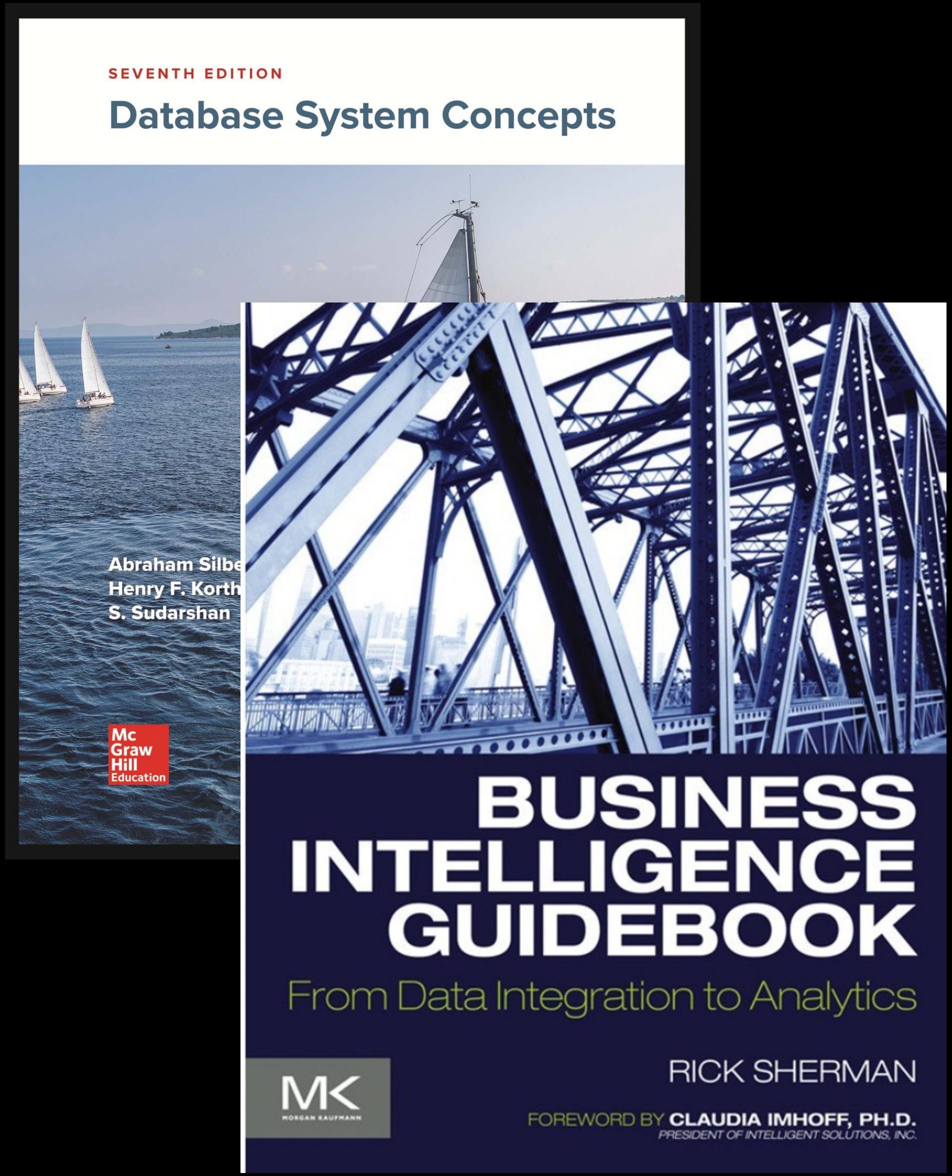
IF2240 – Basis Data

Introduction to Multidimensional Data Model & Data Warehousing

Modified form Silberschatz's slides, Database System Concept, 7th edition

KNOWLEDGE & SOFTWARE ENGINEERING


SEVENTH EDITION
Database System Concepts
Abraham Silbe
Henry F. Korth
S. Sudarshan
Mc Graw Hill Education
BUSINESS INTELLIGENCE GUIDEBOOK
From Data Integration to Analytics
RICK SHERMAN
MK MORGAN KAUFMANN
FOREWORD BY CLAUDIA IMHOFF, PH.D.
PRESIDENT OF INTELLIGENT SOLUTIONS, INC.

References

Abraham Silberschatz, Henry F. Korth, S. Sudarshan : "**Database System Concepts**", 7th Edition

- Chapter 11: Data Analytics

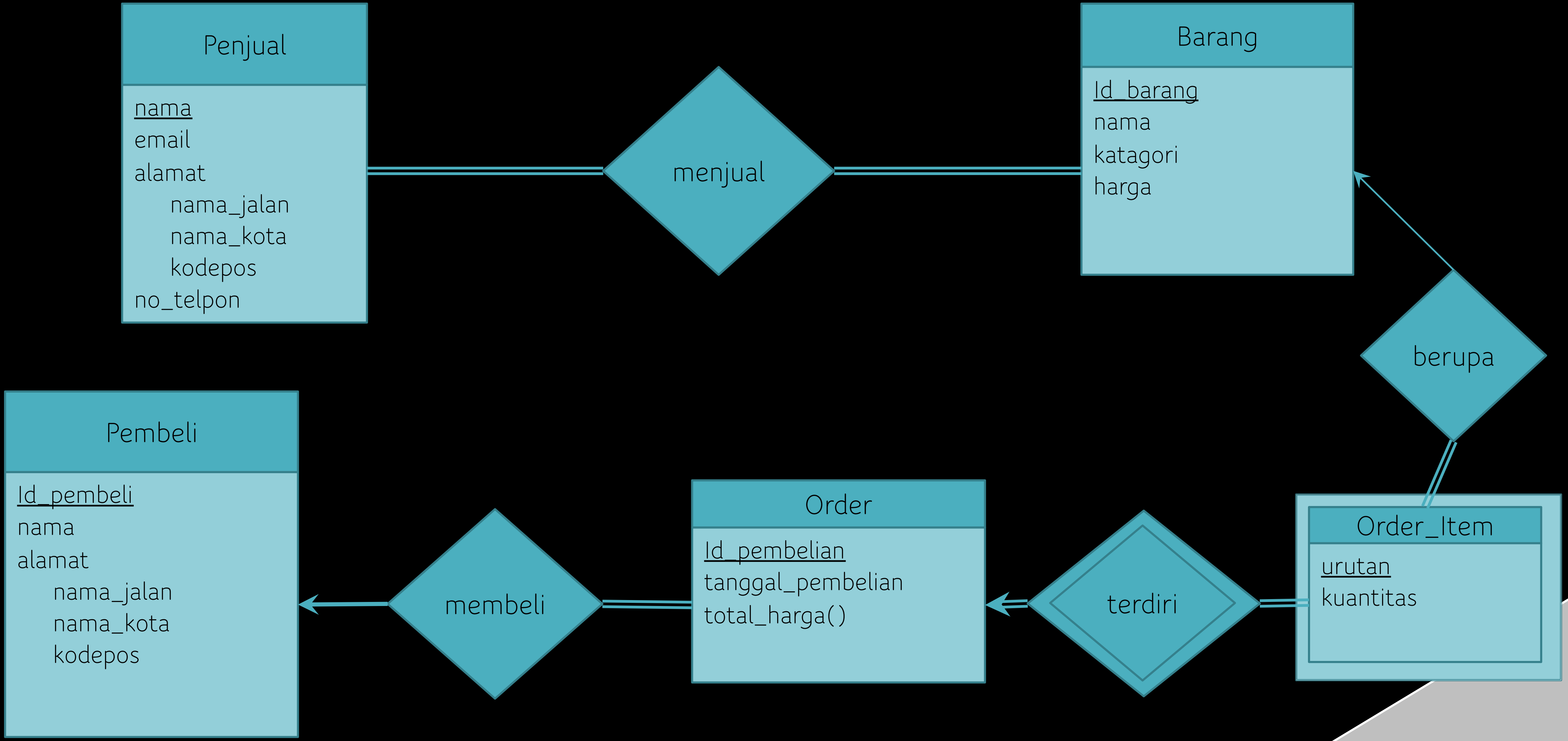
Rich Sherman: "**Business Intelligence Guidebook : From Data Integration to Analytics**", 1st Edition

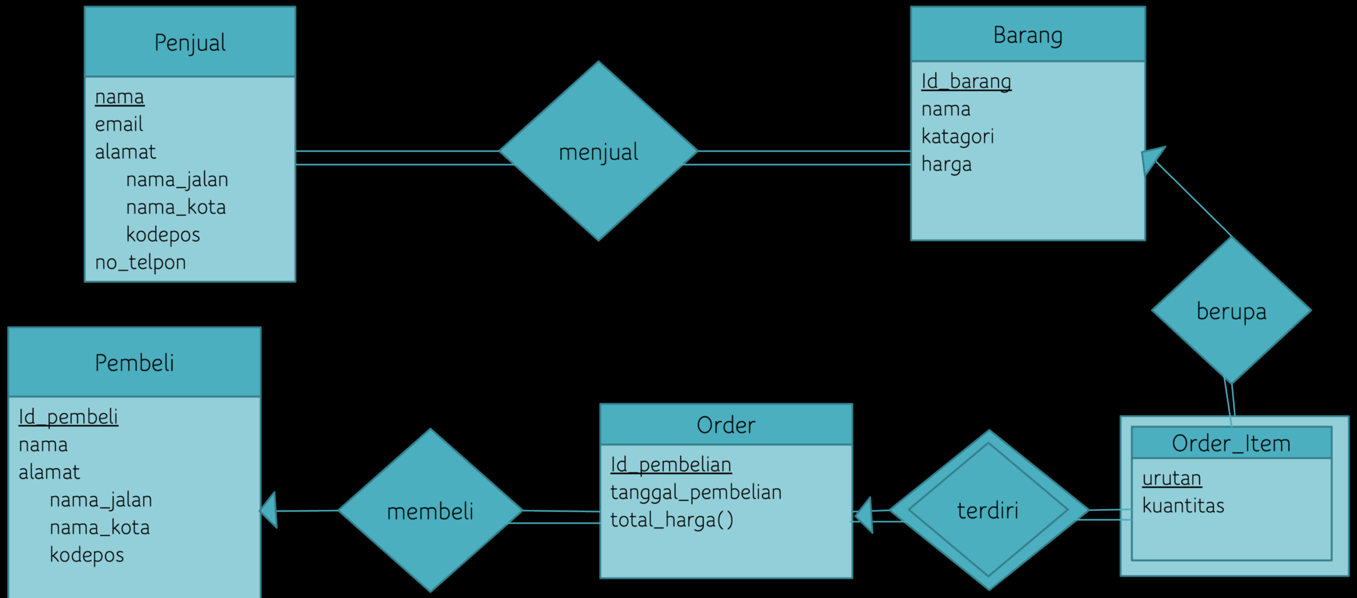
- Chapter 1: The Business Demand for Data, Information, and Analytics

Pemodelan untuk Toko Online

Itsy Bitsy adalah perusahaan e-commerce yang menjual berbagai barang dari berbagai penjual. Saat ini mereka tengah membangun *database* yang akan membantu mereka dalam mencatat **penjual** dan masing-masing **barang** yang mereka jual. Masing-masing penjual memiliki nama yang unik, kontak email, alamat, dan nomor telepon. Penjual dapat **menjual satu atau lebih barang**, masing-masing jenis barang memiliki id yang unik, nama, katagori dan harga.

Pengembangan lebih lanjut, dilakukan penyimpanan informasi **pembeli**, berupa id, nama dan alamat (terdiri atas nama jalan, nama kota, dan kodepos). Setiap pembeli dapat melakukan **pembelian**. Satu kali pembeliaan memiliki id, tanggal pembelian, dan total harga yang dijumlahkan dari harga masing-masing barang yang dibeli. Setiap jenis barang yang dibeli perlu dicatat kuantitasnya.


Penjual
nama
email
alamat
nama_jalan
nama_kota
kodepos
no_telpon
menjual
Barang
Id_barang
nama
katagori
harga
berupa
Pembeli
Id_pembeli
nama
alamat
nama_jalan
nama_kota
kodepos
membeli
Order
Id_pembelian
tanggal_pembelian
total_harga()
terdiri
Order_Item
urutan
kuantitas



Penjual : <u>Nama</u>, email, alamat, alama_nama_jalan, alamat_nama_kota, alamat_kodepos, no_telpon

Barang : <u>id_barang</u>, nama, katagori, harga

Menjual : <u>nama_penjual</u>, <u>id_barang</u>

Order : <u>id_pembelian</u>, tanggal_pembelian, id_pembeli

Order_item : <u>id_pembelian</u>, <u>urutan</u>, kuantitas, id_barang

Pembeli : <u>id_pembeli</u>, nama, alamat_nama_jalan, alamat_nama_kota, alamat_kodepos

FK

Menjual(nama_penjual) → Penjual (nama)

Menjual(id_barang) → Barang(id_barang)

Order(id_pembeli) → Pembeli(id_pembeli)

Order_item(id_pembelian) → Order(id_pembelian)

Order_item(id_barang) → Barang (id_barang)

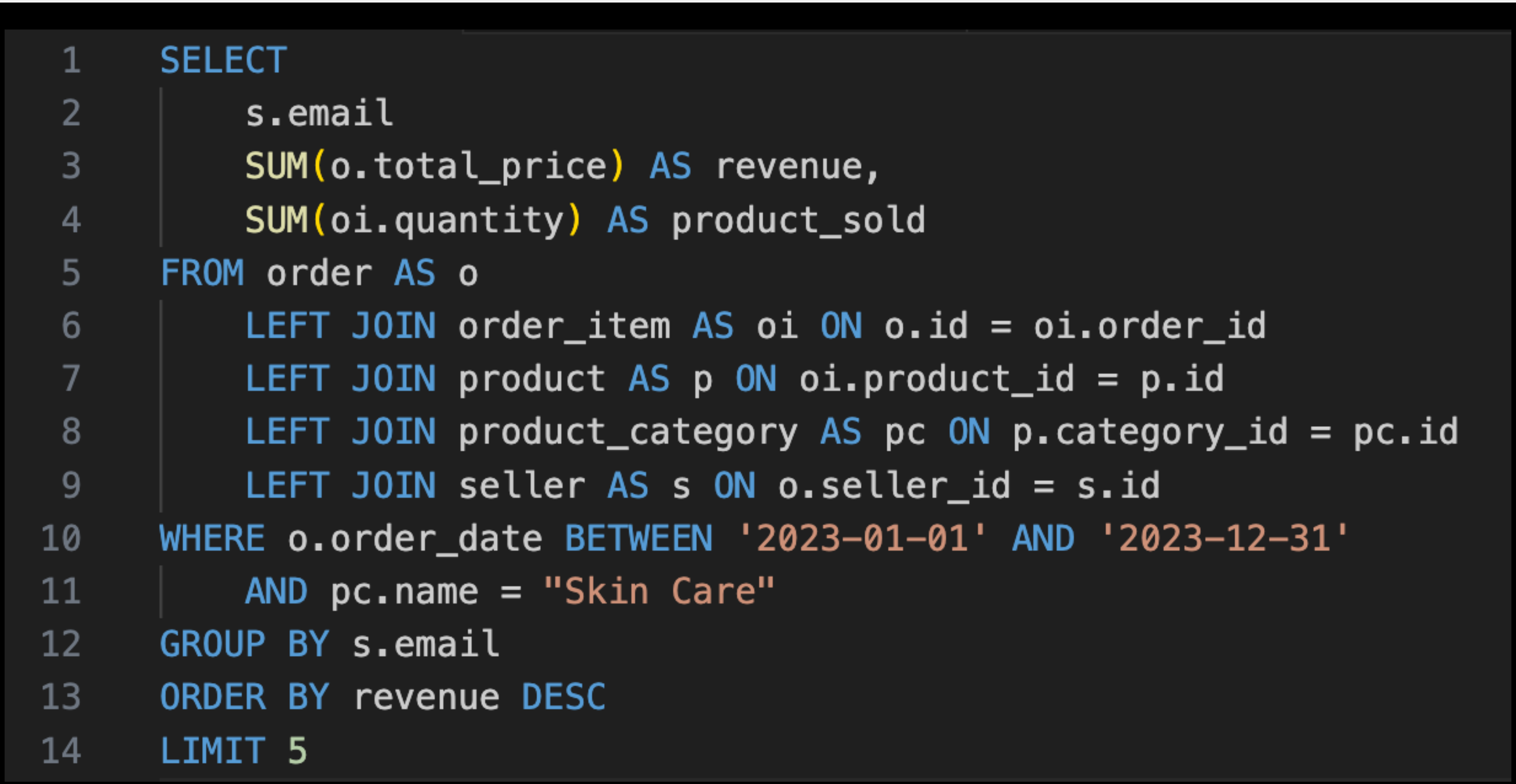
Kebutuhan Informasi untuk Tim Eksekutif (Analysis)

1. Berapa total penjualan barang setiap bulannya?
2. Berapa rata-rata jumlah rupiah penjualan setiap bulannya?
3. Penjual mana yang menjual barang terbanyak bulan ini?
4. Di kota mana market terbesar dari perusahaan?
5. Barang apa yang paling laku terjual?

Contoh masalah yang akan ditimbulkan jika menggunakan model relasional:

1. Peningkatan kompleksitas query
2. Mungin memerlukan waktu eksekusi yang lama dan menganggu proses operasional

Sebuah ilustrasi… dari skema BD lain..
Siapa Penjual Top Skin Care di 2023?



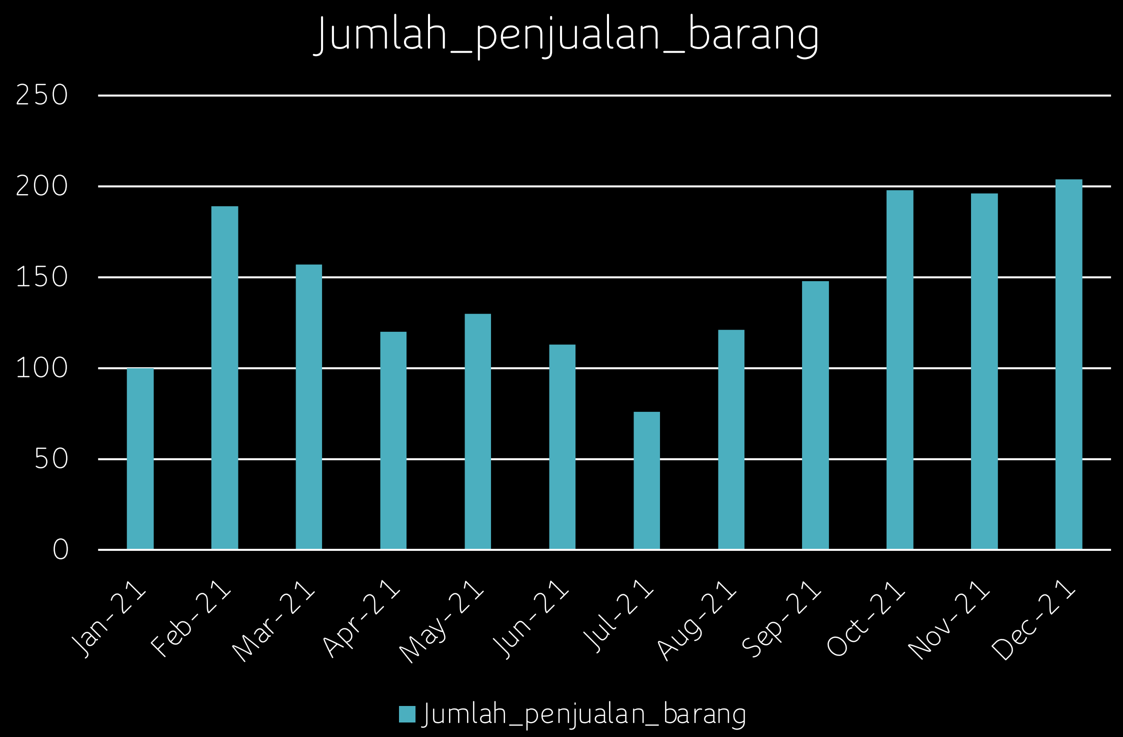
```sql
SELECT
    s.email
    SUM(o.total_price) AS revenue,
    SUM(oi.quantity) AS product_sold
FROM order AS o
    LEFT JOIN order_item AS oi ON o.id = oi.order_id
    LEFT JOIN product AS p ON oi.product_id = p.id
    LEFT JOIN product_category AS pc ON p.category_id = pc.id
    LEFT JOIN seller AS s ON o.seller_id = s.id
WHERE o.order_date BETWEEN '2023-01-01' AND '2023-12-31'
    AND pc.name = "Skin Care"
GROUP BY s.email
ORDER BY revenue DESC
LIMIT 5
```

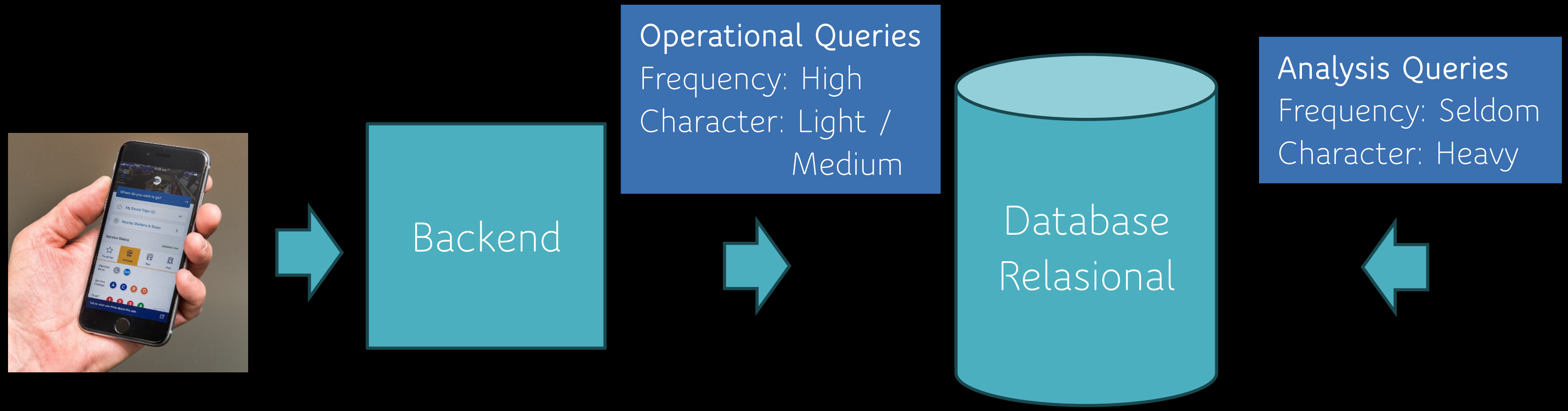
Kebutuhan Informasi Pihak Eksekutif digunakan untuk keperluan analisis sehingga melibatkan data dalam jumlah yang besar…

Filter jenis barang : semua ▼

Filter jenis cuaca : cerah ▼

Filter jenis penjual : semua ▼



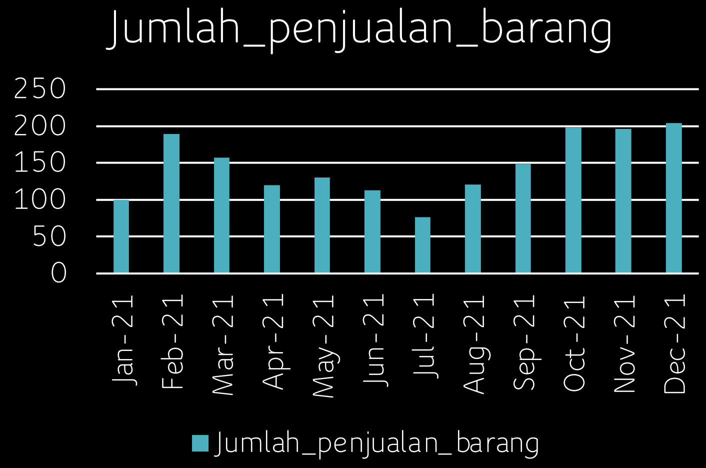


Backend

Operational Queries
Frequency: High
Character: Light / Medium

Database Relasional

Analysis Queries
Frequency: Seldom
Character: Heavy



Contoh masalah yang akan ditimbulkan jika menggunakan model relasional:
Mungin memerlukan waktu eksekusi yang lama dan **menganggu proses operasional**

Multidimensional (1/2)

The multidimensional model begins with the observation that the factors affecting decision-making processes are **enterprise-specific *facts***, such as sales, shipments, hospital admissions, surgeries, and so on. Instances of a fact **correspond to *events* that occurred.** For example, every single sale or shipment carried out is an event. Each fact is described by the values of a set of **relevant *measures* that provide a quantitative description of events**. For example, sales receipts, amounts shipped, hospital admission costs, and surgery time are measures.

Terminology

- **Dimension**: subject label for a row or column
- **Member**: value of dimension
- **Measure**: quantitative variables stored in cells

KNOWLEDGE & SOFTWARE ENGINEERING

Multidimensional (2/2)

Used metaphor of ***cube*****s to represent multidimensional data.**

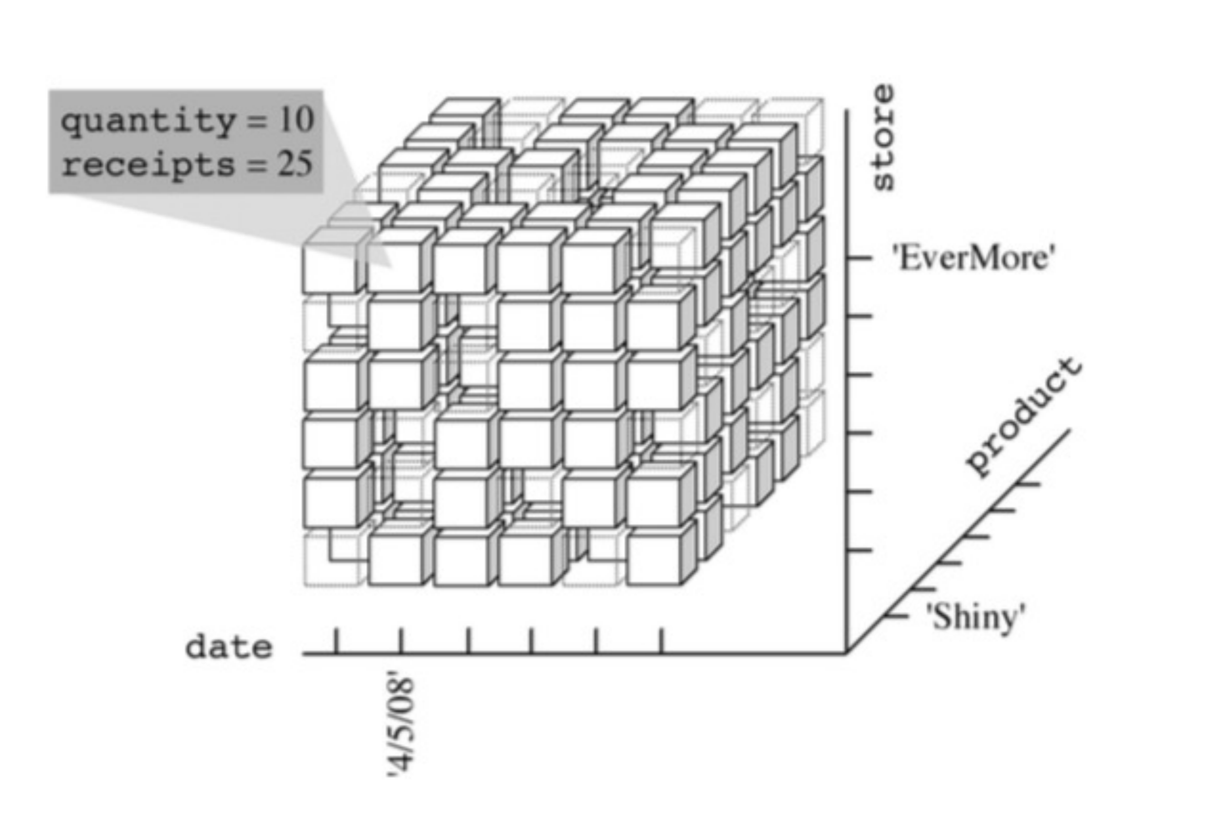
Events are associated with **cube cells.** Each cube cell is given a value for each measure

Cube edges stand for analysis dimensions.

If more than three dimensions exist, the cube is called a *hypercube*.

KNOWLEDGE & SOFTWARE ENGINEERING

Sales Data Cube Example



The three-dimensional cube modeling sales in a store chain:
10 packs of Shiny were sold on 4/5/2008 in the EverMore store, totaling $25

Relational schema :

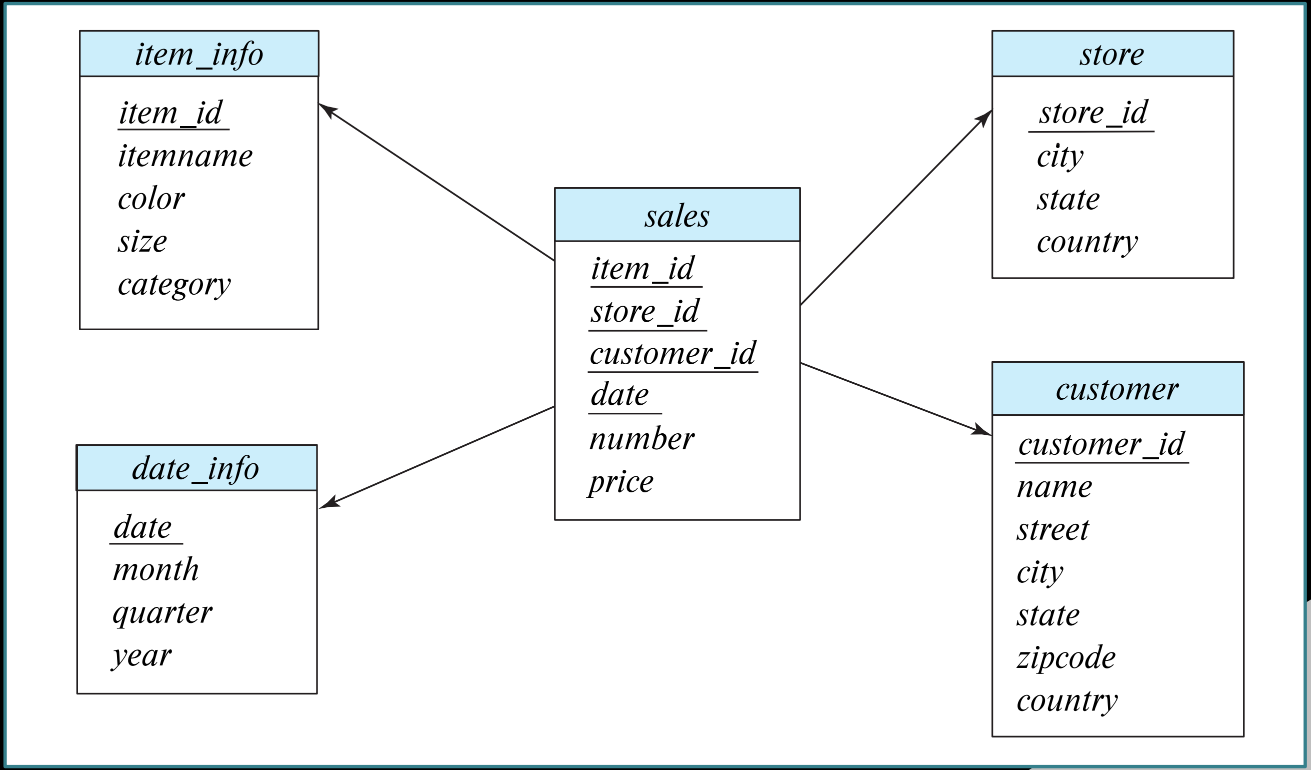
SALES(store, product, date, quantity, receipts)

<'EverMore', 'Shiny', '04/05/08', 10, 25>

Multidimensional Data and Warehouse Schemas (1/2)

Data in warehouse can usually be divided into

- **Fact tables**, which are large
  - E.g, *sales*(*item_id, store_id, customer_id, date, number, price*)
- **Dimension tables**, which are relatively small
  - Store extra information about stores, items, etc.



item_info: *item_id*, *itemname*, *color*, *size*, *category*

store: *store_id*, *city*, *state*, *country*

sales: *item_id*, *store_id*, *customer_id*, *date*, *number*, *price*

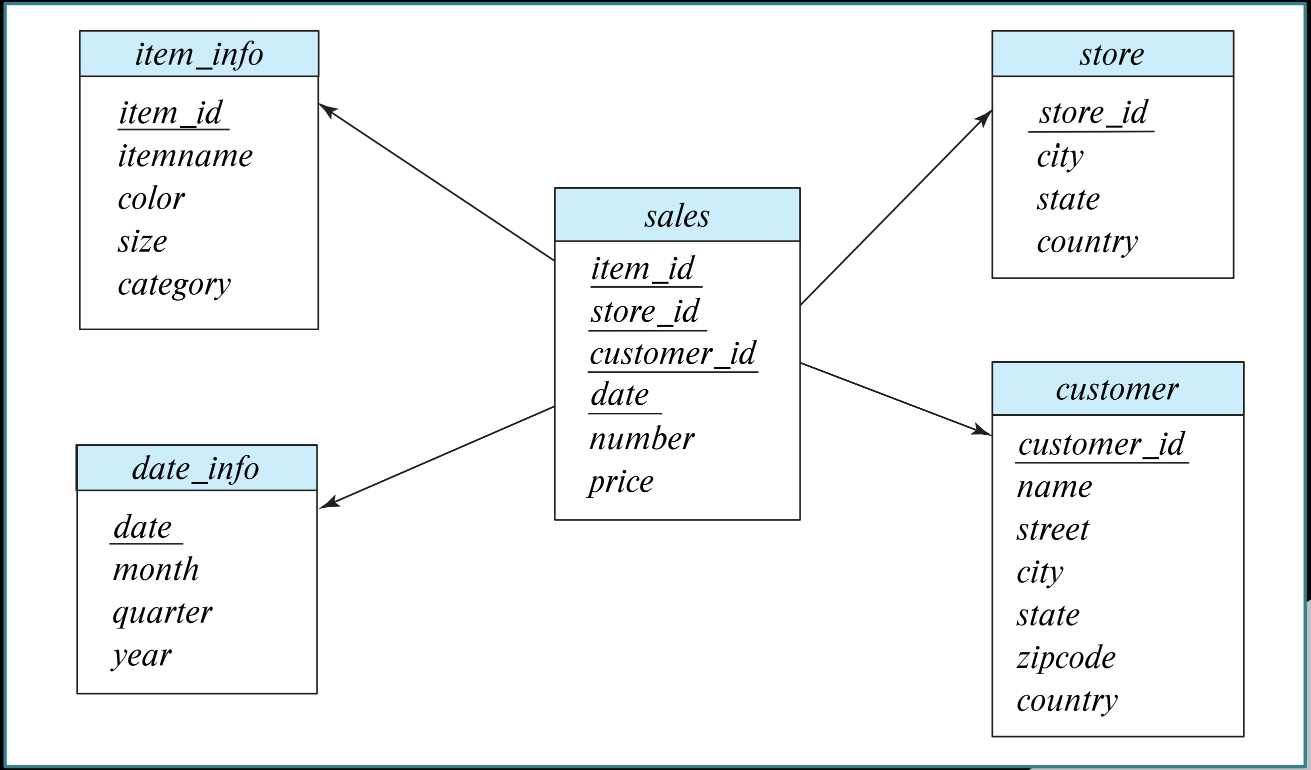
date_info: *date*, *month*, *quarter*, *year*

customer: *customer_id*, *name*, *street*, *city*, *state*, *zipcode*, *country*

Multidimensional Data and Warehouse Schemas (2/2)

Attributes of fact tables can be usually viewed as

- Measure attributes
  - measure some value, and can be aggregated, e.g., the attributes *number* or *price* of the *sales* relation
- Dimension attributes
  - dimensions on which measure attributes are viewed, e.g., attributes *item_id, color,* and *size* of the *sales* relation
  - Usually small ids that are foreign keys to dimension tables



item_info: *item_id*, *itemname*, *color*, *size*, *category*

store: *store_id*, *city*, *state*, *country*

sales: *item_id*, *store_id*, *customer_id*, *date*, *number*, *price*

date_info: *date*, *month*, *quarter*, *year*

customer: *customer_id*, *name*, *street*, *city*, *state*, *zipcode*, *country*

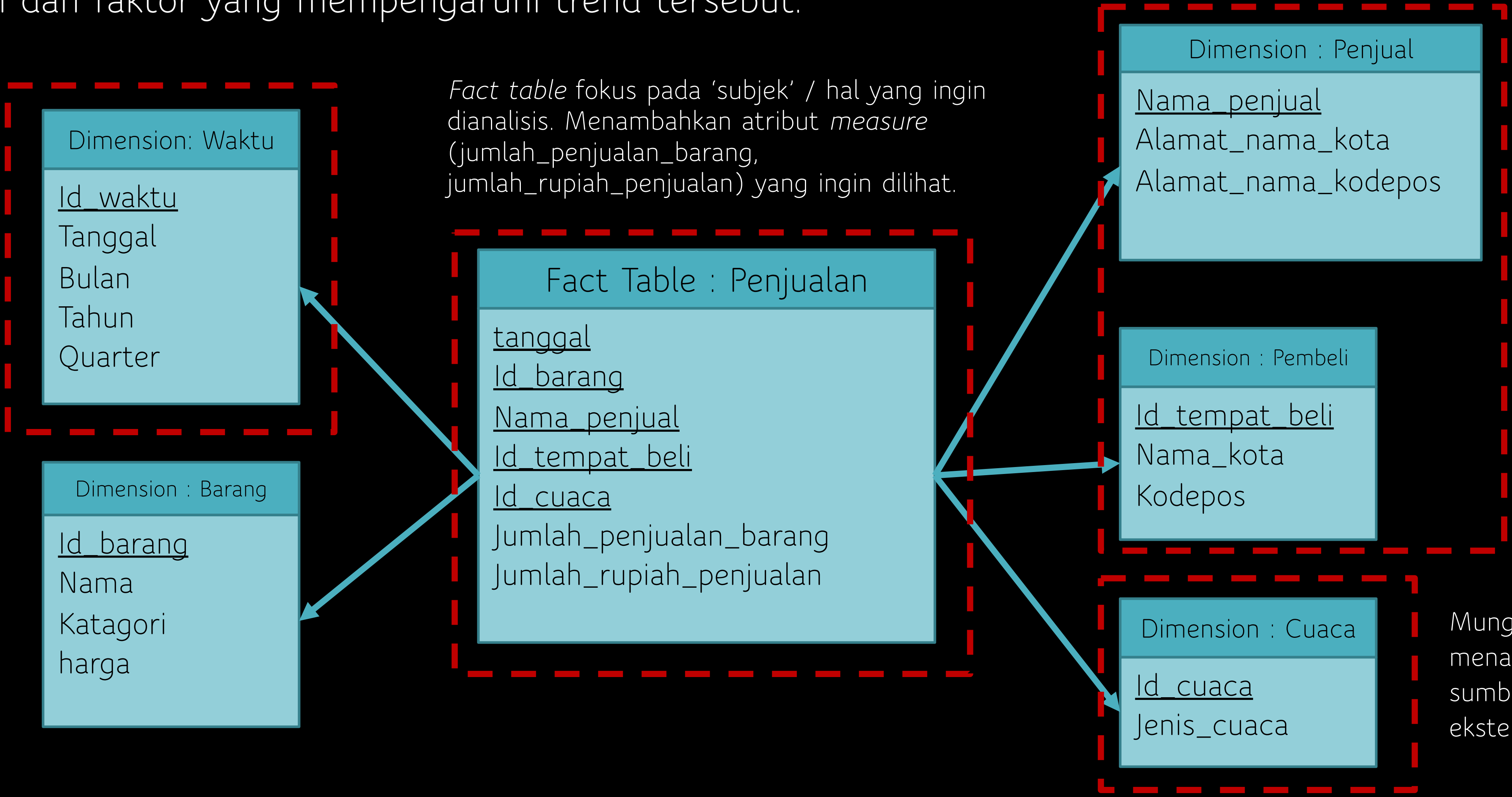
Kebutuhan Informasi untuk Tim Eksekutif

1. Berapa total penjualan barang setiap bulannya?
2. Berapa rata-rata jumlah rupiah penjualan setiap bulannya?
3. Penjual mana yang menjual barang terbanyak bulan ini?
4. Di kota mana market terbesar dari perusahaan?
5. Barang apa yang paling laku terjual?

Dapat diselesaikan dengan pendekatan model multidimensional

Adanya kebutuhan dari pihak manajerial untuk mengetahui trend penjualan dan faktor yang mempengaruhi trend tersebut.

Dimensi 'wajib' yang ada pada multidimensional model. *Granularity* bisa disesuikan dengan kebutuhan.



Fact table fokus pada 'subjek' / hal yang ingin dianalisis. Menambahkan atribut *measure* (jumlah_penjualan_barang, jumlah_rupiah_penjualan) yang ingin dilihat.

Dimension: Waktu

Id_waktu
Tanggal
Bulan
Tahun
Quarter

Dimension : Barang

Id_barang
Nama
Katagori
harga

Fact Table : Penjualan

tanggal
Id_barang
Nama_penjual
Id_tempat_beli
Id_cuaca
Jumlah_penjualan_barang
Jumlah_rupiah_penjualan

Dimension : Penjual

Nama_penjual
Alamat_nama_kota
Alamat_nama_kodepos

Dimension : Pembeli

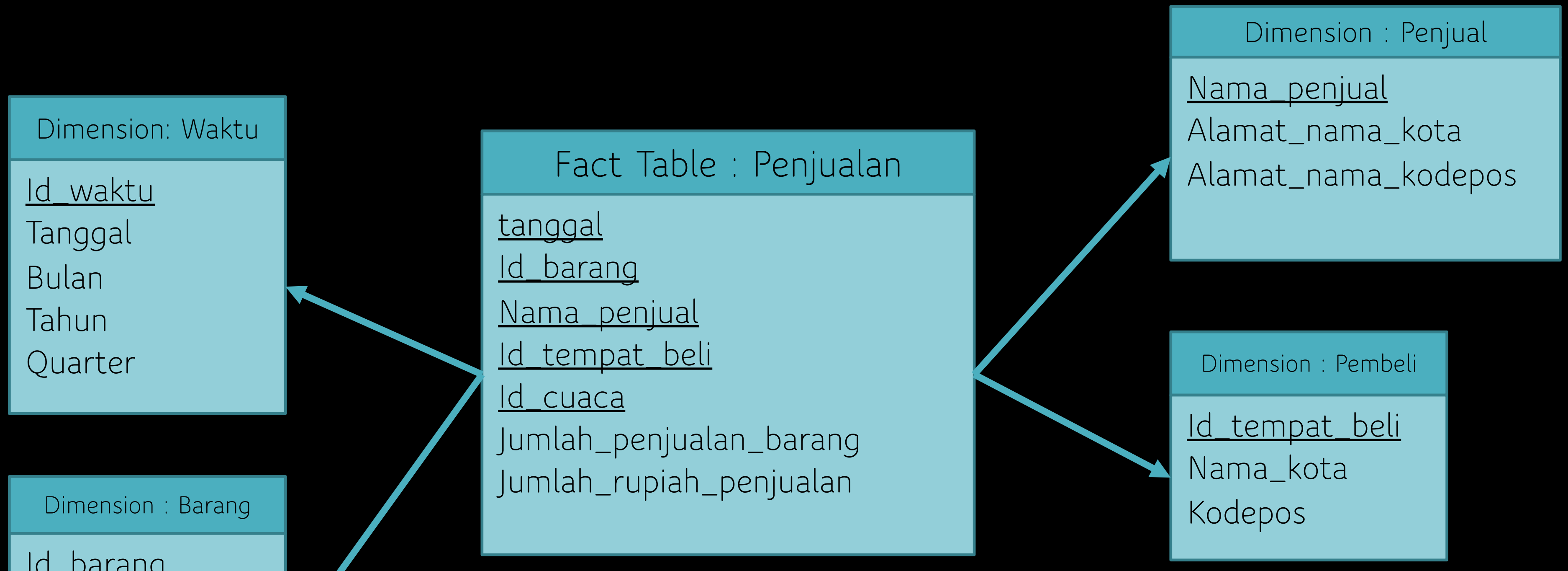
Id_tempat_beli
Nama_kota
Kodepos

Dimension : Cuaca

Id_cuaca
Jenis_cuaca

Tidak perlu detail data. Data 'email penjual' ataupun 'nama pembeli' tidak perlu disimpan karena tidak memiliki makna bila diagregasi

Mungkin menambahkan sumber eksternal



Dimension: Waktu

Id_waktu
Tanggal
Bulan
Tahun
Quarter

Dimension : Barang

Id_barang
Nama
Katagori
harga

Fact Table : Penjualan

tanggal
Id_barang
Nama_penjual
Id_tempat_beli
Id_cuaca
Jumlah_penjualan_barang
Jumlah_rupiah_penjualan

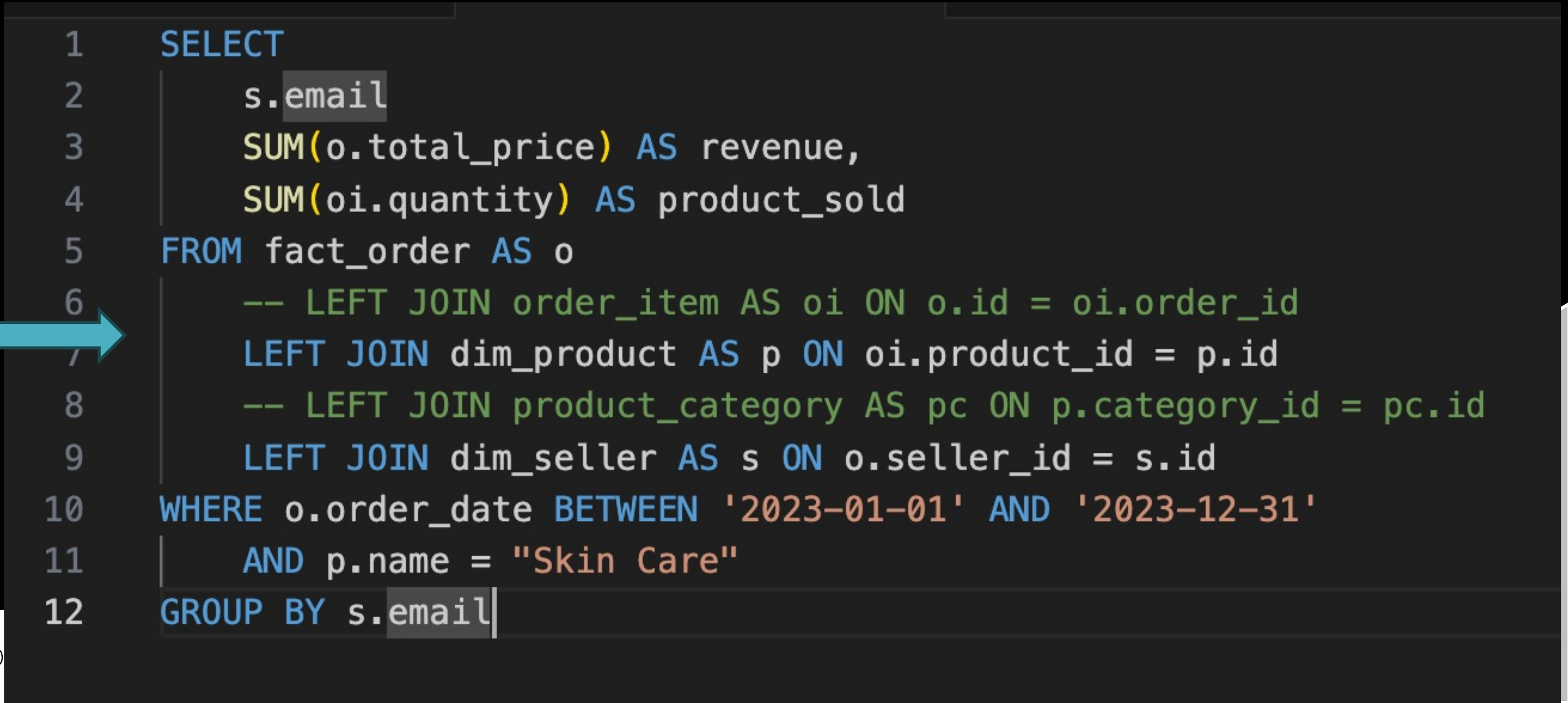
Dimension : Penjual

Nama_penjual
Alamat_nama_kota
Alamat_nama_kodepos

Dimension : Pembeli

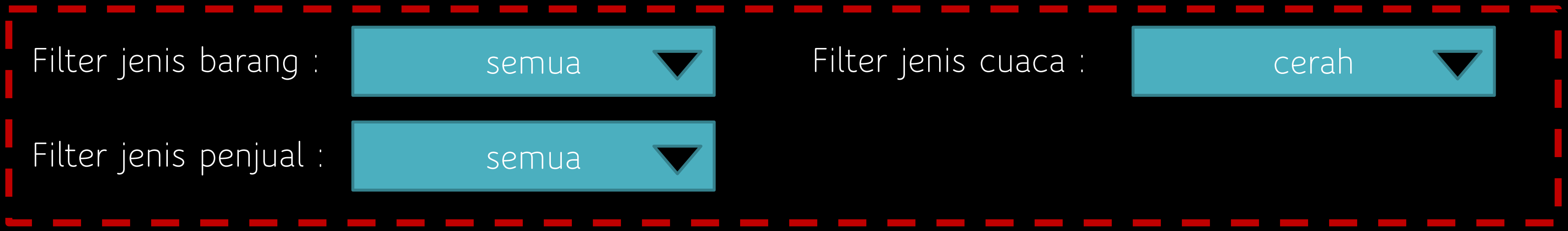
Id_tempat_beli
Nama_kota
Kodepos

Commented lines show how dimensional model simplified the query



```sql
SELECT
    s.email
    SUM(o.total_price) AS revenue,
    SUM(oi.quantity) AS product_sold
FROM fact_order AS o
    -- LEFT JOIN order_item AS oi ON o.id = oi.order_id
    LEFT JOIN dim_product AS p ON oi.product_id = p.id
    -- LEFT JOIN product_category AS pc ON p.category_id = pc.id
    LEFT JOIN dim_seller AS s ON o.seller_id = s.id
WHERE o.order_date BETWEEN '2023-01-01' AND '2023-12-31'
    AND p.name = "Skin Care"
GROUP BY s.email
```

Multidimensional model tersebut dapat membantu meng-generate report untuk analisis trend yang dibutuhkan:

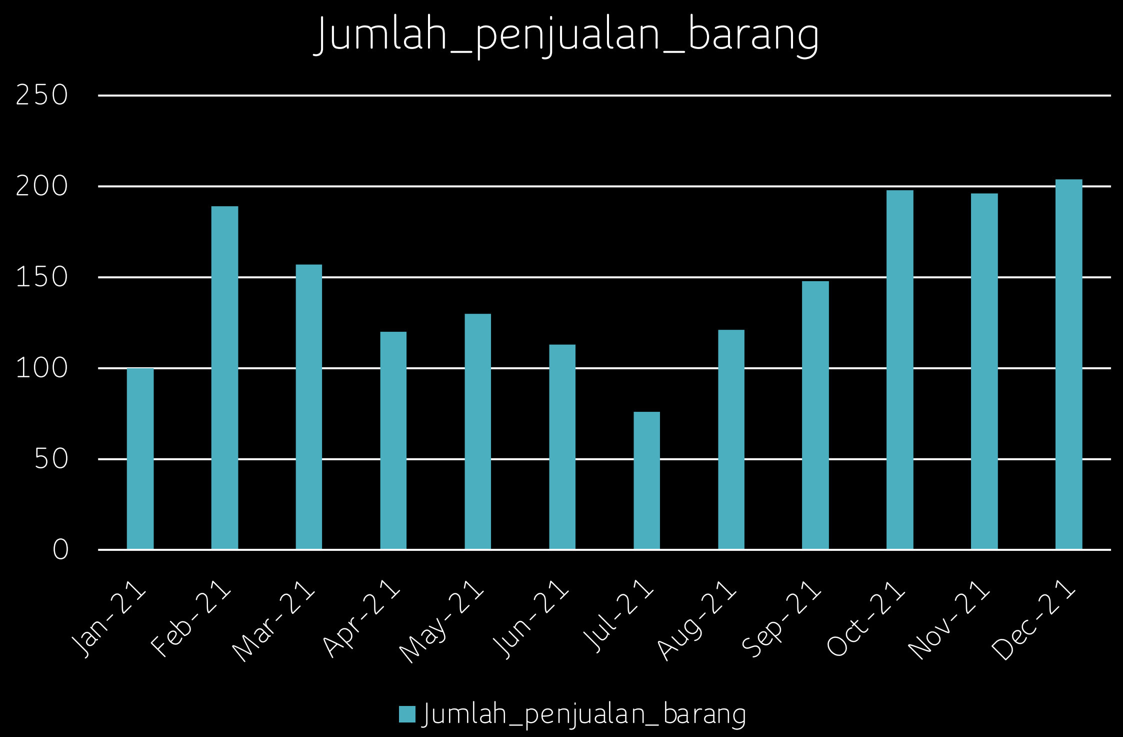


Filter jenis barang : semua ▼

Filter jenis cuaca : cerah ▼

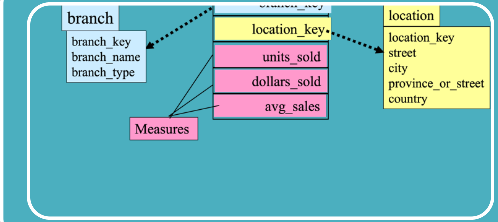
Filter jenis penjual : semua ▼

Bisa menambahkan filter sesuai dengan dimensi yang tersedia



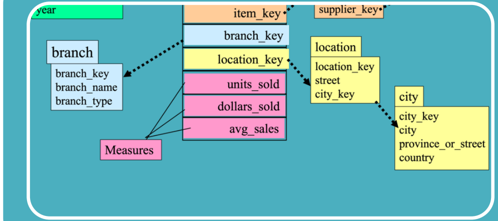
Conceptual Modeling of Data Warehouses

Modeling data warehouses: dimensions & measures



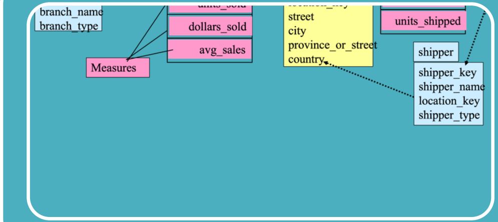
Star schema:

- A fact table in the middle connected to a set of dimension tables



Snowflake schema:

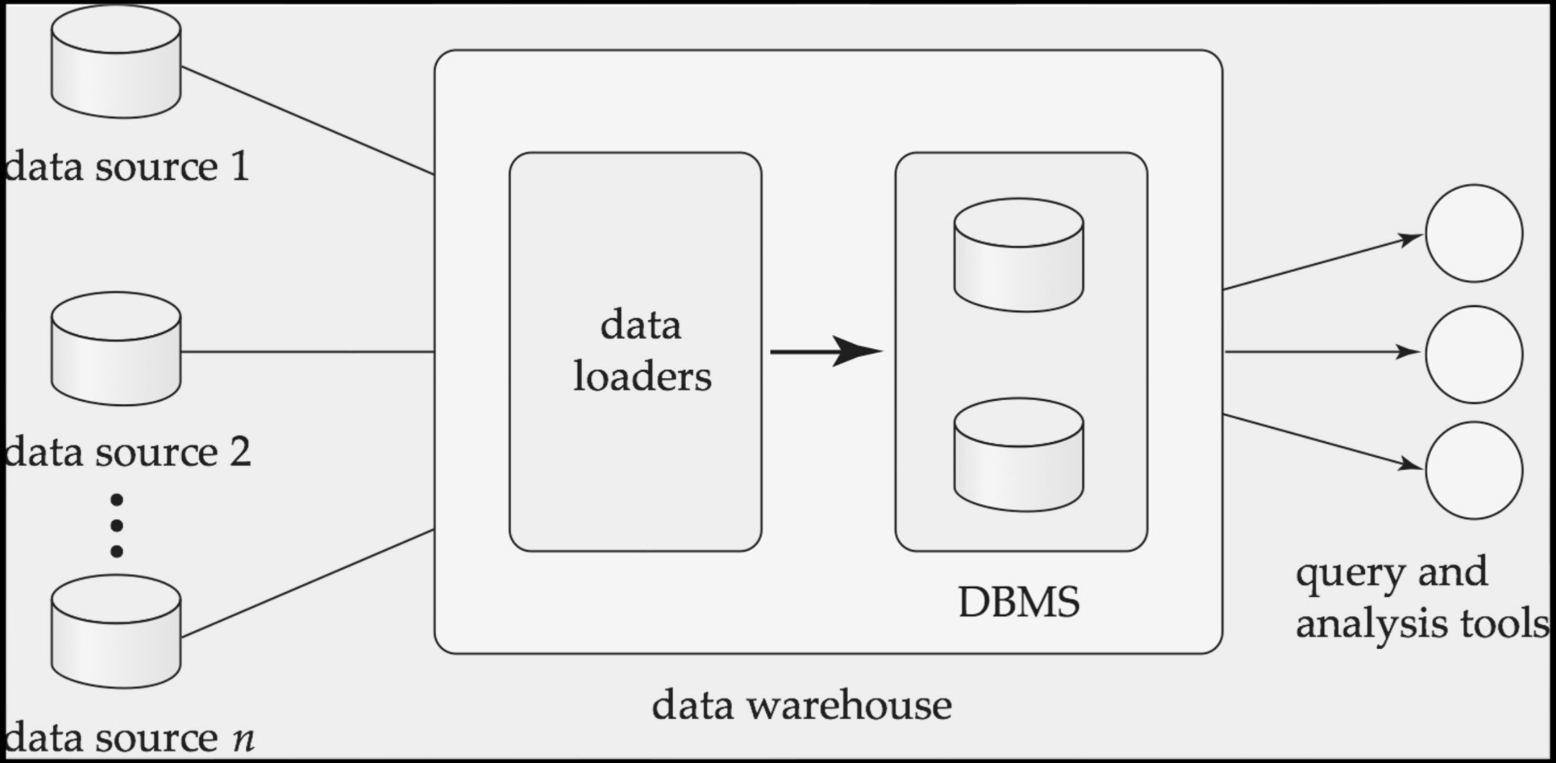
- A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake



Fact constellations:

- Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

Data Warehousing



Data Warehousing

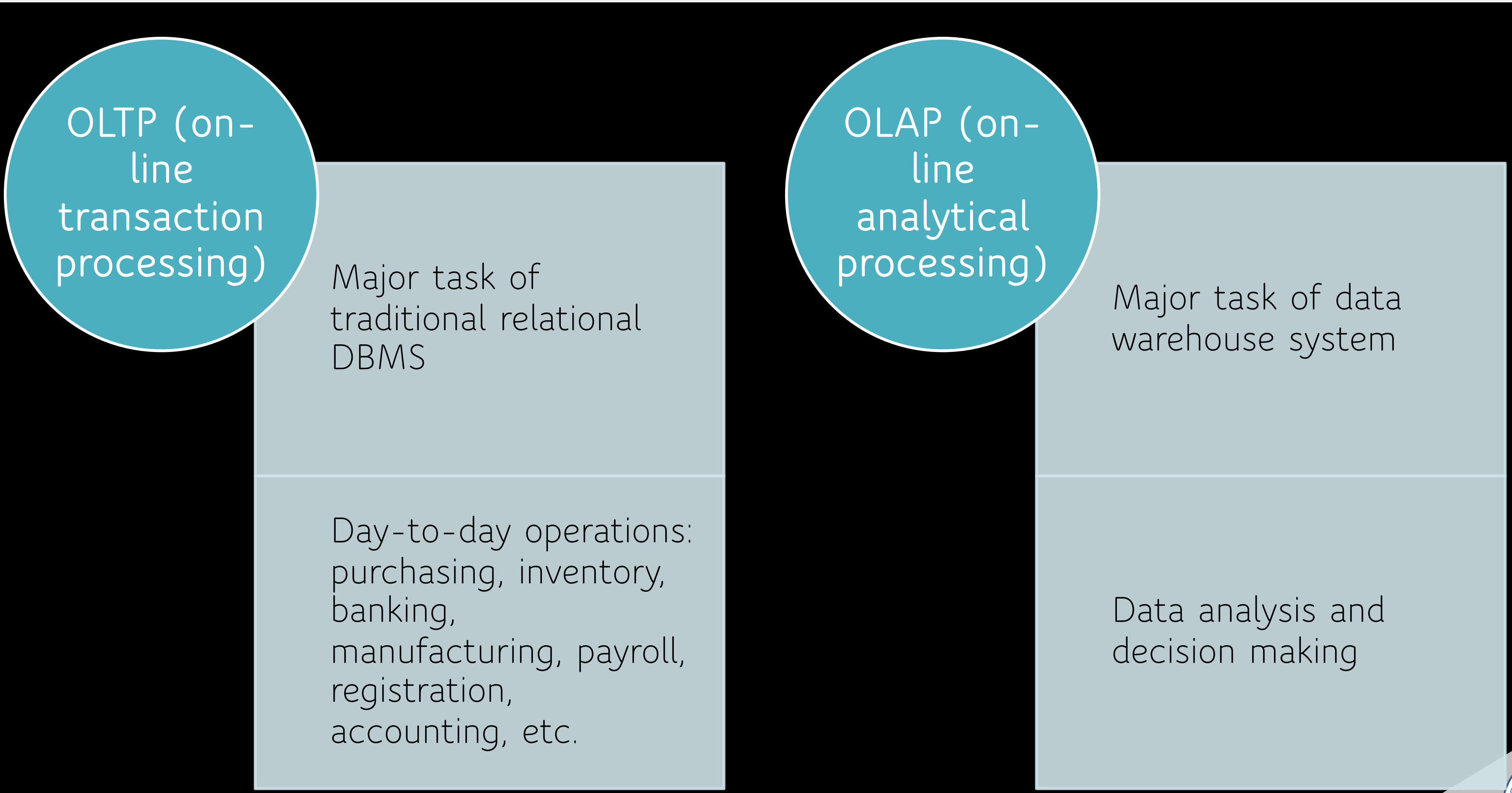
Data sources often store only current data, not historical data

Corporate decision making requires a unified view of all organizational data, including historical data

A **data warehouse** is a repository (archive) of information gathered from multiple sources, stored under a unified schema, at a single site

- Greatly simplifies querying, permits study of historical trends
- Shifts decision support query load away from transaction processing system

Data Warehouse vs. Operational DBMS #01



OLTP (on-line transaction processing)

Major task of traditional relational DBMS

Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.

OLAP (on-line analytical processing)

Major task of data warehouse system

Data analysis and decision making

Data Warehouse vs. Operational DBMS #02

Distinct features (OLTP vs. OLAP):

- **User and system orientation:** customer vs. market
- **Data contents:** current, detailed vs. historical, consolidated
- **Database design:** ER + application vs. star + subject
- **View:** current, local vs. evolutionary, integrated
- **Access patterns:** update vs. read-only but complex queries

OLTP vs. OLAP

	OLTP	OLAP
Users	Clerk, IT Professional	Knowledge worker
Function	Day to day operations	Decision support
DB Design	Application-oriented	Subject-oriented
Data	Current, up-to-date Detailed, flat relational Isolated	Historical Summarized, multi-dimensional Integrated, consolidated
Usage	Repetitive	Ad-hoc
Access	Read/Write Index/hash on primary key	Lots of scans
Unit of Work	Short, simple transaction	Complex query
# Records Accessed	Tens	Millions
# Users	Thousands	Hundreds
DB Size	100MB-GB	100GB-TB
Metric	Transaction throughput	Query throughput, response