Qua. 1) What Is GraphQL? ?

Ans. => GraphQL Is A New API Standard Designed And Developed By Facebook. It Is An Open-Source Server-Side Technology That Is Now Maintained By A Large Community Of Companies And Individuals Worldwide. It Is Also An Execution Engine That Works As A Data Query Language And Used To Fetch Declarative Data.

Qua. 2) What Was The Reason Behind The Development Of GraphQL? ?

Ans. => GraphQL Was Initially Developed By Facebook As An Internal Solution For Their Mobile Apps. It Was Designed To Optimize RESTful API Calls And Provide A Flexible, Robust, And Efficient Alternative To REST. It Is Not A Replacement For REST. It Is An Alternative To Writing APIs Using REST.

After Its Release, It Has Become Prevalent Among Developers And Is Also A Popular Solution For Building Web Services Along With REST.

Qua. 3) What Are The Top Companies That Use GraphQL?

Ans. => There Are Many Big Organizations Such As Facebook, Github, Pinterest, Intuit, Coursera, Shopify, Dailymotion, Yelp Etc. That Uses GraphQL. Actually GraphQL Was Designed And Developed By Facebook Itself.

Qua. 4) What Are The Most Significant Advantages Of Using GraphQL Over REST?

Ans. => A List Of The Most Significant Advantages Of Using GraphQL Over REST:

There Is Only One Endpoint In GraphQL, But REST Has Multiple Endpoints. That's Why GraphQL Is More Cost-Effective Than REST. You Don't Have To Use Your Resources For Various Endpoints.

In REST API, You Have To Use Multiple Requests To Retrieve A Complex Data-Set, But In GraphQL, You Can Execute A Complex Query Easily In Just A Single Request.

You Can Change The Request Data Format In GraphQL, But It Is Not Possible To Do The Same In REST.

The Development Speed In GraphQL Is Faster Than REST.

GraphQL Provides High Consistency Across All Platforms, While In REST, It Is Hard To Get Consistency Across All Platforms.

GraphQL Doesn't Support An Automatic Caching System, While REST Uses Caching Automatically.

Qua.5 What Is The Use Of Resolver In GraphQL?

Ans. => In GraphQL, A Resolver Is Used To Handle Queries And Produce A Response To The GraphQL Query.

Qua. ◆ 6. What Is GraphQL Schema?

Ans. => Every GraphQL Server Has Two Core Parts That Determine How It Works: A Schema And Resolve Functions.

The Schema Is A Model Of The Data That Can Be Fetched Through The GraphQL Server. It Defines What Queries Clients Are Allowed To Make, What Types Of Data Can Be Fetched From The Server, And What The Relationships Between These Types Are.

```
Type Author {
  Id: Int
  Name: String
  Posts: [Post]
}
Type Post {
  Id: Int
  Title: String
  Text: String
  Author: Author
}
Type Query {
  GetAuthor(Id: Int): Author
  GetPostsByTitle(TitleContains: String): [Post]
}
Schema {
  Query: Query
}
```

Qua.7 What Is Query?

Ans. => A Query Is Used To Read Data. Similar To The GET Request Of REST APIs, We Can Fetch Data From A GraphQL Server Using A Query.

Qua. 8. What Do You Mean By Over-Fetching In The GraphQL Language?

Ans. =>

Over Fetching Is Basically A Condition In GraphQL Where The Requested Client Gets More Data Than What Was Requested. This Will Result In An Increase In The Payload Size.

Qua. 9. What Do You Mean By Under-Fetching In The GraphQL Language?

Ans. => Under Fetching Is Basically The Exact Opposite Of Over-Fetching. It Happens When The Requested Client Does Not Get Enough Data And You Are Forced To Send Multiple Calls To Get The Desired Data.

Qua.10 What Is Mutation?

Ans. => Mutation Is Used For The Write Operation. A Mutation Is Used For Operations Like Add, Delete And Edit Data.

Qua.11 How Can You Use Variables In A Query ?

Ans. => Instead Of Writing Arguments In A Query String, We Can Use Variables And Pass Any Arguments To These Queries Directly. For Example :

```
{
Student(Id: "100")
{
Name
Age
}
}
```

.

Qua.12 How Can You Use Variables In A Query ?

Ans. => Instead Of Writing Arguments In A Query String, We Can Use Variables And Pass Any Arguments To These Queries Directly. For Example :

```
Query StudentQuery($Id : Int)
{
Student(Id : $Id){
Name
}
}
```

Qua.13 What Is A Resolver ?

Ans. => Resolver Is Used To Produce A Response To A GraphQL Query. Resolver Is Used To Handle Queries.