

Дополнительная профессиональная программа повышения квалификации «Базовый SQL для анализа данных»

Программа направлена на формирование у слушателей знаний по обработке и анализу данных с применением языка запросов SQL и основных элементов типовой среды разработки баз данных (далее – БД).

№	Наименование	Характеристика
1	Наименование и цели программы	
1.1	Наименование	«Базовый SQL для анализа данных»
1.2	Вид программы	Дополнительная профессиональная программа повышения квалификации
1.3	Цель	Изучение основ языка SQL, его применения в анализе данных, получение необходимых навыков для работы с базами данных.
1.4	Планируемые результаты (формируемые или совершенствующиеся компетенции в соответствии с указанным профстандартом)	Подготовка данных для проведения аналитических работ по исследованию больших данных; Проведение аналитического исследования с применением технологии больших данных в соответствии с требованиями заказчика; Управление качеством больших данных Профессиональный стандарт 06.042 «Специалист по большим данным» (утв. приказом Министерства труда и социальной защиты РФ от 06.07.2020 № 405н). Повышение производительности БД путем оптимизации запросов к БД Профессиональный стандарт 06.011 «Администратор баз данных» (утв. приказом Министерства труда и социальной защиты РФ от 27.04.2023 № 408н).
2	Формат реализации	
2.1	Язык обучения	Русский
2.2	Форма обучения	Очно-заочная с применением ДОТ
2.3	Объем программы	84 академических часа
2.4	Режим занятий	В течение 10 недель в удобное для слушателя время по 4 – 11 ак. часов в неделю. Вебинары – согласно расписанию занятий.
2.5	Выдаваемый документ	Удостоверение о повышении квалификации Академии ФСБ

2.6	Целевая аудитория	Аналитики данных; руководители проектов и менеджеры в области информационных технологий и аналитики; студенты и начинающие специалисты по анализу данных
2.7	Численность обучаемых по программе	Численность слушателей определяется договором об образовании, заключаемым при приеме на обучение
3	Организационно-педагогические условия	
3.1	Учебный план	см. Приложение № 1
3.2	Календарный учебный график	см. Приложение № 2
3.3	Рабочая программа (учебные предметы, курсы, дисциплины (модули))	см. Приложение № 3
3.4	Профессорско-преподавательский состав	Реализация программы обеспечивается квалифицированными научно-педагогическими кадрами, имеющими базовое образование, соответствующее профилю преподаваемой дисциплины, и систематически занимающимися научной и научно-методической деятельностью
3.5	Форма итоговой аттестации	Экзамен в форме выполнения задания
3.6	Оценочные материалы	Пример задания см. Приложение № 4. Результаты выполнения оцениваются на основании 4 критериев, с присвоением баллов по каждому критерию оценки. Максимальный балл, который может набрать слушатель за выполнение итогового задания – 100 баллов. Условием успешного завершения обучения и получения удостоверения о повышении квалификации является выполнение итогового задания с результатом не менее 40 баллов
3.7	Методические материалы	Учебная литература (учебники, пособия, книги) – предоставляется доступ к электронной библиотечной системе «Знаниум» (www.znanium.ru).
4	Контактная информация	
4.1	Руководитель программы	Мубаракшина Алсу Ирековна, директор Центра развития обучения по управлению данными АНО ДПО «Академия ПСБ» a.mubarakshina@psb-academy.ru

УЧЕБНЫЙ ПЛАН
дополнительной профессиональной программы повышения квалификации
«Базовый SQL для анализа данных»

Количество часов – 84 академических часа.

Форма обучения – очно-заочная с применением дистанционных образовательных технологий.

№ п/п	Наименование раздела/ модуля, темы	Общая трудоемкость, час.	Аудиторная работа, час.					С применением дистанционных образовательных технологий, электронного обучения, час.					Самостоятельная работа, час.	Текущий контроль успеваемости ¹	Промежуточная аттестация (форма/час.) ²	Итоговая аттестация (вид/час.)	Код компетенции
			Всего	В форме практической подготовки	В том числе			Всего	В форме практической подготовки	В том числе							
					Лекции/в интерактивной форме	Практические (семинарские) занятия/в интерактивной форме	Контактная самостоятельная работа			Лекции/в интерактивной форме	Практические (семинарские) занятия/в интерактивной форме	Контактная самостоятельная работа					
1	Введение в базы данных (часть 1)	4						3	1	2		1		1			A/03.6
2	Введение в базы данных (часть 2)	3						3	1	2		1					A/03.6
3	Простые операции с одной таблицей	16						16	5	11		5					A/03.6, A/04.6
4	Трансформация таблицы	7						7	3	4		3					A/04.6,

¹ Вид текущего контроля успеваемости: Т – тестирование.

² Вид промежуточной аттестации: ПР – практическая работа.

																	В/07.7
5	Модификация данных	4						4	2	2		2					В/07.7
6	Операции с несколькими таблицами	21						20	5	15		5			1		В/07.7
7	Расширенные возможности	8						8	2	6		2					В/03.5
8	Построение отчетов	12						11	3	8		3			1		А/04.6
9	Работа с хранилищами и витринами данных	7						7	3	4		3					В/03.5
	Итоговая аттестация	2														Э ^{3/2}	А/03.6, А/04.6, В/03.5, В/07.7
	ИТОГО:	84						79	25	54		25		1	2	2	

³ Вид итоговой аттестации: Э – экзамен.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Период обучения:		
84 ак. часа	10 недель	2,5 месяца

РАБОЧИЕ ПРОГРАММЫ РАЗДЕЛОВ / МОДУЛЕЙ / ДИСЦИПЛИН

1. ВВЕДЕНИЕ В БАЗЫ ДАННЫХ (ЧАСТЬ 1)
1.1. Входное тестирование.
1.2. Знакомство с преподавателем.
1.3. Клиент-серверная архитектура. • Понятие «клиент-серверная архитектура». • Отличие клиент-серверной архитектуры от остальных видов.
1.4. Подключение к серверу. • Установка программы базы данных и подключение к учебному серверу.
2. ВВЕДЕНИЕ В БАЗЫ ДАННЫХ (ЧАСТЬ 2)
2.1. Понимание таблиц. • Понимание структуры базы данных: таблицы, строки, столбцы. • Создание связей между таблицами. • Свойства таблиц.
2.2. Работа с таблицами. • Описание работы интерфейса SQL-сервера. • Самостоятельное создание запросов на SQL-сервере.
3. ПРОСТЫЕ ОПЕРАЦИИ С ОДНОЙ ТАБЛИЦЕЙ
3.1. Инструкция SELECT. • Секции в запросах. • Чтение запроса SELECT. • Порядок секций.
3.2. Фильтрация и выборка. • Описание работы фильтрации и выборки. • Фильтрация по столбцам. • Фильтрация по строкам.
3.3. Использование функций. • Использование функций в базовом решении. • Минусы использования оптимизации в запросах.
3.4. Неизвестные значения. • Проблемы реляционных баз данных: null. • Как бороться с системными последствиями после null.
3.5. Сортировка. • Управление порядком строк в запросе.
3.6. Учебные базы данных (SSMS, ADS). • Работа со вторым учебным сервером (T-SQL). • Инструкция для работы с программами SSMS, ADS.
3.7. Отсечка. • Свойства операции «отсечка». • Описание работы отсечки. • Пример решения задач с использованием отсечки.
3.8. Порядок операций в запросе. • Важность порядка операций в запросе. • Описание процесса операций.
3.9. Устранение дубликатов. • Операция DISTINCT. • Пример устранения дубликатов в запросе.

4. ТРАНСФОРМАЦИЯ ТАБЛИЦЫ	
4.1. Трансформирующие операции.	- Понимание списков. - Трансформирующие операции при выполнении сложных запросов.
4.2. Агрегация.	- Описание работы агрегатных функций. - Особенности агрегатных функций. - Различие агрегатных и скалярных функций.
4.3. Группировка.	- Описание работы группировки. - Создание полезных таблиц с помощью группировки. - Примеры решения задач с функцией группировки.
4.4. Фильтрация групп.	- Применение функции HAVING. - Пример решения более сложной задачи с использованием группировки и фильтрации.
5. МОДИФИКАЦИЯ ДАННЫХ	
5.1. Модификация данных.	- Синтаксис команд INSERT, UPDATE, DELETE. - Пример модификации данных.
5.2. Применение модифицирующих операций.	- Аналитическая «песочница». - Быстрое создание временного набора данных. - Массовое добавление данных в таблицу.
6. ОПЕРАЦИИ С НЕСКОЛЬКИМИ ТАБЛИЦАМИ	
6.1. Работа с несколькими таблицами.	Варианты комбинирования нескольких таблиц.
6.2. Операции над множествами.	- Пересечение множеств. - Функции UNION, INTERSECT, EXCEPT.
6.3. Подзапросы и джойны.	- Механизм подзапросов. - Подзапросы с большим количеством таблиц. - Преимущества и недостатки подзапросов. - Механизм джойнов. - Внутренние соединения. - Сложные аспекты джойнов. - Преимущества и недостатки джойнов.
6.4. Коррелированные соединения.	- Применение латеральных джойнов. - Пример задачи и решение.
7. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ	
7.1. Программирование на SQL.	- Введение в переменные. - Условия, операторные скобки.
7.2. Программные модули.	Разновидности программных модулей: VIEW, STORED PROCEDURE, FUNCTION.
7.3. Табличные выражения.	- Примеры использования табличных выражений для упрощения решения задач. - Причины использования табличных выражений.

8. ПОСТРОЕНИЕ ОТЧЁТОВ	
8.1. Построение отчётов.	• Цели построения отчетов. • Приемы оптимизации отчетов.
8.2. Сводные таблицы.	• Механизм сводных таблиц. • Практический пример применения сводных таблиц.
8.3. Подытоги.	• Пример использования подытогов: CUBE, ROLL UP. • GROUPING.
8.4. Ранжирование.	• Применение ранжирования. • Объяснение определения порядка в запросе.
8.5. Сдвиг.	• Описание работы функции сдвига. • Функция LAG.
8.6. Агрегация со скользящим окном.	• Описание работы скользящего окна.
8.7. Постраничный вывод.	• Способы отражения постраничного вывода.
9. РАБОТА С ХРАНИЛИЩАМИ И ВИТРИНАМИ ДАННЫХ	
9.1. Операционные базы данных.	• Задачи операционных баз данных. • Отдельные базы данных для аналитиков данных.
9.2. Структура хранилища.	• Структура аналитических селектов. • Отличительные черты хранилищ от операционных баз данных фактов и справочников.
9.3. Аналитическая обработка больших объёмов данных.	• Подготовка таблиц с данными. • Советы по обработке больших данных.
9.4. Статистический анализ данных.	• Агрегатные функции и группировка. • Оконные функции и скользящее окно. • Составление сложных показателей.

Пример задания итоговой аттестации

Пример задания № 1:

Для развернутого решения задачи Вам потребуется пройти пять шагов:

Шаг 1. Решить задачу.

Шаг 2. Решить другими способами.

Шаг 3. Проверить решения.

Шаг 4. Оценить решения.

Шаг 5. Ответить на вопрос: ваши решения хорошо масштабируются?

Шаг 1. Решить задачу.

Задача: Как зовут продавцов, которые оформили заказов в Берлин больше, чем в Париж?

Пояснение: Вам понадобятся две таблицы (Рис.1):

- Таблица с продавцами (Employees) – из неё вы возьмёте имена продавцов.
- Таблица с заказами (Orders) – в ней вы посчитаете количество заказов с учётом города.

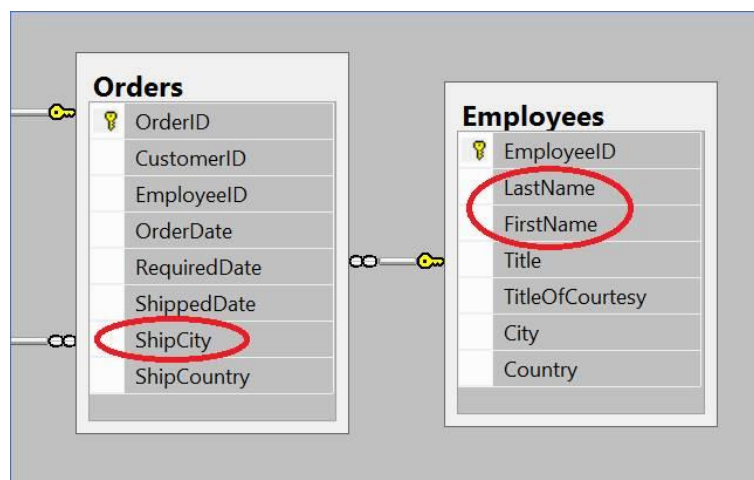


Рисунок 1. Таблицы с продавцами и заказами.

Шаг 2. Решить другими способами.

Найдите несколько разных способов решения этой задачи.

Интересны, разумеется, принципиально разные стратегии решения, а не просто перестановка таблиц в джойне. К примеру, подзапросы и джойны – это два разных способа решения. Постарайтесь решить задачу максимальным числом способов.

Но не переусердствуйте – если набрали уже пять разных способов, остановитесь.

Шаг 3. Проверить решения.

Прежде, чем отправить решения на проверку, обязательно проверьте решения самостоятельно. Подставьте в ваши запросы другие значения, в том числе несуществующие. В этом случае ваши запросы возвращают ожидаемые (и одинаковые) результаты?

Шаг 4. Оценить решения.

Сопроводите каждый вариант решения комментарием, чем оно лучше и хуже других способов, т.е. в каких ситуациях вы бы стали использовать именно этот конкретный способ, а в каких, наоборот, не стали бы.

Шаг 5. Ваши решения хорошо масштабируются?

Дополнительно каждое решение требуется оценить с точки зрения гибкости (универсальности).

Что будет, если в формулировке задачи вместо «больше чем» будет указано «не меньше, чем»?

Напишите, какие изменения вам пришлось бы внести в каждый вариант решения.

Какие из предложенных подходов к решению задачи останутся работоспособными, если в задаче появятся дополнительные параметры, фильтры? А если потребуется оценить не два значения, а три ($A > B > C$)?

По завершению выполнения задания присылайте набор решений (с комментариями по пунктам 4 и 5) на проверку преподавателю.

Вариант решения задания:

Шаг 1. Решение задачи.

Метод подзапросов, диалект Postgres

```
SELECT First_Name || ' ' || Last_Name AS Name
FROM Employees
WHERE (
    SELECT Count (*)
    FROM Orders
    WHERE Ship_City = 'Berlin'
    AND Employee_ID = Employees.Employee_ID
)
>
(
    SELECT Count (*)
    FROM Orders
    WHERE Ship_City = 'Paris'
    AND Employee_ID = Employees.Employee_ID
)
```

Шаг 2. Решение другими способами.

В решении использованы материалы из разделов:

- Фильтрация и выборка;
- Использование функций;
- Агрегация;

- Механизм подзапросов.

Шаг 3. Проверка решения.

1. Добавим в секцию SELECT вывод значений для визуальной проверки.
2. Подставим другие названия городов, например, 'London', 'Rio de Janeiro'.
3. Подставим несуществующие названия городов.

Запрос продолжает работать без ошибок.

Количество считается и выводится.

По несуществующим городам количество равно нулю.

Шаг 4. Оценка решения.

Плюсы решения подзапросами:

- Простота решения;
- Отсутствуют сложные операции, требующие повышенного внимания или связанные с риском ошибок.

Минусы решения подзапросами:

- К таблице с заказами приходится обращаться два раза.

Шаг 5. Масштабируемость решения.

Решение подзапросами позволяет легко добавлять к результату столбцы со статистикой по другим городам (добавляем аналогичный подзапрос). Вместо подсчёта количества легко подставить любую другую агрегатную функцию. Добавление новых фильтров к расчёту делается простым добавлением условий в секцию WHERE вложенного запроса.

Проблемы могут возникнуть при усложнении вложенного запроса, так как все усложнения, добавляемые в один подзапрос, придётся копировать и в другие вложенные SELECT'ы.

Если потребуется заменить операцию сравнения на другую (например "не меньше, чем"), в запросе потребуется только изменить знак операции (на ">=").

Пример задания № 2:

Для развернутого решения задачи Вам потребуется пройти пять шагов:

Шаг 1. Решить задачу.

Шаг 2. Решить другими способами.

Шаг 3. Проверить решения.

Шаг 4. Оценить решения.

Шаг 5. Ответить на вопрос: ваши решения хорошо масштабируются?

Шаг 1. Решить задачу.

Задача: Как зовут покупателей, которые оформили заказов в 1998 году больше, чем в 1997?

Пояснение:

Вам понадобятся две таблицы (Рис 2.):

- Таблица с покупателями (Customers) — из неё вы возьмёте имена покупателей.
- Таблица с заказами (Orders) — в ней вы посчитаете количество заказов с учётом года.

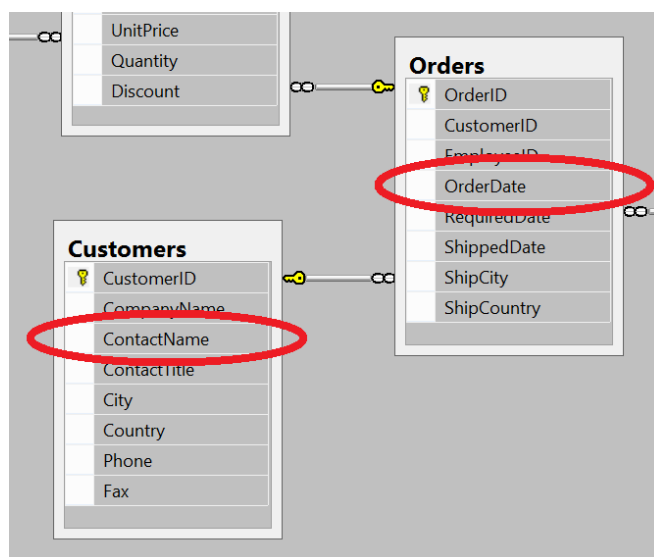


Рисунок 2. Таблицы с продавцами и заказами.

Шаг 2. Решить другими способами.

Найдите несколько разных способов решения этой задачи.

Интересны, разумеется, принципиально разные стратегии решения, а не просто перестановка таблиц в джойне. К примеру, подзапросы и джойны — это два разных способа решения. Постарайтесь решить задачу максимальным числом способов.

Но не переусердствуйте — если набрали уже пять разных способов, остановитесь.

Шаг 3. Проверить решения.

Прежде, чем отправить решения на проверку, обязательно проверьте решения самостоятельно. Подставьте в ваши запросы другие значения, в том числе несуществующие. В этом случае ваши запросы возвращают ожидаемые (и одинаковые) результаты?

Шаг 4. Оценить решения.

Сопроводите каждый вариант решения комментарием, чем оно лучше и хуже других способов, то есть в каких ситуациях вы бы стали использовать именно этот конкретный способ, а в каких, наоборот, не стали бы.

Шаг 5. Ваши решения хорошо масштабируются?

Дополнительно каждое решение требуется оценить с точки зрения гибкости (универсальности).

Что будет, если в формулировке задачи вместо «больше чем» будет указано «не меньше, чем»?

Напишите, какие изменения вам пришлось бы внести в каждый вариант решения.

Какие из предложенных подходов к решению задачи останутся работоспособными, если в задаче появятся дополнительные параметры, фильтры? А если потребуется оценить не два значения, а три ($A > B > C$)?

По завершении выполнения задания присылайте набор решений (с комментариями по пунктам 4 и 5) на проверку преподавателю.

Вариант решения задания:

Шаг 1. Решение задачи.

Метод подзапросов, диалект Postgres

```
SELECT Contact_Name AS Name
FROM Customers
WHERE (
    SELECT Count (*)
    FROM Orders
    WHERE Date_Part ('year', Order_Date) = 1998
    AND Customer_ID = Customers.Customer_ID
)
>
(
    SELECT Count (*)
    FROM Orders
    WHERE Date_Part ('year', Order_Date) = 1997
    AND Customer_ID = Customers.Customer_ID
)
```

Шаг 2. Решение другими способами.

В решении использованы материалы из разделов:

- Фильтрация и выборка;
- Использование функций;
- Агрегация;
- Механизм подзапросов.

Шаг 3. Проверка решения.

1. Добавим в секцию SELECT вывод значений для визуальной проверки.
2. Подставим другие номера годов, например, 1996.
3. Подставим отсутствующие в базе или ещё не наступившие годы.

Запрос продолжает работать без ошибок.

Количество считается и выводится.

По отсутствующим годам количество равно нулю.

Шаг 4. Оценка решения.

Плюсы решения подзапросами:

- Простота решения;
- Отсутствуют сложные операции, требующие повышенного внимания или связанные с риском ошибок.

Минусы решения подзапросами:

- К таблице с заказами приходится обращаться два раза.

Шаг 5. Масштабируемость решения.

Решение подзапросами позволяет легко добавлять к результату столбцы со статистикой по другим городам (добавляем аналогичный подзапрос). Вместо подсчёта количества легко подставить любую другую агрегатную функцию. Добавление новых фильтров к расчёту делается простым добавлением условий в секцию WHERE вложенного запроса.

Проблемы могут возникнуть при усложнении вложенного запроса. Так как все усложнения, добавляемые в один подзапрос, придётся копировать и в другие вложенные SELECT'ы.

Если потребуется заменить операцию сравнения на другую (например "не меньше, чем"), в запросе потребуется только изменить знак операции (на ">=").