**5330: Database Systems I Spring Semester 2015**

**Project 1**

**Name: Kaustubh Mohgaonkar ID: 1001101770**

***TASK1:***

**A. Make a list of the entities for the above process**

Answer: The following are the entities for the process:

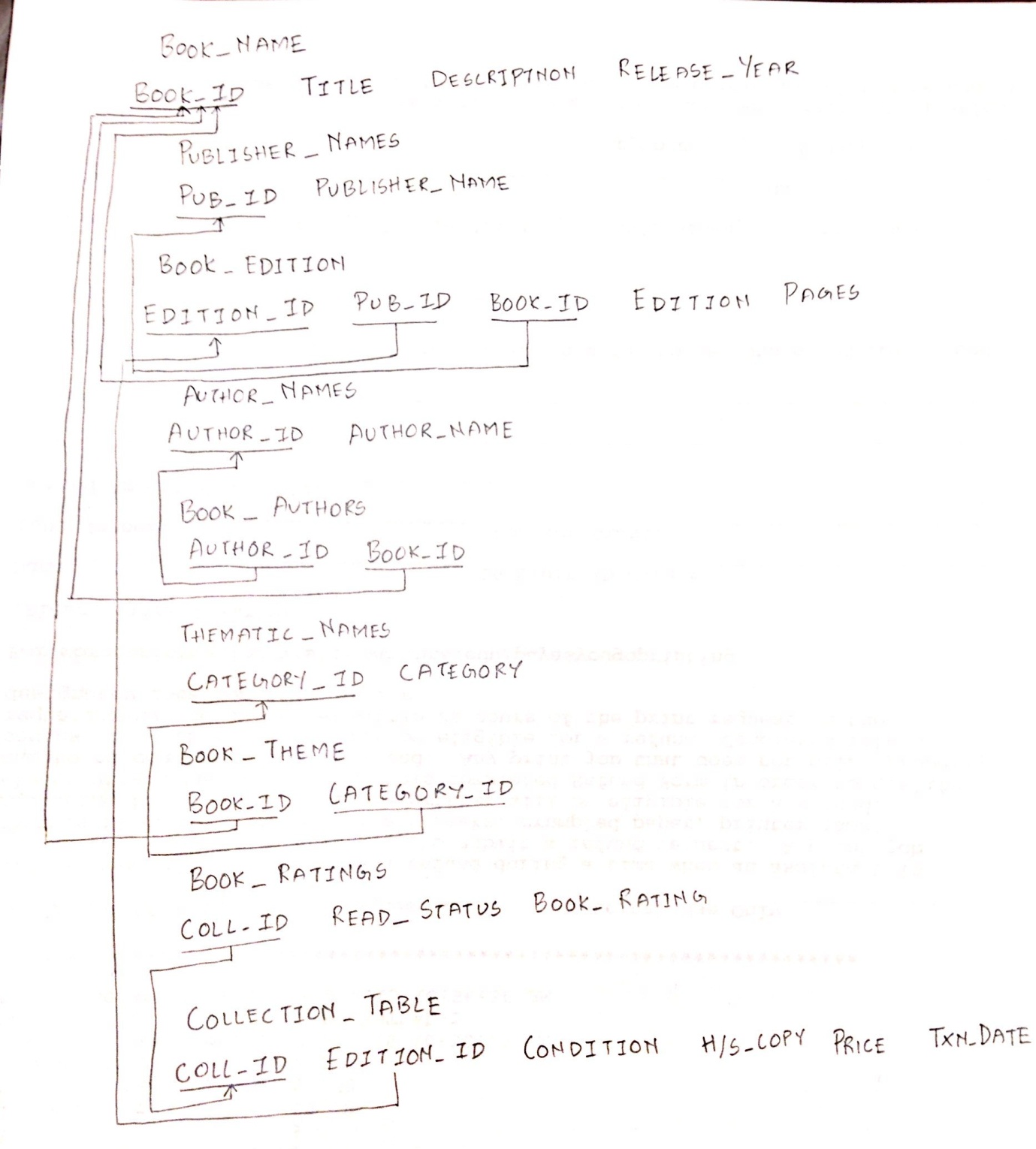
|  |  |
| --- | --- |
| Entities | Description |
| Author\_Names | Stores author names |
| Book\_Authors | Intermediate table to allow many to many relation between authors and books |
| Book\_Edition | Stores the edition and publication details of books |
| Book\_Names | Stores the title and description of books |
| Book\_Ratings | Stores read status and the ratings of read books |
| Book\_Theme | Intermediate table to allow many to many relation between themes and books |
| Collection\_Table | Stores details about user's collection |
| Publisher\_Names | Stores publisher names with id |
| Thematic\_Names | Stores theme names with id |

**B. For each entity define the required attributes and the data type of each of those attributes. You can add any attributes you believe that should be included.**

|  |  |  |
| --- | --- | --- |
| Entity | Attribute | Data Type |
| Author\_Names | Author\_ID | INT |
| Author\_Name | VARCHAR(30) |
| Book\_Authors | Author\_ID | INT |
| Book\_ID | INT |
| Book\_Edition | Edition\_ID | INT |
| Pub\_ID | INT |
| Book\_ID | INT |
| Edition | INT |
| Pages | INT |
| Book\_Names | Book\_ID | INT |
| Title | VARCHAR(45) |
| Book\_Desc | VARCHAR(60) |
| Release\_Year | YEAR |
| Book\_Ratings | Book\_ID | INT |
| Read\_Status | INT |
| Book\_Rating | INT |
| Book\_Theme | Book\_ID | INT |
| Category\_ID | INT |
| Collection\_Table | Coll\_ID | INT |
| Edition\_ID | INT |
| Condition | VARCHAR(5) |
| H/S\_Copy | VARCHAR(5) |
| Price | DECIMAL(10) |
| Txn\_Date | DATE |
| Publisher\_Names | Pub\_ID | INT |
| Publisher\_Name | VARCHAR(45) |
| Thematic\_Names | Category\_ID | INT |
| Category | VARCHAR(10) |

***TASK2:***

**A. Design the database schema (diagram) using arrows to define the references, based on the description. Your schema should be handwritten at this point.**

****

**B. Explain why you took specific decisions regarding:**

**The relations you have created.**

1. Author\_Names

Attributes:

* Author\_ID
* Author\_Name

This table lists author names with each of them assigned an author id. The author id is the primary key in this table as it uniquely identifies all authors. This table is kept separate from the Book\_Authors table to enable assigning one or more authors to a single book.

1. Book\_Authors

Attributes:

* Author\_ID
* Book\_ID

Here Author\_ID and Book\_ID combine to form a composite key which is the primary key for this table.

1. Book\_Edition

Attributes:

* Edition\_ID
* Pub\_ID
* Book\_ID
* Edition
* Pages

This table describes the different editions of a book based on its publisher. Here Pub\_ID, Book\_ID and Edition are combined to form a primary key. Pages attribute signifies pages for each edition.

1. Book\_Names

Attributes:

* Book\_ID
* Title
* Book\_Desc
* Release\_Year

This table describes the book details. Book\_ID is the primary key in this table. A single book can be published more than once, but the title and book description remains the same.

1. Book\_Ratings

Attributes:

* Book\_ID
* Read\_Status
* Book\_Rating

This status allows user to rate his book and also label read or not read as per user preference. Book\_ID is the primary key here.

1. Book\_Theme

Attributes:

* Book\_ID
* Category\_ID

In this table Book\_ID and Category\_ID are combined to form a composite key. Creation of this table enables assignment of one or more theme to a single book.

1. Collection\_Table

Attributes

* Coll\_ID
* Edition\_ID
* Condition
* H/S\_Copy
* Price
* Txn\_Date

This table stores all the books that the user actually owns. Condition field signifies if the book is in poor, good, mint, etc. condition. The HS field denotes if the books has hard cover or a soft cover. Price and Txn\_Date fields denote the complete transaction details of the purchased book. Coll\_ID is the primary key here.

1. Publisher\_Names

Attributes:

* Pub\_ID
* Publisher\_Name

This table denotes the list of all publisher names. Pub\_ID is the primary key here.

1. Thematic\_Names

Attributes:

* Category\_ID
* Category

This table lists all the themes of the book. These can be History, Novel, Poetry and Science. The Category\_ID is the primary key.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Constraint Name** | **Type** | **Information** |
| 1 | Book\_ID | Entity Integrity Constraint | Primary Key for Book\_Names Table |
| 2 | Pub\_ID | Entity Integrity Constraint | Primary Key for Publisher\_names Table |
| 3 | Edition\_ID | Entity Integrity Constraint | Primary Key for Book\_Edition table |
| 4 | Author\_ID | Entity Integrity Constraint | Primary Key for Author\_names table |
| 5 | Category\_ID | Entity Integrity Constraint | Primary Key for Thematic\_names table |
| 6 | Coll\_ID | Entity Integrity Constraint | Primary Key for Collection\_names table |
| 7 | Book\_ID | Referential Integrity | Foreign Key for Book\_Edition table references Book\_names table |
| 8 | Pub\_ID | Referential Integrity | Foreign Key for Book\_Edition table references Publisher\_names table |
| 9 | Author\_ID | Referential Integrity | Foreign Key for Book\_Authors references Author\_names table |
| 10 | Book\_ID | Referential Integrity | Foreign Key for Book\_Authors references book\_names table |
| 11 | Book\_ID | Referential Integrity | Foreign Key for Book\_theme references book\_names table |
| 12 | Category\_ID | Referential Integrity | Foreign Key for Book\_theme references thematic\_names table |
| 13 | Book\_ID | Referential Integrity | Foreign Key for Book\_theme references book\_names table |
| 14 | Edition\_ID | Referential Integrity | Foreign Key for Collection\_names references book\_names table |
| 15 |  | Not Null | Indicates that a column cannot store NULL value |

**C. The cardinality of the relationships and the references you made.**

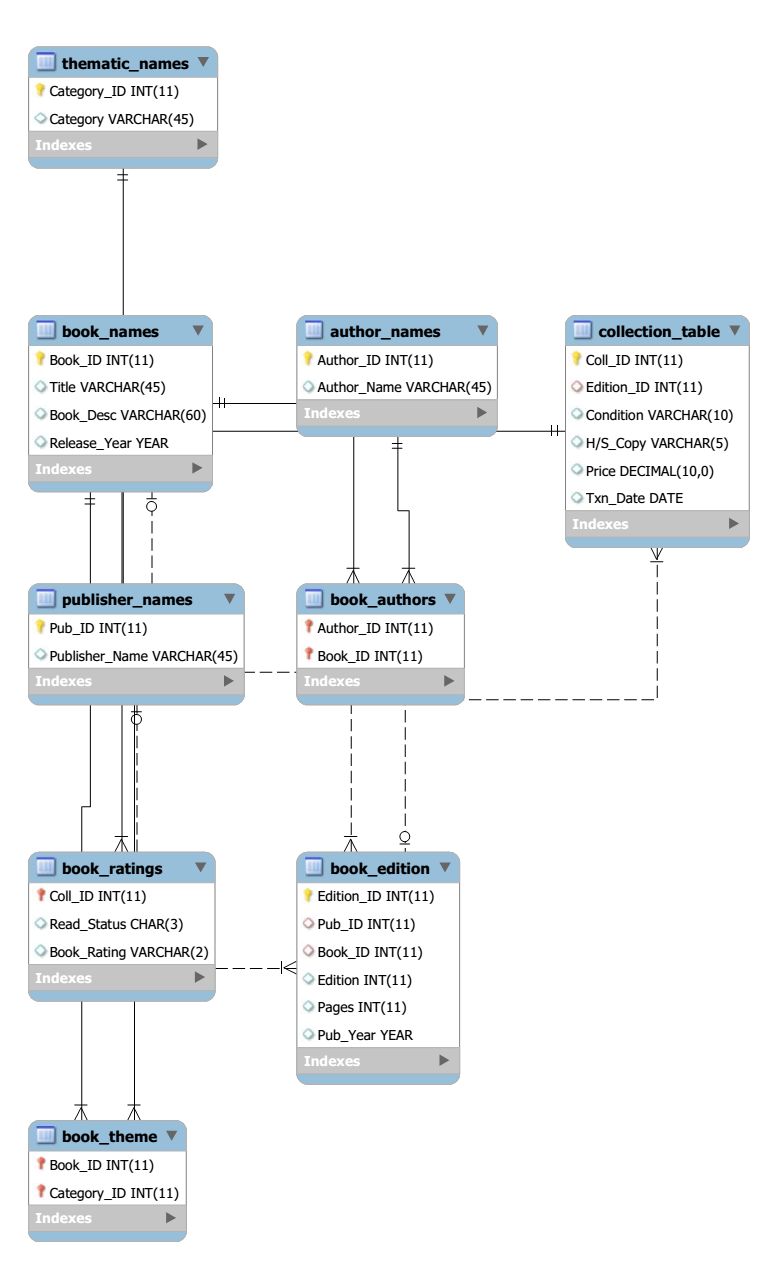
|  |  |  |
| --- | --- | --- |
| Cardinality of Relationships | | |
| Author\_ID | Book\_ID | Many to Many |
| Category\_ID | Book\_ID | Many to Many |
| Book\_ID | Edition\_ID | One to Many |
| Book\_ID | Pub\_ID | Many to Many |

***TASK3:***

**A. SQL CODE for SCHEMA:**

|  |
| --- |
| SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;  SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;  SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';  -- -----------------------------------------------------  -- Schema mydb  -- -----------------------------------------------------  -- -----------------------------------------------------  -- Schema book\_system  -- -----------------------------------------------------  CREATE SCHEMA IF NOT EXISTS `book\_system` DEFAULT CHARACTER SET utf8 ;  USE `book\_system` ;  -- -----------------------------------------------------  -- Table `book\_system`.`author\_names`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`author\_names` (  `Author\_ID` INT(11) NOT NULL AUTO\_INCREMENT,  `Author\_Name` VARCHAR(45) NULL DEFAULT NULL,  PRIMARY KEY (`Author\_ID`))  ENGINE = InnoDB  AUTO\_INCREMENT = 7  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`book\_names`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`book\_names` (  `Book\_ID` INT(11) NOT NULL AUTO\_INCREMENT,  `Title` VARCHAR(45) NULL DEFAULT NULL,  `Book\_Desc` VARCHAR(60) NULL DEFAULT NULL,  `Release\_Year` YEAR NULL DEFAULT NULL,  PRIMARY KEY (`Book\_ID`))  ENGINE = InnoDB  AUTO\_INCREMENT = 26  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`book\_authors`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`book\_authors` (  `Author\_ID` INT(11) NOT NULL,  `Book\_ID` INT(11) NOT NULL,  PRIMARY KEY (`Author\_ID`, `Book\_ID`),  INDEX `Book\_ID\_idx` (`Book\_ID` ASC),  CONSTRAINT `Author\_ID`  FOREIGN KEY (`Author\_ID`)  REFERENCES `book\_system`.`author\_names` (`Author\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE,  CONSTRAINT `Book\_ID`  FOREIGN KEY (`Book\_ID`)  REFERENCES `book\_system`.`book\_names` (`Book\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`publisher\_names`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`publisher\_names` (  `Pub\_ID` INT(11) NOT NULL AUTO\_INCREMENT,  `Publisher\_Name` VARCHAR(45) NULL DEFAULT NULL,  PRIMARY KEY (`Pub\_ID`))  ENGINE = InnoDB  AUTO\_INCREMENT = 13  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`book\_edition`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`book\_edition` (  `Edition\_ID` INT(11) NOT NULL AUTO\_INCREMENT,  `Pub\_ID` INT(11) NULL DEFAULT NULL,  `Book\_ID` INT(11) NULL DEFAULT NULL,  `Edition` INT(11) NULL DEFAULT NULL,  `Pages` INT(11) NULL DEFAULT NULL,  `Pub\_Year` YEAR NULL DEFAULT NULL,  PRIMARY KEY (`Edition\_ID`),  INDEX `Pub\_ID\_idx` (`Pub\_ID` ASC),  INDEX `Book\_ID\_idx` (`Book\_ID` ASC),  CONSTRAINT `Book\_ID1`  FOREIGN KEY (`Book\_ID`)  REFERENCES `book\_system`.`book\_names` (`Book\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE,  CONSTRAINT `Pub\_ID1`  FOREIGN KEY (`Pub\_ID`)  REFERENCES `book\_system`.`publisher\_names` (`Pub\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  AUTO\_INCREMENT = 24  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`collection\_table`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`collection\_table` (  `Coll\_ID` INT(11) NOT NULL AUTO\_INCREMENT,  `Edition\_ID` INT(11) NULL DEFAULT NULL,  `Condition` VARCHAR(10) NULL DEFAULT NULL,  `H/S\_Copy` VARCHAR(5) NULL DEFAULT NULL,  `Price` FLOAT NULL DEFAULT NULL,  `Txn\_Date` DATE NULL DEFAULT NULL,  PRIMARY KEY (`Coll\_ID`),  INDEX `Edition\_ID\_idx` (`Edition\_ID` ASC),  CONSTRAINT `Edition\_ID1`  FOREIGN KEY (`Edition\_ID`)  REFERENCES `book\_system`.`book\_edition` (`Edition\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  AUTO\_INCREMENT = 36  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`book\_ratings`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`book\_ratings` (  `Coll\_ID` INT(11) NOT NULL,  `Read\_Status` CHAR(3) NULL DEFAULT NULL,  `Book\_Rating` VARCHAR(2) NULL DEFAULT NULL,  PRIMARY KEY (`Coll\_ID`),  CONSTRAINT `Coll\_ID1`  FOREIGN KEY (`Coll\_ID`)  REFERENCES `book\_system`.`collection\_table` (`Coll\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`thematic\_names`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`thematic\_names` (  `Category\_ID` INT(11) NOT NULL AUTO\_INCREMENT,  `Category` VARCHAR(45) NULL DEFAULT NULL,  PRIMARY KEY (`Category\_ID`))  ENGINE = InnoDB  AUTO\_INCREMENT = 5  DEFAULT CHARACTER SET = utf8;  -- -----------------------------------------------------  -- Table `book\_system`.`book\_theme`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `book\_system`.`book\_theme` (  `Book\_ID` INT(11) NOT NULL,  `Category\_ID` INT(11) NOT NULL,  PRIMARY KEY (`Book\_ID`, `Category\_ID`),  INDEX `Book\_ID\_idx` (`Book\_ID` ASC),  INDEX `Category\_ID\_idx` (`Category\_ID` ASC),  CONSTRAINT `Book\_ID2`  FOREIGN KEY (`Book\_ID`)  REFERENCES `book\_system`.`book\_names` (`Book\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE,  CONSTRAINT `Category\_ID`  FOREIGN KEY (`Category\_ID`)  REFERENCES `book\_system`.`thematic\_names` (`Category\_ID`)  ON DELETE CASCADE  ON UPDATE CASCADE)  ENGINE = InnoDB  DEFAULT CHARACTER SET = utf8;  SET SQL\_MODE=@OLD\_SQL\_MODE;  SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;  SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS; |

**B. Reverse Engineered Diagram from MySQL:**

****

**C. INSERT Statements:**

|  |
| --- |
| -- Author\_Names Table  INSERT INTO author\_names (Author\_ID, Author\_Name) VALUES (1,'John Terry');  INSERT INTO author\_names (Author\_ID, Author\_Name) VALUES (2,'Eden Hazard');  INSERT INTO author\_names (Author\_ID, Author\_Name) VALUES (3,'Diego Costa');  INSERT INTO author\_names (Author\_ID, Author\_Name) VALUES (4,'Petr Cech');  INSERT INTO author\_names (Author\_ID, Author\_Name) VALUES (5,'Oscar');  INSERT INTO author\_names (Author\_ID, Author\_Name) VALUES (6,'Willian');  -- Book\_Authors Table  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (1,1);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (6,1);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (2,2);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (5,2);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (3,3);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (4,3);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (3,4);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (4,4);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (2,5);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (5,5);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (5,6);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (6,6);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (1,7);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (4,7);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (2,8);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (3,8);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (2,9);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (3,9);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (1,10);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (4,10);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (5,11);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (6,12);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (1,13);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (2,14);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (3,15);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (4,16);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (5,17);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (6,18);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (1,19);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (2,20);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (3,21);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (4,22);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (5,23);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (6,24);  INSERT INTO book\_authors (Author\_ID, Book\_ID) VALUES (1,25);  -- book\_edition table  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (1,5,1,1,900,2001);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (2,5,1,2,950,2002);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (3,5,1,3,1200,2003);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (4,7,2,1,500,2004);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (5,7,2,2,700,2005);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (6,4,2,1,520,2006);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (7,4,2,2,600,2007);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (8,4,2,3,700,2008);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (9,3,3,1,486,2009);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (10,2,4,1,600,2010);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (11,8,3,1,789,2011);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (12,1,5,1,460,2012);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (13,1,5,2,580,2013);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (14,1,6,1,100,2014);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (15,2,7,1,200,2015);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (16,3,8,1,300,2014);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (17,4,9,1,400,2013);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (18,5,10,1,500,2012);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (19,5,1,4,1250,2013);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (20,3,11,1,150,1999);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (21,2,15,1,250,1998);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (22,3,20,1,350,1997);  INSERT INTO book\_edition (Edition\_ID, Pub\_ID, Book\_ID, Edition, Pages, Pub\_Year) VALUES (23,2,25,1,450,1998);  -- book\_names table  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (1,'Databases','Basics of database',1990);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (2,'Economics','Basics of Economics',2001);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (3,'Computer Science','Basics of Computer Science',1988);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (4,'History','Basics of History',1995);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (5,'Law Fundamentals','Basics of Law and order',1999);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (6,'Cloud Computing','Basics of Cloud',2010);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (7,'Algorithms','Basics of Algorithms',2014);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (8,'Management Information Systems','Basics of Management Information Systems',1984);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (9,'Engineering Management','Basics of Engineering Management',1986);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (10,'Business Analytics','Basics of Business Analytics',2003);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (11,'All the Light We Cannot See ','Good Novel',1981);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (12,'Yes Please','Basics of database',1982);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (13,'The Blood of Olympus','Novel 13',1983);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (14,'The Storied Life of A.J. Fikry','Novel 14',1984);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (15,'The Silkworm (Cormoran Strike, #2)','Novel 11',1985);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (16,'The Bone Clocks','Novel 16',1981);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (17,'The Girl with All the Gifts','Novel 17',1982);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (18,'Love Letters to the Dead','Novel 18',1983);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (19,'Leaving Time','Novel 19',1984);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (20,'The Miniaturist','Novel 20',1986);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (21,'The Paying Guests','Novel 21',1934);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (22,'Cruel Beauty','Novel 22',1935);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (23,'The Good Girl','Novel 23',1936);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (24,'Revival','Novel 24',1937);  INSERT INTO book\_names (Book\_ID, Title, Book\_Desc, Release\_Year) VALUES (25,'The One & Only','Novel 25',1938);  -- book\_ratings table  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (1,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (2,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (3,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (4,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (5,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (6,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (7,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (8,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (9,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (10,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (11,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (12,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (13,'1','5');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (14,'1','2');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (15,'1','2');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (16,'1','2');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (17,'1','3');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (18,'1','3');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (19,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (20,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (21,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (22,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (23,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (24,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (25,'1','1');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (26,'1','1');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (27,'1','4');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (28,'1','4');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (29,'1','4');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (30,'1','4');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (31,'1','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (32,'1','4');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (33,'1','4');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (34,'0','NA');  INSERT INTO book\_ratings (Coll\_ID, Read\_Status, Book\_Rating) VALUES (35,'0','NA');  -- book\_theme table  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (1,1);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (1,3);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (1,4);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (2,2);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (2,3);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (3,2);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (3,3);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (4,3);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (4,4);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (5,1);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (5,2);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (6,2);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (7,3);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (8,4);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (9,1);  INSERT INTO book\_theme (Book\_ID, Category\_ID) VALUES (10,2);  -- collection\_table table  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (1,1,'mint','hard',125,'2010-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (2,1,'mint','hard',125,'2011-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (3,1,'mint','hard',125,'2012-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (4,1,'mint','hard',125,'2013-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (5,1,'mint','hard',125,'2014-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (6,2,'good','soft',150,'2014-11-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (7,3,'good','soft',175,'2015-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (8,4,'poor','soft',50,'2006-05-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (9,5,'poor','hard',60,'2006-06-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (10,6,'acceptable','soft',100,'2011-04-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (11,6,'poor','hard',0.50,'2005-06-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (12,7,'poor','hard',0.75,'2005-04-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (13,8,'poor','soft',0.80,'2005-06-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (14,9,'poor','hard',0.55,'2005-04-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (15,10,'very good','soft',200,'2013-06-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (16,11,'poor','hard',220,'2014-04-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (17,12,'very good','soft',240,'2014-06-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (18,13,'very good','hard',260,'2015-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (19,13,'very good','hard',260,'2015-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (20,13,'very good','hard',260,'2015-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (21,4,'mint','soft',350,'2014-05-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (22,5,'very good','hard',360,'2014-06-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (23,14,'poor','soft',75,'2013-01-01');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (24,15,'poor','soft',80,'2013-01-02');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (25,16,'poor','soft',85,'2013-01-03');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (26,17,'good','hard',90,'2013-01-04');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (27,18,'good','soft',95,'2013-01-05');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (28,14,'good','hard',100,'2013-01-06');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (29,15,'acceptable','soft',105,'2013-01-07');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (30,16,'acceptable','hard',110,'2013-01-08');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (31,19,'mint','hard',150,'2015-01-02');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (32,20,'mint','hard',210,'2011-01-03');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (33,21,'good','soft',220,'2013-07-03');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (34,22,'very good','hard',230,'2012-04-03');  INSERT INTO collection\_table (Coll\_ID, Edition\_ID, Condition, `H/S\_Copy`, Price, Txn\_Date) VALUES (35,23,'very good','soft',240,'2011-01-03');  -- publisher\_names table  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (1,'Arcade Publishing');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (2,'Arkham House');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (3,'Cisco Press');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (4,'CRC Press');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (5,'Exact Change');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (6,'Houghton Mifflin');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (7,'John Lane');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (8,'Kensington Books');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (9,'Llewellyn Worldwide');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (10,'Medknow Publications');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (11,'Marion Boyars Publishers');  INSERT INTO publisher\_names (Pub\_ID, Publisher\_Name) VALUES (12,'NavPress');  -- thematic\_names table  INSERT INTO thematic\_names (Category\_ID, Category) VALUES (1,'history');  INSERT INTO thematic\_names (Category\_ID, Category) VALUES (2,'novel');  INSERT INTO thematic\_names (Category\_ID, Category) VALUES (3,'poetry');  INSERT INTO thematic\_names (Category\_ID, Category) VALUES (4,'science'); |

**D. List of 'on update' and 'on delete' constraints**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Constraint Name | On Update | On Delete |
| book\_authors | ba\_Author\_ID | CASCADE | CASCADE |
| ba\_Book\_ID | CASCADE | CASCADE |
| book\_edition | be\_Book\_ID | CASCADE | CASCADE |
| be\_Pub\_ID | CASCADE | CASCADE |
| book\_ratings | br\_Coll\_ID | CASCADE | CASCADE |
| book\_theme | bt\_Book\_ID | CASCADE | CASCADE |
| bt\_Category\_ID | CASCADE | CASCADE |
| collection\_table | ct\_Edition\_ID | CASCADE | CASCADE |

**E. The initial state of the schema will contain the following items:**

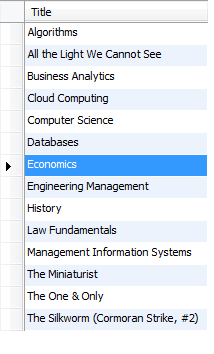
* **Relations**
* **Attributes of relations**
* **Constraints**
  + Entity Integrity Constraint
  + Referential Integrity Constraints
* **Sample Data**

***TASK 4:***

A. Show a list of the books you own sorted alphabetically based on the title

|  |
| --- |
| select book\_names.Title  from book\_names, collection\_table, book\_edition  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_edition.Book\_ID = book\_names.Book\_ID  order by book\_names.Title ASC; |

Output:



B. Show the total number of your books

|  |
| --- |
| select count(collection\_table.Coll\_ID) as 'Total Number of Books'  from collection\_table |

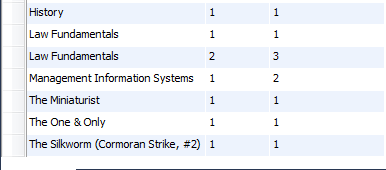
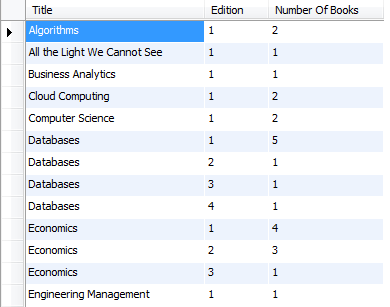
Output:



C. Show for each book the number of times you own it

|  |
| --- |
| select book\_names.Title,book\_edition.Edition , count(book\_names.Title) as NumberOfBooks  from collection\_table, book\_edition, book\_names  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_edition.Book\_ID = book\_names.Book\_ID  group by book\_names.Title, book\_edition.edition; |

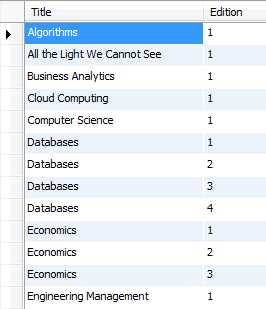
Output:



D. Show your unique publications. More specifically the query should return each book of the same publisher and same edition only once, even though you might own it more than once.

|  |
| --- |
| select book\_names.Title, book\_edition.Edition  from book\_names, collection\_table, book\_edition  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_edition.Book\_ID = book\_names.Book\_ID  group by book\_names.Title, book\_edition.Edition; |

Output:

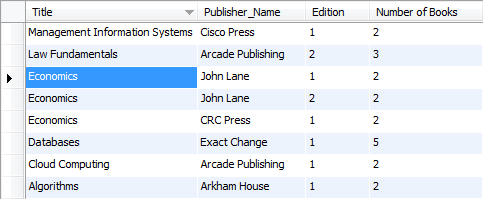




E. Show only those books you own more than once

|  |
| --- |
| select book\_names.Title, publisher\_names.Publisher\_Name, book\_edition.Edition , count(collection\_table.Edition\_ID)  from book\_names, collection\_table, book\_edition, publisher\_names  where book\_names.Book\_ID = book\_edition.Book\_ID  and book\_edition.Edition\_ID = collection\_table.Edition\_ID  and book\_edition.Pub\_ID = publisher\_names.Pub\_ID  group by collection\_table.Edition\_ID  having count(collection\_table.Edition\_ID) > 1; |

Output:



F. Show the publisher for whom you have the largest number of books

|  |
| --- |
| select pub.Publisher\_Name, count(be.Pub\_ID) as Number\_Of\_Books  from collection\_table c, book\_edition be, publisher\_names pub  where c.Edition\_ID = be.Edition\_ID and be.Pub\_ID = pub.Pub\_ID  group by be.Pub\_ID  order by Number\_Of\_Books desc limit 1; |



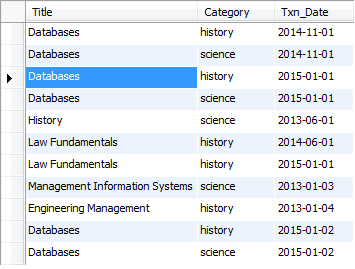
G. Show the book with the most edition updates which you have still not read (but you own).

|  |
| --- |
| select book\_edition.Edition\_ID, book\_names.Title, book\_edition.Edition  from book\_names, book\_edition, collection\_table, book\_ratings  where book\_edition.Edition\_ID = collection\_table.Edition\_ID  and book\_names.Book\_ID = book\_edition.Book\_ID  and collection\_table.Coll\_ID = book\_ratings.Coll\_ID  and book\_ratings.Read\_Status = 'No'  group by book\_names.Title, book\_edition.Edition  order by book\_edition.Edition desc limit 1; |



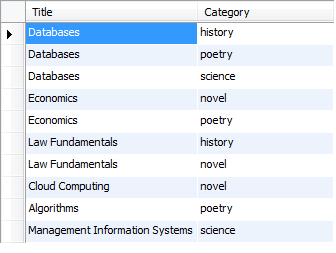
H. Show the books purchased after a given year (you decide) with a thematic category “science” or “history”.

|  |
| --- |
| select book\_names.Title, thematic\_names.Category, collection\_table.Txn\_Date  from book\_names, thematic\_names, book\_theme, collection\_table, book\_edition  where book\_names.Book\_ID = book\_edition.Book\_ID  and book\_edition.Edition\_ID = collection\_table.Edition\_ID  and book\_theme.Book\_ID = book\_names.Book\_ID  and book\_theme.Category\_ID = thematic\_names.Category\_ID  group by collection\_table.Edition\_ID, book\_names.Title, thematic\_names.Category  having year(Txn\_Date) > 2011 and (Category = 'history' or Category = 'science') ; |



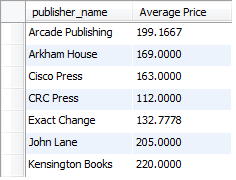
I. Show a list of your books with more than one thematic categories. Show the books with more categories first.

|  |
| --- |
| select distinct book\_names.Title, thematic\_names.Category  from book\_names, thematic\_names, book\_theme, collection\_table, book\_edition  where book\_names.Book\_ID = book\_edition.Book\_ID  and book\_edition.Edition\_ID = collection\_table.Edition\_ID  and book\_theme.Book\_ID = book\_names.Book\_ID  and book\_theme.Category\_ID = thematic\_names.Category\_ID  group by collection\_table.Edition\_ID, book\_names.Title, thematic\_names.Category  having count(category) > 1 ; |



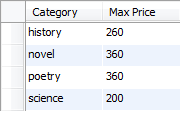
J. Show the average price you have paid per publisher. Consider that all your books need to be taken into account, even the duplicates.

|  |
| --- |
| select pb.publisher\_name, avg(ct.Price) as AveragePrice  from publisher\_names as pb, book\_edition as be, collection\_table as ct  where ct.Edition\_id = be.Edition\_ID  and be.Pub\_ID = pb.Pub\_ID  group by pb.Publisher\_name; |



K. Show the most expensive book per thematic category.

|  |
| --- |
| select thematic\_names.Category, max(collection\_table.price) as MaxPrice  from book\_names, thematic\_names, collection\_table, book\_theme, book\_edition  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_names.Book\_ID = book\_edition.Book\_ID  and thematic\_names.Category\_ID = book\_theme.Category\_ID  and book\_theme.Book\_ID = book\_names.Book\_ID  group by thematic\_names.Category; |



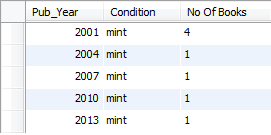
L. Show the least expensive book which is in mint condition.

|  |
| --- |
| select book\_names.Title, collection\_table.Condition , min(collection\_table.Price) as MinPrice  from collection\_table, book\_names, book\_edition  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_names.Book\_ID = book\_edition.Book\_ID  group by collection\_table.Condition  having collection\_table.Condition = 'mint'; |



M. Show how many mint books you have for each publication year

|  |
| --- |
| select book\_edition.Pub\_Year, collection\_table.Condition , count(collection\_table.Condition) as 'No Of Books'  from collection\_table, book\_edition  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  group by book\_edition.Pub\_year, collection\_table.Condition  having collection\_table.Condition = 'mint'; |



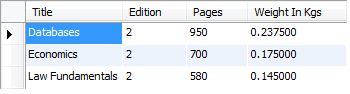
N. What percentage (%) of your single-author books have you read and rated with more than three stars?

|  |
| --- |
| select sum(a.Read\_Status)/count(\*) \* 100 as Percentage  from (select book\_names.Title, author\_names.Author\_Name, book\_ratings.Read\_Status  from book\_names, author\_names, book\_authors, book\_edition, collection\_table, book\_ratings  where book\_names.Book\_ID = book\_authors.Book\_ID  and collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_names.Book\_ID = book\_edition.Book\_ID  and book\_authors.Author\_ID = author\_names.Author\_ID  and collection\_table.Coll\_ID = book\_ratings.Coll\_ID  group by book\_authors.Book\_ID  having count(book\_authors.Book\_ID) = 1) a |



O. If each leaf of the book (1 leaf=2 pages) weights 0.5 gr, show the weight of your second edition books in kilos. Do not consider the hardcover for this query.

|  |
| --- |
| select book\_names.Title, book\_edition.Edition, book\_edition.Pages, ((book\_edition.Pages \* 0.25)/1000) as 'Weight In Kgs'  from book\_names, book\_edition, collection\_table  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_names.Book\_ID = book\_edition.Book\_ID  group by book\_names.Title, book\_edition.Edition  having book\_edition.Edition = 2; |



***TASK 5:***

A. Write a query which would update the number of pages: increase by 10 pages for those books which were written after the year 2000.

|  |
| --- |
| update book\_edition  set book\_edition.Pages = book\_edition.Pages - 10  where book\_edition.Pub\_Year > 2000; |

Table Output Before update query:

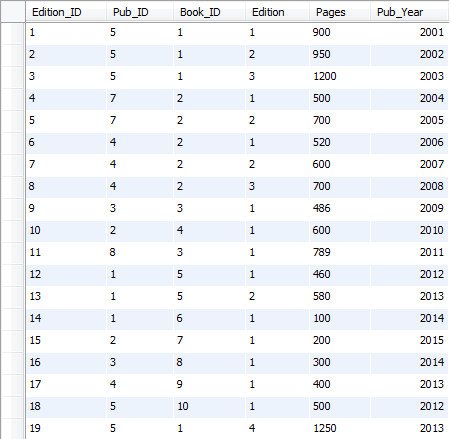
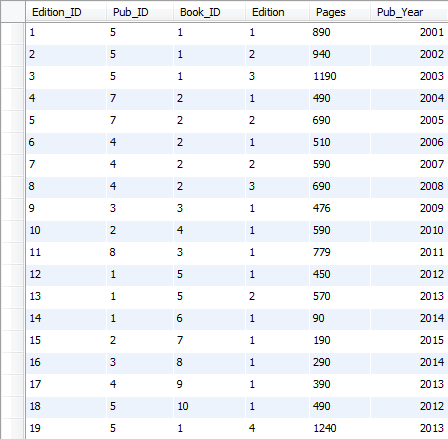


Table output after update query:



B. Write a query which would delete all books that you bought in 2005 in "poor" condition and a cost less than $1.00

|  |
| --- |
| delete from collection\_table  where year(Txn\_Date) = 2005  and collection\_table.Condition = 'poor'  and price < 1; |

Output of Select Query before delete query:

|  |
| --- |
| select \* from collection\_table  where year(Txn\_Date) = 2005  and collection\_table.Condition = 'poor'  and price < 1; |



Output of Select Query after delete query:

|  |
| --- |
| select \* from collection\_table  where year(Txn\_Date) = 2005  and collection\_table.Condition = 'poor'  and price < 1; |

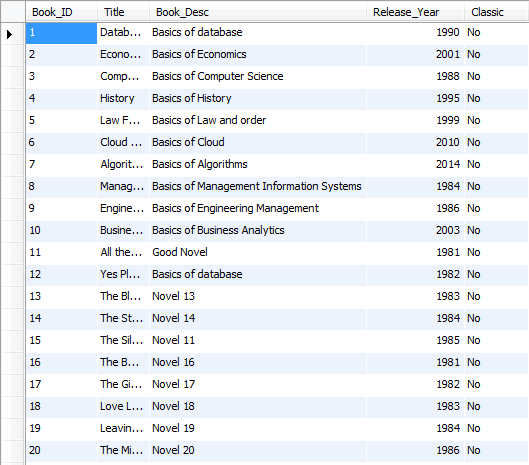
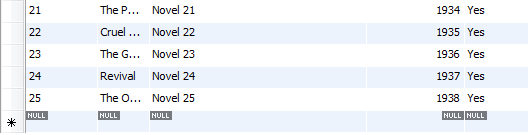


C. Write a query which would delete the relation which stores the information about the authors

|  |
| --- |
| ALTER TABLE `book\_system`.`book\_authors`  DROP FOREIGN KEY `ba\_Book\_ID`,  DROP FOREIGN KEY `ba\_Author\_ID`; |

D. Write a query which would create an extra attribute in the appropriate relation and name it ‘classic’ which automatically takes values ‘true’ if the book was written before 1950 and ‘false’ if the book was written after 1950

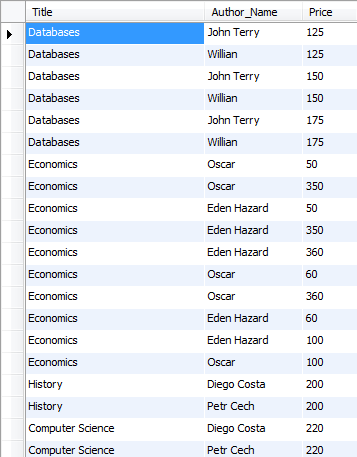
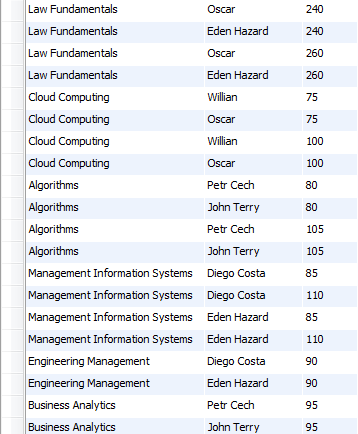
|  |
| --- |
| update book\_names set classic = 'Yes'  where Release\_Year < 1950  update book\_names set classic = 'No'  where Release\_Year > 1950 |

CREATE VIEWS for below queries:

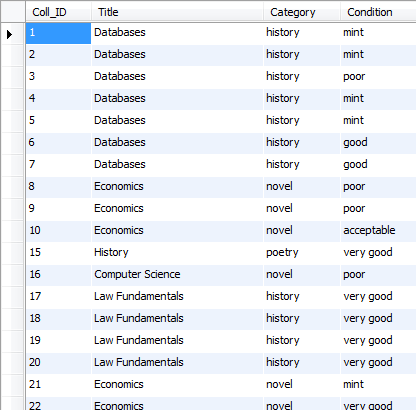
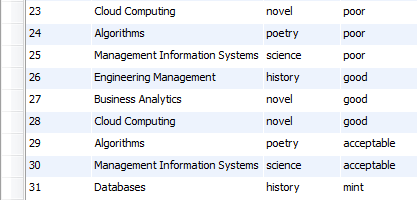
E. The book title with the money paid and the author(s).

|  |
| --- |
| create view Book\_Details\_View  as (select distinct collection\_table.Edition\_ID, book\_names.Title, author\_names.Author\_Name, collection\_table.Price  from book\_names, book\_edition, author\_names, book\_authors, collection\_table  where book\_names.Book\_ID = book\_authors.Book\_ID  and book\_authors.Author\_ID = author\_names.Author\_ID  and book\_names.Book\_ID = book\_edition.Book\_ID  and book\_edition.Edition\_ID = collection\_table.Edition\_ID  order by collection\_table.Edition\_ID); |

F. The book title with the thematic category and the condition.

|  |
| --- |
| create view book\_det\_thematic as  (select collection\_table.Coll\_ID, book\_names.Title, thematic\_names.Category, collection\_table.Condition  from collection\_table, book\_edition, book\_theme, thematic\_names, book\_names  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_edition.Book\_ID = book\_names.Book\_ID  and book\_theme.Book\_ID = book\_names.Book\_ID  and thematic\_names.Category\_ID = book\_theme.Category\_ID  group by collection\_table.Coll\_ID); |

G. The average amount of money paid for each unique book, alongside with info about the thematic category

|  |
| --- |
| create view Book\_name\_AVGPRice as  (select book\_names.Title,thematic\_names.Category , AVG(collection\_table.Price) as 'Average Price'  from collection\_table, book\_names, thematic\_names, book\_theme, book\_edition  where collection\_table.Edition\_ID = book\_edition.Edition\_ID  and book\_edition.Book\_ID = book\_names.Book\_ID  and book\_names.Book\_ID = book\_theme.Book\_ID  and book\_theme.Category\_ID = thematic\_names.Category\_ID  group by book\_names.Title); |

