r2.ReturnQuantity) AS total_return_quantity
       FROM returns r2
       GROUP BY r2.TerritoryKey
   ) sub2
);
```

Finding the max from the summary table having total_return_quantity as a numeric column

DATE Manipulation

```
Date_Format
```

https://dev.mysql.com/doc/refman/8.4/en/date-and-time-functions.html

- AAAA-WW-DD
- 1000-01-01 Till 9999-12-31
- 3 bytes
- TODAY() returns current date,
- NOW() -> TIMESTAMP -> DATETIME

Field	Type	Null	Key	Default	Extra
event_id	int	NO	PRI	NULL	auto_increment
event_name	varchar(100)	YES		NULL	
event_date	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
delivery_date	date	YES		NULL	

```
INSERT INTO Events(event_name,event_date,delivery_date)
VALUES("audit", '2025-06-26 15:00:00', '2025-06-26');

INSERT INTO events (event_name, event_date, delivery_date)
VALUES
('Product Launch', '2025-07-15', '2025-07-10'),
('Annual Meetup', '2025-08-01', '2025-07-25'),
('Marketing Campaign', '2025-07-05', '2025-07-01'),
('Customer Webinar', '2025-06-30', '2025-06-25'),
('Software Deployment', '2025-07-20', '2025-07-18');

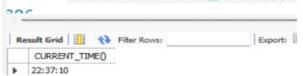
SELECT * FROM Events;
```

event_id	event_name	event_date	delivery_date
1	audit	2025-06-26 15:00:00	2025-06-26
2	Product Launch	2025-07-15 00:00:00	2025-07-10
3	Annual Meetup	2025-08-01 00:00:00	2025-07-25
4	Marketing Campaign	2025-07-05 00:00:00	2025-07-01
5	Customer Webinar	2025-06-30 00:00:00	2025-06-25
6	Software Deployment	2025-07-20 00:00:00	2025-07-18
NULL	NULL	NULL	NULL

INSERT INTO Events(event_name,event_date,delivery_date)
VALUES("Project X", CURRENT_TIMESTAMP() , CURRENT_DATE());

event_id	event_name	event_date	delivery_date
1	audit	2025-06-26 15:00:00	2025-06-26
2	Product Launch	2025-07-15 00:00:00	2025-07-10
3	Annual Meetup	2025-08-01 00:00:00	2025-07-25
4	Marketing Campaign	2025-07-05 00:00:00	2025-07-01
5	Customer Webinar	2025-06-30 00:00:00	2025-06-25
6	Software Deployment	2025-07-20 00:00:00	2025-07-18
7	Project X	2025-06-26 22:35:31	2025-06-26
NULL	NULL	NULL	NULL

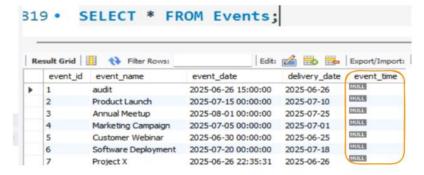
305 • SELECT CURRENT_TIME();



-- Add Event With Current Time
-- Add column Event_time

ALTER TABLE Events
ADD Column event_time TIME;
DESC Events;

Field	Туре	Null	Key	Default	Extra
event_id	int	NO	PRI	HULL	auto_increment
event_name	varchar(100)	YES		NULL	
event_date	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
delivery_date	date	YES		NULL	
event_time	time	YES		HULL	



321 • INSERT INTO Events(event_name, delivery_date, event_time)
322 VALUES('Team Sync', CURRENT_DATE(), CURRENT_TIME());

