**Тестовое задание для аналитиков**

Тестовое задание на знание SQL (только один вариант ответа)

1. Какой оператор используется для добавления данных в таблицу?
2. **INSERT INTO**
3. UPDATE
4. DELETE FROM
5. SELECT
6. Каким оператором можно выбрать данные из нескольких таблиц?
7. UNION ALL
8. **JOIN**
9. INSERT
10. SELECT
11. Каким образом можно удалить таблицу?
12. DELETE
13. UPDATE
14. TRUNCATE
15. **DROP**
16. Что вернет следующий SQL запрос SELECT product\_id, COUNT(order\_id) FROM order\_details GROUP BY product\_id?
17. Количество заказов для каждого клиента в базе данных
18. Количество продуктов в каждом заказе в базе данных
19. **Количество заказов для каждого продукта в базе данных**
20. Суммарное количество заказов и продуктов
21. Какой оператор используется для изменения структуры существующей таблицы в SQL?
22. **ALTER TABLE**
23. UPDATE TABLE
24. MODIFY TABLE
25. CHANGE TABLE
26. Что такое реляционные базы данных?
27. **База данных, в которой данные хранятся в виде двумерных таблиц, связанных между собой**
28. Хранит данные в формате ключ-значение
29. Любая база данных – реляционная
30. Совокупность не связанных данных
31. Как выглядит запрос для вывода всех значений из таблицы Orders?
32. select ALL from Orders
33. **select \* from Orders**
34. select % from Orders
35. select orders from Orders

8. Как использовать функцию COUNT() в SQL?

1. SELECT COUNT(\*) FROM table\_name
2. SELECT COUNT(column\_name) FROM table\_name
3. SELECT COUNT(column\_name) AS count\_name FROM table\_name
4. **Все вышеперечисленные варианты**

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

1. FILTER;
2. SELECT;
3. **WHERE;**
4. HAVING.

10. Как сортировать данные в таблице в порядке возрастания?

1. **ORDER BY ASC;**
2. ORDER BY DESC;
3. ORDER BY;
4. 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.
2. Описать API (в REST) получения всех отелей, номеров в отелях, и бронирования. Должно быть включены: примеры запроса/ответа сервера, входные/выходные данные и тд.
3. Описать словами, как бы вы построили работу команды с точки зрения подхода к разработке. Вопрос не обязателен, но желателен.

**Решение**

1. UML-диаграммы: В приложении.

2. Примеры API (в REST):

- Получение всех отелей: GET /hotels - запрос на получение списка всех отелей. Ответ сервера может быть в виде JSON с данными об отелях, их номерах и ценах.

- Получение номеров в отеле: GET /hotels/{id}/rooms - запрос на получение списка номеров в определенном отеле с идентификатором id. Ответ сервера может содержать информацию о номерах, их типах и доступности.

- Бронирование: POST /hotels/{id}/booking - запрос на бронирование номера в отеле с идентификатором id. Входные данные могут содержать информацию о выбранном номере, дате заезда, количестве гостей и другие необходимые данные. Ответ сервера может содержать подтверждение бронирования или ошибку, если бронирование не удалось.

3. Построение работы команды с точки зрения подхода к разработке:

- Сначала будет проведен анализ требований и определение функциональных и нефункциональных требований системы Заказчика.

- Затем мне совместно с командой нужно создать дизайн системы, включая диаграммы (как упомянуто выше) и планы разработки.

- Команда разработки будет делиться на различные роли, такие как разработчики, тестировщики и дизайнеры. Каждый будет отвечать за свою область работы.

- За основу философии Agile можно взять методологию Scrum, для управления проектом и планирования итераций разработки.

- Разработка будет выполняться поэтапно, с регулярными ревью и тестированием, чтобы обеспечить качество и соответствие требованиям.

- После завершения разработки, команда будет осуществлять передачу разработанной системы в эксплуатацию и поддержку, а также будет собирать обратную связь от пользователей для внесения улучшений и обновлений.