# **Data Warehousing & Data Modeling Doc**

By Deepanshu Kalra ([LinkedIn](https://www.linkedin.com/in/deepanshuk/), [Instagram](http://instagram.com/dkalra_/), [Direct 1:1](http://topmate.io/deepanshu))

Welcome Nerds!

We've made it easy for you, the data nerds, to learn and revise data warehousing basics with this document. SQL and Analytics are the only prerequisites. Firstly, let's talk about WHY?

**The data warehouse is similar to building an eight-lane highway for your data, as opposed to having it travel along narrow, lawless streets. -Deepanshu Kalra**

Data Warehousing helps users gain insights by developing simpler and more efficient queries and increases the speed and efficiency of accessing different data sets. Data Warehouses are essential to maintaining consistency across your organization, otherwise everyone will have their own data standards, which is a serious concern. As it generally stores historical data as well, the data is typically larger and stored in a format that is easier to read (rather than easy to write/update). For other helpful resources on SQL, Python and DSA please head towards: [**nerdsfornerds.in**](http://nerdsfornerds.in/)

Keep Hustling. Keep Growing.

Deepanshu Kalra

**Note: This is an initial draft. I will keep improving it over time.**

# 

# Tabel of Content

[**Data Warehousing & Data Modeling Doc**](#_9hwz5ehp6c8) **1**

[**Tabel of Content**](#_kud35uid8oz4) **2**

[**Data Modeling & Why It is done?**](#_5m8fp2eb217s) **5**

[Why is Data Modeling done?⭐](#_o2j1vbx3x5rt) 5

[Phases of Data Model](#_zic6bjv4m8tv) **5**

[Conceptual Data Model](#_tjs45nugau2t) 5

[Logical Data Model](#_jpbolv6aq07r) 5

[Physical Data Model](#_kotjomta5yv6) 5

[Data Modeling Techniques](#_i05ktp9eq6fc) **6**

[Hierarchical data Modeling](#_cbh2fyv217dp) 6

[Network data modeling](#_c5ljbzchcfd3) 6

[Relational data modeling](#_iqt7pqm4ykje) 6

[Entity Relationship data modeling⭐⭐](#_3r0ig7eg4qi9) 7

[Dimension Data Modeling ⭐⭐⭐⭐](#_vk4wpu5wlg3w) 7

[Object Oriented Data Modeling](#_sif4yckelch4) 7

[Graph Data Modeling](#_5c1mzstf9e84) 7

[Steps in designing a data model ⭐⭐](#_rhxyusach2f0) **7**

[Building blocks of data model ⭐⭐](#_nv8900ahy0bd) **8**

[Entities](#_5avd2obr1koe) 8

[Attributes](#_wpwmrtktlb4r) 8

[Relationship](#_kxmdogdehdlq) 8

[Constraints](#_n2tyf4zf8sxv) 8

[Normalization & Its types ⭐⭐⭐⭐](#_evdu5qivrku1) **8**

[First Normal Form (1NF)](#_4npwe2hrh6l9) 9

[Second Normal Form (2NF)](#_hc9yj1o8yqcx) 9

[Third Normal Form (3NF)](#_9pqknjl1u07b) 9

[Benefits & Drawbacks of Normalization](#_2qfj8dlycn6z) **9**

[Drawbacks of Normalization⭐⭐](#_p3bvk99k5gxo) 10

[Relationship and its Types in DBMS](#_juycwfcsfax1) **10**

[Types of Relationship in DBMS](#_u5m9g0bcnppq) 10

[One to One](#_pcu27pjznkp4) 10

[One to Many](#_mcd604y7kv3x) 10

[Many to Many ⭐⭐](#_etxwr27odkjo) 11

[Many to One](#_73b8n0218aqv) 11

[Cardinality](#_xhcf9umcyeg9) **11**

[In Context of Data Models](#_kyape1d0mvpa) 11

[In Context of Query Optimization⭐](#_eoq5pvwclfvn) 11

[Data Warehousing](#_4umhaqvnkxil) **11**

[Goals of Data Warehousing & Business Intelligence⭐⭐](#_km0nzo1cvo91) 12

[OLTP vs OLAP based data warehouses⭐⭐](#_bmz22llfup3p) 12

[Dimensional Modeling ⭐⭐⭐⭐](#_j29328nmclgl) **13**

[Steps to Create Dimensional Data Modeling](#_g4vrl8irk0oi) **13**

[Fact Tables](#_1xnq1ltbfkn8) **14**

[Types of Fact Tables](#_mvwe0meprbkm) 15

[Transactional](#_b0xa1rqo5bt9) 15

[Periodic snapshots](#_j0w85ckvr9gk) 15

[Accumulating snapshots](#_kd98pp3a9i48) 15

[Factless Fact⭐⭐](#_l4grlttj60mq) 15

[Dimension Tables](#_py3k74joas1i) **16**

[Types of Dimension Tables](#_3dcty9e9scvf) 16

[Slowly Changing Dimension (SCD)⭐⭐](#_q0g1hlyd8g70) 16

[Conformed Dimension:](#_2tn36nnmt3lz) 18

[Junk Dimension ⭐⭐](#_9cnr9vlku0cy) 18

[Degenerate Dimension](#_oxybzmomckho) 18

[Roleplay Dimension:](#_6bb5q1d3wcc8) 18

[Types of Schemas in Dimensional Modeling](#_7wtrywgv22z6) **19**

[Star Schema ⭐⭐⭐⭐](#_ojckuvl1jxxq) 19

[Snowflake Schema⭐⭐⭐⭐](#_goajlgabm89z) 19

[Galaxy Schema](#_5kvskva48zvc) 20

[Fact Constellation Schema:](#_1t6rg81foa66) 20

[**Multi-Dimensional Modeling (can ignore)**](#_amies99kqv6j) **20**

[**ETL (Extract Transform and Load) ⭐⭐⭐⭐**](#_xgg8t2o5tbku) **22**

[ELT (Extract Load and Transform)](#_v459g1hlyksu) **22**

[Difference Between ETL and ELT⭐](#_kts9g27mq1cc) **22**

[Data Mart](#_vbm3cpu1p5l8) **22**

[Benefits of Data Mart:](#_f20079f39k5e) 23

[The difference between data mart, data lake, data warehouse](#_ef2v6z4qg535) **23**

[Cloud Data Warehousing](#_2xbssdvmghu2) **24**

[Advantages:](#_ouupqiblhj1y) 24

[Different Cloud Data Warehousing Platforms](#_onzqbanw4ozx) 26

[**Other References & Authors**](#_i2jd0m2potxm) **27**

# 

# Data Modeling & **Why** It is done?

For building a Data Warehouse you need to understand *Data Warehouse Modeling* or Data Modeling first. Data modeling is the process of creating a visual representation of either a whole **information system** or parts of it to communicate connections between data points and structures. The goal is to illustrate the **types of data used and stored** within the system, the relationships among these data types, the ways the data can be **grouped and organized and its formats and attributes.**

## Why is Data Modeling done?[⭐](#_q0g1hlyd8g70)

Data modeling is a core data management discipline. By providing a visual representation of data sets and their business context, it helps **pinpoint information needs for different business processes**. It then specifies the characteristics of the data elements that will be included in applications and in the database or file system structures used to process, store and manage the data.

# Phases of Data Model

## Conceptual Data Model

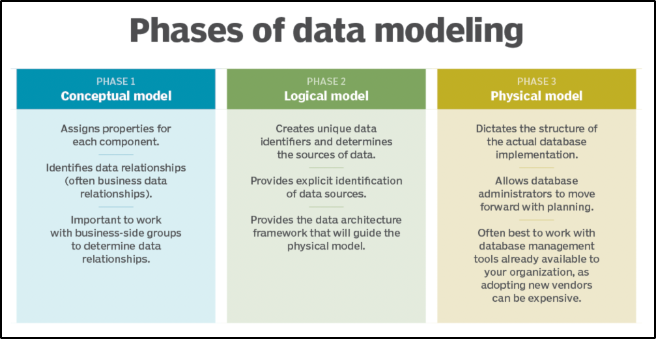
This Data Model defines **WHAT** the system contains. This model is typically created by Business stakeholders and Data Architects. The purpose is to organize, scope and define business concepts and rules.

## Logical Data Model

Defines **HOW** the system should be implemented regardless of the DBMS. This model is typically created by Data Architects and Business Analysts. The purpose is to develop a technical map of rules and data structures.

## Physical Data Model

This Data Model describes HOW the system will be **implemented** using a specific DBMS system. This model is typically created by DBA and developers. The purpose is actual implementation of the database.



# Data Modeling Techniques

## Hierarchical data Modeling

Organize data in a treelike arrangement of **parent and child** records. A child record can have only one parent, making this a one-to-many modeling method.

## Network data modeling

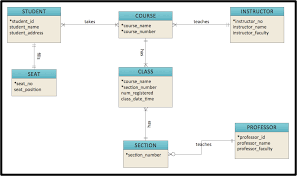
Popular data modeling option in mainframe databases that isn't used as much now. Network data models expanded on hierarchical ones by allowing child records to be connected to **multiple parent** records.

## Relational data modeling

Relational model **map**s the relationships between data elements stored in different tables that contain sets of rows and columns. Relational modeling set the stage for the development of relational databases

## Entity Relationship data modeling[⭐⭐](#_q0g1hlyd8g70)

A variation of the relational model that can also be used with other types of databases, entity-relationship (ER) models **visually map entities, their attributes and the relationships between different entities**.



## Dimension Data Modeling [⭐⭐⭐⭐](#_q0g1hlyd8g70)

Primarily used in data warehouses and data marts that support business intelligence applications. They consist of fact tables that contain data about transactions or other events and dimension tables that list attributes of the entities in the fact tables.

## Object Oriented Data Modeling

The object-oriented approach is similar to the ER method in how it represents data, attributes and relationships, but it abstracts entities into objects. Different objects that have the same attributes and behaviors can be grouped into classes, and new classes can inherit the attributes and behaviors of existing ones.

## Graph Data Modeling

The graph data model is a more modern offshoot of network and hierarchical models. Typically paired with [graph databases](https://www.techtarget.com/whatis/definition/graph-database), it's often used to describe data sets that contain complex relationships. For example, graph data modeling is a popular approach in social networks, recommendation engines and fraud detection applications.

# Steps in designing a data model ⭐⭐

* Identify the business entities that are represented in the data set.
* Identify key properties for each entity to differentiate between them.
* Create a draft entity-relationship model to show how entities are connected.
* Identify the data attributes that need to be incorporated into the model.
* Map the attributes to entities to illustrate the data's business meaning.
* Finalize the data model and validate its accuracy.

# Building blocks of data model ⭐⭐

## Entities

Entities are real time objects that exist. It can be a person, place, object, event, concept. Entities are represented by a rectangle box containing the entity name in it.

Example: Student, employee.

## Attributes

It is the set of characteristics representing an entity. It is represented by an ellipse symbol with attribute name on it.

Example: A student has attributes like name, roll number, age and much more.

## Relationship

It describes the association between two entities. It is represented using a diamond symbol containing a relationship name with it. The data model generally uses three kinds of relationships: one to many, many to many, one to one.

Example: The relationship between two entities Student and Class has many to many relationships.

## Constraints

Constraints are conditions applied on the data. It provides data integrity.

Example: A student can take a maximum of 2 books from the library is applied as a constraint on the student database.

# Normalization & Its types [⭐⭐⭐⭐](#_q0g1hlyd8g70)

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divide larger tables into smaller tables and link them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

<https://www.youtube.com/watch?v=ABwD8IYByfk>

<https://www.guru99.com/database-normalization.html>

<https://towardsdatascience.com/database-normalization-explained-53e60a494495>

* 1NF (First Normal Form)
* 2NF (Second Normal Form)
* 3NF (Third Normal Form)
* BCNF (Boyce-Codd Normal Form)
* 4NF (Fourth Normal Form)
* 5NF (Fifth Normal Form)
* 6NF (Sixth Normal Form)

## First Normal Form (1NF)

* Data is stored in tables with rows uniquely identified by a primary key
* Data within each table is stored in individual columns in its most reduced form
* There are no repeating groups

## Second Normal Form (2NF)

* Everything from 1NF
* Only data that relates to a table’s primary key is stored in each table

## Third Normal Form (3NF)

* Everything from 2NF
* There are no in-table dependencies between the columns in each table

# Benefits & Drawbacks of Normalization

Normalization provides numerous benefits to a database. Some of the major benefits include the following :

* Greater overall database organization
* Reduction of redundant data
* Data consistency within the database
* A much more flexible database design
* A better handle on database security

## Drawbacks of Normalization[⭐⭐](#_q0g1hlyd8g70)

* Reduced database performance
* Requires much more CPU, memory and I/O to process transactions and db queries than does a denormalized database
* Must locate requested tables and join the data from the tables to either get the requested information or to process the desired data

# Relationship and its Types in DBMS

A relationship in a DBMS, is primarily the way two or more data sets are linked. This is so true for Relational Database Management Systems. One dataset may be then termed as the Foreign key and the ones linked to it may be termed as the Primary Key. There may be multiple Foreign and Primary keys linked to each other.

Relationships allow the datasets to share and store the data in separate tables. They also help link disparate data with each other.

## Types of Relationship in DBMS

### One to One

A single record in Table A is related to a single record in Table B. And vice versa

### One to Many

One data in Table A has links to multiple data in Table B. However, a single data in Table B, will have a link to a single data in Table A. One good example to showcase such a relationship is through a bank account. One individual may have multiple bank accounts, but each bank account will be linked to one specific owner or account holder only (provided we are not talking about a joint bank account)

### **Many to Many** [**⭐⭐**](#_q0g1hlyd8g70)

Consider two tables A and B. Each data in Table A is linked to all the data in Table B and vice versa. A great example would be linking a student table to the courses table. Table A has the name of all the students while Table B has the name of all the courses. A particular student may enroll oneself to more than one course whilst a course may have more than one student. Hence, it becomes a many to many relationship.

### Many to One

Many rows of the first table are associated with a single row of the second table. For example, the relationship between student and university is many to one because a university can have many students but a student can only study only in a single university at a time.

# Cardinality

In DBMS you may hear the cardinality term at two different places and it has two different meanings as well.

## In Context of Data Models

In terms of data models, cardinality refers to the relationship between two tables.

## In Context of Query Optimization[⭐](#_q0g1hlyd8g70)

In terms of query, the cardinality refers to the uniqueness of a column in a table. The column with all unique values would be having the high cardinality and the column with all duplicate values would be having the low cardinality. These cardinality scores help in query optimization.

# **Data Warehousing**

A Data Warehousing (DW) is a process for collecting and managing data from varied sources to provide meaningful business insights. A Data warehouse is typically used to connect and analyze business data from heterogeneous sources. The data warehouse is the core of the BI system which is built for data analysis and reporting.

## Goals of Data Warehousing & Business Intelligence[⭐⭐](#_q0g1hlyd8g70)

* It must make information easily accessible.
* It must present information consistently.
* It must adapt to change.
* It must present information in a timely way.
* It must be a secure bastion that protects the information assets.
* It must serve as the authoritative and trustworthy foundation for improved decision making.
* The business community must accept the DW/BI system to deem it successful.

## OLTP vs OLAP based data warehouses[⭐⭐](#_q0g1hlyd8g70)

|  |  |  |
| --- | --- | --- |
|  | **OLTP** | **OLAP** |
| Stands for | Online transaction processing | Online analytical processing |
| Usage pattern | Optimized for fast CRUD(create, read, update, delete) of a small number of rows | Optimized for running select c1, c2, sum(c3),.. where .. group by on a large number of rows (aka analytical queries), and ingesting large amounts of data via bulk import or event stream |
| Storage type | Row oriented | Column-oriented |
| Data modeling | Data modeling is based on normalization | Data modeling is based on denormalization. Some popular ones are dimensional modeling and data vaults |
| Data state | Represents current state of the data | Contains historical events that have already happened |
| Data size | Gigabytes to Terabytes | Terabytes and above |
| Example database | MySQL, Postgres, etc | Clickhouse, AWS Redshift, Snowflake, GCP Bigquery, etc |

# Dimensional Modeling ⭐⭐⭐⭐

Dimensional Data Modeling is one of the data modeling techniques used in data warehouse design.

Dimensional modeling is widely accepted as the preferred technique for presenting analytic data because it addresses two simultaneous requirements:

* Deliver data that’s understandable to the business users.
* Deliver fast query performance.

The concept of Dimensional Modeling was developed by Ralph Kimball which consists of facts and dimension tables. Since the main goal of this modeling is to improve the data retrieval so it is optimized for SELECT OPERATION. The advantage of using this model is that we can store data in such a way that it is easier to store and retrieve the data once stored in a data warehouse. Dimensional model is the data model used by many OLAP systems.

# Steps to Create Dimensional Data Modeling

Step-1: Identifying the business objective –

The first step is to identify the business objective. Sales, HR, Marketing, etc. are some examples as per the need of the organization. Since it is the most important step of Data Modeling, the selection of business objectives also depends on the quality of data available for that process.

Step-2: Identifying Granularity –

Granularity is the lowest level of information stored in the table. The level of detail for a business problem and its solution is described by Grain.

Step-3: Identifying Dimensions and its Attributes –

Dimensions are objects or things. Dimensions categorize and describe data warehouse facts and measures in a way that supports meaningful answers to business questions. A data warehouse organizes descriptive attributes as columns in dimension tables. For Example, the data dimension may contain data like a year, month and weekday.

Step-4: Identifying the Fact –

The measurable data is held by the fact table. Most of the fact table rows are numerical values like price or cost per unit, etc.

Step-5: Building of Schema –

We implement the Dimension Model in this step. A schema is a database structure. There are two popular schemes: [Star Schema](https://www.geeksforgeeks.org/star-schema-in-data-warehouse-modelling/) and [Snowflake Schema](https://www.geeksforgeeks.org/data-warehouse-modelling-snowflake-schema/).

# Fact Tables

The term fact represents a business measure. Imagine standing in the marketplace

watching products being sold and writing down the unit quantity and dollar sales

amount for each product in each sales transaction. These measurements are captured as products are scanned at the register.

Each row in a fact table corresponds to a measurement event. The data on each

row is at a specific level of detail, referred to as the grain, such as one row per product sold on a sales transaction. One of the core tenets of dimensional modeling is that

all the measurement rows in a fact table must be at the same grain. Having the discipline to create fact tables with a single level of detail ensures that measurements aren’t inappropriately double-counted.

A fact table can store different types of measures such as additive, non-additive, semi-additive.

* Additive – As its name implied, additive measures are measures that can be added to all dimensions.
* Non-additive – different from additive measures, non-additive measures are measures that cannot be added to all dimensions.
* Semi-additive – semi-additive measures are measures that can be added to only some dimensions and not across others.

The most useful facts are numeric and additive, such as dollar sales amount.



## Types of Fact Tables

### Transactional

The most basic one is that each grain associated with it is indicated as “one row per line in a transaction”, e.g., every line item appears on an invoice. Transaction fact table stores data of the most detailed level, therefore, it has a high number of dimensions associated with it.

### Periodic snapshots

It stores the data that is a snapshot in a period of time. The source data of the periodic snapshots fact table is data from a transaction fact table where you choose a period to get the output.

### Accumulating snapshots

It describes the activity of a business process that has a clear beginning and end. This type of fact table, therefore, has multiple date columns to represent milestones in the process. A good example of accumulating snapshots in a fact table is the processing of a material. As steps towards handling the material are finished, the corresponding record in the accumulating snapshots fact table gets updated.

### Factless Fact[⭐⭐](#_q0g1hlyd8g70)

A factless fact table is a fact table that does not have any measures, i.e. any numeric fields that can be aggregated.

<https://stackoverflow.com/questions/54421583/what-is-the-difference-between-fact-less-fact-and-fact-table>

# Dimension Tables

Dimension tables are integral companions to a fact table. The dimension tables contain the textual context associated with a business process measurement event. They describe the “who, what, where, when, how, and why” associated with the event. Each dimension is defined by a single primary key.

## Types of Dimension Tables

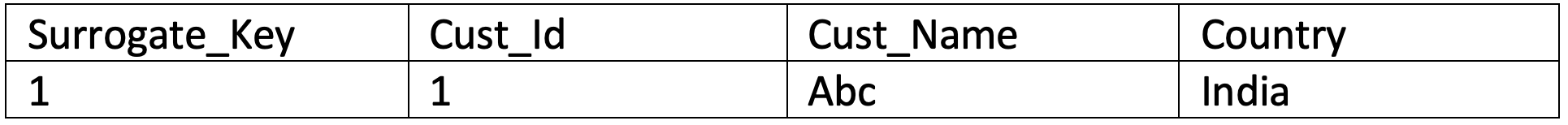
### Slowly Changing Dimension (SCD)[⭐⭐](#_q0g1hlyd8g70)

The dimension attributes that tend to change slowly with time rather than changing in a regular interval of time are called slowly changing dimensions. For e.g. address and phone number changes but not regularly. There are different ways of creating SCDs.

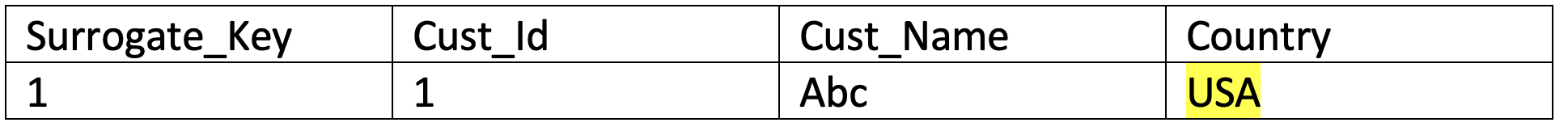
* Type0 (Retain Original)
* Type1 (Overwrite):

SCD type 1 methodology is used when there is no need to store historical data in the dimension table. This method overwrites the old data in the dimension table with the new data. It is used to correct data errors in the dimension.

Before:



After:



* Type2: Add New Row[⭐⭐⭐⭐⭐](#_q0g1hlyd8g70)

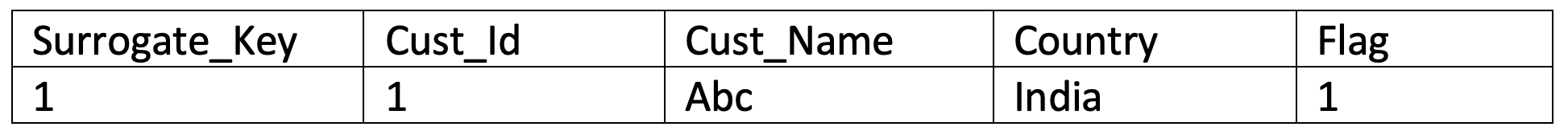
SCD type 2 stores the entire history, the data in the dimension table. With type 2 we can store unlimited history in the dimension table. In type 2, you can store the data in three different ways. They are:

o Versioning

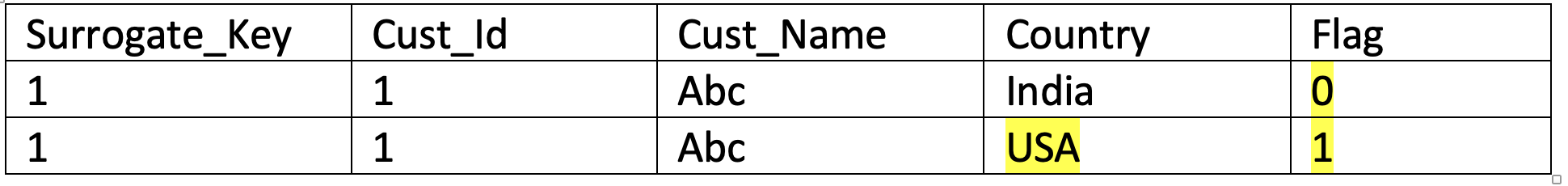
o Flagging (example shown)

o Effective Date

Before:



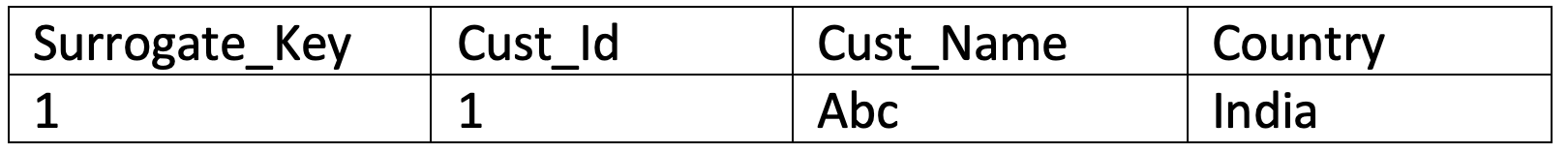
After:



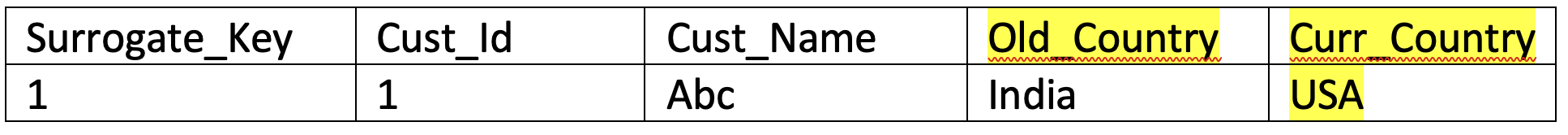
* Type3 (Add New Attribute)

Add a new column. It is the best approach as history can be maintained easily. Only one level of history is stored.

Before:



After:



* Type4 (Add Mini Dimension)
* Type5 (Add Mini Dimension and Type 1 Outrigger)
* Type6 (Add Type 1 Attributes to Type 2 Dimension)
* Type7 (Dual Type1 and Type 2 Dimensions)

<https://en.wikipedia.org/wiki/Slowly_changing_dimension>

<https://www.educba.com/dimension-table/>

<https://www.folkstalk.com/2012/03/slowly-changing-dimensions-scd-types.html>

https://datawarehouse4u.info/SCD-Slowly-Changing-Dimensions.html

### Conformed Dimension:

This dimension is shared among multiple subject areas or data marts. Same can be used in different projects without any modifications done in the same. This is used to maintain consistency. Conformed dimensions are those which are exactly the same or a proper subset of any other dimension.

### Junk Dimension [⭐⭐](#_q0g1hlyd8g70)

A junk dimension is a group of attributes of low cardinality. It contains different or various attributes which are unrelated to any other attribute. These can be used to implement RCD (rapidly changing dimension) such as flags, weights etc.

### Degenerate Dimension

Its attributes are stored in the fact table itself and not as a separate dimension table, those attributes are called degenerate dimension. For e.g. ticket number, invoice number, transaction number etc.

### Roleplay Dimension:

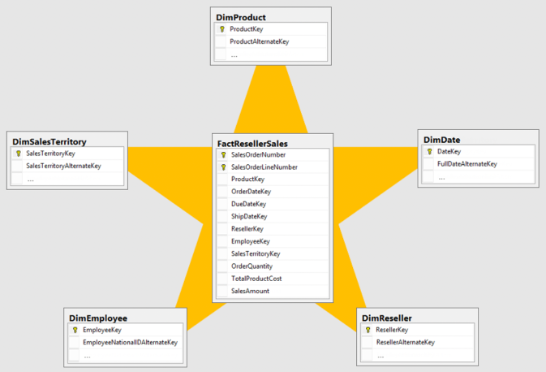
It has multiple relationships with the fact table called role-play dimension. In other words, it is when the same dimension key with all its related attributes is joined to many foreign key presents in the fact table. It can fulfill multiple purposes within the same existing database.

# 

# Types of Schemas in Dimensional Modeling

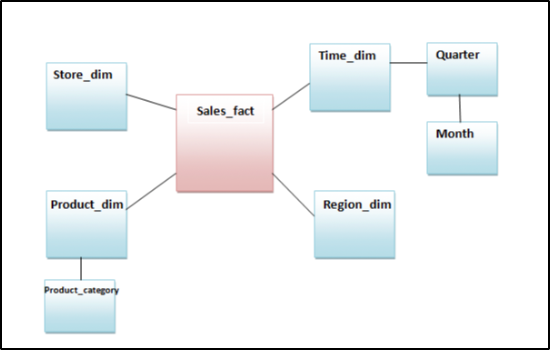
## Star Schema [⭐⭐⭐⭐](#_q0g1hlyd8g70)

A star schema is the one in which a central fact table is surrounded by denormalized dimensional tables. A star schema can be simple or complex. A simple star schema consists of one fact table whereas a complex star schema has more than one fact table.



## Snowflake Schema[⭐⭐⭐⭐](#_q0g1hlyd8g70)

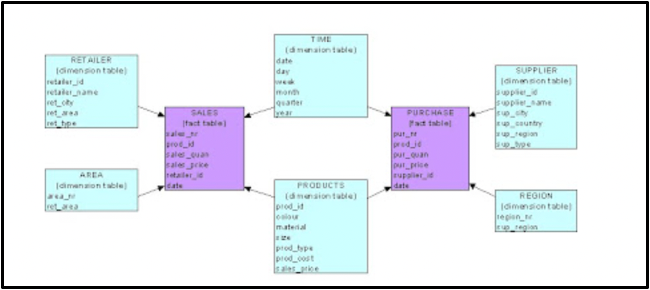
A snowflake schema is an enhancement of a star schema by adding additional dimensions. Snowflake schema are useful when there are low cardinality attributes in the dimensions.



<https://www.guru99.com/star-snowflake-data-warehousing.html>

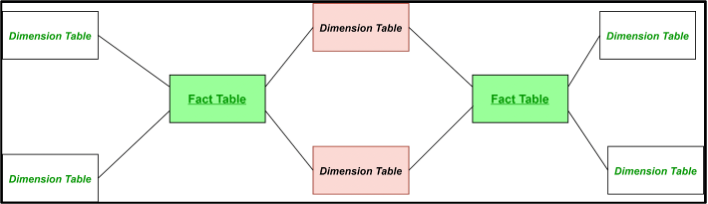
## Galaxy Schema

Galaxy schema contains many fact tables with some common dimensions (conformed dimensions). This schema is a combination of many data marts.



## Fact Constellation Schema:

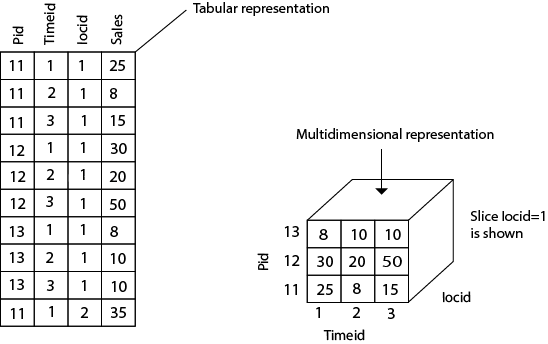
The dimensions in this schema are segregated into independent dimensions based on the levels of hierarchy. For example, if geography has five levels of hierarchy like tertiary, region, country, state and city; constellation schema would have five dimensions instead of one.



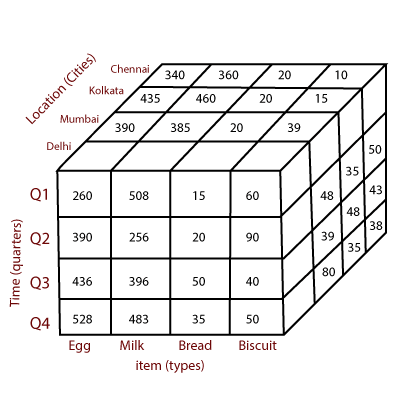
# Multi-Dimensional Modeling (can ignore)

A multidimensional model views data in the form of a data-cube. A data cube enables data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

A multidimensional data model is organized around a central theme, for example, sales. This theme is represented by a fact table. Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.



Imagine the data (generally pre-calculated) to be in the 3 dimensional space rather than a plane row-column based 2 dimensional space.



# ETL (Extract Transform and Load) [⭐⭐⭐⭐](#_q0g1hlyd8g70)

ETL, which stands for extract, transform, and load, is the process data engineers use to extract data from different sources, transform the data into a usable and trusted resource, and load that data into the systems end-users can access and use downstream to solve business problems

* Extract: The process of getting the data from the source system. E.g., a python process to get data from an API, access data from an OLTP database, etc.
* Transform: The process of transforming the extracted data. E.g., changing field types & names, applying business logic to the data set, enriching data, etc.
* Load: The process of loading the transformed data into the data asset used by the end-user.

# ELT (Extract Load and Transform)

ELT is a different method of looking at the tool approach to data movement. Instead of transforming the data before it’s written, ELT lets the target system do the transformation. The data first copied to the target and then transformed in place.

ELT is usually used with no-Sql databases like Hadoop cluster, data appliance or cloud installation.

# Difference Between ETL and ELT[⭐](#_q0g1hlyd8g70)

<https://www.guru99.com/etl-vs-elt.html>

<https://www.startdataengineering.com/post/elt-vs-etl/>

# Data Mart

A [data mart](https://www.oracle.com/in/autonomous-database/departmental-data-warehouse/) is a simple form of data warehouse focused on a single subject or line of business. With a data mart, teams can access data and gain insights faster, because they don’t have to spend time searching within a more complex data warehouse or manually aggregating data from different sources.

## Benefits of Data Mart:

A data mart dedicated to a team or specific line of business offers several benefits:

* A single source of truth
* Quicker access to data
* Faster insights leading to faster decision making
* Simpler and faster implementation
* Creating agile and scalable data management
* Transient analysis

# The difference between data **mart**, data **lake**, data **warehouse**

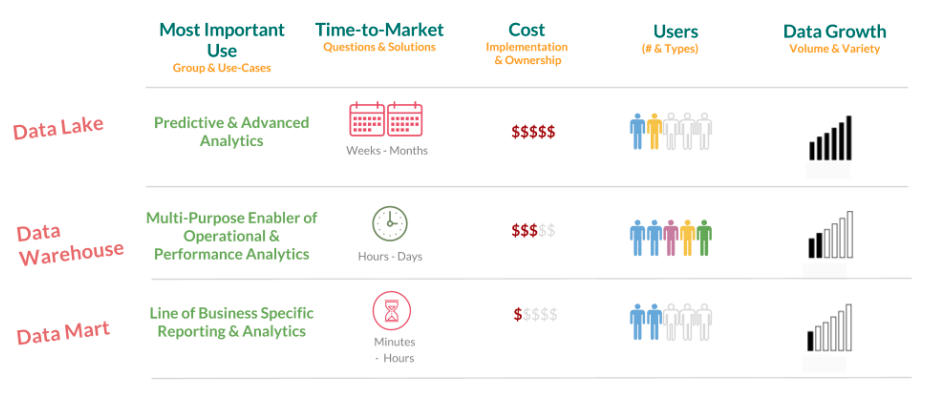
Data marts, data lakes, and data warehouses serve different purposes and needs.

A [data warehouse](https://www.oracle.com/in/database/what-is-a-data-warehouse/) is a data management system designed to support business intelligence and analytics for an entire organization. Data warehouses often contain large amounts of data, including historical data. The data within a data warehouse usually is derived from a wide range of sources, such as application log files and transactional applications. A data warehouse stores structured data, whose purpose is usually well-defined.

A [data lake](https://www.oracle.com/in/big-data/what-is-data-lake/) allows organizations to store large amounts of structured and unstructured data (for example, from social media or clickstream data), and to immediately make it available for real-time analytics, data science, and machine learning use cases. With a data lake, data is ingested in its original form, without alteration.

The key difference between a data lake and a data warehouse is that data lakes store vast amounts of raw data, without a predefined structure. Organizations do not need to know in advance how the data will be used.

A [data mart](https://www.oracle.com/in/autonomous-database/departmental-data-warehouse/) is a simple form of a data warehouse that is focused on a single subject or line of business, such as sales, finance, or marketing. Given their focus, data marts draw data from fewer sources than data warehouses. Data mart sources can include internal operational systems, a central data warehouse, and external data.



<https://www.oracle.com/in/autonomous-database/what-is-data-mart/#:~:text=A%20data%20mart%20is%20a%20simple%20form%20of%20a%20data,fewer%20sources%20than%20data%20warehouses>.

<https://www.guru99.com/data-mart-tutorial.html>

<https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/>

# 

# Cloud Data Warehousing

A cloud data warehouse solution is managed and hosted by a cloud services provider. This gives you the inherent **flexibility** of a cloud environment along with more predictable costs, which can be based on usage or a fixed amount. The up-front **investment is typically much lower and lead times are shorter** than on-premises solutions because you don’t have to buy hardware, thereby reducing CapEx. You can also achieve **operational efficiencies** from the serverless / NoOps nature of cloud data warehouses.

## Key Features

**Massive Parallel Processing (MPP)**: MPP architectures are used in cloud-based data warehouses that support big data projects to provide high-performance queries on large data volumes. MPP architectures are made up of multiple servers that run in parallel to distribute processing and input/output (I/O) loads.

**Columnar data stores**: MPP data warehouses are typically columnar stores, which are the most adaptable and cost-effective for analytics. Columnar databases store and process data in columns rather than rows, allowing aggregate queries, which are commonly used for reporting, to run much faster.

## Advantages

**It’s managed**

A cloud data warehouse lets you outsource the management hassle to cloud providers who must meet service level agreements. This provides operational savings and can keep your in-house team focused on growth initiatives.

**It can provide better uptime compared to on-premises data warehouses**

Cloud providers are obligated to meet SLAs and provide better uptime with reliable cloud infrastructure that scales seamlessly. On-premises data warehouses have scale and resource limitations that could impact performance.

**It’s built for scale**

Cloud data warehouses are elastic, so they can seamlessly scale up or down as your business needs change.

**It’s cost effective**

With cloud, you gain flexible pricing by paying for what you use or choosing a more predictable flat-rate option. Some providers charge by throughput or per hour per node. Others charge a fixed price for a certain amount of resources. In every case, you avoid the mammoth costs incurred by an on-premises data warehouse that runs 24 hours a day, seven days a week, regardless of whether resources are in use or not.

**It supports real-time insights**

Cloud data warehouses support streaming data, allowing you to query data in real time in order to drive fast and informed business decisions.

**It supports machine learning and AI initiatives**

Customers can quickly unlock and operationalize machine learning use cases in order to predict business outcomes.

## 

## 

## Different Cloud Data Warehousing Platforms

|  |  |  |
| --- | --- | --- |
|  | **Best for** | **Description** |
| **Azure Synapse Analytics** | Enterprise data warehousing | Azure Synapse Analytics is good for integrating data from hundreds of data sources across the company’s divisions, subsidiaries, etc. for analytical querying to be performed in seconds. Reporting on all management levels, from C-suite to directors, managers and supervisors, is protected with a fine-grained data access control. |
| **Amazon Redshift** | Big data warehousing | Amazon Redshift enables SQL-querying of exabytes of structured, semi-structured, and unstructured data across the data warehouse, operational data stores, and a data lake with the possibility to further aggregate data with big data analytics and ML services. |
| **Google BigQuery** | **Cost-effective storage of big volume of data with infrequent queries** | **BigQuery allows for cost-effective exabyte-scale storage with tables having up to 10,000 columns. It's most effective when main analytical queries either filter data according to partitioning or clustering or require scanning the entire dataset.** |
| **Azure SQL Database** | Midsize data warehouse | Azure SQL database is a good fit for data warehousing scenarios with up to 8 TB of data volumes and a large number of active users (concurrent requests can reach up to 6,400 with up to 30,000 concurrent sessions). |
| **Snowflake** | **Cloud-agnostic data warehouse** | **Provided as Software-as-a-Service, Snowflake enables companies to concurrently allocate compute resources of different cloud vendors (AWS, Azure, GCP) to the same database for loading and querying data with no impact on the data warehouse performance.** |
| **Azure Cosmos DB + Azure Synapse Analytics** | Operational data warehouse (hybrid transaction/analytical processing) | Azure Cosmos DB and Azure Synapse Analytics enable enterprise teams to run fast, cost-effective no-ETL queries on large operational real-time data sets, without copying data and impacting the performance of the company’s transactional workloads. |

The modeling concepts for cloud and on-premise data are almost the same.

# 

# Other References & Authors

<https://www.guru99.com/data-modelling-conceptual-logical.html>

<https://www.techtarget.com/searchdatamanagement/definition/data-modeling>

<https://opentextbc.ca/dbdesign01/chapter/chapter-5-data-modelling/>

<https://www.geeksforgeeks.org/building-blocks-of-a-data-model/>

<https://www.informit.com/articles/article.aspx?p=30646>

<https://www.educba.com/relationship-in-dbms/>

<https://beginnersbook.com/2015/04/cardinality-in-dbms/>

<https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning>

<https://www.omg.org/retail-depository/arts-odm-73/data_modeling_methodology_and_.htm>

<https://www.guru99.com/er-diagram-tutorial-dbms.html>

# Ending Note

Thank you for reading this. If you found this helpful **consider** posting about these free documents on LinkedIn.

I would love if you will write about it yourself but for the lazy folks like myself, here’s a sample LinkedIn post:

|  |
| --- |
| @Deepanshu Kalra has some amazing free resources on his website nerdsfornerds.in and I just went through one of the documents and really liked it. The document was very informative and well structured.  He has free learning documents for SQL, Python, DSA, Data Warehousing and many other amazing topics. I recommend those to anyone who is learning these topics for their upcoming interviews or for basic learning. |

**If you post this you are a hero. Thanks for helping me help others.**