1. What is a Data Warehouse?
   A data warehouse is a large store of data collected from a wide range of sources within a company and used to guide management decisions. It is subject-oriented, integrated, time-variant, and non-volatile collection of data.

2. What is Data Modelling?
   Data modelling is the process of creating a data model for the data to be stored in a database. It visually represents data objects, the associations between different data objects, and the rules governing these associations.

3. What are the types of Data Models?
   There are mainly three types of data models: Conceptual, Logical, and Physical. The Conceptual Model defines WHAT the system contains. The Logical Model defines WHAT the system does, and the Physical Model defines HOW the system performs.

4. What is the difference between a Database and a Data Warehouse?
   A Database is a collection of related data organized in a structured way. It is designed to handle transactions and enables real-time processing. On the other hand, a data warehouse is designed for analysis and query processing and is intended to support decision making. It contains historical data derived from transaction data but can include data from other sources.

5. What are Fact Tables and Dimension Tables in a Data Warehouse?
   Fact tables store quantitative information for analysis and are often de-normalized. It contains foreign keys referring to candidate keys in Dimension tables. Dimension tables store dimensions of a fact, such as product, location, time, etc., and are often denormalized.

6. What is ETL?
   ETL stands for Extract, Transform, Load. It's a process of extracting data from source systems, transforming it into a format that can be analyzed, and then loading it into a data warehouse or similar system.

7. What is a Star Schema?
   Star schema is a relational schema whose design represents a multidimensional data model. The star schema consists of one or more fact tables referencing any number of dimension tables. The star schema gets its name from the physical model's resemblance to a star shape with a fact table in the middle and the dimension tables surrounding it representing the star's points.

8. What is a Snowflake Schema?
   Snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables. This reduces data redundancy, but at the cost of more complex queries and reduced query performance.

9. What is a Surrogate Key?
   A Surrogate Key is a substitution for the natural primary key. It is a unique identifier for each record in a table. It is beneficial because it is simple, stable, and allows changes to natural keys.

10. What is OLTP vs OLAP?
    OLTP (Online Transactional Processing) is a system that manages transaction-based applications in the real-time scenario that can be used for online database modifying operations. OLAP (Online Analytical Processing) is a system designed for analysis of business data for strategic decision-making.

11. What is a Data Mart?
    A Data Mart is a subset of a data warehouse that relates to specific business line. Data marts are managed by a specific department within an organization.

12. What is a Normalization?
    Normalization is the process of organizing data to minimize redundancy. It usually divides a database into two or more tables and defines relationships between the tables.

13. What is De-normalization?
    Denormalization is the process of adding redundancy to speed up complex queries involving multiple table joins. It is used in the data warehouse to simplify the complex queries and enhance the performance of data fetching.

14. What is ER Diagram?
    ER Diagram or Entity-Relationship Diagram is a visual representation of entities and relationships. It describes how data is related to each other. In ER Modeling, the database structure is represented graphically.

15. What is a Fact Constellation Schema?

Fact Constellation Schema or Galaxy Schema is a model that consists of multiple fact tables sharing dimension tables, viewed as a collection of stars.

16. What is Data Mining?
    Data Mining is a process to extract information from a data set and transform it into a comprehensible structure for further use.

17. What is a Slowly Changing Dimension (SCD)?
    Slowly Changing Dimensions (SCDs) are dimensions in which data changes slowly, rather than changing on a time-based, regular schedule.

18. What are the types of SCDs?
    There are three types of SCDs, defined as SCD Type 1 (overwrite), SCD Type 2 (history), and SCD Type 3 (new column).

19. What is a Factless Fact Table?
    Factless Fact Table is a fact table that does not have any measures. It is essentially an intersection of dimensions (it contains nothing but dimensional keys).

20. What is Dimensional Modelling?
    Dimensional Modeling (DM) is a data structure technique optimized for Data Warehousing tools. The concept of DM is to have a table in a center surrounded by several other lookup tables each joined by a foreign key. It provides a way to improve query performance in relational databases.

21. What is a Hash Partition?
    Hash partitioning is a partitioning technique where a hash key is used to distribute rows evenly across the different partitions. This can help to ensure a good distribution of data.

22. What are Aggregates?
    Aggregates in a Data Warehouse are the results of mathematical operations performed on data. The purpose of storing aggregates is to improve query performance.

23. What is a Link Table?
    A link table is a table that maps many-to-many relationships between two tables.

24. What is Forward Engineering in Data Modelling?

Forward Engineering involves producing physical model and schema from a logical model.

25. What is Reverse Engineering in Data Modelling?
Reverse engineering in data modelling involves creating a logical model from an existing database or physical model.

26. What are Non-Additive facts?
Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table. An example is average or ratios.

27. What are Semi-Additive facts?
Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.

28. What is a CUBE in data warehousing context?
A cube is a multi-dimensional generalization of a two- or three-dimensional spreadsheet. Cubes are used in data warehouses and business intelligence applications to store and analyze data across multiple dimensions.

29. What are Measures?
Measures are the numerical data based on columns in the fact table. They are used in the function of facts.

30. What is a Schema?
A schema is a logical description of the entire database. It includes the name and description of records of all record types, plus all fields of data elements.

31. What is a View?
A view is a tailored representation of the data contained in one or more tables. View can be considered as a virtual table.

32. What is a Lookup table?
Lookup table is a relational table with one record for each item of a type of entity.

33. What are Update strategy transformations?
Update strategy transformations are used to update data in target tables, either to maintain a history of data or just the new changes.

34. **Question**: Your organization has recently acquired another company with a completely different data infrastructure. What steps would you take to integrate their data into your existing data warehouse?

    Answer: This is a complex process that involves understanding the new data's schema and contents, mapping it to the existing data warehouse schema, and potentially transforming or cleaning the data before it can be merged. Special consideration must be given to handling inconsistencies or conflicts between the two datasets, such as duplicate records or incompatible data formats.

35. **Question**: How would you implement a change data capture mechanism in a data warehouse for real-time analytics?

    Answer: Change data capture (CDC) can be achieved by keeping track of changes in the transactional system and updating the warehouse accordingly. This can involve triggers in the source database that track changes, or it can involve keeping a timestamp or version number with each record and regularly scanning for updated records.

36. **Question**: How would you design a data model to efficiently manage a recursive relationship, such as an employee hierarchy where each employee reports to another employee?

    Answer: In such a scenario, you could use a self-referential relation in the data model. The employee table could include an attribute for 'Manager', which refers back to the employee ID within the same table.

37. **Question**: How would you handle large amounts of unstructured data in a data warehouse, such as user reviews or comments?

    Answer: Unstructured data can be handled in a data warehouse by storing it in a suitable format like JSON or XML within a column of a table. You might also consider using a hybrid data warehousing approach, combining a traditional relational data warehouse with a data lake that can store unstructured data.

38. **Question**: How would you deal with a situation where the data quality in your source system is poor, with many missing or inconsistent records?

    Answer: Poor data quality can be addressed through data cleaning as part of the ETL process. This could involve filling in missing values based on certain rules, standardizing and correcting inconsistent records, and identifying and eliminating duplicate records.

39. **Question**: How would you adjust your data model if the organization decided to switch from a product-centric to a customer-centric business model?

    Answer: You might need to redesign your data model to put customers at the center, such as by using a Customer dimension table as a central fact table. This would involve shifting the focus from transactions to customer interactions and behaviors.

40. **Question**: How would you address the challenge of maintaining data privacy and GDPR compliance in a data warehouse?

    Answer: You would need to implement strict data governance policies, including managing who has access to what data, anonymizing or encrypting sensitive data, ensuring data is only stored as long as necessary, and being able to delete data upon request.

41. **Question**: Your data warehouse is suffering from "data silos", with different departments having their separate databases that aren't integrated. How would you address this issue?

    Answer: You would need to establish an organization-wide data strategy, potentially centralizing the data into a single integrated data warehouse or implementing a data federation approach that allows querying across the separate databases.

42. **Question**: How would you ensure data consistency in a data warehouse where multiple ETL processes are running in parallel?

    Answer: You could use techniques like ensuring transactions are atomic and isolated, coordinating the ETL processes to avoid conflicts, or implementing a system of locks or flags to prevent multiple processes from modifying the same data at the same time.

43. **Question**: How would you model many-to-many relationships in a data warehouse?

    Answer: Many-to-many relationships can be modeled in a data warehouse using a bridge table (also known as a linking or junction table), which includes foreign keys to each of the related tables.

44. **Question**: How would you design a data warehouse for a business with multiple branches worldwide, considering factors like latency and data sovereignty laws?

    Answer: This could involve creating separate data marts for each region, which comply with local data laws and reduce latency by being physically closer to the users. These could be regularly synchronized with a central data warehouse.

45. **Question**: How would you handle a situation where your data warehouse needs to be accessible both by advanced users who write SQL queries and by non-technical users who rely on a simple interface?

    Answer: You would need to ensure the data is structured and documented in a way that makes it easy to query, while also providing a user-friendly BI tool that allows non-technical users to generate reports and visualizations.

46. **Question**: How would you design a data model for a highly interconnected dataset, such as a social network where each user can be connected to many other users?

    Answer: This might involve using a graph data model, which is designed to handle highly interconnected data. Each user would be a node in the graph, and each connection between users would be an edge.

47. **Question**: How would you implement a data warehouse that can handle "big data", with very large volumes of data and high velocity of data changes?

    Answer: Handling big data might involve using a distributed data warehouse that can scale horizontally, using data partitioning to distribute the data across multiple nodes. It might also involve using a columnar storage format for efficient storage and querying of large datasets.

48. **Question**: How would you approach designing a data warehouse for an organization that doesn't yet know what questions they want to ask of their data?

    Answer: This involves building a flexible data model that can handle a wide range of potential queries, and potentially adopting a data lake architecture that allows storing raw data until the business knows what it wants to do with it.

49. **Question**: How would you handle a situation where your source system does not maintain a history of changes, but your data warehouse needs to support historical analysis?

    Answer: You would need to implement a system for tracking historical changes within your data warehouse, such as using slowly changing dimensions (SCDs) to keep a record of how the data has changed over time.

50. **Question**: How would you design a data warehouse for an online retailer that wants to analyze customer behavior and recommend products based on past purchases?

    Answer: This involves designing a data model that captures detailed data about customer interactions, such as browsing history, purchases, and product ratings. It might also involve integrating with a machine learning system for generating recommendations.

51. **Question**: How would you address the challenge of integrating real-time streaming data, such as from IoT devices, into a data warehouse?

    Answer: Integrating real-time data can be achieved using a lambda architecture, which combines a batch processing layer for handling large volumes of stored data with a speed layer for processing real-time data.

52. **Question**: How would you model temporal data in a data warehouse, such as tracking the prices of products over time?

    Answer: Temporal data can be modeled using a slowly changing dimension of Type 2 or Type 3, which keeps a record of changes over time. Alternatively, you could use a separate table to track the history of price changes.

53. **Question**: How would you ensure high availability of a data warehouse, to minimize downtime and avoid data loss?

    Answer: Ensuring high availability might involve using redundant hardware and networking, regular backups and disaster recovery plans, and potentially a multi-region setup that can failover to another region if necessary.