|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Фундаментальные науки»

КАФЕДРА «Математическое моделирование»

ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент Лосев Владислав Александрович

*фамилия, имя, отчество*

Группа ФН12-21М

Тип практики научно-исследовательская работа

Название предприятия МГТУ имени Н. Э. Баумана

Студент

Руководитель практики

Лосев В. А.

*подпись, дата фамилия, и.о.*

Вишняков И.Э.

*подпись, дата фамилия, и.о.*

Оценка:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2024 г.

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

Кафедра « Математическое Моделирование » (\_ФН12-21М\_)

З А Д А Н И Е

**на прохождение производственной практики**

на предприятии МГТУ имени Н. Э. Баумана

Студент Лосев Владислав Александрович / Лосев В. А. / ФН12-21М

(фамилия, имя, отчество; инициалы; индекс группы)

Во время прохождения производственной практики студент должен:

1. Построить модель «сущность-связь» для выбранной предметной области.
2. Преобразовать модель «сущность-связь» в реляционную модель.
3. Реализовать БД средствами СУБД SQL Server.

Дата выдачи задания « » 20 г.

Руководитель практики от кафедры\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

**Студент**  **/ \_\_\_\_\_\_\_\_\_\_\_**

(подпись, дата) (Фамилия И.О.)

# Содержание

[Введение](#_bookmark0) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

1. [Проектирование ER-модели](#_bookmark1) . . . . . . . . . . . . . . . . . . .5
2. [Преобразование](#_bookmark3) ER-модели в реляционную . . . . . . . . . . . . . . 7
3. Реализация базы данных средствами СУБД SQL Server . . . . 11

Заключение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .19

1. Приложение А . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .20

# Введение

База данных и система управления базой данных являются неотъемлемой частью информационных систем предприятия. Процесс проектирования базы данных представляет собой последовательность переходов от словесного описания информационной структуры предметной области к формализированному описанию объектов предметной области в терминах некоторой модели. В общем случае выделяют следующие этапы проектирования: анализ и описание предметной области информационной системы, концептуальное моделирование, построение логической модели базы данных, физическое проектирование базы данных. Целью данной научно-исследовательской работы является разработка модели «сущность-связь» и её дальнейшая реализация в SQL Server.

Для достижения поставленной цели необходимо решить следующие задачи:

* выбрать предметную область, соответствующую 4-5 сущностям;
* сформировать требования к предметной области;
* создать модель «сущность-связь» для предметной области с обоснованием выбора кардинальных чисел связи;
* преобразовать модель «сущность-связь» в реляционную модель согласно процедуре преобразования;
* обосновать выбор типов данных, ключей, правил обеспечения ограничений минимальной кардинальности;
* реализовать спроектированную базу данных с помощью средств СУБД SQL Server.

1. **Проектирование ER-модели**

Для реализации модели была выбрана работа управляющей компании (управление капиталом клиентов). Клиентами управляющей компании являются юридические лица, вносящие компании деньги для управления в соответствии с условиями, прописанными в договоре. С помощью полученных данных управляющая компания может:

1. Получить отчет об эффективности управления.
2. Получить информацию о совершенных сделках
3. Получить информацию о сумме внесенных средств и информацию для отчетов по управлению.

На основе описанной в предыдущем пункте предыдущем пункте предметной области создана модель «сущность-связь». В созданную модель были включены пять сущностей:

* Клиент (юридическое лицо) – сущность клиента с идентификатором “ИНН” и атрибутами: правовая форма, юридический адрес, почтовый адрес.
* Договор – сущность с идентификатором “номер договора” и атрибутами: дата заключения, срок, сумма (перечисляемая клиентом управляющей компании), инвестиционная декларация.
* Сделка – сущность с идентификатором “номер сделки” и атрибутами: направление сделки, дата заключения сделки, сумма сделки, количество купленных ценных бумаг, для облигаций купонный период и процент выплат, цена за штуку.
* Выпуск (серия) – сущность типа ценных бумаг с идентификатором ISIN и с атрибутами: торговая площадка, эмитент, текущая цена бумаги, тип ценной бумаги, количество выпущенных бумаг.

Между описанными сущностями были построены связи, согласующиеся с особенностями предметной области.

1. Так как каждый договор связан непосредственно с одним конкретным клиентом при этом каждый клиент может заключать много договоров, а также может быть ситуация, когда договор закончился и новый пока не заключен, в этом случае может быть 0 договоров. Тогда связь между сущностями Клиент и Договор является связью типа «один-ко-многим», а минимальные кардинальные числа равны 1 и 0 соответственно.
2. Так как каждая на основании каждого договора может быть заключено множество сделок, то связь между сущностями “Договор” и “Сделка” является связью типа «один-ко-многим», а минимальные кардинальные числа равны 1 и 0 соответственно.
3. Ценные бумаги с одним ISIN могут несколько раз фигурировать в сделках. В том случае, если их, например, сначала продали, потом купили. Без покупки ценных бумаг сделка не существует, в результате заключения сделки покупается серия акций с одним ISIN, то связь между сущностями Сделка и Выпуск (серия) ценных бумаг является связью типа «многие к одному», а минимальные кардинальные числа равны 0 и 1 соответственно.

Визуализируем представленную модель сущность-связь (рисунок 1):

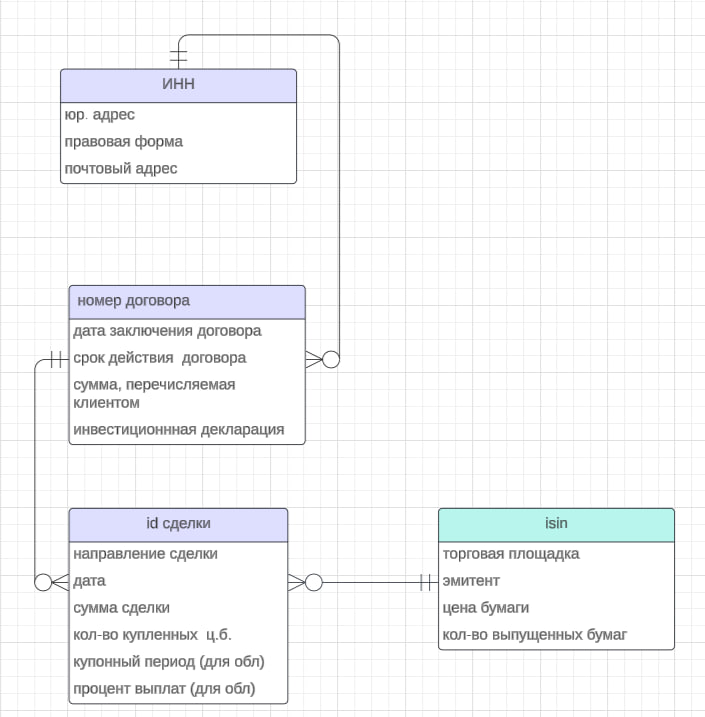


Рисунок 1 - Модель «сущность-связь»

# Преобразование ER-модели в реляционную

На основании модели «сущность-связь» была получена реляционная модель. А также реализованы таблицы для каждой сущности.

В таблице 1 представлены типы данных и их значения по умолчанию для сущности Клиент.

Таблица 1 − CLIENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Type | Key | NULL Status | Remarks |
| ИНН | char(10) | Alternate | NOT NULL | Unique (AK1.1) |
| Юр. адрес | Nvarchar (100) | No | NOT NULL |  |
| Правовая форма | tinyint | No | NULL | 1-ООО  2-ЗАО  3-ОАО |
| Email | Nvarchar (100) | No | NOT NULL |  |
| ID клиента | int | Primary | NOT NULL | Surrogate Key |

В таблице 2 представлены типы данных и их значения по умолчанию для сущности Договор

Таблица 2 − Договор

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Type | Key | NULL Status | Remarks |
| Номер договора | int | Primary | NOT NULL |  |
| Дата заключения | datetime | No | NOT NULL |  |
| Срок (в днях) | int | No | NOT NULL | DEFAULT value = 365 |
| Сумма, перечисляемая клиентом | float | No | NOT NULL |  |
| Инвестиционная декларация (условия) | varchar (100) | No | NOT NULL |  |
| ID клиента | int | Foreign | NOT NULL |  |

В таблице 3 представлены типы данных и их значения по умолчанию для сущности Сделка.

Таблица 3 − Сделка

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Type | Key | NULL Status | Remarks |
| Id сделки | int | Primary | NOT NULL |  |
| Направление сделки | smallint | No | NOT NULL | 1-купили  -1-продали |
| Дата заключения | datetime | No | NULL |  |
| Сумма сделки | float | No | NOT NULL |  |
| Количество цб | int | No | NOT NULL |  |
| Цена за штуку | float | No | NOT NULL |  |
| Купонный период | int | No | NULL | Только для облигаций |
| Процент выплат | float | No | NULL | Только для облигаций |
| ISIN | char(12) | Foreign key | NOT NULL |  |
| Номер договора | int | Foreign Key | NOT NULL |  |

В таблице 4 представлены типы данных и их значения по умолчанию для сущности Выпуск ценных бумаг.

Таблица 4 − Выпуск (серия) ценных бумаг

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Type | Key | NULL Status | Remarks |
| ISIN | char(12) | Primary | NOT NULL |  |
| Торговая площадка | varchar (80) | No | NULL |  |
| Эмитент | varchar (80) | No | NOT NULL |  |
| Текущая цена | float | No | NOT NULL |  |
| Объем выпуска | int | No | NULL |  |

Определим типы кардинальности для каждого отношения (таблица 6)

Таблица 6 − Определение кардинальностей для каждого отношения

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship | | Cardinality | |
| parent | child | max | min |
| Клиент | Договор | 1:N | M-O |
| Договор | Сделка | 1:N | M-O |
| Серия ценных бумаг | Сделка | 1:N | M-O |

Правила ограничения минимальной кардинальности приведены в следующих таблицах (таблица 7, таблица 8, таблица 9).

Таблица 7 - Клиент к договору

|  |  |  |
| --- | --- | --- |
| Операции | Действия для Клиента (родитель) | Действия для Договора (дочерний) |
| Вставка | − | Подбор новой родительской записи. |
| Изменение первичного или внешнего ключа | Запрет (суррогатный ключ.) | Запрет |
| Удаление | Каскадное удаление ребенка. | − |

Таблица 8 - Договор к Сделке

|  |  |  |
| --- | --- | --- |
| Действия | Действия для Договора (родитель) | Действия для Сделки (дочерний) |
| Вставка | − | Подбор новой родительской записи |
| Изменение первичного или внешнего ключа | Запрет | Запрет |
| Удаление | Каскадное удаление ребенка | − |

Таблица 9 - Серия ц.б. к Сделке

|  |  |  |
| --- | --- | --- |
| Действия | Действия для Серии ц.б. (родитель) | Действия для Сделки (дочерний) |
| Вставка | − | Подбор новой родительской записи. |
| Изменение первичного или внешнего ключа | Запрет | Запрет |
| Удаление | Запрет | − |

Полученная реляционная модель представлена на рисунке 2:

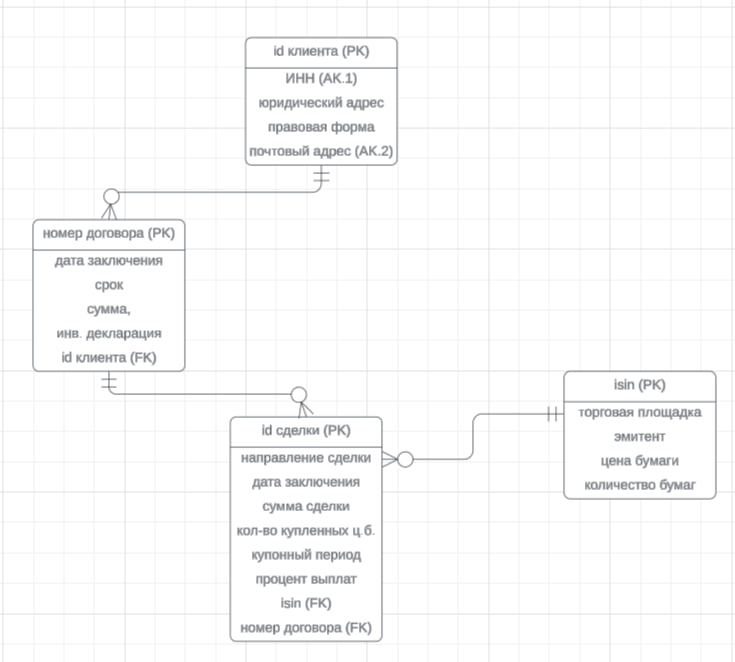


Рисунок 2 - Реляционная модель базы данных

* 1. **Реализация базы данных средствами СУБД SQL Server**

Реализация описанной в предыдущих разделах базы данных была выполнена с использованием языка запросов Transact-SQL.

В рассматриваемой БД были созданы следующие запросы DML для:

* выборки записей (команда SELECT);
* добавления новых записей (команда INSERT), как с помощью непосредственного указания значений, так и с помощью команды SELECT;
* модификации записей (команда UPDATE);
* удаления записей (команда DELETE).

Были проиллюстрированы следующие возможности языка:

* удаление повторяющихся записей (DISTINCT);
* выбор, упорядочивание и именование полей (создание псевдонимов для полей и таблиц);
* соединение таблиц (INNER JOIN LEFT JOIN);
* условия выбора записей (в том числе, условия LIKE, BETWEEN, IN);
* сортировка записей (ORDER BY);
* группировка записей (GROUP BY + HAVING, использование функций агрегирования - COUNT, AVG, SUM, MIN, MAX);
* объединение результатов нескольких запросов;
* вложенные запросы.

Листинг кода представлен в приложении. Для демонстрации выполним следующие задачи:

1. Выберем ценные бумаги, которые использовались в сделках, исключив повторы (рисунок 3)

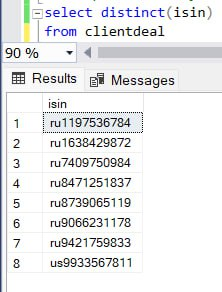


Рисунок 3 - Ценные бумаги, использованные в сделках

1. Выберем самые крупные контракты (рисунок 4)

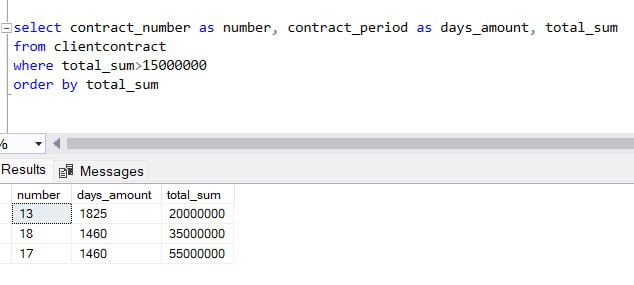


Рисунок 4 - Самые крупные контракты

1. Выделим основную информацию о договорах (рисунок 5)

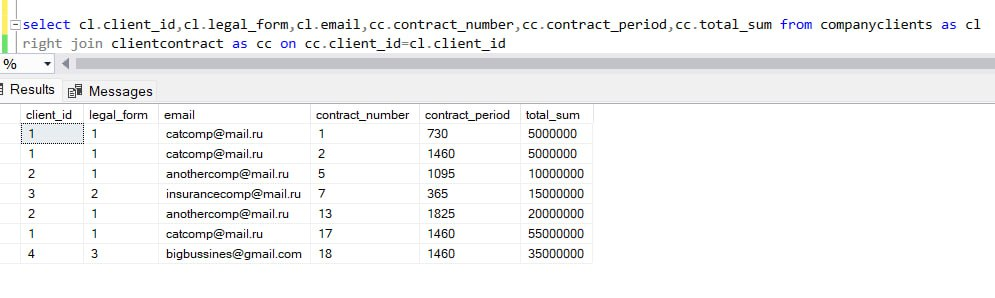


Рисунок 5 - Основная информация о договорах

1. Информация о всех ценных бумагах и контрактах (рисунок 6)

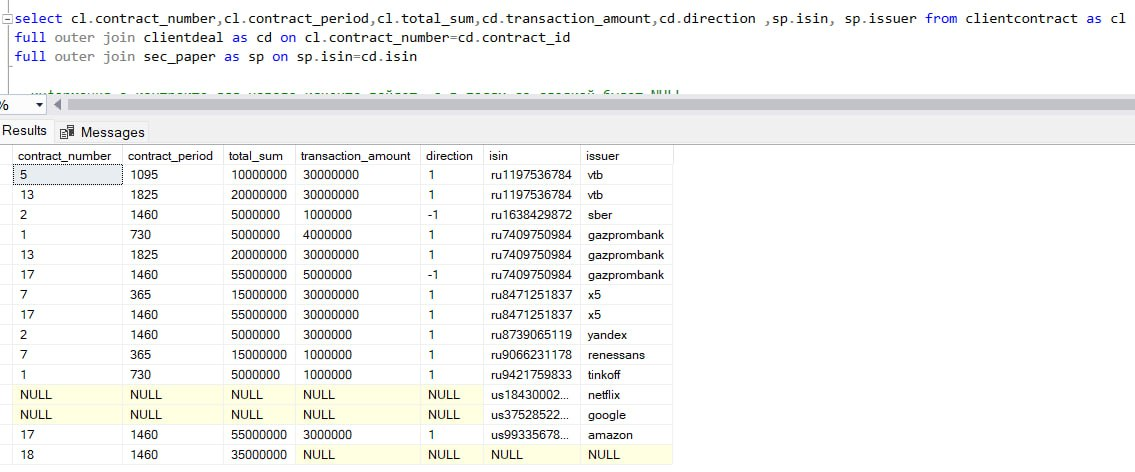


Рисунок 6 - Информация о ценных бумагах и контрактах

1. Все договоры и сделки, заключенные в соответствие с ними (рисунок 7)

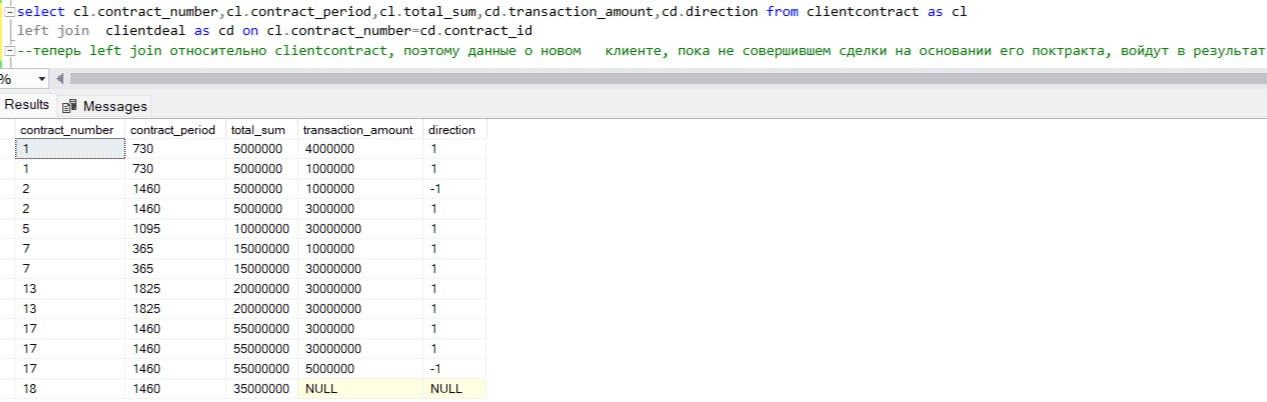


Рисунок 7 - Договоры и связанные с ними сделки

1. Недавно заключенные контракты (рисунок 8)



Рисунок 8 - Контракты, заключенные в текущем году

1. Выведем американские ценные бумаги, участвующие в сделках (рисунок 9)

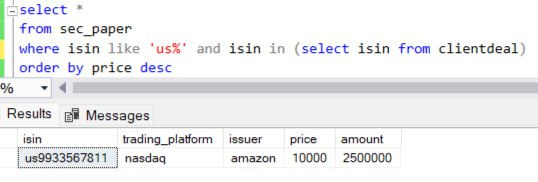


Рисунок 9 - Американские ц.б. в сделках

1. Выведем процент от всех полученных средств, вложенных в акции (рисунок 10)

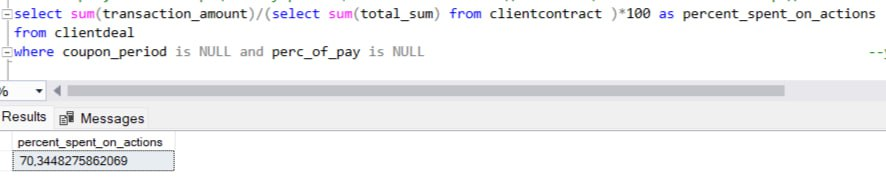


Рисунок 10 - Процент средств, вложенных в акции

1. Найдем все договоры, на основании которых еще не были заключены сделки (рисунок 11)

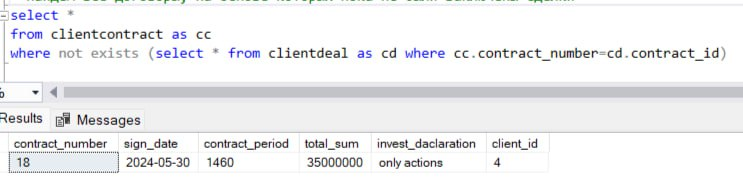


Рисунок 11 - Новые договоры

1. Найдем суммарное количество, общую сумму и среднюю вложенную сумму в сделку для каждого договора (рисунок 12)

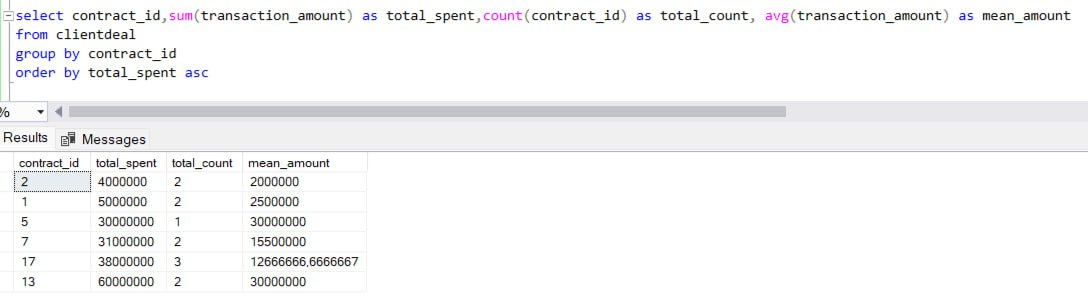


Рисунок 12 - Общая сумма и количество сделок, заключенных в соответствие с контрактом

1. Найдем самые ранние поздние даты заключения сделки, в соответствие с договором (рисунок 13)

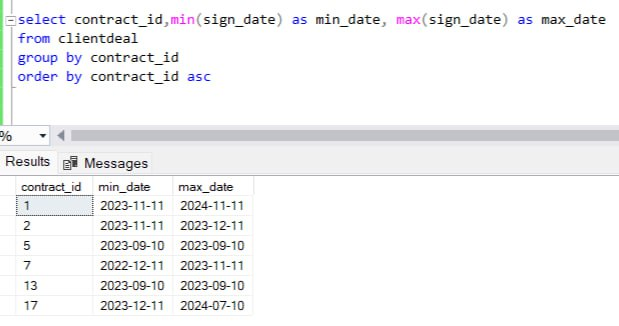


Рисунок 13 - Самые ранние и поздние сделки, для каждого договора

1. Клиенты, заключившие более одного договора (рисунок 14)

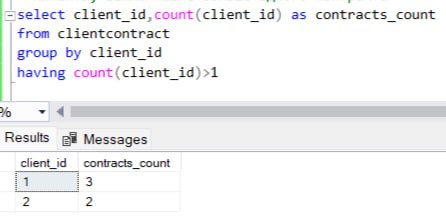


Рисунок 14 - Постоянные клиенты

1. Найдем количество сделок, совершенных для каждого клиента и их суммарную стоимость (рисунок 15)

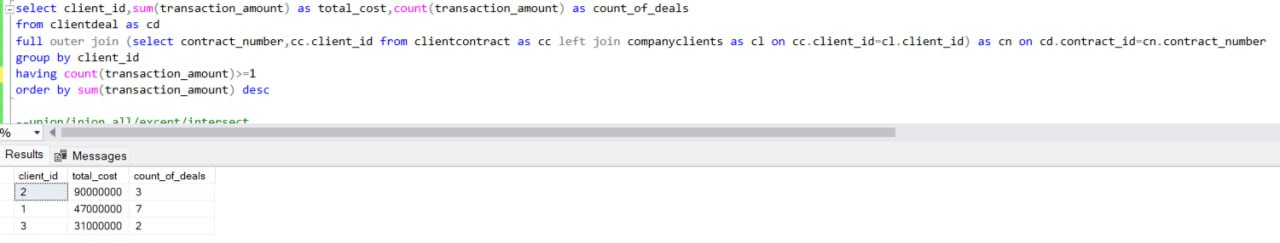


Рисунок 15 - Количество сделок для каждого клиента

1. Выведем данные “крупных” и/или длительных контрактах (рисунок 16)

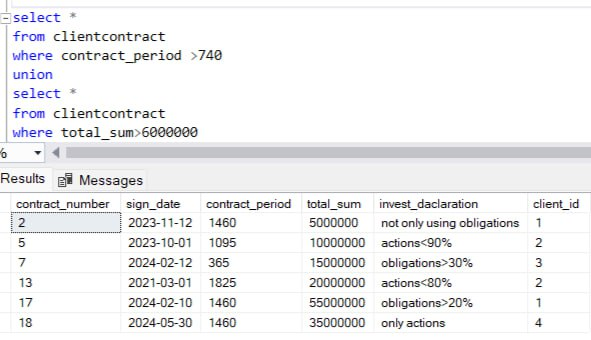


Рисунок 16 - Данные о “крупных” и/или длительных контрактах

1. Данные о “крупных” и/или длительных контрактах

с повторениями, если контракт подходит под оба условия (рисунок 17)

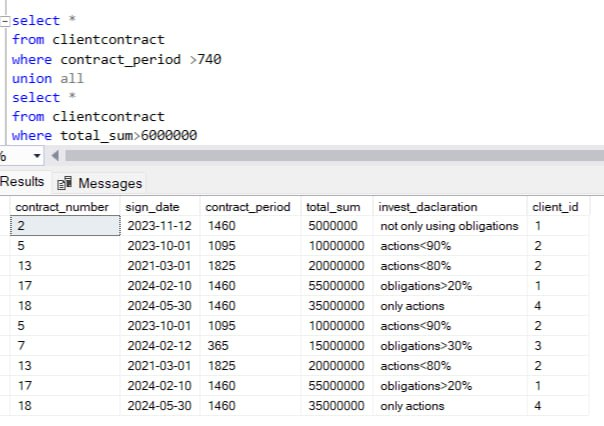


Рисунок 17 - Данные о “крупных” и/или длительных контрактах

17)Выведем данные “крупных” и длительных контрактах (рисунок 18)

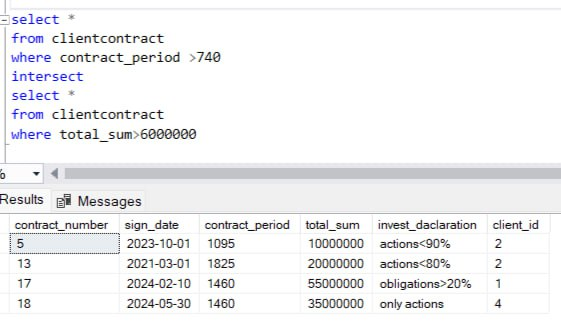


Рисунок 18 - Данные о “крупных” и длительных контрактах

1. Выведем данные о длительных контрактах, за исключением крупных (рисунок 19)

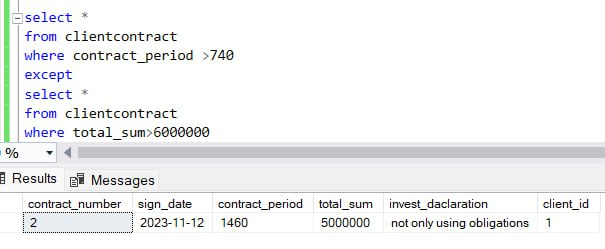


Рисунок 19 - Данные о длительных контрактах, за исключением крупных

18) Доходы от акций и облигаций для каждого клиента (рисунок 20)

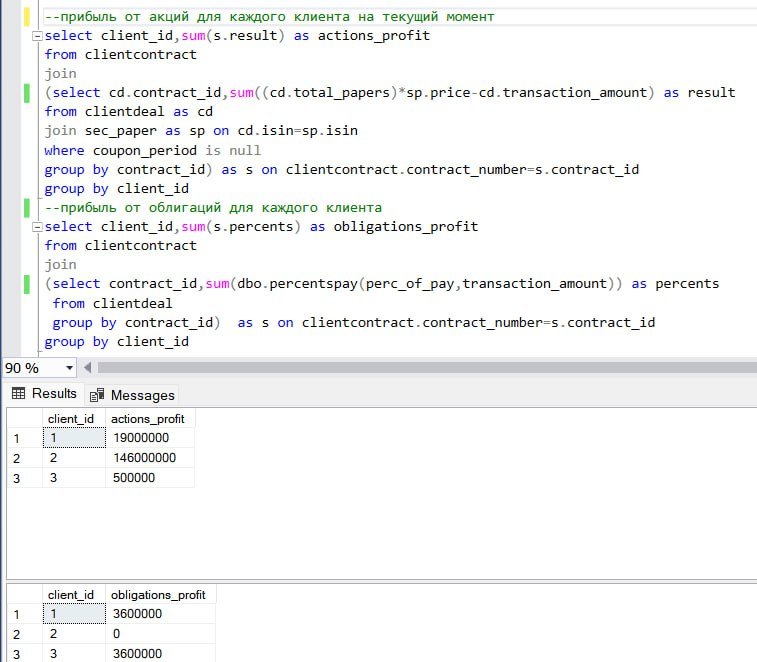


Рисунок 20 - Доходы от акций и облигаций для каждого клиента

**Заключение**

В данной работе была выбрана предметная область для создания базы данных и сформированы требования к ней. Затем была спроектирована ER-модель базы данных "Управляющая компания" и обоснован выбор кардинальных чисел связи. После было проведено преобразование спроектированной ER-модели в реляционную модель с обоснованием типов данных, ключей, правил обеспечения минимальной кардинальности. Изучены основные средства работы языка запросов Transact-SQL. Реализована база данных средствами СУБД SQL Server. В базе данных реализованы все запросы, выделенные в процессе проектирования, описаны механизмы реализации обеспечения целостности данных в выбранной системе управления базами данных.

# 4. Приложение А

Листинг кода:

USE master;

GO

IF DB\_ID (N'management\_company') IS NOT NULL

DROP DATABASE management\_company;

GO

CREATE DATABASE management\_company;

go

USE management\_company;

GO

--создаем таблицу с данными клиентов компании

CREATE TABLE companyclients (client\_id int identity(1,1) not null primary key,--not null primary key,

inn CHAR(10) not null unique,

j\_adress varchar(100) not null,

legal\_form smallint,

email varchar(100) not null unique,

CONSTRAINT check\_legal\_form CHECK(legal\_form in (1,2,3))

)

go

--создадим таблицу с договором клиента

CREATE TABLE clientcontract (contract\_number int not null primary key,

sign\_date date not null,

contract\_period int not null default 365,

total\_sum float not null,

invest\_daclaration varchar(100) not null,

client\_id int not null

constraint fc\_clientcontr foreign key(client\_id) references companyclients(client\_id)

ON DELETE CASCADE

ON UPDATE CASCADE)

go

--создаем таблицу с данными о ценных бумагах

CREATE TABLE sec\_paper (isin char(12) not null primary key,

trading\_platform varchar(100),

issuer varchar(100) not null,

price int not null,

amount int not null)

go

--создаем таблицу с данными о сделках в рамках договора

CREATE TABLE clientdeal (deal\_id int identity(1,1) not null primary key,

direction smallint not null default 1,--направление сделки -1,1

sign\_date date not null,

transaction\_amount float not null ,

total\_papers int not null,

coupon\_period int , --null для акций

perc\_of\_pay float, --null для акций

contract\_id int not null,

isin char(12) not null,

CONSTRAINT check\_direction CHECK(direction in (-1,1)),

constraint fc\_contr foreign key(contract\_id) references clientcontract(contract\_number),

constraint fc\_isin foreign key(isin) references sec\_paper(isin)

ON DELETE CASCADE

ON UPDATE CASCADE )

go

--запретим обновлять поля в сделке

drop trigger if exists clientdeal\_trigger\_upd

go

CREATE TRIGGER clientdeal\_trigger\_upd ON clientdeal

for UPDATE

AS

begin

RAISERROR('You can not update all fields, related to monetary part of deal! ', 16, 1);

rollback transaction;

end

go

--в таблице с данными о ценных бумагах введем запрет на обновление всех полей, за исключением текущей цены

drop trigger if exists papers\_trigger\_upd

go

CREATE TRIGGER papers\_trigger\_upd ON sec\_paper

for UPDATE

AS

begin

IF UPDATE(isin) and UPDATE(trading\_platform) and UPDATE(issuer) and UPDATE(amount)

RAISERROR('You can not update all fields, instead except of current price! ', 16, 1);

if UPDATE(price)

update sec\_paper

set

price = (select price from inserted where inserted.isin = sec\_paper.isin)--т.к. идентиф

where isin in (select isin from inserted where inserted.isin = sec\_paper.isin)

end

go

--запрет на изменение всех полей sec\_paper (кроме текущей цены), так как все остальные поля изменениям не подлежат

--заполним созданные таблицы

insert into companyclients(inn,j\_adress,legal\_form,email) values ('1749308428','Moscow, Paslannikov per., h.5,b.1',1,'catcomp@mail.ru')

, ('5828502174','Moscow, Gospitalny per., h.2',1,'anothercomp@mail.ru')

, ('4936799397','Moscow, Brigadirsky per., h.5,b.3',2,'insurancecomp@mail.ru'),

('4528573910','Moscow, Rubtsov per., h.4,b.1',3,'bigbussines@gmail.com') --новый клиент, пока контракт один заключили, сделок еще нет

insert into clientcontract(contract\_number,sign\_date,contract\_period,total\_sum,invest\_daclaration,client\_id) values (2,'2023-11-12',1460,5000000,'not only using obligations',1)

,(5,'2023-10-01',1095,10000000,'actions<90%',2),

(7,'2024-02-12',365,15000000,'obligations>30%',3),

(1,'2022-06-12',730,5000000,'not only obligations',1),

(13,'2021-03-01',1825,20000000,'actions<80%',2),

(17,'2024-02-10',1460,55000000,'obligations>20%',1) ,

(18,'2024-05-30',1460,35000000,'only actions',4); --контракт с новым клиентов заключили, сделки еще не провели

insert into sec\_paper(isin,trading\_platform,issuer,price,amount) values('ru1638429872','moex','sber',1000,50000),

('ru8739065119','moex','yandex',3000,400000),--облигация

('ru1197536784','moex','vtb',5000,600000),

('ru9066231178','moex','renessans',1500,450000),

('ru7409750984','moex','gazprombank',4000,1000000),

('ru9421759833','moex','tinkoff',5000,700000),

('ru8471251837','moex','x5',5000,1500000),--облигация

('us3752852299','nasdaq','google',10000,1500000),

('us9933567811','nasdaq','amazon',10000,2500000)

insert into clientdeal(direction,sign\_date,transaction\_amount,total\_papers,coupon\_period,perc\_of\_pay,contract\_id,isin) values(-1,'2023-11-11',1000000,1000,NULL,NULL,2,'ru1638429872'),

(1,'2023-12-11',3000000,1000,24,12,2,'ru8739065119'),

(1,'2023-09-10',30000000,25000,NULL,NULL,5,'ru1197536784'),

(1,'2023-11-11',4000000,1000,NULL,NULL,1,'ru7409750984'),

(1,'2023-12-11',3000000,1000,36,8,17,'us9933567811'),

(1,'2023-09-10',30000000,15000,NULL,NULL,13,'ru1197536784'),

(1,'2023-11-11',1000000,1000,NULL,NULL,7,'ru9066231178'),

(1,'2022-12-11',30000000,1000,12,12,7,'ru8471251837'),

(1,'2023-09-10',30000000,9000,NULL,NULL,13,'ru7409750984'),

(1,'2024-11-11',1000000,1000,NULL,NULL,1,'ru9421759833'),

(1,'2023-12-11',30000000,1000,36,10,17,'ru8471251837'),

(-1,'2024-07-10',5000000,5000,NULL,NULL,17,'ru7409750984');

--создадим функцию, которая вычисляет количество дней со времени подписания контракта до текущего дня

drop function if exists passeddays

go

create function passeddays(@contract\_date date)

returns int as begin

declare @current\_date datetime = GETDATE();

declare @passed int;

set @passed =datediff(day,@contract\_date, @current\_date );

return @passed;

end

go

--смотрим, сколько дней по контракту осталось, а какие договоры закончились

select contract\_number,dbo.passeddays(sign\_date) as daysleft

from clientcontract

--теперь создадим функцию для подсчета денег, полученных в качестве процента по облигациям за первый купонный период

drop function if exists percentspay

go

create function percentspay(@perc\_of\_pay float,@transaction\_amount float)

returns int as begin

declare @total int

if @perc\_of\_pay is not null --для облигаций

set @total =@perc\_of\_pay\*@transaction\_amount/100.0

else

set @total=0

return @total

end

go

select contract\_id,sum(dbo.percentspay(perc\_of\_pay,transaction\_amount)) as percents

from clientdeal

group by contract\_id

drop function if exists dealstatus

go

create function dealstatus(@coupon\_period int,@sign\_date date)

returns bit as

begin

declare @current\_date datetime = GETDATE();

declare @passed int;

declare @dstatus char;

set @passed =datediff(month,@sign\_date, @current\_date )

if @passed>@coupon\_period

set @dstatus=1 --done

else

set @dstatus=0 --active

return @dstatus;

end

go

--найдем завершенные сделки. Завершенными считаются сделки по облигациям после завершения купонного периода.

select deal\_id,dbo.dealstatus(coupon\_period,sign\_date) as deal\_status

from clientdeal

--создадим представления для запросов содержащих основную информацию

--основная информация о контрактах, клиентах, количестве внесенных средств

DROP VIEW IF EXISTS clientsview

GO

CREATE VIEW clientsview AS

select cc.client\_id,cc.legal\_form,cc.j\_adress,cc.email,cl.tot\_sum,cl.numb\_of\_contracts

from companyclients as cc

join (select client\_id, sum(total\_sum) as tot\_sum,count(contract\_number) as numb\_of\_contracts

from clientcontract

group by client\_id) as cl on cl.client\_id=cc.client\_id

GO

--создадим представления для таблицы со сделками, отдельно для акций и облигаций

--представление для облигаций

DROP VIEW IF EXISTS obligationsview

GO

CREATE VIEW obligationsview AS

select \*

from clientdeal

where coupon\_period is not NULL and perc\_of\_pay is not NULL

GO

--представление для акций

DROP VIEW IF EXISTS actionsview

GO

CREATE VIEW actionsview AS

select \*

from clientdeal

where coupon\_period is NULL and perc\_of\_pay is NULL

GO

--Теперь создадим представления данные для ценных бумаг. Часто требуется отделить российские ценные бумаги от иностраннх

DROP VIEW IF EXISTS rupapers

GO

CREATE VIEW rupapers AS

select \*

from sec\_paper

where trading\_platform='moex'

GO

select \*

from rupapers

DROP VIEW IF EXISTS foreignpapers

GO

CREATE VIEW foreignpapers AS

select \*

from sec\_paper

where trading\_platform!='moex'

GO

select \*

from foreignpapers

--создадим триггер на вставку для представлений по ценным бумагам

drop trigger if exists insert\_viewpapers\_trigger;

go

create trigger insert\_viewpapers\_trigger

on foreignpapers

instead of insert

as

begin

insert into foreignpapers(isin,trading\_platform,issuer,price,amount)

select isin,trading\_platform,issuer,price,amount from inserted --вставка значений с использованием select

end

go

insert into foreignpapers(isin,trading\_platform,issuer,price,amount) values ('us1843000231','nasdaq','netflix',2000,20000)

--создадим индексы на часто используемые поля при поиске

DROP INDEX IF EXISTS client\_ind ON companyclients

CREATE INDEX client\_ind

ON companyclients(inn)

include(email,legal\_form)

GO

DROP INDEX IF EXISTS contract\_ind ON clientcontract

CREATE INDEX contract\_ind

ON clientcontract(sign\_date)

include(contract\_period)

GO

DROP INDEX IF EXISTS deal\_ind ON clientdeal

CREATE INDEX deal\_ind

ON clientdeal(sign\_date)

GO

--4 часть --обязательные запросы

--выберем ценные бумаги, которые использовались в сделках, исключив повторы

select distinct(isin)

from clientdeal

--выбор, упорядочивание и именование полей (создание псевдонимов для полей и таблиц/представлений)

select contract\_number as number, contract\_period as days\_amount, total\_sum

from clientcontract

where total\_sum>15000000

order by total\_sum

--соединение таблиц

--Выделим основную информацию

select cl.client\_id,cl.legal\_form,cl.email,cc.contract\_number,cc.contract\_period,cc.total\_sum from companyclients as cl

right join clientcontract as cc on cc.client\_id=cl.client\_id

select cl.contract\_number,cl.contract\_period,cl.total\_sum,cd.transaction\_amount,cd.direction ,sp.isin, sp.issuer from clientcontract as cl

full outer join clientdeal as cd on cl.contract\_number=cd.contract\_id

full outer join sec\_paper as sp on sp.isin=cd.isin

--информация о контракте для нового клиента войдет, а в полях со сделкой будет NULL

select cl.contract\_number,cl.contract\_period,cl.total\_sum,cd.transaction\_amount,cd.direction from clientcontract as cl

left join clientdeal as cd on cl.contract\_number=cd.contract\_id

--теперь left join относительно clientcontract, поэтому данные о новом клиенте, пока не совершившем сделки на основании его поктракта, войдут в результат

--условия выбора записей (null/like/between/in/exists

--выберем контракты, заключенные недавно и отсортируем их в порядке убывания

select \* from clientcontract

where sign\_date between '2024-01-01' and '2024-06-01' --between

order by total\_sum desc

--посмотрим, какие ценные бумаги американской фондовой биржи из представленной таблицы уже использовались в сделках

select \* from sec\_paper

where isin like 'us%' and isin in (select isin from clientdeal) --in,like

order by price desc

--посмотрим,сколько в процентах управляющая компания вложила денег в акции относительно всех средств. очень большой % допускать опасно и нужно вкладываться в облигации

select sum(transaction\_amount)/(select sum(total\_sum) from clientcontract )\*100 as percent\_spent\_on\_actions

from clientdeal

where coupon\_period is NULL and perc\_of\_pay is NULL --условия на акции

--exists и псевдонимы таблиц

--найдем все договоры, на основе которых пока не были заключены сделки

select \* from clientcontract as cc

where not exists (select \* from clientdeal as cd where cc.contract\_number=cd.contract\_id)

--сортировка записей order by -asc desc

--группировка записей, использование аггрегирующих функций (sum/avg/count/min/max)

select contract\_id,sum(transaction\_amount) as total\_spent,count(contract\_id) as total\_count, avg(transaction\_amount) as mean\_amount

from clientdeal

group by contract\_id

order by total\_spent asc

--/min/max

select contract\_id,min(sign\_date) as min\_date, max(sign\_date) as max\_date

from clientdeal

group by contract\_id

order by contract\_id asc

--клиенты, заключившие больше одного контракта

select client\_id,count(client\_id) as contracts\_count

from clientcontract

group by client\_id

having count(client\_id)>1

--найдем количество денег от каждого клиента уже было вложено в сделки в соответствии с договором

select client\_id,sum(transaction\_amount) as total\_cost,count(transaction\_amount) as count\_of\_deals

from clientdeal as cd

full outer join (select contract\_number,cc.client\_id from clientcontract as cc left join companyclients as cl on cc.client\_id=cl.client\_id) as cn on cd.contract\_id=cn.contract\_number

group by client\_id

having count(transaction\_amount)>=1 --можно убрать, тогда будет 4 клиент, который пока без сделок

order by sum(transaction\_amount) desc

--union/inion all/except/intersect

--крупные и длительные контракты

select \* from clientcontract

where contract\_period >740 --длительные контракты

union

select \* from clientcontract

where total\_sum>6000000 --крупные контракты

--крупные и длительные контракты с использованием union all

select \* from clientcontract

where contract\_period >740

union all

select \* from clientcontract

where total\_sum>6000000

--одновременно крупные и длительные контракты

select \* from clientcontract

where contract\_period >740

intersect

select \* from clientcontract

where total\_sum>6000000

--длительные контракты, не считая крупных

select \* from clientcontract

where contract\_period >740

except

select \* from clientcontract

where total\_sum>6000000

--обновление данных в price для ценной бумаги

update sec\_paper

set price=3000 where issuer='netflix'

update companyclients

set j\_adress='Moscow, Paslannikov per., h.5,b.9' where client\_id=1

--прибыль от акций по каждой сделки

select cd.deal\_id,cd.transaction\_amount,cd.total\_papers,cd.contract\_id,cd.isin,sp.price, sp.amount,(cd.total\_papers)\*sp.price-cd.transaction\_amount as result

from clientdeal as cd

join sec\_paper as sp on cd.isin=sp.isin

where coupon\_period is null

--прибыль от акций для каждого клиента на текущий момент

select client\_id,sum(s.result) as actions\_profit

from clientcontract

join (select cd.contract\_id,sum((cd.total\_papers)\*sp.price-cd.transaction\_amount) as result

from clientdeal as cd

join sec\_paper as sp on cd.isin=sp.isin

where coupon\_period is null

group by contract\_id) as s on clientcontract.contract\_number=s.contract\_id

group by client\_id

--прибыль от облигаций для каждого клиента

select client\_id,sum(s.percents) as obligations\_profit

from clientcontract

join

(select contract\_id,sum(dbo.percentspay(perc\_of\_pay,transaction\_amount)) as percents

from clientdeal

group by contract\_id) as s on clientcontract.contract\_number=s.contract\_id

group by client\_id