Q3&4

A paragraph on what PaaS, SaaS and IaaS are and the differences between them.
IaaS(Infrastructure-as-a-Service), also known as cloud infrastructure services, moves an existing application or workload from on-premise deployment to a cloud-provider's infrastructure. Iaas is often used when companies already have a mature on-premise solution and they want to move to a cloud environment as soon as possible. A cloud provider provides you with a bare metal machine or virtual machine, often coming with open-sourced install and they own and manage the hardware upon which your software stack runs. The moving process does not require much coding and does not need to modify any existing code or system companies have. IaaS typically provides pay-as-you-go service so you always have the flexibility to control how much service you want to use and only pay the necessary amount for what you use. IaaS platforms are highly flexible and highly scalable, accessible by multiple users, and cost-effective. The downside of Iaas is that there is not much of a difference from on-prem. Examples of Iaas are AWS EC2, Microsoft Azure, Google Compute Engine (GCE), IBM Cloud, Rackspace, Digital Ocean, Oracle Cloud, etc.

PaaS (Platform as a Service), also known as cloud platform services, provides businesses with hardware and software tools to help business build their own applications. It is for developer use. PaaS provides the platform including runtime, interface, editor, monitoring tools, etc for developers to create customizable software. Developers don't need to start from scratch developing applications but they still have to write their own logic to use the tool and process the data. Examples of Paas are AWS Elastic Beanstalk, Windows Azure, Heroku, Google App Engine, OpenShift, SAP Cloud, AWS Lambda, Azure Functions.

SaaS (Software as a Service), also known as cloud application services, are end products ready for consumer use. Consumers no need to worry about how the services is implemented or how it works. They are typically ready-to-use and simply take a login to a user's web browser. Examples of Saas are Salesforce, SAP concur, Shopify, MailChimp, Dropbox, Hubspot, Google Workspace (G Suite).

Differences: IaaS vs. PaaS: IaaS offers you great control over your operating systems. While PaaS allows you to build apps without having to host them on-premise, there's more flexibility but less control. And there's more coding involved in Paas than in IaaS.
PaaS vs. SaaS: PaaS is used as a way to build new products on top of your already existing network. But SaaS takes this one step further. SaaS products are entirely managed by the vendor and ready to use by your teams.
IaaS vs. SaaS: With a SaaS product, you are getting the most service from the provider in terms of software management and maintenance. With IaaS on the other hand, the provider only supplies and maintains core components such as servers or storage.

A paragraph on the differences between ETL and ELT. Also, list the pros and cons of each in a chart. And specify when you'll use which.
ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform) are both data integration methods that move raw data from a source system to a target database. Their main task is to transfer data from one place to another. The main difference is that ETL transforms data before loading it on the server, while ELT transforms it afterward, and ETL transforms data on a separate processing server, while ELT transforms data within the data warehouse itself. ETL extracts data from data sources, deposit data into a staging area, clean and transform data there, and finally store it in a data warehouse. ETL is used when data must be transformed to conform to the data regime of a target database, usually for structured data. ETL is ideal for small data sets with complicated transformation requirements. OLAP data warehouse is an example that only accepts relational data structures and requires ETL. Unlike ETL, ELT does not require data transformations to take place before the loading process. ELT loads raw data directly into a target data warehouse, instead of moving it to a processing server or a staging area for transformation. With the ELT data pipeline, data cleansing and transformation all occur inside the data warehouse itself. Raw data is stored indefinitely in the data warehouse, allowing for multiple transformations. ELT is best suited for large volumes of data implemented in cloud environments.

	ETL	ELT
Pros	**Fast analysis -** Once the data is structured and transformed with ETL, data queries are much more efficient than unstructured data, which leads to faster analysis. **Flexibility of environment-** ETL can be implemented in either on-premise or cloud-based environments. Organizations will often use ETL to take data from on-premise systems and load it to a cloud datastore. **Compliance -** ETL transforms data before it reaches its destination. When companies are subject to data privacy regulations such as GDPR, ETL allows them to remove, mask, or encrypt sensitive data before it's loaded to the data warehouse to ensure compliance. **Maturity -** ETL was developed first and has been in practice for more than two decades. There are more engineers with experience in ETL implementations and more ETL tools in the marketplace to build data pipelines within organizations.	**Great flexibility of data formats** and **ease of storing** new, unstructured data. **Low-Maintenance -** ELT is cloud-based, it utilizes automated solutions instead of relying on the user to initiate manual updates. With fewer systems, the maintenance burden is reduced. **Quicker Loading -** without transformation needed, ELT allows for all of the data to go into the system immediately. **Transformation as needed -** In an ELT model, transformation typically happens only when analysis is needed versus transforming all data before it's loaded, which means greater efficiency of resources. **High availability of data -** With ELT, all data is loaded to the data lake, so it's always available. This allows tools that don't require structured data to interact with the loaded data immediately instead of waiting until it's transformed.
Cons	**Slow loading speed -** Because data must be transformed in a staging area before it's loaded, it is not available as quickly in the datastore as opposed to ELT where data is loaded as soon as it's extracted. **Rigidity of workflow -** If the structure of data in the warehouse does not support new queries or analyses (that are determined to be valuable), then the transformation process and schema of the data warehouse may need to be modified. **Small data volume -** ETL is not ideal for handling large volumes of data given the time needed for transformation. Instead, it's best suited for smaller data sets that require in-depth manipulation and are known to have data relevant for analysis.	**Less mature -** ELT is new and still evolving, with less documentation and experience. It's more difficult to find experienced ELT developers. **Privacy issue -** Direct loading of data requires more privacy safeguards. **Slow analysis -** requires more time to process data for queries since that step only happens as needed versus ETL where the data is instantly queryable after loading. **Less flexibility of environment -** Though theoretically possible in an on-premise environment, the true advantages of ELT are only possible when paired with the storage and processing power of the cloud. **Compliance issue -** Regulations may prohibit companies from storing sensitive data. ELT's integration with the cloud may be a second issue since some regulations prohibit storing information on servers outside of a specific region or country's borders.