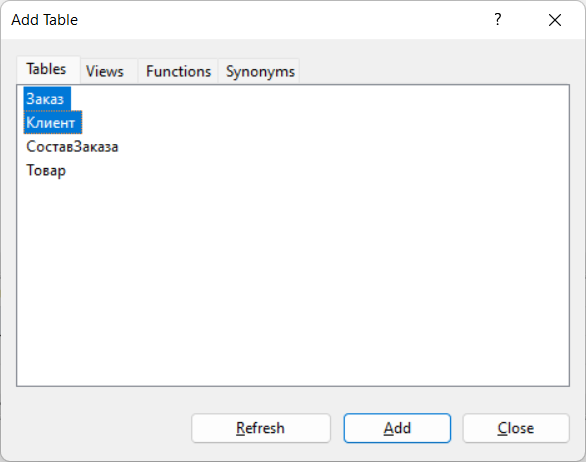
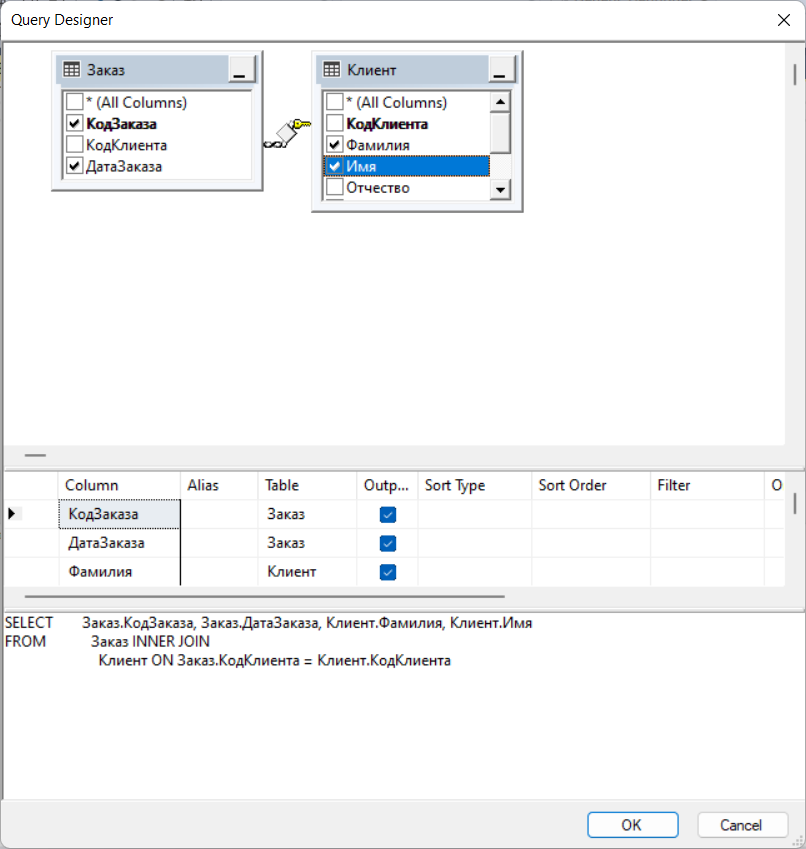
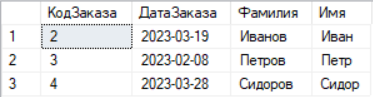
**Многотабличный запрос выборки с сортировкой и отбором данных (INNER JOIN, WHERE, ORDER BY)**

Создал запрос с помощью конструктора запросов:







**Многотабличный запрос выборки с сортировкой и отбором данных (INNER JOIN, WHERE, ORDER BY)**

SELECT Клиент.Фамилия, Клиент.Имя, Заказ.ДатаЗаказа

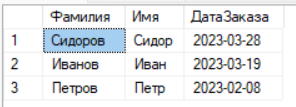
FROM Клиент

INNER JOIN Заказ

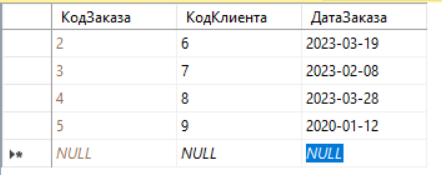
ON Клиент.КодКлиента = Заказ.КодКлиента

WHERE Заказ.ДатаЗаказа > '2022-01-01'

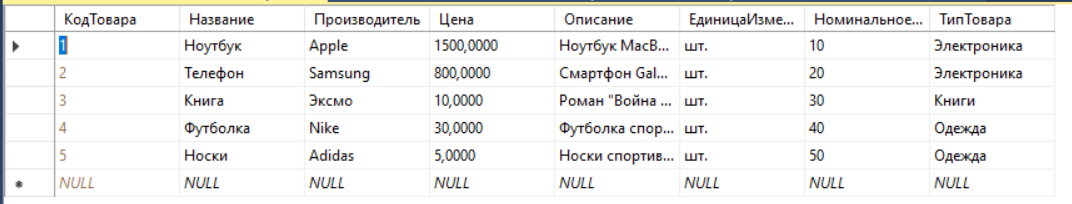
ORDER BY Заказ.ДатаЗаказа DESC;



Данные заказов:



Данные клиентов:



SELECT - определяет поля, подлежащие выводу в выходной набор.

FROM - определяет имена таблиц, включенных в запрос. Т.к запросе участвуют несколько таблиц, то для исключения двусмысленности имена полей следует записывать в полной форме: СоставЗакза.КодТовара.

WHERE - задает условия, которым должны удовлетворять записи в результирующей таблице.

INNER JOIN - это оператор SQL, который используется для соединения двух или более таблиц в базе данных. Он возвращает строки из обеих таблиц, которые удовлетворяют условию соединения. Если строки из обеих таблиц не удовлетворяют условию соединения, они не включаются в результат. Условие соединения указывается после ключевого слова ON

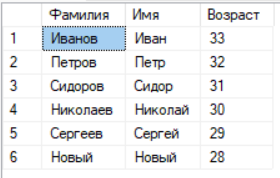
ORDER BY - это оператор SQL, который используется для сортировки результатов запроса по одному или нескольким столбцам.

* ASC — это ключевое слово, которое указывает на сортировку в порядке возрастания (по умолчанию).
* DESC — это ключевое слово, которое указывает на сортировку в порядке убывания.

**Запрос с применением вычисляемых полей (примечание: не применять агрегатные функции), например, вычислить возраст, полную цену, и типа подобное.**

SELECT Фамилия, Имя, DATEDIFF(year, ДатаРождения, GETDATE()) AS Возраст

FROM Клиент;



В DATEDIFF вычисляет разницу между датами. мы указываем, что хотим получить результат запроса в годах, а параметры для вычисления возраста – текучая дата и дата рождения клиента. В результате он возвращает столбец, который мы называем возраст.

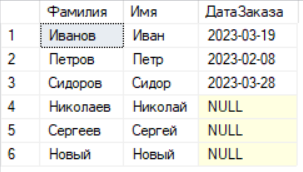
запрос выборки с внешним объединением двух отношений (LEFT|RIGHT JOIN)

SELECT Клиент.Фамилия, Клиент.Имя, Заказ.ДатаЗаказа

FROM Клиент

LEFT JOIN Заказ

ON Клиент.КодКлиента = Заказ.КодКлиента;

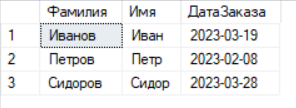


SELECT Клиент.Фамилия, Клиент.Имя, Заказ.ДатаЗаказа

FROM Клиент

RIGHT JOIN Заказ

ON Клиент.КодКлиента = Заказ.КодКлиента;



LEFT JOIN и RIGHT JOIN — это два типа внешних соединений (OUTER JOIN) в SQL. Они используются для соединения двух таблиц и возвращения строк из обеих таблиц, даже если в одной из таблиц нет соответствующих строк.

Основное отличие между LEFT JOIN и RIGHT JOIN заключается в том, какие строки включаются в результат, если в одной из таблиц нет соответствующих строк.

* LEFT JOIN возвращает все строки из левой таблицы и соответствующие строки из правой таблицы. Если в правой таблице нет соответствующих строк, то в результирующей таблице для этих строк будут значения NULL.
* RIGHT JOIN возвращает все строки из правой таблицы и соответствующие строки из левой таблицы. Если в левой таблице нет соответствующих строк, то в результирующей таблице для этих строк будут значения NULL.

**Запрос группировкой, вычислением итогов и отбором данных (GROUP BY, HAVING)**

SELECT Товар.ТипТовара, SUM(СоставЗаказа.Количество) AS ОбщееКоличество

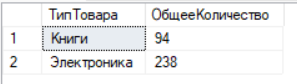
FROM Товар

INNER JOIN СоставЗаказа

ON Товар.КодТовара = СоставЗаказа.КодТовара

GROUP BY Товар.ТипТовара

HAVING SUM(СоставЗаказа.Количество) > 10;





GROUP BY — это оператор SQL, который используется для группировки строк в результирующей таблице по одному или нескольким столбцам. Он часто используется с агрегатными функциями (например, SUM, COUNT, AVG и т. д.), чтобы вычислить итоговые значения для каждой группы.

HAVING — это оператор SQL, который используется для фильтрации групп в результирующей таблице после группировки с помощью оператора GROUP BY. Он аналогичен оператору WHERE, но используется для фильтрации групп, а не отдельных строк.

**Запрос на добавление (INSERT INTO)**

INSERT INTO Клиент (Фамилия, Имя, Отчество, Телефон, Email, ДатаРождения)

VALUES ('Новый', 'Новый', 'Новый', '+375291111111', 'new@example.com', '1995-06-06');

****

INSERT INTO - это оператор SQL, который используется для добавления новых строк в таблицу.

**Запрос на удаление**

DELETE FROM Клиент WHERE Фамилия = 'Новый';

****

DELETE FROM - это оператор SQL, который используется для удаления указанной строки из таблицы

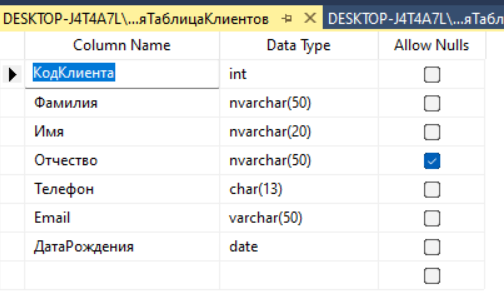
**Запрос на обновление.**

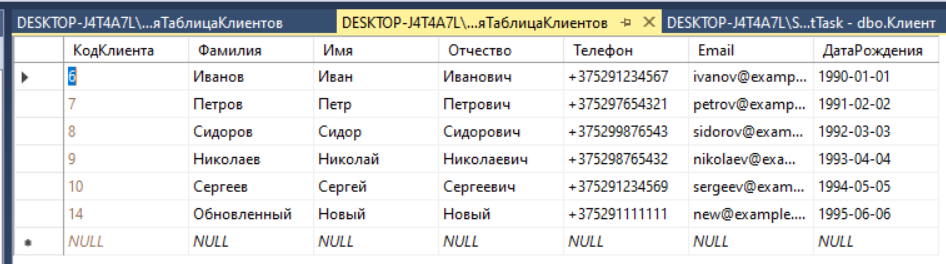
UPDATE Клиент SET Фамилия = 'Обновленный' WHERE Фамилия = 'Новый';

****

UPDATE - это оператор SQL, который используется для обновления существующих строк в таблице.

**Запрос на создание новой таблицы на основе существующей.**

****

****

SELECT \* INTO НоваяТаблицаКлиентов FROM Клиент;

SELECT \* INTO - это оператор SQL, который используется для создания новой таблицы и копирования в нее данных из существующей таблицы.

**Запрос на объединение (UNION), где применимо (только в SQL).**

SELECT Фамилия FROM Клиент

UNION ALL

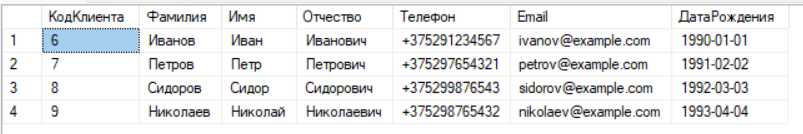
SELECT Название FROM Товар;

****

UNION - это оператор SQL, который используется для объединения результатов двух или более SELECT запросов в одну таблицу. Он возвращает все строки из всех запросов, но удаляет дубликаты. А ALL оставляет все дубликаты.

**Вложенный запрос на SQL (вложение во фразе WHERE).**

SELECT \* FROM Клиент WHERE КодКлиента IN (SELECT КодКлиента FROM Заказ);

****

Вложенный запрос - это запрос, который находится внутри другого запроса и используется для обработки данных, которые будут использованы в основном запросе в качестве условия для дальнейшей обработки данных. В этом конкретном примере вложенный запрос SELECT КодКлиента FROM Заказ выбирает все значения КодКлиента из таблицы Заказ. Затем эти значения используются во внешнем запросе как условие для выбора строк из таблицы Клиент. В результате возвращаются все строки из таблицы Клиент, у которых значение КодКлиента присутствует в таблице Заказ.

\* - означает, что будут выбраны все столбцы.

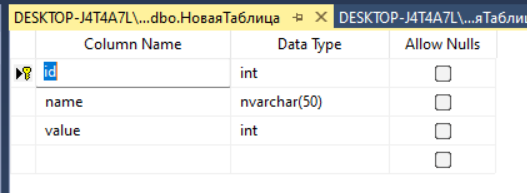
**Запрос на создание новой таблицы.**

CREATE TABLE НоваяТаблица (

id INT PRIMARY KEY,

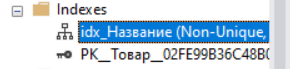
name NVARCHAR(50) NOT NULL,

value INT NOT NULL);

****

**Запрос на создание индекса.**

CREATE INDEX idx\_Название ON Товар (Название);



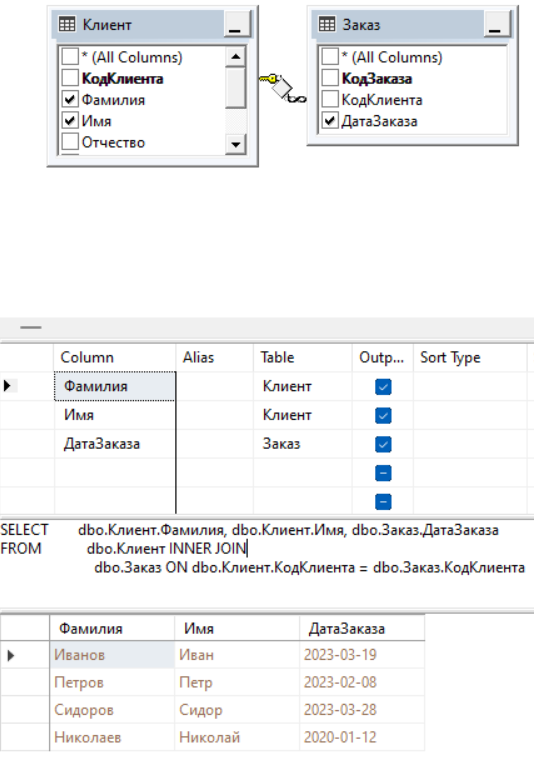
**Запрос на создание представления, объединяющего данные двух таблиц.**

CREATE VIEW ЗаказыКлиентов AS

SELECT Клиент.Фамилия, Клиент.Имя, Заказ.ДатаЗаказа

FROM Клиент

INNER JOIN Заказ ON Клиент.КодКлиента = Заказ.КодКлиента;

****