Q.1 Differentiate between OODBMS and ORDBMS

Features	Object-Oriented DBMS (OODBMS)	Object-Relation DBMS (ORDBMS)
Data Model	Based completely on the Object Oriented model, where data is stored as objects similar to programming languages like Java or C++	Based on the relational model but enhanced with object oriented features like inheritance and user defined types
Structure	User object, class and inheritance to represent real-world entities	Uses tables (relations) but can include complex data types and object references
Query Language	Uses OQL (Object Query Language) for accessing objects	Uses extended SQL that supports object features (like SQL:1999)
Data Storage	Store both data and behavior (methods) together within the object	Primarily stores data, but allows methods or functions through object types
Relationship	Object are linked using object identifiers or pointer	Tables are linked using primary key and foreign key
Usage	Used in engineering, multimedia and CAD/CAM system where complex data is handled	used in business and commercial application where both relational and object data are needed
Example	db4o, Objectstore, Versant ODBMS	Oracle 10g, PostgreSQL, Informix Universal Server

Q.2 Explain the snowflake schema with a suitable example

A snowflake schema is a type of multidimensional database schema used in data warehousing where the dimension tables are normalized into related tables. It is an extension of the star schema, designed to reduce data redundancy and improve data integrity through normalization.

In a snowflake schema the fact table (which stores measurable, quantitive data like sales amount, revenue, etc) is linked to dimension table (which stores descriptive information such as product, customer, time, etc)

However, unlike the star schema, the dimension tables in a snowflake schema are split into smaller sub-dimension tables forming a structure that resembles a snowflake.

Key points of snowflake schema

(i) Normalization:
Dimension tables are normalized into multiple related tables to remove redundancy

(ii) Structure:
The central fact table connects to primary dimension tables and those dimensions futher connect to sub-dimensions

FOR EDUCATIONAL USE

(iii) Storage Efficiency :
It saves storage space because of normalization but may increase query complexity.

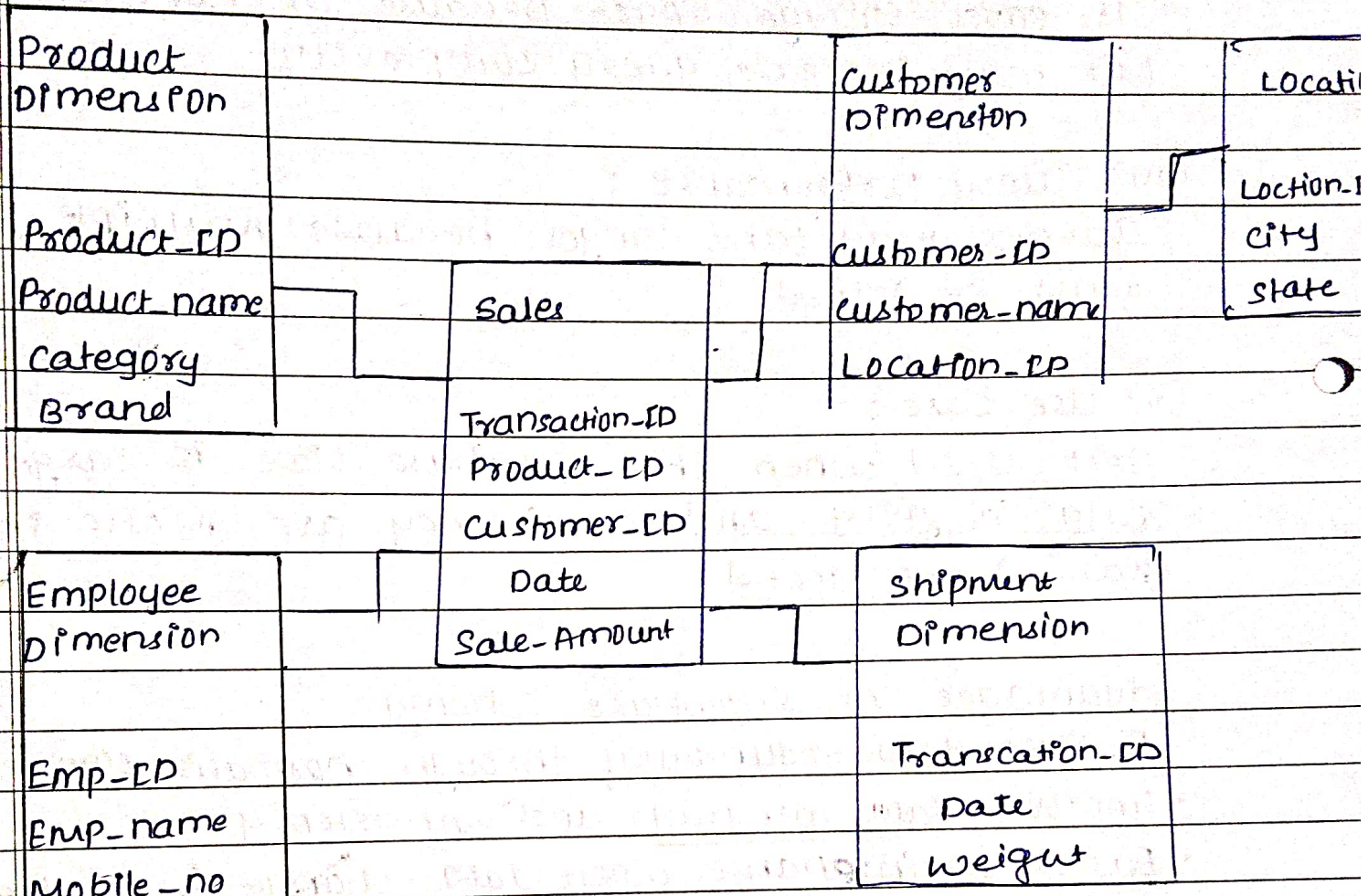
(iv) Query Performance :
Queries may take longer because multiple tables must be joined

(v) Use Case :
Best used when the database size is large and data integrity and consistency are more important than query speed

Advantages of Snowflake schema
- Reduce data redundancy through normalization.
- Improves data integrity and consistency.
- Easier maintenance when data change (e.g. category nam
- More efficient storage because data isn't duplicated

Disadvantages of Snowflake schema
- Complex queries due to multiple table joins
- slower query performance compared to star schema
- More complicated design and understanding

Example



The diagram illustrates a snowflake schema, characterized by central sales fact table linked to normalization dimension tables. The normalization is visible where the customer Dimension breaks down attributes li Location into seperate table

Q. 3 What is Inheritance in OODBMS ?

Inheritance is an OODBMS (Object-Oriented Database Management System) is a mechanism that allows a new class (sub class) to inherit the properties (attributes) and behaviors (methods) of an existing class (superclass)

It is one of the key features of OODBMS, which enables ~~resu~~ reusability, data consistency. and hierarchical organization of object

Key Aspects:

(i) Code and Structure Reuse:
Prevents redundant definitions. Example if you have a Person class with attributes like Name and Address you can create a Student class and an Employee class that automatica inherit those properties. This means you only define Name and Address one in the parent class.

(ii) Schema Hierarchy:
The database schema forms a hierarchy of classes. An object of a subclass is also an instance of its superclass [e.g. a student is also a Person]

(iii) Data Persistence:
When a subclass object (like student) is stored, the database keeps both its own attributes and the inherit ones as a single persistent object.

(iv) Polymorphism:
The same method call can behave differently depending on the class (e.g. DisplayInfo() may differ for Student and Employee)

(v) Attribute Overriding:
Subclasses can redefine inherited attributes to be more specific

(vi) Method Overloading/overriding:
Subclasses may override parent methods to provide specialized behavior (e.g. different CalculatePay() methods for Student and Employee).

(vii) Type/Subtype Relationship
Ensure that operations valid for the superclass are also valid for its subclass, preserving consistency

(viii) Dynamic Binding (Run-Time Dispatch)
The actual method executed is determined at runtime based on the Object's real type

Example: Vehical Hierarchy
A database managing Vehicles might define a superclass Vehicle with attributes like Vehicle_ID, Manufactures and Year and methods such as StartEngine() and CalculateDepreciation()
Subclass like Car, Truck and Motorbike inherit these and may override certain methods to represent their unique behaviors.

Q.4 Explain Roll-up, Drill-down, Slice, Dice and Pivot Operations

1. Roll-up
- Roll-up is an aggregation operation that summarizes data by moving up a hierarchy in a dimension or reducing the number of dimensions
- It provides a higher-level view and helps spot overall trends.
- Removes granularity, making the dataset smaller and aggregated values larger.
- Can also be done by reducing the number of dimensions considered
- Example: Sales data by Pincode → Roll-up to City level. Pincodes 400701, 400702, 400703 combine to show total sales for Navi Mumbai.

2. Drill-down
- Drill-down is the inverse of roll-up, allowing analysts to view more detailed data by descending a hierarchy
- Adds granularity and makes the view more descriptive
- Can also be performed by introducing a new dimension

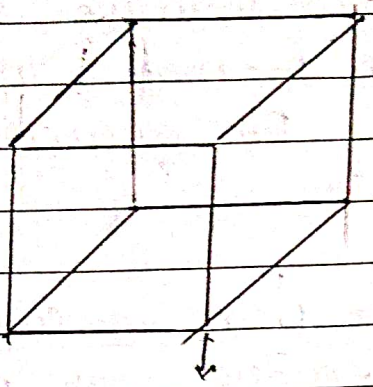
Example: Q3 sales appear low → Drill down by Month (July, August, September) → Drill further to Week in August → Identify weak sales week.

3. Slice
- Slice fixes one dimension at a single value, resulting in a sub-cube smaller than the original.
- Help filter large datasets to a focused view.
- Creates a cross-sectional view by isolating data based on ~~once~~ one criterion.
Example: 4D cube with Time, Product, Region, Customer Age → slice for Product 'Smartphone X' → 3D sub-cube shows sales for that product across Time, Region and Customer Age Group.

4. Dice
- Dice selects specific values or range across two or more dimensions, creating a smaller sub-cube.
- Similar to slice but involves multiple dimensions
- Results in a smaller, multi-dimensional cube ready for analysis
Example: 4D sales cube → Dice for Time = Q1 and Q2 AND Region = South and West → Sub-cube shows sales data meeting both criteria

5. Pivot
- Pivot rotates data axes, changing the visual layout without altering content or aggregation.
- Allows dimensions to be displayed along row, column or filters.
- Helps reveal new insights and comparisons
Example: Report with Time (Months) as rows, Region as columns → Pivot → Region as rows, time as columns → Easier to compare monthly trends with a region.

Roll-up
summarizes data by moving up a hierarchy



Slice
Create a sub-cub for one dimension

Drill-down
moves down hierarchy for more details

Dice
Creates a sub-cube for one dimension

Pivot
Rotates the data axes

Q.5 Write a short note on the ETL Process

ETL is stand for Extract, Transform, Load. It is data integration process that extract large volumes of raw data from multiple sources, transform it into a standard format and load it into a data warehouse for analysis and reporting

1. Extraction :
- Data is extracted from various sources such as databases, files, emails, web pages or application
- Extracted data is placed in a staging area, which acts as a buffer between source system and the data warehouse
- The staging area allows data cleaning and preparation before loading into the warehouse.

Challenges : Differentiating structured and unstructured data; handling multiple formats

Importance : Directly loading data into a warehouse is not feasible due to varying formats, so extraction into a staging area is crucial.

2. Transformation :
- Extracted data is converted into a consistent format for the data warehouse.
- Transformation improves data quality, normalization and compliance

Methods for transformation include :

Filtering : Load only selected attributes

Cleaning : Handle NULL values, standardize values (e.g. USA, United States → USA).
Joining : Combine multiple attributes into one
Splitting : Divide a single attribute into multiple attributes
Sorting : Arrange data based on a key attribute.

3. Loading
- Transformed data is loaded into the data warehouse
- Frequency depends on system requirement : can be frequent or at regular intervals.

Types of Loading
1. Initial Load : First-time population of the data warehouse, including historical and current data.
   • Techniques :- Load Mode (single run) or Load Mode + Append Mode (multiple runs)
2. Incremental Load : Periodic loading of new or updated records without reloading all data
   • Techniques :- Constructive Merge Mode (adds new data while keeping history).

3. Full Refresh : Delete old data and reloads everything from ~~sto~~ scratch.
   • Techniques : Load Mode + Append Mode.