

Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический университет
Петра Великого»
Институт компьютерных наук и технологий
Кафедра Компьютерных систем и программных технологий

ОТЧЕТ
по дисциплине «Базы данных»
GUI-приложение

Студент гр.43501/4
Алексеев Д.М.

Преподаватель
Мяснов А.В.

Санкт-Петербург
2016 год

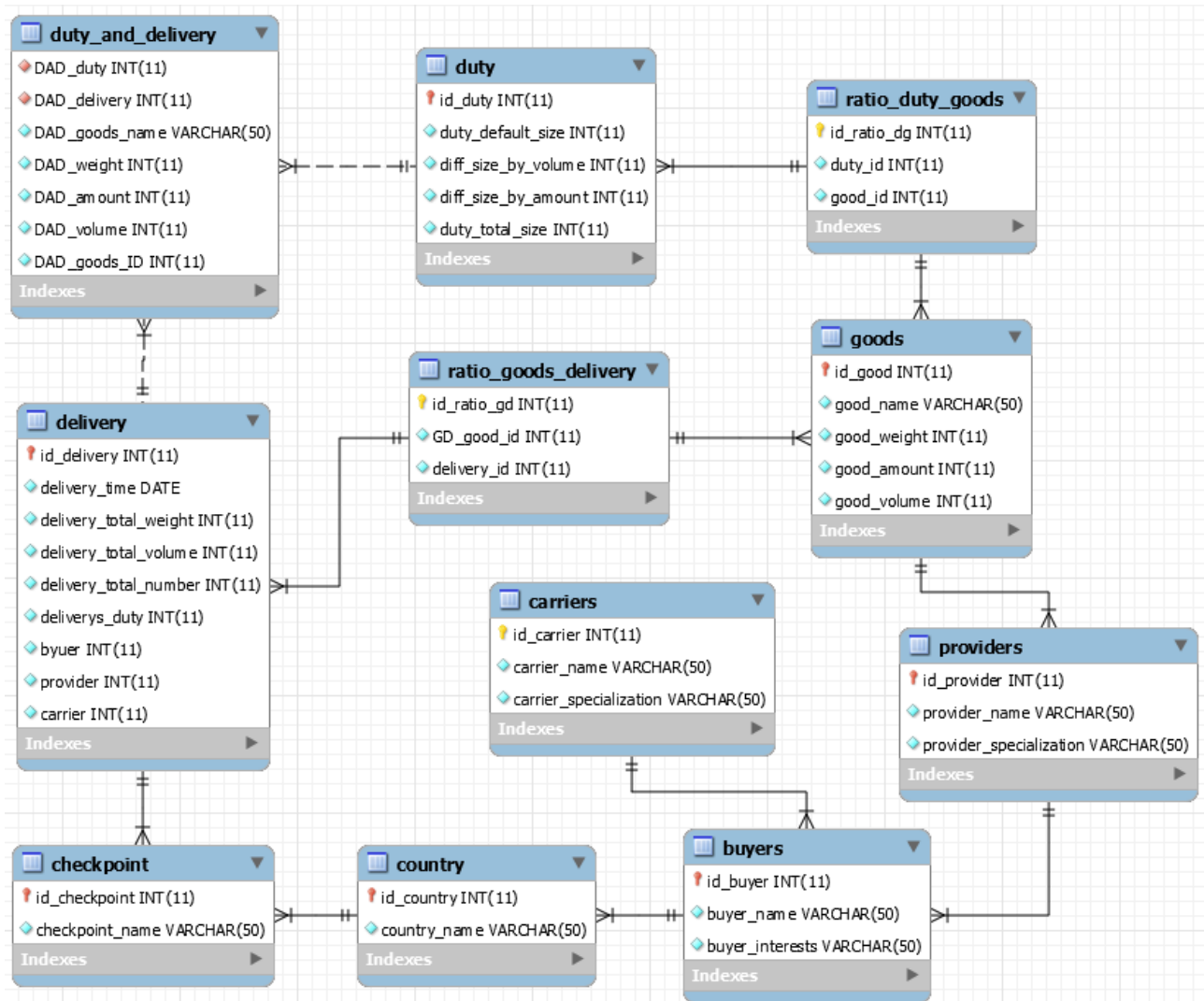
1. Цель работы

Создать GUI-приложение, использующее OLAP-Cube для обработки поступающего запроса на получение 10 самых востребованных товаров по определённому критерию.

2. Программа работы

- Создать две БД: нормализованную и ДЕНормализованную; от последней добиться высокого быстродействия для указанного запроса.
- Написать GUI-приложение, которое может общаться с двумя созданными БД.
- Получение данных проводить от ДЕНормализованной БД; изменение данных – в нормализованной и ДЕНормализованной БД.

На рис. ниже приведена ДЕНормализованная схема БД (в нормализованной отсутствует таблицу duty_and_delivery):



3. Выполнение задания

Добавим 2 триггера к нашим БД:

В **нормализованной** БД при изменении пошлины на товар

(check_duty.sql)/характеристик товара (check_good_params.sql) (т.е. его веса,

количества или объёма) последуют изменения в поставке, которая этим товаром занята (если она ещё не отправлена, т.е. имеет статус del_status = 'wait').

В денормализованной БД помимо этого мы также будем менять содержимое таблицы duty_and_delivery с помощью триггера, который расширяет возможности check_good_parametres - DAD_trigger.sql и вызываемых им ХП.

- **Пошлина**

Триггер (check_duty.sql):

```
delimiter //
DROP TRIGGER IF EXISTS `check_duty`//
CREATE TRIGGER `check_duty` AFTER UPDATE ON `duty` FOR EACH ROW
BEGIN

SET @addition = NEW.duty_total_size-OLD.duty_total_size;
SET @current_good_id = (SELECT id_good FROM goods, ratio_duty_goods, duty WHERE
goods.id_good = ratio_duty_goods.good_id AND ratio_duty_goods.duty_id =
duty.id_duty AND
ratio_duty_goods.duty_id = OLD.id_duty);

CALL change_duty_in_delivery(@current_good_id, @addition);

END//
delimiter ;
```

Вызываемая триггером процедура (change_duty_in_delivery.sql):

```
delimiter //
DROP PROCEDURE IF EXISTS `change_duty_in_delivery`//
CREATE DEFINER=`root`@`localhost` PROCEDURE `change_duty_in_delivery`(IN
`changed_good_ID` INT(11), IN `addition` INT(11))

BEGIN

UPDATE delivery SET delivery.deliveryys_duty = delivery.deliveryys_duty + addition
#Вложенный запрос IN (SELECT T FROM (SELECT )AS T ) - для защиты от error 1093
WHERE delivery.del_status = 'wait' AND delivery.id_delivery IN(SELECT
T.id_delivery
FROM(SELECT delivery.id_delivery FROM goods, ratio_goods_delivery, delivery
WHERE
(delivery.id_delivery = ratio_goods_delivery.delivery_id AND
ratio_goods_delivery.GD_good_id = changed_good_ID)) AS T);

END//
delimiter ;
```

- **Вес/количество/объём** (CHOOSEN_VALUE в коде – это weight, amount или volume в зависимости от изменяемой величины)

Триггер (check_good_parametres.sql):

```
delimiter //
DROP TRIGGER IF EXISTS `check_weight`//
CREATE TRIGGER `check_weight` AFTER UPDATE ON `goods` FOR EACH ROW
BEGIN

SET @addition_weight = NEW.good_weight-OLD.good_weight;
SET @addition_amount = NEW.good_amount-OLD.good_amount;
SET @addition_volume = NEW.good_volume-OLD.good_volume;
```

```

SET @current_good_id = (SELECT id_good FROM goods, ratio_goods_delivery,
delivery WHERE
goods.id_good = ratio_goods_delivery.GD_good_id AND
ratio_goods_delivery.delivery_id = delivery.id_delivery AND
ratio_goods_delivery.GD_good_id = OLD.id_good);

CALL change_weight_in_delivery(@current_good_id, @addition_weight);
CALL change_amount_in_delivery(@current_good_id, @addition_amount);
CALL change_volume_in_delivery(@current_good_id, @addition_volume);

END//
delimiter ;

```

Вызываемые триггером процедуры (с CHOOSSEN_VALUE):

```

delimiter //

DROP PROCEDURE IF EXISTS `change_CHOOSSEN_VALUE_in_delivery`//
CREATE DEFINER=`root`@`localhost` PROCEDURE
`change_CHOOSSEN_VALUE_in_delivery`(IN `changed_good_ID` INT(11), IN `addition`
INT(11))

BEGIN

UPDATE delivery SET delivery.delivery_total_CHOOSSEN_VALUE =
delivery.delivery_total_CHOOSSEN_VALUE + addition
#Вложенный запрос IN (SELECT T FROM (SELECT )AS T ) - для защиты от error 1093
WHERE delivery.del_status = 'wait' AND delivery.id_delivery IN(SELECT
T.id_delivery
FROM(SELECT delivery.id_delivery FROM goods, ratio_goods_delivery, delivery
WHERE
(delivery.id_delivery = ratio_goods_delivery.delivery_id AND
ratio_goods_delivery.GD_good_id = changed_good_ID)) AS T);

END//

delimiter ;

```

Приведённые ниже триггер и ХП характерны только для ДЕНормализованной схемы!

Триггер для duty_and_delivery (trigger_DAD.sql):

```

delimiter //
DROP TRIGGER IF EXISTS `trigger_DAD`//
CREATE TRIGGER `trigger_DAD` AFTER UPDATE ON `goods` FOR EACH ROW
BEGIN

SET @addition_weight = NEW.good_weight-OLD.good_weight;
SET @addition_amount = NEW.good_amount-OLD.good_amount;
SET @addition_volume = NEW.good_volume-OLD.good_volume;
SET @current_good_id = (SELECT id_good FROM goods, ratio_goods_delivery,
delivery WHERE
goods.id_good = ratio_goods_delivery.GD_good_id AND
ratio_goods_delivery.delivery_id = delivery.id_delivery AND
ratio_goods_delivery.GD_good_id = OLD.id_good);

CALL update_DAD(@current_good_id, @addition_weight, @addition_amount,
@addition_volume);
CALL change_weight_in_delivery(@current_good_id, @addition_weight);
CALL change_amount_in_delivery(@current_good_id, @addition_amount);
CALL change_volume_in_delivery(@current_good_id, @addition_volume);

END//
delimiter ;

```

Хранимая процедура для duty_and_delivery (update_DAD):

```
delimiter //

DROP PROCEDURE IF EXISTS `update_DAD`//
CREATE DEFINER=`root`@`localhost` PROCEDURE `update_DAD`(IN `changed_good_ID`
INT(11), IN `addition_weight` INT(11)
, IN `addition_amount` INT(11), IN `addition_volume` INT(11))

BEGIN

UPDATE duty_and_delivery SET duty_and_delivery.DAD_weight =
duty_and_delivery.DAD_weight + addition_weight,
duty_and_delivery.DAD_amount = duty_and_delivery.DAD_amount + addition_amount,
duty_and_delivery.DAD_volume = duty_and_delivery.DAD_volume + addition_volume
#Вложенный запрос IN (SELECT T FROM (SELECT )AS T ) - для защиты от error 1093
WHERE duty_and_delivery.DAD_delivery IN(SELECT T.id_delivery
FROM(SELECT delivery.id_delivery FROM duty, duty_and_delivery, delivery WHERE
(delivery.id_delivery = duty_and_delivery.DAD_delivery AND
duty_and_delivery.DAD_goods_ID = changed_good_ID
AND delivery.del_status = 'wait')) AS T);

END//

delimiter ;
```

Само приложение:

10 самых востребованных товаров за указанное время

	Название товара	Оценка по критерию
1	Good_4507	7284
2	Good_901	7209
3	Good_4791	7097
4	Good_4370	6991
5	Good_2288	6975
6	Good_1688	6961
7	Good_2289	6895
8	Good_919	6852
9	Good_1825	6815
10	Good_2056	6804

Критерий выбора:

☐ Пошлина
☒ Вес
☐ Количество
☐ Объём

Изменение данных о товаре

Изменяемая величина:

☒ Пошлина
☐ Вес
☐ Количество
☐ Объём

ID изменяемого критерия:

0

Новое значение:

0

Показать текущие данные

Изменить!

Начальная дата:

2010-01-01

Конечная дата:

2015-12-30

Послать запрос!

Выберите вид:

☒ Полный вид
☐ Только верх

Кол-во товара за указанное время

Номер товара	Кол-во товара за указанное время
1	7284
2	7209
3	7097
4	6991
5	6975
6	6961
7	6895
8	6852
9	6815
10	6804

После нажатия **кнопки** «Послать запрос» мы получим 10 товаров, которые были отобраны по указанному критерию в группе «Критерий выбора». Поля для **ввода даты** имеют следующие ограничения: максимум – 10 символов; ввести символы, отличные от цифр и знака «-» не получится; также не получится ввести что-то вроде «--0-23244-» - программа такой ввод просто не примет. Полученные значения отображаются в таблице над кнопкой «Критерий выбора».

График построен с использованием бесплатно распространяемого виджета для Qt QCustomPlot. После получения данных оси ординат и абсцисс автоматически подгоняются под полученные значения. Также можно выбрать вид: посмотреть, как выглядит вся гистограмма, или только её верх (т.е. разницу между самым большим и самым маленьким столбцом).

Изменение данных о товаре позволяет нам установить новые значения для пошлины, веса, количества и объёма некоего товара. Ввести в поля «ID изменяемого критерия» и «Новое значение» можно только цифры. После нажатия на кнопку «Изменить!» мы отправим 2 запроса: один к нормализованной БД, другой – к ДЕНормализованной (т.е. данные будут совпадать в обеих схемах).

Кнопка «**Показать текущие данные**» выведет для нас таблицу с информацией, которая нужна для работы (состояние текущих критериев, а также id товара, пошлины и поставки, с которыми они связаны):

Dialog

?

×

Страница95/100

Перейти на страницу:95

Применить

	1	2	3	4	5	6	7	8
1	94001	94001	Good_4001	47	3196	9588	94001	83
2	94002	94002	Good_4002	317	22507	67521	94002	37
3	94003	94003	Good_4003	304	27056	108224	94003	52
4	94004	94004	Good_4004	57	228	912	94004	1
5	94005	94005	Good_4005	345	28290	113160	94005	30
6	94006	94006	Good_4006	70	6720	20160	94006	1
7	94007	94007	Good_4007	192	4032	8064	94007	16
8	94008	94008	Good_4008	31	434	1302	94008	1
9	94009	94009	Good_4009	191	11460	57300	94009	31
10	94010	94010	Good_4010	173	13148	39444	94010	1
11	94011	94011	Good_4011	485	17945	17945	94011	53
12	94012	94012	Good_4012	105	9660	48300	94012	51
13	94013	94013	Good_4013	398	1990	7960	94013	91
14	94014	94014	Good_4014	489	32274	129096	94014	61
15	94015	94015	Good_4015	39	273	546	94015	19
16	94016	94016	Good_4016	210	9240	9240	94016	89
17	94017	94017	Good_4017	105	315	1260	94017	21
18	94018	94018	Good_4018	460	31740	158700	94018	23
19	94019	94019	Good_4019	171	7695	15390	94019	47
20	94020	94020	Good_4020	491	31915	127660	94020	3
21	94021	94021	Good_4021	16	416	1664	94021	27

Отправляемый из клиента **запрос** к денормализованной БД на получение 10 самых востребованных товаров выглядит следующим образом:

```
inquiry_to_DB = "SELECT duty_and_delivery.DAD_goods_name, "
+ criterion +
" FROM duty, duty_and_delivery, delivery\n"
"WHERE duty.id_duty = duty_and_delivery.DAD_duty AND "
"duty_and_delivery.DAD_delivery = delivery.id_delivery\n"
"AND (DATE(delivery_time) BETWEEN "
"'" + date_from + "' AND '" + date_before + "')\n"
"GROUP BY duty_and_delivery.DAD_goods_name\n"
"HAVING " + criterion + ">0\n"
"ORDER BY " + criterion + " DESC\n"
"LIMIT 10;\n";
```

inquiry_to_db – отправляемый к БД запрос.

criterion – вставляемый критерий поиска (пошлина, объём и т.д.)

date_from/date_before – даты, составляющие промежуток поиска.

Исходный код программы, а также скрипты и данные для заполнения программы лежат в репозитории на GitHub.

4. Выводы

OLAP-Cube – это мощный инструмент аналитики, который представляет собой N-мерный объект, по которому можно сделать срезы для получения данных по указанным критериям. В нашем случае мы сделали двухмерный объект (срезы производятся по дате и критерию поиска). Для своей работы он имеет две БД: одну – в нормализованном виде (для удобства ХРАНЕНИЯ данных), а другую – в ДЕнормализованном (для быстроты ВЫПОЛНЕНИЯ запросов).

В реальной жизни частота обновления ДЕнормализованной БД может быть не после каждого обновления, а по истечении определённого срока (30 минут, 12 часов, сутки и т.д.) в зависимости от требования к актуальности предоставляемых данных и имеющейся мощности (если обновляемых данных очень много, сервер с БД вполне может требовать профилактических работ, в течении которых будут осуществлены все работы по обновлению новых).

Напоследок могу сказать, что по окончании этой работы я научился объединять C++ (в частности, Qt) и MySQL, а также писать свои драйвера для Qt (в скачиваемых дистрибутивах отсутствует поддержка MySQL, поэтому нужно писать свой драйвер).