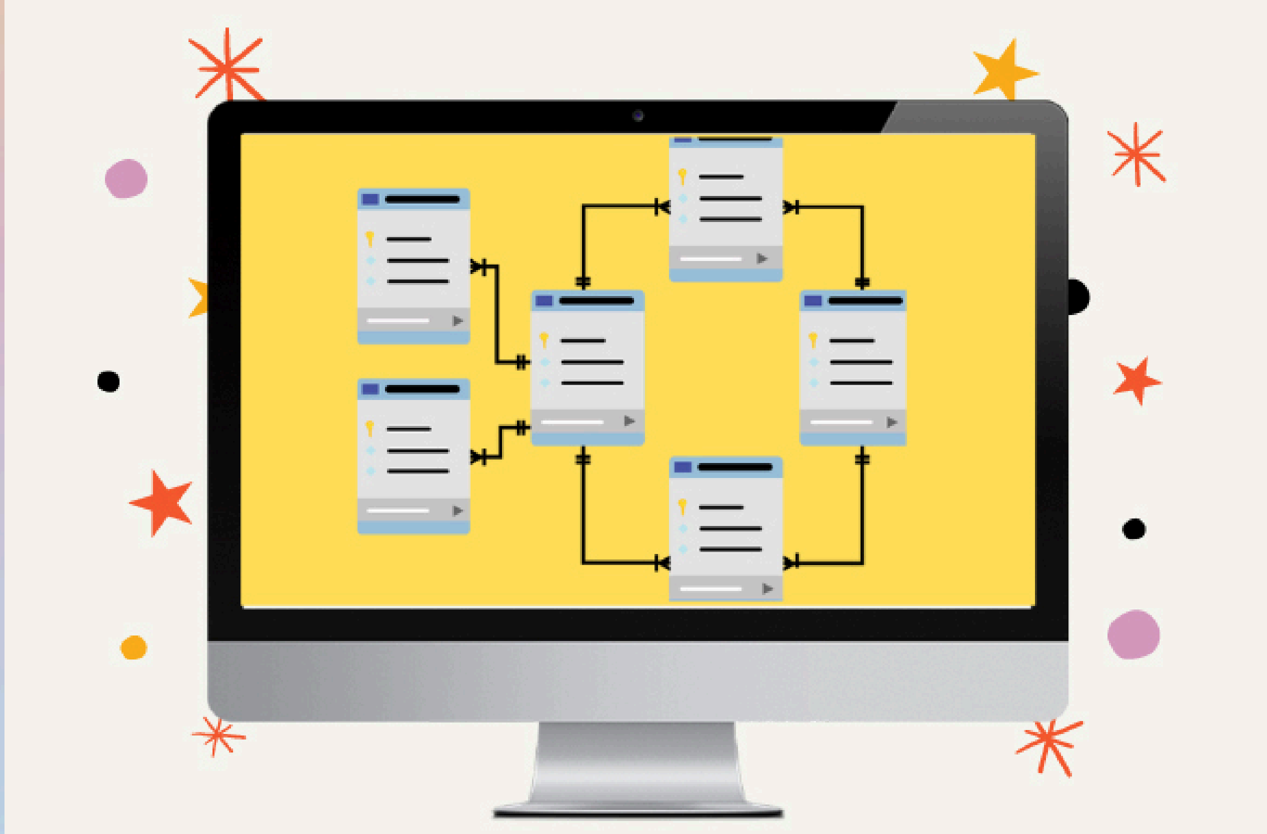
Data Modeling



Define the data model

A data model documents and organizes data, how it is stored and accessed, and the relationships among different types of data. The model may be abstract or concrete.

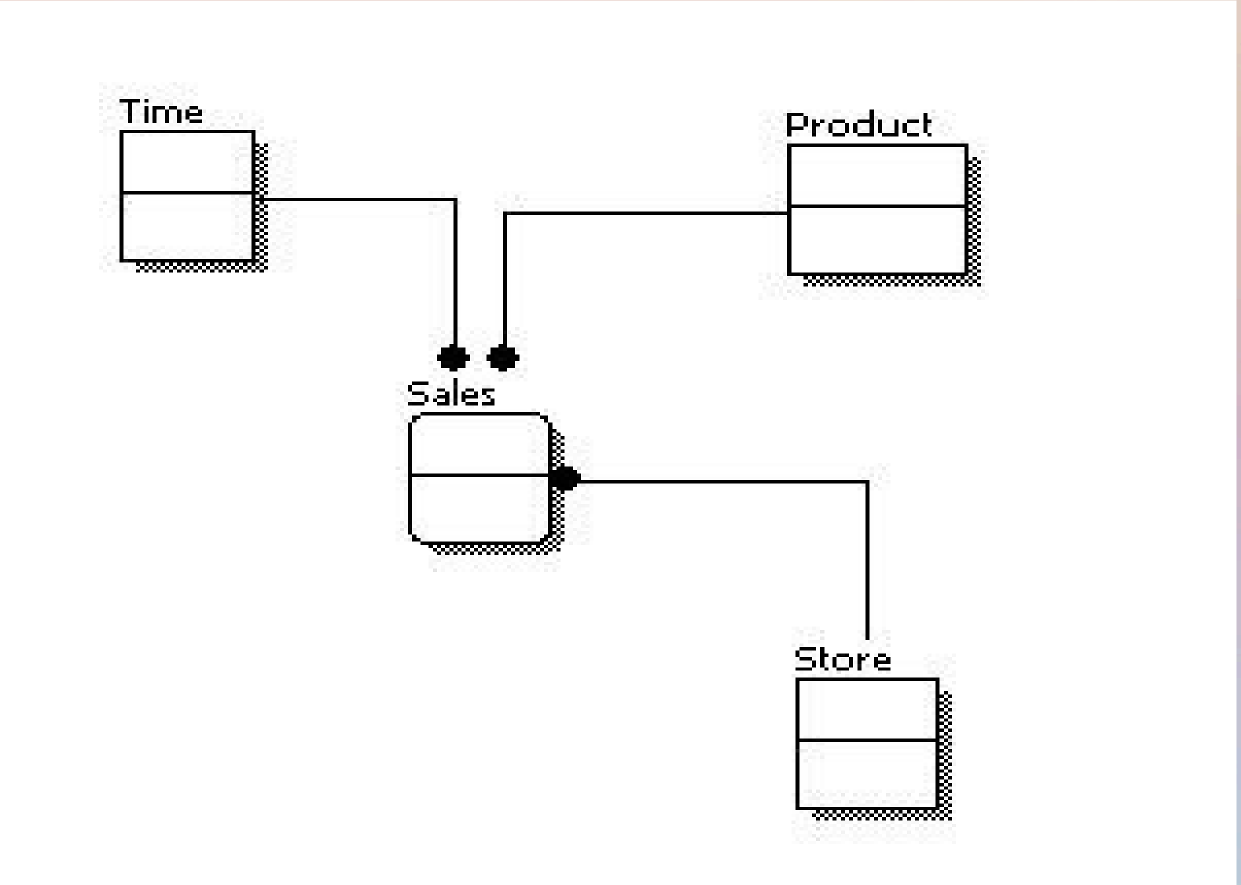
- Identify the different data components- consider raw and processed data, as well as associated metadata (these are called entities)
- Identify the relationships between the different data components (these are called associations)
- Identify anticipated uses of the data (these are called requirements), with recognition that data may be most valuable in the future for unanticipated uses
- Identify the strengths and constraints of the technology (hardware and software) that you plan to use during your project (this is called a technology assessment phase)
- Build a draft model of the entities and their relations, attempting to keep the model independent from any specific uses or technology constraints.

Data Modeling Types

- Conceptual Data
- Modeling Enterprise
- Data Modeling Logical
- Data Modeling Physical
- Data Modeling
- Relational Data
- Modeling Dimensional
- Data Modeling

Conceptual Data Modeling

- A conceptual data model identifies the highest-level relationships between the different entities. Features of conceptual data model include:
- Includes the important entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.

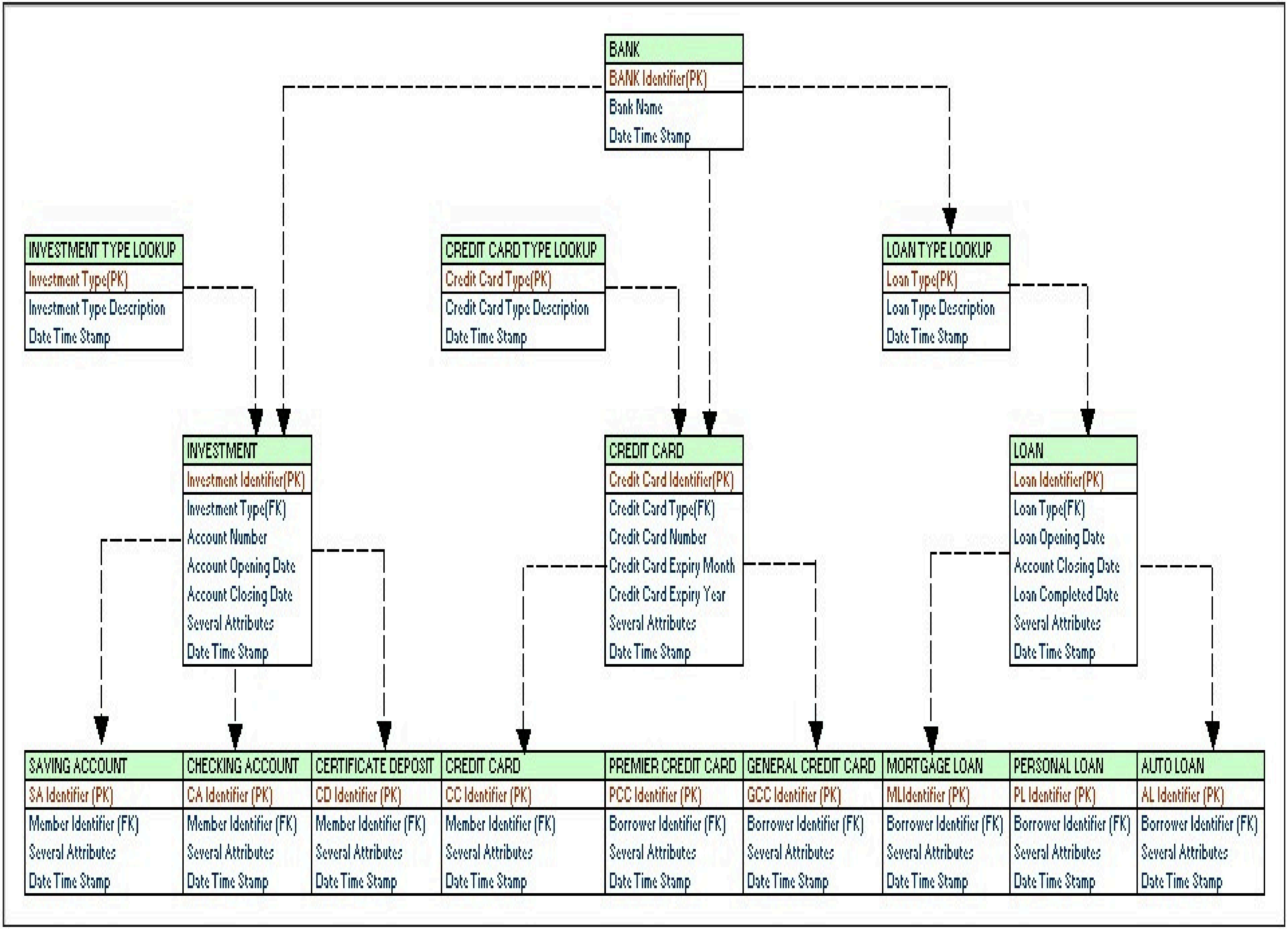


Enterprise Data Model

- Development of a common consistent view and understanding of data elements and their relationships across the enterprise is referred to as enterprise data modeling.

- This type of data modeling provides access to information scattered throughout an enterprise under the control of different divisions or departments with different databases and data models.
- Enterprise data modeling is sometimes called as global business model and the entire information about the enterprise would be captured in the forms of entities.

- When a enterprise logical data model is transformed to a physical data model, SUPERTYPES and SUBTPES may not be as is. I.e the logical and physical structure of super types and subtypes may be entirely different.(Means names of tables and columns changes and tables can break for understand the model.



BANK
BANK Identifier(PK)
Bank Name
Date Time Stamp

INVESTMENT TYPE LOOKUP
Investment Type(PK)
Investment Type Description
Date Time Stamp

CREDIT CARD TYPE LOOKUP
Credit Card Type(PK)
Credit Card Type Description
Date Time Stamp

LOAN TYPE LOOKUP
Loan Type(PK)
Loan Type Description
Date Time Stamp

INVESTMENT
Investment Identifier(PK)
Investment Type(FK)
Account Number
Account Opening Date
Account Closing Date
Several Attributes
Date Time Stamp

CREDIT CARD
Credit Card Identifier(PK)
Credit Card Type(FK)
Credit Card Number
Credit Card Expiry Month
Credit Card Expiry Year
Several Attributes
Date Time Stamp

LOAN
Loan Identifier(PK)
Loan Type(FK)
Loan Opening Date
Account Closing Date
Loan Completed Date
Several Attributes
Date Time Stamp

SAVING ACCOUNT
SA Identifier (PK)
Member Identifier (FK)
Several Attributes
Date Time Stamp

CHECKING ACCOUNT
CA Identifier (PK)
Member Identifier (FK)
Several Attributes
Date Time Stamp

CERTIFICATE DEPOSIT
CD Identifier (PK)
Member Identifier (FK)
Several Attributes
Date Time Stamp

CREDIT CARD
CC Identifier (PK)
Member Identifier (FK)
Several Attributes
Date Time Stamp

PREMIER CREDIT CARD
PCC Identifier (PK)
Borrower Identifier (FK)
Several Attributes
Date Time Stamp

GENERAL CREDIT CARD
GCC Identifier (PK)
Borrower Identifier (FK)
Several Attributes
Date Time Stamp

MORTGAGE LOAN
MLIdentifier (PK)
Borrower Identifier (FK)
Several Attributes
Date Time Stamp

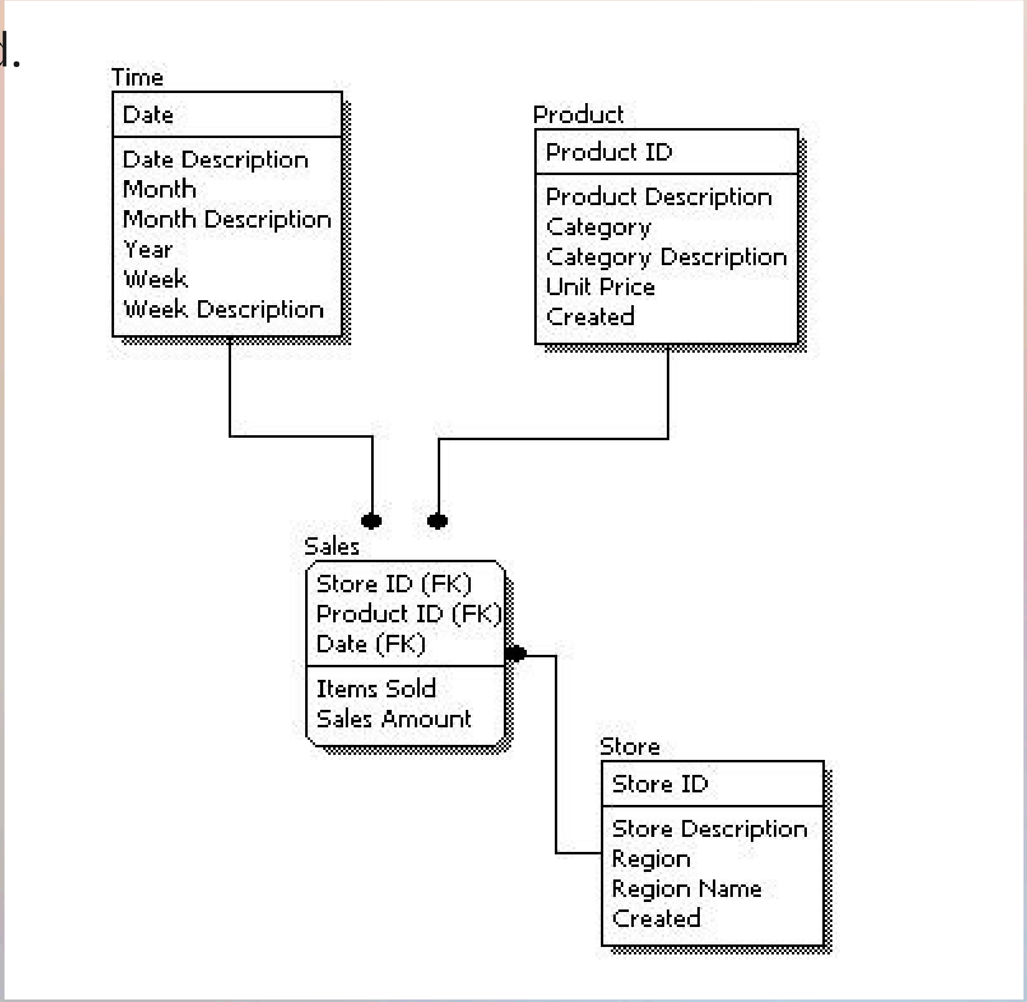
PERSONAL LOAN
PL Identifier (PK)
Borrower Identifier (FK)
Several Attributes
Date Time Stamp

AUTO LOAN
AL Identifier (PK)
Borrower Identifier (FK)
Several Attributes
Date Time Stamp

Logical Data Modeling

A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database. Features of a logical data model include:

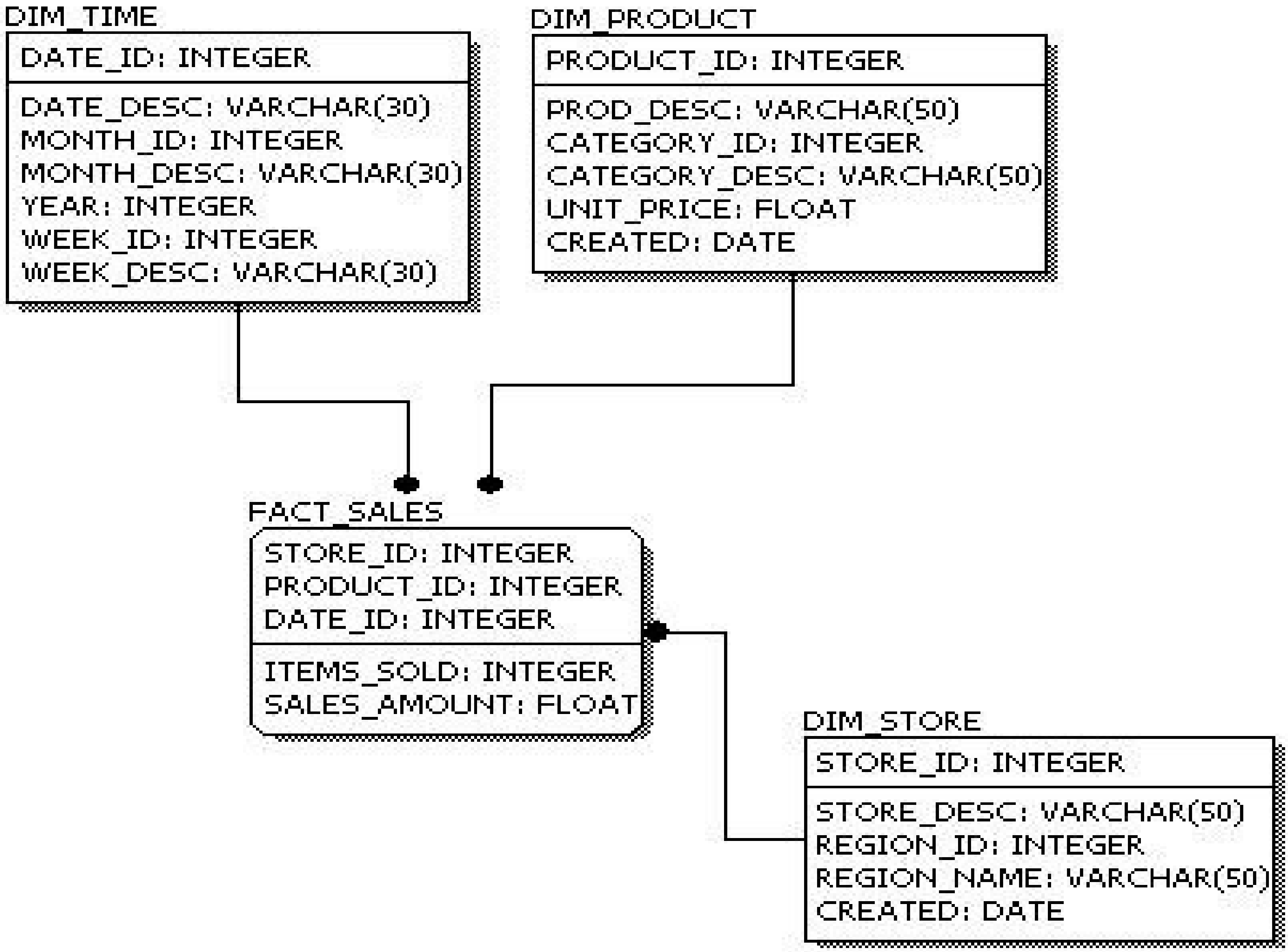
- Includes all entities and relationships among them.
- All attributes for each entity are specified.
- Primary key for each entity is specified.
- Foreign keys are specified.
- Normalization occurs at this level.



Physical Data Modeling

Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Features of a physical data model include:

- Specification all tables and columns.
- Foreign keys are used to identify relationships between tables.
- Denormalization may occur based on user requirements.
- Physical considerations may cause the physical data model to be quite different from the logical data model.
- Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL, SQL Server,Oracle,Postgres etc.



DIM_TIME

DATE_ID: INTEGER

DATE_DESC: VARCHAR(30)
MONTH_ID: INTEGER
MONTH_DESC: VARCHAR(30)
YEAR: INTEGER
WEEK_ID: INTEGER
WEEK_DESC: VARCHAR(30)

DIM_PRODUCT

PRODUCT_ID: INTEGER

PROD_DESC: VARCHAR(50)
CATEGORY_ID: INTEGER
CATEGORY_DESC: VARCHAR(50)
UNIT_PRICE: FLOAT
CREATED: DATE

FACT_SALES

STORE_ID: INTEGER
PRODUCT_ID: INTEGER
DATE_ID: INTEGER

ITEMS_SOLD: INTEGER
SALES_AMOUNT: FLOAT

DIM_STORE

STORE_ID: INTEGER

STORE_DESC: VARCHAR(50)
REGION_ID: INTEGER
REGION_NAME: VARCHAR(50)
CREATED: DATE

Relational Data Modeling

- Relational Data Model is a data model that views the real world as entities and relationships.
- Entities are concepts, real or abstract about which information is collected.
- The goal of relational data model is to normalize data and present it in a good normal form.
- Following are some of questions that arise during development of relational data model,

  What will be the future scope of the data model?

  How to normalize data ?

  How to group attribute and entities?

  How to connect one entity to other?

  How to validate data?

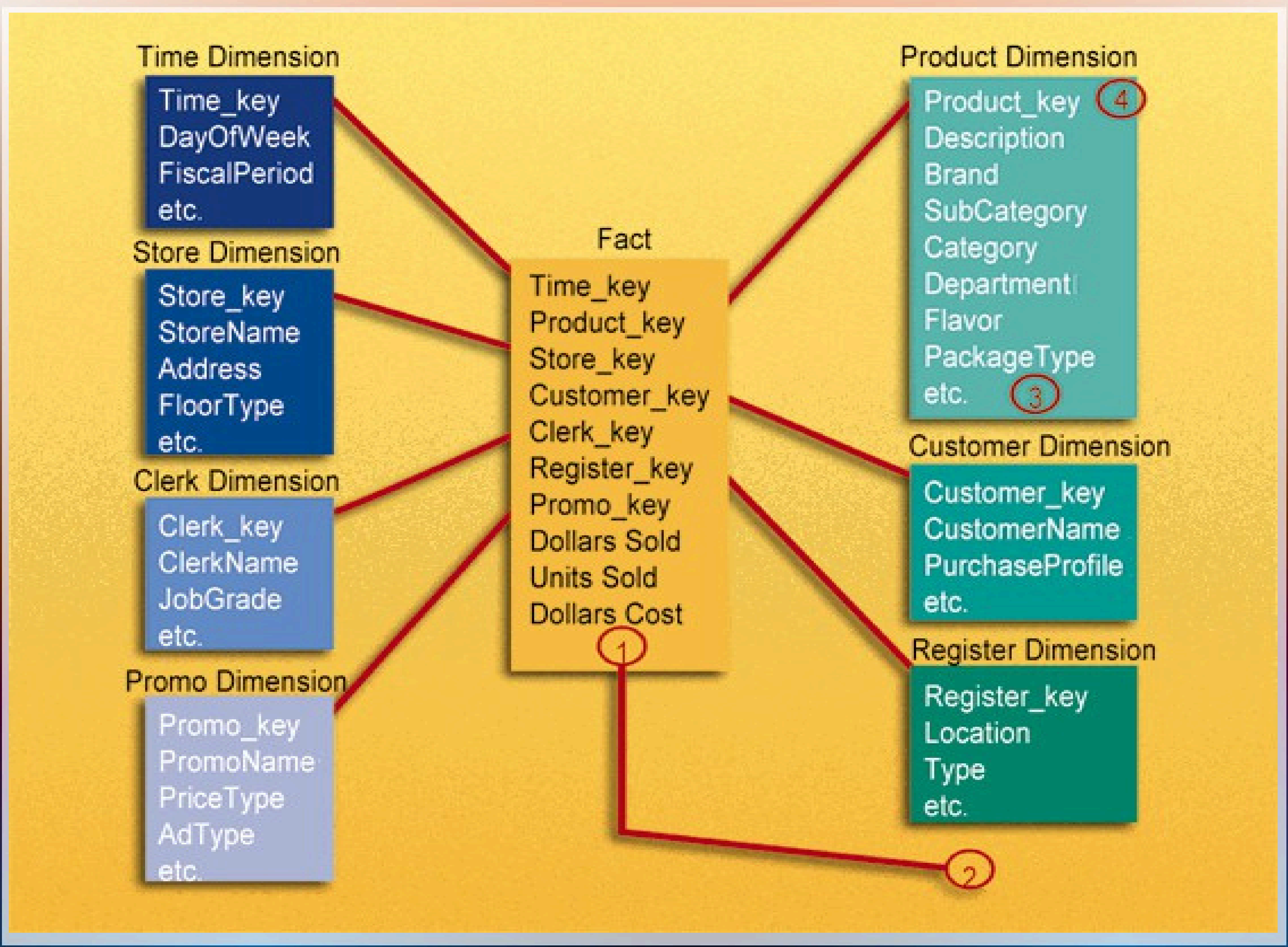
  How to present report?

Dimensional Data Modeling

- DM is a logical design technique that seeks to present the data in a standard, intuitive framework that allows for high-performance access. It is inherently dimensional, and it adheres to a discipline that uses the relational model with some important restrictions. Every dimensional model is composed of one table with a multipart key, called the fact table, and a set of smaller tables called dimension tables. Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multipart key in the fact table. This characteristic “star-like” structure is often called a star join.

- A fact table, because it has a multipart primary key made up of two or more foreign keys, always expresses a many-to-many relationship. The most useful fact tables also contain one or more numerical measures, or “facts,” that occur for the combination of keys that define each record.

- Dimension tables, by contrast, most often contain descriptive textual information. Dimension attributes are used as the source of most of the interesting constraints in data warehouse queries, and they are virtually always the source of the row headers in the SQL answer set.



Time Dimension

Time_key
DayOfWeek
FiscalPeriod
etc.

Store Dimension

Store_key
StoreName
Address
FloorType
etc.

Clerk Dimension

Clerk_key
ClerkName
JobGrade
etc.

Promo Dimension

Promo_key
PromoName
PriceType
AdType
etc.

Fact

Time_key
Product_key
Store_key
Customer_key
Clerk_key
Register_key
Promo_key
Dollars Sold
Units Sold
Dollars Cost

Product Dimension

Product_key
Description
Brand
SubCategory
Category
Department
Flavor
PackageType
etc.

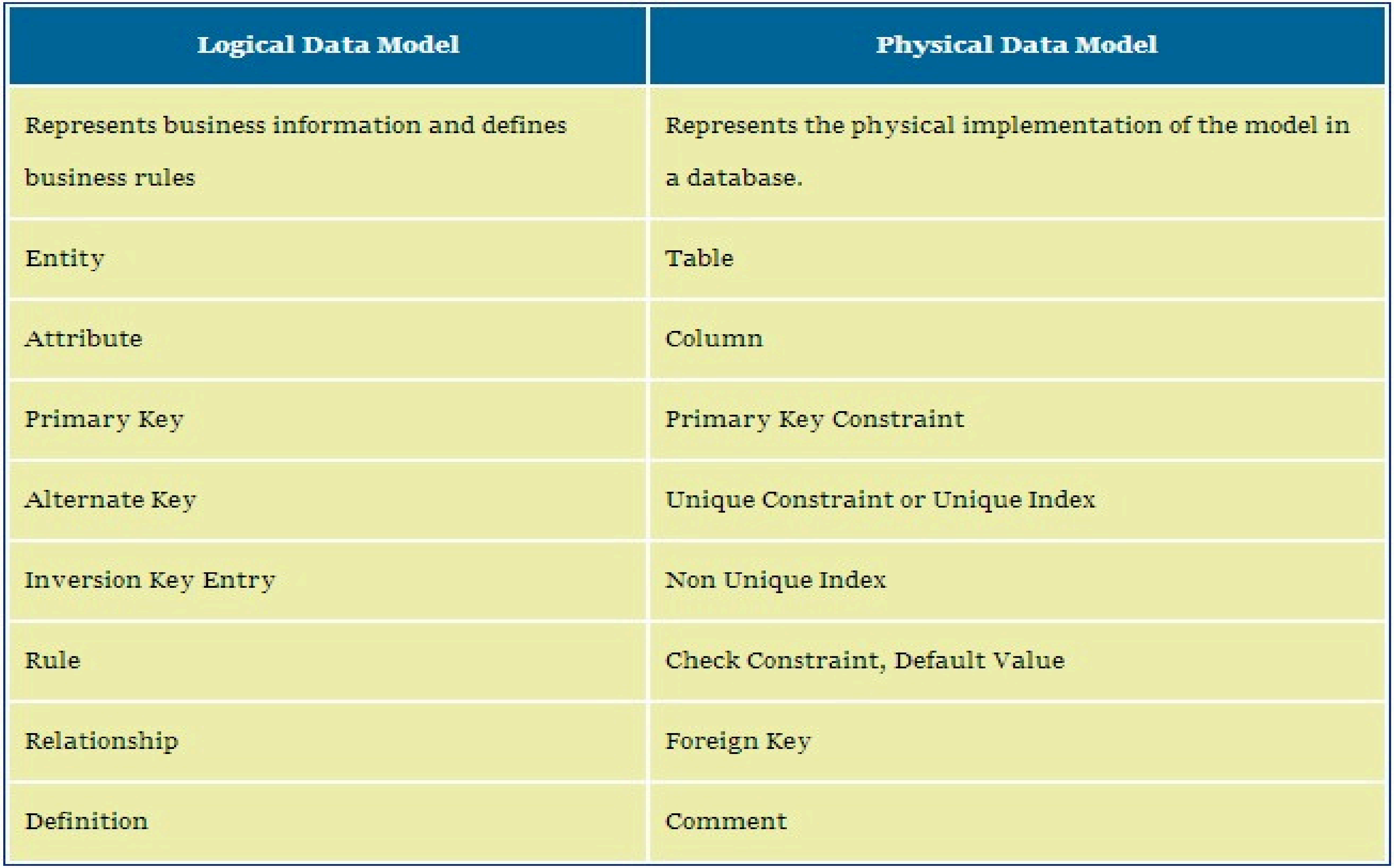
Customer Dimension

Customer_key
CustomerName
PurchaseProfile
etc.

Register Dimension

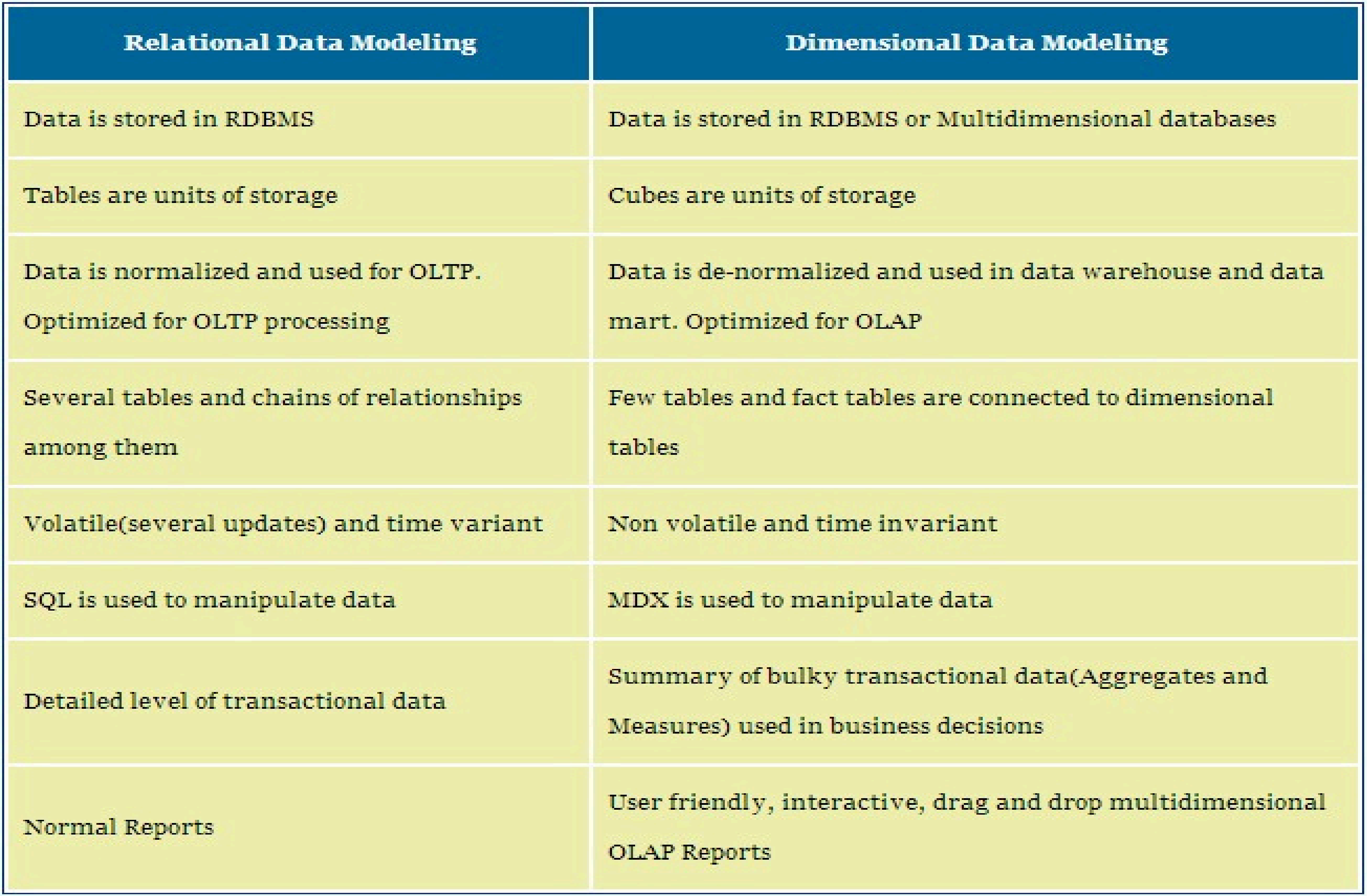
Register_key
Location
Type
etc.

Physical vs Logical



Logical Data Model	Physical Data Model
Represents business information and defines business rules	Represents the physical implementation of the model in a database.
Entity	Table
Attribute	Column
Primary Key	Primary Key Constraint
Alternate Key	Unique Constraint or Unique Index
Inversion Key Entry	Non Unique Index
Rule	Check Constraint, Default Value
Relationship	Foreign Key
Definition	Comment

Relational Vs Dimensional



Relational Data Modeling	Dimensional Data Modeling
Data is stored in RDBMS	Data is stored in RDBMS or Multidimensional databases
Tables are units of storage	Cubes are units of storage
Data is normalized and used for OLTP. Optimized for OLTP processing	Data is de-normalized and used in data warehouse and data mart. Optimized for OLAP
Several tables and chains of relationships among them	Few tables and fact tables are connected to dimensional tables
Volatile(several updates) and time variant	Non volatile and time invariant
SQL is used to manipulate data	MDX is used to manipulate data
Detailed level of transactional data	Summary of bulky transactional data(Aggregates and Measures) used in business decisions
Normal Reports	User friendly, interactive, drag and drop multidimensional OLAP Reports

Data Warehouse

- A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.
- Subject-Oriented: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

  Integrated: A data warehouse integrates data from multiple data sources. For example,

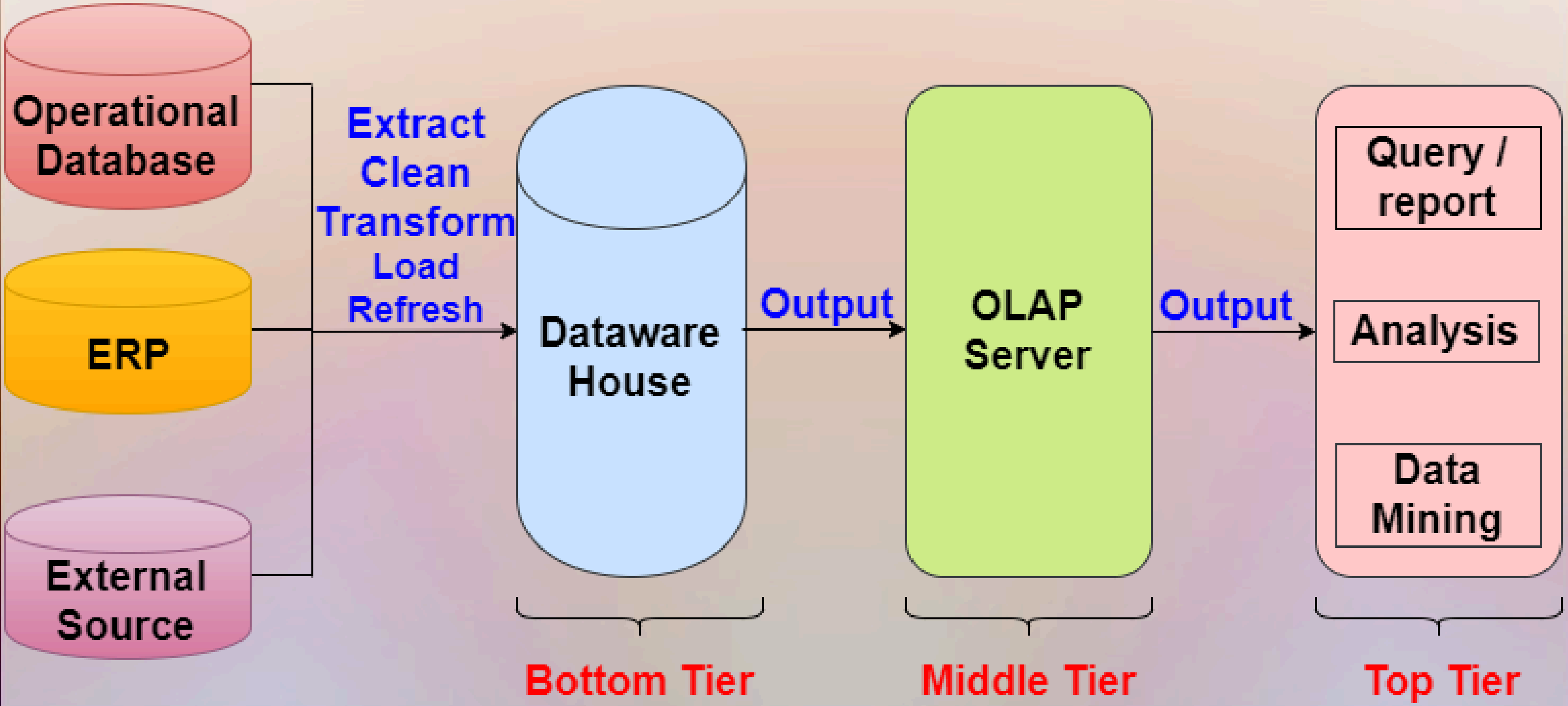
  source
- A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.
- Time-Variant: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse.

  This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.
- Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.
- A data warehouse is a copy of transaction data specifically structured for query and analysis.

Data Warehouse Architecture

All data warehouse systems have the following layers:

- Data Source Layer
- Data Extraction Layer
- Staging Area
- ETL Layer
- Data Storage Layer
- Data Logic Layer
- Data Presentation Layer
- Metadata Layer
- System Operations Layer


Operational Database
ERP
External Source
Extract
Clean
Transform
Load
Refresh
Dataware House
Output
OLAP Server
Output
Query / report
Analysis
Data Mining
Bottom Tier
Middle Tier
Top Tier

Data Source Layer

- This represents the different data sources that feed data into the data warehouse. The data source can be of any format -- plain text file, relational database, other types of database, Excel file, etc., can all act as a data source.
- Many different types of data can be a data source:
- Operations -- such as sales data, HR data, product data, inventory data, marketing data, systems data.
- Web server logs with user browsing data.
- Internal market research data.
- Third-party data, such as census data, demographics data, or survey data.

- **Data Extraction Layer**

  Data gets pulled from the data source into the data warehouse system. There is likely some minimal data cleansing, but there is unlikely any major data transformation.

- **Staging Area**

  This is where data sits prior to being scrubbed and transformed into a data warehouse / data mart. Having one common area makes it easier for subsequent data processing / integration.

- **ETL Layer**

  This is where data gains its "intelligence", as logic is applied to transform the data from a transactional nature to an analytical nature. This layer is also where data cleansing happens. The ETL design phase is often the most time-consuming phase in a data warehousing project, and an ETL tool is often used in this layer.

- **Data Storage Layer**

  This is where the transformed and cleansed data sit. Based on scope and functionality, 3 types of entities can be found here: data warehouse, data mart, and operational data store (ODS). In any given system, you may have just one of the three, two of the three, or all three types.

- **Data Logic Layer**

  This is where business rules are stored. Business rules stored here do not affect the underlying data transformation rules, but do affect what the report looks like.

- **Data Presentation Layer**

  This refers to the information that reaches the users. This can be in a form of a tabular / graphical report in a browser, an emailed report that gets automatically generated and sent everyday, or an alert that warns users of exceptions, among others. Usually an OLAP tool and/or a reporting tool is used in this layer.

- **Metadata Layer**

  This is where information about the data stored in the data warehouse system is stored. A logical data model would be an example of something that's in the metadata layer. A metadata tool is often used to manage metadata.

- **System Operations Layer**

  This layer includes information on how the data warehouse system operates, such as ETL job status, system performance, and user access history.

Database Vs Data Warehouse

Database:

- Used for Online Transactional Processing (OLTP). This records the data from the user for history. The tables and joins are complex since they are normalized. This is done to reduce redundant data and to save storage space.
- Entity – Relational modeling techniques are used for database design.
- Optimized for write operation.
- Performance is low for analysis queries.

Data Warehouse:

- Used for Online Analytical Processing (OLAP). This reads the historical data for the Users for business decisions.
- The Tables and joins are simple since they are de-normalized. This is done to reduce the response time for analytical queries.
- Data – Modeling techniques are used for the Data Warehouse design.
- Optimized for read operations.
- High performance for analytical queries.
- General Data Flow – (Ex: Online Insurance Registration)

Thanks for reading !

Follow my profile for more coding related contents

Like Comment Share

@rganesh0203