Программа курса «STAR: Проектирование и разработка Online-хранилищ данных на базе StarRocks»

О курсе: Курс посвящен современным подходам к построению on-line аналитических платформ, обеспечивающих доступность данных для аналитики с минимальной задержкой после их возникновения в оперативных системах. Слушатели изучат Lambda и Kappa архитектуру, технологии потоковой обработки данных, методы проектирования аналитических моделей и современные аналитические СУБД, ориентированные на online-аналитику.

Аудитория:

- Data Engineer
- BI Engineer
- Data Architect
- Backend-разработчики
- Системные архитекторы
- Руководители аналитических платформ

Уровень подготовки:

- SQL, основы реляционных БД,
- Базовые знания в области аналитических систем

Продолжительность курса: 24 академических часа, 5 дней

Содержание программы

1. Онлайн-хранилища данных и требования Real-Time Analytics

- эволюция аналитических платформ: от пакетного DWH к Near Real-Time Analytics; различия OLTP и OLAP-нагрузок;
- пакетная и потоковая обработка данных;
- требования бизнеса к задержке, полноте, точности и доступности данных;
- компромисс между низкой задержкой и сложностью архитектуры;
- место онлайн-хранилища в современной платформе данных на примерах **StarRocks** и **ClickHouse**

2. Событийная архитектура, Lambda и Kappa

- событие как базовая единица обмена данными;
- Producer, Consumer, Event Broker и модель Publish/Subscribe;
- распределенный журнал событий;
- Batch Layer, Speed Layer и Serving Layer в Lambda Architecture;
- Преимущества и ограничения Lambda Architecture; поток как источник истины и повторная обработка данных в Kappa Architecture;
- сравнение Lambda и Kappa для корпоративных сценариев.
    - **Лабораторная работа:** разработка событийной модели интернет-магазина и высокоуровневой архитектуры доставки данных. Выбор Lambda- или Kappa-подхода с обоснованием решения.

3. Моделирование данных для онлайн-аналитики

- денормализация и ее влияние на скорость запросов и сопровождение данных;
- принцип «рассчитать один раз — использовать многократно»;
- факты и измерения;
- схемы «звезда» и «снежинка»;
- ведение истории изменений и Slowly Changing Dimensions;

- применение SCD Type 2 в потоковых сценариях;
- выбор модели с учетом частых обновлений и JOIN-запросов.
    - **Лабораторная работа:** проектирование аналитической модели интернет-магазина с таблицами фактов, измерениями и историей изменений.

4. Apache Kafka в аналитической платформе

- архитектура Kafka и роль брокера событий;
- Producer и Consumer;
- топики, партиции и consumer groups;
- репликация и отказоустойчивость;
- ключ события и порядок обработки; форматы JSON, Avro и Protobuf;
- Schema Registry и эволюция схем сообщений;
- KTable и представление изменяемого состояния в потоковой архитектуре.
    - **Лабораторная работа:** проектирование топиков, ключей, партиционирования и контрактов событий для интернет-магазина

5. Change Data Capture: доставка изменений из PostgreSQL

- журнал транзакций как источник изменений;
- CDC и его место в архитектуре онлайн-хранилища;
- Kafka Connect и Debezium;
- захват операций INSERT, UPDATE и DELETE из PostgreSQL;
- первичная загрузка и последующая потоковая синхронизация;
- порядок событий, дублирование и повторная обработка;
- изменение схемы источника и контроль качества данных.
    - **Лабораторная работа:** настройка Kafka Connect и Debezium для захвата изменений из PostgreSQL и публикации событий в Kafka

6. Потоковая обработка данных в Apache Flink

- архитектура Apache Flink;
- DataStream API, Table API и Flink SQL;
- Processing Time и Event Time;
- Watermarks и обработка опоздавших событий;
- оконные функции;
- Stateful Processing;
- Checkpoints и восстановление состояния;
- семантика exactly-once;
- потоковые JOIN и агрегации при подготовке витрин.
    - **Лабораторная работа:** потоковая агрегация событий и подготовка аналитической витрины интернет-магазина в Apache Flink.

7. Аналитические СУБД для онлайн-аналитики

- MPP-архитектура и параллельное выполнение запросов;
- особенности колоночного хранения;
- требования к СУБД для потоковых обновлений и интерактивной аналитики;
- изменяемые данные, предварительные вычисления и JOIN-запросы;
- сравнение StarRocks и ClickHouse по модели данных, способам обновления, JOIN, загрузке и типовым сценариям применения;
- критерии выбора аналитической СУБД под требования проекта.

8. StarRocks под капотом

- архитектура Frontend и Backend: роли FE, BE и CN;
- shared-nothing и shared-data режимы развертывания;
- Vectorized Execution Engine и Pipeline Engine;

- Cost-Based Optimizer и статистика;
- модели таблиц StarRocks;
- Primary Key Tables и реализация операций UPSERT, DELETE и частичного обновления;
- партиционирование, распределение данных и бакеты;
- индексы и data skipping;
- алгоритмы JOIN и Runtime Filters;
- синхронные и асинхронные материализованные представления;
- выполнение и анализ плана запроса с EXPLAIN.
    - **Лабораторная работа:** развертывание учебного кластера, создание таблиц разных моделей и исследование планов JOIN-запросов. Сравнение вариантов распределения данных и оптимизация запроса.

9. Загрузка данных в StarRocks

- способы пакетной и потоковой загрузки;
- Stream Load для загрузки файлов и микропакетов;
- Routine Load для непрерывного чтения данных из Kafka;
- Kafka Connector для интеграции с Kafka Connect и обработки CDC-событий;
- Flink Connector для записи результатов потоковой обработки;
- начальная пакетная загрузка и переключение на поток изменений;
- контроль состояния задач загрузки, ошибок и задержки;
- применение UPSERT и DELETE при загрузке в Primary Key Tables.
    - **Лабораторная работа:** настройка непрерывной загрузки событий из Apache Kafka и результатов обработки Apache Flink в аналитическое хранилище StarRocks.

10. Итоговый практический проект

Вы завершите сквозной кейс и соберете работающий прототип онлайн-хранилища данных для интернет-магазина:

- выполните начальную пакетную загрузку данных из PostgreSQL;
- настроите захват последующих изменений через CDC;
- организуете доставку событий через Kafka;
- при необходимости преобразуете и агрегируете потоки во Flink;
- загрузите изменения в Primary Key Tables и аналитические витрины StarRocks;
- проверите корректность UPSERT и DELETE;
- измерите задержку обновления данных;
- выполните аналитические запросы с JOIN и проанализируете план выполнения.
    - **Лабораторная работа:** сборка и проверка полного конвейера PostgreSQL → Debezium → Kafka → Flink → StarRocks.

Результат проекта: данные в витринах StarRocks обновляются с минимальной задержкой относительно их появления в PostgreSQL или Kafka и доступны для интерактивной SQL-аналитики.