

Программа курса

«DAPL: Платформа данных: карта, компоненты, решения»

О курсе: За 5 дней обучения на курсе, мы рассмотрим реальные платформы данных, разберем их по компонентам, сравним подходы и решения и обсудим, что за каждым из них стоит. Вы увидите не одну «правильную» архитектуру, а несколько — и научитесь понимать, почему они разные.

Аудитория:

- Системным и бизнес-аналитикам, менеджерам и владельцам продуктов
- Data-инженерам и разработчикам
- Разработчикам сервисов и приложений
- Техническим руководителям и тимлидам

Уровень подготовки:

Достаточно базового опыта работы с данными или ИТ-системами: понимать, что такое реляционная СУБД, читать простые SQL-запросы и представлять, как приложения, базы и интеграции связаны между собой, а опыт построения платформ и знание конкретных инструментов не нужны.

Продолжительность курса: 24 академических часа, 5 дней

Содержание программы

1. Платформа данных: что стоит за термином

- Почему «платформа данных» — это не синоним DWH, Data Lake или BI-системы
- Что общего у всех платформ данных, как бы они ни выглядели
- Смежные понятия: где заканчивается платформа и начинается инфраструктура, интеграция или аналитика
- Кто и какие задачи решает с помощью платформы данных
- Почему один и тот же термин означает разное у вендоров, консультантов и внутри компаний
- Рабочее определение, которым будем пользоваться весь курс

2. Как мы к этому пришли: три поколения платформ данных

- Почему платформы данных вообще появились: от разделения OLTP и OLAP до идеи единого аналитического контура
- Первое поколение: классическое хранилище данных — централизация, схема «на входе», дорогое железо
- Второе поколение: Data Lake и эра больших данных — гибкость, дешёвое хранение, но потеря смысла
- Третье поколение: Lakehouse и облачные платформы — попытка взять лучшее из двух миров
- Что именно менялось от поколения к поколению: хранение, обработка, модель данных, роли
- Почему понимание поколений помогает читать любую современную архитектуру

3. Карта платформы: семь слоёв

- Семь слоёв платформы данных: источники, приём, хранение, обработка, моделирование, управление, потребление
- Почему «слои» — это не отделы и не команды, а роли, которые могут исполняться разными инструментами
- Компоненты, которые есть почти в любой платформе, и компоненты, которые появляются не всегда
- Как компоненты связаны между собой: поток данных от источника до потребителя
- Типовые архитектурные развилки: batch или streaming, централизация или федерация, облако или on-premise
- Как пользоваться этой картой: три вопроса, которые стоит задать к любой платформе

4. Источники данных и приём

- СУБД, файлы, API, потоковые данные — основные типы источников и их характер
- Чем источники отличаются друг от друга: объём, скорость, структура, надёжность, доступность

- Batch и streaming: почему выбор неочевиден
- Интеграционные паттерны: pull и push, CDC, очереди, шины данных
- Источник как архитектурное ограничение на всю платформу
- Типовые проблемы на стороне источников и почему это архитектурная, а не операционная тема
- Ориентиры по инструментам: Kafka, NiFi, CDC-инструменты

5. Хранение данных и форматы

- Реляционные СУБД и MPP-системы: классика, которая никуда не ушла
- Колоночные хранилища и почему аналитика любит колонки
- Объектные хранилища: дешево, масштабируемо, но не «просто диск»
- Открытые табличные форматы — Iceberg, Delta, Hudi, Paimon: почему формат стал архитектурным решением
- Что такое «открытость» формата и почему вокруг неё столько споров
- Как выбрать хранилище под задачу: критерии, которые действительно важны
- Ориентиры по инструментам: Greenplum, ClickHouse, StarRocks, Trino

6. Обработка данных: от ETL к compute

- ETL, ELT и compute: три парадигмы обработки и логика их смены
- Почему обработка «переехала» внутрь хранилища — и что это изменило
- Batch, streaming и micro-batch: три режима обработки и их место в платформе
- Оркестрация: как управлять цепочками обработки, когда их становятся сотни
- Почему разделение хранения и вычислений изменило облик платформы
- Что значит «compute» сегодня: движки, мощности, стоимость
- Ориентиры по инструментам: Spark, Flink, Airflow

7. Моделирование данных: как данные превращаются в смысл

- Зачем моделировать данные: от «просто таблиц» к управляемому смыслу
- Кимбалл: размерное моделирование — и почему оно стало стандартом де-факто
- Инмон: корпоративная модель — и почему она не прижилась
- Data Vault и Anchor Modeling: ответ на ограничения предыдущих подходов
- OBT: почему «одна большая таблица» снова стала вариантом
- Что выбирают сегодня — и почему единого ответа нет

8. Управление платформой: метаданные, качество, безопасность

- Каталог и метаданные: что это даёт и почему внедрения буксуют
- Lineage: откуда данные и куда они ушли
- Качество данных и data observability
- Data contracts: договорённости между источником и потребителем
- Доступы, безопасность, соответствие требованиям
- Роли и ответственность: владельцы данных, стюарды

9. Потребление данных

- BI и самообслуживание: что требуется от платформы
- Семантический слой: единые метрики и определения
- Отдача данных приложениям: API, reverse ETL, витрины как сервис
- Данные для ML и AI: feature-хранилища, подготовка данных, AI-ready платформа — что в этом реально, а что лозунг
- Data as a product: что это значит на практике

10. Современные тренды: Lakehouse, Data Mesh, Data Fabric

- Lakehouse: попытка примирить хранилище и озеро — что получилось, а что нет
- Data Mesh: организационный сдвиг, а не технология — почему это важно понимать
- Data Fabric: метаданные, каталог и автоматизация — где здесь суть, а где маркетинг
- Bring Your Own Compute: вычисления приходят в вашу среду
- Возвращение одной машины: DuckDB и Polars

 ШКОЛА БОЛЬШИХ ДАННЫХ	ООО «Учебный центр «Коммерсант» ©	
	«Школа Больших Данных» www.bigdataschool.ru 2026	

- AI-ready Data Platform, семантический слой и другие
- Как тренды соотносятся друг с другом: конкуренты, дополнения или разные уровни разговора
- Что стоит за модными словами: реальные проблемы, на которые отвечает каждый тренд
- Как отличить содержательный тренд от маркетингового: несколько простых критериев
- Что может прийти на смену — и как к этому относиться