

Базы данных

Курс предназначен для разработчиков и администраторов, готовых освоить работу с базами данных

Длительность курса: 166 академических часов

1 Сущности и связи

1 **Реляционная модель данных. Реляционная алгебра**

познакомиться с реляционной теорией;
понять принципы проектирования реляционных баз данных;
освоить приемы нормализации данных;
построить свою модель

Домашние задания

1 Строим модель данных

Задача: реализовать модель данных БД, определить сущности, построить связи, выполнить декомпозицию и нормализацию

За основу берем практическую структуру данных с заказчиками интернет магазина (файл `some_customers.csv.gz`).

Текущая структура данных неоптимальна:

- нет типизации - почти все поля хранятся как строки

- данные не нормализованы - данные о адресе и человеке хранятся в одной таблице, на одного человека может приходиться несколько адресов

Попытаться выделить следующие сущности:

- страны

- города

- улицы

- дома

и другие которые посчитаете нужными

Описанные сущности не являются полным и конечным ТЗ (как это и бывает в жизни). Вы как архитектор должны предусмотреть необходимые атрибуты и дополнительные сущности по необходимости. И четко представлять бизнес-задачу которую будет решать эта структура данных.

делаем декомпозицию и нормализацию
в качестве сделанной ДЗ принимается pdf с
начальной и конечной моделью

Решая данное ДЗ вы тренируете навык проектирования БД, выделения сущностей и их атрибутов, построения связей, нормализации данных

2 Проектирование БД

разобрать вопросы и проблемы обслуживания и сопровождения, которые важно учесть на этапе проектирования

Домашние задания

1 Доработка проекта

В первом ДЗ была реализована сущность customers

Реализуйте сущности продукты, категории продуктов, цены, поставщики, производители и покупки

на примере того что было продемонстрировано на занятии

Свои решения для этой схемы приветствуются

В проекте должны быть

- схема
- документация
- примеры бизнес-задач которые решает база
- рекомендации к использованию репликации
- рекомендации к резервному копированию

ДЗ автоматически защищаетеся при защите проекта на последнем занятии модуля

3 **Компоненты современной СУБД**

понять и использовать в работе многообразие компонентов и возможностей СУБД: как управлять пользователями, какие типы коннектов бывают, как создавать индексы, как создавать ограничения (constrains), что такое триггеры, зачем нужны последовательности и очереди, для чего использовать хранимые процедуры и функции, что такое DB links, представления (view) и синонимы, Jobs

Домашние задания

1 Добавляем в модель данных дополнительные индексы и ограничения

1. Проводим анализ возможных запросов\отчетов\поиска данных.
2. Предполагаем возможную кардинальность поля.
3. Создаем дополнительные индексы - простые или композитные.
4. На каждый индекс пишем краткое описание зачем он нужен (почему по этому полю\полям).
5. Думаем какие логические ограничения в БД нужно добавить - например какие поля должны быть уникальны, в какие нужно добавить условия, чтобы не нарушить бизнес логику. Пример - нельзя провести операцию по переводу средств на отрицательную сумму.
6. Создаем ограничения по выбранным полям.

4 **Теорема CAP**

5 **Проблемы миграции**

разобраться в видах миграции;
узнать основные проблемы миграции;
решить проблемы, возникшие при миграции

- 6 **Семинар:
проектирование** рассказать о своем проекте, получить фидбэк от коллег и преподавателя

- 1 Внутренняя архитектура СУБД MySQL** настроить СУБД, влияющие на ее производительность

- 2 DDL: создание, изменение и удаление объектов в MySQL** создавать, модифицировать и удалять тейблспейсы, таблицы, пользователей
настраивать политики безопасности

- 3 Типы данных в MySQL** правильно выбирать типы данных для своих структур,
использовать функции для работы с определенными типами данных

- 4 DML: вставка, обновление, удаление, выборка данных** использовать различные способы выборки данных из одной и нескольких таблиц;
использовать нужный тип JOIN и различные условия в WHERE

Домашние задания

1 SQL выборка

Задачи на выбор

1) по базе voip

Рассчитываем стоимость звонка в БД voip
Строим Select для выборки всех звонков

поля для связи

CDR.src_ip -> oper_ip.IP_OP

oper_ip.OP_ID -> SITE.ID

SITE.rate_o -> RATES.RATE_ID

RATES.CODE_ID -> DEST_CODE.DEST_ID

RATES.price - цена

DEST_CODE.CODE. - код направления

CDR - запись о звонке

SRC_IP - ип адрес оператора от которого пришел звонок

oper_ip - в этой таблице находим этот ип адрес который действовал в это время

SITE - здесь находим подразделение оператора к которому привязан данный ип адрес

RATES здесь по (SITE.rate_o ->

RATES.RATE_ID) находим тариф который действовал в заданный период у данного подразделения оператора

который относится к НАИБОЛЬШЕМУ коду соответствующему номер из

CDR.BILL_NUMBER

коды описаны в DEST_CODE

(RATES.CODE_ID -> DEST_CODE.DEST_ID)

необходимо учитывать дату звонка, дату действия тарифа и дату действия IP оператора

В oper_ip также могут быть дублирующие IP. в том числе и по дате (есть условия по префиксу номера и ip сервера) - для облегчения задачи можно вытащить уникальные записи в отдельную таблицу oper_ip_tm

* сделать выборку с учетом префикса номера в oper_ip

2) по своей базе (пример для базы продуктов)

Отрисовка списка продуктов с указанием всех уровней категории (например Бытовая техника/холодильники)

и списком параметров в JSON или XML по

определенным условием (например все двухкамерные холодильники)

запрос должен. включать в себя 1 или несколько JOIN и условия WHERE

5 DML: агрегация и сортировка, CTE, аналитические функции

группировать и сортировать данные и использовать групповые функции

Домашние задания

1 Создаем отчетную выборку

Построить статистический запрос и показать группировку по странам (по каждому коду необходимо вычислить страну и сгруппировать по стране)

- по суммарному времени
- по кол-ву звонков
- по кол-ву нулевых звонков

должны быть представлены 3 SQL

* Использовать RollUP для развернутого отчета по странам и направлений внутри стран

- можно использовать свою БД и предоставить следующий результат
 - база с данными в докере
 - группировки с использованием CASE, HAVING, ROLLUP, GROUPING SET например для магазина к предыдущему списку продуктов добавить максимальную и минимальную цену и кол-во предложений также сделать выборку показывающую самый дорогой и самый дешевый товар в каждой категории
 - сделать rollup для оценки продаж по категориям товаров
-

6 Транзакции, MVCC, ACID

разбираться в уровнях изоляции транзакций
избегать взаимных блокировок
разбираться в различных видах блокировок
строить транзакции в mysql

7 Индексы

разобраться в типах индексов, плюсах и минусах,
создавать, удалять и мониторить индексы для оптимизации БД

Домашние задания

1 Индексы

Варианты:

1)

В материалах приложен дамп с 4 таблицами по продуктам

```
select * from
positions_models pm,
ppi_position_import ppi,
technical_details td,
technical_details_fields tdf
where pm.mod_ppi_id=ppi.ppi_id
and tdf.tdf_id=td.tch_property_id
and td.tch_mod_id=pm.mod_id
```

Задача - сделать полнотекстовый индекс, который ищет по свойствам, названию товара и названию модели

2) добавляем индексы в свой проект

формат сдачи -

сделать форк репозитория к себе
добавить в докерфайл создание индексов

8 Хранимые процедуры и триггеры

использовать хранимые процедуры и функций для оптимизации работы с БД

Домашние задания

1 Добавляем в базу хранимые процедуры и триггеры

варианты, сделать любой

1)
создаем в базе view

хранимые функции для определения стоимости звонка

хранимую процедуру для расчета данных за один день

- берется последний час которые есть в DAYSTAT
- рассчитываются данные за следующий день
- заносятся в DAYSTAT
- при следующем запуске считается следующий день
- если при первом запуске в DAYSTAT нет данных - считаются данные с того дня который первый присутствует в CDR

2) реализовать pipelined функцию в своей БД

3) реализовать функцию со своими типами данных

4) переписать приложенные функции для расчета звонка с mysql на postgres

- 9 **Оптимизация производительности. Профилирование. Мониторинг.** определять узкие места базы, оптимизировать и профилировать запросы
-
- 10 **Резервное копирование и восстановление** делать бэкап и затем восстанавливаться с него
-
- 11 **Репликация** построить репликацию, использовать proxySQL для балансировки, использовать оркестрацию, переключать реплики

Домашние задания

- 1 Спрогнозировать рост данных и спроектировать модель хранения и архивации

Формируем документ в котором описываем

- прогноз по возможному росту базы

* рост данных

* рост количества пользователей

* всплески одновременных соединений

- описываем возможные угрозы и методы защиты от них

* предлагаем стратегии бэкапа

- репликации

- кластеризации

-
- 12 **Кластеры MySQL: Galera, Innodb cluster** построить несколько видов MySQL кластеров: Galera, InnoDB Cluster
-

13 MySQL Fabric - шардинг

построить шардинг схему с помощью MySQL Fabric

- 1 Внутренняя архитектура СУБД PostgreSQL** настроить СУБД, влияющие на ее производительность

- 2 DDL: создание, изменение и удаление объектов в PostgreSQL**

- 3 Типы данных в PostgreSQL** разбираться в многообразии типов данных использовать встроенные функции для работы с типами данных использовать различные типы данных Postgres, создавать свои типы данных

- 4 DML: вставка, обновление, удаление, выборка данных** использовать различные способы выборки данных из одной и нескольких таблиц; использовать нужный тип JOIN и различные условия в WHERE

- 5 DML: агрегация и сортировка, CTE, аналитические функции** группировать и сортировать данные и использовать групповые функции

- 6 Транзакции, MVCC, ACID** разбираться в уровнях изоляции транзакций избегать взаимных блокировок разбираться в различных видах блокировок строить транзакции в PostgreSQL

| | | |
|----|--|---|
| 7 | Индексы | разобраться в типах индексов, плюсах и минусах, создавать, удалять и мониторить индексы для оптимизации БД |
| 8 | Хранимые процедуры и триггеры | использовать хранимые процедуры и функций для оптимизации работы с БД |
| 9 | Оптимизация производительности. Профилирование. Мониторинг. | определять узкие места базы, оптимизировать и профилировать запросы |
| 10 | Резервное копирование и восстановление | делать бэкап и затем восстанавливаться с него |
| 11 | Репликация | создать реплику базы, сделать ручное переключение ролей, переключится после фэйловера, настроить архивацию логов |
| 12 | Кластеризация: patroni | построить отказоустойчивый кластер с помощью patroni |
| 13 | OLAP: Postgres + pentaho | |

4 NoSQL (Redis, Tarantool, MongoDB, Cassandra)

- 1 NoSQL: Redis**

понимать, когда и для каких целей имеет смысл использовать NoSQL;
применять Redis

- 2 Работа с tarantool**

рассмотрим основы работы с СУБД tarantool.
Рассмотрим ее архитектуру, поймём ее возможности, научимся строить высоконапряженные решения на ее базе.

- 3 Базовые возможности mongodb.**

понятие коллекции, документов и полей.
CRUD-операции.
Фильтрация по полям
sort, skip, limit
Запросы по поддокументам

- 4 Кластерные возможности mongodb**

варианты репликации.
Replica set.
Концепция кворума.
Балансировка.
Подходы к шардированию в БД.
Шардирование в mongodb: mongos
Правильный выбор ключа шардирования.

- 5 map-reduce framework**

концепция map-reduce.
Зачем нужен map-reduce в mongodb
Концепция pipeline.
Структура и синтаксис агрегации.
\$match
Стадия \$group
Стадия \$lookup

6 **оптимизация
производительности
mongodb**

профилирование
Построение индексов.
Виды индексов в mongodb
Оптимизации CRUD.
Дисковые движки.
GridFS
Производительность кластеров.

7 **Утилиты mongodb**

настройки mongodb
Мониторинг mongodb
Обеспечение безопасности кластера.
Обеспечение бэкапа.

Домашние задания

1 Домашняя работа

Цель: На основе mongodb сделать сервис поиска объектов на карте, аналогичную Яндекс Картам. Научиться в радиусе от точки находить объекты по их типам. Настроить реплицирование и шарлирование. Сохранять в том числе и фотографии объектов.

8 **Архитектура и
применение
cassandra**

- DynamoDB-подобные базы
 - Column-family модель
 - LSM-деревья
 - Фильтр Блума
 - Работа Cassandra с диском
 - Уплотнение
 - Какие проекты работают с Cassandra
 - Зачем в этих проектах применяется Cassandra
-

9 **Язык запросов
cassandra**

- Язык CQL и его отличие от SQL
 - Типы данных в Cassandra
 - DDL: Keyspace, table
 - DML: Select, Insert, Update, Delete, Batch
 - Вторичные индексы
 - Materialized Views
 - Скалярные и агрегирующие функции
 - Поддержка JSON
-

10 **Масштабирование и
отказоустойчивость
cassandra. Часть 1**

- Репликация Cassandra и ее стратегии.
 - Отказоустойчивость ЦОД.
 - Выбор replication factor.
 - Понятие eventual consistency.
 - Уровни консистентности.
 - Выбор уровня консистентности.
 - Дисковые движки
-

11 **Масштабирование и
отказоустойчивость
cassandra. Часть 2**

12 Средства мониторинга cassandra

- Безопасность cassandra
- Обзор базовых метрик cassandra
- Настройка grafana для мониторинга cassandra
- Бэкапы
- Инструменты cassandra: sstableverify, sstableutil, sstableupgrade, sstabledump, sstableexpiredblockers, sstablelevelreset, sstableloader

Домашние задания

- 1 Реализовать отказоустойчивую систему учета действий пользователей сайта - логин, логат, переход по страницам. Получить возможность быстрого чтения лога действий.

5 Общие задачи и проблемы

- 1 Итоговое занятие**
 - подвести итоги обучения на курсе
 - узнать, какие еще курсы будут полезны для выпускников
 - определиться с планами по карьерному консультированию
 - дать фидбэк о программе и занятиях

1 Проект

Домашние задания

1 Проект

2 Обсуждение проектов

защитить проект;
ознакомиться с результатами проектов
одногогруппников

Домашние задания

1 Проектная
работа