

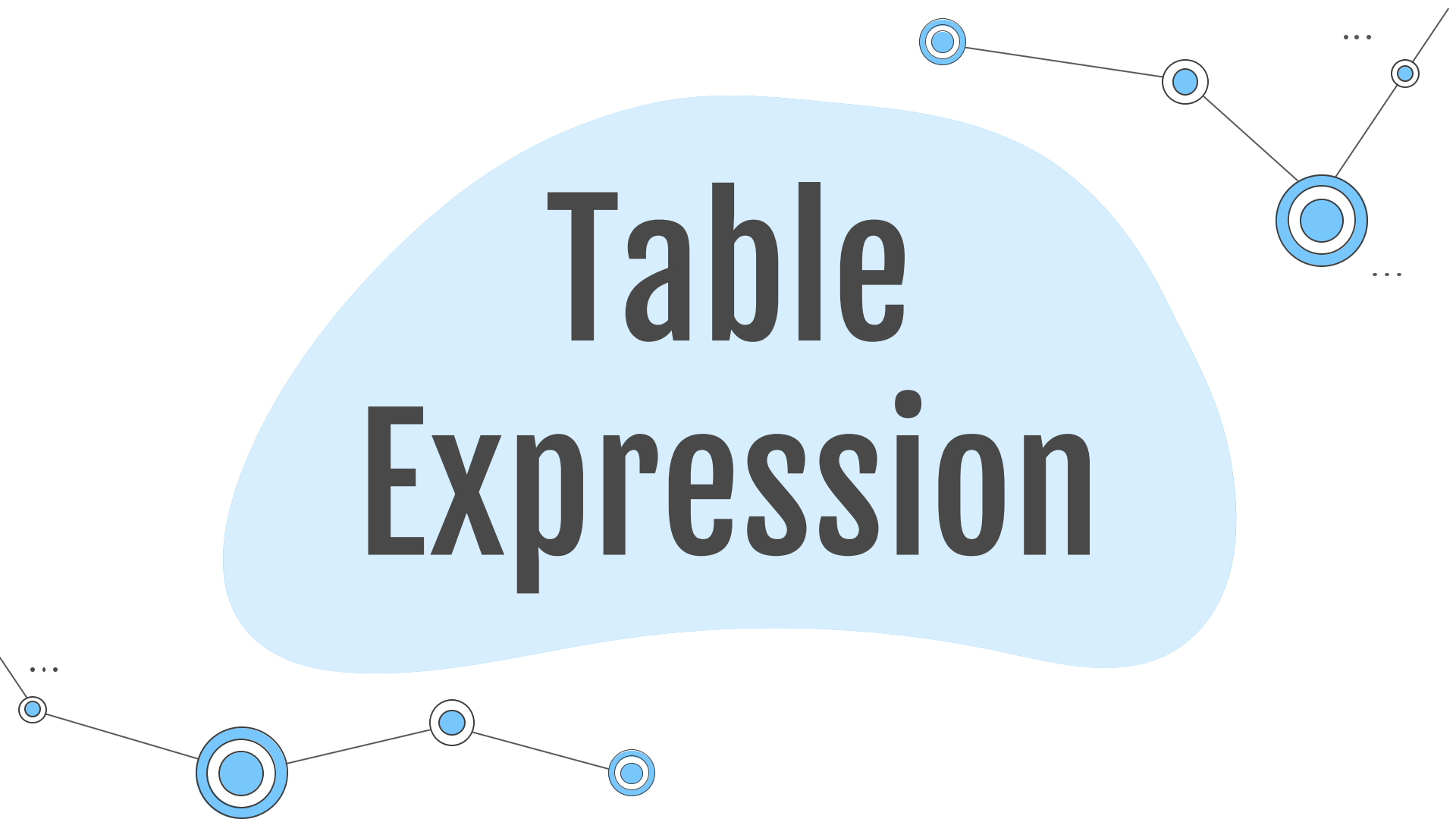
BASIS DATA LANJUT

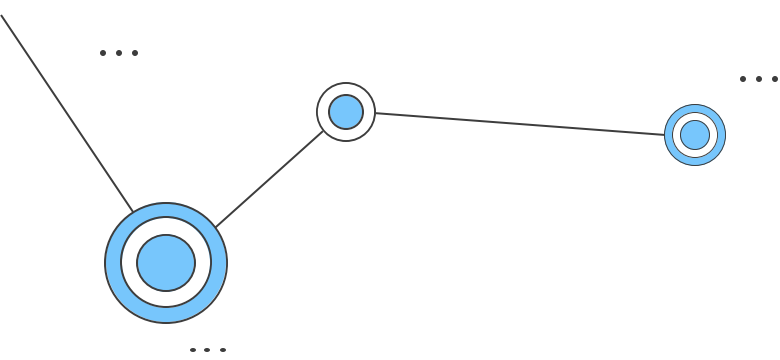
Pertemuan 6

Table Expression

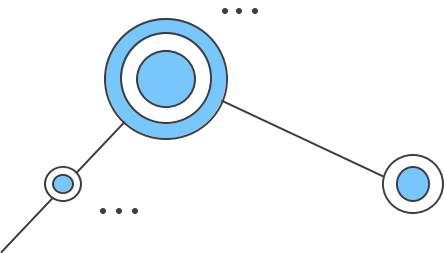
Team Teaching Basis Data Lanjut

Table Expression

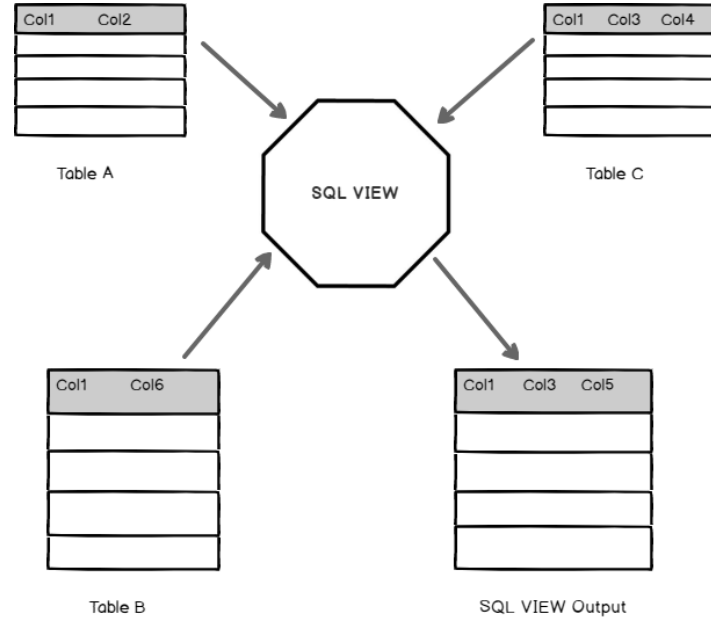




View adalah QUERY yang disimpan dalam database oleh administrator



MEMBUAT VIEW SEDERHANA



MEMBUAT VIEW SEDERHANA

- View adalah QUERY yang disimpan dalam database oleh administrator dan developer
- View didefinisikan dengan satu pernyataan SELECT
- ORDER BY tidak diizinkan dalam definisi View tanpa menggunakan TOP, OFFSET/FETCH, atau FOR XML
- Untuk menyortir output, gunakan ORDER BY di luar QUERY

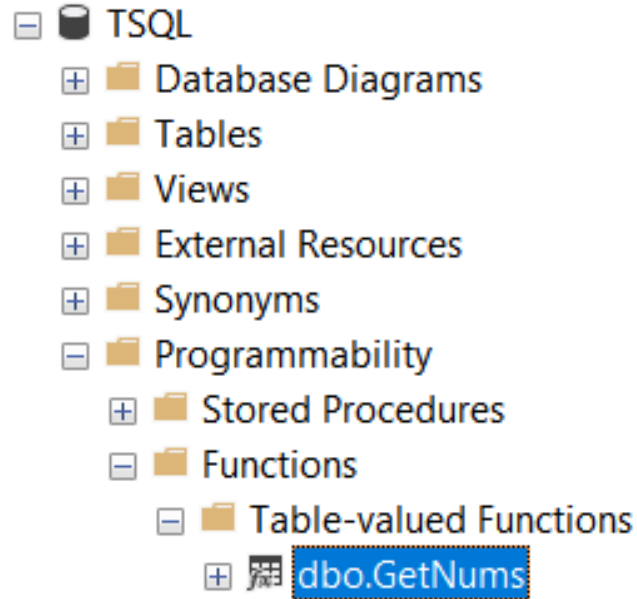
```
CREATE VIEW HumanResources.EmployeeList
AS
SELECT BusinessEntityID, JobTitle, HireDate, VacationHours
FROM HumanResources.Employee;

SELECT * FROM HumanResources.EmployeeList
```

MEMBUAT INLINE TABLE-VALUED FUNCTION

TABLE-VALUED
FUNCTIONS dibuat oleh
administrator dan
developers
TABLE VALUE FUNCTION.

Fungsi tersebut akan
**mengembalikan nilai berupa
table yang berisi record-
record dari hasil eksekusi
QUERY yang didefinisikan
didalam fungsi tersebut**



MEMBUAT INLINE TABLE-VALUED FUNCTION

Membuat dan beri nama fungsi dan parameter opsional dengan CREATE FUNCTION

Membuat dan beri nama fungsi dan parameter opsional dengan CREATE FUNCTION

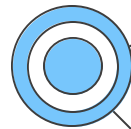
```
CREATE FUNCTION Sales.fn_LineTotal (@SalesOrderID INT)
RETURNS TABLE
AS
RETURN
    SELECT SalesOrderID,
           CAST((OrderQty * UnitPrice * (1 - SpecialOfferID))
           AS DECIMAL(8, 2)) AS LineTotal
    FROM Sales.SalesOrderDetail
    WHERE SalesOrderID = @SalesOrderID ;
```

Perintah yang mendefinisikan TABLE VALUE FUNCTION yaitu perintah RETURNS TABLE. Setelah itu QUERY nya di buat didalam perintah RETURN setelah perintah AS.

...



MENULIS QUERIES DENGAN DERIVED TABLES



...

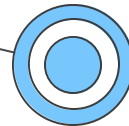
DERIVED TABLES adalah **QUERY yang diberi nama**, dibuat dalam outer SELECT statement

Tidak disimpan pada database – merepresentasikan virtual relational table
Memungkinkan kita menulis modular QUERIES

```
SELECT <column_list>  
FROM (  
    <derived_table_definition>  
) AS <derived_table_alias>;
```

Lingkup dari DERIVED TABLE adalah pada QUERY dimana DERIVED TABLE tersebut dibuat

...



GUIDELINES FOR DRIVED TABLES

DERIVED TABLES Must

- Have an alias
- Have names for all columns
- Have unique names for all columns
- Not use an ORDER BY clause (without TOP or OFFSET/FETCH)
- Not be referred to multiple times in the same QUERY

DERIVED TABLES May

- Use internal or external aliases for columns
- Refer to parameters and/or VARIABLES
- Be nested within other DERIVED TABLES



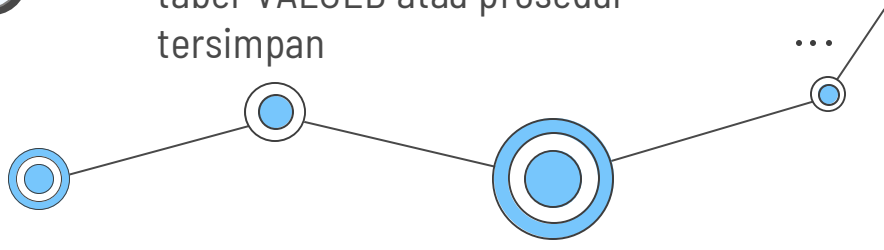
The Slide Title Goes Here!



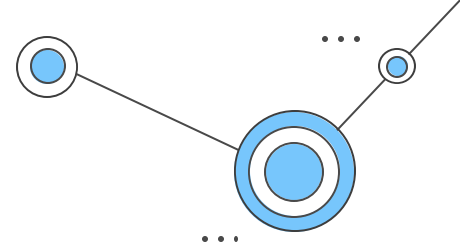
```
DECLARE @emp_id INT = 9;
SELECT orderyear, COUNT(DISTINCT custid) AS
cust_count
FROM (
    SELECT YEAR(orderdate) AS
orderyear, custid
    FROM Sales.Orders
    WHERE empid=@emp_id
) AS derived_year
GROUP BY orderyear;
```

Tabel TURUNAN dapat merujuk pada argumen

Argumen dapat berupa:

- VARIABEL yang dideklarasikan dalam batch yang sama seperti pernyataan SELECT
 - Parameter diteruskan ke fungsi tabel-VALUED atau prosedur tersimpan
- 

CTE



WITH keyword
CTE Name

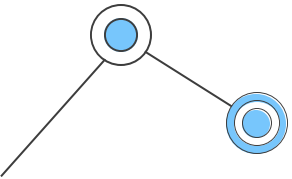
```
WITH avg_price_brand AS  
    (SELECT brand, AVG(price) AS average_for_brand  
     FROM cameras  
     GROUP BY brand)  
SELECT c.id, c.brand, c.model, c.price, avg.average_for_brand  
FROM cameras c  
JOIN avg_price_brand avg  
ON c.brand = avg.brand;
```

CTE body

Use of CTE

CTE keywords

```
with CTE Name as (  
    CTE body  
    Inner query defining the CTE  
)  
  
Outer query using CTE  
as part of the main query
```



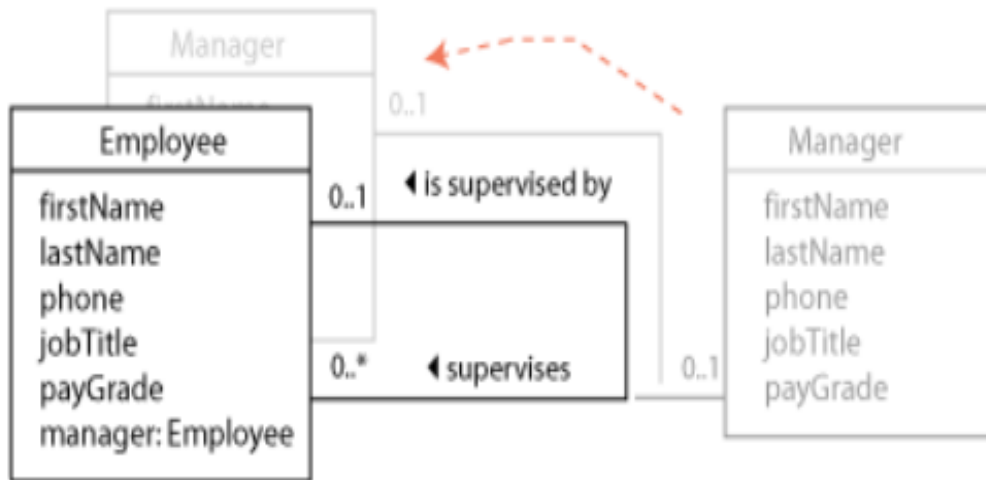
MEMBUAT QUERIES DENGAN CTE

Common Table Expression

(CTE) adalah salah satu bentuk query SQL yang digunakan **untuk menyederhanakan JOIN pada SQL kedalam subqueries dan mampu memberikan query yang bersifat hierarki.**

Hirarki dengan CTE

contohnya adalah design pada masalah employee-manager



Membuat QUERIES dengan common table expressions

```
WITH CTE_year AS
(
  SELECT YEAR(OrderDate) AS OrderYear, customerID
  FROM Sales.SalesOrderHeader
)
SELECT orderyear, COUNT(DISTINCT CustomerID) AS
CustCount
FROM CTE_year
GROUP BY OrderYear;
```

Use WITH clause to create a CTE:

- Definiskan CTE dengan WITH
- Panggil CTE di outer QUERY
- Berikan alias (inline or external)
- Berikan arguments jika diinginkan

Thanks!

Do you have any questions?

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), infographics & images by [Freepik](#) and illustrations by [Stories](#)

