**Contact**
contact@allondata.in
+91-7358866596

ALLONDATA ACADEMY

# GCP Data Engineering Program

From Cloud Fundamentals to Production-Grade ETL Pipelines

A 100% project-oriented learning program. Every service and concept below is taught through one evolving, real-world ETL pipeline, not isolated toy examples, so what you learn is what you'll actually use on the job.

### How this program is structured

**Track 1, Data Engineering Concepts:** the vendor-neutral fundamentals every data engineer needs, regardless of cloud.

**Track 2, GCP Services:** the cloud tools, taught in the order you'll actually pick them up while building a pipeline (storage → security → compute → orchestration → scale → DevOps).

**Track 3, Claude Code and AI-Assisted Data Engineering:** the AI-assisted engineering practices and tools you'll use throughout the program and in your career.

**Track 4, Career Visibility & Personal Branding:** the professional presence, network, outreach, and interview readiness that help you turn your skills into opportunities.

**Track 5, Program Outcomes:** the skills, roles, professional presence, and interview readiness you'll have once the program is complete.

# Contents


**Track 1: Data Engineering Concepts** 3
- Data Fundamentals 3
- File Formats 3
- Data Modeling 3
- Slowly Changing Dimensions (SCD) 3
- Batch vs Stream Processing 3
- Pipeline Orchestration Principles 4
- End-to-End ETL Architecture (Capstone Concept) 4
- SQL & Python for Data Engineers 4

**Track 2: GCP Services** 5
- GCP Foundations 5
- Identity & Access Management (IAM) 5
- Cloud Storage (GCS) 5
- BigQuery 6
- Cloud Run (Functions & Jobs) 8
- Orchestration & Automation Services 9
- Large-Scale Processing: The Decision Layer 9
- DevOps for GCP 9

**Track 3: Claude Code and AI-Assisted Data Engineering** 10
- Why AI-Assisted Engineering, Practically 10
- Claude Code Fundamentals 10
- Claude Code Across the GCP & Data Engineering Tracks 10
- Agents & Subagents 11
- MCP (Model Context Protocol) 11
- Skills 11
- Memory 11
- Context Management 12
- Agentic Teams / Multi-Agent Workflows 12
- Claude Code for Career Visibility 12

**Track 4: Career Visibility & Personal Branding** 13
- Resume Building 13
- LinkedIn Profile Building 13
- Community & Network Building 14
- Outreach & Referral Strategy 14
- Interview Readiness 14

**Track 5: Program Outcomes** 15
- Skills You Will Have 15
- Roles You'll Be Ready For 15
- Professional Presence You Will Have Built 15
- Interview & Career Readiness 16

TRACK 1

# Data Engineering Concepts

*Cloud-agnostic fundamentals. These skills transfer to any platform: AWS, Azure, or GCP.*

## 1. Data Fundamentals

1 item

- Data vs Database vs Data Warehouse vs Data Lake: what each is actually for

> **You'll learn to:** speak the vocabulary of data architecture correctly from day one.

## 2. File Formats

1 item

- CSV, JSON, Avro, Parquet, ORC: structure, compression, and when to use each

> **You'll learn to:** choose the right file format for a pipeline based on cost, performance, and downstream use case.

## 3. Data Modeling

3 items

- Normalization vs Denormalization
- Fact Tables vs Dimension Tables
- Star Schema vs Snowflake Schema

> **You'll learn to:** design warehouse schemas that are fast to query and easy to maintain. This is a top recurring interview topic.

## 4. Slowly Changing Dimensions (SCD)

1 item

- Types 0, 1, 2, 3, 4, and 6: how each handles historical data, and when to use it

> **You'll learn to:** correctly implement change-tracking in dimension tables, a skill that separates junior from mid-level data engineers.

## 5. Batch vs Stream Processing

1 item

- Core characteristics of each approach, and the GCP services that support them

> **You'll learn to:** decide when a pipeline needs to be real-time versus scheduled, and design accordingly.

## 6. Pipeline Orchestration Principles

2 items

- Tasks and dependencies
- Sequential, parallel, and conditional execution patterns

> **You'll learn to:** design pipeline logic that is resilient, efficient, and easy to debug when something breaks.

## 7. End-to-End ETL Architecture *(Capstone Concept)*

2 items

- The layered pipeline design: Source → Landing → Raw → Curated → Target
- Fail-fast validation, error handling, and reusability across multiple pipelines

> **You'll learn to:** architect a complete, production-grade ETL system from scratch. It is the single most valuable skill this program builds toward, and the one interviewers probe hardest.

## 8. SQL & Python for Data Engineers

2 items

- What level of SQL is actually needed (joins, window functions, partitioning/clustering)
- What level of Python is actually needed (data structures, reading/validating LLM-generated code)

> **You'll learn to:** meet the real, practical bar companies expect: not academic mastery, but working fluency.

TRACK 2

# GCP Services

## 1. GCP Foundations

3 items

*Before you touch a single service, you need to know why the cloud exists and where you fit in.*

- What is Google Cloud Platform, and why organizations choose it
- The data engineer's role in the GCP ecosystem
- Global infrastructure: Regions and Zones, and why they matter for latency, availability, and compliance

> **You'll learn to:** explain GCP's value proposition and navigate its global infrastructure model with confidence.

## 2. Identity & Access Management (IAM)

3 items

*Access control comes before real services are handed to you. This is how every GCP project stays secure.*

- Permissions, Roles, and how they combine (Predefined vs Custom Roles)
- The Least Privilege Principle and why over-permissioning causes real production incidents
- Service Accounts: what they are, why applications need their own identity, and how to name/manage them

> **You'll learn to:** design access policies the way cloud security teams expect, and set up service accounts correctly for any application.

## 3. Cloud Storage (GCS)

6 items

*The entry point of virtually every data pipeline. This is where raw data lands first.*

- Buckets, objects, and folder-structure conventions
- Storage Classes (Standard, Nearline, Coldline, Archive) and cost optimization
- Location types: Region, Dual-Region, Multi-Region
- Bucket-level access control and industry naming standards
- Core `gcloud`/`gsutil` operations: create, upload, copy, move, delete
- IAM roles specific to GCS (Object Viewer, Object Admin, Storage Admin)

> **You'll learn to:** stand up and manage a production-ready data lake layer, and choose the right storage tier for any access pattern.

# 4. BigQuery

117 items

*The analytical warehouse, where raw data becomes business insight.*

## Fundamentals

### Basic SELECT & Filtering

- SELECT
- WHERE
- ORDER BY
- DISTINCT
- LIMIT
- Column aliases

### Operators

#### Arithmetic Operators

- +
- -
- *
- /

#### Comparison Operators

- =
- <
- >
- <=
- >=
- <>

#### Logical Operators

- AND
- OR
- NOT

#### Membership Operators

- IN
- NOT IN

#### Range Operator

- BETWEEN ... AND

#### Pattern Matching Operators

- LIKE
- NOT LIKE

### Data Types

- Numeric data types
- String data types
- Date and time data types
- Boolean values
- NULL

### Single-Row Functions

#### CASE / Conditional Logic

- CASE
- IF

#### Character Functions

- UPPER()
- LOWER()
- INITCAP()
- SUBSTR()
- INSTR()
- LPAD()
- RPAD()
- LTRIM()
- RTRIM()
- TRIM()
- CONCAT()
- REPLACE()
- TRANSLATE()
- REVERSE()
- LENGTH()

### Number Functions

- ROUND()
- TRUNC()
- MOD()
- POWER()
- ABS()

### Date Functions

- CURRENT_DATE()
- DATE_ADD()
- DATE_SUB()
- DATE_DIFF()
- DATE_TRUNC()
- EXTRACT()
- LAST_DAY()
- FORMAT_DATE()
- PARSE_DATE()

### Conversion Functions

- CAST()
- EXTRACT()

### General / NULL Functions

- IFNULL()
- NULLIF()
- COALESCE()

## Aggregate Functions & Grouping

### Aggregate Functions

- MIN()
- MAX()
- COUNT()
- AVG()
- SUM()

### Grouping

- GROUP BY
- HAVING

## Joins

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL OUTER JOIN

## Subqueries

- Subqueries in SELECT
- Subqueries in WHERE
- Subqueries in FROM
- IN subqueries
- Correlated subqueries

## Window / Analytical Functions

### Window-Specific Functions

- ROW_NUMBER()
- RANK()
- DENSE_RANK()
- LEAD()
- LAG()

### Aggregate Functions Used as Window Functions

- MIN()
- MAX()
- COUNT()
- AVG()
- SUM()

### Window Clauses

- OVER
- PARTITION BY

## CTEs (WITH Clause)

- WITH clause

### BigQuery & Table Concepts

#### DDL

- `CREATE`
- `ALTER`
- `DROP`
- `TRUNCATE`

#### DML

- `INSERT`
- `UPDATE`
- `DELETE`
- `MERGE`

#### BigQuery Tables

- `Native Tables`
- `Temporary Tables`
- `GCS External Tables`
- `BigLake Tables`

#### Views & Materialized Views

- `Views`
- `Materialized Views`
- `When to use`
- `When not to use`

#### Partitioning & Clustering

- Partitioning
- Clustering

### BigQuery Optimization

- Selecting only required columns
- Efficient filtering
- Proper join usage
- Avoiding unnecessary data processing
- General BigQuery query optimization practices

> **You'll learn to:** design and query a serverless data warehouse at scale, and write cost-efficient SQL that a senior engineer would approve.

---

## 5. Cloud Run (Functions & Jobs)

3 items

*The compute layer. This is where your data actually gets processed.*

- Cloud Run Functions: HTTP vs event triggers, scaling behavior, cold starts, IAM roles
- Packaging code with Docker and publishing to Artifact Registry
- Cloud Run Jobs: building the processing layers of a real ETL pipeline (ingestion, raw load, validation, curated transformation, and delivery to downstream systems)

> **You'll learn to:** containerize and deploy serverless data-processing jobs, and understand exactly how a production ETL pipeline is broken into stages.

## 6. Orchestration & Automation Services

4 items

*Once you have storage and compute, you need something to run them in the right order, automatically.*

- Cloud Scheduler: time-based triggering (cron-style automation)
- Eventarc: event-driven triggering (react the moment something happens)
- Pub/Sub: asynchronous messaging and the backbone of streaming pipelines
- Cloud Workflows: orchestrating multi-step pipelines, and how it compares to Airflow/Cloud Composer

> **You'll learn to:** automate an end-to-end pipeline without babysitting it, and choose the right trigger mechanism for any scenario (scheduled, event-driven, or message-based).

## 7. Large-Scale Processing: The Decision Layer

1 item

*Knowing every tool is less valuable than knowing which one to reach for.*

- Cloud Run vs Dataflow vs Dataproc: a practical decision framework based on data volume, complexity, and team skillset

> **You'll learn to:** justify architecture decisions in a design review or interview: not just use a service, but explain why it was the right choice.

## 8. DevOps for GCP

3 items

*How real teams ship infrastructure and code changes safely.*

- Terraform: Infrastructure as Code, core commands (`init`, `plan`, `apply`)
- Git/GitHub branching strategy for multi-environment projects
- CI/CD pipelines: automated build, test, and deployment across dev/QA/prod

> **You'll learn to:** version and deploy cloud infrastructure the way modern engineering teams do, with no manual console clicking in production.

TRACK 3

# Claude Code and AI-Assisted Data Engineering

*Every track above is taught with Claude Code alongside you. It is not a shortcut, it is how data engineers actually work in 2026. This track covers Claude Code as a skill in its own right, on top of how it's used throughout the program.*

## 1. Why AI-Assisted Engineering, Practically

3 items

- Why professional data engineers use AI coding assistants daily. This isn't "cheating," it's the current baseline expectation on real teams
- What changes and what doesn't: you're still responsible for correctness, cost, and security. The assistant writes, you validate
- The core discipline: read every diff, understand every query, never ship what you can't explain

> **You'll learn to:** use AI assistance the way a senior engineer does: as leverage, not as a replacement for understanding.

## 2. Claude Code Fundamentals

3 items

- Setting up and navigating a real project with Claude Code
- Core workflow: plan → edit → run → review diff → commit
- Effective prompting for engineering tasks: giving the right context, scoping requests, iterating

> **You'll learn to:** operate Claude Code as your daily driver for writing and reviewing code, not just as a chat window.

## 3. Claude Code Across the GCP & Data Engineering Tracks

3 items

- For every service in Track 2 (GCS, BigQuery, Cloud Run, Workflows, Terraform, etc.), you'll first learn *what it is and why it's used*, then use Claude Code to write and adapt the actual Python, SQL, and Terraform for it
- Applying this to real pipeline work: writing ingestion scripts, transformation SQL, validation logic, and infrastructure code with Claude Code assisting throughout
- Debugging real pipeline failures with AI assistance: reading logs, root-causing errors, verifying the fix instead of blindly accepting it

> **You'll learn to:** build the exact same capstone pipeline from Track 1, faster and more confidently, using AI as your pairing partner at every layer.

## 4. Agents & Subagents

3 items

- What an agent is, and when to delegate a task to one instead of doing it inline
- Specialized agents for research, code review, and multi-step implementation work
- Running work in parallel vs sequentially, and when each makes sense

> **You'll learn to:** break down a large engineering task and delegate pieces of it the way a tech lead delegates to a team.

## 5. MCP (Model Context Protocol)

2 items

- What MCP is and why it matters: connecting Claude Code to external tools and data (Jira, Confluence, GitHub, databases, cloud APIs)
- Using MCP-connected tools inside real engineering workflows: pulling tickets, reading docs, querying systems without leaving your editor

> **You'll learn to:** extend Claude Code beyond your local codebase into the real tools your team already uses.

## 6. Skills

2 items

- What a "Skill" is: a packaged, reusable set of instructions for a recurring task or workflow
- Using existing skills effectively, and recognizing when a repeated task should become one

> **You'll learn to:** turn your own repeated workflows (a deployment checklist, a code review pass, a report format) into something Claude Code can run consistently.

## 7. Memory

2 items

- How Claude Code remembers context about you, your project, and your preferences across sessions
- What's worth persisting (decisions, feedback, project context) vs what should stay one-off

> **You'll learn to:** work with an assistant that gets more useful over time instead of starting from zero every session.

## 8. Context Management

2 items

- How long sessions are handled, and why context degrades if it isn't managed
- Practical habits: scoping conversations, summarizing before switching tasks, keeping signal high in long engineering sessions

> **You'll learn to:** run long, complex engineering sessions (like building your capstone pipeline) without losing coherence halfway through.

## 9. Agentic Teams / Multi-Agent Workflows

2 items

- Orchestrating multiple agents toward one larger goal. This directly mirrors the pipeline orchestration concepts from Track 1 (dependencies, sequential vs parallel execution)
- Real scenarios: one agent researching an approach while another implements, or agents reviewing each other's work

> **You'll learn to:** think about AI-assisted work the same way you think about pipeline orchestration: as a system of coordinated tasks, not a single black box.

## 10. Claude Code for Career Visibility

3 items

- Using Claude Code to draft and refine resume bullets from your actual project work (Track 4)
- Drafting LinkedIn "About" sections, Featured post writeups, and outreach messages with AI assistance, then editing for your authentic voice
- Practicing mock interview answers and getting structured feedback

> **You'll learn to:** apply the same AI-assisted workflow to your job search that you use for engineering: faster iteration, still your own judgment in the driver's seat.

TRACK 4

# Career Visibility & Personal Branding

*Skills are necessary but not sufficient. This track makes sure the right people can find you and trust you can do the job.*

## 1. Resume Building

4 items

- Translating your capstone pipeline into resume bullets (impact + metrics, not task lists, e.g. "designed a fail-fast ETL pipeline processing X records/day" vs "used Cloud Run")
- ATS (Applicant Tracking System) formatting: what gets your resume rejected before a human sees it
- Tailoring the same resume for different roles (Data Engineer vs Analytics Engineer vs Cloud Engineer)
- Common resume mistakes GCP/data engineering candidates make (buzzword stuffing, no quantification, listing every tool ever touched)

> **You'll walk away with:** a resume built from your actual project, not a template, and defensible in any interview.

## 2. LinkedIn Profile Building

5 items

- Headline and "About" section that signal data engineer positioning (positioned, not "aspiring")
- Turning your capstone project into a LinkedIn Featured post with a real architecture diagram
- Skills section alignment with what recruiters actually search for
- Requesting and giving recommendations that carry weight
- Open-to-work signaling without looking desperate

> **You'll walk away with:** a profile that shows up in recruiter searches for "GCP Data Engineer" and converts profile views into conversations.

## 3. Community & Network Building

4 items

- Where data engineers actually gather: r/dataengineering, r/googlecloud, r/bigquery, r/cscareerquestions, GCP-focused Discord/Slack communities, local meetup groups
- How to participate without looking like you're farming for a job (answering questions, sharing what you built, genuine engagement before ever asking for anything)
- LinkedIn networking: connecting with alumni, recruiters, and engineers at target companies with a message that gets a reply
- Informational interviews: how to ask for 15 minutes without sounding like a favor

> **You'll walk away with:** a real network of people who've seen your work before you ever apply. This is the single biggest lever for beating the resume black hole.

## 4. Outreach & Referral Strategy

4 items

- Cold-messaging templates that get responses (specific, short, relevant, not "please refer me")
- Turning your network from section 3 into warm referrals
- Working with recruiters: what to say, what not to say, how to stay on their radar
- Tracking outreach so nothing falls through the cracks (simple spreadsheet system)

> **You'll walk away with:** an active pipeline of warm conversations, not just cold applications into a portal.

## 5. Interview Readiness

3 items

- Technical interview prep specific to the topics in Tracks 1 & 2 (SQL, architecture decisions, "walk me through a project")
- Behavioral interview prep (STAR method, framed around your capstone project)
- Mock interview practice and feedback loops

> **You'll walk away with:** the ability to confidently narrate your own project end-to-end under interview pressure.

TRACK 5

# Program Outcomes

*What you walk away with once all four tracks are complete.*

## Skills You Will Have

10 items

- Designed and deployed a complete, production-style ETL pipeline on GCP, end to end
- Built serverless data-processing jobs using Cloud Run and Docker
- Modeled analytical data warehouses using industry-standard schema design
- Automated and orchestrated multi-stage pipelines using GCP-native tooling
- Applied IAM security and least-privilege principles the way real cloud teams do
- Managed infrastructure as code and shipped changes through a CI/CD pipeline
- Made and defended architecture decisions (Cloud Run vs Dataflow vs Dataproc, batch vs streaming)
- Used Claude Code as a daily engineering tool for writing, debugging, and validating real pipeline code with AI assistance
- Worked with agents, MCP-connected tools, Skills, memory, and context management to run complex engineering work effectively
- Applied multi-agent, orchestration-style thinking to AI-assisted workflows, not just to pipelines

## Roles You'll Be Ready For

4 items

- Data Engineer (GCP-focused)
- Cloud Data Engineer / ETL Developer
- Analytics Engineer (BigQuery-centric)
- Junior Data Platform / Data Infrastructure Engineer

## Professional Presence You Will Have Built

5 items

- A resume built from a real, defensible project, not a template
- A LinkedIn profile positioned to surface in recruiter searches and convert views into conversations
- An active network across data engineering communities, built through genuine participation, not cold spam
- A working outreach and referral pipeline instead of a pile of unanswered applications
- Resume bullets, LinkedIn copy, and outreach messages refined with Claude Code: polished, but still authentically yours

## Interview & Career Readiness

5 items

- Core SQL and Python competency aligned to what interviews actually test
- A real, explainable project architecture to discuss in interviews (not a tutorial clone)
- Understanding of production concerns (cost optimization, failure handling, security) that most bootcamp graduates lack
- Practiced technical and behavioral interview delivery, narrating your own project under pressure
- Fluency with AI-assisted engineering tools: a fast-growing expectation in data engineering job descriptions, not just a nice-to-have