Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: “Теория баз данных”

Лабораторная работа №5

“РАЗРАБОТКА СЛОЖНЫХ ЗАПРОСОВ К БАЗЕ ДАННЫХ.

ЗАПРОСЫ НА ОСНОВЕ НЕСКОЛЬКИХ ТАБЛИЦ.

КОРРЕЛИРОВАННЫЕ ВЛОЖЕННЫЕ ПОДЗАПРОСЫ”

Выполнил:

ст.гр. ИС/б-22

Волобуев Ю.С.

Проверила:

Лебедева М.А.

Севастополь

2019

1 ЦЕЛЬ РАБОТЫ

Выполнение лабораторной работы преследует следующие цели:

* изучить способы получения информации из нескольких таблиц;
* записать запросы, демонстрирующие выборки из нескольких таблиц с использованием оператора JOIN и без него;
* изучить способы выполнения и принцип действия рекурсивных запросов;
* научится использовать вложенные подзапросы;
* ознакомиться с возможностями построения вложенных коррелированных подзапросов с применением кванторов;
* ознакомиться с возможностью формирования отчетов в клиентском приложении.

2 ПОСТАНОВКА ЗАДАЧИ

1. Проанализировать созданную в л/р №4 базу данных.
2. Сформулировать и выполнить с помощью SQL-редактора PhpMyAdmin запросы, соединяющие две таблицы с помощью JOIN и без него.
3. Сформулировать и выполнить с помощью SQL-редактора PhpMyAdmin запросы, соединяющие более чем две таблицы с помощью JOIN и без него.
4. Продемонстрировать с помощью SQL-редактора PhpMyAdmin следующие возможности SQL:

* использование псевдонимов на примере рекурсивного запроса;
* запрос с подзапросом;
* использование агрегатных функций в подзапросе;
* подзапросы, возвращающие единственное и множественные значения;
* подзапросы, использующие вычисление;
* использование подзапросов в HAVING;

1. Ознакомится с принципом и продемонстрировать работу коррелированных подзапросов:

* привести пример соединения таблицы со своей копией;
* привести пример коррелированного запроса, использующего две разные таблицы;
* привести пример запроса с оператором EXIST;
* привести пример запроса с оператором ALL;
* привести пример запроса с оператором ANY.

1. Разработать и выполнить с помощью SQL-редактора PhpMyAdmin обязательные запросы, заданные вариантом (Таблица 1).

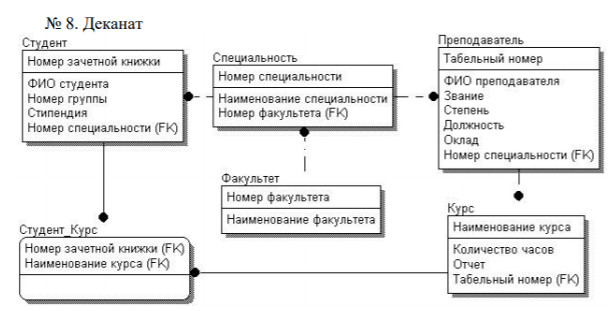


Рисунок 1 – Схема БД по варианту

Таблица 1 – Вариант задания

|  |  |
| --- | --- |
| № Варианта | Запросы к базе данных |
| 2 | 1. Вывести Ф.И.О. студентов факультета АВТ  2. Вывести Ф.И.О. студентов, которые не изучают программирование.  3. Вывести Ф.И.О. студентов ИС, получающих максимальную стипендию. |

3 ХОД РАБОТЫ

1. Сформулировать и выполнить с помощью SQL-редактора PhpMyAdmin запросы, соединяющие две таблицы с помощью JOIN и без него.

Исходные таблицы:



Рисунок 2 – Таблица Студент

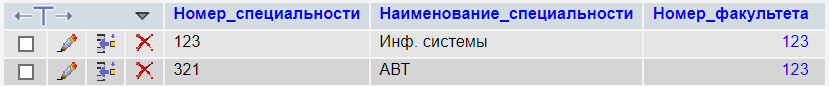


Рисунок 3 – Таблица Специальность

Вывести записи, содержащие: номер книжки, ФИО, номер группы, стипендию, номер специальности и имеющие одинаковые номера зачетки и номер факультета.

Использование JOIN:

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Номер\_зачетной\_книжки`, `ФИО`, `Номер\_группы`, `Стипендия`, `студент`.`Номер\_специальности` FROM `Студент` JOIN `Специальность` ON `студент`.`Номер\_зачетной\_книжки` = `Специальность`.`Номер\_факультета`

Без использования JOIN:

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Номер\_зачетной\_книжки`, `ФИО`, `Номер\_группы`, `Стипендия`, `студент`.`Номер\_специальности`, `специальность`.`Номер\_специальности` FROM `студент`, `специальность` WHERE `студент`.`Номер\_зачетной\_книжки` =`специальность`.`Номер\_специальности`



Рисунок 4 – Результат соединения двух таблиц

2. Сформулировать и выполнить с помощью SQL-редактора PhpMyAdmin запросы, соединяющие более чем две таблицы с помощью JOIN и без него.

Вывести записи, содержащие: номер книжки, ФИО, номер факультета, табельный номер и имеющие одинаковые номер специальности и номер факультета, а также табельный номер и номер зачетной книжки.

Использование JOIN:

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `студент`.`Номер\_зачетной\_книжки`, `студент`.`ФИО`, `специальность`.`Номер\_факультета`, `преподаватель`.`Табельный\_номер` FROM `Студент` JOIN `специальность` ON `студент`.`Номер\_специальности` =`специальность`.`Номер\_факультета` INNER JOIN `преподаватель` ON `преподаватель`.`Табельный\_номер` = `студент`.`Номер\_зачетной\_книжки`

Без использования JOIN:

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `студент`.`Номер\_зачетной\_книжки`, `студент`.`ФИО`, `специальность`.`Номер\_факультета`, `преподаватель`.`Табельный\_номер` FROM `Студент`, `Специальность`, `Преподаватель` WHERE `студент`.`Номер\_специальности`=`специальность`.`Номер\_факультета` && `преподаватель`.`Табельный\_номер` = `студент`.`Номер\_зачетной\_книжки`



Рисунок 5 – Результат соединения трех таблиц

3. Использование псевдонимов на примере рекурсивного запроса

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) one.`ФИО`, two.`ФИО`, one.`Номер\_зачетной\_книжки` FROM `Студент` AS one, `Студент` AS two WHERE one.`Стипендия` = two.`Стипендия` [AND](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) one.`Номер\_зачетной\_книжки` < two.`Номер\_зачетной\_книжки`



Рисунок 6 – Результат использования псевдонимов

4. Запрос с подзапросом + агрегатная функция

Выбрать запись где номер зачетки равен максимальному номеру специальности в таблице Специальность

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `Студент` WHERE `Номер\_зачетной\_книжки` = ([SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [MAX](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_max)(`Номер\_специальности`) FROM `специальность`)



Рисунок 7 – Результат использования подзапроса с агрегатной функцией

5. Подзапросы, использующие вычисление

Вывод студента, у которого номер зачетной книжки + 100 равняется табельному номеру преподавателя.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `студент` WHERE EXISTS( [SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `преподаватель` WHERE `преподаватель`.`Табельный\_номер` = `студент`.`Номер\_зачетной\_книжки`+100)



Рисунок 8 – Результат использование подзапросов с вычислениями

6. Использование подзапросов с HAVING

Вывод списка студентов, номер зачетки которых больше, чем среднее число номеров зачеток в этой таблице.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Номер\_зачетной\_книжки`, `ФИО` FROM `студент` HAVING `Номер\_зачетной\_книжки` > ([SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [AVG](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(`Номер\_зачетной\_книжки`) FROM `Студент`)



Рисунок 9 – Результат использования подзапросов с HAVING

7. Привести пример коррелированного запроса, использующего две разные таблицы.

Вывод список студентов, номер зачетки которых больше 100.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) DISTINCT `студент`.`Номер\_зачетной\_книжки`, `студент`.`ФИО`, `студент`.`Номер\_группы`, `специальность`.`Номер\_специальности` FROM `студент`, `специальность` WHERE `студент`.`Номер\_зачетной\_книжки` =`специальность`.`Номер\_специальности` [AND](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) `специальность`.`Номер\_специальности` > 100

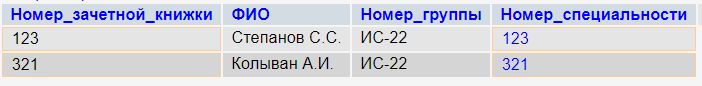


Рисунок 10 – Пример коррелированного запроса

8. Привести пример запроса с оператором ALL.

Вывести всех студентов, у которых номер зачетки больше чем у всех студентов, чей номер специальности = 123.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `студент` WHERE `Номер\_зачетной\_книжки` > ALL ([SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Номер\_зачетной\_книжки` FROM `студент` WHERE `Номер\_специальности` = 123)



Рисунок 11 – Пример использования оператора ALL

9. Привести пример запроса с оператором ANY.

Вывод студентов, совпадающих по номеру зачетки с любым табельным номером переподавателя.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `студент` WHERE `студент`.`Номер\_зачетной\_книжки` = ANY([SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Преподаватель`.`Табельный\_номер` FROM `Преподаватель`)



Рисунок 12 – Пример использования оператора ANY

**Задания по варианту**

10. Вывести Ф.И.О. студентов факультета АВТ.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `ФИО`FROM `Студент` JOIN `Специальность` ON `специальность`.`Номер\_специальности` = 321

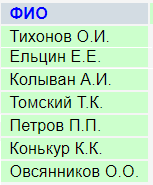


Рисунок 13 – ФИО студентов, обучающихся на АВТ

11. Вывести Ф.И.О. студентов, которые не изучают программирование.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `Студент` WHERE `Студент`.`Номер\_специальности` != ALL ([SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Специальность`.`Номер\_факультета` FROM `Специальность`)



Рисунок 15 – Студенты, не изучающие программирование

12. Вывести Ф.И.О. студентов ИС, получающих максимальную стипендию.

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Студент`.`ФИО` FROM `Студент` JOIN `Специальность` ON `Специальность`.`Наименование\_специальности` = "Инф. системы" && `Студент`.`Стипендия` >= ALL ([SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) `Студент`.`Стипендия` FROM `Студент`)



Рисунок 18 – Результат поиска студента с максимальной стипендией на ИС.

ВЫВОД

В ходе выполнения лабораторной работы были изучены способы получения информации из нескольких таблиц; записаны запросы, демонстрирующие выборки из нескольких таблиц с использованием оператора JOIN и без него; изучены способы выполнения и принцип действия рекурсивных запросов; освоены навыки использования вложенных подзапросов.