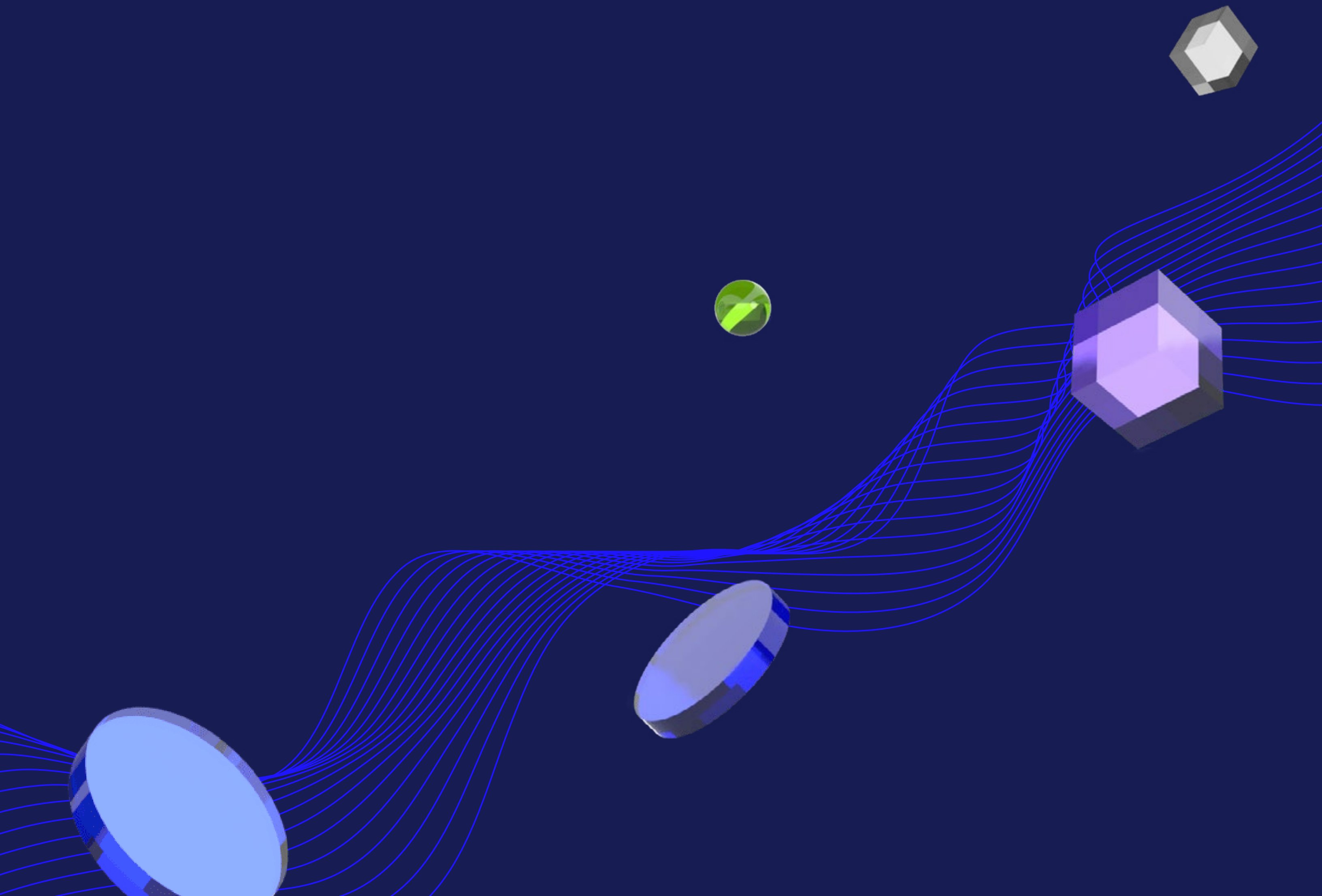
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF DATA SCIENCE

Data Engineering with AWS

Nanodegree Program Syllabus



Overview

Learn to design data models, build data warehouses and data lakes, automate data pipelines, and manage massive datasets.

Learning Objectives

Students will learn to:

- Create user-friendly relational and NoSQL data models.
- Create scalable and efficient data warehouses.
- Work efficiently with massive datasets.
- Build and interact with a cloud-based data lake.
- Automate and monitor data pipelines.
- Develop proficiency in Spark, Airflow, and AWS tools.

Program information

Estimated Time

4 months

Skill Level

Intermediate

Prerequisites

It is recommended that learners have intermediate Python, intermediate SQL, and command line skills.

Required Hardware/Software

There are no software and version requirements to complete this Nanodegree program. All coursework and projects can be completed via Student Workspaces in the Udacity online classroom. Udacity's basic tech requirements can be found at https://www.udacity.com/tech/requirements.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Data Modeling

Learners will create relational and NoSQL data models to fit the diverse needs of data consumers. They'll also use ETL to build databases in Apache Cassandra.

Course Project

Data Modeling with Apache Cassandra

Model event data to create a non-relational database and ETL pipeline for a music streaming app. Learners will define queries and tables for a database built using Apache Cassandra.

Lesson 1

Introduction to Data Modeling

- Understand the purpose of data modeling.
- Identify the strengths and weaknesses of different types of databases and data storage techniques.
- Create a table in Apache Cassandra.

Lesson 2

Relational Data Models

- Understand when to use a relational database.
- Understand the difference between OLAP and OLTP databases.
- Create normalized data tables.
- Implement denormalized schemas (e.g. STAR, Snowflake).

Lesson 3

NoSQL Data Models

- Understand when to use NoSQL databases and how they differ from relational databases.
- Select the appropriate primary key and clustering columns for a given use case.
- Create a NoSQL database in Apache Cassandra.

Course 2

Cloud Data Warehouses

In this course, learners will create cloud-based data warehouses. They will sharpen their data warehousing skills, deepen their understanding of data infrastructure, and be introduced to data engineering on the cloud using Amazon Web Services (AWS).

Course Project

Data Warehouse

In this project, learners will act as a data engineer for a streaming music service. They are tasked with building an ELT pipeline that extracts data from S3, stages it in Redshift, and transforms it into a set of dimensional tables for an analytics team to find insights into what songs their users are listening to.

Lesson 1

Introduction to Data Warehouses

- Explain how OLAP may support certain business users better than OLTP.
- Implement ETL for OLAP Transformations with SQL.
- Describe Data Warehouse Architecture.
- Describe OLAP cube from facts and dimensions to slice, dice, roll-up, and drill down operations.
- Implement OLAP cubes from facts and dimensions to slice, dice, roll-up, and drill down.
- Compare columnar vs. row-oriented approaches.
- Implement columnar vs. row-oriented approaches.

Lesson 2

ELT and Data Warehouse Technology in the Cloud

- Explain the differences between ETL and ELT.
- Differentiate scenarios where ELT is preferred over ETL.
- Implement ETL for OLAP Transformations with SQL.
- Select appropriate cloud data storage solutions.
- Select appropriate cloud pipeline solutions.
- Select appropriate cloud data warehouse solutions.

Lesson 3

AWS Data Technologies

- Describe AWS data warehouse services and technologies.
- Create and configure AWS Storage Resources.
- Create and configure Amazon Redshift resources.
- Implement infrastructure as code for Redshift on AWS.

Lesson 4

Implementing Data Warehouses on AWS

- Describe Redshift data warehouse architecture.
- Run ETL process to extract data from AWS S3 into Redshift.
- Design optimized tables by selecting appropriate distribution styles and sorting keys.

Course 3

Spark & Data Lakes

Learners will build a data lake on AWS and a data catalog following the principles of data lakehouse architecture. They will learn about the big data ecosystem and the power of Apache Spark for data wrangling and transformation. They'll work with AWS data tools and services to extract, load, process, query, and transform semi-structured data in data lakes.

Course Project

STEDI Human Balance Analytics

In this project, learners will act as a data engineer for the STEDI team to build a data lakehouse solution for sensor data that trains a machine learning model. They will build an ELT (Extract, Load, Transform) pipeline for lakehouse architecture, load data from an AWS S3 data lake, process the data into analytics tables using Spark and AWS Glue, and load them back into lakehouse architecture.

Lesson 1

Big Data Ecosystem, Data Lakes, & Spark

- Identify what constitutes the big data ecosystem for data engineering.
- Explain the purpose and evolution of data lakes in the big data ecosystem.
- Compare the Spark framework with Hadoop framework.
- Identify when to use Spark and when not to use it.
- Describe the features of lakehouse architecture.

Lesson 2

Spark Essentials

- Wrangle data with Spark and functional programming to scale across distributed systems.
- Process data with Spark DataFrames and Spark SQL.
- Process data in common formats such as CSV and JSON.
- Use the Spark RDDs API to wrangle data.
- Transform and filter data with Spark.

Lesson 3

Using Spark & Data Lakes in the AWS Cloud

- Use distributed data storage with Amazon S3.
- Identify properties of AWS S3 data lakes.
- Identify service options for using Spark in AWS.
- Configure AWS Glue.
- Create and run Spark Jobs with AWS Glue.

Lesson 4

Ingesting & organizing data in lakehouse architecture on AWS

- Use Spark with AWS Glue to run ELT processes on data of diverse sources, structures, and vintages in lakehouse architecture.
- Create a Glue Data Catalog and Glue Tables.
- Use AWS Athena for ad-hoc queries in a lakehouse.
- Leverage Glue for SQL AWS S3 queries and ELT.
- Ingest data into lakehouse zones.
- Transform and filter data into curated lakehouse zones with Spark and AWS Glue.
- Join and process data into lakehouse zones with Spark and AWS Glue.

Course 4

Automate Data Pipelines

In this course, learners will dive into the concept of data pipelines and how they can use them to accelerate their career as a data engineer. This course will focus on applying the data pipeline concepts students will learn through an open-source tool from Airbnb called Apache Airflow. This course will start by covering concepts including data validation, DAGs, and Airflow. We'll then venture into AWS quality concepts like copying S3 data, connections and hooks, and Redshift Serverless. Next, learners will explore data quality through data lineage, data pipeline schedules, and data partitioning. Finally, they'll put data pipelines into production by extending Airflow with plugins, implementing task boundaries, and refactoring DAGs.

Course Project

Data Pipelines with Airflow

In this project, learners will work to build high grade data pipelines from reusable tasks that can be monitored and provide easy backfills for a music streaming company, Sparkify. They will move JSON logs of user activity and JSON metadata data from S3 and process it in Sparkify's data warehouse in Amazon Redshift. To complete the project, learners will need to create their own custom operators to perform tasks such as staging the data, filling the data warehouse, and running checks on the data as the final step.

Lesson 1

Data Pipelines

- Define and describe a data pipeline and its usage.
- Explain the relationship between DAGs, S3, and Redshift within a given example.
- Employ tasks as instantiated operators.
- Organize task dependencies based on logic flow.
- Apply templating in codebase with kwargs parameter to set runtime variables.

Lesson 2

Airflow & AWS

- Create Airflow Connection to AWS using AWS credentials.
- Create Postgres/Redshift Airflow Connections.
- Leverage hooks to use Connections in DAGs.
- Connect S3 to a Redshift DAG programmatically.

Lesson 3

Data Quality

- Utilize the logic flow of task dependencies to investigate potential errors within data lineage.
- Leverage Airflow catchup to backfill data.
- Extract data from a specific time range by employing the kwargs parameters.
- Create a task to ensure data quality within select tables.

Lesson 4

Production Data Pipelines

- Consolidate repeated code into operator plugins.
- Refactor a complex task into multiple tasks with separate SQL statements.
- Convert an Airflow 1 DAG into an Airflow 2 DAG.
- Construct a DAG and custom operator end-to-end.

Meet your instructors.

Amanda Moran

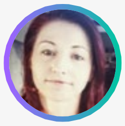
Developer Advocate at DataStax

Amanda is a developer advocate for DataStax after spending the last 6 years as a software engineer on 4 different distributed databases. Her passion is bridging the gap between customers and engineering. She has degrees from the University of Washington and Santa Clara University.

Ben Goldberg

Staff Engineer at SpotHero

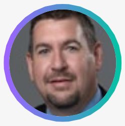
In his career as an engineer, Ben Goldberg has worked in fields ranging from computer vision to natural language processing. At SpotHero, he founded and built out their data engineering team, using Airflow as one of the key technologies.



Valerie Scarlata

Curriculum Manager at Udacity

Valerie is a curriculum manager at Udacity who has developed and taught a broad range of computing curriculum for several colleges and universities. She was a professor and software engineer for over 10 years specializing in web, mobile, voice assistant, and social full-stack application development.



Matt Swaffer

Solutions Architect

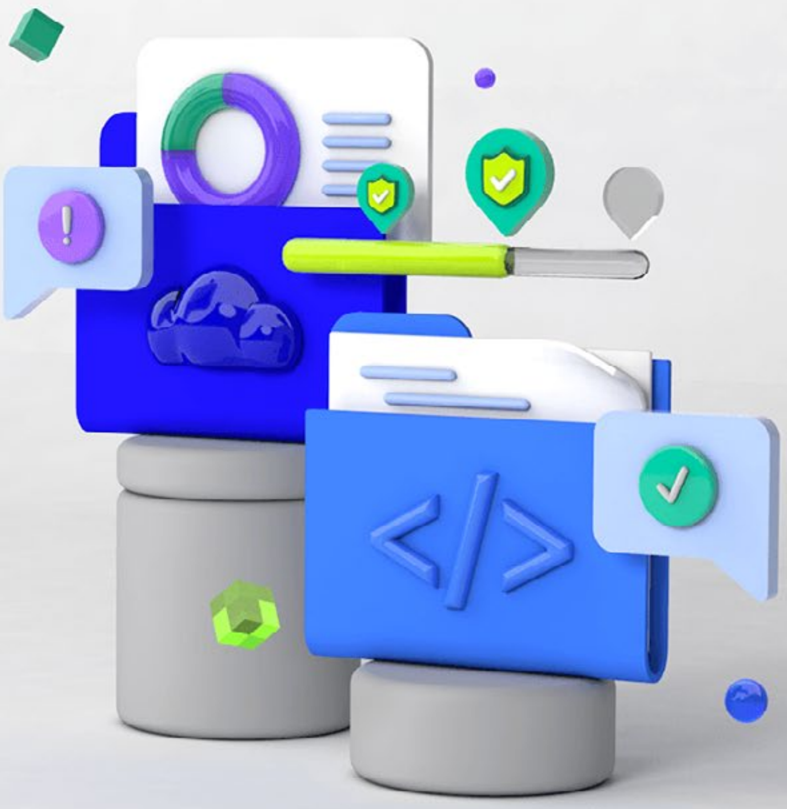
Matt is a software and solutions architect focusing on data science and analytics for managed business solutions. In addition, Matt is an adjunct lecturer, teaching courses in the computer information systems department at the University of Northern Colorado where he received his PhD in educational psychology.

Sean Murdock

Professor at Brigham Young University Idaho

Sean currently teaches cybersecurity and DevOps courses at Brigham Young University Idaho. He has been a software engineer for over 16 years. Some of the most exciting projects he has worked on involved data pipelines for DNA processing and vehicle telematics.

Udacity’s learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren’t just multiple choice questions or step-by-step guides, but instead require critical thinking.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

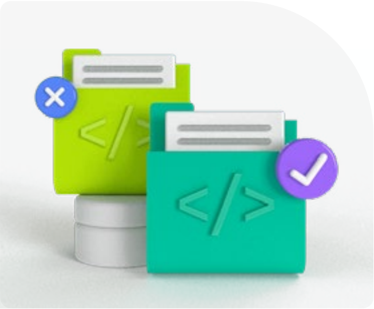
Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

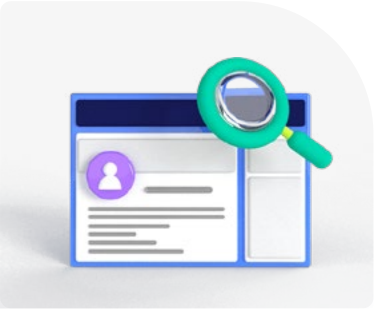
Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →

11.28.22 | V1.0