

Дополнительная профессиональная программа повышения квалификации «Анализ данных для начинающих»

Программа направлена на формирование знаний о методике составления требований к программному продукту, развитие навыков сбора, интерпретации и описания требований для решения аналитических задач, а также на практическое применение языка запросов SQL для аналитической обработки данных.

№	Наименование	Характеристика
1	Наименование и цели программы	
1.1	Наименование	«Анализ данных для начинающих»
1.2	УИН программы	ПК26-010
1.3	Вид программы	Дополнительная профессиональная программа повышения квалификации.
1.4	Цель	Сформировать знания об этапах создания программных продуктов и требованиях к ним, а также навыки моделирования процессов для анализа и представления бизнес-процессов. Научить применять язык SQL для аналитической обработки данных. Сформировать знания в области тестирования программного обеспечения.
1.5	Планируемые результаты (формируемые или совершенствующиеся компетенции в соответствии с указанным профстандартом)	Подготовка к выполнению задания на тестирование ПО. Анализ результатов тестирования ПО на соответствие ожидаемым результатам, оформление и размещение отчета о тестировании в соответствии с жизненным циклом ПО в системе контроля версий. Профессиональный стандарт 06.004 «Специалист по тестированию в области информационных технологий» (утв. приказом Министерства труда и социальной защиты Российской Федерации от 2 августа 2021 г. № 531н).
		Извлечение и описание образцов данных и агрегированных значений из систем-аналогов. Профессиональный стандарт 06.022 «Системный аналитик» (утв. приказом Министерства труда и социальной защиты Российской Федерации от 27 апреля 2023 г. № 367н).
2	Формат реализации	
2.1	Язык обучения	Русский
2.2	Форма обучения	Очно-заочная с применением ДОТ
2.3	Объем программы	50 академических часов

2.4	Режим занятий	В течение 7 недель в удобное для слушателя время по 5 - 10 ак. часов в неделю. Вебинары – согласно расписанию занятий.
2.5	Выдаваемый документ	Удостоверение о повышении квалификации Академии ПСБ.
2.6	Целевая аудитория	Начинающие специалисты в области анализа данных; руководители проектов и менеджеры в области информационных технологий и аналитики; банковские сотрудники, интересующиеся анализом данных и желающие приобрести базовые знания в этой области.
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	Самородов Федор Анатольевич , преподаватель АНО ДПО «Академия ПСБ», преподаватель ОЧУ ДПО «Бауманский компьютерный учебный центр «Специалист.ру»
3.5	Форма итоговой аттестации	Экзамен в форме выполнения задания.
3.6	Оценочные материалы	<p>Пример задания см. Приложение № 4.</p> <p>Результаты выполнения оцениваются на основании 4 критериев, с присвоением баллов по каждому критерию оценки.</p> <p>Максимальный балл, который может набрать слушатель за выполнение задания – 100 баллов.</p> <p>Условием успешного завершения обучения и получения удостоверения о повышении квалификации является прохождение итоговой аттестации с результатом не менее 40 баллов.</p>
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
«Анализ данных для начинающих»

Количество часов – 50 академических часов.

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.)	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					
1	Введение	2					1		1						A/01.4
2	Процесс разработки программного обеспечения	5					3		3		1	1			A/01.4
3	Сбор бизнес-требований	7					5	2	3	2	1	1			A/01.4
4	Разработка и описание требований к программному обеспечению	8					6		6		1	1			A/01.4, B/04.5

¹ Вид текущего контроля успеваемости: ПР – практическая работа.

5	SQL	14						12		12			1	1			A/04.4
6	Тестирование программного обеспечения	10						8	2	6	2		1	1			B/04.5
7	Итоговая аттестация	4														Э ² /4	A/04.4, A/01.4 B/04.5
	ИТОГО:	50						35	4	31	4		6	5		4	

² Вид итоговой аттестации: Э – экзамен.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Период обучения:		
50 ак. часов	7 недель	1,5 месяца

РАБОЧИЕ ПРОГРАММЫ РАЗДЕЛОВ / МОДУЛЕЙ / ДИСЦИПЛИН

1. ВВЕДЕНИЕ
1.1. Входное тестирование.
1.2. Введение в анализ данных.
2. ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
2.1. Процесс разработки ПО.
2.2. Жизненный цикл программного продукта.
2.3. Участники и роли в процессе разработки.
3. СБОР БИЗНЕС-ТРЕБОВАНИЙ
3.1. Понимание и задачи бизнес-требований.
3.2. Сбор бизнес-требований.
3.3. Структурирование бизнес-требований.
4. РАЗРАБОТКА И ОПИСАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ
4.1. Понимание требований к ПО.
4.2. Описание процессов.
4.3. Моделирование требований.
4.4. Инструменты моделирования требований.
4.5. Постановка задачи разработчику.
5. SQL
5.1. SQL.
5.2. Клиент-серверная архитектура.
5.3. Столбцы и строки.
5.4. Понимание таблиц.
5.5. Инструкция SELECT.
5.6. Фильтрация и выборка.
5.7. Использование функций.
5.8. Неизвестные значения.
5.9. Сортировка.
5.10. Отсечка.
5.11. Устранение дубликатов.
5.12. Порядок операций в запросе.
5.13. Трансформирующие операции.
5.14. Агрегация.
5.15. Группировка.

5.16. Фильтрация групп.
6. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
6.1. Понимание тестирования ПО.
6.2. Виды тестирования.
6.3. Описание тест-кейсов.
6.4. Проведение тестирования.
6.5. Документирование ПМИ и ПСИ.

Пример задания итоговой аттестации

Задание № 1.

Описание задания:

На учебном сервере SQL.DataExplorer.ru найдите базу данных "(хакатон Azure ML в Микрософте) 21-22 мая 2016". В этой базе данных исследуйте таблицу "Мать и дитя"."MS_Data05".

Она содержит обезличенные данные по процедурам, которые проводились для пациентов сети клиник "Мать и дитя" в 2014 – 2015 годах.

Каждая строка этой таблицы – это обслуживание одного клиента по одной процедуре в течение одного дня. Если клиенту потребовалось несколько одинаковых процедур в один день, то столбец SERV_COUNT содержит значение больше единицы.

Вопрос по заданию:

Разбить клиентов на сегменты по полу и возрасту, ориентируясь на генерируемую ими выручку.

Уникальный номер пациента	Дата рождения и пол пациента	Дата проведения процедуры	Код процедуры и код типа процедур		Количество процедур и выручка		Номер отделения и регион		
PATIENTS_ID	PATIENTS_BIRTHDAY	PATIENTS_POL	DATE_EXEC	SERV_ID	SERV_GROUP_ID	SERV_COUNT	SERV_SUM	FILIAL_ID	REGION_ID
111752	2006-04-25	0	2014-09-20	20229	2028	1	2560,00	1	1
321330	1987-02-02	1	2014-07-29	31716	2400	1	720,00	116011	9
320916	1983-02-17	1	2014-09-27	31814	2642	1	17920,00	116011	9
178049	2005-11-10	1	2014-07-01	26122	2391	2	3040,00	81312	3
390770	1980-06-26	1	2015-07-16	31357	2026	1	680,00	126836	5

Рисунок 1. Пример из базы данных "Мать и дитя"."MS_Data05"

Задача № 1. Привести к удобному формату.

Исходная таблица довольно большая. Предложите способ (напишите SQL-запрос), который позволит радикально уменьшить число строк.

При этом Ваш запрос должен обязательно сохранить имеющиеся в данных закономерности и заметные особенности в данных, интересные в рамках нашего аналитического сценария.

Задача № 2. Провести аналитическую обработку.

Найдите в данных какую-либо закономерность, интересную в рамках нашего аналитического сценария и покажите эту закономерность наглядно при помощи SQL-запроса. Запрос должен выдавать результат, который будет сразу понятен без какой-либо дополнительной визуализации (без условного форматирования, без графиков и прочего).

Сама таблица, которую выдаст ваш SELECT должна иллюстрировать найденную закономерность. Суть этого задания в осмысленном применении фильтрации, группировки, агрегации, сортировки — тех операций, которые мы изучали, для аналитической задачи.

Задание № 2.

1. Выберите правильные ответы из предложенных вариантов.
2. Обоснуйте, почему Вы считаете правильным именно этот вариант.
3. Обоснуйте, почему Вы считаете остальные варианты не верными.

Описание задачи:

Отметьте три наиболее важные цели тестирования:

1. Исправить ошибки в ПО.
2. Минимизировать риски при внедрении ПО.
3. Обеспечить разработчиков информацией, которую они смогут использовать для исправления ошибок.
4. Обеспечить менеджеров информацией, которую они смогут использовать для оценки рисков использования ПО.
5. Оценить соответствие ПО сформулированным требованиям.
6. Оценить соответствие ПО не сформулированным, но подразумеваемым требованиям.

Пример решения итоговых заданий:

Задание № 1.

Задача № 1.

Идеи для сокращения количества строк:

1) Поскольку в задаче не требуется детализация по медицинским процедурам, можно сгруппировать таблицу, исключив лишнюю детализацию:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

-- Идеи для сокращения количества строк:

-- Поскольку в задаче не требуется разбивка по процедурам,
-- регионам и филиалам, можно сгруппировать таблицу,
-- исключив лишнюю детализацию.

```
SELECT          DATE_EXEC,
                PATIENTS_BIRTHDAY,
                PATIENTS_POL,
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
GROUP BY       DATE_EXEC,
                PATIENTS_BIRTHDAY,
                PATIENTS_POL
```

2) Можно укрупнить дату рождения до года:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

-- Идеи для сокращения количества строк:
-- Можно укрупнить дату рождения до года.

```
SELECT          DATE_EXEC,
                DatePart (year, PATIENTS_BIRTHDAY) AS
Birthday_Year,
                PATIENTS_POL,
                SERV_ID,
                SERV_GROUP_ID,
                FILIAL_ID,
                REGION_ID,
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
GROUP BY       DATE_EXEC,
                DatePart (year, PATIENTS_BIRTHDAY),
                PATIENTS_POL,
                SERV_ID,
                SERV_GROUP_ID,
                FILIAL_ID,
```

REGION_ID

3) Можно укрупнить дату процедуры до месяца:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

- Идеи для сокращения количества строк:
- Можно укрупнить дату процедуры до месяца.

```
SELECT          DatePart (year, DATE_EXEC) AS "Year",
                DatePart (month, DATE_EXEC) AS "Month",
                PATIENTS_BIRTHDAY,
                PATIENTS_POL,
                SERV_ID,
                SERV_GROUP_ID,
                FILIAL_ID,
                REGION_ID,
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
GROUP BY       DatePart (year, DATE_EXEC),
                DatePart (month, DATE_EXEC),
                PATIENTS_BIRTHDAY,
                PATIENTS_POL,
                SERV_ID,
                SERV_GROUP_ID,
                FILIAL_ID,
                REGION_ID
```

4) Совместить эти три подхода — сгруппировать, убрать лишние столбцы и укрупнить периоды:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

- Идеи для сокращения количества строк:
- Совместить три подхода — сгруппировать,
- убрав лишние столбцы и укрупнить периоды.

```
SELECT          DatePart (year, DATE_EXEC) AS "Year",
                DatePart (month, DATE_EXEC) AS "Month",
                DatePart (year, PATIENTS_BIRTHDAY),
                PATIENTS_POL,
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
GROUP BY       DatePart (year, DATE_EXEC),
                DatePart (month, DATE_EXEC),
                DatePart (year, PATIENTS_BIRTHDAY),
                PATIENTS_POL
```

Задание № 1.

Задача № 2.

Пример сегментирования клиентов по полу и возрасту.

1) При делении по возрасту пациента на момент проведения процедуры первые 10 позиций (по выручке) занимают женщины в возрасте около 30 лет. Имеет смысл выделить их в отдельный сегмент:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

- Пример сегментирования клиентов по полу и возрасту.
- При делении по возрасту пациента на момент проведения процедуры
- первые 10 позиций (по выручке) занимают женщины в возрасте около 30 лет.
- Имеет смысл выделить их в отдельный сегмент.

```
SELECT          TOP 10
                DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC)
AS "Возраст",
                PATIENTS_POL,
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
GROUP BY       DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC),
                PATIENTS_POL
ORDER BY       SERV_SUM DESC
```

2) У мужчин особое значение имеет возраст 31 год. В этот период на порядок повышается и выручка и количество процедур:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

- Пример сегментирования клиентов по полу и возрасту.
- У мужчин особое значение имеет возраст 31 год.
- В этот период на порядок повышается и выручка и количество процедур.

```
SELECT          DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC) AS "Возраст",
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
WHERE           PATIENTS_POL = 2
GROUP BY       DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC)
ORDER BY       SERV_SUM DESC
```

3) Для маленьких детей выручка и число процедур не коррелируют с полом. Можно сегментировать только по возрасту:

```
USE          "(хакатон Azure ML в Микрософте) 21-22 мая 2016"
```

- Пример сегментирования клиентов по полу и возрасту.
- Для маленьких детей выручка и число процедур не коррелируют с полом.
- Можно сегментировать только по возрасту.

```

SELECT          DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC) AS "Возраст",
                PATIENTS_POL,
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
WHERE          DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC) BETWEEN 0
AND 8
AND            PATIENTS_POL IN (0, 1)
GROUP BY      DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC),
                PATIENTS_POL
ORDER BY      SERV_SUM DESC

```

4) Грудничкам требуется на порядок больше процедур. Имеет смысл выделить их в отдельный сегмент:

```

USE            "(хакатон Azure ML в Микрософте) 21-22 мая 2016"

```

- Пример сегментирования клиентов по полу и возрасту.
- Грудничкам требуется на порядок больше процедур.
- Имеет смысл выделить их в отдельный сегмент.

```

SELECT          DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC) AS "Возраст",
                Sum (SERV_COUNT) AS SERV_COUNT,
                Sum (SERV_SUM) AS SERV_SUM
FROM            "Мать и дитя"."MS_Data05"
WHERE          DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC) BETWEEN 0
AND 8
GROUP BY      DateDiff (year, PATIENTS_BIRTHDAY, DATE_EXEC)
ORDER BY      SERV_SUM DESC

```

Задание № 2.

1. Правильные ответы из предложенных вариантов: 2, 3, 4.
2. Вышеуказанные варианты являются правильными, потому что:
 - вариант 2: это главная цель тестирования – выявить и оценить риски серьезных проблем при эксплуатации ПО. Тестирование помогает выявлять такие ситуации и обеспечивать разработчиков информацией, которую они смогут использовать для исправления ошибок, а также обеспечивать менеджеров информацией, которую они смогут использовать для оценки рисков использования ПО;
 - варианты 3, 4: для выработки решения разработчикам и менеджерам нужна подробная информация (в связке с вариантом 2 по выявленным рискам).
3. Остальные варианты не правильные, потому что:
 - вариант 1: исправление ошибок – не цель, а один из эффектов, к которому приводит тестирование. Программа может поставляться и эксплуатироваться с имеющимися ошибками, даже если и разработчик, и заказчик в курсе наличия ошибок. Некоторые ошибки исправить невозможно, другие ошибки обойдутся в исправлении слишком дорого, третьи ошибки настолько незначительны, что до их исправления никогда не дойдут. Ценность ПО определяется не наличием ошибок.
 - варианты 5, 6: несоответствие требованиям – это лишь один из возможных рисков.