## Пояснительная записка

**Таблица CON\_CONSUMER (Покупатели).**

1. Функциональный индекс по полю **CITY** (Город), т.е. **UPPER(CITY)**.

Возможен запрос, при котором существует необходимость отфильтровать данные по **городу** проживания покупателей.

Используем **UPPER** для того, чтобы привести данные к верхнему регистру. Данное действие требуется для того, чтобы устранить возможную проблему при обращении к БД.

Например, в поле **CITY** может быть написано название города как: «сочи», «Сочи» e.t.c. Таким образом, образуется проблема, при котором написанный запрос может вернуть не все интересующие пользователя данные.

1. Функциональный индекс по полю **DISPLAY** (ФИО пользователя), т.е. **UPPER(DISPLAY)**.

Возможен запрос, при котором существует необходимость отфильтровать данные по **ФИО** покупателей. По аналогии, см. (пример №1).

1. Составной функциональный индекс (опционально):

**(DISPLAY + CITY) или (DISPLAY + BIRTH\_DATE)**: Возможен запрос, при котором пользователя интересуют данные по конкретному покупателю, который проживает в конкретном городе или имеет указанную дату рождения. Каждое поле требуется «обернуть» в UPPER.

**Таблица CON\_TRANSACTION (Покупки).**

1. Составной функциональный индекс (**CUSTOMER\_ID + ORDER\_DATE**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **CUSTOMER\_ID(Покупатель)** и **ORDER\_DATE(Дата совершения покупки)**, к **ORDER\_DATE** применяем функцию **TRUNC(…)**, чтобы обрезать время у даты, т.к. зачастую пользователя интересует информация за указанную дату (без привязки к конкретному времени).

1. Составной функциональный индекс (**STORE\_ID + ORDER\_DATE**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **STORE\_ID**(Магазин) и **ORDER\_DATE**(Дата совершения продажи), к **ORDER\_DATE** применяем функцию **TRUNC(…)**, чтобы обрезать время у даты, т.к. зачастую пользователя интересует информация за указанную дату (без привязки к конкретному времени).

**Таблица CON\_SHOP (Магазин).**

1. Составной функциональный индекс **(STORE\_NAME + CITY)**:

Возможен запрос, при котором существует необходимость отфильтровать данные по **STORE\_NAME**(Название магазина) и **CITY**(Название города), поля обязательно обернуть в **UPPER**(…). Например, когда пользователь ищет все возможные магазины X, которые располагаются в городе Y.

1. Составной функциональный индекс (**STORE\_NAME + STREET**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **STORE\_NAME**(Название магазина) и STREET (Название улицы), поля обязательно обернуть в **UPPER**(…). Например, когда пользователь ищет магазин X и хочет понять есть ли магазин по указанной улице.

1. Составной функциональный индекс (**STORE\_NAME + CITY + STREET**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **STORE\_NAME**(Название магазина), **CITY**(Название города) и **STREET** (Название улицы), поля обязательно обернуть в **UPPER**(…). Например, когда пользователь пытается точно идентифицировать магазин по введенным данным, чтобы найти электронную почту или номер телефона магазина.

1. Функциональный индекс (**STORE\_NAME**):

Возможен запрос, при котором существует необходимость отфильтровать данные по названию магазина. Например, магазин X может располагаться в разных городах.

1. Функциональный индекс (**CITY**):

Возможен запрос, при котором существует необходимость отфильтровать данные по городу. Например, пользователь A хочет узнать информацию о магазинах (A, B, e.t.c), которые располагаются в конкретном городе.

**Таблица CON\_PURCHGOODS (Купленный товар).**

1. Обычный индекс (**PRODUCT\_ID + DISCOUNT**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **ID продукта** и **скидке** этого продукта.

1. Обычный индекс (**PRODUCT\_ID + LIST\_PRICE**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **ID продукта** и **цене** этого продукта.

1. Обычный индекс(**PRODUCT\_ID + QUANTITY**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **ID продукта** и **количеству** этого продукта.

**Таблица RLZ\_GOODS (Товары).**

1. Функциональный индекс (**PRODUCT\_NAME**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **названию продукта**. Поле **PRODUCT\_NAME** обязательно обернуть в **UPPER**.

1. Составной функциональный индекс (**PRODUCT\_NAME + LIST\_PRICE**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **названию продукта** и **цене**. Поле **PRODUCT\_NAME** обязательно обернуть в **UPPER**.

**Таблица RLZ\_CATEGORY (Категория товара).**

1. Функциональный индекс (**CATEGORY\_NAME**):

Возможен запрос, при котором существует необходимость отфильтровать данные по категории продукта. С одной стороны, можно этот индекс опустить, если будет не так много данных, но с другой: к сущности RLZ\_CATEGORY можно применить нормализацию и атомарно разбить некоторые категории на более конкретные данные. Если «Молочные продукты», то сметана, кефир, сыворотка, ряженка e.t.c. Поле **CATEGORY\_NAME** требуется «обернуть» в **UPPER**(…);

**Таблица RLZ\_WAREHOUSE (Склад).**

1. Составной функциональный индекс (**PRODUCT\_ID + REC\_DATE**):

Возможен запрос, при котором существует необходимость отфильтровать данные по **ID продукта** и **времени** отправки в магазин. Поле **REC\_DATE** можно обернуть в TRUNC(…), чтобы обрезать время.

**Таблица PUR\_MANUFACTURER (Производитель).**

1. Составной функциональный индекс (**PRODUCER\_NAME**):

Возможен запрос, при котором существует необходимость отфильтровать данные по названию производителя.

**Таблица PUR\_VENDOR (Поставщик).**

1. Составной функциональный индекс (**VENDOR\_NAME**):

Возможен запрос, при котором существует необходимость отфильтровать данные по названию поставщика.

**Таблица PUR\_SHIPMENT (Партия товара).**

1. Составной функциональный индекс **(VENDOR\_ID + PART\_DATE)**:

Возможен запрос, при котором существует необходимость отфильтровать данные по **ID поставщика** и **дате** поставки. Поле **PART\_DATE** обернуто в **TRUNC(…)**, т.к. не всегда бывает нужным знать время.

1. Составной функциональный индекс **(VENDOR\_ID + PART\_ID + PART\_DATE)**:

Возможен запрос, при котором существует необходимость отфильтровать данные по **ID поставщика**, **номеру партии** и **дате** поставки. Поле **PART\_DATE** обернуто в **TRUNC(…)**, т.к. не всегда бывает нужным знать время.

1. Составной функциональный индекс **(PART\_ID + PART\_QUANTITY + PART\_DATE)**:

Возможен запрос, при котором существует необходимость отфильтровать данные **по ID поставщика**, **количество поставляемого товара**, **дате поставки**. Поле **PART\_DATE** обернуто в **TRUNC(…)**, т.к. не всегда бывает нужным знать время.

**ЗАМЕТКА:**

К домашней работе приложен файл **.sql**, в котором расписаны все ограничения по таблицам.