Data Engineering Techniques (AID30010) — Unit II

Data Warehousing

Structured Question Bank

PART B: LONG ANSWER QUESTIONS (5 marks)

Introduction to Data Warehouse

Q1: What is a Data Warehouse? Explain its four key characteristics.

Answer:

1. Definition of Data Warehouse

- A single, complete, and consistent store of data obtained from a variety of different sources, made available to end users in a way they can understand and use in a business context — Barry Devlin
- A decision support database that is maintained **separately** from the organization's operational databases
- Supports information processing by providing a solid platform of consolidated, historical data for analysis
- W. H. Inmon's definition: "A data warehouse is a **subject-oriented, integrated, time-variant, and nonvolatile** collection of data in support of management's decision-making process."

2. Subject-Oriented

- Organized around major subjects such as customer, product, and sales
- Focuses on modeling and analysis of data for decision makers — not on daily operations or transaction processing
- Provides a simple and concise view around particular subject issues
- Excludes data that are not useful in the decision support process
- Example: A customer subject area connects all customer-related tables (1985–1991) by a common key — customer ID

3. Integrated

- Constructed by integrating multiple, heterogeneous data sources (relational databases, flat files, online transaction records)
- Data cleaning and data integration techniques are applied before loading
- Ensures consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources

- Example: Hotel price data from different sources may use different currencies, tax structures, and coverage — all must be standardized
- When data is moved to the warehouse, it is converted to a consistent form

4. Time-Variant

- Time horizon for data warehouse is significantly longer than operational systems
- Operational database: current value data (60–90 day time horizon)
- Data warehouse: provides information from a **historical perspective** — e.g., past 5–10 years
- Every key structure in the data warehouse contains an element of time, explicitly or implicitly
- As a result, data warehouse contains much more history than any other environment

5. Nonvolatile

- A physically separate store of data transformed from the operational environment
- Operational update of data does **not** occur in the data warehouse environment
- Does not require transaction processing, recovery, and concurrency control mechanisms
- Requires only two operations in data accessing: **initial loading of data** and **access of data**
- Data is loaded in a snapshot, static format — when changes occur, a new snapshot record is written (history is maintained)
- Operational data is updated record-by-record; warehouse data is loaded en masse and accessed but not updated

---

Q2: Explain the difference between OLTP and OLAP.

Answer:

1. What is OLTP?

- OLTP stands for **On-Line Transaction Processing**
- Major task of traditional relational DBMS
- Handles day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting
- Oriented toward clerks and IT professionals

2. What is OLAP?

- OLAP stands for **On-Line Analytical Processing**
- Major task of data warehouse systems
- Used for data analysis and decision making
- Oriented toward knowledge workers

3. Key Differences — OLTP vs. OLAP

Feature	OLTP	OLAP
Users	Clerks, IT professionals	Knowledge workers, managers
Function	Day-to-day operations	Decision support
DB Design	ER model + application-oriented	Star schema + subject-oriented
Data	Current, up-to-date, detailed, flat relational	Historical, summarized, multidimensional, integrated
Usage	Repetitive transactions	Ad-hoc complex queries
Access	Read/Write	Mostly read-only, complex queries
Records accessed	Tens (one at a time)	Millions (in bulk)
Number of users	Thousands	Hundreds
DB Size	100 MB – GB	100 GB – TB
Metric	Transaction throughput	Query throughput, response time

4. Why a Separate Data Warehouse?

- High performance for both systems — DBMS tuned for OLTP; warehouse tuned for OLAP
- **Missing data:** Decision support requires historical data that operational DBs do not typically maintain
- **Data consolidation:** Decision support requires aggregation and summarization from heterogeneous sources
- **Data quality:** Different sources use inconsistent representations, codes, and formats which must be reconciled

---

Q3: Explain the Data Warehouse Architecture (Three-Tier Architecture).

Answer:

1. Overview

- A data warehouse uses a **three-tier architecture** to manage data from multiple operational sources and deliver it to end users
- The three tiers are: Data Sources, Data Storage, and Front-End Tools

2. Tier 1 — Data Sources

- The bottom layer containing operational databases and external data
- Sources include: relational databases, flat files, online transaction records, external data

3. Tier 2 — Data Storage (Warehouse Server)

- The middle layer where actual data is stored and processed
- Data is extracted, cleaned, transformed, and loaded (ETL process) from source systems
- An **OLAP Server** processes analytical queries on warehouse data
- Also contains **Data Marts** — smaller, subject-specific subsets of the warehouse

4. Tier 3 — Front-End Tools

- The top layer that end users interact with
- Tools include: Analysis, Query & Reporting, Data Mining tools
- Allows users to perform OLAP operations, generate reports, and discover patterns

5. Back-End Tools and Utilities

- **Data Extraction:** Get data from multiple, heterogeneous, and external sources
- **Data Cleaning:** Detect errors and rectify them when possible
- **Data Transformation:** Convert data from legacy/host format to warehouse format
- **Load:** Sort, summarize, consolidate, compute views, check integrity, build indices and partitions
- **Refresh:** Propagate updates from data sources to the warehouse

6. Metadata Repository

- Metadata is the data defining warehouse objects — often called "data about data"
- Stores: description of structure of data warehouse (schema, view, dimensions, hierarchies)
- Stores: operational metadata — data lineage, currency of data (active/archived/purged), monitoring info
- Stores: algorithms used for summarization, mapping from operational environment to warehouse
- Stores: business data — business terms, definitions, ownership, charging policies

---

Q4: Explain the different Data Warehouse Models — Enterprise Warehouse, Data Mart, Virtual Warehouse, and Cloud-Based Warehouse.

Answer:

1. Enterprise Warehouse

- Collects all of the information about subjects spanning the **entire organization**
- Provides a comprehensive, organization-wide view of data

- Serves as the central warehouse for all business areas

2. Data Mart

- A subset of corporate-wide data that is of value to a **specific group of users**
- Its scope is confined to specific, selected groups — e.g., a marketing data mart
- Two types:
  - **Independent Data Mart:** Built directly from source data, without a central enterprise warehouse
  - **Dependent Data Mart:** Built directly from an enterprise data warehouse (more consistent and reliable)

3. Virtual Warehouse

- A set of views over operational databases
- Only some of the possible summary views may be materialized
- Easy to build but requires excess capacity on operational database servers

4. Cloud-Based Data Warehouse

- Hosted on cloud infrastructure (e.g., AWS Redshift, Google BigQuery, Snowflake)
- Can typically perform complex analytical queries much faster because they use **Massively Parallel Processing (MPP)**
- Highly scalable, cost-effective, and easily accessible

Data Warehouse Modeling

Q5: Explain the Multidimensional Data Model and the concept of a Data Cube.

Answer:

1. Multidimensional Data Model

- A data warehouse is based on a multidimensional data model which views data in the form of a **data cube**
- Managers think of business in terms of business dimensions
- Example: A marketing VP is interested in revenue broken down by month, division, customer, product version, sales office, etc.
- These breakdowns are the business dimensions

2. What is a Data Cube?

- A data cube allows data to be modeled and viewed in **multiple dimensions**
- Example: Sales can be viewed across the dimensions of Product, Location, and Time

- Each combination of dimension values defines a cell in the cube that contains a measure (e.g., dollars sold)

3. Dimension Tables and Fact Tables

- **Dimension tables:** Contain descriptive attributes about the dimensions
  - Example: item (item_name, brand, type), time (day, week, month, quarter, year)
- **Fact table:** Contains measures (such as dollars_sold) and foreign keys to each dimension table

4. Cuboids

- An n-D base cube is called a **base cuboid**
- The topmost 0-D cuboid (holds the highest level of summarization) is called the **apex cuboid**
- The lattice of cuboids (from base cuboid to apex) forms the full **data cube**
- Example: For dimensions {product, date, country}:
  - 0-D (apex): all
  - 1-D: {product}, {date}, {country}
  - 2-D: {product, date}, {product, country}, {date, country}
  - 3-D (base): {product, date, country}

5. Formula for Number of Cuboids

- For n dimensions each with Li levels (including "all"):
- Total cuboids = $\prod(Li + 1)$ for i = 1 to n
- Example: 4 dimensions each with 5 levels → (5+1)^4 = 1296... or (4+1)^4 = 625 cuboids

6. Concept Hierarchy in Dimensions

- Dimensions often have hierarchical levels
- Example — Location hierarchy: office → city → country → region → all
- Example — Time hierarchy: day → week → month → quarter → year → all
- Hierarchies allow roll-up and drill-down operations across levels

---

Q6: Explain the Star Schema, Snowflake Schema, and Fact Constellation Schema with their advantages and disadvantages.

Answer:

1. Problems with ER Model for Data Warehouses

- End users cannot understand or remember a complex ER model

- Many data warehouses have failed because of overly complex ER designs
- ER models are not optimized for complex, ad-hoc queries
- Data retrieval becomes difficult due to normalization
- Browsing and analysis become difficult

2. Star Schema

- Has a **single fact table** in the middle connected to a set of **dimension tables**
- Every fact points to one tuple in each of the dimensions and has additional attributes
- Does not capture hierarchies directly
- System-generated (surrogate) keys are used for performance and maintenance
- **Pros:**
    - Simple design — easy for users to understand
    - Fast read performance
    - Easy data aggregation
    - Easy integration with OLAP systems and data cubes
- **Cons:**
    - Redundant data makes for larger storage on disk
    - Potential for data anomalies, errors, and inconsistencies
    - Slower queries in some cases
    - Limited flexibility on non-dimensional data

3. Snowflake Schema

- Similar to star schema but each dimension table may have foreign keys relating to **other dimension tables** (dimension tables are normalized)
- A refinement of the star schema where some dimensional hierarchy is normalized into smaller dimension tables — forming a snowflake shape
- More normalized than star schema — reduces data redundancy
- Used to manage the size of dimension tables
- **Pros:**
    - Saves memory — less data redundancy
    - Normalized structures are easier to update and maintain
- **Cons:**
    - Increases complexity of the schema
    - Browsing becomes more difficult
    - Degrades performance because of additional joins required

4. Fact Constellation Schema (Galaxy Schema)

- Multiple fact tables share dimension tables
- Used when applications are complex and require multiple fact tables
- Viewed as a collection of stars — hence also called **Galaxy Schema**
- Example: A Sales Fact Table and a Shipping Fact Table both share the Time, Item, and Location dimension tables
- **Pros:**
  - Offers more flexibility
  - Multiple fact tables are explicitly assigned to dimension tables
- **Cons:**
  - Structure is more complex and sophisticated
  - Difficult to maintain
  - High number of aggregations

5. Summary Comparison

Schema	Fact Tables	Dimension Tables	Normalization
Star	One	Denormalized (flat)	Low
Snowflake	One	Normalized (multiple levels)	High
Fact Constellation	Multiple	Shared across fact tables	Medium

Q7: Explain the role of Metadata in a Data Warehouse.

Answer:

1. What is Metadata?

- Metadata in a data warehouse is **data that describes and contextualizes other data**
- It provides information about the content, format, structure, and other characteristics of data
- Considered the backbone of effective data management
- Often called "data about data"

2. Metadata — Data Understanding

- Helps users understand the data stored within the system
- Structural metadata describes how different elements of a compound data object are assembled

- Example: page layout of a document, chapter structure of an ebook

3. Metadata — Data Governance

- Helps ensure data accuracy and compliance
- Enforces the definition of business terms to business end-users
- Ensures that business rules and policies are applied consistently

4. Metadata — Data Integration

- Helps validate data transformation and ensure the accuracy of calculations
- Tracks how data moved from operational systems to the warehouse

5. Metadata — Data Accessibility

- Helps improve the accessibility of data
- A data catalog can combine metadata with data management and search tools to help analysts find data

6. What Metadata Stores

- Description of the structure of the data warehouse: schema, view, dimensions, hierarchies, derived data definitions, data mart locations and contents
- Operational metadata: data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (usage statistics, error reports, audit trails)
- Algorithms used for summarization
- Mapping from operational environment to the data warehouse
- Data related to system performance
- Business data: business terms and definitions, ownership of data, charging policies

---

Measures and OLAP Operations

Q8: Explain the three categories of Measures in a Data Cube.

Answer:

1. What are Measures?

- Measures are the numerical values stored in the fact table of a data warehouse
- They represent what is being tracked or analyzed (e.g., sales amount, count of orders)
- Measures are categorized based on how they can be computed across aggregate levels

2. Distributive Measures

- A function is distributive if the result derived by applying it to n aggregate values is the **same** as applying it on all data without partitioning
- In other words, they can be computed by dividing data into partitions and combining partial results
- Examples: **count(), sum(), min(), max()**
- These are the simplest and most efficient to compute in a data cube

3. Algebraic Measures

- Can be computed by an algebraic function with M arguments (where M is a bounded integer), each obtained by applying a distributive aggregate function
- Require some additional information beyond just the partial result
- Examples: **avg(), min_N(), standard_deviation()**
- avg() = sum() / count() — both sum and count are distributive, so avg is algebraic

4. Holistic Measures

- There is **no constant bound** on the storage size needed to describe a subaggregate
- Cannot be computed efficiently by combining partial results
- Examples: **median(), mode(), rank()**
- These are the most expensive to compute in a data cube

5. Importance of Categorization

- Knowing the category of a measure helps decide which ones can be efficiently pre-computed and materialized in the data cube
- Distributive and algebraic measures can be easily pre-computed for all cuboids
- Holistic measures are expensive and are typically computed only when needed

---

Q9: Explain the typical OLAP Operations on a Data Cube.

Answer:

1. What is OLAP?

- On-Line Analytical Processing — used for complex analysis and decision-making on data warehouse data
- OLAP operations allow users to view the data cube from different perspectives and at different levels of detail

2. Roll Up (Drill Up)

- Summarizes data by **climbing up the hierarchy** or by dimension reduction
- Moves from more detailed data to more summarized data

- Example: Rolling up from city → country → region in the location dimension
- Example: Rolling up from monthly sales to quarterly sales to annual sales
- Also possible by dimension reduction — dropping one dimension entirely

3. Drill Down (Roll Down)

- The reverse of roll-up — moves from **higher-level summary to lower-level detail**
- Introduces more detail or adds new dimensions
- Example: Drilling down from annual sales to quarterly to monthly sales
- Example: Drilling down from country → province → city

4. Slice

- Performs a **selection on one dimension** of the data cube, resulting in a sub-cube
- Fixes one dimension to a specific value and shows the remaining dimensions
- Example: Slice the cube where time = "Q1 2024" — shows sales for all products and locations in Q1

5. Dice

- Defines a **sub-cube by specifying a range or set of values** across two or more dimensions
- Like multiple slices at once
- Example: Dice where (time = "Q1" or "Q2") AND (location = "Canada" or "USA") AND (item = "TV" or "PC")

6. Pivot (Rotate)

- Reorients the cube for visualization
- Rotates the data axes to provide an alternative presentation of the data
- Example: Converting a 3D cube into a series of 2D planes for easier viewing

7. Other Operations

- **Drill Across:** Involves queries across more than one fact table
- **Drill Through:** Goes through the bottom level of the cube to its back-end relational tables using SQL
- Retrieves raw, detailed records from the warehouse

OLAP Server Architectures

Q10: Explain ROLAP, MOLAP, and HOLAP with their advantages and disadvantages.

Answer:

1. What are OLAP Server Architectures?

- Different implementations of OLAP servers that determine how data is stored, retrieved, and analyzed
- Three main types: ROLAP, MOLAP, and HOLAP

2. ROLAP — Relational OLAP

- Uses **relational or extended-relational DBMS** to store and manage warehouse data
- OLAP middleware is added on top of the relational database
- Includes optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
- Supports RDBMS products through a metadata layer — avoids the requirement to create a static multidimensional data structure
- Facilitates the creation of multiple multidimensional views of 2D relations
- **Advantage:** Greater scalability — can handle large data volumes
- **Disadvantages:**
    - Performance problems with complex queries requiring multiple passes through relational data
    - Requires development of middleware to support multidimensional applications
    - Slower for complex analytical queries compared to MOLAP

3. MOLAP — Multidimensional OLAP

- Uses specialized data structures and **multidimensional database management systems (MDDBMS)** to organize, navigate, and analyze data
- Sparse array-based multidimensional storage engine that minimizes disk space through sparse data management
- Fast indexing to pre-computed summarized data
- **Advantage:** Very fast query performance due to pre-computed aggregations
- **Disadvantages:**
    - Only a limited amount of data can be efficiently stored and analyzed
    - Navigation and analysis are limited because data is designed according to pre-determined requirements
    - Requires a different set of skills and tools to build and maintain the database

4. HOLAP — Hybrid OLAP

- Combines features of both ROLAP and MOLAP
- Provides limited analysis capability either directly against RDBMS products or via an intermediate MOLAP server
- Delivers selected data directly from DBMS or via MOLAP server to desktop as a data cube

- Example: Microsoft SQL Server
- **Advantage:** Flexibility — low-level data stored relationally; high-level aggregations stored as arrays
- **Disadvantages:**
  - Significant data redundancy — may cause problems for networks with many users
  - Each user building a custom data cube may cause lack of data consistency
  - Only limited amount of data can be efficiently maintained

5. Other OLAP Types

- **WOLAP (Web-Enabled OLAP):** OLAP accessible via a web browser — three-tiered architecture (client, middleware, database server)
- **DOLAP (Desktop OLAP):** User downloads a section of data and works with it locally on their desktop
- **MOLAP (Mobile OLAP):** Users access OLAP data and applications remotely through mobile devices
- **SOLAP (Spatial OLAP):** Combines GIS (Geographic Information Systems) capabilities with OLAP — manages both spatial and non-spatial data

6. Summary Comparison

Feature	ROLAP	MOLAP	HOLAP
Storage	Relational DB	Multidimensional arrays	Both
Performance	Moderate	Fast	Moderate-Fast
Scalability	High	Limited	Moderate
Flexibility	High	Low	High

Data Warehouse Design

Q11: Explain the Data Warehouse Design Process — Top-Down and Bottom-Up approaches.

Answer:

1. Four Views for Data Warehouse Design

- **Top-down view:** Allows selection of the relevant information necessary for the data warehouse
- **Data source view:** Exposes the information being captured, stored, and managed by operational systems
- **Data warehouse view:** Consists of fact tables and dimension tables
- **Business query view:** Sees the perspectives of data from the end-user's view

2. Top-Down Approach

- Starts with the overall design and planning of the entire enterprise warehouse
- A mature, comprehensive approach — the whole enterprise warehouse is designed first, then data marts are built from it
- Also follows the **Waterfall** method: structured and systematic analysis at each step before proceeding to the next
- **Advantages:**
  - Easier to maintain
  - Provides consistent dimensional views of data across data marts (all data marts loaded from the data warehouse)
  - Robust against business changes — creating a new data mart from the warehouse is easy
  - Initial cost is high but subsequent project development cost is lower
- **Disadvantages:**
  - Represents a very large project — significant cost of implementation
  - Time consuming — more time required for initial setup
  - Highly skilled people required

3. Bottom-Up Approach

- Starts with experiments and prototypes — builds individual data marts first, then integrates them into an enterprise warehouse
- Follows the **Spiral** method: rapid generation of increasingly functional systems with quick turnaround
- **Advantages:**
  - Data marts can be delivered quickly
  - Data marts are created first to provide reporting capability
  - Easier to extend — just create new data marts and integrate
  - Less time required — initial setup is very quick
- **Disadvantages:**
  - Initial cost is low but each subsequent phase costs the same
  - The positions of the data warehouse and data marts are reversed
  - Difficult to maintain and often redundant and subject to revisions

4. Typical Data Warehouse Design Steps

1. Choose a business process to model (e.g., orders, invoices, claims)
2. Choose the grain (atomic level of data) of the business process
3. Choose the dimensions that will apply to each fact table record
4. Choose the measures that will populate each fact table record

Q12: Explain Materialized Views and Efficient Data Cube Computation.

Answer:

1. What is Materialization?

- A data cube can be viewed as a lattice of cuboids — from the base cuboid (most detailed) to the apex cuboid (most summarized)
- **Materialization** refers to pre-computing and storing some or all of the cuboids for faster query processing

2. Types of Materialization

- **Full Materialization:** Materialize every possible cuboid — fastest query response but requires the most storage
- **No Materialization:** No cuboids are pre-computed — smallest storage but slowest query response (everything computed on demand)
- **Partial Materialization:** Materialize only selected cuboids that are frequently accessed — a balance between storage and performance

3. Selection of Which Cuboids to Materialize

- Based on:
  - **Size** of the cuboid
  - **Sharing** — cuboids that can be shared across multiple queries
  - **Access frequency** — cuboids that are queried most often

4. Number of Cuboids in a Lattice

- Formula: $T = \prod(Li + 1)$ for $i = 1$ to n
- Where T = total cuboids, n = number of dimensions, Li = number of levels in dimension i
- Example: 4 dimensions each with 5 levels → $T = (5+1)^4 = 1296$... or with 4 levels → $T = (4+1)^4 = 625$

5. Efficient Processing of OLAP Queries

- Determine which operations should be performed on the available cuboids
- Transform drill, roll, slice, dice operations into corresponding SQL and/or OLAP operations
- Determine which materialized cuboid(s) should be selected for each OLAP operation
- Always use the most specific materialized cuboid that satisfies the query to minimize computation

6. What are Materialized Views?

- Pre-computed query results stored physically in the database
- When a query is received, the system checks if a materialized view already contains the result
- Significantly speeds up repeated or common analytical queries
- Must be refreshed periodically to reflect changes in source data

---

Data Lake and Data Lakehouse

Q13: Explain Data Lakes, Data Lakehouses, and compare them with Data Warehouses.

Answer:

1. What is a Data Lake?

- A Data Lake is a **centralized repository** that stores raw data in its original format at any scale
- Think of it like a big lake where all kinds of data flow in — you decide later how to process and analyze it
- **Types of Data Stored:**
  - Structured → tables, CSV, RDBMS data
  - Semi-structured → JSON, XML, logs
  - Unstructured → images, videos, audio, PDFs, text

2. Problems with Data Lakes

- No ACID transactions — data consistency cannot be guaranteed
- Data quality issues — raw, unprocessed data may be unreliable
- Can become a **"data swamp"** — too much unorganized data with no clear structure or governance

3. What is a Data Lakehouse?

- A Data Lakehouse is a **hybrid data architecture** that combines the flexibility of a Data Lake with the performance, reliability, and structure of a Data Warehouse
- Formula: Data Lake + Data Warehouse = Data Lakehouse
- Solves the limitations of both Data Lakes (poor data quality) and Data Warehouses (expensive, rigid schema)

4. Data Lakehouse Architecture (Layered)

- **Storage Layer:** Cloud object storage (S3, ADLS, GCS); open formats like Parquet and ORC
- **Metadata & Table Format Layer:** Delta Lake, Apache Iceberg, Apache Hudi
- **Processing Layer:** Apache Spark, Flink, Trino, Presto
- **Analytics & BI:** Power BI, Tableau, Looker

- **ML & AI:** MLflow, TensorFlow, PyTorch
- **Governance & Security:** Access control, versioning, auditing

5. Key Features of Data Lakehouse

- ACID transactions — ensures data consistency
- Schema enforcement and evolution
- Time travel (data versioning) — can query data as it was at a past point in time
- High query performance
- Single source of truth
- Supports both BI (Business Intelligence) and ML (Machine Learning) together

6. Comparison — Data Lake vs. Data Warehouse vs. Data Lakehouse

Feature	Data Lake	Data Warehouse	Data Lakehouse
Data Type	All (structured, semi, unstructured)	Structured only	All
Schema	On-read (schema defined at query time)	On-write (schema defined before loading)	Flexible (both)
ACID Transactions	❌ No	✅ Yes	✅ Yes
Cost	Low	High	Medium
BI Support	Limited	Excellent	Excellent
ML Support	Excellent	Limited	Excellent

ADDITIONAL PRACTICE QUESTIONS

Q14: Explain the Data Warehouse vs. Heterogeneous DBMS and why an integrated data warehouse is preferred.

Answer:

1. The Problem — Heterogeneous Information Sources

- "Heterogeneities are everywhere" — different data sources use different interfaces, different data representations, and may have duplicate and inconsistent information

- Sources include: personal databases, scientific databases, digital libraries, World Wide Web
- Problems faced:
    - Cannot find the data needed — data is scattered over the network; many versions with subtle differences
    - Cannot get the data needed — need an expert to retrieve it
    - Cannot understand the data found — available data is poorly documented
    - Cannot use the data found — results are unexpected; data needs to be transformed

2. Traditional Heterogeneous DB Integration — Query-Driven

- Build wrappers/mediators on top of heterogeneous databases
- When a query is posed, a meta-dictionary translates the query into queries for individual heterogeneous sites
- Results are integrated into a global answer set
- **Problem:** Complex information filtering; competes for resources; slow performance for complex queries

3. Data Warehouse — Update-Driven Approach

- Information from heterogeneous sources is **integrated in advance** and stored in the warehouse
- Data is available for direct query and analysis at any time
- Much faster and more efficient for analytical queries
- **Advantage:** High performance — data is already integrated and often pre-aggregated

4. Unified Access Benefits

- Collects and combines information from all sources
- Provides an integrated view with a uniform user interface
- Supports sharing of data across the organization

Q15: Explain the concept of Dimensions and Concept Hierarchies in a Data Warehouse with examples.

Answer:

1. What are Dimensions?

- Dimensions are the perspectives or entities with respect to which an organization wants to keep records
- They define the context for the measures (facts) stored in the fact table
- Example: A sales data cube may have dimensions — time, item, branch, and location

2. Business Dimensions — Examples

- **Marketing VP:** Interested in revenue broken down by month, division, customer, demographic, sales office, product version, plan
- **Marketing Manager:** Dimensions are product, product category, time (day/week/month), sale district, distribution channel
- **Financial Controller:** Dimensions are budget line, time (month/quarter/year), district, division

3. Example Dimensions for Insurance Claim Analysis

- Time, Agent, Claim, Insured Party, Policy, Status

4. Example Dimensions for Airline Flyer Flight Analysis

- Time, Customer, Flight, Fare Class, Airport, Status

5. What are Concept Hierarchies?

- A concept hierarchy defines a sequence of mappings from a set of lower-level concepts to higher-level, more general concepts
- They allow data to be viewed at different levels of abstraction
- Used for roll-up and drill-down operations in OLAP

6. Example — Location Concept Hierarchy

- office → city → country → region → all
- Example: L. Chan (office) → Vancouver (city) → Canada (country) → North_America (region) → all

7. Example — Time Concept Hierarchy

- day → week → month → quarter → year → all

8. Example — Product Concept Hierarchy

- product → category → industry → all

9. Hierarchical Summarization Paths

- Industry → Category → Product → (detailed)
- Region → Country → City → Office
- Year → Quarter → Month → Week → Day
- These paths allow managers to summarize and drill through data at any desired level of detail

---

Q16: Explain the Data Warehouse Design Process with a complete example (Star Schema for a University).

Answer:

1. Design Framework Steps

1. Choose a business process to model — e.g., student grade tracking at a university
2. Choose the grain (atomic level) — e.g., a grade for a specific student in a specific course taught by a specific instructor in a specific semester
3. Choose the dimensions — student, course, semester, instructor
4. Choose the measures — count (number of records), avg_grade (average grade)

2. Identifying Dimensions and Levels

- **Student:** student → major → status → university → all
- **Course:** course → dept → subject → school → all
- **Semester:** semester → term → year → program → all
- **Instructor:** instructor → dept → school → faculty → all
- Each dimension has 4 levels (plus "all" = 5 levels total)

3. Number of Cuboids

- T = (4+1)^4 = 5^4 = **625 cuboids**

4. Typical Analytical Queries

- What was the total revenue (or count) by month or year?
- Which course had the highest average grade?
- Do seniors have different grades than freshmen for the same course?
- How does average grade vary by instructor or department?
- Which student segments (by major) perform best?

5. OLAP Operations for a Specific Query

- Query: "List the average grade of CS courses for each Big University student"
- Starting from base cuboid: [student, course, semester, instructor]
- Operations:
    1. Roll-up on **course** from course_id to **department**
    2. Roll-up on **student** from student_id to **university**
    3. **Dice** on course and student with department = "CS" AND university = "Big University"
    4. Drill-down on **student** from university back to **student name**

Q17: Explain the three kinds of Data Warehouse Applications.

Answer:

1. Information Processing

- Supports querying, basic statistical analysis, and reporting
- Uses crosstabs, tables, charts, and graphs
- Straightforward retrieval and presentation of warehouse data
- Example: Monthly sales reports, product inventory summaries

2. Analytical Processing

- Multidimensional analysis of data warehouse data
- Supports basic OLAP operations: slice-dice, drilling (roll-up/drill-down), and pivoting
- Allows managers to explore data from multiple perspectives
- Example: Analyzing sales by product, region, and time simultaneously

3. Data Mining

- Knowledge discovery from hidden patterns in data
- Supports:
  - Associations (e.g., products bought together)
  - Constructing analytical models
  - Performing classification and prediction
  - Presenting mining results using visualization tools
- Example: Predicting which customers are likely to leave (churn prediction), discovering buying patterns

4. Why All Three Are Important

- Information processing answers: "What happened?"
- Analytical processing answers: "Why did it happen?" and "What if?"
- Data mining answers: "What will happen?" and "What patterns are hidden?"
- Together, they support complete business intelligence and decision-making

---

Summary

This question bank covers all major topics in Unit II with:

- 17 Long Answer Questions (5 marks each)
- Complete coverage of all Unit II concepts including:

    - Data Warehouse definition and four key characteristics (Subject-Oriented, Integrated, Time-Variant, Nonvolatile)
    - OLTP vs. OLAP comparison in detail
    - Three-tier Data Warehouse Architecture
    - Data Warehouse Models: Enterprise, Data Mart, Virtual, Cloud-based
    - Multidimensional Data Model and Data Cubes
    - Cuboid lattice and formula for number of cuboids
    - Star Schema, Snowflake Schema, Fact Constellation Schema with pros/cons
    - Role of Metadata in Data Warehouses
    - Three Categories of Measures: Distributive, Algebraic, Holistic
    - Typical OLAP Operations: Roll-up, Drill-down, Slice, Dice, Pivot
    - OLAP Server Architectures: ROLAP, MOLAP, HOLAP
    - Top-Down and Bottom-Up Design Approaches
    - Materialized Views and Efficient Data Cube Computation
    - Concept Hierarchies and Dimensions
    - Data Lakes, Data Lakehouses, and comparison with Data Warehouses
    - Three kinds of Data Warehouse Applications
- Each answer is structured for a 5-mark exam response
- Simple, student-friendly language throughout
- Practical examples and comparisons included wherever possible