**Question 1**

**Network Model:**

It is a type of database model in which data is organized in shape of graphs and every node of the graph is allowed to have more than one parent nodes, unlike the Hierarchical model. This is an advanced form of Hierarchical model of database. This database is mainly used to map many-to-many relationships in the data.   
Example: Integrated Data Store is an example of Network Model.

**Hierarchical Model:**

It is a type of database model in which data is organized in tree-like shape that has only one Root node and all the other nodes (data) are linked to the root node. This represents relations of one-to-many type.  
Example: Data of school in which there are several teachers, each of who then have many students.

**Object Oriented Model:**

This database model works exactly the same way as the Object Oriented Programming. It allows the developers to develop a product and then store the information as objects of that product.   
Example: A database of a store in which “Order” is a class containing all attributes and then objects are made of it.

**Relational Data Model:**

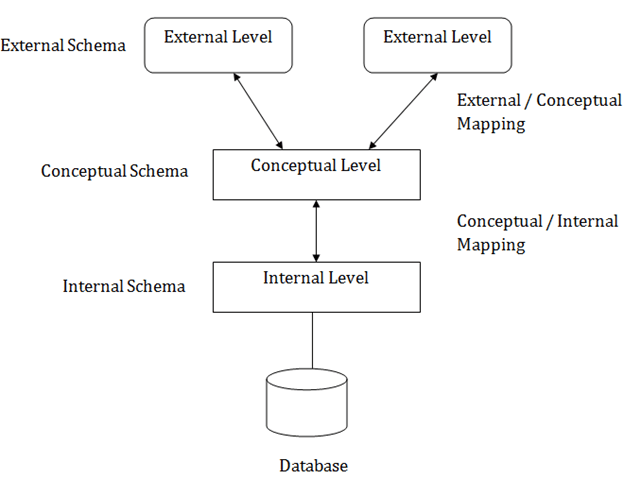
A database in which information is stored in form of collection of relations also called as tables. Every table has a set of rows called the tuples and columns called the attributes of that function.   
Example: SQL Server and Access etc are good examples of Relational Data Model.

**Object Relational Data Model:**

As the name says, this is a combination of Relational Data Model and the Object Oriented Data Model. As a result, it supports objects, classes just like Object Oriented Model and also supports tabular structures etc like the Relational Data Model.  
Example: Oracle Database is an example of such a model.

**Question 2**

**The Three Levels of Schema**

1. **Internal Level:** It has the internal schema of DBMS and explains the physical structure of the database. This physical level is used to explain the complex low-level data structures in detail.
2. **Conceptual Level:** Also known as logical level, this level explains the design of the database at conceptual level. It explains what data is to be stored in the database and what relationships exists among the data to be stored. Internal details are hidden at this level.
3. **External Level:** External Level describes the end user interaction with the database. It has several subschemas which describe different views of the database. It is also known as the view schema, since this is what’s visible to the users. 

**Need of Mapping:**

Mapping between the levels allows for the independency of one layer on another layer whenever any change is required on any level. For example,

1. In case of any change in the architecture of physical structure of database, we can simply change the mapping between the internal and conceptual level.
2. Similarly, a new mapping could be created between conceptual and external level if a new external schema is required, without disturbing the other external schemas.

**Question 3**

* **Data:** Raw facts or figures is known as Data.
* **Database:** Collection of records in an organized form (like in tables) is called Database. Its primary **goal** is to provide a way to store and retrieve information which is convenient and effective.
* **DBMS:** Database Management System, it is a software which is designed to define, manipulate and manage information or record in the database.
* **Database System:** Database System is nothing but Database Management System (DBMS).
* **Database Catalog:** It is a metadata which contains the definitions of database objects such as tables, indexes, users etc.
* **Program-data Independence:** It refers to the capability of the database to keep its data intact and accessible while any kind of changes are made to the database.
* **DBA:** The Database Administrator, the person who’s responsible for managing and maintaining the database.
* **End User:** People who interact with the database with the help of applications or other utilities are the end users. They could be of many types.
* **Meta-data:** The data which provides information about tables, views etc of the database.

**Question 4**

**DDL:** Data Definition Language, a language which is used for the defining of database schemas, data structures etc. DDL statements are used for the modification and creation of database objects such as tables, users etc.

**DML:** Similar to DDL, Data Manipulation Language is the language which consists of the syntax that is used for the manipulation of data of database, such as editing, updating, deleting etc.

**Question 5**

Three-Tier Client for Web Application would make the best choice in this scenario. The client consists of Web User Interface. The Web Server contains the application logic which includes all the rules and regulations related to the reservation process and the issue of tickets; the Database Server contains the DBMS

Two-Tier Client wouldn’t work in such a scenario because it will put an excessive burden on the server.

**Question 6**

* DBA is responsible for ensuring that database backup is being taken properly, and that there is adequate disk, memory etc is available for the database to be running properly and effectively and if there isn’t, then he’s responsible for informing the designer of that.
* Database Designers, on the other hand, are responsible for ensuring that database holds all the information which the end users demand of it.
* They are also responsible for creating and designing physical tables, indexes etc to ensure that queries execute as fast as possible.
* They are responsible for any updating of the database required at any time.

**Question 7**

**Types of Database Users**

* **Application Programmers**: They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal etc. For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database.
* **Sophisticated Users:** They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of query language like SQL.
* **Specialized Users:** These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.
* **Stand-alone Users**: These users will have stand –alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.
* **Native Users:** These are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc which has existing application and users use them to interact with the database to fulfill their requests.

**Question 8**

**Capabilities of DBMS**

* Controlling Redundancy: normalization
* Restricting unauthorized access to database.
* Representing Complex Relationships among Data
* Enforcing integrity constraints
* Providing persistent storage for program objects
* Providing storage structures for efficient query processing
* Backup and recovery

**Question 9**

* Database Systems work with relational databases i.e. structured data, whereas information retrieval system and file system work on unstructured data (such as raw text files/documents).
* File System is more likely to have redundant data compared to Database System.
* For simpler operations such as read, write etc, however, file system and information retrieval system is better than the database as it is faster. But if the data is huge then file system becomes very slow and database system takes the edge in such a case.

**Question 10**

**Centralized and Client-Server architecture for DBMS:**

* The centralized database system consists of a single processor together with it associated data storage devices and other peripherals. It is physical confined to a single location. The data ca be accessed from the multiple sites with the use of a computer network while the database is maintained at the central site.
* Client-Server architecture of database systems has two logical components namely client and server. Client is generally personal computer or workstations whereas server is large workstations, mini range computer system or a mainframe computer system. The application and tools of DBMS run on one or more client platforms, while the DBMS software reside on the server.
* Advantage/Disadvantage: Client-Server architecture is usually better to use since it provides better response time than centralized database system. It is also more flexible. Moreover, centralized database would go down if the central site computer goes down since it is confined to one place, but this wouldn’t happen in case of Client server architecture. However, Client Server architecture is more expensive in terms of programming cost than centralized database system.

**Two-Tier Client-Server Architecture and Three-Tier Client-Server Architecture:**

* Two tire client server architecture consists of two logical layer which are client layer and server layer. There is a communication happens between client and server without any middle ware.
* Three tire client server architecture consists of three logical layer which are client layer, application layer and server layer. There is a communication happens between client and server with middle ware and application layer acts as middle ware.
* Advantage/Disadvantage: Three-Tier Architecture is better when security and performance is concerned, since it provides more security due to indirect access because of three-tier structure and performance loss is also less compared to two-tier. However, two-tier architecture, due to direct access, is faster than three-tier client server architecture.

**Question 11**

**Logical and Physical Database Design**

* Logical database modeling is mainly for gathering information about business needs and does not involve designing a database; whereas physical database modeling is mainly required for actual designing of the database.
* Logical database modeling does not include indexes and constraints; the logical database model for an application can be used across various database software and implementations; whereas physical database modeling is software and hardware specific and has indexes and constraints.
* Logical database modeling includes; ERD, business process diagrams, and user feedback documentation; whereas physical database modeling includes; server model diagram, database design documentation, and user feedback documentation.

**Question 12**

**Six Major Steps in setting up a Database for a particular Enterprise**

Following are the six steps required,

1. Define a model containing all appropriate types of data and data relationships.
2. Define the integrity constraints on the data.
3. Define the conceptual schema for the model.
4. Define the physical level.
5. For each known problem to be solved on a regular basis (e.g., weekly inventory), define a view of the database and write the necessary application programs.
6. Create and initialize the database.

**Question 13**

**Role of High-Level Data Model in Database Design Process**

* High-level data model does not include implementation details, they are easy to understand and are useful in communicating with non-technical users.
* It can also be used as reference to ensure that all user requirements are met and that requirements do not conflict with each other.
* This also enables database designers to concentrate on specifying the properties of data without being concerned with storage details.

**Question 14**

**Why choose Database System over Operating System Files**

* Operating System files do not provide any dictionary of the data that each file contains. Databases have meta data about what they store that u can query as data.
* Database can handle a large set of data, while operating system files will get slower when dealing with large data.
* Database allows indexing based on attributes of the table, which OS files don’t.
* Database provider better security which is also easier to implement, compared to in Operating System
* Redundant data can be avoided in Database system, while it is difficult to avoid in OS files.

**When not to use Database System**

* When you don’t need queries to solve your problem.
* When long-term maintenance of the data is not required.
* When data is simple, not expected to change with time and is also well defined.