

Лабораторная работа № 4.

Выборка и манипулирование данными с помощью команд языка SQL

Подготовительные действия

1. Запустите утилиту SQL Server Management Studio, в окне Обозреватель объектов (*Object Explorer*) в древовидной структуре раскройте папку Базы данных (*Databases*).

2. С помощью команды меню *File - Open - File* загрузите сценарий из файла D:\FIO\TRADE_XXX\CreateDB_Insert_Data.sql в Окно запросов (*Query*).

3. Выполните сценарий, нажав на панели инструментов кнопку Выполнить (*Execute*) (или клавишу F5). В результате будет создана база данных TRADE_XXX.

4. Обновите данные в окне Обозреватель объектов (*Object Explorer*). Для этого используйте команду Обновить (*Refresh*) в контекстном меню папки Базы данных (*Databases*) или соответствующую кнопку в верхней части окна. В результате база данных TRADE_XXX станет видимой в окне Обозреватель объектов (*Object Explorer*).

5. Закройте Окно запросов (*Query*), содержащее сценарий CreateDB_Insert_Data.sql. Затем на панели инструментов нажмите кнопку Новый запрос (*New Query*), и откройте новое пустое Окно запросов (*Query*), предназначенное для формирования нового сценария DataManipulation.sql. Сделайте активной созданную базу данных TRADE_XXX:

```
USE TRADE_XXX  
GO
```

Практические задания

Задание 1. Выборка данных из таблиц и представлений. Оператор SELECT

Выполните различные варианты команды выборки данных SELECT:

1. Выборка всех столбцов и всех строк одной таблицы Locations (Рисунок 52).

```
SELECT * FROM Locations
```

	LocationID	Country	Region	City	Address	Phone
1	101	Беларусь	Витебская	Полоцк	ул.Лесная, 6	+375172691376
2	102	Беларусь	Гродненская	Лида	ул.Моховая, 12	NULL
3	103	Беларусь		Минск	ул.Маркса, 24	NULL
4	201	Россия	Московская	Королев	ул.Труда, 8	387-23-04
5	202	Россия		Москва	ул.Тверская, 25	900-8876
6	301	Украина		Киев	ул. Крещатик, 14	NULL
7	302	Украина	Львовская	Моршин	ул.Франко, 24	NULL

Рисунок 52. Выборка данных из одной таблицы.

2. Выборка некоторых столбцов и всех строк (вертикальный фильтр) из одной таблицы (Рисунок 53).

```
SELECT City, Address, Phone FROM Locations
```

	City	Address	Phone
1	Полоцк	ул.Лесная, 6	+375172691376
2	Лида	ул.Моховая, 12	NULL
3	Минск	ул.Маркса, 24	NULL
4	Королев	ул.Труда, 8	387-23-04
5	Москва	ул.Тверская, 25	900-8876
6	Киев	ул. Крещатик, 14	NULL
7	Моршин	ул.Франко, 24	NULL

Рисунок 53. Вертикальный фильтр.

3. Выборка всех столбцов и некоторых строк (горизонтальный фильтр) из одной таблицы (Рисунок 54).

```
SELECT * FROM Locations
WHERE Country = 'Беларусь' AND City != 'Минск'
```

	LocationID	Country	Region	City	Address	Phone
1	101	Беларусь	Витебская	Полоцк	ул.Лесная, 6	+375172691376
2	102	Беларусь	Гродненская	Лида	ул.Моховая, 12	NULL

Рисунок 54. Горизонтальный фильтр.

4. Выборка некоторых столбцов и некоторых строк (вертикальный и горизонтальный фильтры) из одной таблицы (Рисунок 55).

```
SELECT City, Address, Phone
FROM Locations
WHERE Country = 'Беларусь' AND City != 'Минск'
```

	City	Address	Phone
1	Полоцк	ул.Лесная, 6	+375172691376
2	Лида	ул.Моховая, 12	NULL

Рисунок 55. Вертикальный и горизонтальный фильтры.

5. Выборка с сортировкой строк по столбцу Country, а при совпадении городов - по столбцу City (Рисунок 56).

```
SELECT *
FROM Locations
ORDER BY Country, City
```

	LocationID	Country	Region	City	Address	Phone
1	102	Беларусь	Гродненская	Лида	ул.Моховая, 12	NULL
2	103	Беларусь		Минск	ул.Маркса, 24	NULL
3	101	Беларусь	Витебская	Полоцк	ул.Лесная, 6	+375172691376
4	201	Россия	Московская	Королев	ул.Труда, 8	387-23-04
5	202	Россия		Москва	ул.Тверская, 25	900-8876
6	301	Украина		Киев	ул. Крещатик, 14	NULL
7	302	Украина	Львовская	Моршин	ул.Франко, 24	NULL

Рисунок 56. Выборка с сортировкой по двум полям.

6. Выборка из двух таблиц путем их внутреннего соединения по столбцам Locations.LocationID и Clients.Location с сортировкой по имени клиента (Рисунок 57).

```
SELECT Clients.ClientName, Locations.City, Locations.Phone
FROM Locations INNER JOIN Clients
ON Locations.LocationID = Clients.Location
ORDER BY Clients.ClientName
```

	ClientName	City	Phone
1	ГП "Верас"	Москва	900-8876
2	ИП "Темп"	Полоцк	+375172691376
3	ИП "Темпера"	Лида	NULL
4	ОАО "Рога и копыта"	Киев	NULL
5	ОАО "Старт"	Минск	NULL
6	УП "Вера"	Лида	NULL

Рисунок 57. Выборка из двух таблиц путем их внутреннего соединения.

7. Выборка данных из трех таблиц. Выбираются данные только тех заказов, в которых количество заказанных товаров Quantity больше или равно 2 (Рисунок 58).

```

SELECT Clients.ClientName, Locations.City, Locations.Phone,
       Orders.Quantity, Orders.OrderDate
FROM Locations
INNER JOIN Clients ON Locations.LocationID = Clients.Location
INNER JOIN Orders ON Clients.ClientID = Orders.Client
WHERE Orders.Quantity >= 2
ORDER BY Clients.ClientName, Orders.OrderDate DESC

```

	ClientName	City	Phone
1	ГП "Верас"	Москва	900-8876
2	ИП "Темп"	Полоцк	+375172691376
3	ИП "Темпера"	Лида	NULL
4	ОАО "Рога и копыта"	Киев	NULL
5	ОАО "Старт"	Минск	NULL
6	УП "Вера"	Лида	NULL

Рисунок 58. Выборка данных из трех таблиц.

8. Выборка данных с формированием вычисляемого столбца COST (Рисунок 59).

```

SELECT Products.ProductName, Products.Price, Orders.Quantity,
       Products.Measure, Products.Price * Orders.Quantity AS COST
FROM Products INNER JOIN Orders
ON Products.ProductID = Orders.Product
ORDER BY COST

```

	ProductName	Price	Quantity	Measure	COST
1	Монитор 24"	199,00	14,000	шт	2786.0000000
2	Винчестер HDD 1100GB	119,99	25,000	шт	2999.7500000
3	Винчестер HDD 1100GB	119,99	58,000	шт	6959.4200000
4	Монитор 24"	199,00	37,000	шт	7363.0000000
5	Винчестер HDD 4 TB	299,90	25,000	шт	7497.5000000
6	Винчестер HDD 4 TB	299,90	27,000	шт	8097.3000000
7	Компьютер	999,00	10,000	шт	9990.0000000
8	Монитор 24"	199,00	80,000	шт	15920.0000000
9	Монитор 24"	199,00	122,000	шт	24278.0000000

Рисунок 59. Запрос с вычисляемым полем.

9. Подсчет итоговых данных для столбца Количество в таблице Orders ().

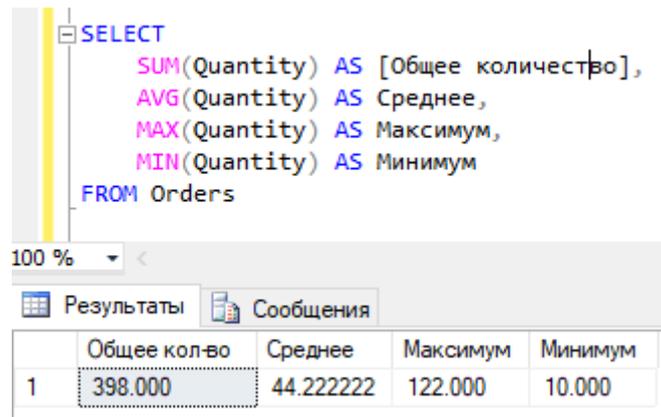


Рисунок 60. Запрос с агрегирующими функциями.

10. Выборка данных из таблицы `Orders` с группировкой по столбцу `ProductID` и подсчетом для каждой группы итоговых данных (Рисунок 61).

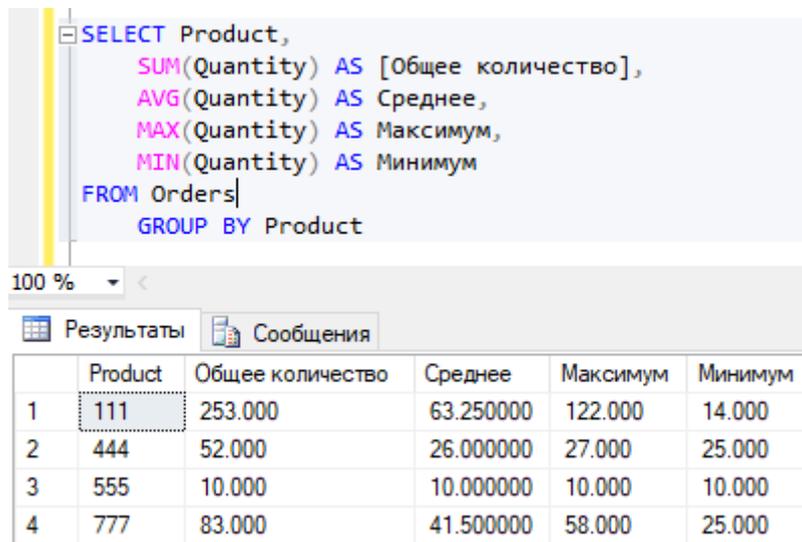


Рисунок 61. Выборка данных с их группировкой и подсчетом для каждой группы итоговых данных.

11. Выборка данных из таблицы `Orders` с группировкой по столбцу `ProductID` и отбором групп, удовлетворяющих условию: общее количество товаров больше 80 (Рисунок 62).

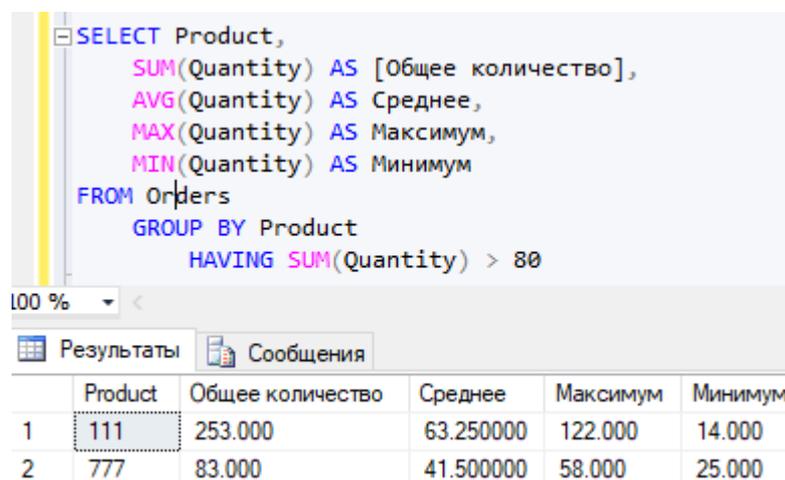
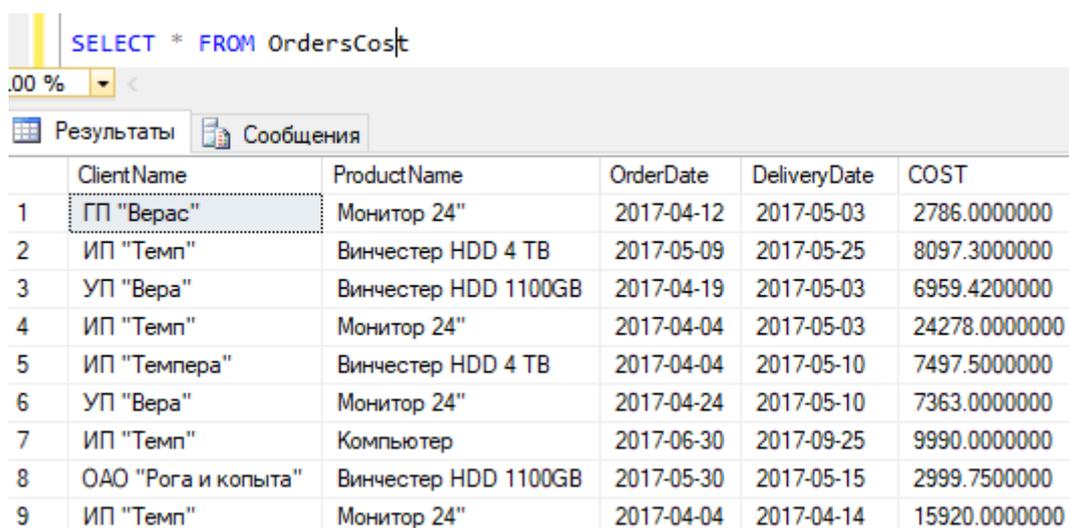


Рисунок 62. Запрос с отбором групп, удовлетворяющих условию.

12. Выборка данных из представления OrdersCost (Рисунок 63).



```
SELECT * FROM OrdersCost
```

	ClientName	ProductName	OrderDate	DeliveryDate	COST
1	ГП "Верас"	Монитор 24"	2017-04-12	2017-05-03	2786.0000000
2	ИП "Темп"	Винчестер HDD 4 TB	2017-05-09	2017-05-25	8097.3000000
3	УП "Вера"	Винчестер HDD 1100GB	2017-04-19	2017-05-03	6959.4200000
4	ИП "Темп"	Монитор 24"	2017-04-04	2017-05-03	24278.0000000
5	ИП "Темпера"	Винчестер HDD 4 TB	2017-04-04	2017-05-10	7497.5000000
6	УП "Вера"	Монитор 24"	2017-04-24	2017-05-10	7363.0000000
7	ИП "Темп"	Компьютер	2017-06-30	2017-09-25	9990.0000000
8	ОАО "Рога и копыта"	Винчестер HDD 1100GB	2017-05-30	2017-05-15	2999.7500000
9	ИП "Темп"	Монитор 24"	2017-04-04	2017-04-14	15920.0000000

Рисунок 63. Выборка данных из представления.

Примечание. При выборке данных из представления можно использовать все предложения команды SELECT.

13. Создание новой таблицы путем отбора данных из другой таблицы. Следующая команда создаст копию таблицы Orders с данными. При создании новой таблицы таким способом копируется только структура таблицы и данные. Ограничения нужно задавать заново.

```
SELECT * INTO Orders1 FROM Orders
```

Следующая команда создаст копию таблицы Products.

```
SELECT * INTO Products1 FROM Products
```

Примечание. При создании новой таблицы можно использовать все предложения команды SELECT.

Задание 2. Выборка системных данных

1. Список учетных записей, которым разрешен доступ к серверу

```
USE master -- переключаемся на системную базу данных master
SELECT name, dbname, password, language FROM syslogins
```

```
USE TRADE_XXX -- переключаемся обратно на базу данных
-- TRADE_XXX
```

2. Список учетных записей, включенных в фиксированные роли сервера

```
EXEC sp_helpsrvrolemember
```

3. Список пользователей базы данных TRADE_XXX

```
EXEC sp_helpuser
```

4. Список ролей (как фиксированных, так и пользовательских) базы данных TRADE_XXX

```
EXEC sp_helprole
```

5. Членство ролей и пользователей в ролях базы данных TRADE_XXX

```
EXEC sp_helprolemember
```

Задание 3. Обновление данных в таблицах и представлениях

1. В таблице Clients заменить все значения NULL в столбце Location на значение 103 с помощью следующей команды:

```
USE TRADE__BD
```

```
SELECT * FROM Clients
```

```
-----
```

```
UPDATE Clients
```

```
    SET Location = 103
```

```
    WHERE Location IS NULL
```

```
-----
```

```
SELECT * FROM Clients
```

Задание 4. Удаление данных из таблиц и представлений

1. В таблице Orders1 удалить все строки, в которых значение поля DeliveryDate не относится к текущему году, следующим образом:

```
SET DATEFORMAT dmy -- задаем привычный формат даты:
```

```
    -- день.месяц.год
```

```
DELETE FROM Orders1
```

```
    WHERE DeliveryDate < '01.01.2017'
```

```
GO
```

Задание 5. Изменение структуры таблицы

Примеры использования команды изменения структуры таблицы:

1. Добавление нового поля Weight в таблицу Products1.

```
ALTER TABLE Products ADD Weight INT NULL
```

```
GO
```

2. Изменение типа данных поля Weight в таблице Products с INT на FLOAT.

```
ALTER TABLE Products
```

```
ALTER COLUMN Weight FLOAT NULL
```

```
GO
```

3. Установка проверочного ограничения для поля `Weight` в таблице `Products`.

```
ALTER TABLE Products
```

```
ADD CONSTRAINT CK_Products_Weight
```

```
CHECK (Weight BETWEEN 10 AND 100)
```

```
GO
```

4. Удаление поля `Weight` из таблицы `Products` (сначала удаляется проверочное ограничение).

```
ALTER TABLE Products
```

```
DROP CONSTRAINT CK_Products_Weight
```

```
ALTER TABLE Products DROP COLUMN Weight
```

```
GO
```

5. Удаление ограничения внешнего ключа `FK_Products_Currency` из таблицы `Products`.

```
ALTER TABLE dbo.Products
```

```
DROP CONSTRAINT FK_Products_Currency
```

6. Добавление ограничения внешнего ключа `FK_Products_Currency` в таблицу `Products`.

```
ALTER TABLE Products
```

```
ADD CONSTRAINT FK_Products_Currency
```

```
FOREIGN KEY (Currency) REFERENCES Currency
```

```
ON UPDATE CASCADE
```

7. Добавим ограничение по первичному ключу для таблицы `Products1`.

```
ALTER TABLE Products1
```

```
ADD CONSTRAINT PK_Products1
```

```
PRIMARY KEY(ProductID)
```

8. Добавление ограничения внешнего ключа `FK_Products1_Order1` в таблицу `Orders1`.

```
ALTER TABLE Orders1
```

```
ADD CONSTRAINT FK_Products1_Orders1
```

```
FOREIGN KEY (Product) REFERENCES Products1
```

```
ON UPDATE CASCADE
```

Задание 6. Удаление таблицы из базы данных

Прежде чем удалить таблицу из базы данных, необходимо разорвать все связи этой таблицы с ее дочерними таблицами, если такие имеются. В противном случае удалить таблицу будет невозможно.

Например, требуется удалить таблицу `Products`. Сначала необходимо провести анализ, есть ли таблицы, для которых таблица `Products` является родительской. Для этого можно, например, использовать системную хранимую процедуру `sp_fkeys`, позволяющую получить информации о связях между таблицами посредством первичных и внешних ключей.

1. Получить информацию о связях родительской таблицы `Products` с ее дочерними таблицами

```
EXEC sp_fkeys 'Products1'
```

У таблицы `Products` имеется одна дочерняя таблица - таблица `Orders`. Та же информация подтверждается, если проанализировать с помощью процедуры `sp_fkeys`, какие родительские таблицы имеются у таблицы `Orders`.

2. Получить информацию о связях дочерней таблицы `Orders` с ее родительскими таблицами

```
EXEC sp_fkeys @fktable_name = 'Orders1'
```

3. После этого становится ясно, что для удаления из базы данных таблицы `Products`, нужно выполнить следующие две команды:

Удаление из базы данных таблицы `Products1` ограничения внешнего ключа `PK_Orders_Products`.

-- Первая команда

```
ALTER TABLE Orders1
```

```
    DROP CONSTRAINT FK_Products1_Orders1
```

-- Вторая команда

```
DROP TABLE Products1
```

Замечание. Сначала была разорвана связь между таблицами `Products1` и `Orders1` путем удаления в таблице `Orders1` ограничения внешнего ключа `FK_Products1_Orders1`. Только после этого была удалена таблица `Products1`.

Задания для самостоятельного выполнения

Предварительно выполните следующие действия.

- Закройте (без сохранения данных) текущее Окно запросов (*Query*) и удалите базу данных `TRADE_XXX`.

- Загрузите сценарий из файла D:\FIO\TRADE_XXX\CreateDB_Insert_Data.sql и создайте базу данных TRADE_XXX заново.
- Обновите данные в окне Обозреватель объектов (*Object Explorer*) и сделайте базу данных TRADE_XXX видимой.
- Закройте окно со сценарием CreateDB_Insert_Data.sql и откройте новое пустое Окно запросов (*Query*), предназначенное для формирования сценария выполнения задания к лабораторной работе.

Все задания выполняйте, используя операторы языка SQL.

Задание 7. Создание новых таблиц

1. Создайте новую таблицу **Suppliers** (Поставщики) (Рисунок 64).

	Имя столбца	Тип данных	Разрешить значения NULL
🔑	SupplierID	int	<input type="checkbox"/>
	SupplierName	varchar(40)	<input type="checkbox"/>
	PaymentTerms	varchar(30)	<input checked="" type="checkbox"/>
	Location	int	<input checked="" type="checkbox"/>
	Note	varchar(MAX)	<input checked="" type="checkbox"/>

Рисунок 64. Таблица **Suppliers** (Поставщики).

При создании таблицы обратите внимание на ее связи с другими таблицами (Рисунок 65). Связи между таблицами *Orders* и *Suppliers* и между таблицами *Locations* и *Suppliers* являются неидентифицирующими. Для создания связи между таблицами *Orders* и *Suppliers* необходимо в таблицу *Orders* добавить новое поле *Supplier*.

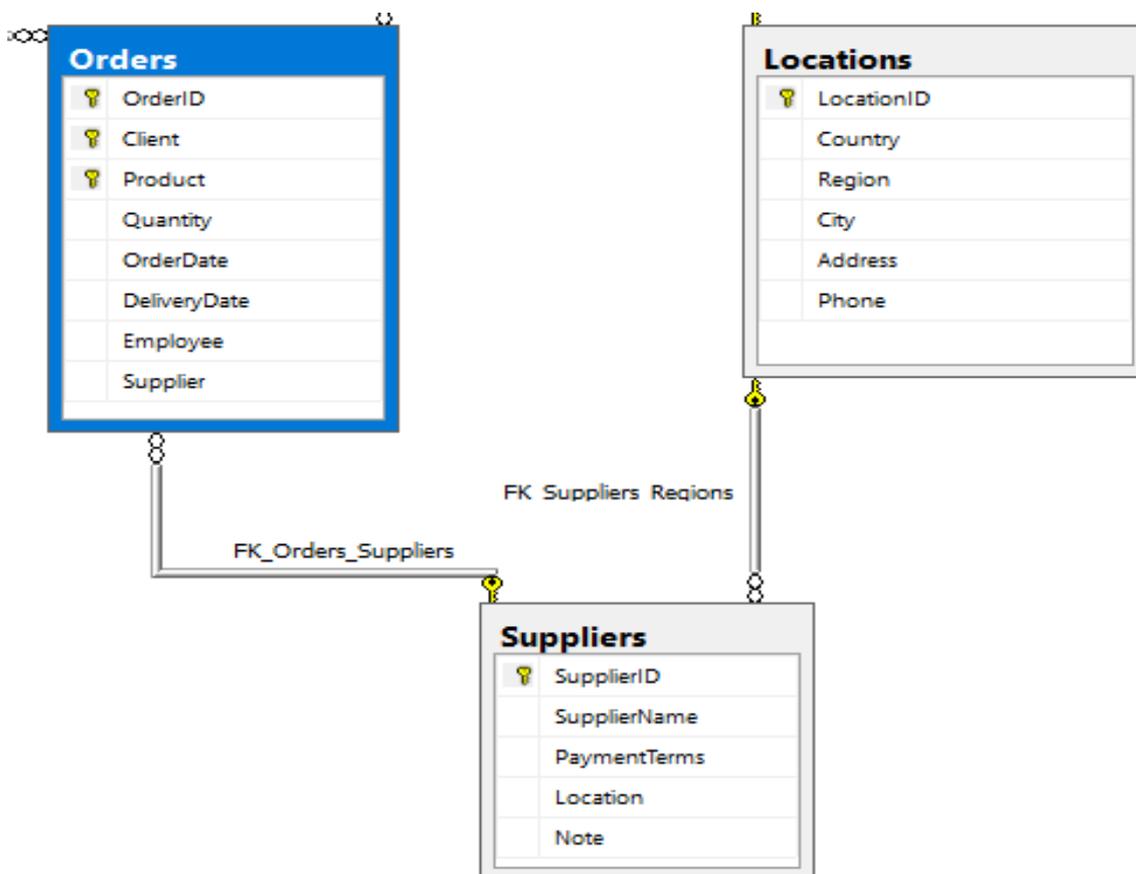


Рисунок 65. Связи таблицы Suppliers с таблицами Orders и Locations.

- Заполните таблицу Suppliers данными (Рисунок 68).

SupplierID	SupplierName	PaymentTerms	Location	Note
123	ЗАО Магистраль	Предоплата	101	NULL
126	ЗАО Магистрал	Предоплата	204	NULL
130	ЗАО Магистр	Предоплата	202	NULL
500	ОАО "Привет"	По факту отгрузки	303	Постоянный поставщик
567	СП "Полихим"	По факту отгрузки	203	Постоянный поставщик
NULL	NULL	NULL	NULL	NULL

Рисунок 66. Данные таблицы Suppliers.

- Заполните поле Supplier таблицы Orders.
2. Создайте новую таблицу **Employees** (Сотрудники) (Рисунок 64).

Имя столбца	Тип данных	Разрешить значения NULL
EmployeeID	int	<input type="checkbox"/>
Name	nvarchar(30)	<input checked="" type="checkbox"/>
SurName	nvarchar(30)	<input checked="" type="checkbox"/>
Title	nvarchar(25)	<input checked="" type="checkbox"/>
Subordination	int	<input checked="" type="checkbox"/>
Birthday	date	<input checked="" type="checkbox"/>

Рисунок 67. Таблица **Employees** (Сотрудники).

При создании таблицы обратите внимание на ее связи с другими таблицами (Рисунок 65) и с самой собой. Для создания связи между таблицами **Orders** и **Employees** необходимо в таблицу **Orders** добавить новое поле **Employee**.

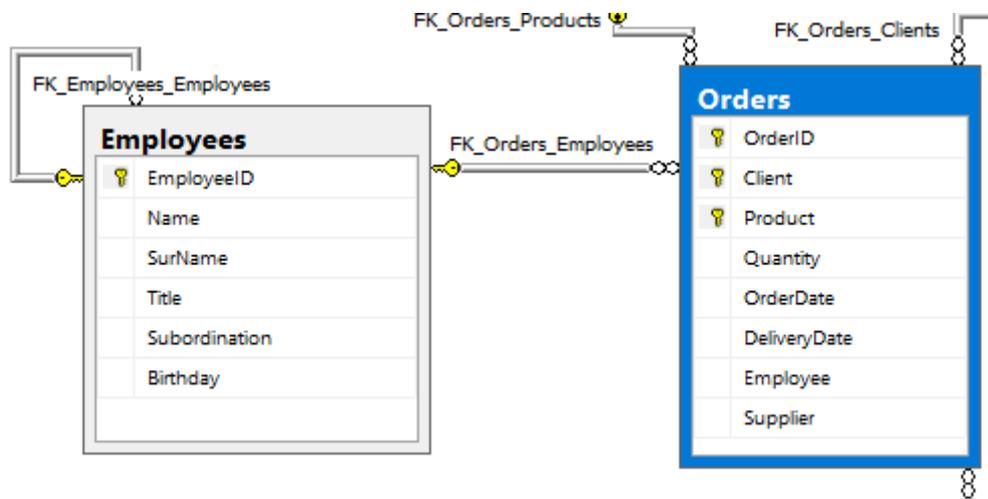


Рисунок 68. Связи таблицы **Employees**.

- Заполните таблицу **Employees** данными (Рисунок 69).

EmployeeID	Name	SurName	Title	Subordination	Birthday
1000	Никита	Котов	Директор	NULL	1975-01-31
1001	Илья	Мудрый	Начальник отдела маркетинга	1000	1970-08-06
1002	Иван	Прохоров	Главный бухгалтер	1000	1981-01-18
1003	Алиса	Кошкина	Секретарь	1000	1990-05-09
1004	Мария	Смехова	Маркетолог	1001	1978-03-27
1005	Алексей	Бусел	Экономист	1002	1968-07-15
1006	Полина	Русакович	Экономист	1002	1985-10-11

Рисунок 69. Данные таблицы **Employees**.

- Заполните поле **Employee** таблицы **Orders**.

После выполнения задания 7 получится диаграмма данных, представленная на Рисунок 70.

3. ***Создайте таблицу **Supplier-Product**, которая реализует на логическом уровне связь типа многие-ко-многим между таблицами **Suppliers** и **Products** (см. лабораторную работу № 1). Самостоятельно заполните ее данными.

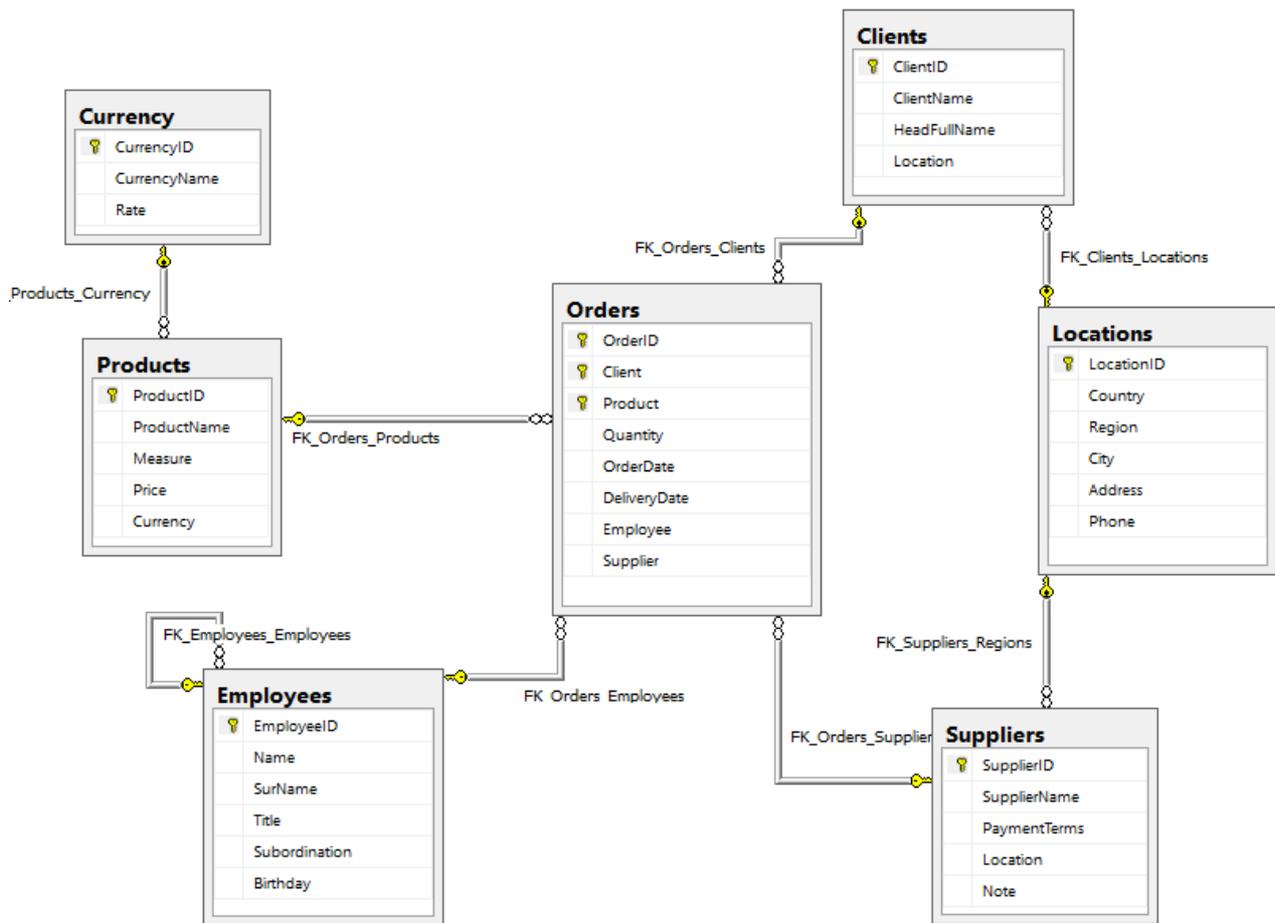


Рисунок 70. Диаграмма данных БД TRADEXXX.

Задание 8. Выборка данных из таблиц и представлений

1. Из таблицы `Clients` выберите все строки, для которых значение поля `ClientName` начинается префиксом «ИП» и значение поля `Location` лежит в диапазоне от 100 до 110 или неизвестно.
2. Из таблицы `Locations` выберите все строки, относящиеся к России или к Беларуси, но не связанные с городом Минском.
3. Из таблицы `Products` выберите все строки, связанные с валютами «Доллары США» или «Евро», для которых значение цены товара лежит в диапазоне от 100 до 300.
4. В таблице `Orders` найдите все те строки, для которых значение поля `Quantity` превышает 25. Однако на экран вместо полей `Client` и `Product` выведите поля `ClientName` из таблицы `Clients` и `ProductName` из таблицы `Products` соответственно, которые заменят малоинформативные коды содержательными наименованиями.

5. В таблице `Orders` найдите все строки, относящиеся к клиентам из Беларуси, а из них, в свою очередь, выберите те строки, для которых с момента заказа прошло не более 60 дней. Кроме того, в полученном наборе строк замените столбец `Product` столбцом `ProductName` из таблицы `Products`. Результирующий набор строк отсортируйте по наименованию товаров, далее – по полю `OrderDate` и далее – по полю `Quantity` в порядке убывания.

6. Измените представление `OrdersCost`, добавив в него поле `Quantity` из таблицы `Orders`. Из представления `OrdersCost` выберите все строки, в которых значение поля `ProductName` содержит в себе подстроку «тер» или «тор» и значение поля `Quantity` больше или равно 30.

7. Подсчитайте стоимость заказов в белорусских рублях: произведение трех величин: количество товаров в заказе (поле `Quantity` таблицы `Orders`), цена товара (поле `Price` таблицы `Products`), значение курса белорусского рубля (поле `Rate` таблицы `Currency`). Выведите наименование товара (поле `ProductName` таблицы `Products`) и подсчитанную стоимость. Отсортируйте по стоимости.

8. Измените запрос предыдущего задания так, чтобы выводилась общая сумма стоимости заказов по каждому товару. Для этого сгруппируйте данные по полю `ProductName` таблицы `Products`. Ограничьте количество знаков после запятой в поле стоимости заказов двумя цифрами.

9. Измените запрос предыдущего задания так, чтобы выводились наименования только тех товаров, суммарная стоимость которых больше заданной величины. Используйте для этого предложение `HAVING` оператора `SELECT`.

10. С помощью команды `SELECT ... INTO` создайте копию таблицы `Clients`, обязательно включив в новую таблицу поле `ClientName`. Дайте имя новой таблице `Clients1`. Измените значения в `ClientName` таблицы `Clients1` так, чтобы часть значений наименований клиентов совпадала со значениями исходной таблицы `Clients`. Выполните над таблицами `Clients` и `Clients1` операции объединения `UNION`, пересечения `INTERSECT`, разности `EXCEPT`. Проанализируйте, как влияют ключевые слова `DISINCT` и `ALL` на результат выполнения оператора `UNION`.

11. Измените таблицу `Clients` так, чтобы у нескольких клиентов в поле `Location` было значение `NULL`, и чтобы не все регионы имели клиентов, то есть чтобы в поле внешнего ключа `Location` отсутствовали некоторые значения из поля `LocationID` таблицы `Location`. Выполните операции внутреннего `INNER JOIN`, левого `LEFT JOIN`, правого `RIGHT JOIN`, полного `FULL JOIN`, перекрестного `CROSS JOIN` соединений таблиц `Clients` и `Locations`.

12. *** Выберите из таблицы Employees имена и фамилии всех руководителей (EmployeeID, Employee.Name, Employee.Surname). Используйте операцию соединения таблицы с самой собой (самосоединение) (Рисунок 71). Обратите внимание на псевдонимы (alias) E1 и E2 таблицы Employees, которые используются в запросе.

```

SELECT E1.EmployeeID, E1.Subordination, E2.EmployeeID, E2.Subordination
FROM Employees E1 INNER JOIN Employees E2
ON E1.EmployeeID = E2.Subordination

```

	EmployeeID	Subordination	EmployeeID	Subordination
1	1000	NULL	1001	1000
2	1000	NULL	1002	1000
3	1000	NULL	1003	1000
4	1001	1000	1004	1001
5	1002	1000	1005	1002
6	1002	1000	1006	1002

Рисунок 71. Операция самосоединения таблицы Employees.

13. *** Подсчитайте, сколько у каждого руководителя подчиненных на основе данных таблицы Employees. Примените групповую операцию.

Задание 9. Обновление данных в таблицах

1. В таблице Clients замените имя клиента «ГП "Верас"» на «ГП "Верас-М"» и фамилию ее руководителя на «Сурков Владимир Петрович».

2. В таблице Products во всех записях, где поле Price в белорусских рублях, замените код валюты на «EUR», а цену уменьшите в 2,3 раз.

3. В таблице Orders обновите поле DeliveryDate следующим образом: если дата заказа раньше 15 апреля текущего года, то срок поставки будет равен дате заказа, увеличенной на 10 дней, однако, после 15 апреля срок поставки должен быть больше даты заказа только на неделю.

4. Добавьте в таблицу Locations новый регион, расположенный в Литовской республике. Дайте ему номер 401. Остальные значения выберите самостоятельно.

Задание 10. Удаление данных из таблиц

1. Добавьте приведенные ниже данные в соответствующие таблицы базы данных с помощью команды INSERT.

Таблица Currency.

CurrencyID	CurrencyName	Rate
GRV	Украинские гривны	0,006

Таблица Products

ProductID	ProductName	Measure	Price	Currency
666	ПК-клавиатура	штука	780	GRV
777	Разъем USB	штука	85	GRV
888	Принтер EPSON	штука	11650	GRV

Таблица Orders

Clients	Product	Quantity	OrderDate	DeliveryDate
4	666	12	по умолчанию	по умолчанию
5	777	15	по умолчанию	по умолчанию
3	888	5	по умолчанию	по умолчанию
5	666	8	по умолчанию	по умолчанию

2. Удалите из таблицы Currency строку с кодом валюты GRV (Украинские гривны).

Замечание. Предварительно проанализируйте свойства связей между указанными выше таблицами, в частности, разрешено или запрещено каскадное (CASCADE) удаление данных в таблицах Products и Orders. Свойства связей между таблицами задаются в команде CREATE TABLE или в команде ALTER TABLE.

Задание 11. Изменение структуры таблиц и представлений

1. Внесите такие изменения в таблицу Products, чтобы для выполнения предыдущего задания, связанного с удалением данных из таблиц, достаточно было использовать только одну команду удаления данных из таблицы Currency.

2. Внесите изменения в нужные таблицы так, чтобы для выполнения задания, связанного с удалением данных из таблиц, требовалось обязательное удаление данные из всех трех таблиц (Orders, Products, Currency).

3. Создайте копию представления OrdersCost с именем OrdersCost1.

Задание 12. Удаления таблиц из базы данных

1. Удалите из базы данных таблицу Orders1.
2. Изучите самостоятельно команду удаления представления (VIEW) из базы данных и удалите существующее представление OrdersCost1.

Сохраните текущий сценарий в файле
D:\FIO\TRADE_XXX\DataManipulation.sql.

Затем удалите базу данных TRADE_XXX. Для этого в ее контекстном меню выберите команду Удалить (*Delete*) и затем в появившемся окне Удаление объекта (*Delete Object*) установите флажок Закрывать существующие соединения (*Close Existing Connections*).