**What is MDM? How to implement?**

MDM is unified data model (Master data Management).Instead of having varied snapshot of important information it is advisable to have Master data for that entity and refer to the same

1. Identify Master data entity with below points

- Data needs to be less volatile compare to transaction data (Which changes less compare to facts)

- Cardinality is more (Number of records are more for i.e. customers)

- Data changes over time then it should be qualified for MDM

- Reuse of Data from MDM (Reusability is high)

**SSIS**

**-error handling**

**-Identifying bottlenecks**

**-Source and DEstination**

**-Nested join and Hash Joins**

**-Multiple source in SSIS jobs**

**-How to optimize job which take huge amount of time?**

**-Challenging job**

**-Incremental load**

**-Protection level**

**-DEployment**

**Evaluation Operation Properties :** Can define the control flow when particular component fails or success it redirect the controls.Allows to define the logical flow of control

**Snowflake and Star Schema :** The most important difference is that the dimension tables in the snowflake schema are normalized. Main differences are with normalization of dimension and Query Complexity

**Ralph vs Inmon**

Kimball uses the dimensional model such as star schemas or snowflakes to organize the data in dimensional data warehouse while Inmon uses ER model in enterprise data warehouse. Inmon only uses dimensional model for data marts only while Kimball uses it for all data

Inmon uses data marts as physical separation from enterprise data warehouse and they are built for departmental uses. While in Kimball’s architecture, it is unnecessary to separate the data marts from the dimensional data warehouse.

In dimensional data warehouse of Kimball, analytic systems can access data directly. While in Inmon’s architecture, analytic systems can only access data in enterprise data warehouse via data marts.

**Partition in SQL Server**

**Duplicates in Job**

**Challange in ETL**

**Error Handling**

**Realtime SSIS Package:**

**SSIS Task vs SSIS Components vs SSIS Transformations**

**Control Flow Elements:** Data Flow Tasks, Data preparation Tasks, Workflow Tasks, Scripting and maintenance tasks.

**Steps to design DW from OLTP**

**In general, building any data warehouse consists of the following steps:**

1. Extracting the transactional data from the data sources into a staging area
2. Transforming the transactional data
3. Loading the transformed data into a dimensional database
4. Building pre-calculated summary values to speed up report generation
5. Building (or purchasing) a front-end reporting tool
6. **Analysis** : Requirement gathering for BI reports and business models. Understanding the objective for which we are designing DW. Analysis process is completely taken care by analysts who will be having a sound knowledge of the process to be followed. In general, this phase is carried forward by concerned personnel sitting at client end.
7. **Design**: Identify the source databases or data, understanding and defining hierarchies. Identifying measures, identifying dimensions and their hierarchies, designing and creating physical models by defining schema of fact and dimension tables.
   1. **Design Staging area for Data**
   2. **Facts and dimensions for Data**
   3. **Data presentation Area where data will be in normalized form.**
8. **Data Gathering:** This is one of the important step in designing the good Data Warehouse system.This is step is nothing but ETL development i.e., Extraction Transformation and Loading.Here the Data is Extracted and by using the transformations we convert this data to normalized form.The normalized format is not so good for understanding purpose.So we go for Data Presentation Area. Data gathering phase involves extracting and cleaning data to prepare it for loading into the warehouse. This phase is called low level design.
9. **Data mart creation:** Here we convert the Normalized model to Dimensional model where the user can understand very easily.Here the data will be in data marts. By combining the data marts we can have a Data Warehouse.

Furthermore,

The last phase, loading phase, is to load the prepared data into the data warehouse.

- Granularity level of dataware house - Requirement speccifci

- CDC of given data.OLTP system which captures change of data and upload perodically to DW can be covered under SCD

**Conformed Dimension:** In data warehousing, a conformed dimension is a dimension that has the same meaning to every fact with which it relates. Conformed dimensions allow facts and measures to be categorized and described in the same way across multiple facts and/or data marts,

**Conformed Fact**: A conformed fact is a shared fact that is designed to be used in the same way. across multiple data marts.

**SCD Type:**

Type 1: overwrite

Type 2: add new row (With Is Active or Effective Dates Column)

Type 3: add new attribute (Add new column to capture the change i.e having previous stateof customer and current state of customer)

Type 4: add history table

**Benefits of Surrogate Keys:**

**Performance**: Surrogate key is always numeric key unlike any production key which is alphanumeric so it can be good qualifier to create index and use in join statements

No Possibilities of duplication and maintenance is quite easy as no composite key creation required.

**Immutability**: Keys can’t be changed

Stay as it is with **requirement changes**( Natural keys can become duplicate on merger surrogate cant)

**Flow of Data**:

**Landing(Extraction from Source)**

**Staging(Data sits in one format from source)**

Data staging is an area where you hold the data temporary on data warehouse server. Data staging includes following steps

Source data extraction and data transformation ( restructuring )

Data transformation (data cleansing, value transformation )

Surrogate key assignments

**ODS (Transformations)**

**DW(Data presentation area)**

**Correlated Subquery to generate running total :**

select Qty,

(select sum(Qty) from t where Qty<=t1.Qty)

from t t1 order by Qty

**Fact less Fact:** Facts which does not contain any mesures and Count keyword only we can operate.This table only contains IDs which links to diemsnion.

Optimize a Query:

-Correct use of where filter

-removing unnecessary tables

-Remove calculated fields from join

-Use temp tables when you are joining large tables

-Using exists when you want to just check existence

-Correct use of IN and JOIN clause

-Usage of functions in Where Clause

-Usage of different data type values in WHERE clause comparisons

-Correlated subqueries removal

- Try to avoid deadlocks

-

**Nested Loops Usage in SQL Server**

Nested Loops Join operates on two tables, an outer table and an inner table, as indicated in the image used to represent this type of operation. Essentially this means that for each row in the outer table (outer loop), SQL Server will perform a criteria based comparison against each row of the inner table (inner loop). The Nested Loops join is sometimes referred to as Nested Iteration Operator.

Nested Loops joins are most efficient when you have a small outer table (not necessarily indexed), and a large but importantly indexed inner table. This will allow SQL Server to very quickly retrieve the required values of the inner (larger) table, even if the smaller (outer) table does not have an index that matches the operator’s requirement. Having an index that supports the criteria of the Nested Loop Join operator on the inner table is crucial in order to maintain an acceptable level of performance.

**When is Hash Match used by SQL Server’s Execution Plan**

A Hash Match is likely to pop-up in the following scenarios:

No covering index is available on the tables involved in join, union or aggregation operations.

One large table is being joined against a much small table, Hash Match sometimes proves to be very efficient in these cases.

**Hash Join**: The most expensive joining methodology. This takes place when the joining columns between two tables are not indexed. Creating indexes on those columns will enable you to get rid of it.

**Nested Loops**: Most cases, this happens when a non-clustered index does not include (Cover) a column that is used in the SELECT column list. In this case, for each member in the non-clustered index column, the database server has to seek into the clustered index to retrieve the other column value specified in the SELECT list. Creating a covered index will enable you to get rid of it.

**CDC**: <https://www.simple-talk.com/sql/learn-sql-server/introduction-to-change-data-capture-cdc-in-sql-server-2008/>

**What are the names of the layers in ETL?**

The first layer in ETL is the source layer and it is the layer where data lands. Second layer is integration layer where data is stored after transformation. Third layer is the dimension layer where actual presentation layer stands.

<http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>

<http://sqlmag.com/database-administration/7-steps-data-warehousing>