**What are the types of backups?**

**Backups are used to make a compressed copy of the data in a database**

Only backup data if it is online, any offline databases can’t be backed up??

-**Full Backup**: Copies all the data in a specific database with enough logs for recovering data

-**Differential Backup**: Records all the data that has been changed or modified since the last Full Backup

-**Transaction Log Backup**: Records all the transaction logs that were not

-**Tail-Backup**: Records the latest log records that have not yet been backed up to prevent data loss

backed up in a previous Log Backup

**What are the recovery models?**

Recovery models are designed to **control transaction log maintenance**.

A recovery model is a database property that controls **how transactions are logged, whether the transaction log requires (and allows) backing up, and what kinds of restore operations are available**. Three recovery models exist: simple, full, and bulk-logged. Typically, a database uses the full recovery model or simple recovery model. A database can be switched to another recovery model at any time.

SQL Server **backup** and **restore** operations **occur within the context of the recovery model of the database.**

**Simple recovery model:**

-This recovery model do not require the backup of transaction log.

-Reclaims space used by logs to keep space requirements small

-Unable to use Log Shipping, AlwaysOn, Database Mirroring, Point in Time Restores, and Media Recovery???

-Operations that require transaction log backups are not supported by the simple recovery model

**Full recovery model**

-Requires log backups

-No work is lost due to a lost or damaged data file.

-Can recover to an arbitrary point in time (for example, prior to application or user error).

**Bulk logged recovery model**

Requires log backups.

An adjunct of the full recovery model that permits high-performance bulk copy operations.

Reduces log space usage by using minimal logging for most bulk operations. For information about operations that can be minimally logged, see The Transaction Log (SQL Server).

**SQL Server Job Agent**

* Tool in SQL that allows one to execute **scheduled** admin tasks or syntax
* Great for performing backups late at night or doing long tasks that would take too much time to execute normally

**System databases**

|  |  |
| --- | --- |
| [master Database](https://msdn.microsoft.com/en-us/library/ms187837.aspx) | Records all the system-level information for an instance of SQL Server.  (This includes instance-wide meadata such as logon accounts, endpoints, linked servers, and system configuration settings. Also, **master** is the database that records the existence of all other databases and the location of those database files and records the initialization information for SQL Server. Therefore, SQL Server cannot start if the **master** database is unavailable.) |
| [msdb Database](https://msdn.microsoft.com/en-us/library/ms187112.aspx) | Is used by SQL Server Agent for scheduling alerts and jobs.  (For example, SQL Server automatically maintains a complete online backup-and-restore history within tables in **msdb**. This information includes the name of the party that performed the backup, the time of the backup, and the devices or files where the backup is stored.) |
| [model Database](https://msdn.microsoft.com/en-us/library/ms186388.aspx) | Is used as the template for all databases created on the instance of SQL Server. Modifications made to the **model** database, such as database size, collation, recovery model, and other database options, are applied to any databases created afterward. |
| [Resource Database](https://msdn.microsoft.com/en-us/library/ms190940.aspx) | Is a read-only database that contains system objects that are included with SQL Server. System objects are physically persisted in the **Resource** database, but they logically appear in the **sys** schema of every database. |
| [tempdb Database](https://msdn.microsoft.com/en-us/library/ms190768.aspx) | Is a workspace for holding temporary objects or intermediate result sets. |

**Bulk Copy Program(BCP)**

BCP uses the CMD of the system, so it is command line driven

Since it uses the CMD, it is avoiding any GUI or software, making it the fastest way for communication between DBMS and RDBMS

Insert and Output

**Bulk Insert (BI)**

Bulk Insert is executed in SQL server and used in a query

Can only INSERT data into SQL, no output

Since being used in queries, it can be logged, use transactions, error handling

Slower than Bulk Copy Program

**Profiler and DTA**

SQL Server Profiler can be used to create and manage traces of a query. It will step through every events that related to execution of a slow running and return a trace file that can later be analyzed database tuning advisor

Query optimizer

-->

Create indexes, or update statistics if the index is already created, check whether there is significant index fragmentation, if yes, reorganize or rebuild

-->

Adjust queries (replace set, sub queries with joins if possible, replace cursor with while loop if possible, use three part naming convention whenever possible, avoid using order by)

-->

If it is a DML operation, check if there are any trigger that drag down the performance, try to remove trigger if possible

Trigger on table, partition table

-->

If this query is used frequent. Consider use Stored procedure, user defined function which enable cached execution plan.

If it is already in the form of store procedure, then make sure the cached exec plan is update.

-->

If we are query a huge data, then consider partition this big table into small groups.

Compare OLTP and OLAP

* OLTP stands for on-line transaction processing, it features large amount online transaction. Common task include insert, update and delete. To design a database is to design a OLTP. An OLTP database stores more recent data. In order to avoid update anomaly and save storage,  unusually the tables are normalized into 3NF
* OLAP stands for on-line analytical processing. It features data analysis on based on historical data, extract business info from past trend and support decisions. Query response time is very important for OLAP systems, therefore tables are often de-normalized

What is dimension modelling?

Dimensional Modeling is the process of designing a data warehouse, which is used to store historical data for analytical reasons.

Generally, it start with identifying the business requirement. What people from different department expect from a new dw. The next is to choose granularity level, which is all about to choose proper time frame to measure data. The final stage is to design the dimensions and facts in dw.

What is ETL?

A process to move data into a data warehouse. ETL stands for extract, transform and load.

Extract just means to grab data from different sources, the sources can be flat files or databases.

Transform is to re-organize the data so that they are consistent and uniform. During Transformation, the common task are data cleansing, data profiling as well data validation

Load just means to populate the data warehousing on the transform finishes.

What is data profiling?

**Data profiling** is the process of examining the data available in an existing data source (e.g. a [database](https://en.wikipedia.org/wiki/Database) or a [file](https://en.wikipedia.org/wiki/Computer_file)) and collecting [statistics](https://en.wikipedia.org/wiki/Descriptive_statistics) and information about that data. The purpose of these statistics may be to:

1. Find out whether existing data can easily be used for other purposes
2. Improve the ability to search the data by [tagging](https://en.wikipedia.org/wiki/Tag_(metadata)) it with [keywords](https://en.wikipedia.org/wiki/Index_term), descriptions, or assigning it to a category
3. Give [metrics](https://en.wikipedia.org/wiki/Software_metric) on [data quality](https://en.wikipedia.org/wiki/Data_quality) including whether the data conforms to particular standards or patterns
4. Assess the risk involved in [integrating data](https://en.wikipedia.org/wiki/Data_integration) for new applications, including the challenges of [joins](https://en.wikipedia.org/wiki/Join)
5. Assess whether [metadata](https://en.wikipedia.org/wiki/Metadata) accurately describes the actual values in the source database
6. Understanding data challenges early in any data intensive project, so that late project surprises are avoided. Finding data problems late in the project can lead to delays and cost overruns.
7. Have an enterprise view of all data, for uses such as [master data management](https://en.wikipedia.org/wiki/Master_data_management) where key data is needed, or [data governance](https://en.wikipedia.org/wiki/Data_governance) for improving data quality.

What is pre-staging and staging?

What are the ETL options?

Checksum

Creates hash values for each row. Two tables are created, Tracking Table & Data Table

The Data table will hold all the info from the tables

The Tracking table will hold the Hash values and Key values to identify the rows. The server will check whether a hash value of a record is already in the data warehouse to determine whether this a new record

Change Data Capture (CDC)

Tracks actually DML activities and events that happen in specified tables

Time Stamps

Binary numbers used to indicate the relative sequence of events in which data modification has taken place

Join & Trigger

Explain Dimension and Facts

A fact table captures the data that measures the organization's business operations. For example, a fact table might contain business sales events such as cash register transactions or the contributions and expenditures of a nonprofit organization. Fact tables usually contain large numbers of rows, sometimes in the hundreds of millions of records when they contain one or more years of history for a large organization. A key characteristic of a fact table is that it contains numerical data (facts) that can be summarized to provide information about the history of the operation of the organization.

Dimension tables contain attributes that describe fact records in the fact table. A dimension table normally holds descriptive and textual data. For example, a dimension containing product information would often contain a hierarchy that separates products into categories such as food, drink, and etc. It is strongly recommended that surrogate keys be created and used for primary keys for all dimension tables.

Explain different type of dimensions

1. Conformed Dimension: A conformed dimension is a dimension that has **exactly the same meaning and content** when being referred from different fact tables. For example, **Time** is a common conformed dimension because its attributes (day, week, month, quarter, year, etc.) have the same meaning when joined to any fact table.

For two dimension tables to be considered as conformed, they must either be identical or one must be a subset of another. For example, we have two customer tables that are exactly the same. One for us customer, another one for Canada customer, since the structure is identical, they are consider to be conform.

1. Role-Playing Dimension

A table with multiple valid relationships between itself and another table is known as a role-playing dimension. This is most commonly seen in dimensions such as Time and Customer. For example, the Sales fact has multiple relationships to the Time query subject on the keys Order Day, Ship Day, and Close Day.

1. JUNK Dimension

In data warehouse design, frequently we run into a situation where there are yes/no indicator fields in the source system. Instead of building a separate dimension for each of these individual attributes, another option is to combine them and build what’s known as a Junk Dimension based on the Cartesian product of each of these attributes and they’re corresponding range of values.

1. Degenerate Dimension

Sometimes a dimension is deﬁned that has no content except for its primary key. For example, when an invoice has multiple line items, the line item fact rows inherit all the descriptive dimension foreign keys of the invoice, and the invoice is left with no unique content. But the invoice number remains a valid dimension key for fact tables at the line item level. This degenerate dimension is placed in the fact table with the explicit acknowledgment that there is no associated dimension table. Degenerate dimensions are most common with transaction and accumulating snapshot fact tables.

1. Rapidly Changing Dimension

A Rapidly Changing dimension is a dimensions whose data values constantly change. We only record the most updated values for this attribute. For example, we record the stock price per minuets. The updated stock price will change rapidly change as time go on. Thus this is a rapid changing dimension. It will not apply to all columns.

1. A rapidly growing dimension means rows are inserted on a constantly high rate, causing the table to expand very fast. Different from rapidly changing dimension, which stores only the most updated value, the rapidly growing dimension keep recording both the historical records and current record. For example, consider a dimension that stores historical price on an hourly basis, each hour there will be new record loaded into the dimension, old records will be kept still.

Slow changing dimension

Type 0: no change

Type1: this type of data can change, but only the most recent value is recorded.

Type2: the date will change, in this case, all changes and modification are recorded.

Type 3: for type 3 data, we tracks changes using separate columns and preserve limited history. There will be two columns to indicate the particular attribute of interest, one indicating the original value, and one indicating the current value. There will also be a column that indicates when the current value become active

Type 4. To record change in type 4 date, we use two table, one being history table and one being recent table. The recent table will record the most updated values while the historical table keep track of all the historical data.

Type 6. Type 6 is a combination of feature from type 1, 2 and 3.

Addictive, Semi-Addictive, Non-addictive

**What are Additive Measures? Give 3 examples**

Values or measurements in a fact table that can be added together, no matter what dimension

For example, sales, cost, profit in absolute value

**What are Semi-Additive Measures? Give 3 Examples**

Values that can be added up to a certain point with some dimensions, but not all

For example, stock left in warehouse, balance in one’s account,

**What are Non-Additive Measures? Give 3 Examples?**

Values that can NOT be added across any dimensions

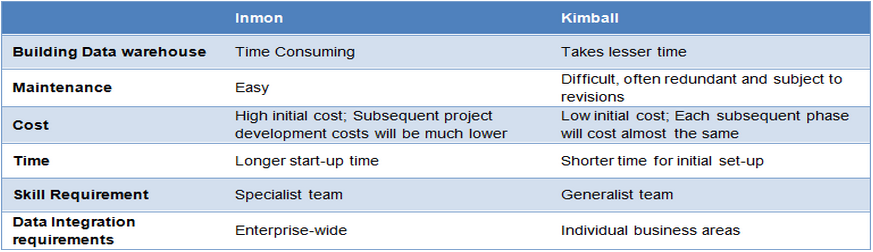
For example, profit in percentage, ratio and date

Top to down, Inmon approach

A normalized data model is designed first. Then the dimensional data marts, which contains data required for specific business processes or specific departments are created from the central data warehouse.

Down to top, Kimball approach

Ralph Kimball’s dimensional design approach (the bottom-up design): The data marts facilitating reports and analysis are created first; these are then combined together to create a broad data warehouse. This model is deformalized



The advantage of Kimball is that it takes less time to implement, can generate report in a short time. The design is de-normalized, which enables faster performance for select statement. But due to de-normalization, it stores redundancy data. On the other hand inmon is more difficult to implement, it requires significant upfront investment and it’s time consuming. The queries can be slow for large amount of data. Compare to Kimball, it’s easier to maintain and the maintain cost is lower.

SSIS design ETL application