ДЗ Производительность индексов

Цель:

В результате выполнения ДЗ вы создадите набор тестовых данных для проведения нагрузочного тестирования, подберете наиболее подходящие индексы и проведете тесты производительности.
В данном задании тренируются навыки:

- генерация тестовых данных;
- работа с индексами;
- нагрузочное тестирование;

Описание/Пошаговая инструкция выполнения домашнего задания:

1. Сгенерировать любым способ 1,000,000 анкет. Имена и Фамилии должны быть реальными (чтобы учитывать селективность индекса)
2. Реализовать функционал поиска анкет по префиксу имени и фамилии (одновременно) в вашей социальной сети (запрос в форме firstName LIKE ? and secondName LIKE ?). Сортировать вывод по id анкеты. Использовать InnoDB движок.
3. Провести нагрузочные тесты по этой странице. Поиграть с количеством одновременных запросов. 1/10/100/1000.
4. Построить графики и сохранить их в отчет
5. Сделать подходящий индекс.
6. Повторить пункт 3 и 4.
7. В качестве результата предоставить отчет в котором должны быть:

- графики latency до индекса;
- графики throughput до индекса;
- графики latency после индекса;
- графики throughput после индекса;
- запрос добавления индекса;
- explain запросов после индекса;
- объяснение почему индекс именно такой;

Отчет проведении тестирования

Структура таблицы

```
CREATE TABLE `account` (
  `Id` varchar(36) NOT NULL,
  `Login` varchar(45) NOT NULL COMMENT '          ',
  `Password` varchar(100) NOT NULL,
  `FirstName` varchar(45) NOT NULL,
  `LastName` varchar(45) NOT NULL,
  `Age` tinyint NOT NULL,
  `Gender` tinyint(1) NOT NULL,
  `Interests` varchar(255) NOT NULL,
  `City` varchar(100) NOT NULL,
  `CreateDate` datetime NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE KEY `Id_UNIQUE` (`Id`),
  UNIQUE KEY `Login_UNIQUE` (`Login`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Генерирование 1 000 000 анкет использовал стороннюю библиотеку Bogus
Запрос для нагрузочного тестирования

```
select * from WebSocial.Account
where FirstName like 'art%' or LastName like 'art%';
```

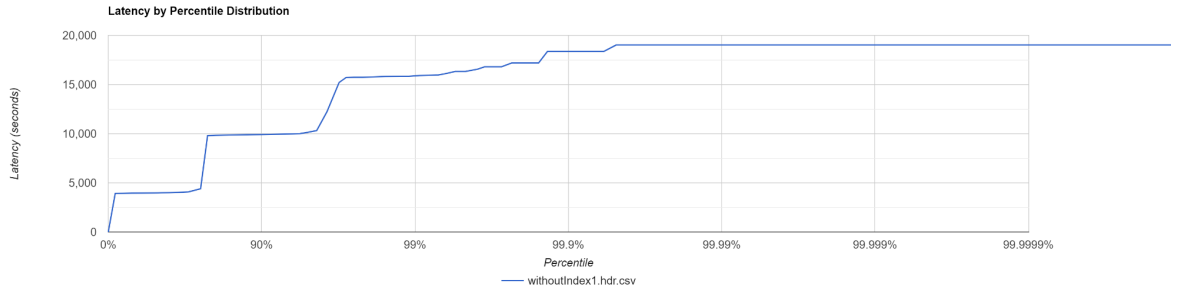
с планом

```
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "108144.42"
    },
    "table": {
      "table_name": "Account",
      "access_type": "ALL",
      "rows_examined_per_scan": 967043,
      "rows_produced_per_join": 202940,
      "filtered": "20.99",
      "cost_info": {
        "read_cost": "87850.37",
        "eval_cost": "20294.05",
        "prefix_cost": "108144.42",
        "data_read_per_join": "489M"
      },
      "used_columns": [
        "Id",
        "Login",
        "Password",
```

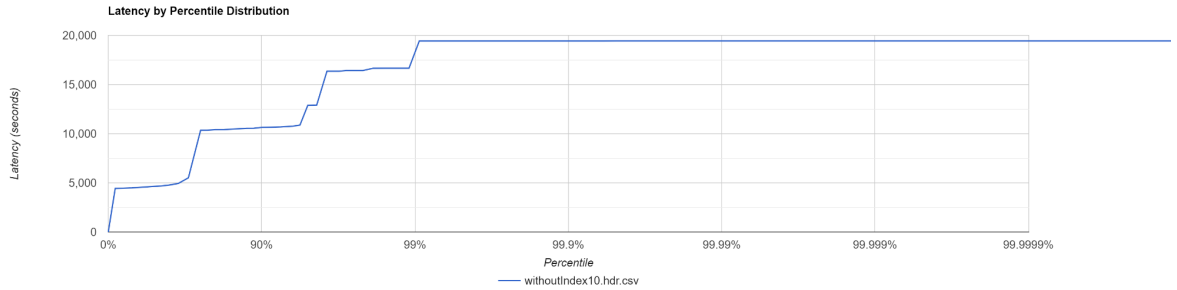
```
        "FirstName",
        "LastName",
        "Age",
        "Gender",
        "Interests",
        "City",
        "CreateDate"
      ],
      "attached_condition": "((`websocial`.`account`.`FirstName` like 'art%') or
(`websocial`.`account`.`LastName` like 'art%'))"
    }
  }
}
```

Результаты тестирования без индекса. Измерение в секундах

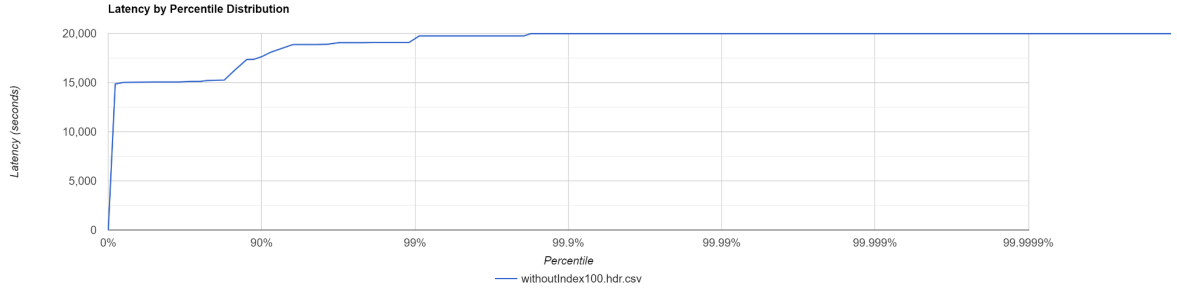
1. Latency при 1 потоке



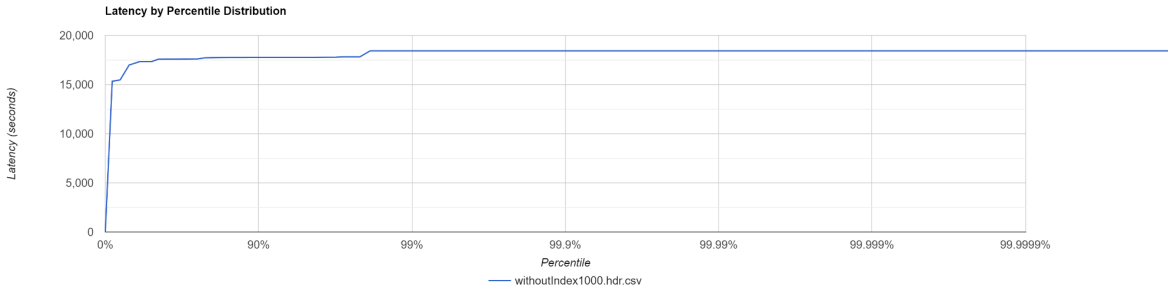
2. Latency при 10 потоков



3. Latency при 100 потоков



4. Latency при 1000 потоков



Для оптимизации запроса решил добавить два индекса

create index first_name_IDX on websocial.Account (FirstName(3)) USING BTREE;

create index last_name_IDX on websocial.Account (LastName(3)) USING BTREE;

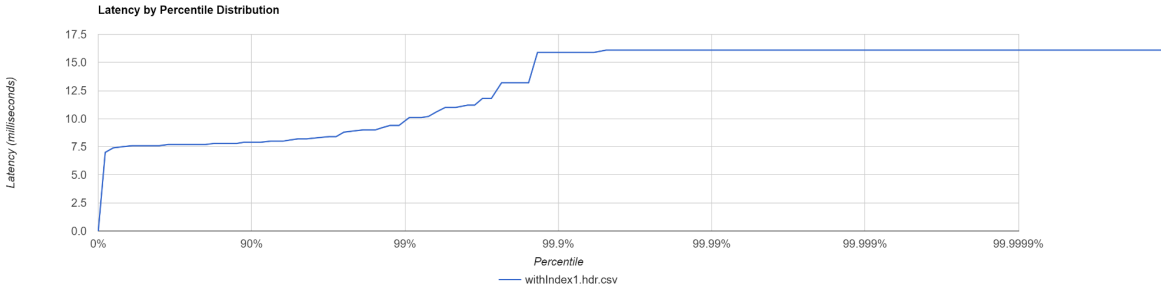
План запроса


```
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "1895.61"
    },
    "table": {
      "table_name": "Account",
      "access_type": "index_merge",
      "possible_keys": [
        "first_name_IDX",
        "last_name_IDX"
      ],
      "key": "sort_union(first_name_IDX,last_name_IDX)",
      "key_length": "14,14",
      "rows_examined_per_scan": 1012,
      "rows_produced_per_join": 1012,
      "filtered": "100.00",
      "cost_info": {
        "read_cost": "1794.41",
        "eval_cost": "101.20",
        "prefix_cost": "1895.61",
        "data_read_per_join": "2M"
      },
      "used_columns": [
        "Id",
        "Login",
        "Password",
        "FirstName",
        "LastName",
        "Age",
        "Gender",
        "Interests",
        "City",
```

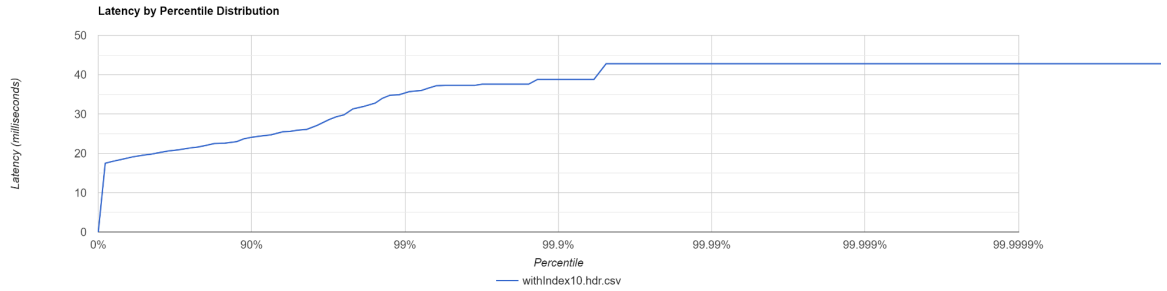
```
        "CreateDate"
      ],
      "attached_condition": "((`websocial`.`account`.`FirstName` like 'art%') or
(`websocial`.`account`.`LastName` like 'art%'))"
    }
  }
}
```

Результаты тестирования после добавления двух индексов. Тут уже измерение в миллисекундах

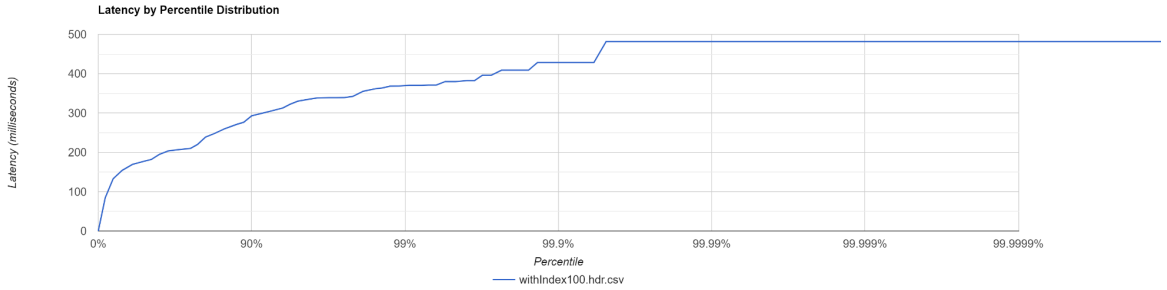
1. Latency при 1 потоке



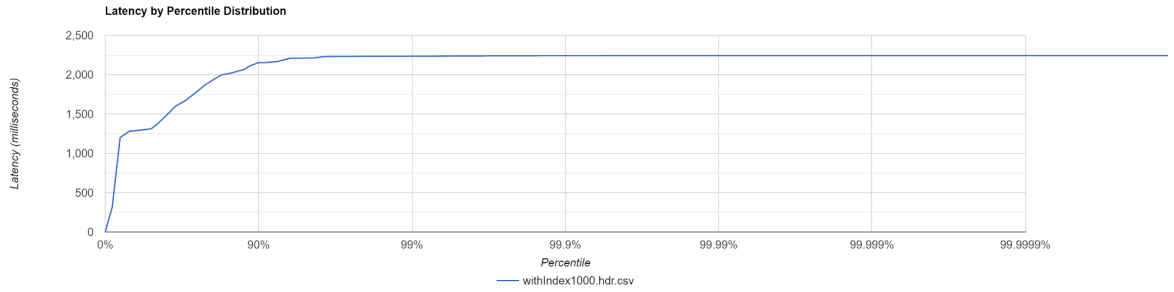
2. Latency при 10 потоков



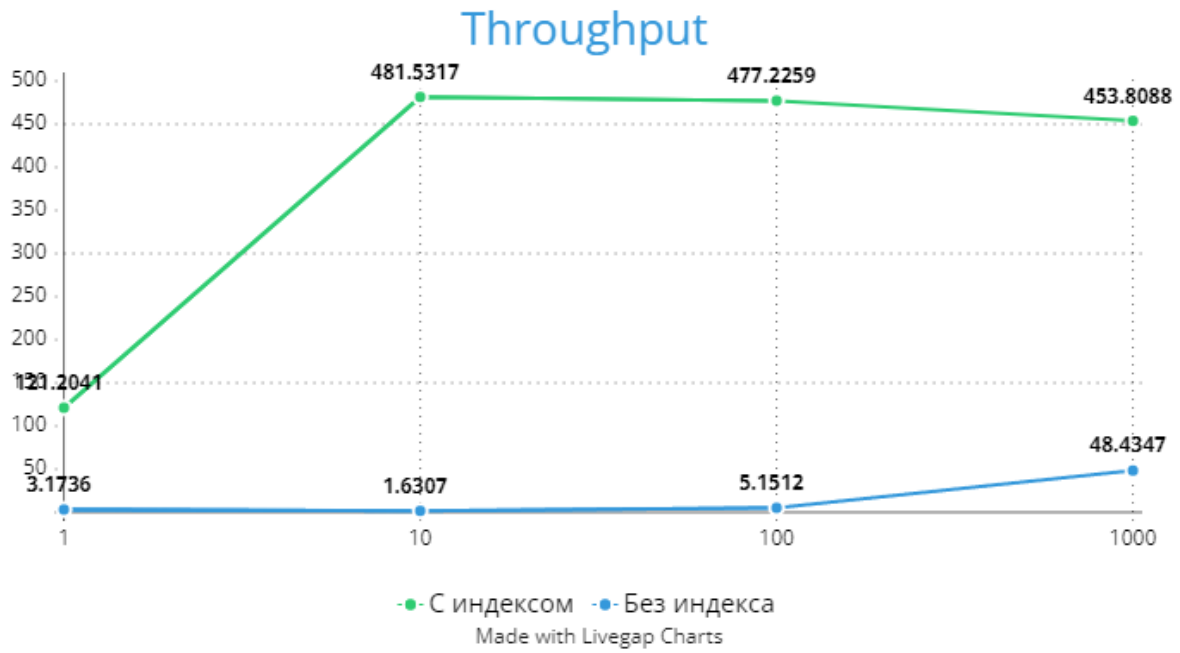
3. Latency при 100 потоков



4. Latency при 1000 потоков



Пропускная способность по количеству соединений



Для нагрузочного тестирования использовал [Hey](), написанный на Go