

Virtual Internship Experience

id/x partners

**Stored Procedures &
Query Scheduling**



Disclaimer

“Dokumen ini memiliki hak cipta. Barang siapa yang menyebarluaskan atau menduplikasi tanpa izin dari instansi terkait dapat diproses sesuai dengan ketentuan hukum yang berlaku.”



Outline

- 1 ***Stored Procedures***
- 2 ***Query Scheduling***
- 3 **Case Study**



1 *Stored Procedures*

Stored Procedures adalah kumpulan pernyataan SQL yang telah dikompilasi sebelumnya yang dapat dieksekusi sebagai satu unit kerja. *Stored Procedures* disimpan dalam *database* dan dapat dipanggil oleh aplikasi atau *Stored Procedures* lainnya. *Stored Procedures* memberikan beberapa manfaat dibandingkan pernyataan SQL ad-hoc, termasuk peningkatan kinerja, peningkatan keamanan, dan penggunaan kembali kode yang lebih baik.

1 *Stored Procedures*

Kenapa kita gunakan *Stored Procedures* pada *SQL Data Warehouse*?



Performance



Security



Reuseability



Abstraction



id/x partners

1 **Stored Procedures**

Membuat *Stored Procedures* pada *SQL Data Warehouse*:
Basic Syntax

CREATE PROCEDURE

[schema_name.]procedure_name

(

@parameter1 datatype1 [= default_value1],

@parameter2 datatype2 [= default_value2],

...

)

AS

BEGIN

-- SQL statements

END;



id/x partners

1 **Stored Procedures**

Membuat *Stored Procedures* pada *SQL Data Warehouse*:

Contoh Syntax

```
CREATE PROCEDURE dbo.CalculateTotalSales
(
    @start_date date,
    @end_date date
)
AS
BEGIN
    SELECT SUM(sales)
    FROM Sales
    WHERE order_date BETWEEN @start_date AND
    @end_date;
END;
```





id/x partners

1 **Stored Procedures**

Membuat *Stored Procedures* pada *SQL Data Warehouse*:

Mengeksekusi *Syntax*

```
EXEC [schema_name.]procedure_name  
[@parameter1 = value1, @parameter2 = value2, ...];
```

Contoh *Syntax*

```
EXEC dbo.CalculateTotalSales [@start_date =  
'2022-01-01', @end_date = '2022-12-31'];
```





2 *Query Scheduling*

Query Scheduling adalah fitur yang memungkinkan Anda mengotomatiskan eksekusi kueri SQL di lingkungan *data warehouse*. *Query Scheduling* memungkinkan Anda untuk mengotomatiskan tugas berulang, seperti ekstraksi data dan *load data*, dan membantu Anda memastikan bahwa data terbaru dan tersedia saat Anda membutuhkannya. *Query Scheduling* biasanya dikelola oleh *scheduler tools*, seperti sistem manajemen basis data (DBMS) atau *job scheduler*.

2 *Query Scheduling*

Kenapa *Query Scheduling* itu penting?



Data Freshness



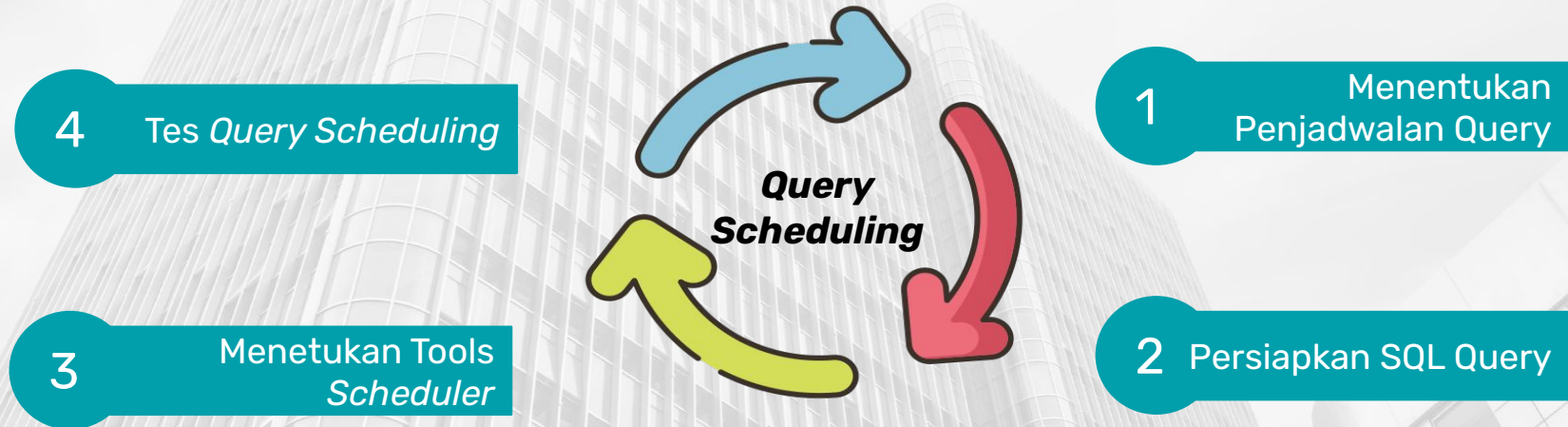
Automation



Performance

2 Query Scheduling

Berikut merupakan cara yang dapat anda lakukan untuk membuat Query Scheduling:





3 Case Study: Problem Statement

Perusahaan client ID/X Partners mengoperasikan online retail store dan mengelola SQL Data Warehouse untuk menyimpan data penjualan. Data penjualan diperbarui setiap hari dan perusahaan ingin memastikan bahwa data tersebut mutakhir dan tersedia saat mereka membutuhkannya. Perusahaan juga ingin mengotomatiskan proses ekstraksi dan pemuatan data penjualan ke dalam data warehouse.



3 Case Study: Solution

Untuk memenuhi persyaratan ini, perusahaan memutuskan untuk menggunakan penjadwalan kueri dengan *Stored Procedure* dan Apache Airflow.

1. *Stored Procedure*: Perusahaan membuat prosedur tersimpan untuk merangkum logika bisnis kompleks untuk mengekstraksi dan memuat data penjualan.
2. Apache Airflow: Perusahaan menyiapkan Apache Airflow untuk menjadwalkan pelaksanaan prosedur tersimpan.
3. Hasil: Data sekarang mutakhir dan tersedia saat mereka membutuhkannya, menghemat waktu dan mengurangi risiko kesalahan manusia.



Thank You!



id/x partners