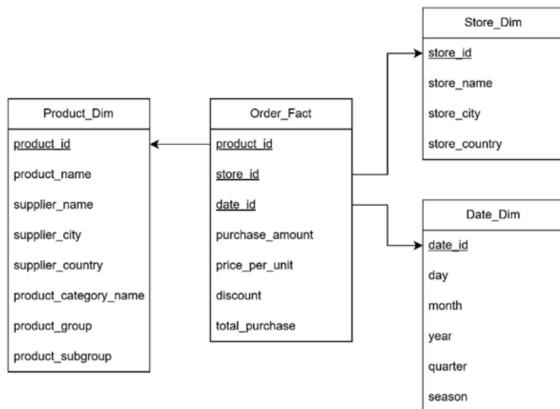


Diketahui sebuah skema basis data relasional operasional yang digunakan dalam sebuah sistem supermarket di bawah ini.



Untuk keperluan analisis bisnis yang dilakukan pihak manajemen yang biasanya terjadi pada setiap quarter (triwulan), pihak supermarket ingin membangun sebuah basisdata analitik yang dimodelkan dengan model multidimensional sebagai berikut.



Basis data analitik akan diisi berdasarkan data dari basis data operasional dengan penjelasan sebagai berikut.

1. Tabel Product_Dim diisi berdasarkan isi tabel Product, Product_Category, dan Supplier.
2. Tabel Store_Dim diisi berdasarkan isi tabel Store.
3. Tabel Date_Dim diisi berdasarkan isi atribut date pada tabel Order.
4. Tabel Order_Fact diisi berdasarkan isi tabel Order dan Order_Details.
5. Atribut product_id, store_id, dan date_id pada basis data analitik merupakan *artificial key*, yaitu atribut baru bertipe integer yang di-generate secara otomatis khusus untuk keperluan basis data analitik.

Tugas Anda:

- 1) Tuliskan 1 (satu) alternatif urutan yang masuk akal untuk melakukan pengisian tabel-tabel pada tabel basis data analitik di atas dan berikan penjelasan.
- 2) Tuliskan mapping untuk memindahkan data dari basis data operasional ke basis data analitik untuk tiap tabel di basis data analitik dengan menggunakan format berikut.

Tabel target	<tabel target di basis data target>
Tabel-tabel sumber	<tabel(-tabel) di basis data sumber yang menjadi sumber data>
Frekuensi load	<kapan data harus di-load/periode melakukan load, misalnya harian, bulanan, dll.>
Deskripsi mapping	<SQL query untuk melakukan pemindahan/load data dari basis data sumber ke basis data target>

- 3) Jelaskan apakah *loading* tabel-tabel pada basis data analitik dari basis data operasional seharusnya dilakukan dalam secara *batch*, *near-real-time*, atau *real-time*? Berikan alasan, minimum berkaitan dengan aspek *latency data*.
- 4) Jelaskan 3 (tiga) hal yang dapat menjadi tantangan dalam proses ETL pada sistem basis data di atas.

1. Sebuah alternatif urutan yang masuk akal untuk melakukan pengisian tabel-tabel pada tabel basis data analitik:

- (1) Tabel Dimensi (Store_Dim, Product_Dim, Date_Dim)
- (2) Tabel Fakta (Order_Fact)

Dalam konsep data *warehouse* dan ETL, tabel fakta berisi *foreign key* yang merujuk pada *primary key* di tabel dimensi. Oleh sebab itu, dimensi wajib sudah terisi terlebih dahulu sebelum fakta di-*load* agar *lookup surrogate key* dapat dilakukan saat pengisian Order_Fact. Sedangkan untuk tabel dimensi sendiri tidak saling bergantung, sehingga urutannya dapat dibebaskan asalkan semuanya diisi sebelum pengisian tabel fakta.

2. *Mapping* untuk memindahkan data dari basis data operasional ke basis data analitik:

(1) Tabel Store_Dim

Komponen	Detail
Tabel Target	Store_Dim
Tabel Sumber	Store
Frekuensi Load	Triwulan
Deskripsi Mapping	<pre>INSERT INTO Store_Dim(store_name, store_city, store_country) SELECT name, city, country FROM Store;</pre>

(2) Tabel Product_Dim

Komponen	Detail
Tabel Target	Product_Dim
Tabel Sumber	Product, Product_Category, Supplier
Frekuensi Load	Triwulan
Deskripsi Mapping	<pre>INSERT INTO Product_Dim (product_name, supplier_name, supplier_city, supplier_country, product_category_name, product_group, product_subgroup) SELECT p.name, s.name, s.city, s.country, pc.name, pc.group, pc.subgroup FROM Product p JOIN Supplier s ON p.supplier_code = s.supplier_code JOIN Product_Category pc ON p.product_category_code = pc.product_category_code;</pre>

(3) Tabel Date_Dim

Komponen	Detail
Tabel Target	Date_Dim
Tabel Sumber	Order
Frekuensi Load	Triwulan
Deskripsi Mapping	<pre>INSERT INTO Date_Dim (day, month, year, quarter, season) SELECT DISTINCT EXTRACT(DAY FROM o.date), EXTRACT(MONTH FROM o.date), EXTRACT(YEAR FROM o.date), EXTRACT(QUARTER FROM o.date), CASE WHEN EXTRACT(MONTH FROM o.date) IN (12, 1, 2) THEN 'Winter' WHEN EXTRACT(MONTH FROM o.date) IN (3, 4, 5) THEN 'Spring' WHEN EXTRACT(MONTH FROM o.date) IN (6, 7, 8) THEN 'Summer' ELSE 'Fall' END AS season FROM "Order" o;</pre>

(4) Tabel Order_Fact

Komponen	Detail
Tabel Target	Order_Fact
Tabel Sumber	Order, Order_Details
Frekuensi Load	Harian
Deskripsi Mapping	<pre>INSERT INTO Order_Fact (product_id, store_id, date_id, purchase_amount, price_per_unit, discount, total_purchase) SELECT pd.product_id, sd.store_id, dd.date_id, od.purchase_amount, od.price_per_unit, od.discount, od.total_purchase FROM Order_Details od JOIN "Order"</pre>

```
o ON od.order_no =  
o.order_no JOIN Product p  
ON od.product_code =  
p.product_code JOIN  
Product_Dim pd ON p.name =  
pd.product_name JOIN  
Store_Dim sd ON  
o.store_code = (SELECT  
store_code FROM Store  
WHERE name =  
sd.store_name) JOIN  
Date_Dim dd ON  
EXTRACT(DAY FROM o.date)  
= dd.day AND  
EXTRACT(MONTH FROM  
o.date) = dd.month AND  
EXTRACT(YEAR FROM  
o.date) = dd.year;
```

3. *Loading* tabel-tabel pada basis data analitik dari basis data operasional seharusnya dilakukan secara *batch*. Hal ini karena manajemen melakukan analisis per kuartal (triwulan), sehingga data tidak dibutuhkan saat itu juga (*real-time*). Latensi tinggi masih dapat diterima selama data tersedia ketika periode analisis. Selain itu, *batch processing* sangat berguna untuk memproses *volume* data besar dalam waktu singkat. Memindahkan data transaksi supermarket lebih efisien dilakukan secara *batch* dibanding membebani sistem operasional dengan *trigger* setiap ada transaksi.

4. Tantangan dalam proses ETL pada sistem basis data tersebut:

(1) Variasi dan kualitas data

Tabel dimensi Product_Dim menggabungkan data dari tabel Product, Supplier, dan Product_Category. Kualitas data di sistem operasional mungkin bervariasi atau tidak konsisten sehingga memerlukan *cleansing* sebelum dimuat.

(2) Keunikan kunci

Sistem analitik menggunakan *artificial key*. Tantangannya adalah memetakan *primary key* dari sistem operasional ke *artificial key*. ETL perlu memastikan saat memuat Order_Fact, sistem menemukan id yang tepat di tabel dimensi.

(3) Performa

Tabel-tabel pada sistem, terutama Order dan Order_Details akan terus bertambah dan bisa menjadi sangat besar. Ini dapat menjadi *bottleneck* pada fase *load*, sehingga perlu strategi untuk meningkatkan performanya.

JAWABAN DOSEN:

<p>Catatan: Sejurus query yang diaksipisikan di sini adalah untuk initial loading (loading data BDA di awal). Query yang dituliskan mengasumsikan bahwa data pada basis data yang berbeda bisa diakses secara langsung dengan menambahkan nama skema basis data di awal nama tabel. Pada kondisi yang sebenarnya, harus dicek pada DBMS masing-masing.</p>	
Tabel target	BDA.Product_Dim
Tabel-tabel sumber	BDO.Product, BDO.Product_Category, BDO.Supplier
Frekuensi load	3 bulan sekali
Deskripsi mapping	<pre> INSERT INTO BDA.Product_Dim (product_name, supplier_name, supplier_city, supplier_country, product_category_name, product_group, product_subgroup) SELECT p.name, s.name, s.city, s.country, pc.name, pc.group, pc.subgroup FROM BDO.Product p INNER JOIN BDO.Product_Category pc ON p.product_category_code = pc.product_category_code INNER JOIN BDO.Supplier s ON p.supplier_code = s.supplier_code -- product_id autogenerated oleh DBMS </pre>
Tabel target	BDA.Store_Dim

Tabel target	BDA.Store_Dim
Tabel-tabel sumber	BDO.Store
Frekuensi load	3 bulan sekali
Deskripsi mapping	<pre> INSERT INTO BDA.Store_Dim (store_name, store_city, store_country) SELECT name, city, country FROM BDO.Store -- store_id autogenerated oleh DBMS </pre>

	-- store_id autogenerated oleh DBMS
Tabel target	BDA.Date_Dim
Tabel-tabel sumber	BDO.Order
Frekuensi load	3 bulan sekali
Deskripsi mapping	Asumsi tambahan: Nilai quarter adalah integer 1 (Jan-Mar); 2 (Apr-Jun); 3 (Jul-Sep); 4 (Oct-Dec). Nilai season adalah winter (Dec, Jan, Feb); spring (Mar, Apr, May); summer (Jun, Jul, Aug); autumn (Sep, Oct, Nov) Fungsi-fungsil: day(D) → menghasilkan tanggal date D month(D) → menghasilkan bulan date D
	year(D) → menghasilkan tahun date D

INSERT INTO BDA.Date_Dim (day, month, year, quarter,

```

year,
quarter,
season)
INSERT INTO BDA.Date_Dim (day, month, year, quarter,
season)
SELECT DISTINCT day(date), month(date), year(date),
CASE WHEN month(date) IN (1,2,3) THEN 1
WHEN month(date) IN (4,5,6) THEN 2
WHEN month(date) IN (7,8,9) THEN 3
ELSE -- month(date) IN (10,11,12)
        4
END, -- quarter
CASE WHEN month(date) IN (12,1,2) THEN 'winter'
WHEN month(date) IN (3,4,5) THEN 'spring'
WHEN month(date) IN (6,7,8) THEN 'summer'
ELSE -- month(date) IN (9,10,11)
        'autumn'
FROM BDO.Order;
```

SELECT pd.product_id, p.product_name
FROM BDA.Product_Dim pd INNER JOIN
(SELECT DISTINCT p.name AS product_name,
s.name AS supplier_name,
s.city AS supplier_city,
s.country AS supplier_country,
pc.name AS product_category_name,
pc.group AS product_group,
pc.subgroup AS product_subgroup
FROM BDO.Product p INNER JOIN BDO.supplier s
ON p.supplier_code = s.supplier_code
INNER JOIN product_category pc
ON p.product_category_code = pc.product_category_code
) AS pr
ON pd.product_name = pr.product_name
AND pd.supplier_name = pr.supplier_name
AND pd.supplier_city = pr.supplier_city
AND pd.supplier_country = pr.supplier_country
AND pd.product_category_name = pr.product_category_name
AND pd.product_group = pr.product_group
AND pd.product_subgroup = pr.product_group
-- bisa diorderkan jika ada countsumsi, misalnya
-- product name unik
, sd AS (-- memungkinkan store_id pakai store_code
SELECT s.store_code, sd.store_id
FROM BDO.store s INNER JOIN BDA.Store Dim sd

```
INSERT INTO BDA.Order_Fact(product_id, store_id, date_id,
Purchase_amount)
SELECT dp.product_id, ds.store_id, dd.date_id,
od.price_per_unit,
SUM(od.purchase_amount),
SUM(od.discount),
SUM(od.total_purchase)
FROM BDO.Order o INNER JOIN BDO.Order_Details od
ON o.order_no = od.order_no
INNER JOIN dp ON dp.product_code = od.product_code
INNER JOIN ds ON ds.store_code = o.store_code
INNER JOIN Date_Dim dd
ON dd.day = day(o.date) AND dd.month = month(o.date)
AND dd.year = year(o.date)
GROUP BY dp.product_id, ds.store_id, dd.date_id,
od.price_per_unit
```

-- asumsi price_per_unit untuk tiap pemesanan sebuah
-- product_code tertentu di suatu store tertentu pada suatu
-- hari/tanggal tertentu adalah sama sehingga query di atas
-- akan tetap aman

-- akan tetapi esan

3. Berdasarkan informasi bahwa analisis bisnis dilakukan tiap triwulan, pengisian data basis data analitik, tidak harus dilakukan secara near-real-time atau real-time. Pengisian data dapat dilakukan secara batch 3 bulan sekali pada periode yang ditetapkan, misalnya 1 minggu sebelum dilakukan proses analisis. Proses pengisian data yang besar terjadi pada saat loading data pertama kali (initial loading) ketika set-up sistem basis data analitik. Selanjutnya pengisian data dimungkinkan dalam bentuk update terhadap data yang belum ada di sistem.
4. Tantangan pada proses ETL yang mungkin terjadi pada sistem basis data di antara:
 - a. Performance pada saat loading data: karena loading data dilakukan secara batch, jika jumlah data yang dipindahkan sangat besar, dapat membutuhkan waktu yang cukup lama.
 - b. Pemindahan data secara serial (berurutan) bisa membutuhkan waktu lama jika data yang dipindahkan sangat besar. Pengisian beberapa tabel di basis data analitik dapat dilakukan secara parallel, yaitu tabel Store_Dim, Product_Dim, dan Date_Dim karena tidak ada ketergantungan satu sama lain.
 - c. Pemindahan data dalam jumlah besar secara batch dapat menimbulkan masalah jika terjadi failure di tengah jalan sehingga membutuhkan mekanisme untuk dapat melakukan recovery.

berikan variansi data dan uniqueness

* gangguan dalam sistem ini karena sumbuas