Факултет по математика и информатика

към

Софийски университет „Св. Климент Охридски“

гр. София

**ПРОЕКТ**

по

Бази от данни

Тема: Създаване на база от данни за издателска къща

Изготвили:

Атанас Троянов, фн. 71228

Виталий Филипов, фн. 71256

Даниел Иванов, фн. 71225

Деница Генчева, фн. 71249

Ивана Христова, фн. 71265

Кирил Димитров, фн. 71229

Константин Добрев, фн. 71238

Мартин Анев, фн. 71232

Минко Гечев, фн. 71202

Цветан Чулев, фн. 71208

Дата на предаване: 03.06.2011 г.

Съдържание

1. [Описание на проекта](#проект)

2. [Описание на множествата същности](#същности)

3. Описание на E/R модела - по модули

3.1.[Описание на модула „Продукция“](#продукция)

3.2. [Описание на модула „Търговски отдел“](#търговия)

3.3. [Описание на модула „Счетоводно-правен отдел“](#счетоводство)

4. [E\R диаграма на модела на БД (Картинка на диаграмата)](#диаграма)

5. [Преобразуване от E\R модел към релационен модел](#преобразуване)

6. [Описание на нормализацията на БД](#нормализация)

7. [Картинка на релационния модел от Data Studio](#картинка)

8. Разпечетване на скриптовете:

8.1. [Create](#create)

8.2. [Alter table](#alter)

8.3. [Select](#select)

Описание на проекта

Проекта, който Ви представяме, представлява създаване на база от данни за издателска къща. След опознаване на материята на издателската къща и нейната бюрокрация създадохме примерен модел на базата. Установихме, че можем да разделим проекта на 3 основни модула, които са: [Продукция](#продукция) , [Счетоводно-правен отдел](#счетоводство) и [Търговски отдел](#търговия).

Дейността, която се извършва от нашата издателска къща, е редактиране, външно оформление и издаване на художествена литература, като изпечатването на книгите се извършва от печатница/ци, която/които имат сключен договор за услуга с издателската къща.

Описание на множествата същности

Contract

Всеки един човек или фирма, които обменят ресурси с нашата фирма, трябва да имат подписан договор.

Атрибутите на Contract са:

**Number** – идентификационен номер, който се получава от подредбата на вписване на договорите.

**start date** - дата на влизане в сила.

**date expired** – краен срок.

**description** – кратко описание.

**type** – тип на договора - граждански, трудов, за услуга и т. н.

Employee

В множеството същности Employee се включват всички преки служители на издателската къща, т.е. тези, които имат подписан трудов догор.

Атрибутите са:

**EGN** – ключ за множеството същности.

**Name** – трите имена на служителя

**Salary –** сътветната стойност на заплатата договорена при подписването на договора или при обновяване на заплатите

**Telephone –** личен или служебен телефон на служителя

Partner

Partner представлява всички юридически и физически лица, които не са част от служебната структура на фирмата.

**Name**: Името на съответното физическо лице, с което сме подписали договор или името на фирмата, с която сме сключили договор

**Type** : По стойностите на атрибута се разбира дали партньора е физическо или юридическо лице.

**Address**: Съответния адрес на цетралния офис на дадена фирма, независимо дали сме сключили договор със съответно лице от фирмата или със самата фирма.

Book (ISBN, Title, Author, NumPages, OriginalLanguage)

В тази таблица се съхранява информацията за всяка една книга, по която издателската къща ще работи и ще дава на определени печатници за изготвянена копия.Релацията съдържа следните атрибути:

**ISBN** – първичният ключ в таблицата.Чрез този атрибут в таблицата се съхранява информация за международния International Book Standard Number, определящ всяка книга, чиято дължина може да бъде 10 или 13 символа, които в повечето случаи са цифри(с изключение на тези номера, където последният символ е 'X', значещ 10)

**Title** - съдържа информация за името на конкретна книгат

**Author** - съдържа информация за автора на книгата(в случаите на съавторство в него се записват имената на всички участващи писатели)

**NumPages** - съдържа общия брой страници, от които ще се състои съответното издание

**OriginalLanguage** - Съдържа информация за оригиналният език, на който е написана съответната книга

PrintingHouse(Name, Address, Contract\_Number)

В тази таблица се съхранява информация за печатниците, с които издателската къща сключва договори за отпечатването на определен брой копия на дадена книга.Релацията съдържа следните атрибути:

**Name** – първичният ключ в таблицата, чрез който се съхранява информация за името на определена печатница

**Address** – съдържа информация за адреса на конкретната печатница, на която ще се предостави книгата за отпечатване

**Contract\_Number** – вторичен ключ, референциращ атрибута *Number* от релацията *Contract*, който описва уникалния номер на договора, чрез който се сключва споразумение между печатницата и издателската къща за съвместна работа

Orders(DateOrdered, Deadline, Book\_ISBN, Partner\_name, NumberOrdered)

Същността Orders е описание на поръчките към логистичен център, направени от даден партньор. Същността е слаба, тъй като не може да бъде определена еднозначно от атрибутите си дата на поръчка (dateordered) и краен срок за изпълнение на поръчката (deadline). Затова при преобразуването в релационния модел тя взима като външни ключове името на партньора, който е заявил дадена книга, и ключа на релацията Book – ISBN. Тоест за Orders има две поддържащи същности(Partner и Book). В релацията Orders също се включват и атрибутите за заявената бройка книги и краен срок, които са функционално определени от първичния ключ.

Asset

В Assets се попълват данните за всеки актив.([виж стр. 6](#счетоводство))

Атрибутите на Asset са:

**Number** - инвентарен номер

**Description** – описани на актива

**Year** – година на въвеждане в експлоатация

**Origin** – описание на произхода

**Initial value** – първоначална стойност

**Annual depreciation** – амортизационна норма (%)

Bill

В книгите за приходи и разходи се водят счетоводни записи, в който се вписват номера на всички първични счетоводни документи, на които съответства същността Bill. Те се издават при извършване на разплащане или друг тип счетоводна дейност.

Чрез тях може да се установят приходите и разходите.

Атрибутите на Bill:

**Number** – всеки вид счетоводни документи се номерират отделно.

**Type** - вид на счетоводния документ.

**Description**- за какво е издаден дадения документ.

**Money** – парите, които сумата, която постъпва в касата или разплащателната сметка или съответно какви разходи се изплащат от тях.

**Date** – дата на издаване на първичния документ.

Logistic centre

Същността Logistic centre има два атрибута - име на логистичния център и неговият ядрес.

Описание на модула „Продукция“

Чрез този модул се представя целият процес по подготвянето на определена книга в издателската къща за печатане, като се включват и дейностите по редактиране, превод от чужд език и др.

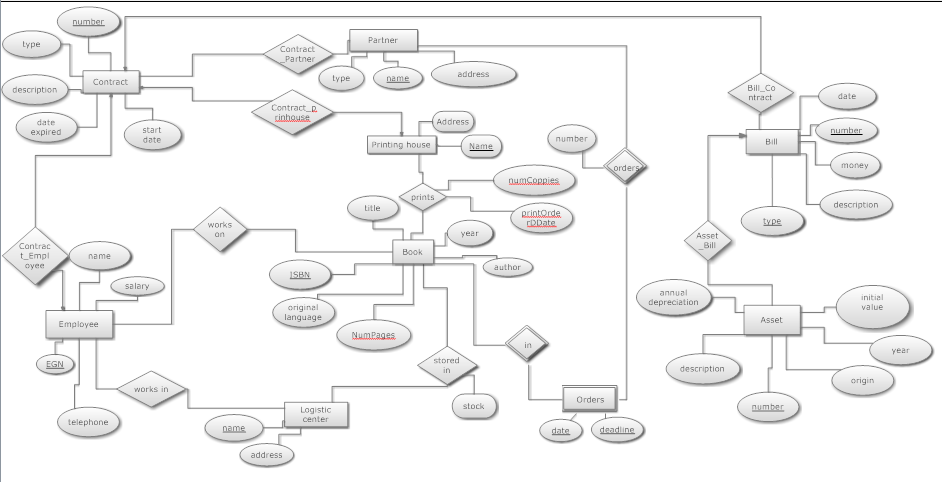
Описание на модула „Търговски отдел“

Този модул представя продажбата на продукцията на издателската къща. Нашата издателска къща има няколко логистични центъра, в които се съхраняват наличните книги. Отдела се грижи за изпънението на всички поръчки направени от нашите партньори.

Описание на модула „Счетоводно-правен отдел“

Счетоводно-правния модул е представен от множествата същности Assest, Bill, Contract, Partner. Те включват най-необходимите за организирането на счетоводство данни при водене на едностранно счетоводство. По закон се предвижда при една фирма да се използва книга за приходите и за разходите, инвентарна книга и други спомагателни. В инвентарната книга счетоводителят вписва всеки дълготраен актив и се отбелязва годишната амортизационна норма.

E/R диаграма на модела на базата данни



Преобразуване от E\R модел към релационен модел

Модул „Продукция“

Book (ISBN, Title, Author, NumPages, OriginalLanguage)

WorkingOn(BookISBN, EmpEGN) – релация за връзката м/у множеството същности Employee и Book

PrintingHouse(Name, Address, Contract\_Number)

Printed (PrHouseName, BISBN, PrintOrderDate, NumCoppies) – релация за връзката м/у множествата същности PrintingHouse и Book

Тук не е нужно да премахваме релации, т.к. връзките м/у същностите са М:М

Модул „Счедоводно-правен отдел“

CONTRACT(NUMBER, TYPE, DESCRIPTION, START\_DATE, END\_DATE);  
PARTNER (NAME, TYPE, ADDRESS);  
BILL (TYPE, NUMBER,DATE, MONEY,  DESCRIPTION, CONTRACT\_NUMBER);  
EMPLOYEE (EGN, NAME, SALARY, TELEPHONE, CONTRACT\_NUMBER, IBAN);  
PARTNER\_CONTRACT(CONTRACT\_NUMBER, PARTNER\_NAME);  
ASSET (YEAR, ORIGIN, INITIAL VALUE, ANNUAL\_DEPRECIATION, NUMBER);  
BILL\_CONTRACT(CONTRACT \_ NUMBER, BILL\_ TYPE, BILL\_ NUMBER);

CONTRACT\_ EMPLOYEE (CONTRACT\_NUMBER, EMPLOYEE\_EGN);

ASSET \_ BILL(BILL TYPE, BILL NUMBER, ASSET NUMBER);

Тук ще премахнем някой релации, а именно тези, които представляват ралции на връзки м/у множество същности М:1 или 1:1, т.к. е достатъчно да добавим ключовите атрибути на множеството към което връзката е 1 към другото множество (когато връзката е 1:1 ние избираме към кое релацията на кое множество същности да добавим атрибитите на дугото, т.е. кое ще ни е дете и кое родител). След преобразуване получаваме следните релации:

CONTRACT(NUMBER, TYPE, DESCRIPTION, START\_DATE, END\_DATE);  
PARTNER (NAME, TYPE, ADDRESS);  
BILL (TYPE, NUMBER,DATE, MONEY,  DESCRIPTION, CONTRACT\_NUMBER);  
EMPLOYEE (EGN, NAME, SALARY, TELEPHONE, CONTRACT\_NUMBER, IBAN);  
PARTNER\_CONTRACT(CONTRACT\_NUMBER, PARTNER\_NAME);  
ASSET (YEAR, ORIGIN, INITIAL VALUE, ANNUAL\_DEPRECIATION, NUMBER)

Модул „Търговски отдел“

WorkinIn (Employee\_EGN, LogisticCentre\_name)

LogisticCentre (Name, Address)

StoredIn (LogisticCentre\_name, Book\_ISBN, stock)

Order (Book\_ISBN, Order\_date, Order\_deadline, Order\_number, Partner\_name)

Тук при преминаването от множеството същности Order към релационнен модел не създаваме връзките към Book и Partner, т.к. имаме слабо множество същности, което се поддържа и от двете други множества.

Тук всички връзки са ни М:М и затова нямаме възможност да премахмен някоя от релациите.

Описание на нормализацията на базата данни

CONTRACT(NUMBER, TYPE, DESCRIPTION, START\_DATE, END\_DATE);  
  
В релацията contract може да бъде забелязана функционалната зависимост  
number -> type, description, start\_date, end\_date.  
Както се вижда номерът на договора определя функционално всички останали  
негови атрибути.  
Понеже number определя функционално всички атрибути на contract, то  
number е ключ на релацията (тъй като е и минималното подмножество на атрибутите на contract с това свойство), а от там и суперключ. Няма друг атрибут, който да може да определя някой от останалите. От тук следва, че във всеки елемент от обвивката на функционалните зависимости, в лявата страна трябва да присъства number. От тук следва, че contract е в BCNF.

PARTNER (NAME, TYPE, ADDRESS);

Единственият атрибут на релацията partner, който притежава свойствата на ключ е name. Никой от останалите атрибути не може да присъства самостоятелно в лявата част на някоя от функционалните зависимости (освен случая, в който даден атрибут определя функционално себе си). Следователно лявата част ще бъде ключ, а от там и суперключ. От всичко това следва, че релацията е в BCNF.

BILL (BILL\_TYPE, BILL\_NUMBER,DATE, MONEY,  DESCRIPTION, CONTRACT\_NUMBER);  
  
Аналогично BILL е в BCNF, но с разликата, че тук ключът е съставен от два атрибута  
(BILL\_TYPE, BILL\_NUMBER). Както се вижда никой от атрибутите, различни от BILL\_TYPE, BILL\_NUMBER, не е в състояние да определя някой от останалите. От тук следва, че във всяка функционална зависимост лявата страна ще бъде ключ и суперключ (съставен от атрибутите BILL\_TYPE и BILL\_NUMBER), следователно релацията е в BCNF.  
  
EMPLOYEE (EGN, NAME, SALARY, TELEPHONE, CONTRACT\_NUMBER, IBAN);  
  
Нормализацията на EMPLOYEE стои по малко по - сложен начин. Тук очевидно EGN определя функционално всички останали атрибути, следвователно е ключ (следователно и суперключ). Но друга функционална зависимост, която може да бъде намерена е IBAN -> NAME (както и CONTRACT\_NUMBER -> NAME). Тук IBAN и CONTRACT\_NUMBER не са част от ключа, откъдето е нарушена BCNF и 3 НФ. Тъй като най-подходящото място за съхранение на IBAN и връзката с договора е таблицата съхраняваща самия служител преценихме, че нормализация до 2 НФ е достатъчна.

PARTNER\_CONTRACT(CONTRACT\_NUMBER, PARTNER\_NAME);  
  
Очевидно релацията е в BCNF поради наличието на само два атрибута, които образуват и първичният ключ.

ASSET (YEAR, ORIGIN, INITIAL VALUE, ANNUAL\_DEPRECIATION, NUMBER)

Във всяка функционална зависимост, от дадената релация, принадлежаща на обвививката от функционални зависимости, в лявата страна трябва да присъства NUMBER, тъй като няма друг атрибути, в състояние да пределят кой да е друг атрибут освен себе си. Тъй като NUMBER е ключ (понеже определя всички останали атрибути и няма негово подмножество, което да има същите свойства) той е и суперключ. От казаното следва, че релацията е в BCNF.

Orders(DateOrdered, Deadline, Book\_ISBN, Partner\_name, NumberOrdered)

В релацията Orders всеки атрибут има атомарно значение (първа нормална форма), но неключовият атрибут NumberOrdered зависи от подмножество на зададения клюя. Правим декомпозиция, за да премахнем атрибура Deadline, който не е необходим в ключа.След нея, неключовите атрибути са в пълна функционална зависимост от ключа, следователно релацията е във втора нормална форма. Също така лявата страна е суперключ, затова тя е и в трета нормална форма.

Следните заявки са приложени с цел отърваване от атрибута, който нарушава условията за втора нормална форма при релацията Orders.

ALTER TABLE Orders

DROP CONSTRAINT pk\_Order;

ALTER TABLE Orders

ADD CONSTRAINT pk\_Order PRIMARY KEY(DateOrdered, Partner\_name, Book\_ISBN);

Logistic\_Centre( Name, Address)

Релацията Logistic\_Centre е нормализирана, тъй като всяка релация с два атрибута е в BCNF.

Еmployee\_LogisticCentre(Employee\_EGN, Logisticcentre\_Name)

Също е в BCNF (поради горе описаната причина).

LogisticCentre\_Book( Logisticcentre\_Name, Book\_ISBN, Stock)

Релацията LogisticCentre\_Book е във втора нормална форма защото данните в нея са с атомарно значение и единствения неключов атрибут stock е в пълна функционална зависимост от ключа на релацията.

Book (ISBN, Title, Author, NumPages, OriginalLanguage)

За тази релация могат да бъдат открити следните нетривиални функционални зависимости:

1. ISBN -> Title, Author, NumPages, OriginalLanguage
2. Title, Author -> OriginalLanguage

На основата на откритите нетривиални функционални зависимости за релацията Book могат да бъдат направени следните заключения:

1.Тя е в 1NF, поради отсъствието на атрибути с неатомарно значение.

2.Тя е във 2NF, защото е в 1NF и всеки неин неключов атрибут се определя от целия ключ, а не от някое негово подмножество(в конкретния случай такова подмножество може да бъде единствено празното множество).

3.Тя не е в 3NF, поради откритата функционална зависимост 2), чиято лява част не е суперключ и чиято дясна част не е първичен атрибут.

4.Тя не е в BCNF, защото не е в 3NF.

WorkingOn(BookISBN, EmpEGN)

За тази релация не могат да бъдат открити нетривиални зависимости, поради наличието на 2 атрибута, и двата от които участват във формирането на първичен ключ.На основата на този факт могат да бъдат направени следните заключения:

1.Тя е в 1NF, поради отсъствието на атрибути с неатомарно значение.

2.Тя е във 2NF, защото е в 1NF и в нея не могат да бъдат открити нетривиални функционални зависимости, което гарантира липсата на атрибути, които да се определят функционално от подмножество на ключа.

3.Тя е в 3NF, защото е в 1NF, 2NF и в нея не могат да бъдат открити нетривиални функционални зависимости, което гарантира отсъствието на функционални зависимости, в които лявата част не е суперключ и дясната е първичен атрибут.

4.Тя е в BCNF, защото е в 1NF, 2NF, 3NF и не могат да бъдат открити нетривиални функционални зависимости, което гарантира изпълнението на условието на BCNF.

PrintingHouse(Name, Address, Contract\_Number)

За тази релация могат да бъдат открити следните нетривиални функционални зависимости:

1. Name -> Address, Contract\_Number
2. Contract\_Number -> Name, Address

На основата на откритите нетривиални функционални зависимости за релацията Book могат да бъдат направени следните заключения:

1.Тя е в 1NF, поради отсъствието на атрибути с неатомарно значение.

2.Тя е във 2NF, защото е в 1NF и всеки неин неключов атрибут може да се определя от целия ключ, а не от някое негово подмножество(в конкретния случай такова подмножество може да бъде единствено празното множество).

3.Тя не е в 3NF, поради откритата функционална зависимост 2), чиято лява част не е суперключ и чиято дясна част не е първичен атрибут, което нарушава условието на 3NF.

4.Тя не е в BCNF, защото не е в 3NF.

Printed (PrHouseName, BISBN, PrintOrderDate, NumCoppies)

За тази релация могат да бъдат открити следните нетривиални функционални зависимости:

1. PrHouseName, BISBN, PrintOrderDate -> NumCoppies

На основата на откритите нетривиални функционални зависимости за релацията Book могат да бъдат направени следните заключения:

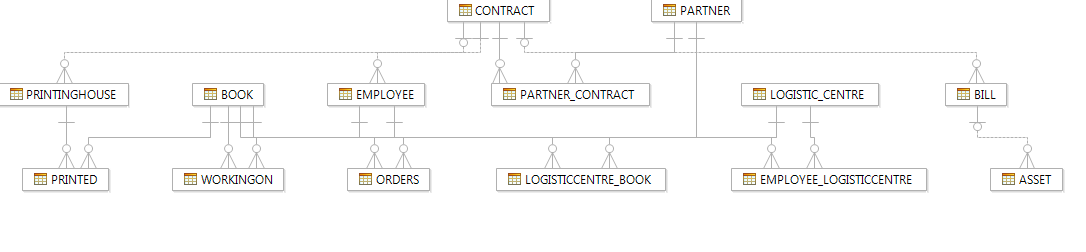
1.Тя е в 1NF, поради отсъствието на атрибути с неатомарно значение.

2.Тя е във 2NF, защото е в 1NF и всеки неин неключов атрибут може да се определя от целия ключ, а не от някое негово подмножество.

3.Тя е в 3NF, защото е в 1NF, 2NF и в нея не могат да бъдат открити функционални зависимости, в които лявата част не е суперключ и дясната е първичен атрибут.

4.Тя е в BCNF, защото е в 1NF, 2NF, 3NF и може да бъде само нетривиалната функционална зависимост 1), чиято нейна лява страна е ключ(следователно и суперключ).

Картинка на релационния модел от Data Studio



**Create заявки**

CREATE SCHEMA PUBLISHINGHOUSE!

CREATE TABLE LOGISTIC\_CENTRE (

NAME VARCHAR(50) NOT NULL,

ADDRESS VARCHAR(150)

)

CREATE TABLE PARTNER\_CONTRACT (

CONTRACT\_NUMBER INTEGER NOT NULL,

PARTNER\_NAME VARCHAR(50) NOT NULL

)

CREATE TABLE ORDERS (

DATEORDERED DATE DEFAULT CURRENT DATE NOT NULL,

DEADLINE DATE NOT NULL,

PARTNER\_NAME VARCHAR(50) NOT NULL,

BOOK\_ISBN VARCHAR(13) NOT NULL,

NUMBERORDERED INTEGER NOT NULL

)

CREATE TABLE WORKINGON (

BOOKISBN VARCHAR(13) NOT NULL,

EMPEGN CHAR(10) NOT NULL

)

CREATE TABLE LOGISTICCENTRE\_BOOK (

LOGISTICCENTRE\_NAME VARCHAR(50) NOT NULL,

BOOK\_ISBN VARCHAR(13) NOT NULL,

STOCK INTEGER

)

CREATE TABLE CONTRACT (

TYPE VARCHAR(50) NOT NULL,

NUMBER INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY ( START WITH 1000 INCREMENT BY 1 MINVALUE 1000 MAXVALUE 2147483647 NO CYCLE CACHE 20),

DESCRIPTION VARCHAR(2000) NOT NULL,

START\_DATE DATE DEFAULT CURRENT DATE NOT NULL,

END\_DATE DATE DEFAULT NULL

)

CREATE TABLE BILL (

NUMBER INTEGER NOT NULL,

TYPE VARCHAR(60) NOT NULL,

DESCRIPTION VARCHAR(200),

MONEY DOUBLE NOT NULL,

DATE DATE NOT NULL,

CONTRACT\_NUMBER INTEGER

)

CREATE TABLE EMPLOYEE\_LOGISTICCENTRE (

EMPLOYEE\_EGN CHAR(10) NOT NULL,

LOGISTICCENTRE\_NAME VARCHAR(50) NOT NULL

)

CREATE TABLE ASSET (

NUMBER INTEGER NOT NULL,

DOCUMENT\_TYPE VARCHAR(40),

DOCUMENT\_NUMBER INTEGER,

ORIGIN VARCHAR(40) NOT NULL,

YEAR INTEGER NOT NULL,

INIT\_VALUE DECIMAL(5 , 0) NOT NULL,

ANNUAL\_DEPRECIATION DECIMAL(5 , 0) NOT NULL,

DESCRIPTION VARCHAR(2000) NOT NULL

)

CREATE TABLE PARTNER (

NAME VARCHAR(50) NOT NULL,

TYPE VARCHAR(50) NOT NULL,

ADDRESS VARCHAR(100) NOT NULL

)

CREATE TABLE BOOK (

ISBN VARCHAR(13) NOT NULL,

TITLE VARCHAR(96) NOT NULL,

AUTHOR VARCHAR(49) NOT NULL,

NUMPAGES INTEGER NOT NULL,

ORIGINALLANGUAGE VARCHAR(25) NOT NULL

)

CREATE TABLE PRINTED (

PRHOUSENAME VARCHAR(150) NOT NULL,

BISBN VARCHAR(13) NOT NULL,

PRINTORDERDATE DATE NOT NULL,

NUMCOPPIES INTEGER NOT NULL

)

CREATE TABLE PRINTINGHOUSE (

NAME CHAR(150) NOT NULL,

ADDRESS VARCHAR(150) NOT NULL,

CONTRACT\_NUMBER INTEGER

)

CREATE TABLE EMPLOYEE (

EGN CHAR(10) NOT NULL,

NAME VARCHAR(50) NOT NULL,

SALARY DECIMAL(5 , 0),

TELEPHONE VARCHAR(15) NOT NULL,

CONTRACT\_NUMBER INTEGER NOT NULL,

IBAN VARCHAR(35) NOT NULL

)

**ALTER TABLE заявки**

ALTER TABLE LOGISTIC\_CENTRE ADD CONSTRAINT PK\_LOGCEN PRIMARY KEY

(NAME)!

ALTER TABLE PARTNER\_CONTRACT ADD CONSTRAINT PARTNER\_CONTRACT\_PK PRIMARY KEY

(CONTRACT\_NUMBER,

PARTNER\_NAME)!

ALTER TABLE ORDERS ADD CONSTRAINT PK\_ORDER PRIMARY KEY

(DATEORDERED,

PARTNER\_NAME,

BOOK\_ISBN)!

ALTER TABLE ORDERS ADD CONSTRAINT SQL110603083737880 CHECK ( LENGTH(BOOK\_ISBN)=10 OR LENGTH(BOOK\_ISBN)=13 )!

ALTER TABLE ORDERS ADD CONSTRAINT SQL110603083737930 CHECK (NUMBERORDERED>1)!

ALTER TABLE WORKINGON ADD CONSTRAINT SQL110603151127840 PRIMARY KEY

(BOOKISBN,

EMPEGN)!

ALTER TABLE LOGISTICCENTRE\_BOOK ADD CONSTRAINT SQL110603151201700 PRIMARY KEY

(LOGISTICCENTRE\_NAME,

BOOK\_ISBN)!

ALTER TABLE CONTRACT ADD CONSTRAINT CONTRACT\_PK PRIMARY KEY

(NUMBER)!

ALTER TABLE BILL ADD CONSTRAINT PK\_BL PRIMARY KEY

(NUMBER,

TYPE)!

ALTER TABLE BILL ADD CONSTRAINT SQL110603152033890 CHECK (TYPE IN ('Приходно банково бордеро','Разходно банково бордеро','Приходен касов ордер ','Разходен касов ордер'))!

ALTER TABLE EMPLOYEE\_LOGISTICCENTRE ADD CONSTRAINT SQL110603152046000 PRIMARY KEY

(EMPLOYEE\_EGN,

LOGISTICCENTRE\_NAME)!

ALTER TABLE ASSET ADD CONSTRAINT ASSET\_PK PRIMARY KEY

(NUMBER)!

ALTER TABLE ASSET ADD CONSTRAINT PERCENT CHECK (ANNUAL\_DEPRECIATION BETWEEN 0 AND 100)!

ALTER TABLE ASSET ADD CONSTRAINT POSITIVE\_NUMBER CHECK (DOCUMENT\_NUMBER > 0)!

ALTER TABLE ASSET ADD CONSTRAINT YEAR\_CHK CHECK (YEAR>2000)!

ALTER TABLE PARTNER ADD CONSTRAINT PARTNER\_PK PRIMARY KEY

(NAME)!

ALTER TABLE PARTNER ADD CONSTRAINT CHECK\_TYPE CHECK (TYPE IN (

'Юридическо лице','Физическо лице'));

ALTER TABLE BOOK ADD CONSTRAINT SQL110531085214180 PRIMARY KEY

(ISBN)!

ALTER TABLE BOOK ADD CONSTRAINT SQL110531085214210 CHECK (LENGTH(ISBN)=10 OR LENGTH(ISBN)=13)!

ALTER TABLE BOOK ADD CONSTRAINT SQL110531085214230 CHECK (NUMPAGES>0)!

ALTER TABLE PRINTED ADD CONSTRAINT SQL110531085439440 PRIMARY KEY

(PRHOUSENAME,

BISBN,

PRINTORDERDATE)

ALTER TABLE PRINTED ADD CONSTRAINT SQL110531085439510 CHECK (YEAR(PRINTORDERDATE)>1990)!

ALTER TABLE PRINTED ADD CONSTRAINT SQL110531085439530 CHECK (NUMCOPPIES>=50)!

ALTER TABLE DOG ADD CONSTRAINT SQL110602102901810 PRIMARY KEY

(ID)!

ALTER TABLE PRINTINGHOUSE ADD CONSTRAINT SQL110602073847310 PRIMARY KEY

(NAME)!

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE\_PK PRIMARY KEY

(EGN)!

ALTER TABLE PARTNER\_CONTRACT ADD CONSTRAINT CONTRACT\_FK FOREIGN KEY

(CONTRACT\_NUMBER)

REFERENCES CONTRACT

(NUMBER)!

ALTER TABLE PARTNER\_CONTRACT ADD CONSTRAINT PARTNER\_FK FOREIGN KEY

(PARTNER\_NAME)

REFERENCES PARTNER

(NAME)!

ALTER TABLE ORDERS ADD CONSTRAINT FK\_BOOK FOREIGN KEY

(BOOK\_ISBN)

REFERENCES BOOK

(ISBN)!

ALTER TABLE ORDERS ADD CONSTRAINT FK\_PARTNERSHIP FOREIGN KEY

(PARTNER\_NAME)

REFERENCES PARTNER

(NAME)!

ALTER TABLE WORKINGON ADD CONSTRAINT SQL110603151127860 FOREIGN KEY

(BOOKISBN)

REFERENCES BOOK

(ISBN)!

ALTER TABLE WORKINGON ADD CONSTRAINT SQL110603151127910 FOREIGN KEY

(EMPEGN)

REFERENCES EMPLOYEE

(EGN)!

ALTER TABLE LOGISTICCENTRE\_BOOK ADD CONSTRAINT SQL110603151201730 FOREIGN KEY

(LOGISTICCENTRE\_NAME)

REFERENCES LOGISTIC\_CENTRE

(NAME)!

ALTER TABLE LOGISTICCENTRE\_BOOK ADD CONSTRAINT SQL110603151201770 FOREIGN KEY

(BOOK\_ISBN)

REFERENCES BOOK

(ISBN)!

ALTER TABLE BILL ADD CONSTRAINT FK\_COREFERENCES FOREIGN KEY

(CONTRACT\_NUMBER)

REFERENCES CONTRACT

(NUMBER)!

ALTER TABLE EMPLOYEE\_LOGISTICCENTRE ADD CONSTRAINT SQL110603152046040 FOREIGN KEY

(EMPLOYEE\_EGN)

REFERENCES EMPLOYEE

(EGN)!

ALTER TABLE EMPLOYEE\_LOGISTICCENTRE ADD CONSTRAINT SQL110603152046070 FOREIGN KEY

(LOGISTICCENTRE\_NAME)

REFERENCES LOGISTIC\_CENTRE

(NAME)!

ALTER TABLE ASSET ADD CONSTRAINT BILL\_FK FOREIGN KEY

(DOCUMENT\_NUMBER,

DOCUMENT\_TYPE)

REFERENCES BILL

(NUMBER,

TYPE)!

ALTER TABLE PRINTED ADD CONSTRAINT FK\_PRHN FOREIGN KEY

(PRHOUSENAME)

REFERENCES PRINTINGHOUSE

(NAME)!

ALTER TABLE PRINTED ADD CONSTRAINT SQL110531085439490 FOREIGN KEY

(BISBN)

REFERENCES BOOK

(ISBN)!

ALTER TABLE PRINTINGHOUSE ADD CONSTRAINT CONTRACT\_FK FOREIGN KEY

(CONTRACT\_NUMBER)

REFERENCES CONTRACT

(NUMBER)!

ALTER TABLE EMPLOYEE ADD CONSTRAINT CONTRACT\_FK FOREIGN KEY

(CONTRACT\_NUMBER)

REFERENCES CONTRACT

(NUMBER)!

**Select заявки**

--Всички служители с актуални договори

SELECT NAME FROM EMPLOYEE

JOIN CONTRACT ON CONTRACT.NUMBER = EMPLOYEE.CONTRACT\_NUMBER

WHERE CONTRACT.END\_DATE > CURRENT\_DATE;

--Всички киниги от бивши служители (с изтекъл договор)

SELECT B1.TITLE

FROM BOOK AS B1

JOIN WORKINGON AS W1 ON W1.BOOKISBN = B1.ISBN

JOIN EMPLOYEE AS E1 ON E1.EGN = W1.EMPEGN

JOIN CONTRACT AS C1 ON C1.NUMBER = E1.CONTRACT\_NUMBER

WHERE C1.END\_DATE < CURRENT\_DATE;

--Имената на книжарниците, които са сключили договор с издаделската къща

SELECT O1.PARTNER\_NAME

FROM ORDERS AS O1

JOIN PARTNER AS P1 ON P1.NAME = O1.PARTNER\_NAME

JOIN PARTNER\_CONTRACT AS PC1 ON PC1.PARTNER\_NAME = P1.NAME

GROUP BY O1.PARTNER\_NAME;

--Името на книжарницата и броят на книгите, които е поръчала от издателската къща

SELECT O1.PARTNER\_NAME, SUM(O1.NUMBERORDERED)

FROM ORDERS AS O1

JOIN PARTNER AS P1 ON P1.NAME = O1.PARTNER\_NAME

JOIN PARTNER\_CONTRACT AS PC1 ON PC1.PARTNER\_NAME = P1.NAME

GROUP BY O1.PARTNER\_NAME;

--Имената на фирмите и сумите които издателството им е изплатило

SELECT NAME, SUM(MONEY)

FROM PARTNER

JOIN PARTNER\_CONTRACT PC

ON NAME = PARTNER\_NAME

JOIN CONTRACT CN

ON CN.NUMBER = PC.CONTRACT\_NUMBER

JOIN BILL

ON CN.NUMBER = BILL.CONTRACT\_NUMBER

WHERE PARTNER.TYPE LIKE 'Юридическо лице' and BILL.TYPE LIKE 'Разход%'

GROUP BY NAME;

--Извежда всички партньори, които са правили заявка за повече от 7 бройки от дадена книга--

select Partner\_name, count(Book\_ISBN)

from Orders

where Partner\_name in ( select Partner\_name from Orders

group by Partner\_name

having count(Book\_ISBN)>7 )

group by Partner\_name;