NEXT DATA CRUNCH

DATABASE OBJECT DIFFERENCES

VIEWS, MATERIALIZED VIEWS,
TABLES, AND TEMPORARY TABLES

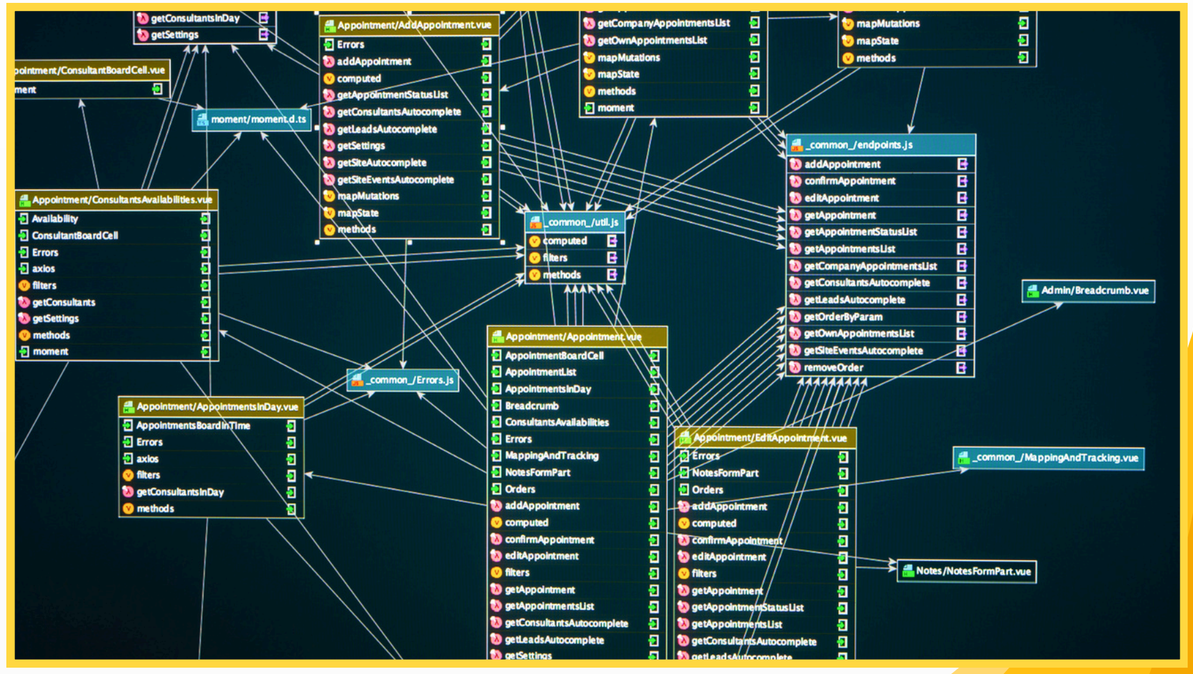
NUR AZMI PRASETYO

NEXT DATA CRUNCH

INTRO - UNRAVEL

- Databases are powerful tools for **managing and organizing data**. Different types of database objects allow us to **store**, **process**, and **query** data in a **variety** of ways.

- In this episode, we will explore **four key database objects** that are essential for data management and querying: **Views**, **Materialized Views**, **Temporary Tables**, and **Tables**.



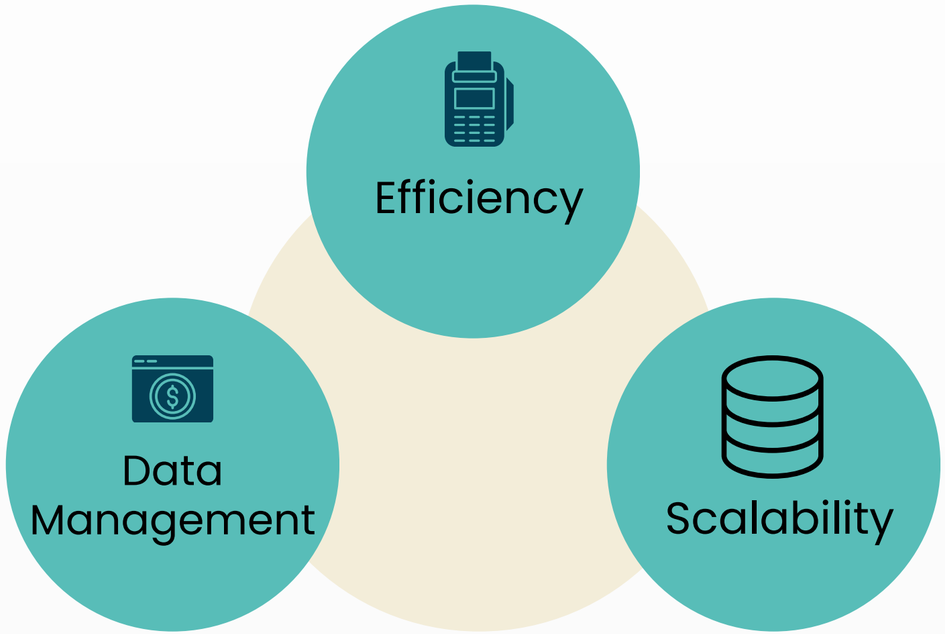
NEXT DATA CRUNCH

WHY DOES IT MATTER?

- Knowing the **right object** to use for different scenarios can **save time** and **resources**. For example, using a **materialized view** can speed up **complex queries** that don't need real-time data.

- Properly organizing data with views or temporary tables can **simplify** your **workflow** and make data **processing smoother**.

- As data grows, **optimizing queries** with the right objects ensures that your systems can handle **increased load** and **complexity**.



NEXT DATA CRUNCH

WHAT IS A VIEW?

A view is a **virtual table** that represents the result of a query. It **doesn't store data** but fetches data dynamically from underlying tables **when queried**.

```sql
CREATE VIEW CustomerSummary AS
SELECT
customer_id,
COUNT(order_id) AS total_orders,
SUM(order_value) AS total_spent
FROM Orders
GROUP BY customer_id;
```

Key Characteristics:

- Data **is not stored** but **queried on the fly** from underlying tables.
- Since the data is retrieved in real-time, it **reflects the latest changes** in the base tables.
- Complex operations can be stored as views, **reducing the need for repetitive code** in your queries.

When to use:

- When you need to hide complexity from end-users.
- When you want to **simplify repetitive queries.**
- For security reasons, when you want to **limit access** to specific columns or rows of a table.

NEXT DATA CRUNCH

MATERIALIZED VIEW (MV)

A materialized view **stores** the **result of a query physically.** This allows for **faster access** to the data since the query doesn't have to be executed every time.

```
Run on active connection | Select block
CREATE MATERIALIZED VIEW SalesSummary AS
SELECT product_id,
SUM(sales_amount) AS total_sales
FROM Sales
GROUP BY product_id;

-- Refresh the materialized view when necessary
REFRESH MATERIALIZED VIEW SalesSummary;
```

Key Characteristics:

- Materialized views **store the result of a query**, unlike regular views which only define a query.
- Querying a materialized view is **faster than querying a view.**
- Materialized views **don’t automatically update** when the base data changes, so they **must be refreshed periodically.**

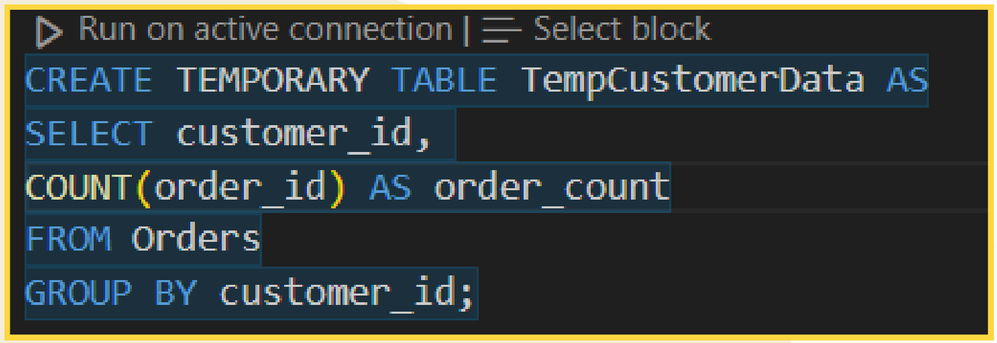
When to use:

- When you need to speed up complex queries that **don’t require real-time accuracy.**
- For read-heavy scenarios where **performance is a priority over fresh data.**

NEXT DATA CRUNCH

TEMPORARY TABLE

A temporary table **exists only during the session** or transaction in which it is created.



Key Characteristics:

- Data in temporary tables is **only available within the session** or transaction that created it.
- The database **automatically removes temporary tables when the session ends**, so there's no need for manual cleanup.
- Useful when you need to **store intermediate results** that don't need to persist.

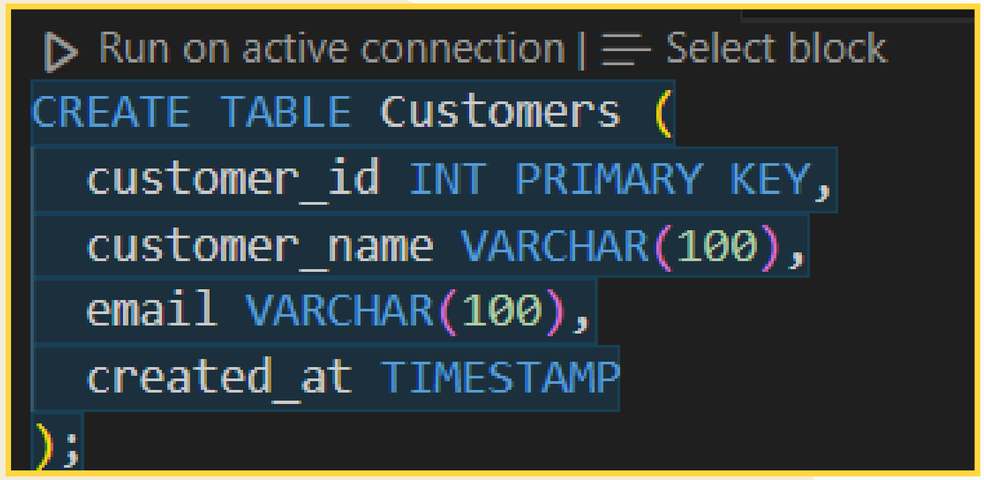
When to use:

- During data transformation processes where **intermediate results are required.**
- When performing **temporary calculations** or staging data for analysis.

NEXT DATA CRUNCH

WHAT IS A TABLE?

Table is the most **fundamental database object**, and **it stores data persistently.**



Key Characteristics:

- Data is **stored permanently** in a table until it is explicitly deleted.
- Tables **can be indexed** to improve performance during data retrieval.
- You can perform all **CRUD** (**Create**, **Read**, **Update**, **Delete**) **operations** on tables.

When to use:

- When you need to **store** large amounts of data that should **persist**.
- For operational or transactional systems that require **constant data updates**.

NEXT DATA CRUNCH

KEY DIFFERENCES SIDE-BY-SIDE COMPARISON

Feature	View	Materialized View	Temporary Table	Table
Data Storage	No (Virtual)	Yes (Physical)	No (Session-based)	Yes (Physical)
Performance	Low (real-time query)	High (precomputed)	Moderate (session-based)	High (persistent)
Data Refresh	Real-time	Periodic/Manual	None	N/A
Use Case	Simplifying queries	Speeding up complex queries	Session-based data storage	Long-term storage

NEXT DATA CRUNCH

CHOOSING THE RIGHT OBJECT FOR THE JOB

01

VIEWS

Use views when the data needs to be up-to-date and doesn't require complex computation or frequent refreshing.

02

MATERIALIZED VIEWS

Choose materialized views for queries that require expensive computations or aggregations but don't need real-time data.

03

TEMPORARY TABLES

Use temporary tables for storing intermediate data that should not persist.

04

TABLES

Tables are the foundation of data storage and should be used for permanent storage of transactional or historical data.

NEXT DATA CRUNCH

THANK YOU

Thank you for your attention today. I appreciate you taking the time to go through this deck.

Feel free to reach out for any questions, clarifications, or further discussions on database objects and data management strategies.