Name: Pradip Bochare

➕ **Notes**

* **CI/CD**

- CI/CD pipeline is a concept central to software. It spans a whole field of processes, testing methods and tooling all facilitated by Git code versioning process.

* Continuous Integration
- Every time you add new track piece, you immediately test it by running toy train (data) through it.

- This ensures that your new addition didn't introduce any problem. If there's issue, you know instantly and can fix it before it becomes bigger problem.

* Continuous deployment
- Once confirmed that new piece fits & train runs smoothly, immediately let everyone see and use the updated track.

- CI checks and tests every new piece of code you add to your data pipeline

- CD ensures that once tested and approved this code gets added to live system without manual intervention

* Continuous Integration (CI) in data pipelines.

- Automated Testing
Automated tests check integrity and quality of data
transformations, ensuring data is processed as
excepted & any error is spotted early.

- Version control
Data pipeline code is stored in repositories like Git,
allowing tracking and managing changes.

- Consistent Environment
CI tools can run tests in environments that mirror
production, ensuring that differences in configuration
or dependencies don't introduce errors.

- Data Quality checks.
This might include checks for null values, data range
violations, data type mismatches or other custom
quality rules.

* Continuous data pipeline deployment

- Automated Deployment
Once code changes pass all CI checks, CD tools can
automate their deployment to production, ensuring
seamless data flow.

- Monitoring and Alerts.
Once deployed, monitoring tools keep track of the pipeline's performance, data quality, and any potential issues. Automated alerts can notify on discrepancies.

- Rollbacks
In case an issue is identified post-deployment CD processes allow for quick rollbacks to a previously stable state of data pipeline.

- Infrastructure as Code (IaC)
Many CD tools supports IaC. Ex. cloud resources such as storage or compute can be provisioned automatically as part of deployment process.

* Git
- Git is a distributed version control system that facilitates collaborative software development by tracking changes across multiple contributors.

- Git can be paired with data orchestration tools and integrated into CI/CD workflows, providing benefits of streamlined deployment and consistency in data engineering tasks.