Санкт-Петербургский государственный политехнический университет

Институт информационных технологий и управления

Кафедра компьютерных систем и программных технологий

Отчёт

по лабораторной работе № 3

“Запросы SQL-DML”

по дисциплине «Базы данных»

Работу выполнил студент гр.43501/13

Михайлов Д.М. \_\_\_\_\_

Преподаватель:

Моисеев М.Ю. \_\_\_\_\_

Санкт-Петербург

2013

1. **Цель работы**

Познакомиться с языком создания запросов управления данными SQL-DML.

1. **Программа работы**

Познакомиться с возможностями реализации обработки данных на стороне сервера БД с помощью запросов на примере общих простых задач.

Индивидуальные задания:

1. Создать запрос, который выводит 10 клиентов, бывших в данном отеле наибольшее число раз (бывших в отеле – означает у путевки есть бронирование). Продемонстрировать его корректную работу.
2. Создать запрос, который определяет рейтинг курорта через рейтинги отелей курорта. Продемонстрировать его корректную работу.
3. **Выполнение программы работы**

**3.1. 1) Сделайте выборку всех данных из каждой таблицы**

connect 'C:\Users\Danya\Documents\Database\tour\_named\tour\_named.fdb' user 'SYSDBA' password 'masterkey';

SELECT \* FROM resort;

SELECT \* FROM hotel;

SELECT \* FROM room\_info;

SELECT \* FROM booking;

SELECT \* FROM tickets;

SELECT \* FROM tour;

SELECT \* FROM review;

SELECT \* FROM tour;

SELECT \* FROM card;

SELECT \* FROM client;

**2) Сделайте выборку данных из одной таблицы при нескольких условиях**

**с использованием логических операций,LIKE,BETWEEN,IN**

SELECT r\_name, place FROM resort WHERE r\_name LIKE '%Германии%';

SELECT place FROM resort WHERE r\_id BETWEEN '15' AND '20';

SELECT name, stars FROM hotel WHERE h\_cost\_min IN ('80','75','77');

**3) Создайте в запросе вычисляемое поле**

SELECT name, MIN(f\_name) FROM client GROUP BY name;

Выводит имена клиентов с первыми в алфавитном порядке фамилиями.

**4) Сделайте выборку всех данных с сортировкой по нескольким полям**

SELECT card.card\_id, room\_info.ri\_room\_type, hotel.name

FROM card, room\_info, hotel, booking

WHERE card.card\_id = booking.CARD\_BOOKING\_ID AND room\_info.ri\_id = booking.ri\_booking\_id

AND room\_info.h\_ri\_id = hotel.h\_id

ORDER BY card.card\_id ASC, hotel.name DESC;

Делается выборка соответствующих id кард, типа комнаты и названия отеля

ASC значит что индекс будет построен по возрастанию ключей. DESC соответственно – по убыванию.

5) Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

SELECT COUNT(\*),tour\_date FROM tour WHERE tour\_date = (SELECT MIN(tour\_date) FROM tour

WHERE t\_id BETWEEN '10' AND '15')

GROUP BY tour.tour\_date;

Ищем наиболее ранние даты из диапазона

6) Сделайте выборку данных из связанных таблиц (не менее двух примеров)

/\*client\_tour\_booking\*/

CREATE VIEW client\_tour\_booking1 as

SELECT client.Name, client.Surname, tour.tour\_cost, booking.B\_duration

FROM client,tour,booking,card

WHERE client.p\_id = card.p\_card\_id AND card.t\_card\_id = tour.t\_id AND

card.card\_id = booking.card\_booking\_id

ORDER BY tour\_cost ASC;

/\*client\_room\_hotel\*/

SELECT client.name, client.surname, booking.b\_duration, Room\_info.ri\_room\_type,

hotel.name

FROM client, card, booking, room\_info, hotel

WHERE client.p\_id = card.p\_card\_id AND card.card\_id = booking.card\_booking\_id

AND booking.ri\_booking\_id = Room\_info.ri\_id AND Room\_info.h\_ri\_id = hotel.h\_id

ORDER BY booking.b\_duration;

**7) Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки .**

**посчитаем, сколько клиентов приходится на каждый тур, вывод число клиентов > 10**

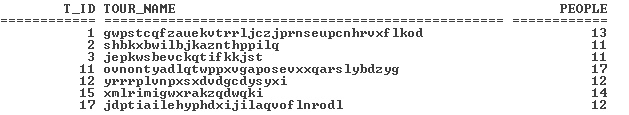
SELECT tour.t\_id, tour.tour\_name, COUNT(client.p\_id) AS people

FROM client,card,tour

WHERE client.p\_id = card.p\_card\_id AND card.t\_card\_id = tour.t\_id

GROUP BY tour.t\_id,tour.tour\_name

HAVING COUNT(client.p\_id) > 10;

****

**8)** **Придумайте и реализуйте пример использования вложенного запроса**

**Люди, заплатившие за тур(без учета остальных стоимостей т.е. не итоговая стоимость card.cost) стоимостью больше параметра**

SELECT client.name, client.surname, tour.tour\_cost, tour.t\_id, card.card\_id FROM client,card,tour WHERE card.t\_card\_id IN

(SELECT t\_id FROM tour WHERE tour\_cost > 4800) AND card.p\_card\_id = client.p\_id

ORDER BY tour.tour\_cost;

**9) С помощью оператора INSERT добавьте в каждую таблицу по одной записи**

**Добавляем последовательно новые основные таблицы**

INSERT INTO Resort VALUES ((SELECT MAX(r\_id) FROM resort) + 1, 'Валь Торанс','Французские Альпы — часть Западных Альп, расположенная во Франции.','[Климат]','[Погода]',0);

INSERT INTO HOTEL VALUES ((SELECT MAX(h\_id) FROM hotel) + 1, (SELECT MAX(r\_id) FROM resort),'Aurora','Place\_2',2,476,500,1);

INSERT INTO ROOM\_INFO VALUES ((SELECT MAX(ri\_id) FROM room\_info) + 1, (SELECT MAX(h\_id) FROM hotel),'vip',331);

INSERT INTO CLIENT VALUES ((SELECT MAX(p\_id) FROM client) + 1, 'Konstantin','Dostovalov','Alexeevich','11-AUG-1993','21961390972812559');

INSERT INTO TOUR VALUES ((SELECT MAX(t\_id) FROM tour) + 1, 'What a nice tour','So much inclusive','23-AUG-2013',1,329);

INSERT INTO CARD VALUES ((SELECT MAX(card\_id) FROM card) + 1, (SELECT MAX(p\_id) FROM client), (SELECT MAX(t\_id) FROM tour), 0);

INSERT INTO BOOKING VALUES ((SELECT MAX(card\_id) FROM card),(SELECT MAX(ri\_id) FROM room\_info),'11-DEC-2013',1);

COMMIT;

**10)** **С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию**

UPDATE hotel SET name = 'Better\_than\_home' WHERE name like 'Voyage';

UPDATE resort SET climate = 'Wet' WHERE r\_id between '30' and '60';

UPDATE client SET cl\_phone = '+79602751783' WHERE surname like 'Mikhailov';

**11) С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики**

**Удаляем наиболее старый booking**

DELETE FROM booking WHERE booking\_date = (SELECT MIN(booking\_date) FROM booking);

**Удаляем наиболее старый tour**

**Тур не удаляется из-за связей**

DELETE FROM tour WHERE t\_id = (SELECT t\_id FROM tour WHERE tour\_date = (SELECT MIN(tour\_date) FROM tour));

**12) С помощью оператора DELETE удалите записи в главной таблице,**

**на которые не ссылается подчиненная таблица (используя вложенный запрос)**

INSERT INTO CLIENT VALUES ((SELECT MAX(p\_id) FROM client) + 1, 'Konstantin','Dostovalov','Alexeevich','11-AUG-1993','21961390972812559');

DELETE FROM CLIENT WHERE p\_id = (SELECT MAX(p\_id) FROM client);

**Т.к. в client был добавлен клиент, который потом нигде не добавлялся,**

**его можно без проблем удалить.**

**3.2. 10 клиентов, бывших в данном отеле наибольшее число раз**

Делаем выборку первых 10 значений имени, фамилии клиента и вычисляемого поля, в котором считается число соотношений клиент – отель. Производится сортировка по убыванию.

connect 'C:\Users\Danya\Documents\Database\tour\_named\tour\_named\_5000.fdb' user 'SYSDBA' password 'masterkey';

SELECT FIRST 10 client.name,client.surname, COUNT(\*) AS times

FROM client, card, booking, room\_info, hotel

WHERE client.p\_id = card.p\_card\_id

AND card.card\_id = booking.card\_booking\_id

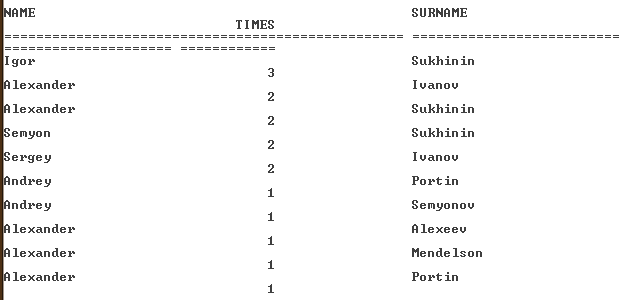
AND booking.ri\_booking\_id = room\_info.ri\_id

AND room\_info.h\_ri\_id = hotel.h\_id

AND hotel.h\_id = '294'

GROUP BY client.name,client.surname

ORDER BY times desc;

****

**Рис.1 Результат выполнения выборки.**

**3.3. Вычислить рейтинг курорта через рейтинги отелей курорта.**

Запрос создает виртуальную таблицу view для возможности передачи его поля в качестве параметра вычисления среднего значения рейтингов - результатов выборки клиент – рейтинг – отель – курорт. В качестве примера используется курорт с id = 1.

/\*Рейтинг курорта через рейтинги отелей курорта\*/

connect 'C:\Users\Danya\Documents\Database\tour\_named\tour\_named.fdb' user 'SYSDBA' password 'masterkey';

drop view ratings;

CREATE VIEW ratings (r) AS

SELECT hotel\_rating.h\_rating

FROM client, hotel\_rating, hotel, resort

WHERE client.p\_id = hotel\_rating.p\_hotel\_rating\_id

AND hotel\_rating.h\_hotel\_rating\_id = hotel.h\_id

AND resort.r\_id = hotel.r\_hotel\_id

AND resort.r\_id = '1';

select \* from ratings;

UPDATE resort SET cl\_r\_rating = (select (avg(ratings.r)) from ratings where resort.r\_id = '1');

**Для тестирования функциональности используем выходные данные текущего запроса :**

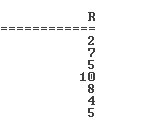


Рис.2

**и запроса вывода значения рейтинга курорта с id = 1 :**

SELECT hotel\_rating.h\_rating

FROM client, hotel\_rating, hotel, resort

WHERE client.p\_id = hotel\_rating.p\_hotel\_rating\_id

AND hotel\_rating.h\_hotel\_rating\_id = hotel.h\_id

AND resort.r\_id = hotel.r\_hotel\_id

AND resort.r\_id = '1';

SELECT resort.cl\_r\_rating

FROM resort

where resort.r\_id = '1';

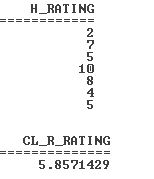


Рис.3

Получаем среднее значение всех рейтингов отелей, записанное в соответствующий курорт.

1. **Выводы**

В данной работе мы познакомились с языком создания запросов и управления данными SQL-DML.

DML = Data Manipulation Language, используется пользователями [баз данных](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) для получения, вставки, удаления или изменения данных в [базах данных](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). SQL-DML, применяемый в запросах, — декларативный язык. Это означает, что программа на SQL-DML содержит только описание требуемого результата, а не алгоритма действий.

Языки DML разделяются в основном на два типа:

* Procedural DMLs — описывают действия над данными.
* Declarative DMLs — описывают сами данные.

Выборка данных SELECT может расширяться предусловиям WHERE и постусловием HAVING для выборки по вычисляемым признакам.

Кроме этого, для определения совокупных характеристик различных отношений существует агрегирование (функции над выборкой) и группировка(организация вывода). Также удобной является возможность сортировки и, применяемая, только в этом контексте операция получения только начальных кортежей.

Следует выделить возможность создания вычисляемого поля в качестве параметра и создания виртуальной таблицы, создающейся с помощью копирования выбранных записей таблицы.

Операции изменения или удаления данных не позволяют, к примеру, удалить запись вместе с зависимыми, однако их функциональности достаточно для выполнения всех необходимых манипуляций.