Chapter #6:-

Conceptual Design

⇒ At conceptual design phase, it is decided that what kind of information the data mart will hold and how it will be organized.

Three main ways to approach conceptual design:

- Data driven
- Requirement driven
- Mixed approach

Goal of Conceptual design Phase:
To create a data mart that is:

Efficient: easily updated and accessed.

Relevant: Provides information that users need.

Consistent: Align with existing data sources and business processes.

Entity-Relationship schema-based Design:

Data-driven Conceptual design:

⇒ The technique used to derive complaint conceptual design of a data mart based on an operational source Entity-relationship schema includes the following steps:

1. Define facts
2. For each fact:
   (a) Build an attribute tree
   (b) Prune & graft attribute tree
   (c) Define dimensions
   (d) Define measures
   (e) Create a fact schema

1- Define facts:

- identify the important facts from source ER schema that will include in data mart.
- Each fact identified in a source schema becomes the root of different fact schema.

2. Building attribute trees:

⇒ Create a hierarchical structure attributes for each fact to organize & identify relevant information.

Steps to create attribute tree:

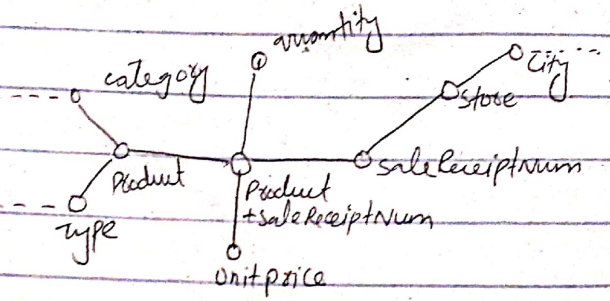
1. Create a root node that represent the fact itself.
2. Identify dimensions
3. Create child nodes.
4. Add more level.

⇒ Each node corresponds to a source schema attribute.

⇒ The root corresponds to the identifier of the F entity.

Ex:

Attribute tree for the sale:



Sale
- Time
  - Year
  - Quarter
  - Month
  - Day
- Product
  - category
  - Brand
  - Product ID
- Customer
  - Customer ID
  - Region
  - Age
- Store
  - store ID
  - Location

Why important AT?

⇒ Data Organization
⇒ Drill-down analysis
⇒ Efficient Querying
⇒ improved data quality

Prune and Graft Tree:

⇒ Refine the attribute tree by removing unnecessary attributes and adding missing ones to ensure it accurately represent the data that need.

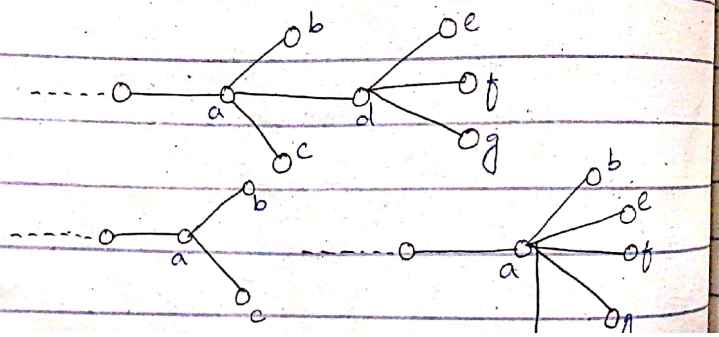
⇒ This is done, because all the attributes in the tree are not relevant to the data mart.

⇒ Therefore, tree is manipulated to delete unnecessary levels of detail.

⇒ To remove node v, should remove the entire sub-tree rooted in v.

⇒ Grafting is need to retain the descendents of a node in tree.

⇒ To graft a node v, whose parent is called v', should link all the children of v directly to v' and then remove v.



One-to-One Relationships:

⇒ A one-to-one relationship may be seen as a special case of a many-to-one relationship.

⇒ If issue an OLAP query to drill down a one-to-one association, this does not add further useful detail.

⇒ These one-to-one relationships can be removed by either:

Grafting: if the attribute has relevant descendents, it can be grafted onto its parent node.

Pruning: if the attribute has no relevant descendents, it can be removed entirely.

Defining Dimensions:

Dimensions are used to categorize and segment data.

They are typically selected from the root child nodes of the attribute tree.

When selecting dimensions, it is important to consider the desire

level of granularity.
⇒ Selecting dimensions is crucial design because it defines the granularity of primary events.

Timing Dimension:

⇒ A fact schema should always contain at least one time dimension.
⇒ Source operational databases can be classified into:
- Transient → define current state of an application domain
- Temporal → define evolution of an application domain

depending on the way they are related to time.
⇒ it allows for analysis of data over specific time periods.

Defining measures:

⇒ Measures are quantitative values associated with facts. They are used to analyze and understand the data.
⇒ Not all the measures can be aggregated using simple arithmetic operations like SUM or average. These are called non-additive measures.
⇒ To identify non-additive measures, determine can the measure be aggregated across different dimensions if not then measure is likely non-additive.
for example, The average price of a product cannot be simply summed up to get the average price of a group of products.
Handling non-additive measures:
- Store detailed data at the lowest level of granularity.
- Use appropriate aggregation functions
- Create calculated measures.

Generating fact schema:-

once the dimensions and measures are defined next step is to create fact schema. This involves:
Translating the attribute tree into a fact schema.
The dimensions and measures are assigned to the fact schema.
Identify fact granularity to store data in fact table.

⇒ if there are cross dimensional attributes, then linked them to appropriate entities.

Lossless Grained fact schema

⇒ When the granularity of the fact matches the granularity of the source data, then directly link descriptive attributes to the fact.

Lossy-Grained fact schema

⇒ When the granularity do not match, prune or aggregate attributes to avoid ambiguity. This can result in loss of detail.

Relational-database schema

⇒ How to structure a database to efficiently store and retrieve information.

ex:

Relational schema for the sales operational database:

PRODUCTS (Product, weight, size, dist. brand : BRANDS , Type : TYPE)

1. Defining facts:

⇒ In relational schema, a fact corresponds to a relation.

- for ex, a fact product sale is represented by the SALES relation.

2. Building attribute Trees

⇒ When design source is the relational schema for operational database, then attribute tree will be built as follows.

- Each node corresponds to one or more schema attributes.
- The root corresponds to the primary key of F.

- For each node v, the corresponding attributes functionally determines all the attributes that correspond to the descendants of v.

⇒ The tree building procedure is based on the principle of functional dependencies.

Other Phases:

⇒ Other Phases are same like Entity-Relationship schema design.

- Retain or graft any nodes
- Modify, add or delete functional dependency.

XML Schema-based Design

⇒ Data that is used in decision-making processes can be stored in XML form.

⇒ The structure of XML consists of nested tags, defined by user. These tags can define the meaning of the data represented.

⇒ XML design is suitable for exchanging data on web without losing semantics.

⇒ XML is considered as a special syntax for the exchange of semi-structured data.

⇒ Conceptual design of data models from XML sources presents two basic problems:

- The first problem is that there are different ways to represent how elements are connected in DTDs and XML schemas. This can make it tricky to understand the relationship between data.

- The second problem is that XML data is often not fully structured, so it can be hard to extract the necessary information for designing a data mart.

1. Modeling XML Associations:

⇒ XML associations, are connections between different parts of an XML document.

⇒ They can be represented in different ways, such as using sub-elements or attributes.

DTD & XML schema:

⇒ These are like blueprints for XML documents, defining their structure and rules.

⇒ An XML schema is validated if it has an associated schema (DTD & XML).

⇒ A DTD defines:

- The elements and attributes that an XML document allows.
- Element nesting rules
- Element occurrences

ID, IDRET & IDREFS:

These are special types of attributes used to create associations between elements.

2- Modeling preliminary Phases:

⇒ Before choose a fact and build it Specific Tree, there should be some preliminary Phases:

- Simplify source DTD
- Create a DTD graph

⇒ Sub-elements in DTD may be stated in complex and redundant way, if this, then simplify DTD through transformation — that convert nested representation into flat representation.

⇒ Then DTD graph is created that visually represent the structure of the simplified DTD.

⇒ This graph helps in identifying the key elements and attributes for the data mart.

3. Selecting fact and Build attribute trees:

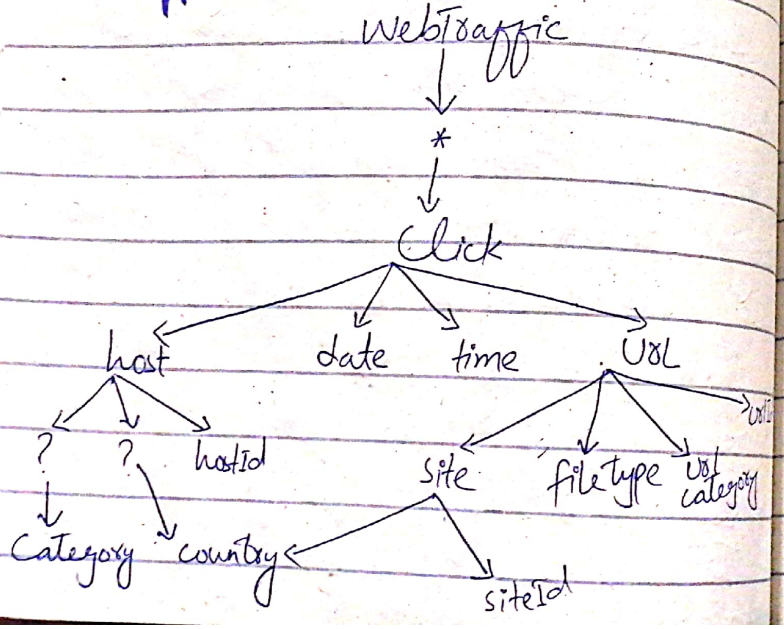
⇒ Designers choose one or more DFG graph nodes as facts.

⇒ Each chosen node becomes root of an attribute tree.

⇒ The tree is built by following the relationships in the DFD graph considering cardinality and functional dependencies.

⇒ This tree form the basis for data marts.

ex: DFD graph for web traffic R/S



Mixed-approach Design

⇒ This approach combines different techniques for data warehouse design

⇒ it uses diagrams to visualize relationships between different parts of the data warehouse.

⇒ In the conceptual design phase pairs the requirements derived from organizational and decision-making needs with source operational schema to get a conceptual schema for data mart. The procedure is divided into three phases:

1. Requirement mapping Phase:

The facts, dimensions and measures found in the decision making modeling Phase are associated with entities in the operational schema.

2. Fact schema building Phase:

a draft conceptual schema is created. Organizes the key facts in a way that makes sense for analysis.

4. Refinement Phase:

- Draft conceptual sch... is refine to meet uress expectations.
- The aim of Phase to re... final schemata to make them ... suitable for users.

Requirement-Driven Approach:

⇒ In mixed and data-driven appro... the starting point is the structure of t... existing operational databases.

⇒ In the requirement-driven approach, the focus is on the analyti... needs, and the data warehouse desig... is more independent of the operati... databases.

Steps in Requirement-Driven Design:

1. identify functional dependencies b/w dimensions:-

- Figuring out how dimensions are related to each other.

Mark optional dimensions:

- Some dimensions might be optional. They don't apply to all data.
- Merge measures with the same aggregation operators: if have multiple measures that are calculated in the same way, then combine them.
- Merge dimensions or fragment parts: This involves deciding whether to combine dimensions or split parts into smaller pieces based on specific criteria.