***Описание запросов***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | | | **Имя запроса** | **Текст запроса** | **Запрос на SQL** |
| 1 | | | Научитесь создавать простые запросы, используя *Мастер запросов*. | Выводит все данные из таблицы Троллейбус | SELECT Троллейбус.\*  FROM Троллейбус; |
| 2 | | | Изучите работу *Конструктора запросов*. Просмотрите в режиме *Конструктора* запросы, созданные *Мастером запросов* | Выводит все данные из таблицы Водитель | SELECT \*  FROM Троллейбус; |
| 3 | | | Создайте запросы с выражениями, в которых используются условный оператор. | Вывожу остановки, у которых название = «Дом Мод» OR «Мтв» | SELECT Остановка.\*, \*  FROM Остановка  WHERE (((Остановка.Название)="Дом Мод") OR ((Остановка.Название)="Мтв")); |
| 4 | | | Создайте запрос с предикатом TOP | Вывести первое значение из таблицы | SELECT TOP 1 Троллейбус.[Код троллейбуса], Троллейбус.[Номер троллейбуса], Троллейбус.[Тип]  FROM Троллейбус; |
| 5 | | | Вложенный подзапрос | Выводит строку по отсортированным двум таблицами | SELECT Остановка.Название  FROM Остановка  WHERE Остановка.[Название остановки] IN (  SELECT Состоит.[Название остановки]  FROM Состоит  WHERE Состоит.[Название остановки] = true  ); |
| 5 | | | Внутреннее соединение | Результатом запроса будет набор данных, содержащий все строки из обеих таблиц, у которых значения в поле "Код водителя" совпадают. | SELECT \*  FROM Рейс INNER JOIN Водитель ON Водитель.[Код водителя]=Рейс.[Код водителя]; |
| 6 | | | Внешнее соединение | Результатом запроса будет набор данных, содержащий все строки из таблицы "Водитель" и соответствующие строки из таблицы "Рейс", если совпадение по полю "Код водителя" есть. Если совпадение не найдено, то значения полей из таблицы "Рейс" будут пустыми. | SELECT \*  FROM Водитель LEFT JOIN Рейс ON Водитель.[Код водителя]=Рейс.[Код водителя]; |
| 7 | | | Self-соединение | Показывает строки, где имена совпадают | SELECT V1.Имя, V1.Удостоверение, V2.Имя, V2.Удостоверение  FROM Водитель AS V1, Водитель AS V2  WHERE V1.Удостоверение=V2.Удостоверение And V1.Имя<>V2.Имя; |
| 8 | | | Union | Результатом выполнения всего запроса будет набор данных с двумя строками, содержащими названия "Дом Мод" и "Мадагаскар". | SELECT Остановка.Название  FROM Остановка  WHERE (((Остановка.Название)="Дом Мод"));    UNION    SELECT Остановка.Название  FROM Остановка  WHERE (((Остановка.Название)="Мадагаскар")); |
| 9 | | | Создайте запрос из фильтра. Создайте расширенный фильтр и сохраните его в виде запроса | Вывести из таблицы строку, которая = «Мадагаскар» | SELECT Остановка.Название  FROM Остановка  WHERE (((Остановка.Название)="Мадагаскар")); |
| 10 | | | Создайте запросы на изменение | | |
| 11 | | CREATE TABLE | Создаю таблицу Инструменты | CREATE TABLE INSTRUMENTS  (  Идентификатор COUNTER PRIMARY KEY,  Имя VARCHAR(50),  Мощность INTEGER  ); |
| 12 | INSERT INTO | Добавляю в таблицу инструменты новое поле | INSERT INTO INSTRUMENTS ( Идентификатор, Имя, Мощность )  SELECT 2 AS Выражение1, "Болгарка" AS Выражение2, 90 AS Выражение3; |
| 13 | Создайте запрос на обновление записей ( UPDATE SET) | Обновляю значение поля Мощность в таблице INSTRUMENTS, у инструмента, идентификатор которого = 1. | UPDATE INSTRUMENTS SET INSTRUMENTS.Мощность = 150  WHERE (((INSTRUMENTS.[Идентификатор])=1)); |
| 14 | Создайте запрос на на удаление записей ( DELETE) | Удаляю значение поля Мощность в таблице INSTRUMENTS, у инструмента, идентификатор которого = 1. | DELETE FROM INSTRUMENTS  WHERE Идентификатор = 1; |
| 15 | Вычислений итоговых значений по всем записям | Группирует кондукторов по именам | SELECT Имя  FROM Кондуктор  GROUP BY Имя; |
| 16 | EXIST | Возвращает значение, если оно существует в таблице | SELECT \*  FROM Водитель  WHERE EXISTS  (SELECT \*  FROM Рейс  WHERE Рейс.[Код водителя] = Водитель.[Код водителя]); |
| 17 | Вычисляемое поле | Создает новый столбец с вычисляемым значением | SELECT INSTRUMENTS.Мощность, INSTRUMENTS.[Мощность]/1000 AS кВТ, \*  FROM INSTRUMENTS; |
| 18 | Перекрестный запрос | «Транспонирует» таблицу | TRANSFORM Min(Остановка.[Координаты]) AS [Min-Координаты]  SELECT Остановка.[Остановка], Min(Остановка.[Координаты]) AS [Итоговое значение Координаты]  FROM Остановка  GROUP BY Остановка.[Остановка]  PIVOT Остановка.[Название]; |
| 19 | Запрос с параметром | Выводит значения, которые совпадают с введенным параметром | SELECT \*  FROM Кондуктор  WHERE Имя = Имя\_кондуктора; |
| 20 | Запрос сравнения с образцом | Выводит остановки, которые совпадают с образцом | SELECT Остановка.Название  FROM Остановка  WHERE (((Остановка.Название)="Мадагаскар")); |
| 21 | Запрос со сравнением | Выводит остановки, если какое-то совпадает с условием | SELECT Остановка.\*, \*  FROM Остановка  WHERE (((Остановка.Название)="Дом Мод") OR ((Остановка.Название)="Мтв")); |
| 22 | Запрос с оператором строковых значений | Выводит кондукторов, если выполняется условие на проверку строки | SELECT \*  FROM Кондуктор  WHERE [Имя] LIKE 'Г\*'; |
| 23 | Запрос с оператором обработки дат | Выводит рейс, если время начала меньше заданного | SELECT \*  FROM Рейс  WHERE Рейс.[Время начала] < DATE(); |
| 24 | Запрос с оператором условных выражений | Проверяет на условие номер троллейбуса, и выводит чет./нечет. | SELECT IIF ([Номер троллейбуса] MOD 2 = 0, "Четный", "Нечетный") AS ["Тип"], \*  FROM Троллейбус; |
| 25 | Запрос с комплексными текстовыми константами и константными датами | Выводит строки, где дата меньше сегодняшней | SELECT \*  FROM Рейс  WHERE Рейс.[Время начала] < #03/15/1990#); |
| 26 | Запрос для вычислений итоговых значений по группам записей | Выводит отсортированные имена по определенным параметрам | SELECT Имя  FROM Кондуктор  GROUP BY Имя  HAVING Имя LIKE 'Г\*'; |
| 27 | Запрос с сохраненным именованным запросом как источником исходных данных. | Выводит троллейбус с номером 8 из запроса {Top 1} | SELECT \*  FROM [Top 1]  WHERE [Номер троллейбуса] = 8; |

|  |  |  |
| --- | --- | --- |
| **№** | **Вид запроса** | **Шифр (ы)** |
| 1 | Запросы с выражениями, использующие   1. операторы строковых значений 2. все операторы сравнения с образцом 3. операторы обработки дат 4. операторы условных выражений 5. статистические функции по подмножеству 6. комплексные текстовые константы и константы даты | 21  19 20  23  22 24  20    25 |
| 2 | Использование логических операторов, включающих в себя комбинацию «И», «ИЛИ» | 3 21 |
| 3 | Запрос с вычисляемым полем | 17 |
| 4 | Запрос с предикатом TOP | 4 |
| 5 | Запрос с внутренним соединением по нескольким полям | 10 |
| 6 | Запрос с внутренним   неэквивалентным соединением | 5 |
| 7 | Запрос с использованием  внешнего соединения (LEFT, RIGHT) | 6 |
| 8 | Запрос с self-соединением (самообъединение) | 7 |
| 9 | Подзапросы   * вложенный подзапрос * коррелированный подзапрос | 5  6 |
| 10 | Запрос с оператором UNION | 8 |
| 12 | Запрос с предикатом EXISTS | 16 |
| 13 | Запрос с сохраненным именованным запросом как источником исходных данных | 27 |
| 14 | Перекрестный запрос | 18 |
| 15 | Запросы с параметрами | 19 |
| 16 | Запросы на группирование данных   * Вычислений итоговых значений по всем записям * Вычислений итоговых значений по группам записей | 15    26 |
| 17 | Запросы на удаление записи | 14 |
| 18 | Запросы на обновление записи | 13 |
| 19 | Запросы на добавление записей | 12 |
| 20 | Запросы на создание таблицы | 11 |