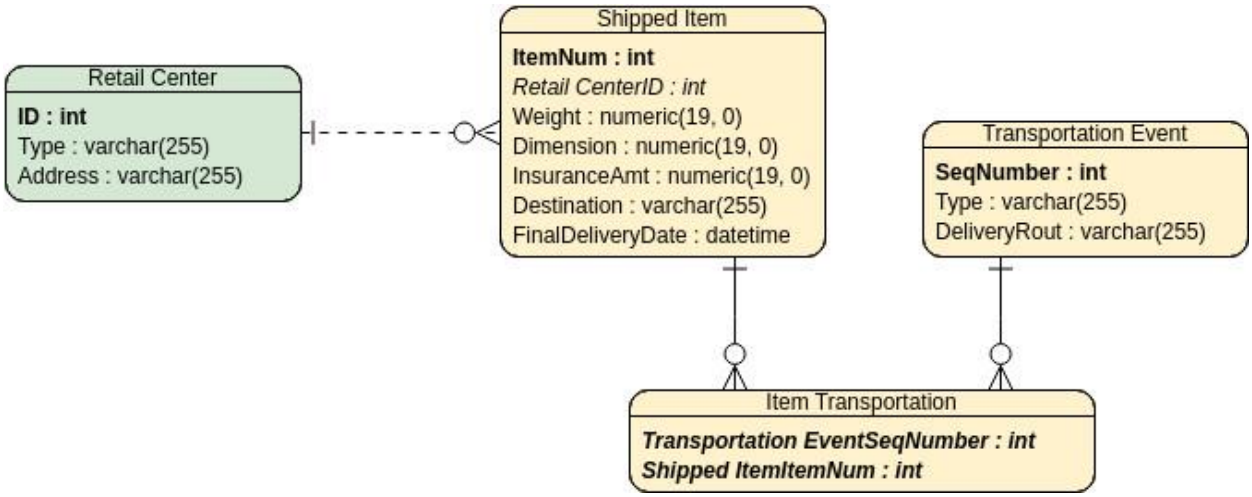
Практическое задание №3.1 Проектирование аналитической схемы базы данных

Постановка задачи

Седьмое практическое задание связано с проектированием схемы базы данных для аналитики. Будем исходить из того, что приложение, для которого была сделана база данных в задании стала очень популярной и по ней каждый день можно собирать большой объем статистической информации. Результатом данного практического задания являются: скрипты создания базы данных, хранимая процедура (генератор) для ее заполнения, анализ плана выполнения запроса.

Пример используемой ER диаграммы (система доставки UPS)



ссылка. Можно использовать эту или любую другую похожую по сложности диаграмму аналитической БД.

Требования к БД

- Как минимум одна таблица должна содержать не меньше 10 млн. записей, которые со временем теряют актуальность.
- Другая таблица, связанная с первой, должна содержать не меньше 1 млн. записей.
- В одной из таблиц с количество записей больше 1 млн. должна быть колонка с текстом, по которой будет необходимо настроить полнотекстовый поиск.

Практическая часть включает:
1) наполнение таблицы, для этого нужно написать хранимую процедуру - генератор на языке ***plpython3u***, которая использует словари (для строковых типов), случайные значения (для строковых, числовых типов).
2) оценку скорости выполнения запросов.
Для этого могут быть использованы механизмы секционирования, наследования и индексов.
Необходимо подготовить два запроса:

- Запрос к одной таблице, содержащий фильтрацию по нескольким полям.
- Запрос к нескольким связанным таблицам, содержащий фильтрацию по нескольким полям.

Для каждого из этих запросов необходимо провести следующие шаги:

- Получить план выполнения запроса без использования индексов (удаление индекса или отключение его использования в плане запроса).

- Получить статистику (IO и Time) выполнения запроса без использования индексов.
- Создать нужные индексы, позволяющие ускорить запрос.
- Получить план выполнения запроса с использованием индексов и сравнить с первоначальным планом.
- Получить статистику выполнения запроса с использованием индексов и сравнить с первоначальной статистикой.
- Оценить эффективность выполнения оптимизированного запроса.

Также необходимо продемонстрировать полезность индексов для организации полнотекстового поиска.

Для таблицы объемом 10 млн. записей произвести оптимизацию, позволяющую быстро удалять старые данные, ускорить вставку и чтение данных.

Темы для проработки

- Наполнение базы данных
  https://postgrespro.ru/docs/postgrespro/14/populate
- EXPLAIN
  https://postgrespro.ru/docs/postgrespro/14/using-explain
  https://postgrespro.ru/docs/postgrespro/14/planner-stats
  https://postgrespro.ru/docs/postgrespro/14/explicit-joins
- ANALYZE
  https://postgrespro.ru/docs/postgrespro/14/routine-vacuuming#VACUUM-FORSTATISTICS
- Индексы
  https://postgrespro.ru/docs/postgrespro/14/indexes
- Полнотекстовый поиск
  https://postgrespro.ru/docs/postgrespro/14/textsearch
- Наследование таблиц
  https://postgrespro.ru/docs/postgrespro/14/ddl-inherit
- Секционирование таблиц
  https://postgrespro.ru/docs/postgrespro/14/ddl-partitioning
- Полное описание синтаксиса встретившихся команд
  https://postgrespro.ru/docs/postgrespro/14/sql-commands

Примеры вопросов

- В чем отличие первичного ключа и уникального индекса?
- В каких случаях имеет смысл создавать индексы? Какие колонки следует включать в индекс и почему?
- Какие существуют способы внутренней организации индексов?
- Рассказать о проблеме фрагментации индексов. Как бороться с фрагментацией?
- Имеет ли значение порядок указания колонок при создании индекса?
- В чем разница между Index Scan и Index Seek?
- В чем разница между секционированием и наследованием?
- Зачем нужен ANALYZE?
- Исправить ошибки в подготовленных выборках.
- Могут ли индексы ухудшить производительность? Если да, то продемонстрировать это.
- На что влияет порядок сортировки (ASC\DESC) при создании индекса? Продемонстрировать это.

- Продемонстрировать полезность индекса по выражению.
- Продемонстрировать полезность частичного индекса.

Практическое задание №3.2 Использование документно-ориентированных объектов типа Json.

Постановка задачи

PostgreSQL стала первой реляционной базой данных, поддерживающей слабоструктурированные данные. В PostgreSQL для этого используется JSON (JavaScript Object Notation, Запись объекта JavaScript RFC 7159), который имеет два представления: json и jsonb. Для реализации эффективного механизма запросов к этим типам данных в Postgres также имеется тип jsonpath. Официально JSON появился в PostgreSQL в 2014 году. PostgreSQL с JSONB совмещает гибкость NoSQL , а также надёжность и богатство функциональности реляционных СУБД.

В практической части необходимо:

- создать БД IMDB test, использующую стандартные атрибуты и атрибут jsonb. Ссылка на интефейсы. Таблицы находятся на я.диске в папке DataSet. Описание атрибута jsonb:

```
{
    "nconst": "nm0000151",
    "primaryName": “Morgan Freeman”,
    "roles": [
         {
             "title": "The Shawshank Redemption",
             "year": "1994",
             "character name": "Ellis Boyd 'Red' Redding"
         },
         {
             "title": "Unforgiven",
             "year": "1992",
             "character name": "Ned Logan"
         }
         {
             "title": "Through the Wormhole",
             "series name": "Are Aliens Inside Us? (#6.5)"
             "year": "2010",
             "character name": "Himself - Narrator"
         }
     ],
     "birthYear": “1937”,
     "deathYear": “\N”,
}
```

  где nconst, birthYear, deathYear это записи таблицы *name.basics.tsv.gz,* roles загружать из таблицы
- Составить 3-4 запроса с использованием jsonb.
- Измерить время доступа к ключу для каждой строчки (в виде таблицы или графика). Оценить влияние длины строки на скорость доступа (линейная зависимость). Как можно это влияние уменьшить.
- Составить запрос на изменение PrimaryName у актера. Сравнить изменение объема БД для актера с малым кол-вом ролей и актера с большим количеством ролей (toasted roles).

Темы для проработки

- Денормализация
  https://habr.com/ru/company/latera/blog/281262
  https://habr.com/ru/post/64524
- Json
  https://postgrespro.ru/docs/postgrespro/14/datatype-json
  https://postgrespro.ru/docs/postgrespro/14/functions-json
  https://habr.com/ru/company/oleg-bunin/blog/646987/

Примеры вопросов

- В чем отличие типов json и jsonb?
- Какие типы индексации поддерживает jsonb?
- Что такое jsonpath?
- Что такое toasted object?