Учреждение образования “БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ”

Лабораторная работа № 6. Группировка данных

Выполнил:

студент 2 курса, 7 группы

Бобрович Глеб Сергеевич

Минск 2022ПроТТеоре

|  |  |
| --- | --- |
| **Задание** |  |
| 1.На основе таблицы **AUDITORIUM** разработать SELECT-запрос, вычисляющий максимальную, минимальную и среднюю вместимость аудиторий, суммарную вместимость всех аудиторий и общее количество аудиторий.  2.На основе таблиц **AUDITORIUM** и **AUDITORIUM\_TYPE** разработать запрос, вычисляющий для каждого типа аудиторий максимальную, минимальную, среднюю вместимость аудиторий, суммарную вместимость всех аудиторий и общее количество аудиторий данного типа.  Результирующий набор должен содержать столбец с наименованием типа аудиторий (столбец **AUDITORIUM\_TYPE.AU-DITORIUM\_TYPENAME**) и столбцы с вычисленными величинами. Использовать внутреннее соединение таблиц, секцию GROUP BY и агрегатные функции.  3.Разработать запрос на основе таблицы **PROGRESS**, который содержит количество экзаменационных оценок в заданном интервале. При этом учесть, что сортировка строк должна осуществляться в порядке, обратном величине оценки; сумма значений в столбце **количество** должна быть равна количеству строк в таблице **PROGRESS**.  Использовать подзапрос в секции FROM, в подзапросе применить GROUP BY, сортировку осуществить во внешнем запросе. В секции GROUP BY, в SELECT-списке подзапроса и в ORDER BY внешнего запроса применить CASE. | | **Задание 1**  USE ЛР6  SELECT min(AUDITORIUM\_CAPACITY) [Мин. вмест.],  max(AUDITORIUM\_CAPACITY) [Макс. вмест.],  avg(AUDITORIUM\_CAPACITY) [Средняя вмест.],  count (\*) [Кол-во аудиторий],  sum(AUDITORIUM\_CAPACITY) [Суммарное кол-во]  FROM AUDITORIUM;  **Задание 2**  USE ЛР6  SELECT AUDITORIUM.AUDITORIUM\_TYPE,  min(AUDITORIUM\_CAPACITY) [Мин. вмест.],  max(AUDITORIUM\_CAPACITY) [Макс. вмест.],  avg(AUDITORIUM\_CAPACITY) [Средняя вмест.],  count (\*) [Кол-во аудиторий],  sum(AUDITORIUM\_CAPACITY) [Суммарное кол-во]  FROM AUDITORIUM Inner Join AUDITORIUM\_TYPE  On AUDITORIUM.AUDITORIUM\_TYPE = AUDITORIUM\_TYPE.AUDITORIUM\_TYPE  GROUP BY AUDITORIUM.AUDITORIUM\_TYPE;    **Задание 3**  USE ЛР6  SELECT \*  FROM(SELECT CASE  WHEN NOTE = 10 then '10'  WHEN NOTE between 8 and 9 then '8-9'  WHEN NOTE between 6 and 7 then '6-7'  WHEN NOTE between 4 and 5 then '4-5'  END [Диапазон],  COUNT(\*) as [Кол-во]  FROM PROGRESS GROUP BY CASE  WHEN NOTE = 10 then '10'  WHEN NOTE between 8 and 9 then '8-9'  WHEN NOTE between 6 and 7 then '6-7'  WHEN NOTE between 4 and 5 then '4-5'  END) AS T  ORDER BY Case[Диапазон]  WHEN '10' then 4  WHEN '8-9' then 3  WHEN '6-7' then 2  WHEN '4-5' then 1  ELSE 0  END; | |
| 4. Разработать SELECT-запроса на основе таблиц **FACULTY**, **GROUPS**, **STUDENT** и **PROGRESS**, который содержит среднюю экзаменационную оценку для каждого курса каждой специальности. Строки отсортировать в порядке убывания средней оценки.  При этом следует учесть, что средняя оценка должна рассчитываться с точностью до двух знаков после запятой. Использовать внутреннее соединение таблиц, агрегатную функцию AVG и встроенные функции CAST и ROUND.  Переписать SELECT-запрос, разработанный в задании 4 так, чтобы в расчете среднего значения оценок использовались оценки только по дисциплинам с кодами **БД** и **ОАиП**. Использовать WHERE. | | **Задание 4**  USE ЛР6  SELECT FACULTY.FACULTY\_NAME [Факультет],  GROUPS.PROFESSION [Специальность],  STUDENT.IDGROUP [Номер группы],  round(avg(cast(PROGRESS.NOTE AS float(4))), 2) [Средний балл]  FROM FACULTY INNER JOIN GROUPS  ON FACULTY.FACULTY = GROUPS.FACULTY  inner join STUDENT  ON STUDENT.IDGROUP = GROUPS.IDGROUP  inner join PROGRESS  ON PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT  GROUP BY FACULTY.FACULTY\_NAME,  GROUPS.PROFESSION,  STUDENT.IDGROUP | |
| 5. На основе таблиц **FACULTY**, **GROUPS**, **STUDENT** и **PROGRESS** разработать SELECT-запрос, в котором выводятся специальность, дисциплины и средние оценки при сдаче экзаменов на факультете ТОВ. Использовать группировку по полям FACULTY, PROFESSION, SUBJECT.  Добавить в запрос конструкцию **ROLLUP** и проанализировать результат.  **6.** Выполнить исходный SELECT-запрос п.5 с использованием **CUBE**-группировки. Проанализировать результат. | | **Задание 5**  USE ЛР6  SELECT GROUPS.FACULTY [Факультет],  GROUPS.PROFESSION [Специальность],  PROGRESS.SUBJECT [Предмет],  round(avg(cast(PROGRESS.NOTE AS float(4))), 2) [Средняя отметка]  FROM GROUPS, STUDENT, PROFESSION, PROGRESS  WHERE GROUPS.FACULTY in ('ТОВ')  GROUP BY ROLLUP (GROUPS.FACULTY,  GROUPS.PROFESSION,  PROGRESS.SUBJECT);    **Задание 6**  USE ЛР6  SELECT GROUPS.FACULTY [Факультет],  GROUPS.PROFESSION [Специальность],  PROGRESS.SUBJECT [Предмет],  round(avg(cast(PROGRESS.NOTE AS float(4))), 2) [Средняя отметка]  FROM GROUPS, STUDENT, PROFESSION, PROGRESS  WHERE GROUPS.FACULTY in ('ТОВ')  GROUP BY CUBE (GROUPS.FACULTY,  GROUPS.PROFESSION,  PROGRESS.SUBJECT) | |

|  |  |
| --- | --- |
| 7. На основе таблиц **GROUPS**, **STUDENT** и **PROGRESS** разработать SELECT-запрос, в котором определяются результаты сдачи экзаменов.  В запросе должны отражаться специальности, дисциплины, средние оценки студентов на факультете ТОВ.  Отдельно разработать запрос, в котором определяются результаты сдачи экзаменов на факультете ХТиТ.  Объединить результаты двух запросов с использованием операторов UNION и UNION ALL. Объяснить результаты.  8.Получить пересечение двух множеств строк, созданных в результате выполнения запросов пункта 8. Объяснить результат.  Использовать оператор INTERSECT.  9.Получить разницу между множеством строк, созданных в результате запросов пункта 8.  Объяснить результат.  Использовать оператор EXCEPT. | **Задание 7**  USE ЛР6  SELECT  FACULTY.FACULTY,  GROUPS.PROFESSION,  round(avg(cast(PROGRESS.NOTE AS float(4))), 2) as AVG\_NOTE  FROM GROUPS  inner join FACULTY  ON FACULTY.FACULTY = GROUPS.FACULTY  inner join STUDENT  ON STUDENT.IDGROUP = GROUPS.IDGROUP  inner join PROGRESS  ON PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT  WHERE FACULTY.FACULTY in ('ТОВ')  GROUP BY FACULTY.FACULTY, GROUPS.PROFESSION  UNION  SELECT  FACULTY.FACULTY,  GROUPS.PROFESSION,  round(avg(cast(PROGRESS.NOTE AS float(4))), 2) as AVG\_NOTE  FROM GROUPS  inner join FACULTY  ON FACULTY.FACULTY = GROUPS.FACULTY  inner join STUDENT  ON STUDENT.IDGROUP = GROUPS.IDGROUP  inner join PROGRESS  ON PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT  WHERE FACULTY.FACULTY in ('ЛХФ')  GROUP BY FACULTY.FACULTY, GROUPS.PROFESSION  **Задание 8**  USE ЛР6  Select GROUPS.PROFESSION, PROGRESS.SUBJECT,GROUPS.FACULTY,  round (avg (cast (PROGRESS.NOTE as float(4))),2) as 'средняя оценка'  from GROUPS inner join FACULTY  on GROUPS.FACULTY = FACULTY.FACULTY  inner join STUDENT  on STUDENT.IDGROUP = GROUPS.IDGROUP  inner join PROGRESS  on PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT  where GROUPS.FACULTY like 'ИДиП'  group by GROUPS.PROFESSION, PROGRESS.SUBJECT, GROUPS.FACULTY  intersect  Select GROUPS.PROFESSION, PROGRESS.SUBJECT,GROUPS.FACULTY,  round (avg (cast (PROGRESS.NOTE as float(4))),2) as 'средняя оценка'  from GROUPS inner join FACULTY  on GROUPS.FACULTY = FACULTY.FACULTY  inner join STUDENT  on STUDENT.IDGROUP = GROUPS.IDGROUP  inner join PROGRESS  on PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT  where GROUPS.FACULTY like 'ИТ'  group by GROUPS.PROFESSION, PROGRESS.SUBJECT, GROUPS.FACULTY  **Задание 9**  USE ЛР6  SELECT g.PROFESSION, p.SUBJECT, round(avg(cast(p.NOTE as float(4))),2) [Средняя оценка]  From FACULTY f inner join GROUPS g  on f.FACULTY = g.FACULTY  inner join STUDENT s  on g.IDGROUP = s.IDGROUP  inner join PROGRESS p  on s.IDSTUDENT = p.IDSTUDENT  WHERE g.FACULTY='ИДиП'  GROUP BY g.PROFESSION, f.FACULTY, p.SUBJECT  EXCEPT  SELECT g.PROFESSION, p.SUBJECT, round(avg(cast(p.NOTE as float(4))),2) [Средняя оценка]  From FACULTY f inner join GROUPS g  on f.FACULTY = g.FACULTY  inner join STUDENT s  on g.IDGROUP = s.IDGROUP  inner join PROGRESS p  on s.IDSTUDENT = p.IDSTUDENT  WHERE g.FACULTY='ХТиТ'  GROUP BY g.PROFESSION, f.FACULTY, p.SUBJECT    **Задание 10**  USE ЛР6  SELECT PROGRESS.SUBJECT [Предмет],  PROGRESS.NOTE [Отметка],  (SELECT COUNT(\*) FROM STUDENT, PROGRESS where PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT ) [Кол-во студентов]  FROM PROGRESS  GROUP BY PROGRESS.NOTE, PROGRESS.SUBJECT  HAVING PROGRESS.NOTE > 6 |
| 10.На основе таблицы **PROGRESS** определить для каждой дисциплины количество студентов, получивших оценки 8 и 9.  Использовать группировку, секцию HAVING, сортировку. |  |