Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина**: Базы данных

**Тема**: Ознакомление с основами SQL-DML

Выполнил студент гр. 43501/3 Супронович П.В.

(подпись)

Руководитель Мяснов А.В.

(подпись)

Санкт-Петербург

2016

**Программа работы**

1. Изучите SQL-DML
2. Выполните все запросы из списка стандартных запросов. Продемонстрируйте результаты преподавателю.
3. Получите у преподавателя и реализуйте SQL-запросы в соответствии с **индивидуальным** заданием. Продемонстрируйте результаты преподавателю.
4. Выполненные запросы SELECT сохраните в БД в виде представлений, запросы INSERT, UPDATE или DELETE -- в виде ХП. Выложите скрипт в Subversion.

**Список стандартных запросов**

* Сделайте выборку всех данных из каждой таблицы
* Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)
* Создайте в запросе вычисляемое поле
* Сделайте выборку всех данных с сортировкой по нескольким полям
* Создайте запрос, вычисляющий несколько совокупных характеристик таблиц
* Сделайте выборку данных из связанных таблиц (не менее двух примеров)
* Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки
* Придумайте и реализуйте пример использования вложенного запроса
* С помощью оператора INSERT добавьте в каждую таблицу по одной записи
* С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию
* С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики
* С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

**Ход работы:**

* Сделайте выборку всех данных из каждой таблицы

Этот запрос реализуется при помощи операции SELECT.

**select** **\*** **from** client**;**

**select** **\*** **from** provider**;**

**select** **\*** **from** **type;**

**select** **\*** **from** load**;**

**select** **\*** **from** transport**;**

**select** **\*** **from** insurance**;**

**select** **\*** **from** **ord;**

**select** **\*** **from** transportation**;**

**select** **\*** **from** shedule**;**

* Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)

**select** **\*** **from** shedule **where** arr\_date **in** **(**'29.11.2016' **,** '29.12.2016'**);**

- выведет расписание на 29 ноября или 29 декабря.

**select** **\*** **from** client **where** name **like** 'D%'**;** - выведет всех клиентов, чьё имя начинается с буквы «D».

**select** **\*** **from** **ord** **where** amount\_cost **between** 100 **and** 1000**;** - выведет заказы, которые стоят от 100 до 1000.

* Создайте в запросе вычисляемое поле

**select** load**.**id\_type**,** **sum(** load**.**cost **)** sum\_cost **from** load **group** **by** load**.**id\_type**;** - суммарная стоимость всех грузов определённого типа.

* Сделайте выборку всех данных с сортировкой по нескольким полям

**select** **\*** **from** transport **order** **by** capacity**,** name**;** - сортируется по вместимости, а потом по алфавиту

* Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

**select** load**.**name**,** **type.**name **from** load**,** **type** **where** load**.**id\_type **=** **type.**id**;**

- делает выборку из таблицы грузов и типов. Выводит название груза и его тип.

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

**select** **\*** **from** shedule **as** p1 **inner** **join** transportation **as** p2 **on** p1**.**id\_transportation **=** p2**.**id **and** p1**.**arr\_date **=**'29.11.2016'**;** - выводит «склеенными» таблицы транспортировок и расписания в следующем соответствии: если в расписании транспортировка производится 29 ноября 2016 – то конкатенировать данные транспортировки к данным расписания и вывести.

**select** **\*** **from** load **inner** **join** insurance **as** p1 **on** p1**.**id\_load **=** load**.**id **and** load**.**cost **>** 1000**;** - вывести напротив грузов, стоимостью больше 1000 их страховки.

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

**select** load**.**id\_type**,** **avg(**load**.**cost**)** count\_of\_type **from** load **group** **by** load**.**id\_type **having** **avg(**load**.**cost**)** **>** 50000**;** - сгруппировать типы груза и количество грузов этого типа стоимостью больше 100 по типу груза.

* Придумайте и реализуйте пример использования вложенного запроса

**select** name **from** load **where** id\_type **=** **(select** id **from** **type** **where** name **=** 'build matherials'**);** - вывести название всех грузов, тип которых стройматериалы.

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

**create** **procedure** ins\_transport**(**i **integer,** name **varchar(**20**),** capacity **int)**

**begin**

**insert** **into** transport **(**id**,** name**,** capacity**)** **values** **(:**i**,** **:**name**,** **:**capacity**);**

**end**

**create** **procedure** ins\_provider**(**i **integer,** name **varchar(**20**),** surname **varchar(**20**),** oldname **varchar(**20**),** firm **varchar(**20**),** addr **varchar(**100**),** phone\_num **varchar(**11**))**

**begin**

**insert** **into** provider **(**id**,** name**,** surname**,** oldname**,** firm**,** address**,** phone\_number**)** **values** **(:**i**,** **:**name**,** **:**surname**,** **:**oldname**,** **:**firm**,** **:**addr**,** **:**phone\_num**);**

**end**

**create** **procedure** ins\_client**(**i **int,** name **varchar(**20**),** surname **varchar(**20**),** oldname **varchar(**20**),** addr **varchar(**100**),** phone\_num **varchar(**11**))**

**begin**

**insert** **into** client **(**id**,** name**,** surname**,** oldname**,** address**,** phone\_number**)** **values** **(:**i**,** **:**name**,** **:**surname**,** **:**oldname**,** **:**addr**,** **:**phone\_num**);**

**end**

**create** **procedure** ins\_type**(**i **int,** name **varchar(**20**),** cost **float)**

**begin**

**insert** **into** **type** **(**id**,** name**,** cost**)** **values** **(:**i**,** **:**name**,** **:**cost**);**

**end**

**create** **procedure** ins\_load**(**i **int,** name **varchar(**20**),** id\_type **int,** weight **int,** **length** **float,** width **float,** height **float,** cost **float)**

**begin**

**insert** **into** load **(**id**,** name**,** id\_type**,** weight**,** **length,** width**,** height**,** cost**,** lost**)** **values** **(:**i**,** **:**name**,** **:**id\_type**,** **:**weight**,** **:**lenght**,** **:**width**,** **:**height**,** **:**cost**);**

**end**

**create** **procedure** ins\_insurance**(**i **int,** id\_load **int,** return\_cost **float)**

**begin**

**insert** **into** insurance **(**id**,** id\_load**,** return\_cost**)** **values** **(:**i**,** **:**id\_load**,** **:**return\_cost**);**

**end**

**create** **procedure** ins\_ord**(**i **int,** id\_provider **int,** id\_client **int,** id\_load **int,** quantity **int,** amount\_cost **float,** id\_insurance **int)**

**begin**

**insert** **into** **ord** **(**id**,** id\_provider**,** id\_client**,** id\_load**,** quantity**,** amount\_cost**)** **values** **(:**i**,** **:**id\_provider**,** **:**id\_client**,** **:**id\_load**,** **:**quantity**,** **:**amount\_cost**);**

**end**

**create** **procedure** ins\_transportation**(**i **int,** id\_transport **int,** id\_ord **int)**

**begin**

**insert** **into** transportation **(**id**,** id\_transport**,** id\_ord**)** **values** **(:**i**,** **:**id\_transport**,** **:**id\_ord**);**

**end**

**create** **procedure** ins\_shedule**(**i **int,** id\_transportation **int,** arr\_date tymestamp**)**

**begin**

**insert** **into** shedule **(**id**,** id\_transportation**,** arr\_date**)** **values** **(:**i**,** **:**id\_transportation**,** **:**arr\_date**);**

**end**

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

**create** **procedure** update\_load**(**id\_type **int)** **as**

**begin**

**update** load

**set**

cost **=** cost **+** 100

**where** id\_type **=** **:**id\_type**;**

**end**

Увеличивает стоимость грузов определённого типа.

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

**create** **procedure** del\_ord**(**i **int)** **as**

**begin**

**delete** **from** **ord** **where** id\_client **=** **:**i **and** amount\_cost**=(select** **min(**amount\_cost**)** **from** **ord** **where** id\_client **=** **:**i**);**

**end**

Удалит заказ определённого клиента с минимальной стоимостью.

* С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

**create** **procedure** del\_trans **as**

**begin**

**delete** **from** transportation **where** id **not** **in** **(select** id\_transportation **from** shedule**);**

**end**

**Индивидуальное задание.**

Выполнить следующие запросы.

1. Вывести 5 наиболее популярных типов грузов и общий объем перевозок каждого за заданный интервал времени.
2. Вывести поставщика, грузы от которого участвовали в максимальном количестве перевозок.
3. Удалить неиспользуемые типы грузов.

1. Вывести 5 наиболее популярных типов грузов и общий объём перевозок каждого за заданный интервал времени.

**select** **first** 5 **type.**id**, type.**name**,** **sum(ord.**quantity**)** **as** sum\_quantity **from** **type,** load**,** **ord**

**where** load**.**id **=** **ord.**id\_load **and** **type.**id **=** load**.**id\_type **group** **by** **type.**id**, type.**name **order** **by** sum\_quantity **desc;**

2. Вывести поставщика, грузы от которого участвовали в максимальном количестве перевозок.

**select** provider**.**id**,** provider**.**name**,** provider**.**surname**,** provider**.**oldname**, count(ord.**id\_provider**)** **as** counter **from** provider**,** **ord** **where** provider**.**id **=** **ord.**id\_provider **group** **by** provider**.**id**,** provider**.**name**,** provider**.**surname**,** provider**.**oldname **order** **by** counter **desc;**

3.Удалить неиспользуемые грузы.

**create** **procedure** del\_load **as**

**begin**

**delete** **from** load **where** id **not** **in** **(select** **ord.**id\_load **from** **ord)**

**end;**

**Вывод:**

Язык SQL-DML позволяет производить операции над данными, хранящимися в базе. Основные операции языка SQL-DML:

* INSERT – добавление записи в таблицу
* SELECT – выборка данных из таблицы
* UPDATE – обновление данных, находящихся в таблице
* DELETE – удаление данных из таблицы.
* JOIN – объединение таблиц для выборки
* WHERE – наложение условия на выборку
* ORDER BY - группировка выбранных данных

Комбинируя эти операции, можно создавать сложные запросы, объединяющие в себе множество условий, вычислений (таких как определение максимума, минимума, среднего, суммы и количества), группировок.

Однако необходимо задумываться о производительности используемых запросов и об их оптимизации, так как при увеличении количества записей в таблицах, увеличивается и объем выполняемых операций, что может привести к снижению производительности. Например, если запрос выполняется за 0,01 секунды, а таких кол-во запросов доходит до миллиона, стоит попробовать оптимизировать запрос или базу данных. Объединить некоторые таблицы или не выводить, или упростить сам запрос, не выводя некоторые данные.