ИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)   
  
  
Факультет информатики  
Кафедра программных систем  
  
Дисциплина  
**Базы данных  
  
  
  
ОТЧЕТ**по лабораторной работе №4  
  
Вариант №14

Студент: Михайлова К.В  
Группа: 6314-020302D  
  
Преподаватель: Повова-Коварцева Д. А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2024

**Вариант № 14**

Предметная область: Бухгалтерия (расчет стипендии).

Основные предметно-значимые сущности: Студенты, Группы студентов, Результаты сдачи сессии.

Основные предметно-значимые атрибуты сущностей:

* студенты – фамилия, имя, отчество;
* группы студентов – название или номер группы;
* результаты сдачи сессии – студент, семестр, название категории (не сдал, сдал на 3, сдал на 4-5, сдал на 5).

Основные требования к функциям системы:

* назначить размер стипендии студентов за последнюю сессию в соответствие с действующими правилами;
* вывести группы, в которых студенты не получают стипендии;
* выбрать всех студентов, обучающихся по целевым договорам, сдавших на 3 или вообще не сдавших последнюю сессию, по группам;
* выбрать всех студентов, сдавших последнюю сессию на 5, по группам;
* подсчитать сумму стипендий студентов по курсам;
* вывести размер назначенной стипендии студентов, обучающихся по целевым договорам, по предприятиям;
* подсчитать сумму стипендий по группам.

Тема работы: использование подзапросов и представлений.

Цель работы: приобретение навыков использования подзапросов в команде SQL SELECT. Создание и работа с представлениями.

Содержание работы:

1. Изучение основ написания подзапросов в различных частях команды SQL SELECT.
2. Изучение основ создания виртуальных таблиц (представлений). Создание обновляемых представлений. Использование команды WITH CHECK OPTION.
3. Подготовка ответов на вопросы к лабораторной работе.
4. Подготовка отчета о проделанной работе.

Последовательность выполнения работы:

1. Придумайте и напишите SQL запросы, которые будут необходимы для предметной области (в соответствии с вариантом задания):
   1. Запрос с применением подзапроса в части WHERE команды SQL SELECT и внутри предложения HAVING.
   2. . Запрос с применением подзапроса с применением следующих операторов: ALL, EXISTS, ANY.
2. Создайте представления, основанные на запросах из пункта 1.
3. Создайте представления с выборкой, сортировкой, группировкой, левым, правым и внешним объединением.
4. Создайте обновляемые представления для всех таблиц. Проверьте работоспособность созданных представлений командами SQL: Select, Insert, Update и Delete.
5. Для обновляемого представления примените команду WITH CHECK OPTION. Объясните смысл ее применения.
6. Оформите отчет о выполнении лабораторной работы.

1. На рисунке 1 представлен пример выполнения подзапроса с использованием WHERE и HAVING. Для начала была найдена средняя стипендия, далее были объединены 3 таблицы базы данных. Затем при условии, что категория меньше максимальной, была выполнена группировка по фамилии, а потом выведены средние размеры стипендий для определенных студентов.

SELECT Студент.Фамилия AS Студент,AVG(Размер\_стипендии) AS Средняя

FROM Бухгалтерия.dbo.Категория

JOIN Бухгалтерия.dbo.Результат\_сдачи ON Категория.ID\_категории=Результат\_сдачи.ID\_категории

JOIN Бухгалтерия.dbo.Студент ON Результат\_сдачи.ID\_студента = Студент.ID\_студента

WHERE Категория.ID\_категории < (SELECT MAX(ID\_категории) FROM Категория)

GROUP BY Студент.Фамилия

HAVING AVG(Размер\_стипендии) < (SELECT AVG(Размер\_стипендии) FROM Категория)

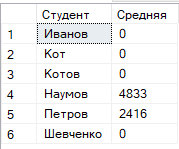


Рисунок 1 – пример выполнения

2. На рисунке 2 пример выполнения подзапроса с использованием ALL. Выводится ID студента, который больше или равен максимальному из всех ID студентов.

SELECT \* FROM Студент

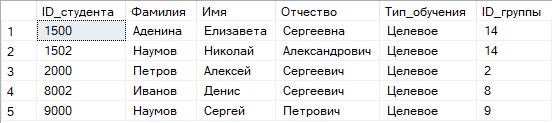
WHERE ID\_студента >= ALL (SELECT ID\_студента FROM Студент)

  
Рисунок 2 – пример выполнения

3. На рисунке 3 представлен пример выполнения подзапроса с использованием EXISTS. Здесь выводятся все студенты, которые обучаются по целевому договору.

SELECT\* FROM Студент

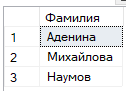
WHERE EXISTS (SELECT\* FROM Группа WHERE Студент.Тип\_обучения= 'Целевое')

  
Рисунок 3 – пример выполнения

4. На рисунке 4 пример выполнения подзапроса с использованием ANY. Здесь будут выводиться любые студенты, у которых ID группы 14.

SELECT Фамилия FROM Студент

WHERE ID\_группы = ANY (SELECT ID\_группы FROM Группа WHERE ID\_группы=14)

  
Рисунок 4 – пример выполнения

5. На рисунке 5 пример создания представления по запросу 1. Здесь также использовалось внутреннее объединение.

CREATE VIEW v AS SELECT Студент.Фамилия AS Студент,AVG(Размер\_стипендии) AS Средняя

FROM Бухгалтерия.dbo.Категория

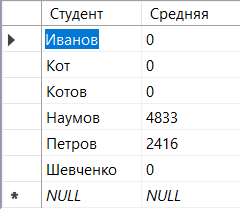
JOIN Бухгалтерия.dbo.Результат\_сдачи ON Категория.ID\_категории=Результат\_сдачи.ID\_категории

JOIN Бухгалтерия.dbo.Студент ON Результат\_сдачи.ID\_студента = Студент.ID\_студента

WHERE Категория.ID\_категории < (SELECT MAX(ID\_категории) FROM Категория)

GROUP BY Студент.Фамилия

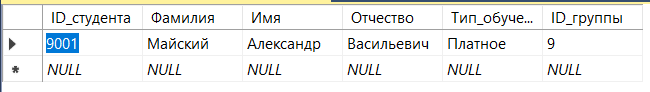
HAVING AVG(Размер\_стипендии) < (SELECT AVG(Размер\_стипендии) FROM Категория)

  
Рисунок 5 – пример выполнения

6. На рисунке 6 пример создания представления по запросу 2.

CREATE VIEW v1 AS SELECT \* FROM Студент

WHERE ID\_студента >= ALL (SELECT ID\_студента FROM Студент)

  
Рисунок 6 – пример выполнения

7. На рисунке 7 пример создания представления по запросу 3.

CREATE VIEW v2 AS SELECT\* FROM Студент

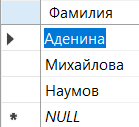
WHERE EXISTS (SELECT\* FROM Группа WHERE Студент.Тип\_обучения= 'Целевое')

  
Рисунок 7 – пример выполнения

8. На рисунке 8 пример создания представления по запросу 4.

CREATE VIEW v3 AS SELECT Фамилия FROM Студент

WHERE ID\_группы = ANY (SELECT ID\_группы FROM Группа WHERE ID\_группы=14)

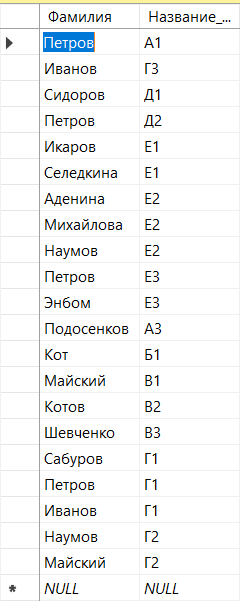
  
Рисунок 8 – пример выполнения

9. На рисунке 9 пример создания представления с левым внешним объединением.

CREATE VIEW v4 AS SELECT Студент.Фамилия, Группа.Название\_номер

FROM Бухгалтерия.dbo.Студент

LEFT OUTER JOIN Бухгалтерия.dbo.Группа ON Студент.ID\_группы=Группа.ID\_группы

  
Рисунок 9 – пример выполнения

10. На рисунке 10 пример создания представления с правым внешним объединением.

CREATE VIEW v5 AS SELECT Студент.Имя, Группа.ID\_группы

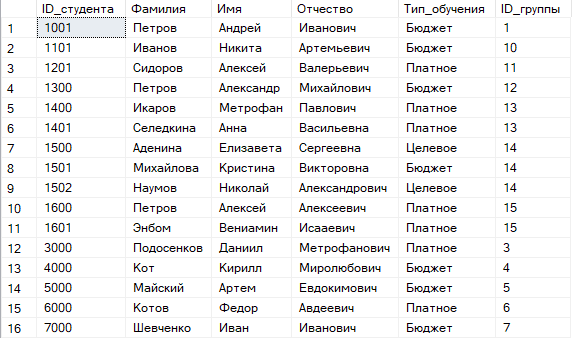
FROM Бухгалтерия.dbo.Студент

RIGHT OUTER JOIN Бухгалтерия.dbo.Группа ON Студент.ID\_группы=Группа.ID\_группы

  
Рисунок 10 – пример выполнения

11. На рисунке 11 пример проверки работоспособности представления Student командой SELECT.

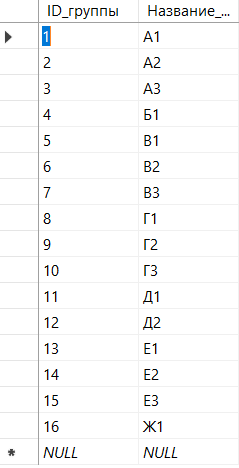
SELECT\* FROM Student

  
Рисунок 11 – пример выполнения

12. На рисунке 12 представлена проверка представления Groupp с помощью команды INSERT.

INSERT INTO Бухгалтерия.dbo.Groupp (ID\_группы, Название\_номер)

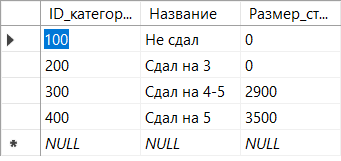
VALUES (16, 'Ж1')

  
Рисунок 12 – пример выполнения

13. На рисунке 13 продемонстрирована работоспособность представления Category с помощью команды UPDATE.

UPDATE Бухгалтерия.dbo.Category

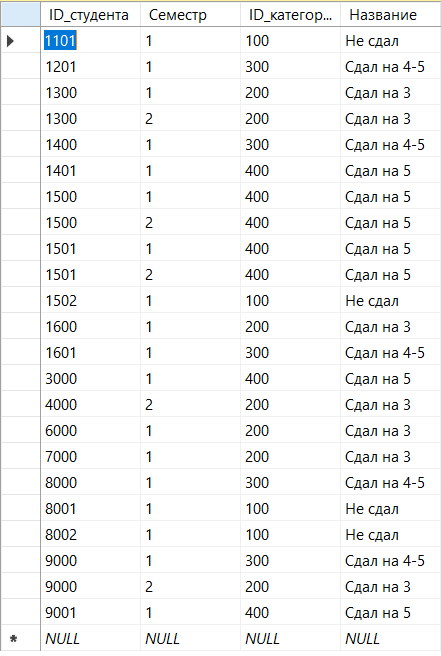
SET Размер\_стипендии=Размер\_стипендии/5

  
Рисунок 13 – пример выполнения

14. На рисунке 14 пример выполнения удаления с помощью DELETE из представления Result.

DELETE FROM Бухгалтерия.dbo.Result

WHERE ID\_студента='1001'

  
Рисунок 14 – пример выполнения

15. На рисунке 15 пример обновляемого представления к которому была применена команда WITH CHECK OPTION.

CREATE VIEW v6

AS SELECT\* FROM Студент

WHERE (ID\_группы) BETWEEN 1 AND 5

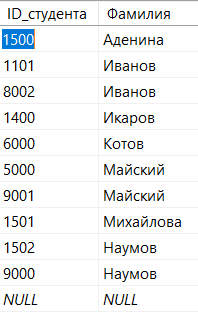
WITH CHECK OPTION

  
Рисунок 15 – пример выполнения

16. На рисунке 16 продемонстрировано представление с сортировкой.

CREATE VIEW v7 AS SELECT TOP(10) ID\_студента, Фамилия FROM Студент

ORDER BY Фамилия

  
Рисунок 16 – пример выполнения