Санкт-Петербургский государственный политехнический университет

Кафедра компьютерных систем и программных технологий

**Отчёт лабораторной работе №3**

**Дисциплина**: Базы данных

**Тема**: SQL-DML

Выполнил студент гр. 43501/1 А.М. Зинченко

(подпись)

Руководитель А.В. Мяснов

(подпись)

“ ” 2014 г.

Санкт - Петербург

2014

1. **Цель работы**

Выполнить различные типы запросов к БД. Изучить команды добавления, редактирования и удаления записей.

1. **Проведение работы**

Выборка всех данных из таблицы выполняется командой

Select \* from <имя таблицы>

Определим условия для выборки

Вывод tickets\_id и стоимость билетов, стоимость которых меньше 3

select tickets\_id,cost from tickets where cost<3;

Вывод tickets\_id, которые куплены на вагоны 1 и 2

select tickets\_id from tickets where vag in(1,2);

Вывод tickets\_id, у которых точка отправления в диапазоне [1,3]

select tickets\_id from tickets where st between 1 and 3;

Вывод числа результатов предыдущего запроса

select count(tickets\_id) from tickets where st between 1 and 3;

Сортировка по нескольким полям

select \* from vagons\_types order by vagons\_types\_names asc, numb\_of\_places desc;

Вывод нескольких совокупных характеристик таблиц

select count(tickets.tickets\_id),max(tickets.cost), count\_places

from tickets,(select count(places.places\_id) count\_places from places)

group by count\_places

Выберем пользователей, которые купили больше 7 билетов + число купленных этими пользователями билетов

select tickets.usr,count( tickets.tickets\_id )from tickets

group by tickets.usr having count(tickets.usr) > 7

Вывод фамилий пользователей, которые купили билеты стоимость которых < 5

select secname from users where passport in(

select usr from tickets where cost<5);

Вывод id билетов, которые входят в заказ такой, что его стоимость меньше 5

select tickets\_id from tickets where order\_id in(

select order\_id from "ORDER" where cost < 5);

Добавление в таблицу выполняется командой

insert into places (places\_id,names\_of\_places) values (-1,'абвгдейка');

или короче

insert into places values (-2,'абвгдейка2');

Пользователь под 5 номером подтвердил все свои заказы

update "ORDER" set "ORDER"."COMMIT"='yes' where owner=5;

Удалить самого старшего пользователя

delete from users where users.passport in (

select first 1 users.passport from users

order by birth)

В результате СУБД не смогла удалить этого пользователя, поскольку он есть в таблице билетов и в таблице заказов. Следует установить правило (например, приравнять полю user/owner NULL, при удалении этого пользователя).

Удалить те билеты, начальные/конечные станции которых отсутствуют в маршрутах.

delete from tickets where st not in(select st from routes);

delete from tickets where fin not in(select st from routes);

1. **Дополнительное задание**

-Вывести пять станций, с которых чаще всего отправлялись пассажиры за выбранный промежуток времени и количество отправленных пассажиров.

select

places.names\_of\_places,

count\_of\_tickets\_id

from

(select first 5

tickets.st ts,

count( tickets.tickets\_id) count\_of\_tickets\_id

from tickets

where trip in(

select trip\_id from trip where dt between '2014-10-15' and '2014-11-01')

group by tickets.st

order by count( tickets.tickets\_id) desc),places, routes

where ts=routes.routes\_id and routes.st= places.places\_id

-Удалить станции, по которым не проходят маршруты.

delete from places where places\_id not in(

select st from routes);

-Вывести десять маршрутов, сумма стоимости билетов в денежном исчислении по которым была максимальна за выбранный промежуток времени и сумму стоимости билетов по каждому маршруту.

select first 10 name\_routes.name, sum(tickets.cost) from trip, tickets,name\_routes

where trip.tm between '00:00:00' and '01:00:00' and trip.trip\_id=tickets.trip

and trip.name\_off\_routes=name\_routes.name\_routes\_id

group by name\_routes.name

order by sum(tickets.cost) desc

1. **Вывод**

В данной работе были изучены инструменты по созданию запросов к БД, удалению и модификации записей. Отметим множество критериев по выделению записей, возможность проводить различные арифметические операции над столбцами результирующего отношения, а также наличие агрегатных функций. Для операций

DML мне было удобнее использовать SQL, чем графические средства, поскольку они оказались не столь «гибкими». При выполнении дополнительного задания стало ясно, что зачастую вложенные команды select можно заменить одной такой командой с множеством параметров, что существенно сократит код, однако понять запрос будет сложнее.