

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ

Кузнецов Дмитрий Алексеевич

Курсовая работа

Смартфоны Apple

Направление 010302

Прикладная математика и информатика

Преподаватель

Филиппов Р.О.

Санкт-Петербург
2017

Содержание

Глава 1. Описание и структура.....	2
Глава 2. Легкие запросы.....	4
Глава 3. Средние запросы.....	5
Глава 4. Сложные запросы.....	6

Все файлы доступны по адресу:

github.com/DmitryKuznetsov1/database_iphone

ГЛАВА 1. ОПИСАНИЕ И СТРУКТУРА

Данная схема описывает базу данных всех вышедших на данный момент смартфонов компании

Apple. В ней представлено 6 объектов,

1) Model - Модель телефона.

Поля:

- model_id - идентификатор модели телефона (primary key)
- name - название модели (например, iPhone 5C), характеристики которой представлены в других полях этой таблицы
- memory, Gb - количество оперативной памяти модели в гигабайтах
- battery capacity, mA*h - емкость аккумулятора модели в миллиампер*часах
- display_id - идентификатор экрана модели, характеристики которого представлены в таблице display
- cpu_id - идентификатор процессора, характеристики которого представлены в таблице cpu

2) Color - Цвет корпуса (список всех)

Поля:

- color_id - идентификатор цвета (primary key)
- color - наименование цвета (маркетинговое название)

3) Storage - Объем памяти (список всех)

Поля:

- storage_id - идентификатор объема памяти (primary key)
- storage, Gb - объем памяти в гигабайтах

4) Cpu manufacturer - Основная информация о производителе процессора

Поля:

- cpu_manufacturer_id - идентификатор компании производителя процессора (primary key)
- manufacturer name - название компании
- location country - страна, в которой располагается штаб-квартира компании
- year of foundation - год основания компании
- current CEO - имя текущего генерального директора компании

5) Display - Экран (с основными характеристиками)

Поля:

- display_id - идентификатор экрана (primary key)
- screen size, inch - размер экрана в дюймах
- type of display - тип экрана (LED, OLED и др.)
- pixel resolution - разрешение экрана
- pixels per inch - количество пикселей на дюйм

6) Cpu - процессор (с основными характеристиками)

Поля:

- cpu_id - идентификатор процессора (primary key)
- cpu name - название процессора
- frequency, Ghz - частота процессора в гигагерцах
- number of cores - количество ядер процессора
- cpu manufacturer id - идентификатор компании производителя процессора

В табличной схеме представлены по одной таблице для каждого объекта, а также еще 2 таблицы:

1) Available colors - Доступные цвета моделей (у каждой модели может быть один или несколько доступных цветов, при этом один и тот же цвет может быть доступен для нескольких моделей)

Поля:

- model_id - идентификатор модели
- color_id - идентификатор цвета

2) Available storage - Доступные объемы памяти (у каждой модели может быть один или несколько доступных объемов памяти, при этом один и тот же объем памяти может быть доступен для нескольких моделей)

Поля:

- model_id - идентификатор модели
- storage_id - идентификатор объема памяти

Таким образом, в схеме представлены отношения:

а) Доступный цвет m:m модель

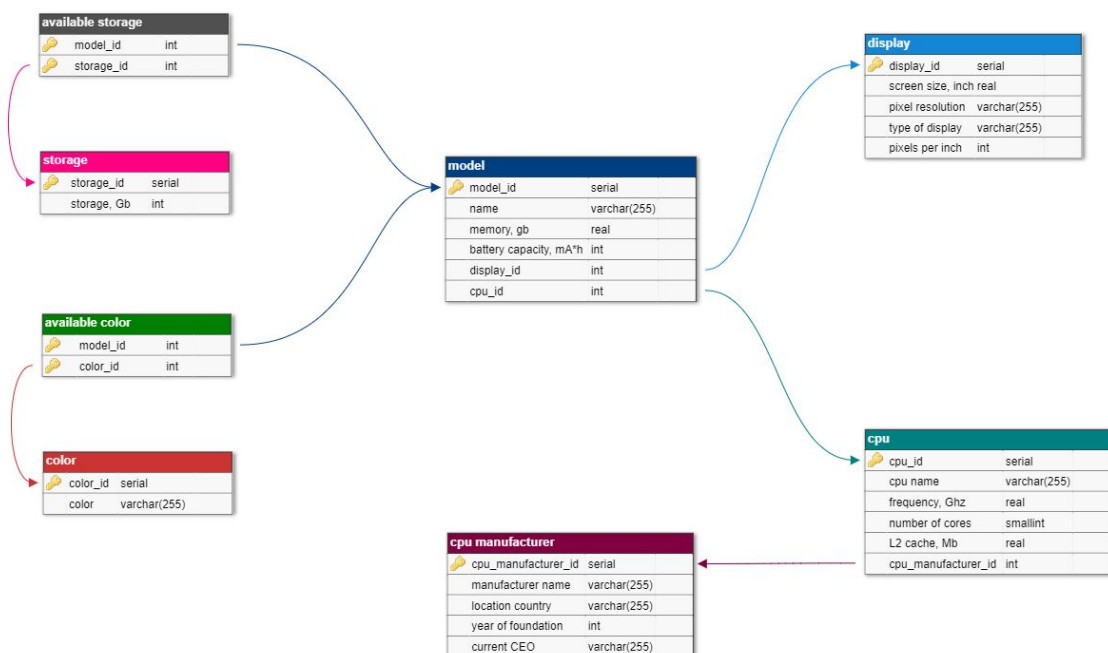
б) Доступный объем памяти m:m модель

в) Модель m:1 процессор

г) Экран m:1 модель

д) Процессор m:1 производитель процессора

Ниже представлена табличная схема, описанная выше.



ГЛАВА 2. ЛЕГКИЕ ЗАПРОСЫ

1. Показывает названия всех моделей, память которых больше 1 Гб, а также объем памяти.

```
SELECT DISTINCT name AS "Name", memory_gb AS "Memory, Gb"  
FROM model  
WHERE memory_gb > 1;
```

Оптимизация:

Добавлен индекс **model_memory_gb_idx** для фильтрации телефонов по количеству памяти
CREATE INDEX ON model(memory_gb) WHERE memory_gb>1;

2. Показывает первый процессор, в котором количество ядер равно 2
P.S Предполагается, что чем больше id, тем позже был выпущен процессор.

```
SELECT cpu_name AS "Name"  
FROM cpu  
WHERE number_of_cores=2  
ORDER BY cpu_id  
LIMIT 1;
```

Оптимизация:

Добавлен индекс **cpu_number_of_cores_idx** для фильтрации процессоров по количеству ядер.
CREATE INDEX ON cpu(number_of_cores) WHERE number_of_cores=2;

3. Сортирует все модели по объему батареи (по убыванию), при этом выводит название каждой модели и соответствующий объем.

```
SELECT name AS "Name", battery_capacity_mah AS "Battery capacity, mA*h"  
FROM model  
ORDER BY battery_capacity_mah DESC;
```

Оптимизация:

Добавлен индекс **model_battery_capacity_mah_idx** для сортировки моделей по объему батареи.

CREATE INDEX ON model(battery_capacity_mah);

4. Показывает количество процессоров, имеющих один и тот же объем L2 кэша, а также для удобства сортирует по убыванию объема L2 кэша

```
SELECT l2_cache_mb AS "L2 cache, Mb", count(*) AS "Number of cpu"  
FROM cpu  
GROUP BY l2_cache_mb  
ORDER BY l2_cache_mb;
```

Оптимизация:

Добавлен индекс **cpu_l2_cache_mb_idx** для фильтрации процессоров по объему L2 кэша.
CREATE INDEX ON cpu(l2_cache_mb);

ГЛАВА 3. СРЕДНИЕ ЗАПРОСЫ

1. Показывает доступные цвета для iPhone 8.

```
SELECT color.color AS "Available iPhone 8 colors"
FROM model
JOIN available_color ON model.model_id=available_color.model_id
JOIN color ON available_color.color_id=color.color_id
WHERE model.name='iPhone 8';
```

Оптимизация:

Используются индексы **color_pkey**, **model_pkey** и **available_color_pkey** для оптимизации объединения таблиц.

2. Показывает количество процессоров (в смысле различных моделей), произведенных компанией TSMC.

```
SELECT cpu_manufacturer.manufacturer_name AS "Manufacturer name", count (*) AS
"Number of cpu made"
FROM cpu
JOIN cpu_manufacturer ON
cpu_manufacturer.cpu_manufacturer_id=cpu.cpu_manufacturer_id
WHERE cpu_manufacturer.manufacturer_name='TSMC'
GROUP BY cpu_manufacturer.manufacturer_name;
```

Оптимизация:

Используются индексы **cpu_pkey** и **cpu_manufacturer_pkey** для оптимизации объединения таблиц.

Добавлен индекс **cpu_manufacturer_manufacturer_name_idx** для фильтрации компаний по названию.

```
CREATE INDEX ON cpu_manufacturer(manufacturer_name) WHERE
manufacturer_name='TSMC';
```

3. Показывает процессор, который чаще остальных используется в телефонах Apple. Если таких несколько - показывает все в порядке убывания id номеров

```
SELECT cpu.cpu_id, cpu.cpu_name, count(*) AS number_of_models_using_this_cpu
FROM model
JOIN cpu ON model.cpu_id=cpu.cpu_id
GROUP BY cpu.cpu_id
HAVING count(*)=(SELECT MAX(number_of_models_using_this_cpu)
FROM (SELECT cpu.cpu_id, count(*) AS number_of_models_using_this_cpu
FROM model
JOIN cpu ON model.cpu_id=cpu.cpu_id
GROUP BY cpu.cpu_id) AS t1)
ORDER BY cpu_id;
```

Оптимизация:

Используются индексы **cpu_pkey** и **model_pkey**, добавлен индекс **model_cpu_id_idx** для оптимизации объединения таблиц.

```
CREATE INDEX ON model(cpu_id);
```

ГЛАВА 4. СЛОЖНЫЕ ЗАПРОСЫ

1. Выводит названия всех производителей и количество произведенных (в смысле различных моделей) процессоров, имеющих 1 ядро

```
SELECT m.manufacturer_name, count(c.cpu_id) AS number_of_models
FROM cpu_manufacturer AS m
LEFT JOIN cpu AS c
ON (m.cpu_manufacturer_id=c.cpu_manufacturer_id AND number_of_cores=1)
GROUP BY m.manufacturer_name;
```

2. Выводит максимальное значение объема памяти, доступного как для iPhone X, так и для iPhone SE

```
SELECT MAX(storage_gb)
FROM (SELECT storage_gb, count(*)
FROM (SELECT storage.storage_gb
FROM model
JOIN available_storage ON model.model_id=available_storage.model_id
JOIN storage ON available_storage.storage_id=storage.storage_id
WHERE model.name = 'iPhone X' OR model.name = 'iPhone SE') AS q1
GROUP BY storage_gb
HAVING count(*)=2) AS q2;
```

3. Показывает сколько существует вариантов сборки каждого телефона (по цвету и объему памяти), т.е количество пар (цвет, объем памяти) для каждой модели

```
SELECT t3.name, (t3.number_of_available_storage * t4.number_of_available_color) AS
number_of_options FROM
(SELECT name, count(*) AS number_of_available_storage
FROM(
SELECT m.name, s.storage_gb
FROM model AS m
JOIN available_storage AS ast ON m.model_id=ast.model_id
JOIN storage AS s ON ast.storage_id=s.storage_id) AS t1
GROUP BY t1.name) AS t3
JOIN
(SELECT name, count(*) AS number_of_available_color
FROM(
SELECT m.name, c.color
FROM model AS m
JOIN available_color AS ac ON m.model_id=ac.model_id
JOIN color AS c ON ac.color_id=c.color_id) AS t2
GROUP BY t2.name) AS t4
ON t3.name=t4.name
ORDER BY number_of_options DESC;
```

