

## Упражнение 3 Моделиране на данните

В практиката съществуват редица модели – йерархичен, релационен, мрежови, диаграми със същности и връзки (Entity-Relationship Diagrams), диаграми на структурите данни (Data-Structure Diagrams). Всеки един от първите три посочени модела са приложими само при конкретна организация на базите данни, което нарушава изискването за независимост на логически модел от физическия. Моделът със същности и връзки и диаграмите на структурите данни задоволяват напълно всички поставени изисквания и поради това са приложими във всякакви ситуации. Преди да преминем към конкретното разглеждане на тези два модела ще отделим конкретно разглеждане на тези два модела ще отделим внимание на някои основни термини, свързани с базите данни.

- Обект (същност; Entity, Object) – всеки обект, за когото събираме данни.
- Тип (Entity Set, Object Type, Entity Type) – множество от възможните стойности, които приема даден обект.
- Атрибут (поле, елемент, Attribute, Field, Data Item, Data Element) – определена характеристика на обекта
- Стойност (Value) – точно определено значение, което притежава всеки атрибут на всеки обект.
- Ключ (ключов атрибут, първичен ключ; Key, Key Attribute) – атрибут, който определя еднозначно всеки един обект от дадено множество. В някои случаи това не може да бъде реализирано единствено от един атрибут, поради което ключът включва два или повече атрибута. Тогава говорим за конкатениран ключ (Concatenated Key).
- Външен ключ (Foreign Key) – атрибут, който принадлежи на даден тип и в същото време служи като ключов атрибут на друг тип. Чрез външните ключове се осъществяват връзките между различните типове данни в базата.

### Диаграма на структурите данни

Примерна диаграма на структурите данни е показана на Фигура 1. Този пример съдържа описание на част от данните, използвани от информационната система за факултетна канцелария.

За обозначаване на типовете в диаграмата на структурите данните се използват правоъгълници, като на всяко хранилище от диаграмите на потоците данни се съпоставя съответен тип в диаграмата на структурите данни. За всеки тип вътре в правоъгълника се посочват съответните атрибути, като ключовият атрибут или атрибути са подчертани. Стрелките, посредством, които се свързват отделните типове се наричат логически указатели. Логическите указатели отразяват съществуващите в базата данни връзки и тяхната посока. Основната характеристика на логическите указатели е тяхната кардиналност. Кардиналността се отбелязва със символи, поставени над стрелките и отразява броя на връзките между двата типа. Различаваме три типа кардиналност:

1:1 съответства на връзка от тип „едно-към-едно“ и в двете посоки.

1:M описва връзката “едно-към-много” в едната посока и „едно-към-едно“ в обратната.

M:N съответства на връзката с тип „едно-към-много“ и в двете посоки. Тази връзка е известна и като „много-към-много“.

### Създаване на диаграми на структурите данни

Създаването на една диаграма на структурите данни преминава през девет основни фази:

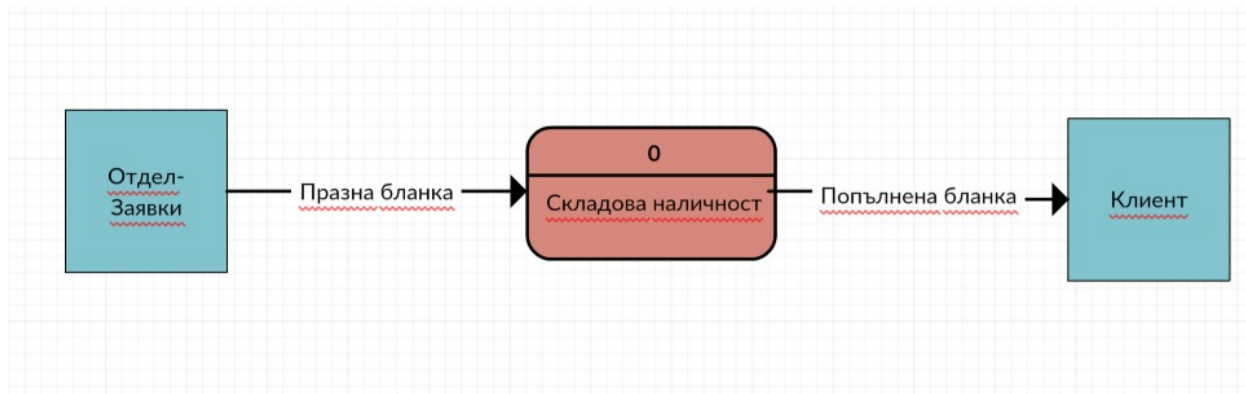
1. Определяне на типовете обекти. На всяко хранилище и всеки входен поток, използван за обновяване на хранилище съответства по един тип. За изходните потоци не се определят типове.
2. Определяне на ключовите атрибути за всеки тип. В този случай, ако атрибутите на даден тип не позволяват да бъде открит ключов атрибут, такъв се създава изкуствено.
3. Определяне на неключовите атрибути за всеки тип. Изброяват се всички атрибути, принадлежащи на съответния тип, като в случаите, когато имаме повтаряемост на атрибут или група атрибути, за нейното обозначение се използва символа {}.
4. Определяне на логическите указатели и тяхната кардиналност.
5. Създаване на първа нормална форма. Първата нормална форма изисква всеки тип да включва само еднозначни атрибути. Премахването на многозначните атрибути става чрез отделянето им като нов тип, за който също определяме ключов атрибут.
6. Създаване на втора нормална форма. Втора нормална форма изисква всеки един атрибут от тип с конкатениран ключ да зависи от целият ключ, а не само от част от него. За да бъде типът във втора нормална форма. Атрибутите, които не отговарят на поставеното условие се отделят в нов тип, заедно с онази част от ключа, от която те зависят.

Ако базата данни не е във втора нормална форма възникват следните четири проблема:

- (1) Поради повтаряемост на данните, тяхната промяна е много трудна и може да доведе до грешки, тъй като едни и същи данни се съхраняват на няколко различни места.
  - (2) Поради дублирането на информацията е възможно едни и същи данни да имат различно съдържание. В този случай имаме несъвместимост на данните.
  - (3) Проблеми с въвеждането на нови елементи.
  - (4) Проблеми при изтриване на елементи.
7. Създаване на трета нормална форма. Трета нормална форма гласи, че стойността на неключовите атрибути не трябва да зависи от други неключови атрибути. За да бъде типът в трета нормална форма атрибути, между които съществуват такива зависимости се отделят като нов тип.
  8. Премахване на всички външни ключове, които не са част от конкатениран ключ. Външните ключове се премахват, тъй като в случая ролята им се поема от логически указатели. Външни ключове, които са част от конкатениран ключ се запазват, за да не се наруши изискването за уникална идентификация на атрибутите в типа, посредством използване на ключ.
  9. Актуализиране на речника на данните и диаграмите на потоците данни, така че да съответстват на новата структура на данните.

## Пример за съставяне на диаграма на структурите данни

Като пример ще разгледаме система за складова наличност, чиято контекстна диаграма е представена на фигура 1.



Фигура 1. Контекстна диаграма

Принципът на работа на тази система е следният:

1. За всяка заявка от страна на клиента се добавя запис във файла „ЗАЯВКИ“. Този запис съдържа номера на поръчката и друга информация.
2. За всяка заявка Отдел-Заявки подготвя и изпраща към склада бланка, описваща подадената заявка. Видът на бланката е показан на Фигура 2.

БЛАНКА N 123			
КЛИЕНТ N	005		
КЛИЕНТ ИМЕ	Иванов		
Артикул N	Артикул име	Заявено количество	Доставено количество
567	Радио	3	3
382	Стол	5	0

Фигура 2. Вид на бланката

3. Работниците в склада проверяват ръчно дали посочените стоки са в наличност, след което попълват бланката.
4. След като бланката бъде попълнена оригиналът се запазва, а клиентът получава копие.
5. На базата на оригинала информационната система актуализира файла „ЗАЯВКИ“ файла „Складова Наличност“. Структурата на тези два файла е показана съответно на Фигура 3 и Фигура 4.

Запис N	Заявка N	Клиент N	Клиент име	Артикул N	Артикул име	Заявено Количество	Доставено количество	Ед. Цена	Обща сума
1	12	005	Иванов	567	Радио	3	3	105	315

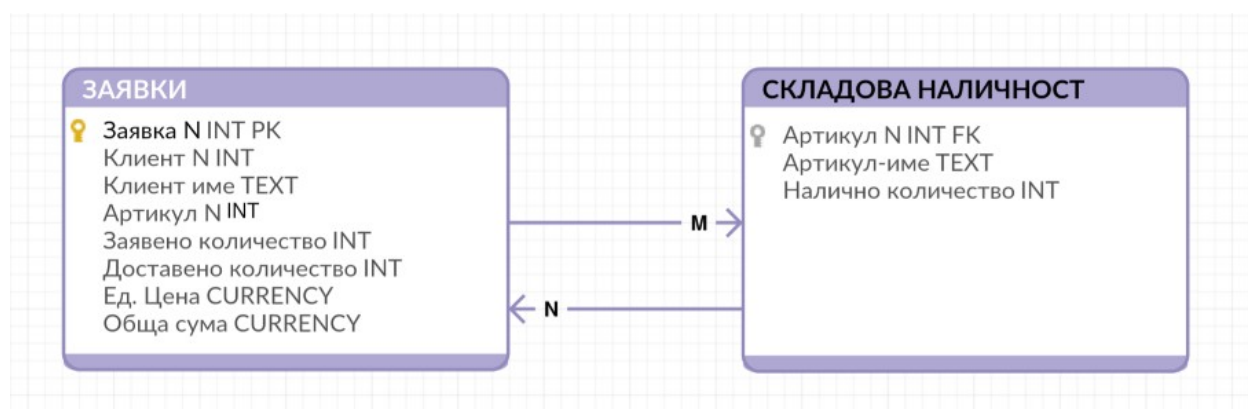
2	13	005	Иванов	382	Стол	5	0	70	0
3	14	006	Петров	567	Радио	6	4	105	420
4	15	132	Стояно в	201	Лампа	1	1	5	5

Фигура 3. Файл ЗАЯВКИ

Запис N	Артикул N	Артикул име	Налично количество
1	567	Радио	7
2	382	Стол	0
3	12	Маса	14
4	354	Печка	6
5	201	Лампа	1

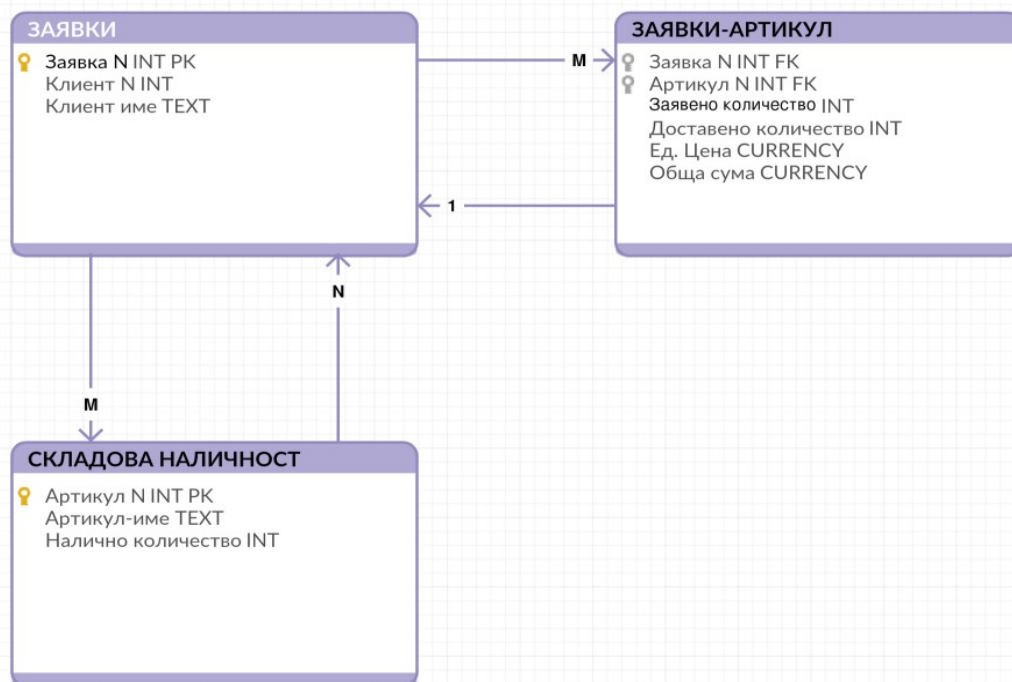
Фигура 4. Файл СКЛАДОВА НАЛИЧНОСТ

6. Ще приложим алгоритъма за построяване на диаграми на структурите данни, за да опишем така представената информационна система за складова наличност.
  1. Типовете обекти са два – СКЛАДОВА НАЛИЧНОСТ и ЗАЯВКИ.
  2. Ключов атрибут за първия обект е полето Заявка N, а за втория Артикул N.
  3. Неключови атрибути за първият обект са: Клиент N, Клиент име, Артикул N, Артикул име, Заявено количество, Ед. Цена и Обща сума. За втория обект неключови са атрибутите: Артикул име и Налично количество.
  4. За така описаните обекти определяме логическите указатели и тяхната кардиналност. Получената структура е представена на Фигура 5.



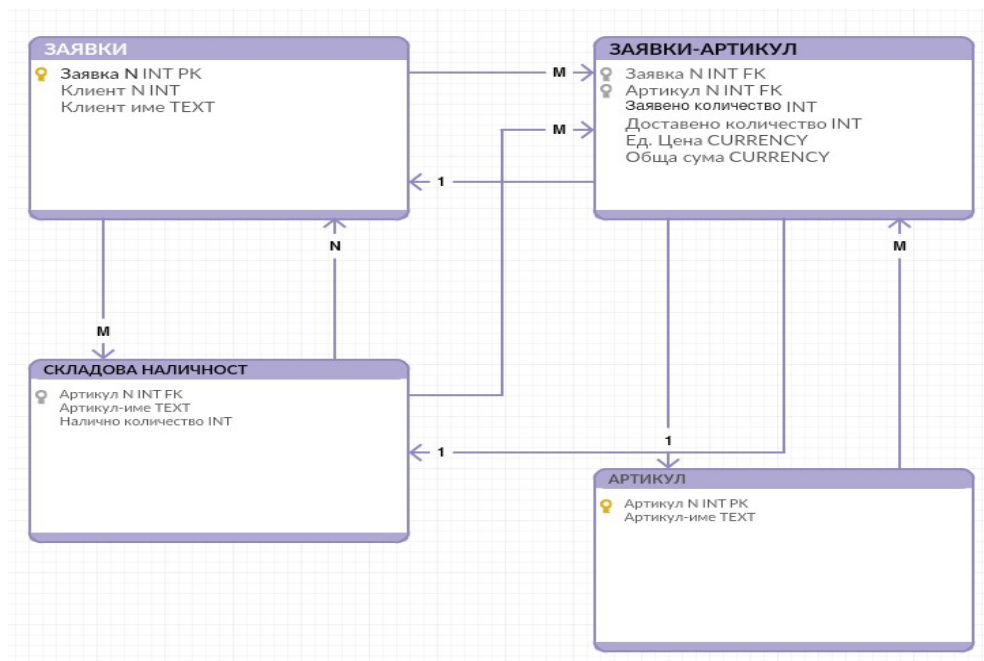
Фигура 5. Първоначална структура от данни

5. Поради повтаряемостта на групата {Артикул N, Артикул име, Заявено количество, Доставено количество, Ед. Цена и Обща сума} в обекта ЗАЯВКИ, тя се отделя като самостоятелен обект, наречен ЗАЯВКА-АРТИКУЛ. Ключът съответстващ на този обект е конкатениран и включва полетата Заявка N и Артикул N. Първата нормална форма има вида представен на Фигура 6.



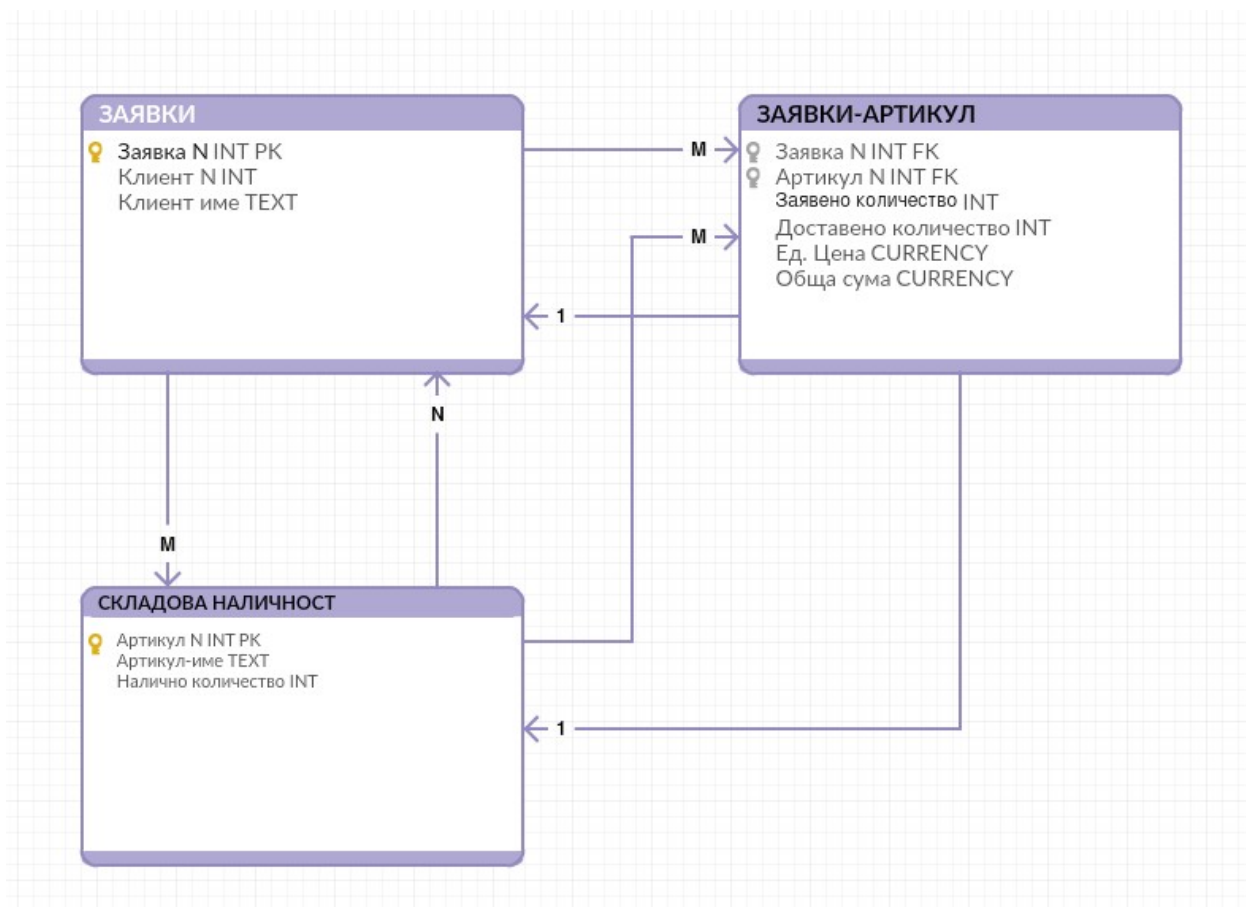
Фигура 6. Първа нормална форма

6. Втора нормална форма е представена на Фигура 7, а на Фигура 8 е представена междинна втора нормална форма.

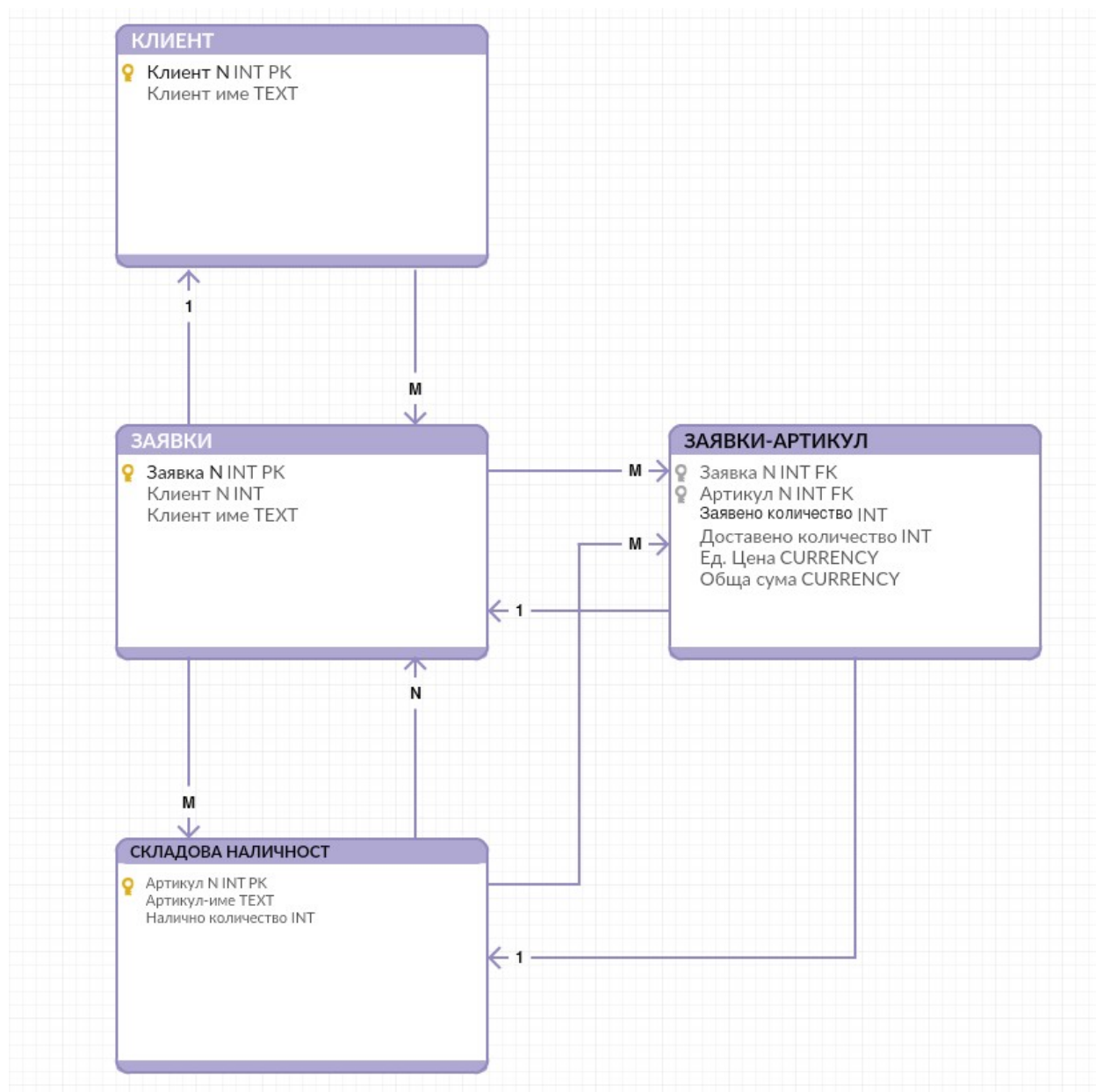


Фигура 8. Междинна втора нормална форма

7. Третата нормална форма има вида, представен на Фигура 9.



Фигура 7. Окончателна втора нормална форма



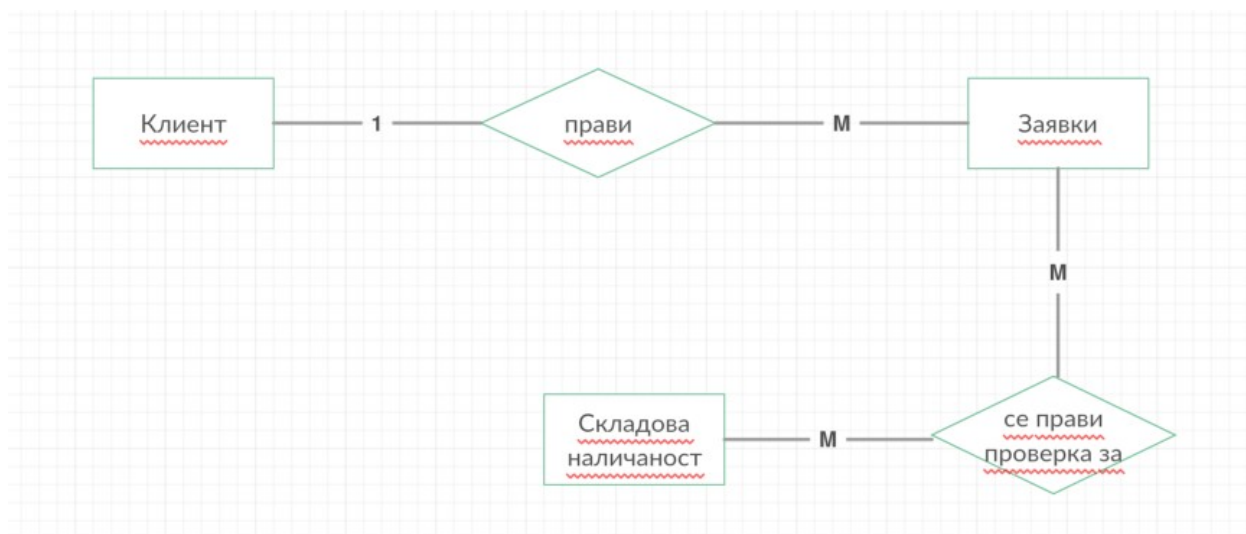
Фигура 9. Трета нормална форма

Създаване на диаграми със същности и връзки

При създаване на диаграми със същности и връзки следваме същият алгоритъм, както при диаграмите на структурите данни, като се съобразяваме със следните условия:

- (1) Атрибутите не се посочват на диаграмата.
- (2) При създаване на типове имаме възможност за създаване на композитни типове, подтипове и супер типове.
- (3) Логическите указатели са заменени от връзки.
- (4) Имаме възможност да създаваме рекурсивни връзки и да представяме повече от една връзка между два типа.

Пример за създаване на диаграма със същности и връзки



Фигура 10. Диаграма същности и връзки

#### Задачи:

Задача 1: Проектира се база от данни, която да съдържа информация за поръчки на клиенти. Да се разгледа следната таблица и да се нормализира. Използвайте [www.creately.com](http://www.creately.com) за да начертаете диаграмата. От възможните диаграми изберете Database diagrams.

Поръчки на клиенти	
Идентификатор на поръчка	
Дата на поръчка	
Идентификатор на клиент	
Наименование на клиент	
Адрес на клиент	
Идентификатор на продукт	
Наименование на продукт	
Категория на продукт	
Цена на продукт	
Поръчано количество от продукт	



Задача 2. Проектира се база данни, която обработва информацията, необходима на една обществена библиотека. Да се разгледа следната таблица и да се нормализира.

Наемане на книги	
Идентификатор на наемане	
Дата на наемане на книги	
Идентификатор на читател	
Име на читател	
Презиме на читател	
Фамилия на читател	
Адрес на читател	
Идентификатор на книга	
Заглавие	
Автор(и)	
Издателство	
ISBN	
Година на издаване	
Дата на връщане на книга	

Задача 3. Да се състави диаграма със същности и връзки на база данни имаща четири типа: УПРАВИТЕЛ, СЛУЖИТЕЛ, ФИЛИАЛ и КЛИЕНТ. В диаграмата да бъдат представени и връзките: управител ръководи филиал. Служител работи във филиал, управител е началник на служител и клиент е обслужван в филиал. На диаграмата да бъдат отразени и кардиналностите на връзките.