Lecture Notes

Data Warehousing and ETL

Introduction to Dimensional Modelling

OLTP

The purpose of a database defines the type of data storage or data processing for which a particular database is used. It also defines the users of that database. Based on this information, we design the tables of the database and define the relations between those tables. Here, we will discuss the following two main reasons why every enterprise stores data:

1. **To run the daily business**
2. **To make data-driven decisions**

Now, you have an understanding of relational databases, specifically, what their use is and how they can be modelled. Relational databases are optimised for updation queries and concurrent usage, and, thus, they can be used to handle various real-time transactions to run the daily business.

On-**L**ine **T**ransactional **P**rocessing (OLTP) systems include the interface through which customers avail a service or purchase a product from a company. The role of OLTP systems is to provide a great user experience to every customer. Now, since OLTP systems have to store real-time transactional data and handle concurrent usage, they use relational databases for these purposes.

However, these systems are not optimised for analysis. To perform data analysis effectively, you need to run queries on large amounts of data stored by companies. Also, since relational models have tables in the 2NF or the 3NF form, data is present across multiple tables. And query results are not quite fast when running queries on large amounts of data stored in multiple tables.

Data Warehouses

On-**L**ine **A**nalytical **P**rocessing (OLAP) includes various processes that are required for storing and analysing data. It includes ETL processes and data warehouses. Now, to analyse data, it needs to be stored in a format that supports faster analytics on integrated data. For this purpose, we use data warehouses, which provide a structured format wherein you store the key numerical data that is to be analysed. Some major features of a data warehouse are:

Subject-oriented: A data warehouse stores information about one particular subject.
Integrated: The entire data pertaining to one subject is stored in one location.
Non-volatile: The data stored in a data warehouse does not change.
Time-variant: Historical data collected by companies is stored in a data warehouse.

Introduction to Dimensional Modelling

We use dimensional models to organise data into tables and define the relations between the tables in a data warehouse.

The process of building a dimensional model involves the following steps:

1. **Defining the business process:** Data that is relevant to a business process is stored in a data warehouse. The business process defines the key metrics that a company is looking to use for the purpose of analysis.
2. **Defining the grain:** A grain is defined to gain an understanding of the amount of detailed data that a company wants to store in a data warehouse.
3. **Defining facts and dimensions:** Fact tables store numeric data, whereas dimension tables store descriptions of the numeric data.

Now, consider the first step in building the dimensional model: Defining the business process. A business process is any business activity that a company wants to analyse.

Suppose there is a retail company, upGrad Fashions, and we have to build a dimensional model for the data warehouses of this company. The process that the company wants to analyse is sales achieved at each of its stores at various locations across the country. So, the process for the dimensional model in this case is sales.

Define the Grain

Granularity defines whether the dimensional model will store atomic data or summarised data. If the analytics is more detailed, then the data has higher granularity. This means if the information is highly detailed and atomic, then the data is more granular. In contrast, if the data is more summarised, then it has less granularity. **Now, suppose upGrad Fashions wants to analyse its sales at the month grain only.** In this case, there will be only one record for one month for each store of the company, and this record will have information pertaining to the total sales for that particular month.
Now, suppose upGrad Fashions wants to analyse its sales on a daily basis. In this case, there will be one record for each store of the company daily. The data in this case is more granular than month grain.

Next, suppose upGrad Fashions wants to analyse the product sales achieved in a day on the basis of each transaction. This means that for every transaction that is carried out at each store, there will be a record in the data warehouse.

Now, suppose upGrad Fashions wants to analyse each transaction in detail; in this case, the analysis can include the products sold for each transaction. This means there will be a record for each product sold at every store in a data warehouse.

For the upGrad sales process, we will take the grain as each product sales.

Facts and Dimensions

Fact tables store numeric data. They contain foreign keys to all related dimensions. On the other hand, dimension tables are used to store descriptions of numeric data. Now, let's discuss the upGrad Dimension Model that we have created. There are four dimension tables, which are as follows:

1. **Store Dimension**, which contains information about each store.
2. **Product Dimension**, which contains information about each product.
3. **Date Dimension**, which stores various attributes related to the date on which a product is sold.
4. **Promotion Dimension**, which stores the promotional details pertaining to a sold product.

Dimensional Models and Data Marts

Factless Tables and Different Attributes

The three-level architecture of a DBMS describes the storage and management of data in a database.

Factless fact tables are need for the following two main reasons:

1. **To record events that contain only descriptive data**
2. **To record events that did not take place**

Consider additive, semi-additive and non-additive attributes.

- **Additive attributes:** Additive attributes can be added across all the dimensions. The total sales amount is an additive attribute.

- **Semi-additive attributes:** Semi-additive attributes can be added across some of the dimensions only. The total quantity remaining is a semi-additive attribute.
- **Non-additive attributes:** Non-additive attributes cannot be added across any dimension. Ratios and percentages are non-additive attributes.

Derived attributes can be derived from other attributes using some mathematical operations.

Slowly Changing Dimensionns

The data stored in a data warehouse is generally non-volatile. This means once stored, the data does not change and, in fact, it should not, since it is the historical data stored by the company. Nevertheless, there may be many changes within the business that may introduce inconsistency in the data stored in a data warehouse.

Let’s say the address of one of the stores has changed. This means the region in which the products are selling is different. Now, if we continue using the same address, then we would incorrectly analyse the sales in that region. So, there must be ways to handle such changes to data.

Following are the three ways of handling slowly changing dimensions:

1. **Type 1:** You change the value stored in the row. This method is used when the data warehouse does not want to keep track of the changes to the data.
2. **Type 2:** You create a new row with all the attributes having the same value, except the attribute whose value needs to be updated. This method is used when the data warehouse needs to keep track of all the changes to the data.
3. **Type 3:** This method is used when the data warehouse needs to keep track of the recent changes to the data. You add another column, which stores the previous value of an attribute.

Snowflake Schema

A **snowflake schema** has dimension tables in the 2NF or the 3NF form. Each dimension can store some of its attributes in new tables, which are related to the dimension table through foreign keys. A dimension table categorises their attributes into different tables using functional dependencies and transitive dependencies in the dimension table. The fact table is related to only one of these tables, whereas the other tables are related to this one table.

Why are star schemas used more?

1. The number of tables is more in case of a snowflake schema and the query results are not fast.
2. A dimension table is smaller than a fact table. This means even if data gets repeated in the star schema, because of functional and transitive dependencies, it does not pose a significant challenge as fact tables expand in size as new data is added to the data warehouses.
3. Star schemas paint a complete picture of one dimension in one table.

Snowflake schemas are also used in the following cases:

1. If a dimension table is not used too frequently, then it can be divided into smaller tables based on the functional and transitive dependencies.
2. If a dimension table is too wide, then we can split it into smaller tables.

Data Marts

Data marts contain information about a specific topic. Now, suppose a data warehouse stores information pertaining to product performance. In this case, data marts could store information pertaining to each category of products and how all the products in that particular category have been performing.

Data marts are easy to set up, they contain specific information, and you can extract data from files that contain that information.

Based on the data source, data marts are of the following three types:

1. **Independent data marts:** These data marts use original files, which contain information as data sources.
2. **Dependent data marts:** These data marts use data warehouses as data sources.
3. **Hybrid data marts:** These data marts use both data warehouses and original files as data sources.

Data warehouses can be designed in the following two ways:

1. **Bill Inmon's architecture:** In this architecture, you first build a data warehouse for an entire organisation and then build data marts for specific topics.
2. **Kimball's architecture:** In this architecture, you first build data marts for every subject and then build a data warehouse using the data marts.

ETL and ELT

The Process of ETL

The ETL process is a collection of three steps: Data extraction, Data transformation and Data loading.

Now, there are certain concepts related to the ETL process that you must know about. These concepts are as follows:

1. **Business requirements:** It is important to have a discussion with the users of this entire system and a discussion with the data modelling team working on the designing dimensions and facts.
2. **Data profiling:** Data profiling is the process of examining different data sources for quality and context of use.
3. **Exception handling:** If any of the steps of the ETL process throws an exception, then there must be a

record to tell which step has thrown which exception.

4. **Data security:** You do not want the business users to access the data at the different stages of data warehouse development.
5. **Data archiving:** Data derived from different sources may require some changes during the process. Hence, it is helpful to maintain a separate staging area for recording the changes made to the data, so that you do not have to start from the data source again if you need to reprocess some data.

Data Extraction

Data extraction includes understanding the data sources, using business requirements and data profiling to choose the right data sources, connecting to those data sources, and then bringing the data into the main processing stream.

The two main data extraction processes are as follows:

1. **Full extraction:** In this process, the entire data is extracted from the data source whenever the data source undergoes any updates.
2. **Incremental extraction:** In this process, only the newly updated data is extracted whenever the data source undergoes any updates. Various triggers are used to indicate whether or not the data source has undergone any updates.

Data Transformation and Loading

Data transformation includes data structuring, data cleaning and data enrichment.

Data cleaning includes:

1. Removing multiple copies of the same data,
2. Checking whether permitted values are used in each field, and
3. Checking for null values.

Data structuring includes:

1. Understanding the format of each file that is extracted,
2. Extracting the required data from the file, and
3. Structuring the data according to the schema used in the data warehouse.

Data enrichment includes:

1. Checking for data quality according to the business rules,

2. Checking whether the data present in all the fields complies with the business rules, and
3. Deriving key attributes based on business requirements.

Data loading includes:

1. Loading data into the dimension tables first,
2. Loading the foreign keys into the fact tables, and
3. Loading data into the fact tables.

The Process of ELT

There are several issues with the ETL process, which include the following:

1. **Specifying the business subject first**
   Although it is helpful to specify the subject first, data extraction and transformation according to every business subject are not fast when data is available in huge volumes in multiple different formats.
2. **Need for servers to perform data transformation**
   To perform transformation processes before loading the data, you need to build a staging area where you can perform data transformations.
3. **Less flexibility**
   Data sources have to be cleaned and then data transformation has to be performed first, according to the defined business process. Data from every source has to be transformed first, even though it may not be used frequently.

So, to handle these issues, companies use another process: the ELT process. In this process, data is extracted from different data sources and is first loaded into a central repository. Whenever data analysis is to be performed, the required data is transformed and cleaned.

Data Lakes

Just like a data warehouse, a data lake is a centralised repository that stores the data collected from different sources in the organisational ecosystem; the only difference is that a data lake stores both structured and unstructured data. Data can be dumped into this repository in raw format without any preprocessing.

Data lakes help address the various issues of data warehouses, which include the following:

1. **Specific schema**
   You need to define a data warehouse schema with facts and dimensions before loading the data into the warehouse.
2. **Highly structured**

   Data warehouses can store only structured data. Unstructured data sources have to be structured first before loading the data into the data warehouse.

3. **Less flexibility**
   Over the years, there has been an increase in data formats. Weblogs, XML formats, and many other forms of semi-structured and unstructured data are prominent sources of data for analysis. Each source of unstructured data has to be transformed into a structured format for loading into data warehouses.

Both data warehouses and data lakes are used by organisations for different purposes. A data warehouse is mainly used for reporting various financial data. On the other hand, data lakes are used as data repositories where organisations can load all of their data, be it structured, semi-structured or unstructured. For reporting, the data from data lakes can be transformed and loaded into data warehouses in a structured format.

Disclaimer: *All content and material on the upGrad website is copyrighted material, belonging to either upGrad or its bona fide contributors, and is purely for the dissemination of education. You are permitted to access, print, and download extracts from this site purely for your own education only and on the following basis:*

- *You can download this document from the website for self-use only.*
- *Any copies of this document, in part or full, saved to disc or to any other storage medium, may be used for subsequent, self-viewing purposes or to print an individual extract or copy for non-commercial personal use only.*
- *Any further dissemination, distribution, reproduction, and copying of the content of the document herein, or the uploading thereof on other websites, or use of the content for any other commercial/unauthorized purposes in any way that could infringe the intellectual property rights of upGrad or its contributors is strictly prohibited.*
- *No graphics, images, or photographs from any accompanying text in this document will be used separately for unauthorized purposes.*
- *No material in this document will be modified, adapted, or altered in any way.*
- *No part of this document or upGrad content may be reproduced or stored on any other website or included in any public or private electronic retrieval system or service without upGrad’s prior written permission.*
- *Any rights not expressly granted in these terms are reserved.*