* **What is the difference between SSAS 2005 and SSAS2008?**

1. In 2005 its not possible to create an empty cube but in 2008 we can create an empty cube.
2. A new feature in Analysis Services 2008 is the Attribute Relationships tab in the Dimension Designer . to implement attribute relationship is complex in ssas 2005
3. we can create ONLY 2000 partitions per Measure Group in ssas 2005 and the same limit of partitions is removed in ssas 2008.

You can answer more but if you end this with these then the interviewer feel that you are REAL EXPERIENCED.

* **What is datawarehouse in short DWH?**

The datawarehouse is an informational environment that

* Provides an integrated and total view of the enterprise
* Makes the enterprise’s current and historical information easily available for decision making
* Makes decision-support transactions possible without hindering operational systems
* Renders the organization’s information consistent
* Presents a flexible and interactive source of strategic information

OR a warehouse is a

* Subject oriented
* Integrated
* Time variant
* Non volatile for doing decision support

OR

Collection of data in support of management’s decision making process”. He defined the terms in the sentence as follows.

OR

Subject oriented:

It define the specific business domain ex: banking, retail, insurance, etc…..

Integrated:

It should be in a position to integrated data from various source systems

Ex: sql,oracle,db2 etc……

Time variant:

It should be in a position to maintain the data the various time periods.

Non volatile:

Once data is inserted it can’t be changed

* **What is data mart?**

A data mart is a subset of an organizational data store, usually oriented to a specific purpose or major data subject that may be distributed to support business needs. Data marts are analytical data stores designed to focus on specific business functions for a specific community within an organization.

Data marts are often derived from subsets of data in a [data warehouse](http://en.wikipedia.org/wiki/Data_warehouse), though in the bottom-up data warehouse design methodology the data warehouse is created from the union of organizational data marts.

They are 3 types of data mart they are

1. Dependent
2. Independent
3. Logical data mart

* **What are the difference between data mart and data warehouse?**

Datawarehouse is complete data where as Data mart is Subset of the same.

Ex:

All the organisation data may related to finance department, HR, banking dept are stored in data warehouse where as in data mart only finance data or HR department data will be stored. So data warehouse is a collection of different data marts.

* **Have you ever worked on performance tuning, if yes what are the steps involved in it?**

We need to identify the bottlenecks to tune the performance, to overcome the bottleneck we need to following the following.

1. Avoid named queries
2. Unnecessary relationships between tables
3. Proper attribute relationships to be given
4. Proper aggregation design
5. Proper partitioning of data
6. Proper dimension usage design
7. Avoid unnecessary many to many relationships
8. Avoid unnecessary measures
9. Set AttributeHierarchyEnabled = FALSE to Attributes that is not required
10. Won’t take even single measure which is not necessary.

* **What are the difficulties faced in cube development?**

This question is either to test whether you are really experienced or when he doesnot have any questions to ask .. ;)

You can tell any area where you feel difficult to work. But always the best answers will be the following.

1. Giving attribute relationships
2. Calculations
3. Giving dimension usage (many to many relationship)
4. Analyzing the requirements

* **Explain the flow of creating a cube?**

Steps to create a cube in ssas

1. Create  a data source.
2. Create a datasource view.
3. Create Dimensions
4. Create a cube.
5. Deploy and Process the cube.

* **What is a datasource or DS?**

The data source is the Physical Connection information that analysis service uses to connect to the database that host the data. The data source contains the connection string which specifies the server and the database hosting the data as well as any necessary authentication credentials.

* **What is datasourceview or DSV?**

A data source view is a persistent set of tables from a data source that supply the data for a particular cube. BIDS also includes a wizard for creating data source views, which you can invoke by right-clicking on the Data Source Views folder in Solution Explorer.

1. Datasource view is the logical view of the data in the data source.
2. Data source view  is the only thing a cube can see.

* **What is named calculation?**

A named calculation is a SQL expression represented as a calculated column. This expression appears and behaves as a column in the table. A named calculation lets you extend the relational schema of existing tables or views in a data source view without modifying the tables or views in the underlying data source.

Named calculation is used to create a new column in the DSV using hard coded values or by using existing columns or even with both.

* **What is named query?**

Named query in DSV is similar to View in Database. This is used to create Virtual table in DSV which will not impact the underlying database. Named query is mainly used to merge the two or more table in the datasource view or to filter columns of a table.

* **Why we need named queries?**

A named query is used to join multiple tables, to remove unnecessary columns from a table of a database. You can achieve the same in database using Views but this Named Queries will be the best bet whe you don’t have access to create Views in database.

* **How will you add a new column to an existing table in data source view?**

By using named calculations we can add a new column to an existing table in the data source view. Named Calculation is explained above.

* **What is dimension table?**

A dimension table contains hierarchical data by which you’d like to summarize. A dimension table contains specific business information, a dimension table that contains the specific name of each member of the dimension. The name of the dimension member is called an “attribute”

The key attribute in the dimension must contain a unique value for each member of the dimension. This key attribute is called “primary key column”

The primary key column of each dimension table corresponding to the one of the key column  in any related fact table.

* **What is fact table?**

A fact table contains the basic information that you wish to summarize. The table that stores the detailed value for measure is called fact table. In simple and best we can define as “The table which contains METRICS” that are used to analyse the business.

It consists of 2 sections

1) Foregine key to the dimesion

2) measures/facts(a numerical value that used to monitor business activity)

* **What is Factless fact table?**

This is very important interview question. The “Factless Fact Table” is a table which is similar to Fact Table except for having any measure; I mean that this table just has the links to the dimensions. These tables enable you to track events; indeed they are for recording events.

Factless fact tables are used for tracking a process or collecting stats. They are called so because, the fact table does not have aggregatable numeric values or information. They are mere key values with reference to the dimensions from which the stats can be collected

* **What is attribute relationships, why we need it?**

Attribute relationships are the way of telling the analysis service engine that how the attributes are related with each other. It will help to relate two or more  attributes to each other.Processing time will be decreased if proper relationships are given. This increases the Cube Processing performance and MDX query performance too.

In Microsoft SQL Server Analysis Services, attributes within a dimension are always related either directly or indirectly to the key attribute. When you define a dimension based on a star schema, which is where all dimension attributes are derived from the same relational table, an attribute relationship is automatically defined between the key attribute and each non-key attribute of the dimension. When you define a dimension based on a snowflake schema, which is where dimension attributes are derived from multiple related tables, an attribute relationship is automatically defined as follows:

* Between the key attribute and each non-key attribute bound to columns in the main dimension table.
* Between the key attribute and the attribute bound to the foreign key in the secondary table that links the underlying dimension tables.
* Between the attribute bound to foreign key in the secondary table and each non-key attribute bound to columns from the secondary table.
* **How many types of attribute relationships are there?**

They are 2 types of attribute relationships they are

1. Rigid
2. Flexible

**Rigid:**In Rigid relationships  where the relationship between the attributes is fixed, attributes will not change levels or their respective attribute relationships.

Example: The time dimension. We know that month “January 2009″ will ONLY belong to Year “2009″ and it wont be moved to any other year.

**Flexible :**In Flexible relationship between the attributes is changed.

Example: An employee and department. An employee can be in accounts department today but it is possible that the employee will be in Marketing department tomorrow.

* **How many types of dimensions are there and what are they?**

They are 3 types of dimensions:

1. confirm dimension
2. junk dimension
3. degenerate attribute

* **What are confirmed dimensions, junk dimension and degenerated dimensions?**

Confirm dimension: It is the dimension which is sharable across the multiple facts or data model. This is also called as Role Playing Dimensions.

junk dimension: A number of very small dimensions might be lumped (a small irregularly shaped) together to form a single dimension, a junk dimension – the attributes are not closely related. Grouping of Random flags and text Attributes in a dimension and moving them to a separate sub dimension is known as junk dimension.

Degenerated dimension: In this degenerate dimension contains their values in fact table and the dimension id not available in dimension table. Degenerated Dimension is a dimension key without corresponding dimension.

Example: In the PointOfSale Transaction Fact table, we have:

Date Key (FK), Product Key (FK), Store Key (FK), Promotion Key (FP), and POS Transaction Number

Date Dimension corresponds to Date Key, Production Dimension corresponds to Production Key. In a traditional parent-child database, POS Transactional Number would be the key to the transaction header record that contains all the info valid for the transaction as a whole, such as the transaction date and store identifier. But in this dimensional model, we have already extracted this info into other dimension. Therefore, POS Transaction Number looks like a dimension key in the fact table but does not have the corresponding dimension table.

* **What are the types of database schema?**

They are 3 types of database schema they are

1. Star
2. Snowflake
3. Starflake

* **What is star, snowflake and star flake schema?**

**Star schema:** In star schema fact table will be directly linked with all dimension tables. The star schema’s dimensions are denormalized with each dimension being represented by a single table. In a star schema a central fact table connects a number of individual dimension tables.

**Snowflake:**The snowflake schema is an extension of the [star schema](http://www.1keydata.com/datawarehousing/www.1keydata.com/datawarehousing/star-schema.html), where each point of the star explodes into more points. In a star schema, each dimension is represented by a single dimensional table, whereas in a snowflake schema, that dimensional table is normalized into multiple lookup tables, each representing a level in the dimensional hierarchy. In snow flake schema fact table will be linked directly as well as there will be some intermediate dimension tables between fact and dimension tables.

**Star flake:** A hybrid structure that contains a mixture of star(denormalized) and snowflake(normalized) schema’s.

* **How will you hide an attribute?**

We can hide the attribute by selecting “AttributeHierarchyVisible = False” in properties of the attribute.

* **How will you make an attribute not process?**

By selecting  “ AttributeHierarchyEnabled = False”, we can make an  attribute not in process.

* **What is use of IsAggregatable property?**

In Analysis Service we generally see all dimension has All member. This is because of IsAggregatable property of the attribute. You can set its value to false, so that it will not show All member. Its default member for that attribute. If you hide this member than you will have to set other attribute value to default member else it will pick some value as default and this will create confusion in browsing data if someone is not known to change in default member.

* **What are key, name and value columns of an attribute?**

**Key column of any attribute:**Contains the column or columns that represent the key for the attribute, which is the column in the underlying relational table in the data source view to which the attribute is bound. The value of this column for each member is displayed to users unless a value is specified for the NameColumn property.

**Name  column of an attribute:** Identifies the column that provides the name of the attribute that is displayed to users, instead of the value in the key column for the attribute. This column is used when the key column value for an attribute member is cryptic or not otherwise useful to the user, or when the key column is based on a composite key. The NameColumn property is not used in parent-child hierarchies; instead, the NameColumn property for child members is used as the member names in a parent-child hierarchy.

**Value columns of an attribute:**Identifies the column that provides the value of the attribute. If the NameColumn element of the attribute is specified, the same DataItem values are used as default values for the ValueColumn element. If the NameColumn element of the attribute is not specified and the KeyColumns collection of the attribute contains a single KeyColumn element representing a key column with a string data type, the same DataItem values are used as default values for the ValueColumn element.

* **What is hierarchy, what are its types and difference between them?**

A hierarchy is a very important part of any OLAP engine and allows users to drill down from  summary levels hierarchies represent the way user expect to explore data at more detailed level

hierarchies  is made up of multipule levels creating the structure based on end user requirements.

->years->quarter->month->week ,are all the levels of calender hierarchy

They are 2 types of hierarchies they are

1. Natural hierarchy
2. Unnatural hierarchy

**Natural hierarchy:**This means that the attributes are intuitively related to one another. There is a clear relationship from the top of the hierarchy to the bottom.

Example: An example of this would be date: year, quarter and month follow from each other, and in part, define each other.

**Unnatural hierarchy:**This means that the attributes are not clearly related.

Example: An example of this might be geography; we may have country -> state -> city, but it is not clear where Province might sit.

* **What is Attribute hierarchy?**

An attribute hierarchy is created for every attribute in a dimension, and each hierarchy is available for dimensioning fact data. This hierarchy consists of an “All” level and a detail level containing all members of the hierarchy.

you can organize attributes into user-defined hierarchies to provide navigation paths in a cube. Under certain circumstances, you may want to disable or hide some attributes and their hierarchies.

* **What is use of AttributeHierarchyDisplayFolder property ?**

**AttributeHierarchyDisplayFolder:** Identifies the folder in which to display the associated attribute hierarchy to end users. For example if I set the property value as “Test” to all the Attributes of a dimension then a folder with the name “Test” will be created and all the Attributes will be placed into the same.

* **What is use of AttributeHierarchyEnabled?**

**AttributeHierarchyEnabled:**Determines whether an attribute hierarchy is generated by Analysis Services for the attribute. If the attribute hierarchy is not enabled, the attribute cannot be used in a user-defined hierarchy and the attribute hierarchy cannot be referenced in Multidimensional Expressions (MDX) statements.

* **What is use of AttributeHierarchyOptimizedState?**

**AttributeHierarchyOptimizedState:**Determines the level of optimization applied to the attribute hierarchy. By default, an attribute hierarchy is FullyOptimized, which means that Analysis Services builds indexes for the attribute hierarchy to improve query performance. The other option, NotOptimized, means that no indexes are built for the attribute hierarchy. Using NotOptimized is useful if the attribute hierarchy is used for purposes other than querying, because no additional indexes are built for the attribute. Other uses for an attribute hierarchy can be helping to order another attribute.

* **What is use of AttributeHierarchyOrdered ?**

**AttributeHierarchyOrdered:**Determines whether the associated attribute hierarchy is ordered. The default value is True. However, if an attribute hierarchy will not be used for querying, you can save processing time by changing the value of this property to False.

* **What is the use of AttributeHierarchyVisible ?**

**AttributeHierarchyVisible :**Determines whether the attribute hierarchy is visible to client applications. The default value is True. However, if an attribute hierarchy will not be used for querying, you can save processing time by changing the value of this property to False.

* **What are types of storage modes?**

There are three standard storage modes in OLAP applications

1. MOLAP
2. ROLAP
3. HOLAP

* **Compare the Three Storage Modes ?**

Summary and comparison

| Basic Storage Mode | Storage Location for Detail Data | Storage Location for Summary/ Aggregations | Storage space requirement | Query Response Time | Processing Time | Latency |
| --- | --- | --- | --- | --- | --- | --- |
| MOLAP | Multidimensional Format | Multidimensional Format | MediumBecause detail data is stored in compressed format. | Fast | Fast | High |
| HOLAP | Relational Database | Multidimensional Format | Small | Medium | Fast | Medium |
| ROLAP | Relational Database | Relational Database | Large | Slow | Slow | Low |

* **What is MOLAP and its advantage?**

MOLAP (Multi dimensional Online Analytical Processing) : MOLAP is the most used storage type. Its designed to offer maximum query performance to the users. the data and aggregations are stored in a multidimensional format, compressed and optimized for performance. This is both good and bad. When a cube with MOLAP storage is processed, the data is pulled from the relational database, the aggregations are performed, and the data is stored in the AS database. The data inside the cube will refresh only when the cube is processed, so latency is high.

Advantages:

1. Since the data is stored on the OLAP server in optimized format, queries (even complex calculations) are faster than ROLAP.
2. The data is compressed so it takes up less space.
3. And because the data is stored on the OLAP server, you don’t need to keep the connection to the relational database.
4. Cube browsing is fastest using MOLAP.

* **What is ROLAP and its advantage?**

ROLAP (Relational Online Analytical Processing) : ROLAP does not have the high latency disadvantage of MOLAP. With ROLAP, the data and aggregations are stored in relational format. This means that there will be zero latency between the relational source database and the cube.

Disadvantage of this mode is the performance, this type gives the poorest query performance because no objects benefit from multi dimensional storage.

Advantages:

1. Since the data is kept in the relational database instead of on the OLAP server, you can view the data in almost real time.
2. Also, since the data is kept in the relational database, it allows for much larger amounts of data, which can mean better scalability.
3. Low latency.

* **What is HOLAP and its advantage?**

Hybrid Online Analytical Processing (HOLAP): HOLAP is a combination of MOLAP and ROLAP. HOLAP stores the detail data in the relational database but stores the aggregations in multidimensional format. Because of this, the aggregations will need to be processed when changes are occur. With HOLAP you kind of have medium query performance: not as slow as ROLAP, but not as fast as MOLAP. If, however, you were only querying aggregated data or using a cached query, query performance would be similar to MOLAP. But when you need to get that detail data, performance is closer to ROLAP.

Advantages:

1. HOLAP is best used when large amounts of aggregations are queried often with little detail data, offering high performance and lower storage requirements.
2. Cubes are smaller than MOLAP since the detail data is kept in the relational database.
3. Processing time is less than MOLAP since only aggregations are stored in multidimensional format.
4. Low latency since processing takes place when changes occur and detail data is kept in the relational database.

* **What are Translations and its use?**

**Translation:** The translation feature in analysis service allows you to display caption and attributes names that correspond to a specific language. It helps in providing GLOBALIZATION to the Cube.

* **What is Database dimension?**

All the dimensions that are created using NEW DIMENSION Wizard are database dimensions. In other words, the dimensions which are at Database level are called Database Dimensions.

* What is Cube dimension?

A cube dimension is an instance of a database dimension within a cube is called as cube dimension. A database dimension can be used in multiple cubes, and multiple cube dimensions can be based on a single database dimension

* **Difference between Database dimension and Cube dimension?**

1. The Database dimension has only Name and ID properties, whereas a Cube dimension has several more properties.
2. Database dimension is created one where as Cube dimension is referenced from database dimension.
3. Database dimension exists only once.where as Cube dimensions can be created more than one using ROLE PLAYING Dimensions concept.

* **How will you add a dimension to cube?**

To add a dimension to a cube follow these steps.

1. In Solution Explorer, right-click the cube, and then click View Designer.
2. In the Design tab for the cube, click the Dimension Usage tab.
3. Either click the Add Cube Dimension button, or right-click anywhere on the work surface  and then click Add Cube Dimension.
4. In the Add Cube Dimension dialog box, use one of the following steps:
5. To add an existing dimension, select the dimension, and then click OK.
6. To create a new dimension to add to the cube, click New dimension, and then follow the steps in the Dimension Wizard.

* **What is SCD (slowly changing dimension)?**

Slowly changing dimensions (SCD) determine how the historical changes in the dimension tables are handled. Implementing the SCD mechanism enables users to know to which category an item belonged to in any given date.

* **What are types of SCD?**

It is a concept of STORING Historical Changes and when ever an IT guy finds a new way to store then a new Type will come into picture. Basically there are 3 types of SCD they are given below

1. SCD type1
2. SCD type2
3. SCD type3

* **What  is Type1, Type2, Type3 of SCD?**

**Type 1:**In Type 1 Slowly Changing Dimension, the new information simply overwrites the original information. In other words, no history is kept.

In our example, recall we originally have the following table:

| Customer Key | Name | State |
| --- | --- | --- |
| 1001 | Christina | Illinois |

After Christina moved from Illinois to California, the new information replaces the new record, and we have the following table:

| Customer Key | Name | State |
| --- | --- | --- |
| 1001 | Christina | California |

Advantages: This is the easiest way to handle the Slowly Changing Dimension problem, since there is no need to keep track of the old information.

Disadvantages: All history is lost. By applying this methodology, it is not possible to trace back in history.

Usage:  About 50% of the time.

When to use Type 1: Type 1 slowly changing dimension should be used when it is not necessary for the data warehouse to keep track of historical changes.

**Type 2:**In Type 2 Slowly Changing Dimension, a new record is added to the table to represent the new information. Therefore, both the original and the new record will be present. The new record gets its own primary key.

In our example, recall we originally have the following table:

| Customer Key | Name | State |
| --- | --- | --- |
| 1001 | Christina | Illinois |

After Christina moved from Illinois to California, we add the new information as a new row into the table:

| Customer Key | Name | State |
| --- | --- | --- |
| 1001 | Christina | Illinois |
| 1005 | Christina | California |

Advantages: This allows us to accurately keep all historical information.

Disadvantages:

1. This will cause the size of the table to grow fast. In cases where the number of rows for the table is very high to start with, storage and performance can become a concern.
2. This necessarily complicates the ETL process.

Usage: About 50% of the time.

**Type3 :**In Type 3 Slowly Changing Dimension, 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 becomes active.

In our example, recall we originally have the following table:

| Customer Key | Name | State |
| --- | --- | --- |
| 1001 | Christina | Illinois |

To accommodate Type 3 Slowly Changing Dimension, we will now have the following columns:

Customer Key,Name,OriginalState,CurrentState,Effective Date

After Christina moved from Illinois to California, the original information gets updated, and we have the following table (assuming the effective date of change is January 15, 2003):

| Customer Key | Name | OriginalState | CurrentState | Effective Date |
| --- | --- | --- | --- | --- |
| 1001 | Christina | Illinois | California | 15-JAN-2003 |

Advantages:

1. This does not increase the size of the table, since new information is updated.
2. This allows us to keep some part of history.

Disadvantages: Type 3 will not be able to keep all history where an attribute is changed more than once. For example, if Christina later moves to Texas on December 15, 2003, the California information will be lost.

Usage: Type 3 is rarely used in actual practice.

* **What is role playing dimension with two examples?**

**Role play dimensions:** We already discussed about this. This is nothing but CONFIRMED Dimensions. A dimension can play different role in a fact table you can recognize a roleplay dimension when there are multiple columns in a fact table that each have foreign keys to the same dimension table.

Ex1: There are three dimension keys in the factinternalsales,factresellersales tables which all refer to the dimtime table,the same time dimension is used  to track sales by that contain either of these fact table,the corresponding  role-playing dimension are automatically added to the cube.

Ex2 : In retail banking, for checking account cube we could have transaction date dimension and effective date dimension. Both dimensions have date, month, quarter and year attributes. The formats of attributes are the same on both dimensions, for example the date attribute is in ‘dd-mm-yyyy’ format. Both dimensions have members from 1993 to 2010.

* **What is measure group, measure?**

**Measure groups :**These measure groups can contain different dimensions and be at different granularity  but so long as you model your cube correctly, your users will be able to use measures from each of these measure groups in their queries easily and without worrying about the underlying complexity.

Creating multiple measure groups : To create a new measure group in the Cube Editor, go to the Cube Structure tab and right-click on the cube name in the Measures pane and select ‘New Measure Group’. You’ll then need to select the fact table to create the measure group from and then the new measure group will be created; any columns that aren’t used as foreign key columns in the DSV will automatically be created as measures, and you’ll also get an extra measure of aggregation type Count. It’s a good idea to delete any measures you are not going to use at this stage.

**Measures :**  Measures are the numeric values that our users want to aggregate, slice, dice and otherwise analyze, and as a result, it’s important to make sure they behave the way we want them to. One of the fundamental reasons for using Analysis Services is that, unlike a relational database it allows us to build into our cube design business rules about measures: how they should be formatted, how they should aggregate up, how they interact with specific dimensions and so on.

* **What is attribute?**

An attribute is a specification that defines a property of an object, element, or file. It may also refer to or set the specific value for a given instance of such.

* **What is surrogate key?**

A surrogate key is the SQL generated key which acts like an alternate primary key for the table in database, Data warehouses commonly use a surrogate key to uniquely identify an entity. A surrogate is not generated by the user but by the system. A primary difference between a primary key and surrogate key in few databases is that primarykey uniquely identifies a record while a Surrogatekey uniquely identifies an entity.

Ex: An employee may be recruited before the year 2000 while another employee with the same name may be recruited after the year 2000. Here, the primary key will uniquely identify the record while the surrogate key will be generated by the system (say a serial number) since the SK is NOT derived from the data.

* **How many types of relations are there between dimension and measure group?**

They are six relation between the dimension and measure group, they are

1. No Relationship
2. Regular
3. Refernce
4. Many to Many
5. Data Mining
6. Fact

* **What is regular type, no relation type, fact type, referenced type, many-to-many type with example?**

**No relationship:** The dimension and measure group are not related.

**Regular:** The dimension table is joined directly to the fact table.

**Referenced:** The dimension table is joined to an intermediate table, which in turn,is joined to the fact table.

**Many to many:**The dimension table is to an intermediate fact table,the intermediate fact table is joined , in turn, to an intermediate dimension table to which the fact table is joined.

**Data mining:**The target dimension is based on a mining model built from the source dimension. The source dimension must also be included in the cube.

**Fact table:**The dimension table is the fact table.

* **What are calculated members and what is its use?**

Calculations are item in the cube that are eveluated at runtime

Calculated members: You can create customized measures or dimension members, called calculated members, by combining cube data, arithmetic operators, numbers, and/or functions.

Example: You can create a calculated member called Marks that converts dollars to marks by multiplying an existing dollar measure by a conversion rate. Marks can then be displayed to end users in a separate row or column. Calculated member definitions are stored, but their values exist only in memory. In the preceding example, values in marks are displayed to end users but are not stored as cube data.

* **What are KPIs and what is its use?**

In Analysis Services, a KPI is a collection of calculations that are associated with a measure group in a cube that are used to evaluate business success. We use KPI to see the business at the particular point, this is represents with some graphical items such as traffic signals,ganze etc

* **What are actions, how many types of actions are there, explain with example?**

Actions are powerful way of extending the value of SSAS cubes for the end user. They can click on a cube or portion of a cube to start an application with the selected item as a parameter, or to retrieve information about the selected item.

One of the objects supported by a SQL Server Analysis Services cube is the action. An action is an event that a user can initiate when accessing cube data. The event can take a number of forms. For example, a user might be able to view a Reporting Services report, open a Web page, or drill through to detailed information related to the cube data

Analysis Services supports three types of actions..

**Report action:**Report action Returns a Reporting Services report that is associated with the cube data on which the action is based.

**Drill through:**Drillthrough Returns a result set that provides detailed information related to the cube data on which the action is based.

**Standard:**Standard has five action subtypes that are based on the specified cube data.

**Dataset:** Returns a mutlidimensional dataset.

**Proprietary:** Returns a string that can be interpreted by a client application.

**Rowset:** Returns a tabular rowset.

**Statement:** Returns a command string that can be run by a client application.

**URL:**  Returns a URL that can be opened by a client application, usually a browser.

* **What is partition, how will you implement it?**

You can use the Partition Wizard to define partitions for a measure group in a cube. By default, a single partition is defined for each measure group in a cube. Access and processing performance, however, can degrade for large partitions. By creating multiple partitions, each containing a portion of the data for a measure group, you can improve the access and processing performance for that measure group.

* **What is the minimum and maximum number of partitions required for a measure group?**

In 2005 a MAX of 2000 partitions can be created per measure group and that limit is lifted in later versions.

In any version the MINIMUM is ONE Partition per measure group.

* **What are Aggregations and its use?**

Aggregations provide performance improvements by allowing Microsoft SQL Server Analysis Services (SSAS) to retrieve pre-calculated totals directly from cube storage instead of having to recalculate data from an underlying data source for each query. To design these aggregations, you can use the Aggregation Design Wizard. This wizard guides you through the following steps:

1. Selecting standard or custom settings for the storage and caching options of a partition, measure group, or cube.
2. Providing estimated or actual counts for objects referenced by the partition, measure group, or cube.
3. Specifying aggregation options and limits to optimize the storage and query performance delivered by designed aggregations.
4. Saving and optionally processing the partition, measure group, or cube to generate the defined aggregations.
5. After you use the Aggregation Design Wizard, you can use the Usage-Based Optimization Wizard to design aggregations based on the usage patterns of the business users and client applications that query the cube.

* **What is perspective, have you ever created perspective?**

Perspectives are a way to reduce the complexity of cubes by hidden elements like measure groups, measures, dimensions, hierarchies etc. It’s nothing but slicing of a cube, for ex we are having retail and hospital data and end user is subscribed to see only hospital data, then we can create perspective according to it.

* **What is deploy, process and build?**

**Bulid:** Verifies the project files and create several local files.

**Deploy:** Deploy the structure of the cube(Skeleton) to the server.

**Process:** Read the data from the source and build the dimesions and cube structures

Elaborating the same is given below.

**Build:**Its is a used to process the data of the cube database. Build is a version of a [program](http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci212834,00.html). As a rule, a build is a pre-release version and as such is identified by a build number, rather than by a release number. Reiterative (repeated) builds are an important part of the development process. Throughout development, application components are collected and repeatedly [compiled](http://whatis.techtarget.com/definition/0,,sid9_gci211824,00.html) for testing purposes, to ensure a reliable final product. Build tools, such as [make](http://whatis.techtarget.com/definition/0,,sid9_gci873189,00.html) or [Ant](http://searchenterpriselinux.techtarget.com/sDefinition/0,,sid39_gci873167,00.html), enable developers to automate some programming tasks. As a verb, to build can mean either to write [code](http://whatis.techtarget.com/definition/0,,sid9_gci213934,00.html) or to put individual coded components of a program together.

**Deployment:**During development of an Analysis Services project in Business Intelligence Development Studio, you frequently deploy the project to a development server in order to create the Analysis Services database defined by the project. This is required to test the project.

for example, to browse cells in the cube, browse dimension members, or verify key performance indicators (KPIs) formulas.

* **What is the maximum size of a dimension?**

The maximum size of the dimension is **4 gb**.

* **What are the types of processing and explain each?**

They are 6 types of processing in ssas ,they are

* Process Full
* Process Data
* Process Index
* Process Incremental
* Process Structure
* UnProcess

**Process Full:**Processes an Analysis Services object and all the objects that it contains. When Process Full is executed against an object that has already been processed, Analysis Services drops all data in the object, and then processes the object. This kind of processing is required when a structural change has been made to an object, for example, when an attribute hierarchy is added, deleted, or renamed. This processing option is supported for cubes, databases, dimensions, measure groups, mining models, mining structures, and partitions.

**Process Data:**Processes data only without building aggregations or indexes. If there is data is in the partitions, it will be dropped before re-populating the partition with source data. This processing option is supported for dimensions, cubes, measure groups, and partitions.

**Process Index:**Creates or rebuilds indexes and aggregations for all processed partitions. This option causes an error on unprocessed objects. This processing option is supported for cubes, dimensions, measure groups, and partitions.

**Process Increment: A**dds newly available fact data and process only to the relevant partitions. This processing option is supported for measure groups, and partitions.

**Process Structure:**If the cube is unprocessed, Analysis Services will process, if it is necessary, all the cube’s dimensions. After that, Analysis Services will create only cube definitions. If this option is applied to a mining structure, it populates the mining structure with source data. The difference between this option and the Process Full option is that this option does not iterate the processing down to the mining models themselves. This processing option is supported for cubes and mining structures.

**Unprocess :**Drops the data in the object specified and any lower-level constituent objects. After the data is dropped, it is not reloaded. This processing option is supported for cubes, databases, dimensions, measure groups, mining models, mining structures, and partitions.

**Process Default:**Detects the process state of an object, and performs processing necessary to deliver unprocessed or partially processed objects to a fully processed state. This processing option is supported for cubes, databases, dimensions, measure groups, mining models, mining structures, and partitions.

* **What is a cube?**

The basic unit of storage and analysis in Analysis Services is the *cube*. A cube is a collection of data that’s been aggregated to allow queries to return data quickly.

For example, a cube of order data might be aggregated by time period and by title, making the cube fast when you ask questions concerning orders by week or orders by title.

* **What is AMO?**

The full form of AMO is Analysis Managament Objects. This is used to create or alter cubes from .NET code.

* **After creating the cube, if  we added a new column to the OLTP table then how you add this new attribute to the cube?**

Just open the datasourceview and on right click we find the option REFRESH. Click the REFRESH then it will add new attributes to the table which can be added to Cube.

**REAL TIME INTERVIEW QUESTIONS -**

* **What is the size of the Cube in your last Project?**

Answer to this question varies from project to project and mainly depends on how BIG is your database and how COMPLEX the database design is. Generally for the database with a TRANSACTION TABLE of 50 crore records, the cube size will be around 100GB. So, better go with 100GB as answer to this question.

* **What is size of the database in your last Project?**

You can expect this question immediately after you answer 100GB to the last question. The database size will be 600 to 800GB for which the cube will come to 100 GB. So go with 800GB for this question.

* **What is size of the fact(Transaction) table in your last Project?**

This will be the next question if you answer 800GB as your dataabase size. Here he is not expecting SIZE in GBs but the interviewer will be expecting NUMBER OF ROWS in the Transaction table. Go with 57Crore records for this question.

* **How frequently you process the cube?**

You have to be very careful here. Frequency of processing cube depends on HOW FREQUENTLY YOU ARE GETTING NEW DATA. Once the new data comes then SSIS team loads it and send a mail to SSAS team after load is completed successfully. Once SSAS team receives the mail then these guys will look for best time to PROCESS.

Typically we get data either Weekly or Monthly. So you can say that the processing of the cube will be done either Weekly or monthly.

* **How frequently you get DATA from clients?**

This answer should be based on your last answer. IF you answered WEEKLY to last question then the Answer to this question also should be WEEKLY. IF MONTHLY for last question then this answer also should be MONTHLY.

* **What type of Processing Options you used to process the cube in your Project?**

This is the toughest question to answer. This depends on DATA you have and CLIENTS requirements. Let me explain here.

1. If the database is SMALL, let’s say it has only 1 crore records then people do FULL PROCESS as it wont take much time.
2. If the database is MEDIUM, let’s say it has only 15 crore records then people prefer to do INCREMENTAL PROCESS unless CLIENTS ask us to do FULL PROCESS as it takes little bit of time.
3. If the database is HUGE, let’s say it has more than 35 to 40 crore records then people prefer to do INCREMENTAL PROCESS unless CLIENTS ask us to do FULL PROCESS as it takes lot of time. In this case we TRY to convince clients for INCREMENTAL and if they don’t agree then we don’t have any other option.
4. Incremental process will come into picture ONLY when there is no updates to the OLD data i.e no changes to already existing data else NO OTHER OPTION than FULL PROCESS.

* **How you provide security to cube?**

**By defining roles we provide security to cubes. Using roles we can restrict users from accessing restricted data. Procedure as follows -**

1. Define Role
2. Set Permission
3. Add appropriate Users to the role

* **How you move the cube from one server to another?**

There are many ways to do the same. Let me explain four here and cleverly you can say “I worked on 4 SSAS projects till date and implemented different types in all the four.”

1. Backup and restore – This is the simplest way. Take the Backup from development server and copy the backup to FTP folder of clients. After doing this drop a mail to Client’s Admin and he will take care of RESTORE part.
2. Directly PROCESS the cube in PRODUCTION environment. For this you need access to Production which will not be given by clients unless the clients are \*\*\*\*\*\*\*\*\*. One of the client I worked for given FULL access to me .. ;)
3. Under Srart –> All Programs –> Sql Server –> Analysis Services you can see deployment wizard. This is one way of moving the cube. This method has some steps to follow. First deploy your cube and FOUR files will be created in BIN folder of PROJECT folder. Copy those FOUR files and paste in Production server in any directory. Then OPEN this DEPLOYMENT Wizard in production and when it ask for Database file then point to the location where you copied the files. After that NEXT,NEXT,NEXT … OK .. Cube will be deployed and processed.
4. This way is most beautiful one. Synchronization, In this we will first deploy and process the cube in STAGING ENVIRONMENT and then we will go to production server. Connect to Analysis services in SSMS and select Synchronize by right clicking on Databases folder in SSMS of analysis services. Then select source as STAGING SERVER and then click on OK. The changes in the cube present in the Staging server will be copied to the production server.

* **What is the toughest challenge you face in your Project?**

There are couple of this where we face difficulty.

1. While working on RELATIONSHIPS between Measure Groups and Dimensions.
2. Working on Complex calculations
3. Performance tuning

* **How you created Partitions of the cube in your Last Project?**

Partitions can be created on different data. Few people do it on PRODUCT NAME wise and many prefer to do it on DATE data wise.  you go with DATE wise.

In dates, we can create MONTH wise,WEEK wise,QUARTER wise and some times YEAR wise. This all depends on how much data you are coming per WEEK or MONTH or QUARTER … If you are getting 50 lakhs records per month then tell you do MONTH wise.

* **How many dimensions in your last cube?**

47 to 50.

* **How many measure groups in your last cube?**

Total 10 and in that 4 are Fact tables and remaining 6 are Fact less fact tables.

* **What is the Schema of your last cube?**

Snowflake

* **Why not STAR Schema ?**

My data base design doesn’t support STAR Schema.

* **What are the different relationships that you are used in your cube?**

1. Regular
2. Referenced
3. Many to Many
4. Fact
5. No Relationship

* **Have you created the KPI’s , If then Explain?**

Don’t add much to this as the questions in this will be tricky. Just tell that you worked on couple of KPI and you have basic knowledge on this. (Don’t worry, this is not MANDATORY)

* **How you define Aggregations in your Project?**

We defined the aggregations for MOST FREQUENTLY USED data in SSRS reports.

* **Size of SSAS team in your last Project?**

Just 2 guys as we guys are really in demand and lot of scarcity:)

* **How many Resources worked on same Cube in your Project?**

Only 2 and one in morning shift and another in Evening shift.

* **How much time it take to Process the Cube?**

This is Very very important question. This again depends on the SIZE of database,Complexity of the database and your server settings. For database with 50 cr transaction records, it generally takes 3.5 hrs.

* **How many Calculation you done in Your Project?**

I answer more than 5000 and if you tell the same then you are caught unless you are super good in MDX. Best answer for you is “Worked on 50 calculations”.