**Тестовое задание для аналитиков**

Тестовое задание SQL (только один вариант ответа):

1. Какой оператор используется для добавления данных в таблицу?

a) INSERT INTO

b) UPDATE

c) DELETE FROM

d) SELECT

1. Каким оператором можно выбрать данные из нескольких таблиц?

a) UNION ALL

b) JOIN

c) INSERT

d) SELECT

1. Каким образом можно удалить таблицу?

a) DELETE

b) UPDATE

c) TRUNCATE

d) DROP

4. Что вернет следующий SQL запрос SELECT product\_id, COUNT(order\_id) FROM order\_details GROUP BY product\_id?  
a) Количество заказов для каждого клиента в базе данных  
b) Количество продуктов в каждом заказе в базе данных  
c) Количество заказов для каждого продукта в базе данных

d) Суммарное количество заказов и продуктов

1. Какой оператор используется для изменения структуры существующей таблицы в SQL?  
   a) ALTER TABLE  
   b) UPDATE TABLE  
   c) MODIFY TABLE  
   d) CHANGE TABLE

1. Что такое реляционные базы данных?
2. База данных, в которой данные хранятся в виде двумерных таблиц, связанных между собой
3. Хранит данные в формате ключ-значение
4. Любая база данных – реляционная
5. Совокупность не связанных данных

1. Как выглядит вопрос для вывода всех значений из таблицы Orders?

a) select ALL from Orders  
b) select \* from Orders  
c) select % from Orders  
d) select orders from Orders

8. Как использовать функцию COUNT() в SQL?  
a) SELECT COUNT(\*) FROM table\_name  
b) SELECT COUNT(column\_name) FROM table\_name  
c) SELECT COUNT(column\_name) AS count\_name FROM table\_name  
d) Все вышеперечисленные варианты

9. Какое ключевое слово используется в SQL для фильтрации данных в таблице?

a) FILTER;

b) SELECT;

c) WHERE;

d) HAVING.

10. Как сортировать данные в таблице в порядке возрастания?

a) ORDER BY ASC;

b) ORDER BY DESC;

c) ORDER BY;

d) GROUP BY.

**Кейс:**

Крупная компания планирует создать Систему для бронирования отелей. Система позволяет бронировать номера в отелях по всем странам. В системе можно бронировать номера в отелях, с возможностью предоплаты и без.

**Список функциональных требований:**  
1. Регистрация пользователей и создание профилей как для пользователей, так и для владельцев отелей;

2. Поиск отелей по странам и городам;

3. Просмотр информации об отеле, номерах и ценах для бронирования;

4. Бронирование номеров с возможностью предоплаты или без;

5. Роли в системе: пользователь, админ отеля;

6. Возможность отказаться от предоплаты и оплатить номер при заселении;

7. Оплата комиссии системе за бронирование;

8. Получение электронного подтверждения бронирования;

9. Интеграция с системами платежных сервисов;

10. Отчетность для администраторов, описывающая количество забронированных номеров, частоту бронирований и т.д.

**Задание:**

1. Описать систему с помощью UML-диаграмм как: Диаграмма классов, диаграмма последовательностей, ER-диаграмма.

Диаграммы доступны в гитхабе (ссылка прилагается)

1. Описать API (в REST) получения всех отелей, номеров в отелях, и бронирования. Должно быть включены: примеры запроса/ответа сервера, входные/выходные данные и тд.

API (в REST) доступны в гитхабе (ссылка прилагается)

1. Описать словами, как бы вы построили работу команды с точки зрения подхода к разработке. Вопрос не обязателен, но желателен.

Для эффективной работы команды в разработке данного проекта, я бы предложила следующий подход:

1. Определение требований и функциональности: команда должна провести встречи с заказчиком, чтобы определить все требования к проекту и функциональность, которую необходимо реализовать. Это поможет определить основные задачи и цели проекта.

2. Разработка технического задания: на основе собранных требований, команда должна разработать техническое задание, которое будет использоваться для планирования, оценки и управления проектом.

3. Разработка макетов и дизайна: команда должна разработать макеты и дизайн интерфейса, чтобы обеспечить удобство использования приложения.

4. Разработка и тестирование: команда должна разработать приложение, используя agile методологию разработки, проводить тестирование приложения и внедрять необходимые изменения и улучшения. Разбить базовый функционал на односоставные задачи и заполнить черновой вариант первых спринтов. В последствие двигаться спринтами корректируя их на регулярных встречах или созвонах с командой.

5. Развертывание и поддержка: после успешной разработки и тестирования приложения, команда должна развернуть его на сервере и начать работу по его поддержке и обновлению.

6. Мониторинг и улучшение: команда должна мониторить работу приложения и собирать данные с целью улучшения его функциональности, производительности и устранения ошибок.

7. Обратная связь: команда должна постоянно поддерживать связь с заказчиком и пользователями приложения, чтобы получать обратную связь и улучшать его функциональность и удобство использования.

Все эти шаги должны выполняться в тесном сотрудничестве и коммуникации между членами команды, чтобы обеспечить эффективность и качество работы.

Для построения работы команды с точки зрения подхода к разработке я бы рекомендовал использовать Agile-методологию, такую как Scrum.