Отчёт по практической работе № 3-4 студента Смирнова Константин группы ПКС-303

Тема: SQL и его инструментарий. Групповые функции в SQL.

Цель: Научиться писать запросы на SQL

Выполнение работы

1. добавить компетенции «Программные решения для бизнеса» и «Web-разработка»

Изображение выглядит как текст

Автоматически созданное описание

1. добавить наше учебное заведение в список учебных заведений,

Изображение выглядит как текст

Автоматически созданное описание

1. зарегистрировать участников из нашего учебного заведения (не менее 5) в компетенции «Программные решения для бизнеса», среди них должен быть студент с фамилией, совпадающей с Вашей.

Изображение выглядит как стол

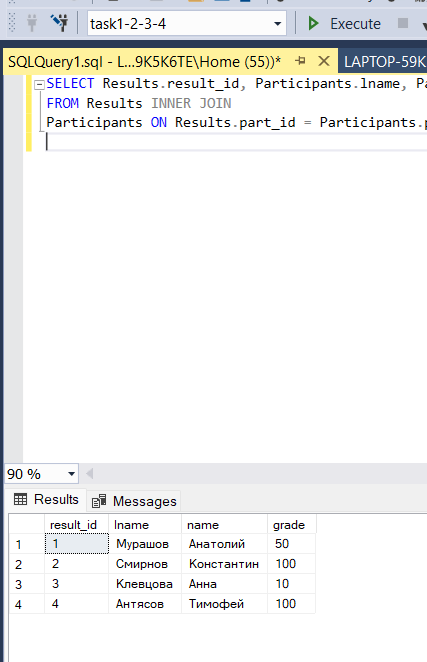
Автоматически созданное описание

1. ввести результаты выполнения модулей задания для каждого участника, \*предусмотреть, чтобы один участник зарегистрировался, но в конкурсе не принимал участия по семейным обстоятельствам,

SELECT Results.result\_id, Participants.lname, Participants.name, Results.grade

FROM Results INNER JOIN

Participants ON Results.part\_id = Participants.part\_id



1. вывести фамилии первых 5 участников, которые являются Вашими ровесниками,

SELECT Top(5) Participants.lname, Participants.name

FROM Participants

WHERE dateofbirth = '2003-11-24'

Изображение выглядит как текст

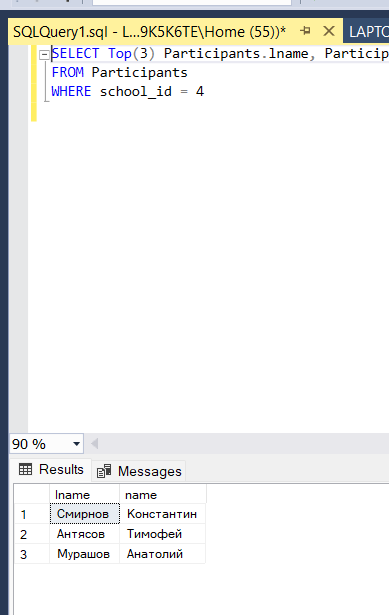
Автоматически созданное описание

1. вывести фамилии и имена первых 3 участников, которые представляли Ваше учебное заведение.

SELECT Top(3) Participants.lname, Participants.name

FROM Participants

WHERE school\_id = 4

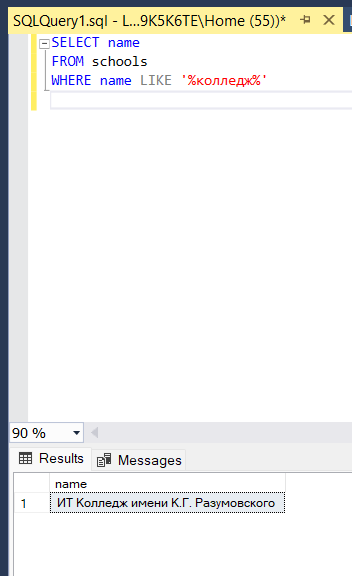


1. вывести названия учебных заведений, которые являются колледжами,

SELECT name

FROM schools

WHERE name LIKE '%колледж%'



1. вывести названия учебных заведений, студенты которых принимали участие в номинациях «Программные решения для бизнеса» и «Web-разработка», список отсортировать по названию по алфавиту,

SELECT DISTINCT Schools.name

FROM Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id INNER JOIN

Participants\_Activities ON Participants.part\_id = Participants\_Activities.part\_id

WHERE competence\_id = 4 OR competence\_id = 5

ORDER BY name ASC

Изображение выглядит как текст

Автоматически созданное описание

1. вывести названия учебных заведений, фамилии, имена, возраст студентов, которые принимали участие в компетенциях «Программные решения для бизнеса» и «Web-разработка», список отсортировать по названию учебного заведения и фамилии участников по алфавиту,

SELECT Schools.name, Participants.lname, Participants.name

FROM Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id INNER JOIN

Participants\_Activities ON Participants.part\_id = Participants\_Activities.part\_id

WHERE competence\_id = 4 OR competence\_id = 5

ORDER BY Schools.name, Participants.lname ASC

Изображение выглядит как текст

Автоматически созданное описание

1. вывести минимальный, максимальный и средний возраст участников,

SELECT MIN(DATEDIFF(YEAR, Participants.dateofbirth, GETDATE())) AS minage,

AVG(DATEDIFF(YEAR, Participants.dateofbirth, GETDATE())) AS avgage,

MAX(DATEDIFF(YEAR, Participants.dateofbirth, GETDATE())) AS maxage

FROM Participants

1. \*вывести названия учебных заведений, студенты которых не принимали участие в компетенциях «Программные решения для бизнеса» и «Web-разработка», список отсортировать по названию по алфавиту,

SELECT DISTINCT Schools.name

FROM Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id INNER JOIN

Participants\_Activities ON Participants.part\_id = Participants\_Activities.part\_id

WHERE competence\_id != 4 AND competence\_id != 5

ORDER BY name ASC

Изображение выглядит как текст

Автоматически созданное описание

1. \*вывести названия учебных заведений, студенты которых не принимали участие ни в одной из компетенций,

SELECT DISTINCT Schools.name

FROM Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id INNER JOIN

Participants\_Activities ON Participants.part\_id = Participants\_Activities.part\_id

WHERE competence\_id != 4 AND competence\_id != 5 AND competence\_id != 1 AND competence\_id != 2 AND competence\_id != 3

ORDER BY name ASC

Изображение выглядит как текст

Автоматически созданное описание

1. осуществить поиск участников по фамилии и названию учебного заведения,

SELECT Participants.name, Participants.lname, Schools.name

FROM Participants INNER JOIN

Schools ON Participants.school\_id = Schools.school\_id

WHERE Participants.lname = 'Петров' AND Schools.name = '№1'

Изображение выглядит как текст

Автоматически созданное описание

1. вывести названия учебных заведений, фамилии, имена, название компетенции, название задания (модуля), количество набранных баллов студентов, которые принимали участие в компетенциях «Программные решения для бизнеса» и «Web-разработка», список отсортировать по названию учебного заведения, названию компетенции, названию задания(модуля) и фамилии участников по алфавиту,

SELECT Schools.name, Participants.lname, Participants.name AS pname, Competencies.name AS cname, Results.module\_number, Results.grade

FROM Participants\_Activities INNER JOIN

Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id ON Participants\_Activities.part\_id = Participants.part\_id INNER JOIN

Results ON Participants.part\_id = Results.part\_id CROSS JOIN

Competencies INNER JOIN

Competencies\_Activities ON Competencies.competence\_id = Competencies\_Activities.competence\_id

WHERE Competencies.competence\_id = 4 OR Competencies.competence\_id = 5

ORDER BY Schools.name, Competencies.name, Participants.lname ASC

Изображение выглядит как текст

Автоматически созданное описание

1. \*вывести названия учебных заведений, фамилии, имена, название компетенции, общее количество набранных баллов по всем заданиям(модулям) компетенций «Программные решения для бизнеса» и «Web-разработка», список отсортировать по названию учебного заведения по алфавиту, названию компетенции по алфавиту, общему количеству набранных баллов по убыванию,

SELECT DISTINCT Schools.name, Participants.lname, Participants.name AS pname, Competencies.name AS cname, Results.module\_number, SUM(Results.grade) AS sumgrade

FROM Participants\_Activities INNER JOIN

Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id ON Participants\_Activities.part\_id = Participants.part\_id INNER JOIN

Results ON Participants.part\_id = Results.part\_id CROSS JOIN

Competencies INNER JOIN

Competencies\_Activities ON Competencies.competence\_id = Competencies\_Activities.competence\_id

WHERE Competencies.competence\_id = 4 OR Competencies.competence\_id = 5

GROUP BY Schools.name, Participants.lname, Participants.name, Competencies.name, Results.module\_number

ORDER BY Schools.name, Competencies.name ASC, sumgrade DESC

Изображение выглядит как текст

Автоматически созданное описание

1. \*вывести список участников в каждой компетенции с выводом количества баллов в порядке убывания баллов,

SELECT DISTINCT Competencies.name AS cname, Participants.lname, Participants.name AS pname, SUM(Results.grade) AS sumgrade

FROM Participants\_Activities INNER JOIN

Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id ON Participants\_Activities.part\_id = Participants.part\_id INNER JOIN

Results ON Participants.part\_id = Results.part\_id CROSS JOIN

Competencies INNER JOIN

Competencies\_Activities ON Competencies.competence\_id = Competencies\_Activities.competence\_id

WHERE Competencies.competence\_id = 4 OR Competencies.competence\_id = 5

GROUP BY Participants.lname, Participants.name, Competencies.name

ORDER BY sumgrade DESC

Изображение выглядит как текст

Автоматически созданное описание

1. \*\*\*вывести список победителей (первых 3-х участников), набравших максимальное количество баллов в каждой компетенции;

SELECT DISTINCT top(3) Competencies.name AS cname, Participants.lname, Participants.name AS pname, SUM(Results.grade) AS sumgrade

FROM Participants\_Activities INNER JOIN

Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id ON Participants\_Activities.part\_id = Participants.part\_id INNER JOIN

Results ON Participants.part\_id = Results.part\_id CROSS JOIN

Competencies INNER JOIN

Competencies\_Activities ON Competencies.competence\_id = Competencies\_Activities.competence\_id

WHERE Competencies.competence\_id = 4 OR Competencies.competence\_id = 5

GROUP BY Participants.lname, Participants.name, Competencies.name

HAVING MAX(Results.grade) = MAX(Results.grade)

ORDER BY sumgrade DESC

Изображение выглядит как текст

Автоматически созданное описание

1. \*вывести список победителей (первых 3-х участников), набравших максимальное количество баллов в компетенциях «Программные решения для бизнеса» и «Web-разработка».

SELECT DISTINCT top(3) Competencies.name AS cname, Participants.lname, Participants.name AS pname, SUM(Results.grade) AS sumgrade

FROM Participants\_Activities INNER JOIN

Schools INNER JOIN

Participants ON Schools.school\_id = Participants.school\_id ON Participants\_Activities.part\_id = Participants.part\_id INNER JOIN

Results ON Participants.part\_id = Results.part\_id CROSS JOIN

Competencies INNER JOIN

Competencies\_Activities ON Competencies.competence\_id = Competencies\_Activities.competence\_id

WHERE Competencies.competence\_id = 4 OR Competencies.competence\_id = 5

GROUP BY Participants.lname, Participants.name, Competencies.name

HAVING MAX(Results.grade) = MAX(Results.grade)

ORDER BY sumgrade DESC

Изображение выглядит как текст

Автоматически созданное описание

**Выводы о полученных практических навыках**

Я получил опыт написания запросов на SQL языке.

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_ оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись

замечание преподавателя