Міністерство освіти і науки

Кіровоградський державний педагогічний університет

імені Володимира Винниченка

Звіт виконаної роботи

з предмету : «Розподілені інформаційно-аналітичні системи»

на тему: «Інформаційна система бібліотечного фонду міста»

студенток 46 групи

фізико-математичного факультету

Коршунової Ксенії

Мельніченко Дар‘ї

Кропивницький 2017

**Предметна область**

Бібліотечний фонд міста складають бібліотеки, розташовані на території міста. Кожна бібліотека включає в себе абонементи і читальні зали. Користувачами бібліотек є різні категорії читачів: студенти, науковці, викладачі, школярі, робітники, пенсіонери та інші жителі міста. Кожна категорія читачів може володіти непересічними характеристиками-атрибутами: для студентів це назва навчального закладу, факультет, курс, номер групи, для науковця - назва організації, наукова тема і т.д. Кожен читач, будучи зареєстрованим в одній з бібліотек, має доступ до всього бібліотечного фонду міста.

Бібліотечний фонд (книги, журнали, газети, збірники статей, збірки віршів, дисертації, реферати, збірники доповідей та тез доповідей і ін.) Розміщений в залах-сховищах різних бібліотек на певних місцях зберігання (номер залу, стелажа, полки) і ідентифікується номенклатурними номерами. При цьому існують різні правила щодо тих чи інших видань: якісь підлягають тільки читанню в читальних залах бібліотек, для тих, що видаються, може бути встановлений різний термін видачі і т.д. З одного боку, бібліотечний фонд може поповнюватися, з іншого, - з плином часу відбувається його списання.

Твори авторів, що складають бібліотечний фонд, також можна розділити на різні категорії, які характеризуються власним набором атрибутів: підручники, повісті, романи, статті, вірші, дисертації, реферати, тези доповідей і т.д.

Співробітники бібліотеки, що працюють в різних залах різних бібліотек, ведуть облік читачів, а також облік розміщення та видачі літератури.

***У БД повинна зберігатися інформація про:***

*Читачі* – код читача, ПІП, дата народження, домашня адреса, номер телефону, код бібліотеки.

*Студент* – код студента, код читача, навчальний заклад, факультет, курс, група.

*Викладач* – код викладача, код читача, робоче місце, наукова ступінь.

*Пенсіонер* – код пенсіонера, код читача, пільги.

*Школяр* – код школяра, код читача, школа, клас.

*Безробітній* – код безробітного, код читача.

*Робітник* – код робітника, код читача, робоче місце,

*Вчений* – код вченого, код читача, місце роботи, наукова степінь.

*Бібліотека* – код бібліотеки, назва, адреса, номер телефону.

*Бібліотекар* – код робітника, ПІП, місце роботи, код бібліотеки.

*Книга* – код книги, код автора книги, назва книги, рік, кількість сторінок, жанр, код місця зберігання, код бібліотеки.

*Автор книги* – код автора книги, ПІП, додатковий автор.

*Місце зберігання* – код місця зберігання, номер залу, номер стелажу, номер полиці, код бібліотеки.

*Надходження* *книг –* код надходження, дата першого надходження, дата останнього надходження, код книги.

*Видача* *книг* – код видачі, код книги, дата видачі книги, дата повернення книги, видана книга(так/ні).

***При проектуванні слід врахувати наступне:***

Одна книга може мати багато авторів, а автор може мати багато книг.

Одна поставка книг може мати багато книг, а одна книга надходить тільки під час одної поставки.

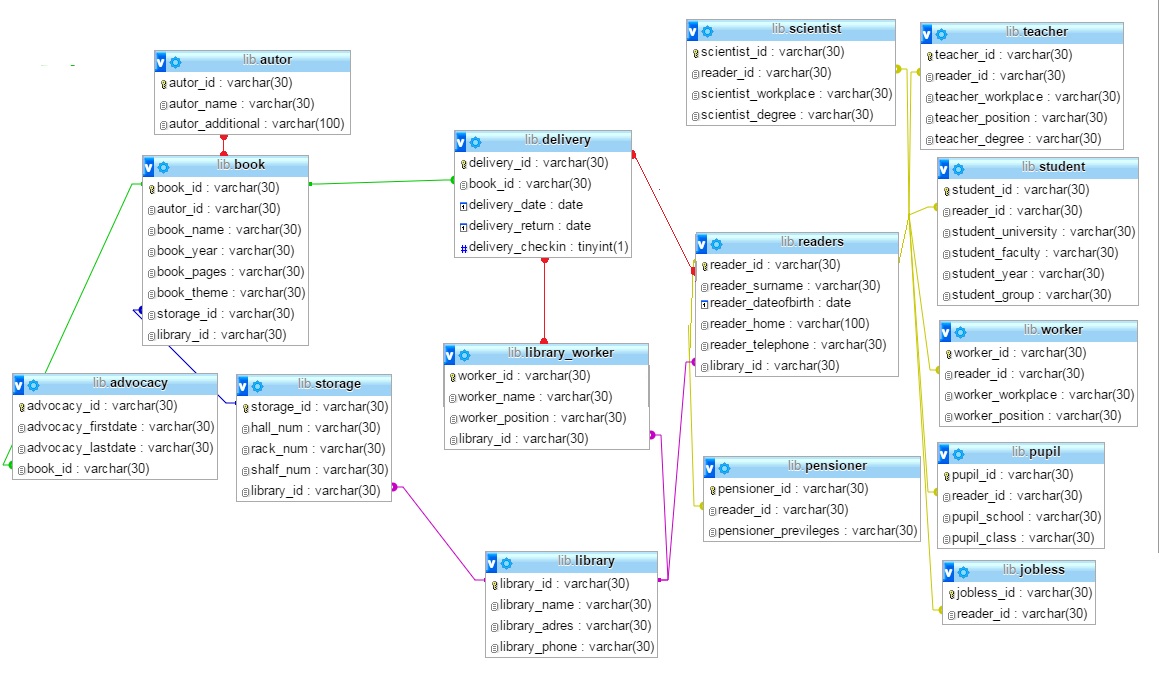
Одна книга може видаватися різним читачам, один читач може брати багато книг.

Одна книга може бути видана багатьма працівниками, один працівник може видати багато книг.

***Крім того слід врахувати:***

Читач бібліотеки може мати свої індивідуальні характеристики – може бути студентом, працівником, науковцем, пенсіонером, безробітним і т.п..

**Логічна модель**

******

**Процес нормалізації**

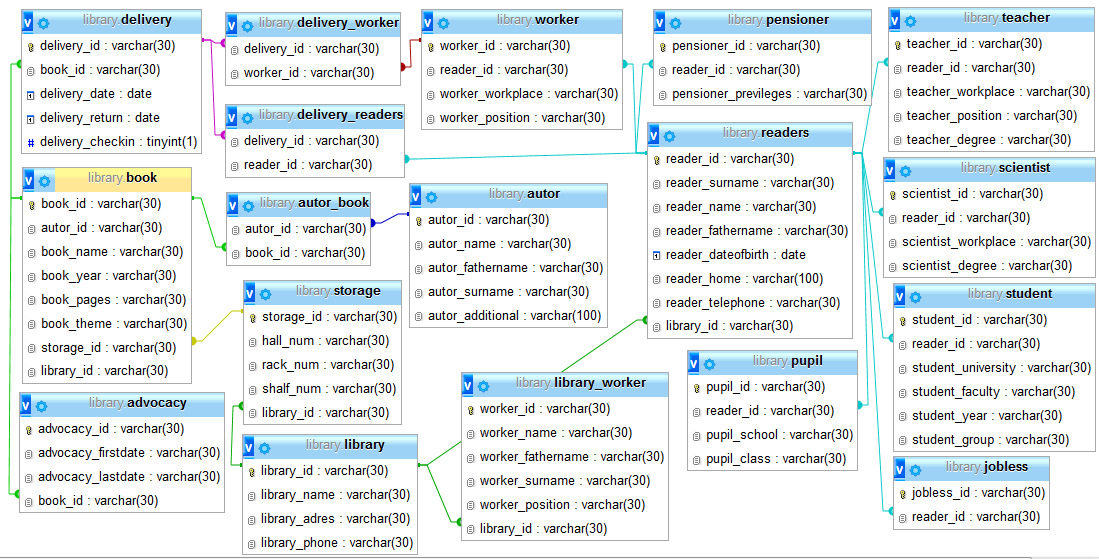
1 НФ

1. Під час нормалізації виявилось, що не всі атрибути є атомарними. Тому було розбито на атрибути:

* Таблиця читачі: ПІП читача
* Таблиця бібліотекар: ПІП бібліотекаря
* Таблиця авторів: ПІП автора

1. Створено проміжні таблиці:

* Автор книги
* Видача книг працівниками
* Видача книг читачам



**Запити**

1. **Отримати список читачів із заданими характеристиками: студентів вказаного навчального закладу, факультету, наукових робітників з певної тематики і т.д.**

SELECT `student`.`student\_university`, `readers`.`reader\_surname`

FROM

`readers` INNER JOIN `student` ON `readers`.`reader\_id`=`student`.`reader\_id`

WHERE `student`.`student\_university`='Universitas Bengkulu'

UNION

SELECT `scientist`.`scientist\_workplace`, `readers`.`reader\_surname`

FROM

`readers`INNER JOIN `scientist` ON `readers`.`reader\_id`=`scientist`.`reader\_id`

WHERE `scientist`.`scientist\_workplace`='Services'

1. **Видати перелік читачів, на руках у яких перебуває зазначений твір.**

SELECT `readers`.`reader\_surname`

from

`book` INNER JOIN (

`delivery`INNER JOIN (

`delivery\_readers` INNER JOIN `readers` on `delivery\_readers`.`reader\_id`=`readers`.`reader\_id`)

ON `delivery`.`delivery\_id` = `delivery\_readers`.`delivery\_id`)

ON `book`.`book\_id`=`delivery`.`book\_id`

where (`delivery`.`delivery\_checkin`=1) and (`book`.`book\_name`='Connelly and Sons')

1. **Отримати список читачів, на руках у яких знаходиться вказане видання (книга, журнал тощо).**

SELECT `readers`.`reader\_surname`

from

`book` INNER JOIN (

`delivery`INNER JOIN (

`delivery\_readers` INNER JOIN `readers` on `delivery\_readers`.`reader\_id`=`readers`.`reader\_id`)

ON `delivery`.`delivery\_id` = `delivery\_readers`.`delivery\_id`)

ON `book`.`book\_id`=`delivery`.`book\_id`

where (`delivery`.`delivery\_checkin`=1) and (`book`.`book\_theme`='novel')

1. **Отримати перелік читачів, які протягом зазначеного проміжку часу отримували видання з деяким твором, і назва цього видання.**

SELECT `readers`.`reader\_surname`,`book`.`book\_name`

from

`book` INNER JOIN (

`delivery`INNER JOIN (

`delivery\_readers` INNER JOIN `readers` on `delivery\_readers`.`reader\_id`=`readers`.`reader\_id`)

ON `delivery`.`delivery\_id` = `delivery\_readers`.`delivery\_id`)

ON `book`.`book\_id`=`delivery`.`book\_id`

where TO\_DAYS(NOW()) - TO\_DAYS(`delivery`.`delivery\_date`) <= 90

and (`book`.`book\_theme`='novel')

1. **Видати список видань, які протягом деякого часу отримував зазначений читач з фонду бібліотеки, де він зареєстрований.**

SELECT `book`.`book\_name`

from

`library`INNER JOIN (

`book` INNER JOIN (

`delivery`INNER JOIN (

`delivery\_readers` INNER JOIN `readers` on `delivery\_readers`.`reader\_id`=`readers`.`reader\_id`)

ON `delivery`.`delivery\_id` = `delivery\_readers`.`delivery\_id`)

ON `book`.`book\_id`=`delivery`.`book\_id`)

ON `book`.`library\_id`=`library`.`library\_id`

where TO\_DAYS(NOW()) - TO\_DAYS(`delivery`.`delivery\_date`) <= 90

and `readers`.`library\_id`= `book`.`library\_id`

1. **Отримати перелік видань, якими протягом деякого часу користувався вказаний читач з фонду бібліотеки, де він не зареєстрований.**

SELECT `book`.`book\_name`

from

`library`INNER JOIN (

`book` INNER JOIN (

`delivery`INNER JOIN (

`delivery\_readers` INNER JOIN `readers` on `delivery\_readers`.`reader\_id`=`readers`.`reader\_id`)

ON `delivery`.`delivery\_id` = `delivery\_readers`.`delivery\_id`)

ON `book`.`book\_id`=`delivery`.`book\_id`)

ON `book`.`library\_id`=`library`.`library\_id`

where TO\_DAYS(NOW()) - TO\_DAYS(`delivery`.`delivery\_date`) <= 90

and `readers`.`library\_id`!= `book`.`library\_id`

1. **Отримати список літератури, яка в даний момент видана з певною полки деякої бібліотеки.**

SELECT `book`.`book\_name`

FROM `delivery` INNER JOIN (

`book` INNER JOIN

(`storage` INNER JOIN `library` ON `storage`.`library\_id` = `library`.`library\_id`)

ON `book`.`storage\_id`=`storage`.`storage\_id`)

ON `delivery`.`book\_id`=`book`.`book\_id`

WHERE `delivery`.`delivery\_checkin`= 1

AND TO\_DAYS( NOW( ) ) - TO\_DAYS( `delivery`.`delivery\_date`) <=1

AND `storage`.`shalf\_num`= 2

AND `library`.`library\_name`= 'Threda'

1. **Видати список читачів, які протягом означеного періоду були обслужені зазначеним бібліотекарем.**

SELECT `readers`.`reader\_surname`, `readers`.`reader\_name`

FROM

`library\_worker` INNER JOIN (

`library` INNER JOIN (

`readers` INNER JOIN (

`delivery\_readers` INNER JOIN `delivery` ON `delivery`.`delivery\_id`=`delivery\_readers`.`delivery\_id`)

ON `readers`.`reader\_id`=`delivery\_readers`.`reader\_id`)

ON `library`.`library\_id`=`readers`.`reader\_id`)

ON `library\_worker`.`library\_id`=`library`.`library\_id`

WHERE `library\_worker`.`worker\_surname`='Davis' AND TO\_DAYS( NOW( ) ) - TO\_DAYS( `delivery`.`delivery\_date`) <=14

1. **Отримати дані про відпрацювання бібліотекарів (число обслужених читачів в зазначений період часу).**

SELECT COUNT( delivery\_id )

FROM delivery

WHERE delivery\_date

BETWEEN '2017-04-18'

AND '2017-04-20'

1. **Отримати список читачів з простроченим терміном літератури.**

SELECT r.reader\_surname, r.reader\_name   
FROM readers r   
INNER JOIN delivery\_readers dr   
ON r.reader\_id = dr.reader\_id   
INNER JOIN delivery d   
ON dr.delivery\_id = d.delivery\_id   
WHERE d.delivery\_return < CURDATE()

1. **Отримати перелік зазначеної літератури, яка надійшла (була списана) протягом деякого періоду.**

SELECT `book`.`book\_name`

FROM

`book` INNER JOIN `advocacy` ON `book`.`book\_id`=`advocacy`.`book\_id`

WHERE `advocacy`.`advocacy\_firstdate` BETWEEN '2017-04-01' AND '2017-04-20'

1. **Видати список бібліотекарів, які працюють в зазначеному читальному залі деякої бібліотеки.**

SELECT lw.worker\_name, lw.worker\_surname

FROM library\_worker lw

INNER JOIN library l

ON l.library\_id = lw.library\_id

INNER JOIN storage s

ON s.library\_id = l.library\_id

WHERE s.storage\_id = 1

1. **Отримати список читачів, які не відвідували бібліотеку протягом зазначеного часу.**

SELECT r.reader\_name, r.reader\_surname

FROM readers r

INNER JOIN delivery\_readers dr

ON dr.reader\_id = r.reader\_id

INNER JOIN delivery d

ON dr.delivery\_id = d.delivery\_id

WHERE d.delivery\_date NOT BETWEEN '2017-04-18' AND '2017-04-20'

1. **Отримати список інвентарних номерів і назв з бібліотечного фонду, в яких міститься вказаний твір.**

SELECT l.library\_name, b.book\_id

FROM book b

INNER JOIN library l

ON b.library\_id = l.library\_id

WHERE b.book\_name = "Flatley Inc"

1. **Видати список інвентарних номерів і назв з бібліотечного фонду, в яких містяться твори зазначеного автора.**

SELECT l.library\_name, b.book\_id

FROM book b

INNER JOIN library l

ON b.library\_id = l.library\_id

INNER JOIN autor\_book ab

ON b.autor\_id = ab.autor\_id

INNER JOIN autor a

ON a.autor\_id = ab.autor\_id

WHERE a.autor\_name = "Stephen" AND a.autor\_surname = "Hamilton"

1. **Отримати список найпопулярніших творів.**

SELECT b.book\_name, COUNT(\*) num

FROM delivery d

INNER JOIN book b

ON b.book\_id = d.book\_id

GROUP BY d.book\_id

ORDER BY num DESC

LIMIT 10

**База даних**

-- phpMyAdmin SQL Dump

-- version 4.4.15.5

-- http://www.phpmyadmin.net

--

-- Host: 127.0.0.1:3306

-- Generation Time: Apr 25, 2017 at 09:46 AM

-- Server version: 5.6.34-log

-- PHP Version: 7.0.13

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `library`

--

-- --------------------------------------------------------

--

-- Table structure for table `advocacy`

--

CREATE TABLE IF NOT EXISTS `advocacy` (

`advocacy\_id` varchar(30) NOT NULL,

`advocacy\_firstdate` varchar(30) NOT NULL,

`advocacy\_lastdate` varchar(30) NOT NULL,

`book\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

--

-- Table structure for table `autor`

--

CREATE TABLE IF NOT EXISTS `autor` (

`autor\_id` varchar(30) NOT NULL,

`autor\_name` varchar(30) NOT NULL,

`autor\_fathername` varchar(30) NOT NULL,

`autor\_surname` varchar(30) NOT NULL,

`autor\_additional` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Dumping data for table `autor`

--

-- --------------------------------------------------------

--

-- Table structure for table `autor\_book`

--

CREATE TABLE IF NOT EXISTS `autor\_book` (

`autor\_id` varchar(30) NOT NULL,

`book\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `book`

--

CREATE TABLE IF NOT EXISTS `book` (

`book\_id` varchar(30) NOT NULL,

`autor\_id` varchar(30) NOT NULL,

`book\_name` varchar(30) NOT NULL,

`book\_year` varchar(30) NOT NULL,

`book\_pages` varchar(30) NOT NULL,

`book\_theme` varchar(30) NOT NULL,

`storage\_id` varchar(30) NOT NULL,

`library\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `delivery`

--

CREATE TABLE IF NOT EXISTS `delivery` (

`delivery\_id` varchar(30) NOT NULL,

`book\_id` varchar(30) NOT NULL,

`delivery\_date` date NOT NULL,

`delivery\_return` date NOT NULL,

`delivery\_checkin` tinyint(1) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `delivery\_readers`

--

CREATE TABLE IF NOT EXISTS `delivery\_readers` (

`delivery\_id` varchar(30) NOT NULL,

`reader\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `delivery\_worker`

--

CREATE TABLE IF NOT EXISTS `delivery\_worker` (

`delivery\_id` varchar(30) NOT NULL,

`worker\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `jobless`

--

CREATE TABLE IF NOT EXISTS `jobless` (

`jobless\_id` varchar(30) NOT NULL,

`reader\_id` varchar(30) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `library`

--

CREATE TABLE IF NOT EXISTS `library` (

`library\_id` varchar(30) NOT NULL,

`library\_name` varchar(30) NOT NULL,

`library\_adres` varchar(30) NOT NULL,

`library\_phone` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `library\_worker`

--

CREATE TABLE IF NOT EXISTS `library\_worker` (

`worker\_id` varchar(30) NOT NULL,

`worker\_name` varchar(30) NOT NULL,

`worker\_fathername` varchar(30) NOT NULL,

`worker\_surname` varchar(30) NOT NULL,

`worker\_position` varchar(30) NOT NULL,

`library\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `pensioner`

--

CREATE TABLE IF NOT EXISTS `pensioner` (

`pensioner\_id` varchar(30) NOT NULL,

`reader\_id` varchar(30) DEFAULT NULL,

`pensioner\_previleges` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `pupil`

--

CREATE TABLE IF NOT EXISTS `pupil` (

`pupil\_id` varchar(30) NOT NULL,

`reader\_id` varchar(30) DEFAULT NULL,

`pupil\_school` varchar(30) NOT NULL,

`pupil\_class` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `readers`

--

CREATE TABLE IF NOT EXISTS `readers` (

`reader\_id` varchar(30) NOT NULL,

`reader\_surname` varchar(30) NOT NULL,

`reader\_name` varchar(30) NOT NULL,

`reader\_fathername` varchar(30) NOT NULL,

`reader\_dateofbirth` date NOT NULL,

`reader\_home` varchar(100) NOT NULL,

`reader\_telephone` varchar(30) NOT NULL,

`library\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `scientist`

--

CREATE TABLE IF NOT EXISTS `scientist` (

`scientist\_id` varchar(30) NOT NULL,

`reader\_id` varchar(30) DEFAULT NULL,

`scientist\_workplace` varchar(30) NOT NULL,

`scientist\_degree` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `storage`

--

CREATE TABLE IF NOT EXISTS `storage` (

`storage\_id` varchar(30) NOT NULL,

`hall\_num` varchar(30) NOT NULL,

`rack\_num` varchar(30) NOT NULL,

`shalf\_num` varchar(30) NOT NULL,

`library\_id` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `student`

--

CREATE TABLE IF NOT EXISTS `student` (

`student\_id` varchar(30) NOT NULL,

`reader\_id` varchar(30) DEFAULT NULL,

`student\_university` varchar(30) NOT NULL,

`student\_faculty` varchar(30) NOT NULL,

`student\_year` varchar(30) NOT NULL,

`student\_group` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `teacher`

--

CREATE TABLE IF NOT EXISTS `teacher` (

`teacher\_id` varchar(30) NOT NULL DEFAULT '',

`reader\_id` varchar(30) NOT NULL,

`teacher\_workplace` varchar(30) NOT NULL,

`teacher\_position` varchar(30) NOT NULL,

`teacher\_degree` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Table structure for table `worker`

--

CREATE TABLE IF NOT EXISTS `worker` (

`worker\_id` varchar(30) NOT NULL DEFAULT '',

`reader\_id` varchar(30) NOT NULL,

`worker\_workplace` varchar(30) NOT NULL,

`worker\_position` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `advocacy`

--

ALTER TABLE `advocacy`

ADD PRIMARY KEY (`advocacy\_id`),

ADD KEY `book\_id` (`book\_id`);

--

-- Indexes for table `autor`

--

ALTER TABLE `autor`

ADD PRIMARY KEY (`autor\_id`);

--

-- Indexes for table `autor\_book`

--

ALTER TABLE `autor\_book`

ADD KEY `autor\_id` (`autor\_id`),

ADD KEY `book\_id` (`book\_id`);

--

-- Indexes for table `book`

--

ALTER TABLE `book`

ADD PRIMARY KEY (`book\_id`),

ADD KEY `storage\_id` (`storage\_id`);

--

-- Indexes for table `delivery`

--

ALTER TABLE `delivery`

ADD PRIMARY KEY (`delivery\_id`),

ADD KEY `book\_id` (`book\_id`);

--

-- Indexes for table `delivery\_readers`

--

ALTER TABLE `delivery\_readers`

ADD KEY `delivery\_id` (`delivery\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `delivery\_worker`

--

ALTER TABLE `delivery\_worker`

ADD KEY `delivery\_id` (`delivery\_id`),

ADD KEY `worker\_id` (`worker\_id`);

--

-- Indexes for table `jobless`

--

ALTER TABLE `jobless`

ADD PRIMARY KEY (`jobless\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `library`

--

ALTER TABLE `library`

ADD PRIMARY KEY (`library\_id`);

--

-- Indexes for table `library\_worker`

--

ALTER TABLE `library\_worker`

ADD PRIMARY KEY (`worker\_id`),

ADD KEY `library\_id` (`library\_id`);

--

-- Indexes for table `pensioner`

--

ALTER TABLE `pensioner`

ADD PRIMARY KEY (`pensioner\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `pupil`

--

ALTER TABLE `pupil`

ADD PRIMARY KEY (`pupil\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `readers`

--

ALTER TABLE `readers`

ADD PRIMARY KEY (`reader\_id`),

ADD KEY `fk\_readerslibr` (`library\_id`);

--

-- Indexes for table `scientist`

--

ALTER TABLE `scientist`

ADD PRIMARY KEY (`scientist\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `storage`

--

ALTER TABLE `storage`

ADD PRIMARY KEY (`storage\_id`),

ADD KEY `library\_id` (`library\_id`);

--

-- Indexes for table `student`

--

ALTER TABLE `student`

ADD PRIMARY KEY (`student\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `teacher`

--

ALTER TABLE `teacher`

ADD PRIMARY KEY (`teacher\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Indexes for table `worker`

--

ALTER TABLE `worker`

ADD PRIMARY KEY (`worker\_id`),

ADD KEY `reader\_id` (`reader\_id`);

--

-- Constraints for dumped tables

--

--

-- Constraints for table `advocacy`

--

ALTER TABLE `advocacy`

ADD CONSTRAINT `advocacy\_ibfk\_1` FOREIGN KEY (`book\_id`) REFERENCES `book` (`book\_id`);

--

-- Constraints for table `autor\_book`

--

ALTER TABLE `autor\_book`

ADD CONSTRAINT `autor\_book\_ibfk\_1` FOREIGN KEY (`autor\_id`) REFERENCES `autor` (`autor\_id`),

ADD CONSTRAINT `autor\_book\_ibfk\_2` FOREIGN KEY (`book\_id`) REFERENCES `book` (`book\_id`);

--

-- Constraints for table `book`

--

ALTER TABLE `book`

ADD CONSTRAINT `book\_ibfk\_1` FOREIGN KEY (`storage\_id`) REFERENCES `storage` (`storage\_id`);

--

-- Constraints for table `delivery`

--

ALTER TABLE `delivery`

ADD CONSTRAINT `delivery\_ibfk\_1` FOREIGN KEY (`book\_id`) REFERENCES `book` (`book\_id`);

--

-- Constraints for table `delivery\_readers`

--

ALTER TABLE `delivery\_readers`

ADD CONSTRAINT `delivery\_readers\_ibfk\_1` FOREIGN KEY (`delivery\_id`) REFERENCES `delivery` (`delivery\_id`),

ADD CONSTRAINT `delivery\_readers\_ibfk\_2` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `delivery\_worker`

--

ALTER TABLE `delivery\_worker`

ADD CONSTRAINT `delivery\_worker\_ibfk\_1` FOREIGN KEY (`delivery\_id`) REFERENCES `delivery` (`delivery\_id`),

ADD CONSTRAINT `delivery\_worker\_ibfk\_2` FOREIGN KEY (`worker\_id`) REFERENCES `worker` (`worker\_id`);

--

-- Constraints for table `jobless`

--

ALTER TABLE `jobless`

ADD CONSTRAINT `jobless\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `library\_worker`

--

ALTER TABLE `library\_worker`

ADD CONSTRAINT `library\_worker\_ibfk\_1` FOREIGN KEY (`library\_id`) REFERENCES `library` (`library\_id`);

--

-- Constraints for table `pensioner`

--

ALTER TABLE `pensioner`

ADD CONSTRAINT `pensioner\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `pupil`

--

ALTER TABLE `pupil`

ADD CONSTRAINT `pupil\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `readers`

--

ALTER TABLE `readers`

ADD CONSTRAINT `fk\_readerslibr` FOREIGN KEY (`library\_id`) REFERENCES `library` (`library\_id`);

--

-- Constraints for table `scientist`

--

ALTER TABLE `scientist`

ADD CONSTRAINT `scientist\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `storage`

--

ALTER TABLE `storage`

ADD CONSTRAINT `storage\_ibfk\_1` FOREIGN KEY (`library\_id`) REFERENCES `library` (`library\_id`);

--

-- Constraints for table `student`

--

ALTER TABLE `student`

ADD CONSTRAINT `student\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `teacher`

--

ALTER TABLE `teacher`

ADD CONSTRAINT `teacher\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

--

-- Constraints for table `worker`

--

ALTER TABLE `worker`

ADD CONSTRAINT `worker\_ibfk\_1` FOREIGN KEY (`reader\_id`) REFERENCES `readers` (`reader\_id`);

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;