“**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，由前 Palantir 工程师组成的 **Ontologize** 团队制作 1, 2。该课程的核心目标是指导用户在 Palantir Foundry 中利用 **Pipeline Builder** 应用，从零开始构建一个完整的数据生产管道 3, 4。
以下是基于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）”的定义：它是一系列输入数据集通过一系列转换（**Transformations**）最终生成输出数据集的过程 5。

- 构建目的：管道通常是为了使原始数据（Raw Data）变得可用，例如为分析做准备、支撑本体（Ontology）对象类型，或作为机器学习模型的输入 5, 6。
- 开发环境：主要在 **Pipeline Builder** 中进行，这是一个低代码/图形化的开发环境 3, 4。

2. 数据准备与清洗流程 (Preprocessing)

课程通过三种不同格式的原始数据（产品 CSV/Parquet、客户结构化数据、交易 JSON）演示了数据清洗的最佳实践：

- 基础转换：包括修剪字符串空格（Trim Whitespace）以确保聚合准确，以及将数据类型（如价格）从字符串转换为双精度浮点型（Double） 7-10。
- 复杂结构处理：教授如何处理“结构体（Struct）”数据 11。例如，将嵌套的 JSON 地址信息展平（Flatten），解析后重新拼接成易读的街道/城市/州字符串，并删除冗余的中间列 12-16。
- **AI** 辅助功能：展示了如何利用 **Foundry AI** 助手 自动生成复杂的日期转换格式字符串（Timestamp Casting），减少了查阅技术文档的负担 17-20。

3. 数据集成与逻辑优化 (Joins & Aggregations)

在清洗完各分支数据后，课程进入了数据整合阶段：

- 多表连接 **(Joins)**：将交易记录、产品信息和客户资料通过左连接（Left Join）合并 21, 22。
- 故障排除与调试：课程特别强调了如何使用 **View Stats**（查看统计数据）功能来识别数据重复（Multiplicity）问题 23, 24。例如，当发现 Join 导致行数非预期增加时，引导用户将连接键从 product_id 优化为更精确的 product_variation_id 24, 25。
- 高级聚合：通过数学运算（单价 × 数量）计算收入（Revenue），并按客户进行分组聚合，最终生成客户终生价值 **(Customer Lifetime Value)** 数据集 26, 27。

4. 部署、监控与最佳实践

课程不仅涵盖了逻辑编写，还包括了管道的运维管理：

- 部署 **(Deployment)**：逻辑编写完成后，必须通过“部署”操作才能在 Foundry 中物化（Materialize）出实际的数据集 28, 29。
- 作业追踪 **(Job Tracker)**：介绍如何使用 Job Tracker 查看后台 Spark 作业的执行进度和详细细节 29。
- 管道分段 **(Segmentation)**：讲解了将大型管道拆分为多个逻辑段（如原始层、清洗层、富化层、本体层）的最佳实践，以便于维护和复用 30, 31。
- 数据血缘 **(Data Lineage)**：通过血缘视图提供管道的端到端可视化，确保数据的来源和去向清晰透明 31, 32。

5. 学习建议

- 权限准备：用户需要拥有 Foundry 项目的“编辑（Editor）”权限，并能访问 Pipeline Builder 应用 33。
- 实验环境：建议在“临时训练工件（Temporary Training Artifacts）”或特定的教学文件夹中进行操作，以保持环境整洁 4, 6, 33。

通过学习该课程，用户不仅能掌握 **Pipeline Builder** 的功能，还能学会如何构建一个高性能、受治理且可扩展的工业级数据管道 32。