

## **Informatica Question & Answers**

**Q1) Tell me what exactly, what was your role?**

**A1) I worked as ETL Developer. I was also involved in requirement gathering, developing mappings, checking source data. I did Unit testing (using TOAD), helped in User Acceptance Testing.**

**Q2) What kind of challenges did you come across in your project?**

**A2) Mostly the challenges were to finalize the requirements in such a way**

**So that different stakeholders come to a common agreement about the scope and expectations from the project.**

**Q3) Tell me what the size of your database was?**

**A3) Around 3 TB. There were other separate systems, but the one I was mainly using was around 3 TB.**

**Q4) what was the daily volume of records?**

**A4) It used to vary, We processed around 100K-200K records on a daily basis, on weekends, and it used to be higher sometimes around 1+ Million records.**

**Q5) So tell me what your sources were?**

**A5) Our Sources were mainly flat files, relational databases.**

**Q6) What tools did you use for FTP/UNIX?**

**A6) For UNIX, I used Open Source tool called Putty and for FTP, I used WINSXP, Filezilla.**

**Q7) Tell me how did you gather requirements?**

**A7) We used to have meetings and design sessions with end users. The users used to give us sketchy requirements and after that we used to do further analysis and used to create detailed Requirement Specification Documents (RSD).**

**Q8) Did you follow any formal process or methodology for Requirement gathering?**

**A8) As such we did not follow strict SDLC approach because requirement gathering is an iterative process. But after creating the detailed Requirement Specification Documents, we used to take User signoff.**

**Q9) How did you do Error handling in Informatica?**

**A9) Typically we can set the error flag in mapping based on business requirements and for each type of error, we can associate an error code and error description and write all errors to a separate error table so that we capture all rejects correctly.**

**Also we need to capture all source fields in a ERR\_DATA table so that if we need to correct the erroneous data fields and Re-RUN the corrected data if needed.**

**Usually there could be a separate mapping to handle such error data file.**

**Typical errors that we come across are**

**1) Non Numeric data in Numeric fields.**

**2) Incorrect Year / Months in Date fields from Flat files or varchar2 fields.**

**Q10) Did you work in Team Based environments?**

**A10) Yes, we had versioning enabled in Repository.**

**Q11) Tell me what are the steps involved in Application Development?**

**A11) In Application Development, we usually follow following steps:  
ADDTIP.**

**a) A - Analysis or User Requirement Gathering**

**b) D - Designing and Architecture**

- c) D - Development
- d) T - Testing (which involves Unit Testing, System Integration Testing, UAT - User Acceptance Testing )
- e) I - Implementation (also called deployment to production)
- f) P - Production Support / Warranty

**Q12) What are the drawbacks of Waterfall Approach ?**

**A12) This approach assumes that all the User Requirements will be perfect before start of design and development. That is not the case most of the time. Users can change their mind to add few more detailed requirements or worse change the requirements drastically. So in those cases this approach (waterfall) is likely to cause a delay in project which is a RISK to the project.**

**Q13) what is mapping design document?**

**A13) In a mapping design document, we map source to target field, also document any special business logic that needs to be implemented in the mapping.**

**Q14) What are different Data Warehousing Methodologies that you are familiar with?**

**A14) In Data Warehousing, two methodologies are popular, 1st one is Ralph Kimball and 2nd one is Bill Inmon.**

**We mainly followed Ralph Kimball's methodology in my last project.**

**In this methodology, we have a fact tables in the middle, surrounded by dimension tables.**

**This is also a basic STAR Schema which is the basic dimensional model.**

**A Snowflake schema. In a snowflake schema, we normalize one of the dimension tables.**

**Q15) What do you do in Bill Inmon Approach?**

**A15) In Bill Inmon's approach, we try to create an Enterprise Data Warehouse using 3rd NF, and then Data Marts are mainly STAR Schemas in 2nd NF.**

**Q16) How many mappings have you done?**

**A16) I did over 35+ mappings, around 10+ were complex mappings.**

**Q17) What are Test cases or how did you do testing of Informatica Mappings?**

**A17) Basically we take the SQL from Source Qualifier and check the source / target data in Toad.**

**Then we try to spot check data for various conditions according to mapping document and look for any error in mappings.**

**For example, there may be a condition that if customer account does not exist then filter out that record and write it to a reject file.**

**Q18) What are the other error handlings that you did in mappings?**

**A18) I mainly looked for non-numeric data in numeric fields, layout of a flat file may be different. Also dates from flat file come as a string**

**Q19) How did you debug your mappings?**

**A19) I used Informatica Debugger to check for any flags being set incorrectly. We see if the logic / expressions are working or not. We may be expecting data  
We use Wizard to configure the debugger.**

**Q20) Give me an example of a tough situation that you came across in Informatica Mappings and how did you handle it?**

**A20) Basically one of our colleagues had created a mapping that was using Joiner and mapping was taking a lot of time to run, but the Join was in such a way that we could do the Join at Database Level (Oracle Level). So I suggested and implemented that change and it reduced the run time by 40%.**

**Q21) Tell me what various transformations that you have used are?**

**A21) I have used Lookup, Joiner, Update Strategy, Aggregator, Sorter etc.**

**Q22) How will you categorize various types of transformation?**

**A22) Transformations can be connected or unconnected. Active or passive.**

**Q23) What are the different types of Transformations?**

**A23) Transformations can be active transformation or passive transformations. If the number of output rows is different than number of input rows then the transformation is an active transformation.**

**Like a Filter / Aggregator Transformation. Filter Transformation can filter out some records based on condition defined in filter transformation.**

**Similarly, in an aggregator transformation, number of output rows can be less than input rows as after applying the aggregate function like SUM, we could have fewer records.**

**Q24) What is a lookup transformation?**

**A24) We can use a Lookup transformation to look up data in a flat file or a relational table, view, or synonym.**

**We can use multiple Lookup transformations in a mapping.**

**The Power Center Server queries the lookup source based on the lookup ports in the transformation. It compares Lookup transformation port values to lookup source column values based on the lookup condition.**

**We can use the Lookup transformation to perform many tasks, including:**

- 1) Get a related value.**
- 2) Perform a calculation.**
- 3) Update slowly changing dimension tables.**

**Q25) Did you use unconnected Lookup Transformation? If yes, then explain.**

**A25) Yes. An Unconnected Lookup receives input value as a result of: LKP Expression in another transformation. It is not connected to any other transformation. Instead, it has input ports, output ports and a Return Port.**

**An Unconnected Lookup can have ONLY ONE Return PORT.**

**Q26) What is Lookup Cache?**

**A26) The Power Center Server builds a cache in memory when it processes the first row of data in a cached Lookup transformation.**

**It allocates the memory based on amount configured in the session. Default is 2M Bytes for Data Cache and 1M bytes for Index Cache.**

**We can change the default Cache size if needed.**

**Condition values are stored in Index Cache and output values in Data cache.**

**Q27) What happens if the Lookup table is larger than the Lookup Cache?**

**A27) If the data does not fit in the memory cache, the Power Center Server stores the overflow values in the cache files.**

**To avoid writing the overflow values to cache files, we can increase the default cache size.**

**When the session completes, the Power Center Server releases cache memory and deletes the cache files.**

**If you use a flat file lookup, the Power Center Server always caches the lookup source.**

**Q28) What is meant by "Lookup caching enabled"?**

**A28) By checking "Lookup caching enabled" option, we are instructing Informatica Server to Cache lookup values during the session.**

**Q29) What are the different types of Lookup?**

**A29) When configuring a lookup cache, you can specify any of the following options:**

**a) Persistent cache. You can save the lookup cache files and reuse them the next time the Power Center Server processes a Lookup transformation configured to use the cache.**

**b) Recache from source. If the persistent cache is not synchronized with the lookup table, you can configure**

the Lookup transformation to rebuild the lookup cache.

c) Static cache. You can configure a static, or read-only, cache for any lookup source.

By default, the Power Center Server creates a static cache. It caches the lookup file or table and looks up values in the cache for each row that comes into the transformation.

When the lookup condition is true, the Power Center Server returns a value from the lookup cache. The Power Center Server does not update the cache while it processes the Lookup Transformation.

d) Dynamic cache. If you want to cache the target table and insert new rows or update existing rows in the cache and the target, you can create a Lookup transformation to use a dynamic cache.

The Power Center Server dynamically inserts or updates data in the lookup cache and passes data to the target table.

You cannot use a dynamic cache with a flat file lookup.

e) Shared cache. You can share the lookup cache between multiple transformations. You can share an unnamed cache between transformations in the same mapping. You can share a named cache between transformations in the same or different mappings.

**Q30) What is a Router Transformation?**

**A30) A Router transformation is similar to a Filter transformation because both transformations allow you to use a condition to test data. A Filter transformation tests data for one condition and drops the rows of data that do not meet the condition.**

**However, a Router transformation tests data for one or more conditions and gives you the option to route rows of data that do not meet any of the conditions to a default output group.**

**Q31) What is a sorter transformation?**

**A31) The Sorter transformation allows you to sort data. You can sort data in ascending or descending order according to a specified sort key. You can also configure the Sorter transformation for case-sensitive sorting, and specify whether the output rows should be distinct. The Sorter transformation is an active transformation.**

**It must be connected to the data flow.**

**Q32) What is a UNION Transformation?**

**A32) The Union transformation is a multiple input group transformation that you can use to merge data from multiple pipelines or pipeline branches into one pipeline branch. It merges data from multiple sources similar to the UNION ALL SQL statement to combine the results from two or more SQL statements. Similar to the UNION ALL statement, the Union transformation does not remove duplicate rows.**

**You can connect heterogeneous sources to a Union transformation. The Union transformation merges sources with matching ports and outputs the data from one output group with the same ports as the input groups.**

**Q33) What is Update Strategy?**

**A33) Update strategy is used to decide on how you will handle updates in your project. When you design your data warehouse, you need to decide what type of information to store in targets. As part of your target table design, you need to determine whether to maintain all the historic data or just the most recent changes.**

**For example, you might have a target table, T\_CUSTOMERS that contains customer data. When customers address changes, you may want to save the original address in the table instead of updating that portion of the customer row. In this case, you would create a new row containing the updated address, and preserve the original row with the old customer address.**

**This illustrates how you might store historical information in a target table. However, if you want the T\_CUSTOMERS table to be a snapshot of current customer data, you would update the existing customer row and lose the original address.**

**The model you choose determines how you handle changes to existing rows.**

**In Power Center, you set your update strategy at two different levels:**

**1) Within a session. When you configure a session, you can instruct the Power Center Server to either treat all rows in the same way (for example, treat all rows as inserts), or use instructions coded into the session mapping to flag rows for different database operations.**

2) Within a mapping. Within a mapping, you use the Update Strategy transformation to flag rows for insert, delete, update, or reject.

**Note:** You can also use the Custom transformation to flag rows for insert, delete, update, or reject.

**Q34) Joiner transformation?**

**A34) A Joiner transformation joins two related heterogeneous sources residing in different location. The combination of sources can be varied like**

- two relational tables existing in separate database.
- two flat files in potentially different file systems.
- two different ODBC sources.
- two instances of the same XML sources.
- a relational table and a flat file source.
- a relational table and a XML source.

**Q35) How many types of Joins can you use in a Joiner?**

**A35) There can be 4 types of joins**

a) Normal Join (Equi Join)

b) Master Outer Join - In master outer join you get all rows from Detail table

c) Detail Outer Join - In Detail Outer Join you get all rows from Master table

d) FULL Outer Join

**Q36) What are Mapping Parameter & variables ?**

**A36) We Use Mapping parameter and variables to make mappings more flexible.**

**Value of a parameter does not change during session, whereas the value stored in a variable can change.**

**Q37) TELL ME ABOUT PERFORMANCE TUNING IN INFORMATICA?**

**A37) Basically Performance Tuning is an Iterative process, we can do lot of tuning at database level and if database queries are faster than Informatica workflows will be automatically faster.**

**For Performance tuning, first we try to identify the source / target bottlenecks. Meaning that first we see what can be doing so that Source data is being retrieved as fast possible.**

**We try to filter as much data in SOURCE QUALIFIER as possible. If we have to use a filter then filtering records should be done as early in the mapping as possible.**

**If we are using an aggregator transformation then we can pass the sorted input to aggregator. We need to ideally sort the ports on which the GROUP BY is being done.**

**Depending on data an unconnected Lookup can be faster than a connected Lookup.**

**Also there should be as less transformations as possible. Also in Source Qualifier, we should bring only the ports which are being used.**

**For optimizing the TARGET, we can disable the constraints in PRE-SESSION SQL and use BULK LOADING.**

**IF the TARGET Table has any indexes like primary key or any other indexes / constraints then BULK Loading will fail. So in order to utilize the BULK Loading, we need to disable the indexes.**

**In case of Aggregator transformation, we can use incremental loading depending on requirements.**

**Q38) What kind of workflows or tasks have you used?**

**A38) I have used session, email task, command task, event wait tasks.**

**Q39) Explain the process that happens when a WORKFLOW Starts?**

**A39) when a workflow starts, the Informatica server retrieves mappings, workflows & session metadata from the repository to extract the data from the source, transform it & load it into Target.**

- it also runs the task in the workflow.
- The informatica server uses load manager & Data Transformation manager (DTM) process to run the workflow.
- The informatica server can combine data from different platforms & source types. For ex. joins data from flat file & an oracle source. It can also load data to different platforms & target types. For ex. can load, transform data to both a FF target & a MS SQL server db in same session.

**Q40) What all tasks can we perform in a Repository Manager?**

**A40) The Repository Manager allows you to navigate through multiple folders & repositories & perform basic repository tasks.**

**Some examples of these tasks are:**

- Add or remove a repository
- work with repository connections: can connect to one repository or multiple repositories.
- View object dependencies: b4 you remove or change an object can view dependencies to see the impact on other objects.
- terminate user connections: can use the repo manager to view & terminate residual user connections
- Exchange metadata with other BI tools: can export & import metadata from other BI tools like cognos, BO..

**IN REPOSITORY MANAGER NAVIGATOR WINDOW, WE FIND OBJECTS LIKE:**

- \_ Repositories: can be standalone, local or global.
- Deployment groups: contain collections of objects for deployment to another repository in the domain.
- Folders: can be non-shared.
- Nodes: can include sessions, sources, targets, transformation, mapplets, workflows, tasks, worklets & mappings.
- Repository objects: same as nodes along with workflow logs & sessions logs.

**Q41) Did you work on ETL strategy?**

**A41) Yes, my Data modeler & ETL lead along with developers analyzed & worked on dependencies between tasks (workflows).**

**well there are Push & Pull strategies which are used to determine how the data comes from source systems to ETL server.**

**Push strategy: with this strategy, the source system pushes data (or send the data) to the ETL server.**

**Pull strategy: with this strategy, the ETL server pulls the data (or gets the data) from the source system.**

**Q42) How did you migrate from Dev environment to UAT / PROD Environment?**

**A42) We can do a folder copy or export the mapping in XML Format and then Import it another Repository or folder.**

**In my last project we used Deployment groups.**

**Q43) External Scheduler?**

**A43) with external schedulers, we used to run informatica jobs like workflows using pmcmd command in parallel with some oracle jobs like stored procedures. There were various kinds of external schedulers available in market like AUtosys, Maestro, and Control M. So we can use for mix & match for informatica & oracle jobs using external schedulers.**

**Q44) What is a Slowly Changing Dimension?**

**A44) In a Data Warehouse, usually the updates in Dimension tables don't happen frequently.**

So if we want to capture changes to a dimension, we usually resolve it with Type 2 or Type 3 SCD. So basically we keep historical data with SCD.

Q11) Explain SLOWLY CHANGING DIMENSION (SCD) Type, which one did you use?

A11) There are 3 ways to resolve SCD. First one is Type 1, in which we overwrite the changes, so we lose history.

Type 1

OLD RECORD

=====

Surr Dim Cust\_Id Cust Name  
Key (Natural Key)

=====

1 C01 ABC Roofing

NEW RECORD

=====

Surr Dim Cust\_Id Cust Name  
Key (Natural Key)

=====

1 C01 XYZ Roofing

I mainly used Type 2 SCD.

In Type 2 SCD, we keep effective date and expiration date.

For older record, we update the exp date as the The current Date - 1, if the changes happened today.

In the current Record, we keep Current Date as

Surr Dim Cust\_Id Cust Name Eff Date Exp Date

Key (Natural Key)

=====

1 C01 ABC Roofing 1/1/0001 12/31/9999

Suppose on 1st Oct, 2007 a small business name changes from ABC Roofing to XYZ Roofing, so if we want to store the old name, we will store data as below:

Surr Dim Cust\_Id Cust Name Eff Date Exp Date

Key (Natural Key)

=====

1 C01 ABC Roofing 1/1/0001 09/30/2007

101 C01 XYZ Roofing 10/1/2007 12/31/9999

We can implement TYPE 2 as a CURRENT RECORD FLAG Also

In the current Record, we keep Current Date as

Surr Dim Cust\_Id Cust Name Current\_Record

Key (Natural Key) Flag

=====

1 C01 ABC Roofing Y

Suppose on 1st Oct, 2007 a small business name changes from ABC Roofing to XYZ Roofing, so if we want to store the old name, we will store data as below:

Surr Dim Cust\_Id Cust Name Current\_Record  
Key (Natural Key) Flag

=====

1	C01	ABC	Roofing	N
101	C01	XYZ	Roofing	Y

**Q45) What is a Mapplets? Can you use an active transformation in a Mapplet?**

**A45) A mapplet has one input and output transformation and in between we can have various mappings.**

**A mapplet is a reusable object that you create in the Mapplet Designer. It contains a set of transformations and allows you to reuse that transformation logic in multiple mappings.**

**Yes we can use active transformation in a Mapplet.**

**1) A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.**

**In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.**

**A common way of introducing data warehousing is to refer to the characteristics of a data warehouse as set forth by William Inmon:**

**Subject Oriented**

**Integrated**

**Nonvolatile**

**Time Variant**

**2) Surrogate Key**

**Data warehouses typically use a surrogate, (also known as artificial or identity key), key for the dimension tables primary keys. They can use Infa sequence generator, or Oracle sequence, or SQL Server Identity values for the surrogate key.**

**There are actually two cases where the need for a "dummy" dimension key arises:**

**1) the fact row has no relationship to the dimension (as in your example), and**

**2) the dimension key cannot be derived from the source system data.**

**3) Facts & Dimensions form the heart of a data warehouse. Facts are the metrics that business users would use for making business decisions. Generally, facts are mere numbers. The facts cannot be used without their dimensions. Dimensions are those attributes that qualify facts. They give structure to the facts. Dimensions give different views of the facts. In our example of employee expenses, the employee expense forms a fact. The Dimensions like department, employee, and location qualify it. This was mentioned so as to give an idea of what facts are.**

**Facts are like skeletons of a body.**

**Skin forms the dimensions. The dimensions give structure to the facts.**

**The fact tables are normalized to the maximum extent.**

**Whereas the Dimension tables are de-normalized since their growth would be very less.**

**4) Type 2 Slowly Changing Dimension**

**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. Type 2 slowly changing dimension should be used when it is necessary for the data warehouse to track historical changes.**

**SCD Type 2**

**Slowly changing dimension Type 2 is a model where the whole history is stored in the database. An additional dimension record is created and the segmenting between the old record values and the new (current) value is**



easy to extract and the history is clear.

The fields 'effective date' and 'current indicator' are very often used in that dimension and the fact table usually stores dimension key and version number.

#### **4) CRC Key**

Cyclic redundancy check, or CRC, is a data encoding method (noncryptographic) originally developed for detecting errors or corruption in data that has been transmitted over a data communications line.

During ETL processing for the dimension table, all relevant columns needed to determine change of content from the source system (s) are combined and encoded through use of a CRC algorithm. The encoded CRC value is stored in a column on the dimension table as operational meta data. During subsequent ETL processing cycles, new source system(s) records have their relevant data content values combined and encoded into CRC values during ETL processing. The source system CRC values are compared against CRC values already computed for the same production/natural key on the dimension table. If the production/natural key of an incoming source record are the same but the CRC values are different, the record is processed as a new SCD record on the dimension table. The advantage here is that CRCs are small, usually 16 or 32 bytes in length, and easier to compare during ETL processing versus the contents of numerous data columns or large variable length columns.

5) Data partitioning, a new feature added to SQL Server 2005, provides a way to divide large tables and indexes into smaller parts. By doing so, it makes the life of a database administrator easier when doing backups, loading data, recovery and query processing.

Data partitioning improves the performance, reduces contention and increases availability of data.

Objects that may be partitioned are:

- Base tables
- Indexes (clustered and nonclustered)
- Indexed views

#### **Q46) Why we use stored procedure transformation?**

A46) Stored Procedure transformation is an important tool for populating and maintaining databases.

Database administrators create stored procedures to automate time-consuming tasks that are too complicated for standard SQL statements.

You might use stored procedures to do the following tasks:

Check the status of a target database before loading data into it.

Determine if enough space exists in a database.

Perform a specialized calculation.

Drop and recreate indexes.

#### **Q47) What is source qualifier transformation?**

A47) When you add a relational or a flat file source definition to a mapping, you need to connect it to a Source Qualifier transformation. The Source Qualifier represents the rows that the Informatica Server reads when it executes a session.

The Transformation which Converts the source (relational or flat) data type to

Informatica datatype. So it works as an intermediary between source and Informatica server.

Tasks performed by qualifier transformation: -

1. Join data originating from the same source database.
2. Filter records when the Informatica Server reads source data.
3. Specify an outer join rather than the default inner join.
4. Specify sorted ports.
5. Select only distinct values from the source.
6. Create a custom query to issue a special SELECT statement for the Informatica Server to read source data.

**Q48) What is CDC, changed data capture?**

**A48) Whenever any source data is changed we need to capture it in the target system also this can be basically in 3 ways**

**Target record is completely replaced with new record.**

**Complete changes can be captured as different records & stored in the target table.**

**Only last change & present data can be captured.**

**CDC can be done generally by using a timestamp or version key**

**Q49) What is Load Manager and DTM (Data Transformation Manager)?**

**A49) Load manager and DTM are the components of Informatica server. Load manager manages the load on the server by maintaining a queue of sessions and release the session based on first come and first serve.**

**When the session is released from the load manager it initializes the master process called DTM. DTM modifies the data according to the instructions coded in the session mapping.**

**The Load Manager creates one DTM process for each session in the workflow. It performs the following tasks:**

**Reads session information from the repository.**

**Expands the server session and mapping variables and parameters.**

**Creates the session log file.**

**Validates source and target code pages.**

**Verifies connection object permissions.**

**Runs pre-session shell commands stored procedures and SQL.**

**Creates and run mapping reader writer and transformation threads to extract transform and load data.**

**Runs post-session stored procedures SQL and shell commands.**

**Sends post-session email.**