ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К СИСТЕМЕ РЕКОМЕНДАЦЙ ДЛЯ СТРИМЕНГОВОГО СЕРВИСА

РЕДИНОВ дМИТРИЙ мАКСИМОВИЧ

Минаева Мирра Дмитриевна

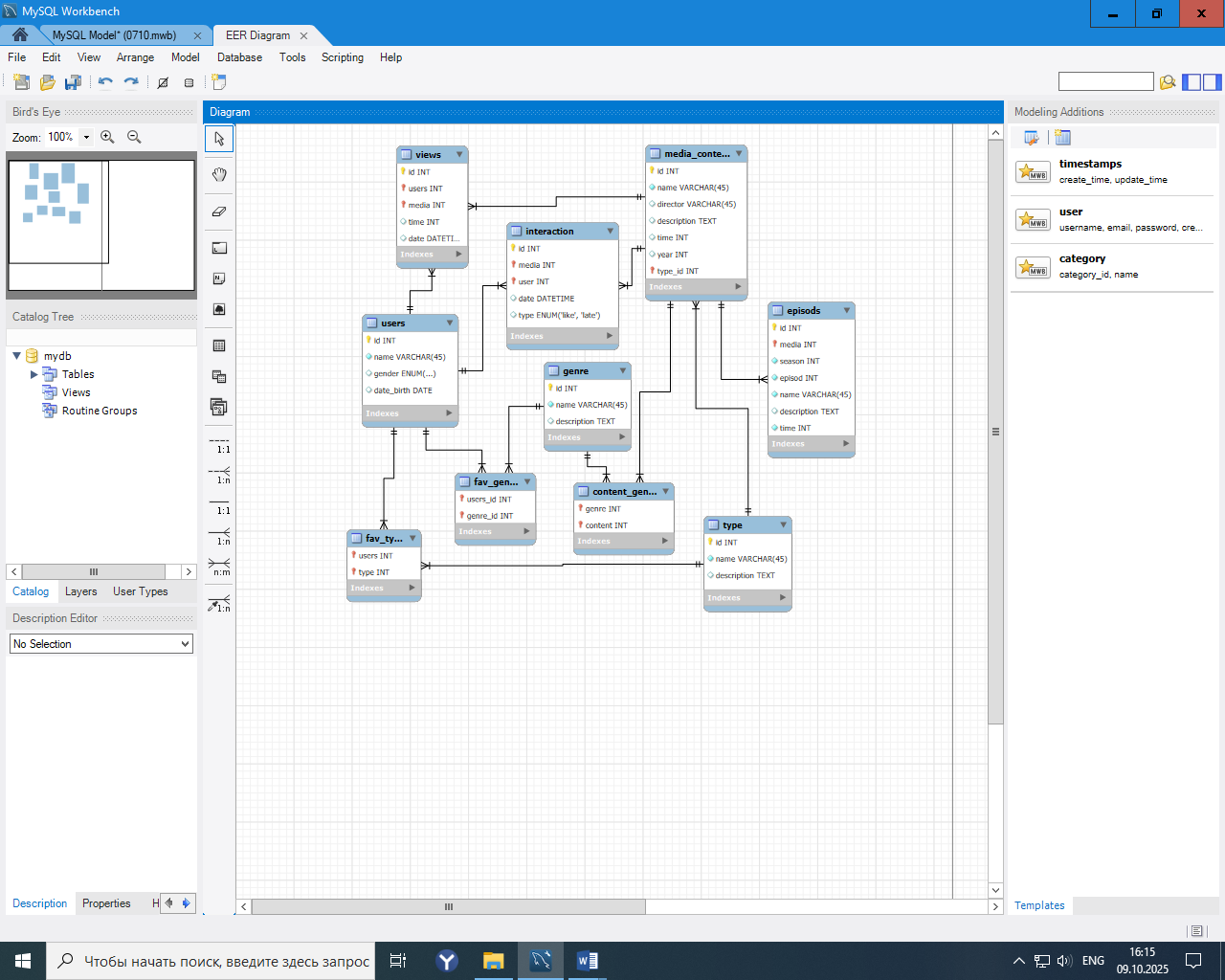
**1. Общее описание:**

Данная система предназначена для обеспечения функционала персонализированных рекомендаций контента в стриминговом сервисе. Её основная бизнес-цель - повышение пользовательской вовлеченности и удержания за счет своевременного предложения релевантного кино- и видеоконтента, соответствующего интересам и поведению каждого конкретного пользователя.

Ключевые технологические цели системы:

* Сбор и надежное хранение в реальном времени сырых данных о взаимодействиях пользователей с платформой (просмотры, лайки, продолжительность).
* Эффективная обработка больших объемов накопленных данных для выявления паттернов поведения и построения матрицы предпочтений "пользователь-контент".
* Сверхбыстрая выдача актуальных персонализированных рекомендаций в ответ на действия пользователя в интерфейсе сервиса.

**2. Логическая модель данных:**



**3. Выбор СУБД и обоснование:**

**3.1 Оценка нагрузки:**

Предполагаемое количество пользователей. Общее количество людей, зарегистрированных на сервисе 2,6. Пиковая одновременная нагрузка: В вечерние часы активны до 40% пользователей, а именно 1 040 000 пользователей начинают и заканчивают просмотр. Как минимум 40% (416 000) из них как-либо взаимодействуют с контентом (оставляют “лайки”, или добавляют в список “смотреть позже”).

Объем данных и их рост. За один вечерний сеанс пользователь может посмотреть 3-5 единиц контента (фильм, серии). Для каждого просмотра генерируется 1 событие: "просмотр". Допустим пользователь взаимодействует с каждым просмотренным контентом, и того 10 событий на пользователя в час в пик. Пиковая скорость записи: 416 000 пользователей \* 10 событий/час = 4 160 000 событий в час. Это примерно 1 155 событий в секунду. В сутки: 4 млн событий/час \* 4 пиковых часа + (оставшиеся 20 часов с низкой нагрузкой, допустим, 1/5 от пика) = 16 млн + 16 млн = 32 миллиона событий в день. В месяц: 32 млн \* 30 дней = 960 млн. В год: примерно11 миллиардов событий.

Типы и частота операций. Операции записи происходят постоянным потоком по 1 тысячи RPS в пике. Чтение происходит при выводе персонализированной подборки для пользователя. Примерное соотношение 85% записей и 15% чтений. Пиковые нагрузки в вечернее время и выходные дни.

**3.2 Выбор СУБД:**

Выбор: Гибридный подход.

Для хранения сырых событий и аналитики: PostgreSQL. Это объектно-реляционная СУБД. Позволяет гибко управлять базами данных и идеально подходит для сбора данных и сложной аналитики.

Для хранения готовых рекомендаций и быстрого доступа: Redis. Нереляционная система управления базами данных. В отличие от реляционных СУБД, работает со структурами типа «ключ-значение». Что хорошо подходит для хранения рекомендованного контента в виде списка для каждого пользователя.

Архитектура потока данных:

* Пользователь взаимодействует с контентом через веб-сервис
* Запись о взаимодействии заноситься в PostgreSQL
* На уровне PostgesSQL проводиться аналитика.
* На основе аналитики формируется список рекомендаций и заноситься в Redis
* Рекомендации выводятся пользователю через веб-сервис

**4. Нефункциональные требования:**

Производительность:

* + Запись событий: 95% запросов на вставку в PostgreSQL должны выполняться < 10 мс.
  + Чтение рекомендаций: 99% запросов к Redis на получение списка рекомендаций для пользователя должны выполняться < 5 мс.
  + Обучение модели: Пакетный пересчет рекомендаций должен завершаться не более чем за 4 часа.

Доступность:

* + Цель: 99.9% (порядка 8-9 часов простоя в год). Сервис рекомендаций не является абсолютно критическим для самого просмотра (стриминг может работать без него), но его недоступность напрямую влияет на вовлеченность и удержание пользователей.

**5. Аппаратные требования.**

Для Сервера PostgreSQL:

* CPU: Много ядер для параллельных запросов. Параллельные операции ввода-вывода при потоковой записи, а также выполнение тяжелых аналитических запросов при обучении моделей.
* RAM: Большой объем для кэширования БД. Чтобы максимально ускорить аналитические запросы, необходимо закэшировать в оперативной памяти "горячие" данные. Цель — чтобы рабочий набор данных помещался в RAM. Для 500 ГБ данных нужно не менее 128-256 ГБ RAM.
* Disk: Высокая скорость (IOPS) для операций ввода-вывода. Постоянная запись потока событий и интенсивное чтение при аналитике требуют очень быстрого диска. NVMe SSD обязательны.

Для Сервера Redis:

* CPU: Высокая тактовая частота для сложных вычислений (одного ядра). Redis в основном однопоточен. Важна высокая производительность одного ядра для быстрой обработки каждого запроса.
* RAM: Большой объем для хранения всех данных. Все данные Redis должны полностью помещаться в оперативной памяти. Рассчитаем: 5 млн пользователей \* (1 ключ user\_id + 100 рекомендованных content\_id \* 8 байт) = 4 ГБ. Плюс накладные расходы. 16-32 ГБ RAM.
* Disk: Скорость не критична, надежность — да. Высокие IOPS не требуются, подойдет стандартный SSD. Надежность обеспечивается репликацией на другой сервер, а не диском.

**6. Требования к резервному копированию и восстановлению:**

Для PostgreSQL (основного хранилища событий)

RPO: 15-30 минут. Потеря даже 30 минут данных о просмотрах в пиковый вечер — это тысячи неучтенных взаимодействий, которые повлияют на качество будущих рекомендаций

RTO: 2-4 часа. Система рекомендаций важна, но не является абсолютно критической для функционала стриминга. За 2-4 часа можно восстановить данные и проверить целостность.

Частота:

* + Ежедневные полные бэкапы (Full Backups): Выполняются раз в сутки, в период наименьшей нагрузки.
  + Непрерывное WAL-архивирование (Write-Ahead Logging): Это ключевая функция PostgreSQL. Все изменения записываются в WAL-файлы. Эти файлы постоянно архивируются в удаленное хранилище. Это позволяет достичь RPO в 15-30 минут.

Тип бэкапов:

* Физические бэкапы (Physical Backups): Копия бинарных файлов данных PostgreSQL. Восстановление происходит быстрее. Идеально в паре с WAL-архивированием для Point-in-Time Recovery (PITR) — восстановления на любой конкретный момент времени в пределах архивированных WAL-файлов.

Для Redis (списка рекомендаций)

RPO: 6-12 часов. Потеря кэша означает, что какое-то время рекомендации будут нерелевантными. Допустима потеря данных за полдня, так как основной массив данных цел в PostgreSQL.

RTO: 15-30 минут. Восстановление Redis должно быть очень быстрым. Цель — минимизировать время, когда пользователи получают некачественные рекомендации. Процесс восстановления простой.

Частота:

* + RDB (Redis Database) снапшоты**:** Выполняются каждые 6-12 часов. RDB — это моментальный снимок набора данных в определенный момент времени. Он компактный и целостный.

Тип бэкапов:

* RDB-снапшоты**.**