ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Приказом Ректора АНО ВО
«Центральный университет»
Ивашкевич Е.В.
от «19» января 2024 г. № 0119.37

Рабочая программа дисциплины (модуля) «SQL и базы данных» дополнительной профессиональной программы – программы профессиональной переподготовки «Академия data science»

Траектория: Машинное обучение

Москва
2024

Электронный документ

Содержание

1. Краткая характеристика дисциплины (модуля)

Изучение дисциплины (модуля) «SQL и базы данных» является ключевым для понимания того, как эффективно управлять, хранить и извлекать данные, что критически важно в современном мире, где информация играет центральную роль в принятии решений. Освоение SQL позволяет специалистам анализировать большие объемы данных, оптимизировать процессы и разрабатывать надежные решения для бизнеса и науки.

Цель изучения дисциплины (модуля): овладение навыками проектирования, управления и эффективного извлечения данных из реляционных баз данных с использованием языка SQL для решения практических задач в различных сферах.

Задачи изучения дисциплины (модуля):

— формирование знаний и развитие понимания по темам: основы хранения данных, типы данных и способы их преобразования, основные типы баз данных, операторы SQL, синтаксис SQL на уровне продвинутого пользователя, Data Definition Language, Data manipulation language, базовые модели хранения данных, способы сокращения количества используемых ресурсов в SQL запросах, способы преобразования данных с помощью SQL: агрегация, соединение, подзапросы, оконные функции, функции и процедуры;

— освоение и развитие умений решать задачи с помощью SQL на уровне продвинутого пользователя, читать план запроса и проводить оптимизацию SQL-кода, переводить бизнес-смысл задачи в код SQL, использовать Data Definition Language для решения задач, решать задачи с использованием операций преобразования данных: агрегация, соединение, подзапросы, оконные функции, функции и процедуры;

— формирование навыков решения бизнес задач с помощью SQL, оптимизации используемого количества ресурсов в SQL запросах; декомпозиции сложной задачи на более простые и понятные подзадачи, которые слушатели могут самостоятельно перевести в код SQL.

2. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)			
1	Основы баз данных	3	3		11	Тренажер, Подготовка к семинару
2	Основы SQL	3	3		11	Тренажер, Подготовка к семинару
3	Агрегация	3	3		11	Тренажер, Подготовка к семинару
4	Соединение таблиц	3	3		11	Тренажер, Подготовка к семинару
5	Подзапросы и CTE	3	3		11	Тренажер, Подготовка к семинару
6	Оконные функции	3	3		11	Тренажер, Подготовка к семинару
7	Определение данных	3	4		11	Тренажер, Подготовка к семинару
8	Организация данных	3	4		11	Тренажер, Подготовка к семинару
9	Продвинутые механизмы SQL	4	4		11	Тренажер, Подготовка к семинару
10	Greenplum	4	4		11	Тренажер, Подготовка к семинару
11	Clickhouse	4	4		11	Тренажер, Подготовка к семинару
12	Оптимизация запросов	4	4		11	Тренажер, Подготовка к семинару
	Зачет с оценкой			6		Защита проекта
	Итого:	***40***	***42***	***6***	***132***	
Объем дисциплины (модуля) (в ак. ч.)		***220***				

3. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы баз данных	История развития систем хранения информации. Терминология. Основы баз данных. Реляционная модель данных. OLAP и OLTP. Основы построения хранилищ данных.
2	Основы SQL	Типы данных. Основные операторы SQL. Функции для работы с текстом, датами, числами.
3	Агрегация	GROUP BY, HAVING и функции агрегации (COUNT, SUM, MIN, MAX, AVG, PERCENTILE_CONT).
4	Соединение таблиц	JOINS, UNION и другие операторы работы со множествами.
5	Подзапросы и CTE	Nested Subqueries, Correlated Subqueries. CTE.
6	Оконные функции	Синтаксис оконных функций. Оконные функции для ранжирования. LAG/LEAD. FIRST_VALUE, LAST_VALUE, NTH_VALUE. Группировка в рамках оконных функций: COUNT, SUM, AVG, MIN, MAX. Границы окна: unbounded, preceding и following.
7	Определение данных	DDL. Data constraints. VIEWS.
8	Организация данных	Правила нормализации данных. Методологии проектирования баз данных.
9	Продвинутые механизмы SQL	Индексы. Транзакции. Способы масштабирования баз данных.
10	Greenplum	СУБД Greenplum. Особенности синтаксиса SQL.
11	Clickhouse	СУБД Clickhouse. Особенности синтаксиса SQL.
12	Оптимизация запросов	Оптимизация запросов: основные понятия и инструменты.

4. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый слушатель в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Слушателям обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Кара-Ушанов В.Ю. SQL — язык реляционных баз данных : учебное пособие / В.Ю. Кара-Ушанов.— Екатеринбург : Изд-во Урал. ун-та, 2016.— 156 с. .— ISBN 978-5-7996-1622-9.

Дополнительная литература:

1. Бьюли А. Изучаем SQL. – Пер. с англ. – СПб: Символ Плюс, 2007. – 312 с., ил. – ISBN 0-596-00727-2.

5. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

— столами и стульями;

— компьютерной техникой;

— механическими калькуляторами;

— специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru	https://elibrary.ru/defaultx.asp

Электронный документ

	библиотека	
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое

Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

6. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «SQL и базы данных» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, практические занятия, тренажеры, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в практическом занятии (аудиторная работа) – активная работа слушателя на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре слушателям рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Тренажер – домашние задания разного уровня сложности с возможностью самопроверки.

Тренажер позволяет оперативно оценивать усвоение материала и выявлять пробелы в знаниях через тесты и практические задачи. Такой формат способствует регулярной самопроверке и повышает мотивацию к изучению дисциплины (модуля).

Проект – это целенаправленная деятельность, имеющая определенные цели, задачи и временные рамки, в результате которой создается уникальный продукт или услуга.

Для успешной подготовки проекта рекомендуется выполнять следующие рекомендации:

- четко определите цель и задачи проекта, чтобы понимать, какой результат вы хотите достичь;
- составьте план работы, разбив проект на этапы с указанием сроков выполнения каждого из них;
- используйте разнообразные источники информации и инструменты для исследования темы, чтобы обеспечить качественную основу для вашего проекта;
- регулярно проверяйте прогресс и вносите коррективы в план, если это необходимо, чтобы оставаться на правильном пути к завершению проекта.

Самостоятельная работа – работа слушателей, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы слушатели взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи слушателя включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***зачета с оценкой***.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	**Пятибалльная оценка**	**Оценка за зачет**	**Общая характеристика результата обучения по дисциплине (модулю)**
10	Отлично	Зачтено	Слушатель полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Слушатель хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Слушатель обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать,
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Слушатель хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Слушатель обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Слушатель способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Слушатель не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «SQL и базы данных» оценивается следующим образом:

Активность	Вес	Количество	Описание
Аудиторная работа	50%	14	Активное участие в семинарах: отвечать на вопросы преподавателя и задавать свои
Тренажер	30%	14	Домашние задания разного уровня сложности с возможностью самопроверки
Зачет с оценкой (Защита проекта)	20%	1	Проект на основе реальных рабочих задач

Формула расчёта итоговой оценки по дисциплине (модулю) «SQL и базы данных»: «0,5 × аудиторная работа + 0,3 × среднее за тренажеры + 0,2 × зачет с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)
Примерные задания для подготовки к семинарам

Тема: Операторы SELECT и FROM

Задача 1.

Вывести из таблицы продуктов (shop_products) все уникальные пары «Название продукта» — «Цена».

Задача 2.

Тебе необходимо подготовить выборку для обзвона клиентов. В выборке нужно вывести из таблицы клиентов (shop_customers) следующую информацию:

- ID клиента (customer_id);
- имя клиента (first_name);
- номер телефона (phone_number).

Задача 3.

Вывести все столбцы из таблицы платежей (shop_payments).

Задача 4.

Вывести все уникальные статусы платежей из таблицы shop_payments. Задать в операторе FROM alias для таблицы.

Тема: Функции агрегации

Задача 1.

По таблице *databases_and_sql.shop_orders* посчитать общее количество заказов, сумму заказов и количество уникальных клиентов при условии, что заказ совершён в ноябре 2024 года или позже.

Что требуется в ответе

Ответ должен содержать колонки:

1) cnt_orders = количество заказов;
2) total_amount = сумма заказов;
3) cntd_customers = количество уникальных клиентов.

Задача 2.

По таблице *databases_and_sql.shop_products* найти общее количество товаров, их суммарную стоимость, а также количество уникальных категорий товаров при условии, что товар добавлен в 2022 году.

Что требуется в ответе

Ответ должен содержать колонки:

1) cnt_products = количество товаров;
2) total_price = суммарная стоимость;
3) cntd_categories = количество уникальных категорий.

Задача 4.

По таблице *databases_and_sql.shop_payments* посчитать минимальную, среднюю и максимальную сумму платежа при условии, что платёж был совершён с помощью карты не ранее 1 июня 2024 года.

Что требуется в ответе

Ответ должен содержать колонки:

1) min_amount = минимальная сумма платежа;
2) avg_amount = средняя сумма платежа;
3) max_amount = максимальная сумма платежа.

Задача 5.

По таблице *databases_and_sql.shop_orders* посчитать среднюю и медианную сумму заказа при условии, что заказ не был отменён. Округлить значения до целых.

Что требуется в ответе

Ответ должен содержать колонки:

1) avg_amount = средняя сумма заказа;
2) median_amount = медианная сумма заказа.

Задача 6.

По таблице *databases_and_sql.shop_payments* посчитать среднюю сумму платежа со статусом 'Success', а также 10-й и 90-й перцентиль суммы платежа со статусом 'Success'. Округлить значения до целых.

Что требуется в ответе

Ответ должен содержать колонки:

1) avg_amount = средняя сумма платежа;

2) perc10 = 10-й перцентиль суммы платежа;
3) perc90 = 90-й перцентиль суммы платежа.

Тема: INNER JOIN

Задача 1.

По таблице *databases_and_sql.shop_customers* определить клиентов, которые совершали заказ за последние 3 месяца.

Что требуется в ответе

Ответ должен содержать колонки:

- ID клиента,
- имя,
- фамилия.

Задача 2.

1) Определить для клиентов из таблицы *databases_and_sql.shop_customers* их адрес из таблицы *databases_and_sql.shop_addresses*. Если у клиента нет адреса, то информацию о нём выводить не нужно.

Что требуется в ответе

Ответ должен содержать колонки:

- ID клиента,
- имя,
- фамилия,
- адрес клиента.

2) Посчитать число уникальных адресов на клиента.

Что требуется в ответе

Ответ должен содержать колонки:

- id клиента,
- число его адресов.

Задача 3.

Определить число заказов для клиентов из таблицы *databases_and_sql.shop_customers*, которые за последние 3 месяца совершили более одного заказа.

Что требуется в ответе

Ответ должен содержать колонки:

- ID клиента,
- имя,
- фамилия,
- число заказов клиента.

Задача 4.

Определить клиентов из таблицы *databases_and_sql.shop_customers*, которые оставляли отзывы за последние полгода (таблица *databases_and_sql.shop_reviews*). Также рассчитай число отзывов, оставленных клиентами, и среднюю оценку, которую они ставили.

Что требуется в ответе

Ответ должен содержать колонки:

- ID клиента,
- имя
- фамилия,
- число отзывов,
- средняя оценка.

Тема: Подзапросы в SELECT и WHERE. Подзапросы в FROM и JOIN

Задача 1.

Из таблицы *databases_and_sql.shop_products* выведи все ID продуктов, их названия, цену, а также среднюю цену по всем продуктам и разницу между ценой товара и средней ценой. Оставь только продукты, цена которых выше среднего значения.

Что требуется в ответе

Ответ должен содержать колонки:

- product_id;
- name;
- price;
- avg_price = средняя цена по всем продуктам;
- diff = разница между ценой продукта и средней ценой всех продуктов.

Задача 2.

По таблице databases_and_sql.shop_payments для каждого метода оплаты посчитай количество платежей, общее количество платежей по всем методам, а также долю этого метода платежа от всех платежей.

Что требуется в ответе

Ответ должен содержать колонки:

- payment_method;
- cnt_payments = количество платежей для метода оплаты;
- cnt_all = общее количество платежей по всем методам оплаты;
- pct = доля количества платежей для метода оплаты от общего количества платежей по всем методам.

Задача 3.

Выведи названия категорий и среднюю цену товаров в них при условии, что средняя цена выше 100 000.

Что требуется в ответе

Ответ должен содержать колонки:

- name;
- avg_price = средняя цена товаров категории.

Задача 4.

Выведи фамилии и имена клиентов, а также общую сумму их заказов при условии, что сумма превышает 1 000 000. Отсортируй по убыванию суммы.

Что требуется в ответе

Ответ должен содержать колонки:

- first_name;
- last_name;
- sum_amount = общая сумма заказов клиента.

Тема: Оконные функции

Задание 1.

Выделите топ-5 продуктов, которые имеют наибольший относительный вклад по суммарной выручке внутри своей категории.

Что требуется в ответе

Ответ должен содержать колонки:

1. **name** - название продукта
2. **share_in_cat** - процентный вклад данного продукта в выручку по его категории

В ответе должно быть 5 строк.

Задание 2.

Пронумеруй все заказы в рамках каждого клиента по времени, однако нумерация должна начинаться только после того, как пользователь совершил две покупки со статусом *Success*.

Что требуется в ответе

Выведи строки результирующей таблицы, соответствующие клиенту с *customer_id = 116*.

Ответ должен содержать колонки:

- **order_id**;
- **customer_id**;
- **order_date**;
- **total_amount**;
- **rn** — номер заказа в рамках клиента.

Задание 3.

Постройте динамику суммарных выплат по дням.

Постройте скользящее среднее 3 видов:

- MA(3): За 3 последних наблюдения
- MA(10): За последние 5 и следующие 5 наблюдений
- Месячное скользящее среднее: За последние 15 и следующие 15 наблюдений.

Дайте колонкам названия в вашем SQL-запросе:

- payment_date - дата платежа
- sum_amt - суммарные выплаты
- avg_3 - MA(3)
- avg_55 - MA(10)
- avg_month - MA(30)

Исполните ячейку ниже для визуализации своего результата.

Задание 4.

Необходимо исследовать динамику того, как пользователи уходят и возвращаются.

Будем считать, что если промежуток между покупками клиента составлял больше 90 дней, то он «впадал в спячку», после чего вернулся к нам, если совершил покупку.

Необходимо для каждой покупки каждого клиента разметить, какой она была по счёту у данного клиента, а также в данном цикле активности (то есть «спячка» обнуляет цикл активности).

Например, если пользователь X совершал покупки в даты [01.01.2023,01.01.2024,02.01.2024], то:

- общая нумерация будет [1,2,3];
- нумерация по циклам будет [1,1,2], так как между первым и вторым заказом пользователя прошло больше дней.

Что требуется в ответе

Выведи строки результирующей таблицы, соответствующие клиенту с *customer_id = 91*.

Ответ должен содержать колонки:

- **order_id**;
- **customer_id;**
- **order_date**;
- **rn_loc** — «локальный» номер/ранг в рамках покупательского цикла клиента;
- **rn_glob** — «глобальный» номер/ранг в рамках клиента.

Задание 5.

Для каждого пользователя рассчитайте, сколько раз он переходил с одного типа оплаты на другой за всё время.

Выделите топ-5 пользователей, у которых наибольшая доля "смен" среди всех покупок.

Возьмите только тех пользователей, которые совершили строго больше 5 покупок, а также имели строго больше 1 смены типа оплаты.

Что требуется в ответе

Ответ должен содержать колонки:

- **customer_id**
- **share_change - доля "смен" среди всех покупок**.

В ответе должно быть 5 строк.

Примерные задания для тренажеров

Тренажер 1.

Задача 1.

Создай таблицу со следующей информацией о клиентах:

- customer_id — ID клиента (первичный ключ, тип данных int);
- fio — ФИО клиента (не может быть пустым, тип данных VARCHAR);
- avg_rating — средняя оценка, которую оставил данный клиент (может принимать значения от 0 до 5, тип данных float).

Таблицу назвать customer_x_ratings.

Задача 2.

Заполни созданную в первом задании таблицу данными из таблиц shop_customers и shop_reviews.

После чего выведи топ-20 клиентов по среднему рейтингу (в случае равного среднего рейтинга отсортируй по ФИО).

Задача 3.

Добавь в созданную ранее таблицу столбец lifetime (INTERVAL), представляющий собой разницу между текущей датой и датой создания клиента в БД (поле created_at).

После чего заполни таблицу, используя данные из учебной БД. В завершении также выведи топ-20 клиентов по среднему рейтингу.

Задача 4.

1. Создай таблицу, содержащую следующую информацию:
   - category_id — ID категории, является первичным ключом, тип данных int;
   - category_name — название категории, тип данных VARCHAR;
   - product_cnt — число товаров в категории, тип данных int (и добавь ограничение, что значение больше 0);
   - avg_price — средняя цена товара в категории, тип данных float.

   Таблицу назвать category_info.
2. Добавь в созданную таблицу строку:
   - category_id = 1;
   - category_name = Телефоны;
   - product_cnt = 10;
   - avg_price = 49523,12.

Задача 5.

1. Создай таблицу, содержащую следующую информацию:
   - product_id — ID продукта, является первичным ключом, тип данных int;
   - product_name — название товара, тип данных VARCHAR;
   - category_name — название категории, тип данных VARCHAR;
   - orders_cnt — число заказов, тип данных int (добавь ограничение, что поле не может быть отрицательным).

   Таблицу необходимо назвать product_info.
2. Заполни таблицу данными из таблиц shop_products, shop_categories и shop_orders.
3. Удали информацию по категории «Ноутбуки».

Тренажер 2.

Задача 1.

Условие задачи

Найти всех покупателей, которые возвращали заказы в 2024 году, и для каждого клиента подсчитать количество и сумму возвращённых заказов. Ограничьтесь только теми покупателями, которые произвели строго больше одного возврата (т.е. было больше 1

order_id со статусом 'Refunded').
Выведите топ-3 покупателя, у которых сумма заказов была больше всех.

Результирующие столбцы

1. *customer_id* - идентификатор пользователя.
2. *cnt_orders* - количество заказов (в штуках).
3. *total_amount* - сумма заказов (в рублях).

Задача 2.

Условие задачи

Для каждого месяца из таблицы *databases_and_sql.shop_orders* посчитай среднюю сумму заказа.
Округли результаты до двух знаков после запятой и отсортируй по возрастанию даты.
Ограничься только первыми шестью месяцами 2023 года.

Результирующие столбцы

1. *dt* - месяц совершения заказа (в формате даты).
2. *avg_amount* - средняя сумма заказа, округлённая до двух знаков после запятой.

Задача 3.

Условие задачи

При работе с данными важно понимать, какие поля имеют пропущенные значения.
Для каждого столбца в таблице *databases_and_sql.shop_orders* посчитай, сколько в нём **заполненных** значений, и выведи результаты в одну строку.
Используй свойства оператора *COUNT*.

Результирующие столбцы

1. *order_id_not_nulls* - кол-во не нулловых значений поля order_id.
2. *customer_id_not_nulls* - кол-во не нулловых значений поля customer_id.
3. *order_date_not_nulls* - кол-во не нулловых значений поля order_date.
4. *order_status_not_nulls* - кол-во не нулловых значений поля order_status.
5. *delivery_date_not_nulls* - кол-во не нулловых значений поля delivery_date.
6. *total_amount_not_nulls* - кол-во не нулловых значений поля total_amount.
7. *shipping_address_id_not_nulls* - кол-во не нулловых значений поля shipping_address_id.

(В результирующей таблице будет только одна строка)

Задача 4.

Условие задачи

По таблице *databases_and_sql.shop_orders* для каждого года заказа найди общее количество заказов и количество доставленных заказов (т.е. заказов с с указанной датой доставки), а также конверсию из заказа в доставку (т.е. отношение вышеуказанных полей).
Не забудь привести конверсию к формату *float*.
Используй свойства оператора *COUNT*.
Отсортируй результат по возрастанию даты.

Результирующие столбцы

1. *dt* - год заказа (в формате даты).
2. *cnt_orders* - всего заказов.
3. *cnt_delivered_orders* - заказов с датой доставки.
4. *share_delivered* - доля доставленных заказов.

Задача 5.

Условие задачи

Для каждого клиента рассчитай количество заказов, которые он сделал за всё время, а также долю успешных заказов (т.е. со статусом Completed).
Ограничься только теми клиентами, которые совершили строго больше 3 заказов за всё время, а также хотя бы один заказ без статуса Completed.
Выведи топ-3 клиента по доле заказов со статусом Completed среди всех заказов клиента.

Результирующие столбцы

1. *customer_id* - идентификатор пользователя
2. *cnt_orders* - количество заказов пользователя.
3. *share_compl* - доля заказов со статусом Completed.

Задача 6.

Условие задачи

По таблице *databases_and_sql.shop_orders* найди два значения:

- среднее количество дней между датой доставки и датой заказа;
- среднее количество дней между датой доставки и датой заказа при условии, что в случае NULL-значения даты доставки его нужно заменить на {дата заказа + 3 дня}.

Округли значения до трёх знаков после запятой.

Результирующие столбцы

1. *general_avg* - среднее количество дней.
2. *coalesce_avg* - скорректированное среднее количество дней.

Задача 7.

Условие задачи

Для каждого месяца заказа рассчитай общее количество заказов, количество заказов с доставкой больше (строго) трёх дней, а также долю заказов с доставкой дольше 3 дней среди всех заказов.

Ограничься только вторым полугодием 2024 года.

Долю необходимо округлить до 3 знаков после запятой.

Результирующие столбцы

1. *dt* - месяц заказа (в формате даты).
2. *cnt_orders* - количество заказов.
3. *cnt_long_orders* - количество заказов с доставкой больше 3 дней.
4. *share_long* - доля заказов с доставкой больше 3 дней из всех заказов.

Задача 8.

Условие задачи

Для каждого календарного месяца (т.е. месяца в числовом формате) выведи количество заказов, количество уникальных клиентов, а также рассчитай среднее количество заказов на одного клиента.

Выведи три самых "занятых" месяца, т.е. когда среднее количество заказов на клиента наибольшее.

Результирующие столбцы

1. *dt* - месяц заказа (в числовом формате).
2. *cnt_orders* - количествло заказов.
3. *cntd_customers* - количество уникальных клиентов.
4. *cnt_per_cust* - среднее количество заказов на одного клиента.

Задача 9.

Условие задачи

Таблица *databases_and_sql.shop_orderitems* отражает связь заказа (*order_id*) и товаров в заказе (*product_id*).

Найди такие заказы, в которых было три или более одинаковых товаров, и выведи идентификатор заказа, товара и количество товаров.

Отранжируйте результат по возрастанию идентификатора заказа.

Результирующие столбцы

1. *order_id* - идентификатор заказа.
2. *product_id* - идентификатор продукта.
3. *cnt_products* - количество продуктов в заказе.

Задача 10.

Условие задачи

Для каждого месяца заказа за 2024 год из
таблицы *databases_and_sql.shop_orders* определи:

- минимальное количество дней между датой доставки и датой создания заказа;
- максимальное количество дней между датой доставки и датой создания заказа;
- среднее количество дней между датой доставки и датой создания заказа;
- медианное количество дней между датой доставки и датой создания заказа;
- разницу между средним и медианным значением.

Округли все значения до целых и отсортируй по возрастанию даты.

Результирующие столбцы

1. *dt* - месяц заказа.
2. *min_diff* - минимальное количество дней.
3. *max_diff* - максимальное количество дней.
4. *avg_diff* - среднее количество дней.
5. *median_diff* - медианное количество дней.
6. *avg_median_diff* - разница между средним и медианным значением.

Примерное задание к зачету с оценкой

Задача 1.

Клиенты магазина совершают покупки, в результате чего формируются заказы. У каждого заказа есть уникальный номер order_id, время покупки order_date, общая сумма заказа order_total.

К заказам можно писать отзывы. Отзыв тоже имеет уникальный идентификатор review_id. Вместе с текстом review_text сохраняется автор user_login, время записи review_ts и номер заказа.

Таблица, в которой менеджеры регистрируют заказы, имеет следующую структуру:

SALES (order_id, order_date, order_total, review_id, user_login, review_text, review_ts)

Что требуется в ответе

Напиши DDL-скрипт (create table …), с помощью которого можно будет создать нормализованную базу данных (3NF). Обязательно указывай первичный ключ.

Используй только названия полей из задания. Список полей приводится ниже:

- order_id,
- order_date,
- order_total,
- review_id,
- user_login,
- review_text,
- review_ts.

Задача 2.

Изучи описание ниже и попробуй составить список таблиц, которые можно использовать в базе данных интернет-магазина.

Описание

Как только появляется новый заказ, ему назначается новый уникальный номер (order_id), определяется итоговая сумма к оплате (order_total), время создания (order_date). В заказе может быть несколько товаров. Если при этом в заказ включается несколько одинаковых товаров, то соответствующее количество записывается в поле qty.

У каждого товара есть цена за единицу (unit_price), название (product_desc).

Можно оставлять отзывы к товарам. При этом у каждого отзыва есть уникальный номер (review_id), текст (review_text), автор (user_login), а также время отправки отзыва (review_ts).

Сейчас данные хранятся в таблицах со следующей структурой:

Orders (order_id, order_date, order_total, product_id, product_desc, unit_price, qty)

ProductReviews (product_id, review_id, user_login, review_text, review_ts)

Тебя попросили исправить структуру таблиц.

Что требуется в ответе

Напиши DDL-скрипт, с помощью которого можно будет создать нормализованную базу данных (3NF). Используй только названия полей из задания.

Задача 3.

Задача на проектирование базы данных для приложения Random Coffee, которое в случайном порядке формирует пары или тройки людей для проведения неформальных встреч.

Краткое описание работы сервиса Каждую неделю запускается новый раунд встреч, каждый из которых имеет две даты: дату открытия приглашения пользователей отметить согласие на участие и дату публикации пар или троек пользователей.

Пользователи регистрируются в приложении Random Coffee. В личном кабинете они могут включить или выключить своё участие. Каждый пользователь во время регистрации указывает своё имя и email. Информация о созданных парах (или тройках) в новом раунде встреч доступна в интерфейсе.

Перед каждым раундом сервис открывает запись и отправляет приглашение на участие во встречах. Пользователи отмечают своё согласие или отказ участвовать в раунде.

Затем, когда наступает дата публикации групп для встреч, программа формирует набор предложений. В одном раунде они нумеруются, начиная с 1. Также записывается время создания предложения и предполагаемые участники встречи. Обычно участников два, но иногда может быть и больше. Эта информация доступна пользователям в интерфейсе сервиса.

Раунд может быть помечен как закрытый или открытый. Это нужно для корректного отображения текущего раунда встреч.

Что требуется в ответе

Разработай скрипт DDL для создания базы данных, которую можно будет использовать в описанном выше сервисе. Ты можешь воспользоваться ERDplus для проектирования ER-модели с последующим переходом к SQL DDL.

Можно выбирать только поля из списка ниже:

- user_id — номер пользователя;
- user_name — имя пользователя;
- email — адрес электронной почты;
- active_flg — признак активности или доступности;
- round_id — номер раунда;
- open_date — дата открытия сбора заявок на участие;
- publish_date — дата публикации предложений;
- meeting_no — номер встречи (предложения);
- created_at — время создания;
- confirm_flg — признак подтверждения.

Вставь скрипт для проверки своего решения и запусти ячейку.

Примерное описание и критерии к проекту

Описание проекта

Ты — аналитик в интернет-магазине техники. Перед командой аналитики руководство поставило задачу построения дашборда, который позволит наблюдать за основными продуктовыми метриками.

Тебе необходимо подготовить данные для построения дашборда в BI-системе, используемой в компании. На нём будут 4 графика. Каждый график строится на основе одной таблицы. Твоя задача — написать 4 SQL-запроса, которые будут выводить необходимые данные.

Ниже ты найдёшь бизнес-требования от заказчика.

График 1

Необходимо вывести следующие метрики в разбивке по дням, неделям, месяцам:

- *общее кол-во заказов за период (order_cnt);*
- *кол-во уникальных покупателей за период (customer_cnt);*
- *средний чек за период (avg_price);*
- *общую выручку за период (revenue);*
- *относительное изменение выручки к предыдущему периоду (revenue_diff).*

Нас интересуют **только успешно выполненные (*Completed*) заказы**.

Так как необходимо построить только одну таблицу для трёх уровней группировок (день, неделя, месяц), то необходим столбец, который будет использоваться в качестве фильтра для управления группировкой данных.

График 2

Необходимо вывести **топ-5 самых продаваемых товаров** за 2024 год и посчитать по ним следующие метрики:

- *кол-во заказов, в которых этот товар покупали (orders_cnt);*
- *кол-во проданных единиц этого товара (order_items_cnt);*
- *кол-во заказов, где этот товар был в первом заказе покупателя (first_orders_cnt);*
- *выручку, которую принёс этот товар (revenue);*
- *долю, которую составляет выручка от продаж этого товара среди общей выручки от всех заказов (revenue_perc).*

В расчётах должны участвовать только **успешно доставленные (*Completed*) заказы**. Товары необходимо отсортировать **по убыванию количества проданных товаров**.

График 3

Необходимо рассчитать для всех городов, в которые производили доставки заказов, следующие метрики:

- *кол-во заказов с доставкой в этот город (orders_cnt);*
- *кол-во возвращённых заказов (refunded_orders_cnt);*
- *кол-во оплат успешно завершённых заказов с помощью СБП (payment_sbp_cnt);*
- *долю оплат успешно завершённых заказов с помощью СБП среди всех оплат успешных заказов (payment_sbp_perc);*
- *среднее кол-во дней от создания заказа до его доставки (avg_delivery_days_cnt) вне зависимости от статуса заказа.*

Оставить требуется только топ-5 городов, отсортированных в порядке убывания количества заказов в эти города *(orders_cnt).*

Несколько нюансов:

- используй только заказы за 2024 год;
- тип населённого пункта должен быть именно «Городом» или его сокращениями. Сёла, деревни и тому подобные не подойдут.

График 4

Рассчитай уровень удержания пользователей (*Retention Rate*) за первые 6 месяцев после их первого заказа. *Retention Rate* определяется как процент пользователей, которые совершили повторные заказы в каждом из месяцев относительно месяца их первого заказа. Есть несколько нюансов:

- для расчётов используем только успешно завершённые заказы (*Completed*);
- расчёты производим по когортам. Группируем пользователей по месяцам их первого заказа;
- в расчётах должны участвовать только «созревшие» пользователи. Например, для расчёта

Retention Rate 1-го месяца необходимо брать только тех пользователей, у которых прошло не менее 2 месяцев со дня первого заказа, для *Retention Rate* 2-го месяца — 3 месяца.

Пример расчёта *Retention Rate* 1-го месяца

1. Берём всех пользователей, которые совершили заказ в январе. Все эти пользователи попадут в одну когорту.
2. Рассчитываем, кто из пользователей когорты «созрел». Считаем, сколько дней прошло с момента первого заказа.
3. Считаем, сколько созревших пользователей в нашей когорте.
4. Считаем, сколько пользователей когорты совершило повторный заказ в период с 28-го по 55-й день включительно после первого заказа.
5. Делим число, полученное в пункте 4, на число из пункта 3. Приводим к процентам. Округляем до одного знака после запятой.

Результат должен содержать три колонки:

- *cohort* — когорта;
- *retention_month* — порядковый номер месяца после первого заказа: от 1 до 6;
- *retention_rate* — процент удержания, округлённый до 1 знака после запятой.

Пример таблицы

cohort	**retention_month**	**retention_rate**
2024-01-01	1	25.0
2024-01-01	2	15.3
2024-02-01	1	22.7

Итого

В рамках проекта тебе необходимо написать SQL-запросы и сдать их в виде файла с расширением *.ipynb (jupyter notebook)*, который будет содержать 4 отдельных SQL-запроса по одному на каждый из графиков.

Критерии оценки

Максимальная оценка — **10 баллов**. Из них:

- подготовка данных для «Графика 1» — **2,5 балла** (по 0,5 балла за каждую метрику);
- подготовка данных для «Графика 2» — **2,5 балла** (по 0,5 балла за каждую метрику);
- подготовка данных для «Графика 3» — **2,5 балла** (по 0,5 балла за каждую метрику);
- подготовка данных для «Графика 4» — **2,5 балла**. Задачу необходимо решить целиком.