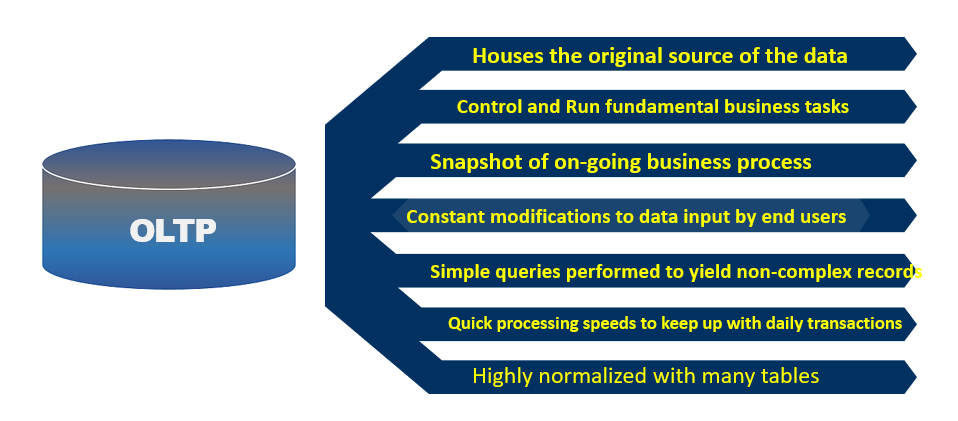
OLTP/OLAP

* Data processing solutions often fall into one of two broad categories: analytical systems, and transaction processing systems.
  + OLTP: Online is a transactional processing.
  + OLAP: Online analytical processing system

**OLTP:**

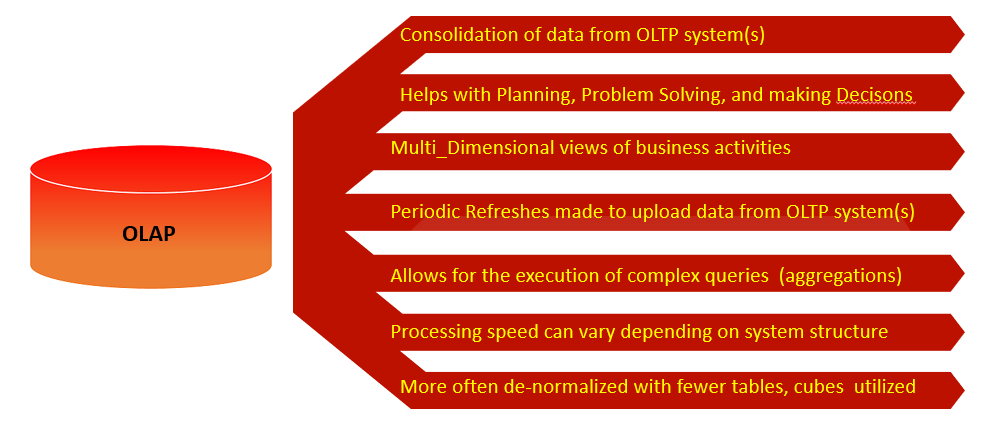
* OLTP's are operational systems which help execute and record the day-to-day operations of a business



* Normalization is done to reduce data redundancy. Data Divided into many relational tables
* Normalization maximizes speed of processing as DML operations touch single transaction, so fast throughput for transactions but it can make querying more complex as it involves joins on multiple tables.

**OLAP:**

* Analytical system is designed to support business users who need to query data and gain a *big picture* view of the information held in a database.



* De-normalization introduces data redundancy but simplifies the data Model.
* It has Faster query response time as less joins are involved.
* Most analytical data processing systems need to perform similar tasks: data ingestion, data transformation, data querying, and data visualization.

Table

Description automatically generated

**Traditional Data Warehouse**

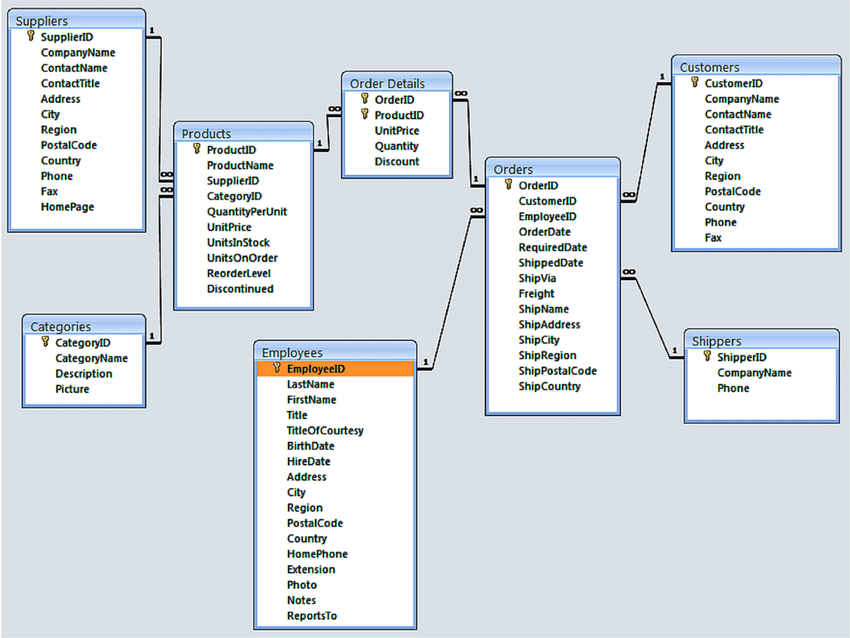
Diagram

Description automatically generated

Understanding Dimension Modeling

**Normal Form Model:**

* Reduce Data Redundancy
* Improve Data Integrity



**Dimension Modeling :**

* Database design method optimized for data warehouse Solutions.
* A set of guidelines to design database table structure for **easier** and **faster data retrieval**
* Database is **optimized** for **read operations** to support end user queries. Best used for **Reporting**
* **Dimensions** and **Facts** are basic building blocks of Datawarehouse.
* **Facts** Provide Quantitative information about Business Process
* **Dimensions** provide Descriptive Information about business process.

|  |  |
| --- | --- |
| **Graphical user interface, application  Description automatically generated** | Diagram  Description automatically generated |

**Fact Table:**

* Fact Table is a table which contains Measures which need to be analyzed.

**Example:** Sales Amount, Total earnings, Profit, stock balances, exchange rates, temperatures

* It contains numeric measure column which can be aggregated for analysis and **dimension key columns** that relate to dimension tables
* Each fact table relates to one or more dimensions tables using Foreign Keys.
* Granularity is defined by related dimensions.
* It contains Primary key to uniquely Identify the row

Example:

|  |
| --- |
| FactSales |
| CustomerId |
| ProductId |
| TimeId |
| SalesAmount |
| SalesQty |

**Dimension table:**

* Dimensions are descriptors which define the facts.

Example: Product, Customer, Store dimension

* It contains the attributes based on which you need to summarize or analyze the data.

Example: Analysing in details about the **Customers** who bought items from specific store based on **(Age,Gender,Address,distanceFromStore etc)**

* It contains textual, descriptive information associated with business process measurement event.
* It answers questions like who, what, when, where, how, why associated with event.
* It allows us to Filter, Group, slice and dice the data.
* They are generally Denormalized.
* The most consistent table you'll find in a star schema is a **date dimension** table

Example:People, Product, Place, Time, Customer etc.

|  |
| --- |
| Product |
| ProductId |
| ProductName |
| Colour |

Generally, dimension tables contain a relatively small number of rows. Fact tables, on the other hand, can contain a very large number of rows and continue to grow over time.

**Normal Form Model vs Dimension Model**

|  |  |  |
| --- | --- | --- |
|  | Normal Form Model | Dimension Model |
| Purpose | Transaction System  Designed for less redundancy and Data integrity | Reporting  Designed for Faster retrieval |
| Structure | Complex | Less Complex (Denormalized Data) |
| Operations | Transactional operations  (Insert, Update, Delete, Select) | Mostly Select |
| Performance | Good  More joins involved | Better  Less Joins involved |

***Assignment :*** *Identify Dimension and Fact*

***Retail store contains transactional information about the Sale***

***It contains information about Date, Store, Product, Quantity, Unit Price, Sales Amount, Invoice number***

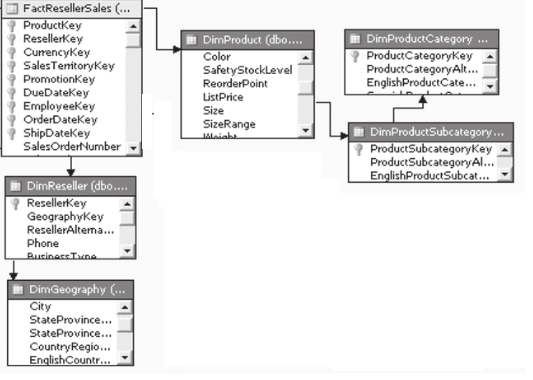
**Star Schema:**

* A star schema is a type of relational database schema that is composed of a single, central fact table that is surrounded by dimension tables.
* A star schema can have any number of dimension tables. The branches at the end of the links connecting the tables indicate a many-to-one relationship between the fact table and each dimension table.

|  |  |
| --- | --- |
| **Graphical user interface, application  Description automatically generated** | Diagram  Description automatically generated |

**SnowFlake Schema:**

* The *snowflake schema* consists of one fact table that is connected to many dimension tables, which can be connected to other dimension tables through a many-to-one relationship.
* Tables in a snowflake schema are usually normalized to the third normal form. Each dimension table represents exactly one level in a hierarchy.



**Data Mart:**

* Data Mart is a subset of Data Warehouse that is generally oriented for a specific purpose, i.e designed for a particular department of any organization
* Data Marts are built for specific user groups.  They contain a subset of rows and columns that are of interest to the particular audience.
* By providing decision makers with only a subset of the data from the Data Warehouse, privacy, performance, and clarity objectives can be attained.

**Example:** The Sales department manager needs to see data on products, customers, and the sales team's performance metrics.

Dimension Modelling Design Process

Identify Granularity

Build Schema

Identify Fact

Identify Dimension and Attributes

Select Business Process

**Business Process:**

* Activity performed by company to accomplish certain Goal.

Example: Taking phone orders, Receiving payments, Receiving support calls,Paying to the vendors

* The process measures something like Sales, Profit etc

**Identifying the Grain(Level of Detail):**

* What is the detail level of at which facts are measured ?
* The purpose is to describe, how a single row in fact table look like?
* Grain must be declared before choosing dimensions or facts.

|  |  |  |
| --- | --- | --- |
| Example | Grain | Question |
| Analyse Sales Per day  Sales.Day | One row at the end of the day | Is information about how much each customer paid important?  Do you want to analyse sales per customer,per day,per time? |
| Sales/Day/Customer/Order | One row for every finished order | Is information about, What was the most sold product,Important?  Do you need to analyse  Avg no of products sold in each order |
| Sales/Day/Customer/Order/Product | One row for every product sold within an order | Is this information enough? |

**Identify Dimensions:**

* How the Data need to be descried in the business process?
* These are answers to questions like Who, What, Where, When, How etc. associated with action or event

**Identify Facts:**

* What we need to measure?
* All the measurement(facts) need to have same granularity.

Example: SalesAmount, Unit Sold

Understanding Dimension

**Dimension Table Components**

**Surrogate Key/Non-Natural :**

* Sequential integer values,automatically incremented when row is added
* No business meaning
* Used to join to fact tables

**Why Surrogate Keys?**

* Allows data from multiple sources which might have overlapping primary keys
* Track changes to an attribute over time
* An add additional dimension value that not exists in source table(Handle Null or Unknown).Handles situation where link does not exists between Fact and Dimension.
* Improves performance.

|  |  |
| --- | --- |
|  | Table  Description automatically generated with medium confidence |

**Natural Key/Business Keys:**

* Primary key value from source table
* Ties dimension record back to source
* If data coming from multiple sources it can be prefixed with source system code(POS|432 OR CSV|546)
* Modeled as attribute in dimension table.

**Attributes:**

* Descriptive characteristics used by end users to group, filter and report on data.

Example: Age, Gender

**DateStamps:**

* Date and time columns helps in auditing, tracking history of data.

Example:

**Auditing:**Insert\_dt,Update\_dt

**Tracking History: Start**\_dt,End\_dt

Slowly Changing Dimensions

* Slowly changing dimensions (SCD) are tables in a dimensional model that handle changes to dimension values over time.
* Example: Customer phone no or Company name might change.
* You need a way to update the existing record or to keep track of all historical changes to the values.

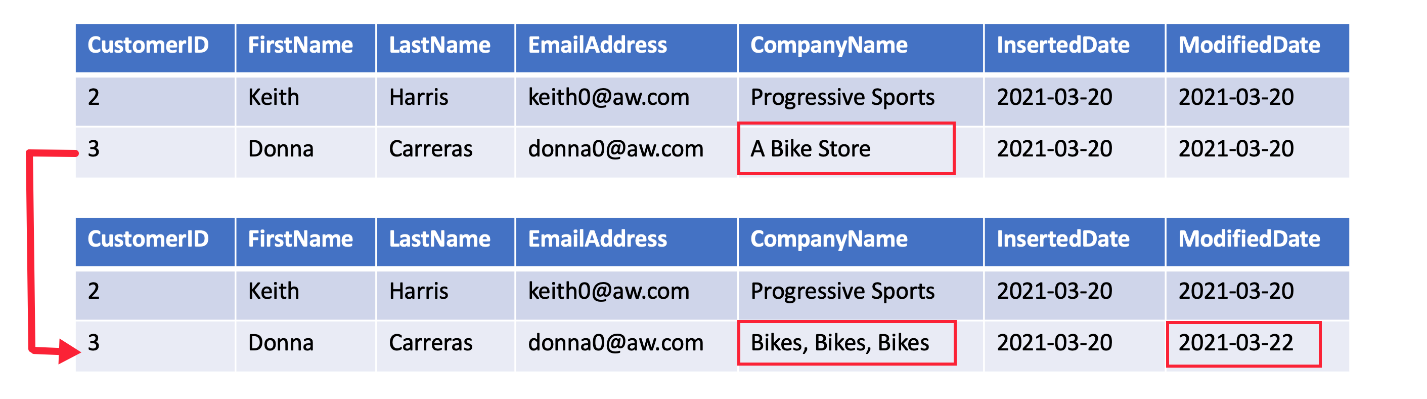


* **(1)** represents the columns where values might change over a period of time
* (2) represents the columns which can be used to track changes.

**SCD Types:**

**Type 1 SCD**

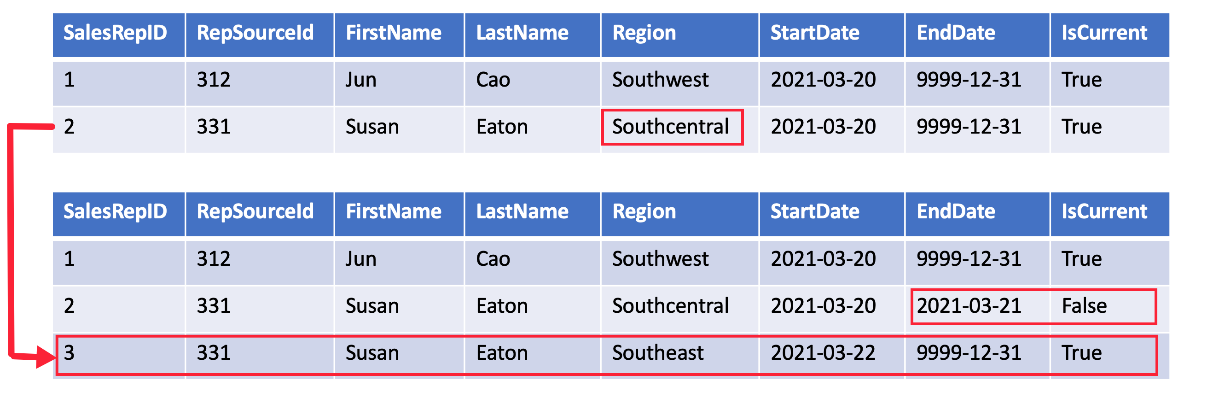
* A **Type 1 SCD** always reflects the latest values, and when changes in source data are detected, the dimension table data is overwritten.
* This design approach is common for columns that store supplementary values, like the email address or phone number of a customer. It is Simple and easy to implement

.

* The key field, such as CustomerID, would stay the same so the records in the fact table automatically link to the updated customer record.
* This is useful when historical reporting is not needed or when we just want to support Spelling correction.

**Type 2 SCD**

* A **Type 2 SCD** supports versioning of dimension members. Often the source system doesn't store versions, so the data warehouse load process detects and manages changes in a dimension table
* Dimension table must use a surrogate key to provide a unique reference to a version of the dimension member
* It also includes columns that define the date range validity of the version (for example, StartDate and EndDate) and possibly a flag column (for example, IsCurrent) to easily filter by current dimension members.



* Changes are recorder by updating EndDate and Inserting new row for changed version of record.
* Related facts must use a time-based lookup to retrieve the dimension key value relevant to the fact date.
* It preserves unlimited history.

## Type 3 SCD

* A **Type 3 SCD** supports storing two versions of a dimension member as separate columns.



* Type 3 uses additional columns to track one key instance of history, rather than storing additional rows to track each change like in a Type 2 SCD.
* This type of tracking may be used for one or two columns in a dimension table.
* Limited History is preserved.