

2. ER модел (Entity Relationship Model)

При проектиране на база от данни много често се използва низходящ подход. Започва се на по-високо ниво на абстракция с изясняване на семантиката на нещата в предметната област и след това се преминава на сравнително по-ниско ниво на абстракция, представено от модела на данните. Съществува група от модели, известни като “семантичен” или “информационно-логически”, чиято основна роля при проектиране на БД е да изясни смисловото значение на данните, които ще се съхраняват в БД и трябва възможно най-точно да представят фактите в ПО. Един от най-популярните такива модели е ER модела, предложен от Чен през 1976г. и след това нееднократно усъвършенстван от Чен и от други изследователи в областта на БД. Характерно за този модел е използването на графична нотация, т.е. представяне на ER модела чрез ER диаграма. В литературата се използват леко различаващи се графични нотации за ER диаграмите. Ние ще използваме нотация, която е близка до оригиналната, предложена от Чен.

ER моделът е неформален модел, т.е. използва неформални, интуитивни понятия. Следва изложение на основните понятия в модела.

Entity / същност / обект

Това е нещо, което съществува реално, различимо е, представлява интерес за ПО и за което ще се съхранява информация в БД. Обектът може да е нещо конкретно – служител, студент, банкова сметка, клиент, доставчик, стока или абстрактно понятие.

Entity set / множество същности / клас обекти / тип обекти

Това е множеството от всички подобни обекти в ПО, т.е. обекти притежаващи общи свойства. Например, всички служители в предприятието, всички клиенти на фирмата, всички студенти във факултета. В ER диаграма клас обекти се представя чрез именован правоъгълник.

Attribute / атрибут / свойство

Обектите се характеризират чрез свойства, т.е. свойството е някакъв факт за обекта. Напр., обект служител може да има свойствата – име, ЕГН, пол, длъжност, адрес и т.н. Свойство (свойства), чието значение еднозначно идентифицира обект в рамките на определен клас обекти се нарича ключово свойство или **ключ**. За всеки клас обекти трябва да може да се определи поне един ключ, тъй като обектите по дефиниция са различими. Напр., за клас обекти служители ключ може да е ЕГН или някакъв вътрешен (служебен) номер на служителя, който се използва в предприятието; ключ на банкова сметка е номера, ключ за студента е факултетния номер. Могат да се различават следните видове свойства (в зависимост от значенията, които могат да приемат):

- Ключово или не ключово свойство

- Просто или съставно. Ако значението на свойството е неделимо от гледна точка на ПО, то е просто, в противен случай е съставно. Напр., свойството “име на служител” може да се разглежда като съставно, тъй като се състои от простите свойства “първо име”, “презиме” и “фамилия”.

- Еднозначно или многозначно. Свойството е многозначно, ако може да има няколко значения за един обект.

В ER диаграма свойство се представя чрез именована елипса, която е свързана чрез линия с правоъгълника на класа обекти. Ключът за всеки клас обекти се показва чрез подчертаване.

Relationship / връзка

Връзката свързва или асоциира обекти (2, 3 или повече) от различни класове или от един и същи клас. Най-често се използват бинарни връзки, т.е. всяка връзка

свързва два обекта. Напр., връзка между служител и отдел се установява, когато служителят бъде назначен в отдела, връзка между клиент и сметка - когато клиентът си открие сметка. Връзките също могат да имат свойства, т.е. факт, който характеризира връзката между обектите, а не самите обекти. Напр., свойство на връзката между служител и отдел може да е дата на назначаване в отдела. Връзките също характеризират обектите.

Relationship set / множество връзки / клас връзки / тип връзки

Това е множеството от всички подобни връзки, които съществуват между два или повече класа обекти и имат еднаква семантика.

Вид на връзка

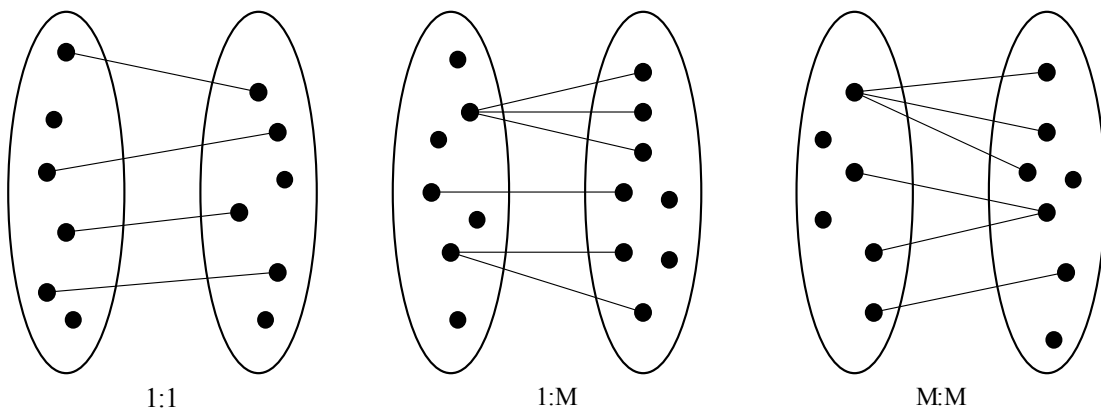
Класовете връзките имат различен вид в зависимост от това, обект от единия клас с колко обекта от другия може да бъде свързан в определен момент и обратно, и дали участието на обект във връзка от определен клас е задължително или не. Според първия критерий връзките биват от вид 1:1 (едно към едно), 1:M (едно към много) или M:M (много към много). Нека R е клас връзки между класовете обекти E1 и E2.

R е от вид 1:1 ако всеки обект от E1 е свързан най-много с един обект от E2 и обратно.

R е от вид 1:M ако всеки обект от E1 може да е свързан с много обекти от E2, но всеки обект от E2 е свързан най-много с един обект от E1.

R е от вид M:M ако всеки обект от E1 може да е свързан с много обекти E2 и обратно, т.е. няма ограничения за броя на връзките, в които обект може да участва.

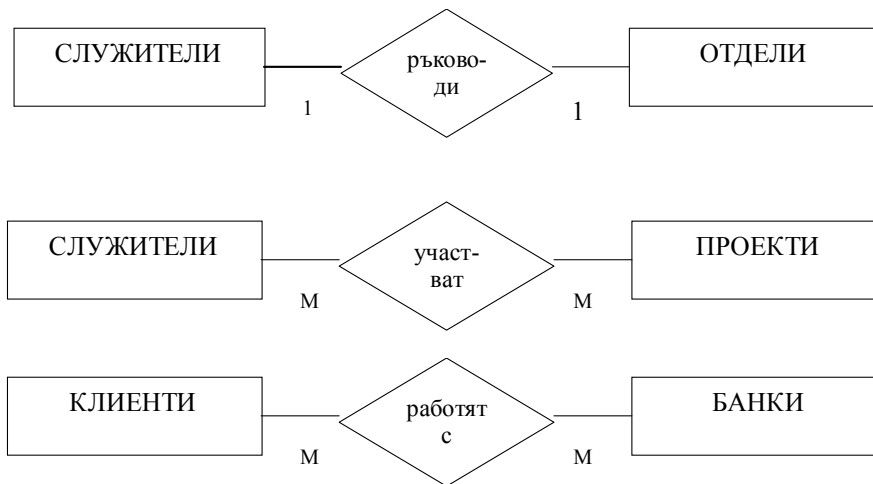
Според втория критерий (Participation constraint) участието на клас обекти в клас връзки може да е пълно (total) или частично (partial). Ако всеки обект от E1 участва поне в една връзка от клас R, се казва че участието на E1 в R е пълно, иначе участието е частично.



В ER диаграма клас връзки се представя чрез именован ромб, който е свързан с класовете обекти чрез линии, а вида на връзката се обозначава чрез надпис над линиите. Когато участието на класа обекти е пълно линията е двойна.

Примери за различни видове връзки.





Слаб клас обекти / Weak entity set

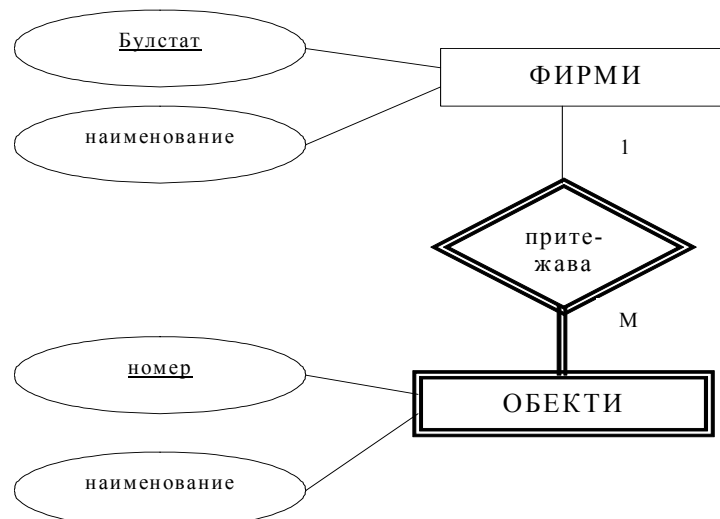
Класовете обекти в ПО могат да бъдат класифицирани на независими класове (наричат ги още силни) и на слаби класове. Обектите от независим клас имат независимо съществуване в ПО. Слаб обект се нарича обект, който не съществува самостоятелно в ПО, т.е. съществуването му зависи от съществуването на друг обект. Някои ги наричат още характеризираещ клас, защото често целта на тяхното съществуване в ПО е да характеризират обектите от някой друг клас.

Например, в една данъчна БД независим клас обекти са фирмите. Всяка фирма има ред свойства, като данъчен номер, Булстат, наименование, адрес на регистрация и т.н. Фирмите притежават различни обекти – магазини, заведения, хотели и др., като всяка фирма може да притежава няколко обекта, но всеки обект е собственост на една фирма и няма независимо съществуване.

Един друг пример, където може да се използва слаб клас, е за представяне на многозначно, съставно свойство. Напр., образованието или квалификацията на служителите, може да се разглежда като слаб клас със свойства – учебно заведение, дата на завършване, специалност, степен, и др.

Много често ключът на слабите обекти се формира от ключа на обекта, от когото зависи плюс собствени свойства на самия слаб обект. Напр., ключ на обект от клас ОБЕКТИ се състои от Булстат на фирмата и номер на обекта във фирмата.

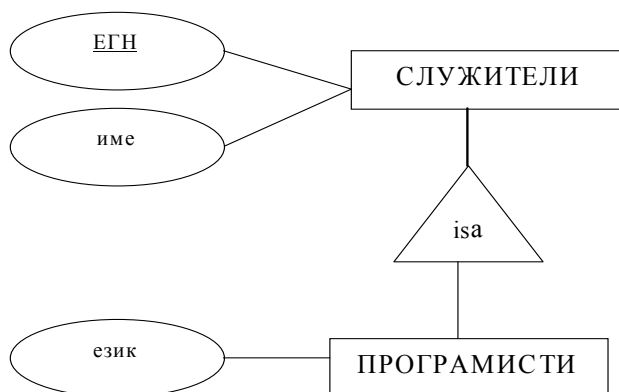
В ER диаграма слаб клас се изобразява чрез двойна линия на правоъгълника. Връзката между слабия клас и класа, от когото той зависи, се изобразява чрез ромб с двойна линия.





Връзка isa / подклас обекти

Ако обектите в един клас могат да бъдат групирани в подкласове, всеки от които има собствени специфични свойства в допълнение към свойствата характеризиращи целия клас, тогава в ER модела може да се изгради йерархия от класове обекти, свързани чрез специална връзка “isa”.



Всеки програмист е и служител, т.е. притежава всички свойства на класа СЛУЖИТЕЛИ, може да участва във всички връзки, в които участват служители. Казано по друг начин множеството на програмистите е подмножество на служителите. Това е концепция позната от обектно-ориентирания подход и според терминологията там СЛУЖИТЕЛИ е суперклас, а ПРОГРАМИСТИ е негов подклас. Подкласът наследява всички свойства и връзки от суперкласа.

ER моделът и диаграмата на определена ПО може да бъде изграден по два начина – възходящ и низходящ. При низходящо изграждане най-напред се определят основните класове обекти. След това се определят връзките, които съществуват между обектите и вида им. Най-накрая се детайлизират свойствата на обектите и връзките, като обикновено това не е еднократно действие. Възможно е по-късно да се добавят нови свойства с цел удовлетворяване на нови информационни потребности.

Една и съща концепция от реалния свят може да се възприема като обект от един потребител и като връзка от друг. Например, какво е “брак”: връзка между двама човека, която притежава ред свойства, като дата и място на сключване, номер на свидетелство, или е обект. Аналогично, дали образованието на служител да се

възприема като слаб клас обекти или като многозначно свойство на класа обекти служители. Когато се изгражда ER модел на ПО, решението се взема от АБД и следователно е субективно. На Рис.3 е представена ER диаграма на БД-служители, която по-нататък ще се използва като пример.

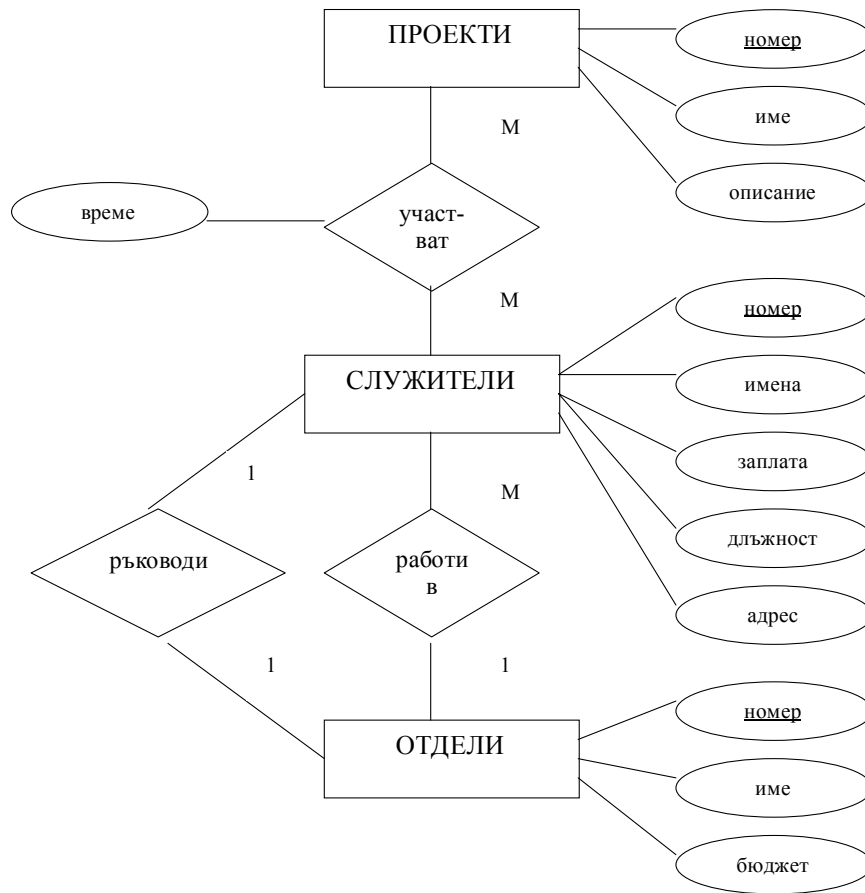


Рис.3. ER диаграма на БД-служители