“**Build with Us | Deep Dive: Building Your First Pipeline**”是发布在 learn.palantir.com 上的一门深度实战课程，由 **Ontologize** 团队（由前 Palantir 工程师组成）提供 1。该课程旨在指导用户如何在 Palantir Foundry 中通过核心工作流，从零开始构建、部署并管理一个完整的数据生产管道 1。
以下是基于提供的来源对该课程内容的详细介绍：

1. 核心概念与管道定义

课程首先明确了 Foundry 中“管道（Pipeline）”的基础定义：

- 定义：管道是一系列输入数据集（或单个数据集），通过一系列**转换（Transformations）**处理后，生成一个或多个输出数据集的过程 2。
- 循环特性：输出数据集可以进一步作为下一个管道环节的输入，直到生成能够支撑最终业务流的数据集序列 2。
- 构建目的：管道通常用于清洗原始数据、为数据分析做准备、为本体（**Ontology**）对象类型提供支撑，或作为机器学习模型的输入 3。

2. 开发环境与准备工作

- 工具：课程的核心操作是在 **Pipeline Builder** 应用中完成的，这是一个用于构建数据管道的图形化/低代码环境 4, 5。
- 权限要求：用户需要拥有对应项目的 **Editor**（编辑） 角色，以便能够添加和管理 Pipeline Builder 构件 6。
- 建议存储位置：通常在名为 temporary training artifacts 或 tutorial practice artifacts 的公共训练项目下创建个人文件夹进行练习 3, 6。

3. 数据摄取与初始预处理

课程通过三种不同格式的原始数据（产品、客户、交易数据）演示了摄取流程：

- 支持格式：包括 **CSV**、**Parquet**（一种专为 Spark 引擎优化的列式存储格式）和 **JSON** 7。
- 数据清洗（**Cleaning**）：
- 去除空格（**Trim Whitespace**）：确保字段名称一致，防止在后续聚合分组时出现错误 8, 9。
- 类型转换（**Cast**）：将字符串类型的字段（如价格 Price 或单位 Units）转换为数值类型（如 Double），以便进行数学运算 8, 10, 11。
- 过滤（**Filter**）：移除包含空值（Null）或无效数据的行，以维护数据质量 12。

4. 高级转换与 AI 集成

课程涵盖了一些复杂的处理技巧：

- 处理结构化数据（**Structs**）：教授如何解析嵌套的 JSON 数据，使用 **Flatten struct**（展平结构体）和 **Concatenate**（拼接）功能将复杂的地址字段转换为易读的字符串 13, 14, 15。
- **AIP** 助手辅助开发：
- 自动生成转换逻辑：用户可以利用 **Foundry AI** 助手 通过自然语言生成复杂的转换逻辑，例如将具有特殊格式的字符串日期转换为标准的时间戳（Timestamp） 16, 11。
- 提示词构建：AI 助手可以根据用户提供的示例值自动推断并生成日期格式字符串，极大简化了开发难度 16。

5. 数据集成与逻辑优化

在单表清洗完成后，课程进入数据整合阶段：

- 多表连接（**Joins**）：通过 **Left Join**（左连接）将交易、产品和客户数据合并，构建全局视图 17, 18。
- 故障排除（**Troubleshooting**）：教授如何利用 **View Stats**（查看统计数据）功能识别“连接爆炸”或数据重复（Multiplicity）问题，并据此调整连接键（如将 product_id 更改为更精确的 product_variation_id） 19, 20, 21。
- 数据聚合（**Aggregations**）：执行类似 SQL 中 Group By 的操作，通过计算“单价 × 数量”得到收入，并按客户维度汇总生成**客户终生价值（Customer Lifetime Value）**数据集 22, 23。

6. 部署、监控与最佳实践

- 部署（**Deploy**）：在 Pipeline Builder 中，逻辑编写完成后必须点击“Deploy”才能在 Foundry 中实际运行并生成物化数据集 24。
- 作业追踪（**Job Tracker**）：课程引导用户通过 Job Tracker 监控后台 Spark 作业的执行进度和成功状态 25。
- 管道分段（**Pipeline Segmentation**）：作为最佳实践，建议将大型管道拆分为多个逻辑段（如原始层 Raw、清洗层 Clean、富化层 Enriched、本体层 Ontology），以便于维护和协作 26, 27。
- 数据血缘（**Data Lineage**）：通过血缘应用可视化展示从原始数据到最终分析应用的全过程，确保数据的可追溯性 27。

通过本教程，用户能够掌握在 Foundry 平台中构建高性能、受治理且可扩展的工业级数据管道所需的端到端技能 28。