**Технологическая карта № 5**

**Наименование:** Групповые операции. Агрегатные функции.

**1. Цель работы:**

1.1. Сформировать навыки и умения реализации запроса с использованием групповых операций и агрегатных функций.

**2. Подготовка к работе:**

3.1. Изучить предложенную литературу.

3.2. Подготовить бланк отчёта.

**3. Литература:**

2.1. Фуфаев Э.В., Фуфаев Д.Э**.** Разработка и эксплуатация удаленных баз данных: учебник для студ. сред. проф. образования / под ред. Э.В. Фуфаев, Д.Э. Фуфаев. – Москва: Издательский центр «Академия», 2008. – 256 с.

2.2. Кунтаева Г.Х. Конспект лекций по дисциплине: «Разработка и эксплуатация удаленных баз данных».

**4. Перечень необходимого оборудования:**    
4.1. Персональный IBM PC.

**5. Задание на работу:**

5.1. Вывести среднюю оценку по таблице Успеваемость для студента с номером ‘01’.

5.2. Определить в таблице Студент количество студентов в таблице.

5.3. Определить для каждого студента по таблице Успеваемость минимальную оценку.

5.4. Вывести наименьшую оценку полученную студентом ‘01’ по каждому предмету.

5.5. Вывести по алфавиту список фамилий преподавателей.

5.6. По таблице Успеваемость определить среднюю оценку в каждой группе.

5.7. По таблице Успеваемость определить количество оценок в каждой группе.

5.8. По таблице Изучение определить суммарное количество часов по предметам в каждой группе.

5.9. По таблице Изучение определить суммарное количество часов по предметам с vidz =’пр’ в каждой группе.

5.10. Определить количество групп, изучающих предмет с кодом ‘01’.

5.11. Определите количество преподавателей на каждой кафедре.

5.12 Определить сумму часов и количество предметов с видом занятия vidz =’лек’ по каждому преподавателю. Вывести список преподавателей, кол-во часов и кол-во предметов.

5.13 Определить максимальный проходной балл среди студентов групп 101, 102 и 105. Соответствующе подписать поле, в которое выводится результат.

5.14 Определить сумму часов по предметам для преподавателей, у которых табельный номер начинается с цифр «1», «2» и «4». Отсортировать результат по убыванию по полю табельный номер преподавателя.

**6. Порядок проведения лабораторной работы:**

6.1. Получить допуск к работе.

6.2. Выполнить задания в соответствие с приложением.

6.3. Ответить на контрольные вопросы.

6.4. Оформление отчета.

**7. Содержание отчета:**

7.1. Наименование и цель работы.

7.2. Запись результата выполненных заданий в отчёт.

7.3. Ответы контрольные вопросы.

7.4. Выводы о проделанной работе.

**8. Контрольные вопросы**:

8.1. Для чего используют агрегатные функции?

* 1. С помощью какого оператора можно сгруппировать данные?
  2. В каком случае используют оператор Having?
  3. Какой оператор используют для сортировки данных?

**Приложение**

*1. Агрегатные функции*

Запросы могут производить обобщенное групповое значение полей точно также как и значение одного поля. Это делает с помощью агрегатных функций. Агрегатные функции производят одиночное значение для всей группы таблицы. Имеется список этих функций:

* COUNT производит номера строк или не-NULL значения полей которые выбрал запрос.
* SUM производит арифметическую сумму всех выбранных значений данного поля.
* AVG производит усреднение всех выбранных значений данного поля.
* MAX производит наибольшее из всех выбранных значений данного поля.
* MIN производит наименьшее из всех выбранных значений данного поля.

Только числовые поля могут использоваться с SUM и AVG. С COUNT, MAX, и MIN, могут использоваться и числовые или символьные поля.

*Задание 1.*Вывести среднюю оценку по таблице Успеваемость для студента с номером ‘01’

USE uch\_proc

*SELECT* AVG(ozenka) *FROM* uspevaemost

*where* ns='01'

Функция COUNT несколько отличается от всех. Она считает число значений в данном столбце, или число строк в таблице. Когда она считает значения столбца, она используется с DISTINCT чтобы производить счет чисел различных значений в данном поле.

*Задание 2.*Определить в таблице Студент количество студентов в таблице

USE uch\_proc

*SELECT* count(ns) *FROM* student

Чтобы подсчитать общее число строк в таблице, используйте функцию COUNT со звездочкой вместо имени поля, как в следующем примере:

USE uch\_proc

*SELECT* count(\*) *FROM* student

*2. Групповые операции*

Предложение GROUP BY позволяет вам определять подмножество значений в особом поле в терминах другого поля, и применять функцию агрегата к подмножеству. Это дает вам возможность объединять поля и агрегатные функции в едином предложении SELECT.

*Задание 3.*Определить для каждого студента по таблице Успеваемость минимальную оценку.

USE uch\_proc

*SELECT* ns, min(ozenka) as 'наихудшая'

*FROM* uspevaemost

*group by* ns

GROUP BY применяет агрегатные функции независимо от серий групп, которые определяются с помощью значения поля в целом. В этом случае, каждая группа состоит из всех строк с тем же самым значением поля ns, и MIN функция применяется отдельно для каждой такой группы. Это значение поля, к которому применяется GROUP BY, имеет, по определению, только одно значение на группу вывода, также как это делает агрегатная функция. Результатом является совместимость, которая позволяет агрегатам и полям объединяться таким образом.

Вы можете также использовать GROUP BY с многочисленными полями.

*Задание 4.* Совершенствуя вышеупомянутый пример далее, предположим что вы хотите увидеть наименьшую оценку полученную каждым студентом по каждому предмету. Чтобы сделать это, вы должны сгруппировать таблицу uspevaemost по студентам и кодам предметов, и применить функцию MIN к каждой такой группе, подобно этому:

USE uch\_proc

*SELECT* ns, min(ozenka) as 'наихудшая' , kp

*FROM* uspevaemost

*group by* ns, kp

*3.Предложение HAVING*

Предположим, что в предыдущем примере, вы хотели бы увидеть MIN оценку только студента ‘01’ Вы не сможете использовать агрегатную функцию в предложении WHERE

Вы не сможете сделать что-нибудь подобно следующему:

USE uch\_proc

*SELECT* ns, min(ozenka) as 'наихудшая' , kp

*FROM* uspevaemost

*group by* ns, kp

*WHERE* ns = ‘01’

Предложение HAVING определяет критерии, используемые чтобы удалять определенные группы из вывода, точно также как предложение WHERE делает это для индивидуальных строк.

*Задание 5.* Вывести наименьшую оценку полученную студентом ‘01’ по каждому предмету.

USE uch\_proc

*SELECT* ns, min(ozenka) as 'наихудшая' , kp

*FROM* uspevaemost

*group by* ns, kp

*HAVING* ns = ‘01’

*4. Упорядочение вывода полей.**Команда ORDER BY*

Таблицы - это неупорядоченные наборы данных, и данные которые выходят из их, не обязательно появляются в какой-то определенной последовательности. SQL использует команду ORDER BY чтобы позволять вам упорядочивать ваш вывод. Эта команда упорядочивает вывод запроса согласно значениям в том или ином количестве выбранных столбцов. Многочисленные столбцы упорядочиваются один внутри другого, также как с GROUP BY, и вы можете определять возрастание ( ASC ) или убывание ( DESC ) для каждого столбца. По умолчанию установлено - возрастание.

Обратите внимание что, во всех случаях, столбцы которые упорядочиваются должны быть указаны в выборе SELECT.

*Задание 6.* Вывести по алфавиту список фамилий преподавателей

USE uch\_proc

*SELECT* fio

*FROM* prepodavatel

*order by* fio

ORDER BY может кроме того, использоваться с GROUP BY для упорядочения групп. Если это так, то ORDER BY всегда приходит последним. Вот пример из Задания 4 с добавлением предложения ORDER BY. Перед группированием вывода, порядок студентов был произвольным.

*Задание 7.* Вывести по порядку номера студентов в задании 4

USE uch\_proc

*SELECT* ns, min(ozenka) as 'наихудшая' , kp

*FROM* uspevaemost

*group by* ns, kp

order by ns