Вопросы

1. Дайте развернутое определение понятию база данных, предметная область, данные.
2. Иерархическая модель данных. Основные понятия, графическое изображение, примеры.
3. Сетевая модель данных. Основные понятия, графическое изображение, примеры.
4. Реляционная модель данных. Основные понятия, графическое изображение, примеры.
5. Термины и определения реляционных баз данных.
6. Основные компоненты систем управления реляционными базами данных.
7. Нормализация таблиц реляционной базы данных. Определение, виды, понятия.
8. Первая нормальная форма реляционной модели данных. Определение, требования, примеры.
9. Вторая нормальная форма реляционной модели данных. Определение, требования, примеры.
10. Третья нормальная форма реляционной модели данных. Определение, требования, примеры.
11. Проектирование связей между таблицами. Назначение, основные правила, варианты поведения зависимой таблицы.
12. Концептуальное проектирование БД
13. Первичный ключ. Определение, назначение, примеры.
14. Внешний ключ. Определение, назначение, примеры.
15. Методы организации целостности данных. Описание, примеры.
16. Типы данных. Группы и краткое описание.
17. Логическое проектирование.
18. Создание, модификация и удаление таблиц.
19. Создание, модификация, удаление базы данных. Синтаксис, параметры.
20. Порядок написания команд в запросе.
21. Порядок выполнения команд в запросе. Обосновать.
22. Оператор выбора: синтаксис, ключевые слова и псевдонимы.
23. Предложение группировки: синтаксис, принцип работы.
24. Предложение фильтрации: синтаксис, принцип работы.
25. Сортировка. Синтаксис, принцип работы.
26. Предикаты, создание связей между таблицами.
27. Использование функций агрегирования.
28. Групповая фильтрация. Синтаксис, примеры.
29. Вложенные запросы. Виды и отличия.
30. Простые вложенные запросы. Синтаксис, примеры.
31. Коррелированные вложенные запросы. Синтаксис, примеры.
32. Операторы манипулирования данными (INSERT, UPDATE, DELETE). Примеры использования.
33. Элементы Microsoft SQL Server: представления. Примеры создания, использования и удаления средствами языка.
34. Элементы Microsoft SQL Server: ограничения. Примеры создания, использования и удаления средствами языка.
35. Использование имен корреляции (алиасов, псевдонимов).
36. Использование агрегатных функций в запросах, с группировками.
37. Специальные операторы SQL IN, BETWEEN, IS NULL.
38. Специальный оператор LIKE.
39. Объединение таблиц с использованием оператора JOIN.
40. SQL: запрос выборки данных, функции агрегирования AVG, SUM, MAX, MIN.
41. Форматирование выходных данных запроса, секции GROUP BY и HAVING.
42. Язык Transact-SQL: синтаксис создания вложенных подзапросов.
43. Язык Transact-SQL: синтаксис создания связанных подзапросов. Оператор EXISTS.
44. Вложенные и связанные подзапросы. Операторы ANY, SOME, ALL.
45. SQL: запрос выборки данных по нескольким таблицам, оператор JOIN, левое, правое и внутреннее соединение.
46. В чем заключается разница между предложениями ON и WHERE?
47. В чем заключается разница между предложениями WHERE и HAVING?
48. Возможно ли обращение к псевдониму столбца, определенному предложением SELECT в предложении WHERE? Ответ обосновать.
49. Используется ли тип данных FLOAT для представления цены единицы товара? Ответ обосновать.
50. Методы импорта данных из таблиц Excel в MS SQL.

Ответы

1. База данных - организованная коллекция информации, которая хранится и управляется с помощью системы управления базами данных.

Предметная область - это сфера знаний, для которой создается база данных. Данные - факты, представленные в виде символов, чисел или текста.

1. Иерархическая модель данных - структура данных, организованная как иерархия, где каждый элемент данных имеет родителя и дочерние элементы. Примером иерархической модели данных может быть организационная структура компании.
2. Сетевая модель данных - структура данных, где каждый элемент может иметь несколько родителей и дочерних элементов, образуя сеть связей. Примером сетевой модели данных может служить информационная база книжного магазина, где книги могут быть связаны с несколькими авторами
3. Реляционная модель данных - структура данных, организованная в виде таблиц, связанных между собой с помощью ключей. Примером реляционной модели данных может быть база данных клиентов и заказов в интернет-магазине.
4. Термины и определения реляционных баз данных включают в себя термины такие как таблица, столбец, строка, первичный ключ, внешний ключ и т.д.
5. Основные компоненты систем управления реляционными базами данных включают в себя систему управления базами данных, язык структурированных запросов SQL, среду разработки и администрирования баз данных и др.
6. Нормализация таблиц реляционной базы данных - процесс организации данных в таблицах таким образом, чтобы убрать излишнюю информацию и избежать аномалий при добавлении, изменении и удалении данных.
7. Первая нормальная форма реляционной модели данных - требует, чтобы все значения в столбцах таблицы были атомарными и не повторялись.
8. Вторая нормальная форма реляционной модели данных - требует, чтобы каждый столбец в таблице зависел от ее первичного ключа.
9. Третья нормальная форма реляционной модели данных - требует, чтобы все неключевые столбцы в таблице зависели только от первичного ключа.
10. Проектирование связей между таблицами - процесс определения связей между таблицами в базе данных, определяющий как данные между ними будут связаны.
11. Концептуальное проектирование БД - процесс определения структуры и организации данных в базе данных на уровне концепций и отношений между ними.
12. Первичный ключ - уникальный идентификатор записи в таблице, который обеспечивает его уникальность.
13. Внешний ключ - поле или набор полей, значение которых связано с значением первичного ключа в другой таблице.
14. Методы организации целостности данных включают в себя ограничения целостности, триггеры и хранимые процедуры.
15. Типы данных - различные категории данных, которые могут быть хранены в базе данных, такие как числа, строки, даты и др.
16. Логическое проектирование - процесс определения структуры и связей между данными на уровне схемы базы данных.
17. Создание, модификация и удаление таблиц - операции, которые позволяют управлять структурой базы данных путем создания новых таблиц, изменения их структуры и удаления ненужных таблиц.
18. Создание – CREATE, Модификация – ALTER, Удаление – DROP
19. Порядок написания команд в запросе: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY
20. Порядок выполнения команд в запросе:

1. FROM - выбор таблицы или представления.

2. WHERE - фильтрация строк.

3. GROUP BY - группировка строк.

4. HAVING - фильтрация групп.

5. SELECT - выбор столбцов.

6. ORDER BY - сортировка результатов.

22. Оператор выбора:

SELECT column1, column2

FROM table\_name

WHERE condition;

23. Предложение группировки:

SELECT column1, COUNT(\*)

FROM table\_name

GROUP BY column1;

24. Предложение фильтрации:

SELECT column1

FROM table\_name

WHERE condition;

25. Сортировка:

SELECT column1

FROM table\_name

ORDER BY column1 DESC;

26. Предикаты и создание связей между таблицами.

SELECT column1

FROM table1

INNER JOIN table2

ON table1.column\_id = table2.column\_id;

27. Использование функций агрегирования:

SELECT MAX(column1)

FROM table\_name;

28. Групповая фильтрация:

SELECT column1, COUNT(\*)

FROM table\_name

GROUP BY column1

HAVING COUNT(\*) > 1;

29. Вложенные запросы - это запросы SQL, вложенные в другие запросы для более сложной обработки данных. Отличия между простыми и коррелированными вложенными запросами в том, что коррелированные запросы могут обращаться к внешним таблицам.

30. Простые вложенные запросы:

SELECT column1

FROM table\_name

WHERE column1 IN (SELECT column1 FROM table2);

31. Коррелированные вложенные запросы:

SELECT column1

FROM table\_name t1

WHERE EXISTS (SELECT column1 FROM table2 t2 WHERE t1.column\_id = t2.column\_id);

32. Операторы манипулирования данными:

INSERT - добавление в таблицу, UPDATE - изменение в таблице, DELETE - удаление из таблицы.

33. Представления (Views) в Microsoft SQL Server позволяют создавать виртуальные таблицы на основе запросов. Пример создания:

CREATE VIEW view\_name AS

SELECT column1, column2

FROM table\_name;

34. Ограничения (Constraints) в Microsoft SQL Server позволяют определять правила целостности данных. Пример создания:

ALTER TABLE table\_name

ADD CONSTRAINT constraint\_name PRIMARY KEY (column1);

35. Использование имен корреляции (алиасов, псевдонимов) позволяет задавать краткие имена таблицам и столбцам для удобства чтения запросов SQL.

SELECT t1.column1

FROM table\_name t1;

36. Использование агрегатных функций в запросах с группировками:

SELECT column1, COUNT(\*)

FROM table\_name

GROUP BY column1;

37. Операторы SQL IN, BETWEEN, IS NULL:

- IN используется для проверки значения в наборе значений.

- BETWEEN используется для проверки, находится ли значение в диапазоне.

- IS NULL используется для проверки, равно ли значение NULL.

38. Специальный оператор LIKE используется для поиска строк, которые соответствуют определенному шаблону.

Пример использования:

SELECT column1

FROM table\_name

WHERE column1 LIKE 'A%';

39. Объединение таблиц с использованием оператора JOIN:

SELECT column1

FROM table1

INNER JOIN table2

ON table1.column\_id = table2.column\_id;

40. Запрос выборки данных с функциями агрегирования:

SELECT AVG(column1), SUM(column2), MAX(column3), MIN(column4)

FROM table\_name;

41. Форматирование выходных данных с GROUP BY и HAVING:

SELECT column1, COUNT(\*) as count

FROM table\_name

GROUP BY column1

HAVING COUNT(\*) > 1;

42. Создание вложенных подзапросов в Transact-SQL:

SELECT column1

FROM table\_name

WHERE column1 IN (SELECT column1 FROM table2);

43. Создание связанных подзапросов с оператором EXISTS:

SELECT column1

FROM table\_name t1

WHERE EXISTS (SELECT column1 FROM table2 t2 WHERE t1.column\_id = t2.column\_id);

44. Разница между вложенными и связанными подзапросами заключается в том, что связанный подзапрос зависит от внешнего запроса, в то время как вложенный подзапрос может быть независимым.

45. Запрос выборки данных по нескольким таблицам с использованием оператора JOIN:

SELECT table1.column1, table2.column2

FROM table1

LEFT JOIN table2 ON table1.column\_id = table2.column\_id;

46. Разница между предложениями ON и WHERE при использовании JOIN заключается в том, что ON определяет условие объединения таблиц, в то время как WHERE определяет фильтрацию строк.

47. Разница между предложениями WHERE и HAVING заключается в том, что WHERE применяется к отдельным строкам, а HAVING применяется к результатам группировки.

48. Да, обращение к псевдониму столбца, определенному в предложении SELECT, возможно при использовании подзапросов. Операции обязательно выполняются после формирования данных исходной таблицы.

49. Тип данных FLOAT не рекомендуется для представления цен единицы товара, так как он может привести к неточным вычислениям из-за плавающей запятой. Рекомендуется использовать DECIMAL или MONEY для точного представления цен.

50. Для импорта данных из таблиц Excel в MS SQL можно использовать различные методы, такие как использование встроенных средств импорта данных в SQL Server Management Studio, написание скриптов на Transact-SQL для импорта данных, или использование инструментов ETL (Extract, Transform, Load).