**INFSCI 2710 Database Management, fall 2016 Monday**

**Homework 1: Relational Algebra, SQL**

**100 pts**

**Due Date: Oct 4, at the beginning of the class. Hard copy, please type and print your answers.**

Consider the following schema:

author (ID, FirstName, LastName, YearOfBirth, Gender, LivingCityID)

book (ID, Name, Type, YearPublished, PublisherID)

writes (BookID, AuthorID)

city (ID, CityName, Country)

publisher (ID, PublisherName, LocationCityID)

Underlines attributes in bold are the primary keys. Assume that one book can be published by only one publisher; one author can write several books; one book may have several authors.

**Q1 [20 pt]:** Write SQL DDL statements to create the above tables. Make sure that you capture the primary and foreign key constraints (if applicable), choose appropriate domain (data) type and constraints for each attribute. Try to define the rules for foreign key on deleting and updating.

Answer: Let the database name is “bookinfo”;

CREATE TABLE `bookinfo`.`city` (

`ID` INT NOT NULL,

`CityName` VARCHAR(45) NOT NULL,

`Country` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ID`));

CREATE TABLE `bookinfo`.`publisher` (

`ID` INT NOT NULL,

`PublisherName` VARCHAR(45) NOT NULL,

`LocationCityID` INT NOT NULL,

PRIMARY KEY (`ID`),

INDEX `locate\_idx` (`LocationCityID` ASC),

CONSTRAINT `locate`

FOREIGN KEY (`LocationCityID`)

REFERENCES `bookinfo`.`city` (`ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE `bookinfo`.`author` (

`ID` INT NOT NULL,

`FirstName` VARCHAR(45) NOT NULL,

`LastName` VARCHAR(45) NOT NULL,

`YearOfBirth` INT NOT NULL,

`Gender` VARCHAR(45) NOT NULL,

`LivingCityID` INT NOT NULL,

PRIMARY KEY (`ID`),

CONSTRAINT `livein`

FOREIGN KEY (`ID`)

REFERENCES `bookinfo`.`city` (`ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE `bookinfo`.`book` (

`ID` INT NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`Type` VARCHAR(45) NOT NULL,

`YearPublished` INT NOT NULL,

`PublisherID` INT NOT NULL,

PRIMARY KEY (`ID`),

INDEX `publishedBy\_idx` (`PublisherID` ASC),

CONSTRAINT `publishedBy`

FOREIGN KEY (`PublisherID`)

REFERENCES `bookinfo`.`publisher` (`ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE `bookinfo`.`writes` (

`BookID` INT NOT NULL,

`AuthorID` INT NOT NULL,

PRIMARY KEY (`BookID`, `AuthorID`),

INDEX `writenBy\_idx` (`AuthorID` ASC),

CONSTRAINT `bookIs`

FOREIGN KEY (`BookID`)

REFERENCES `bookinfo`.`book` (`ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `writenBy`

FOREIGN KEY (`AuthorID`)

REFERENCES `bookinfo`.`author` (`ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

**Q2 [5 pt]:** Specify a relational algebra expression and an SQL query to find all the book names in the dataset.

SELECT book.Name FROM bookinfo.book;

**Q3 [10 pt]:** Specify a relational algebra expression and an SQL query to find the IDs of authors that born in 1964.

SELECT author.ID FROM bookinfo.author where YearOfBirth=1964;

**Q4 [10 pt]:** Specify a relational algebra expression and an SQL query to find the IDs of authors that lives in Pittsburgh.

SELECT author.ID FROM bookinfo.author, bookinfo.city

where city.CityName='pittsburgh' and city.ID=author.LivingCityID

**Q5 [10 pt]:** Specify a relational algebra expression and an SQL query to find the names of all books written by the author whose first name is ‘Peter’ and last name is ‘Lee’.

SELECT Name FROM bookinfo.writes, bookinfo.book, bookinfo.author

where writes.BookID=book.ID and writes.AuthorID=author.ID

and author.FirstName='Peter' and author.LastName='Lee'

**Q6 [15 pt]:** Specify a relational algebra expression and an SQL query to find the books whose author publish it in the local press. If the book has several authors, at least one author’s living city should be same with the publisher’s location. Please output the book ID, and remove the duplicated IDs.

SELECT distinct(BookID) FROM bookinfo.writes, bookinfo.book, bookinfo.author, bookinfo.publisher

where writes.BookID=book.ID and writes.AuthorID=author.ID

and book.PublisherID=publisher.ID and author.LivingCityID=publisher.LocationCityID

**Q7 [15 pt]:** When the data is loaded into the database, there is one error. One book is published before the author is born. Please find out this tuple. Specify the SQL query.

SELECT \* FROM bookinfo.writes, bookinfo.book, bookinfo.author

where writes.BookID=book.ID and writes.AuthorID=author.ID

and book.YearPublished<author.YearOfBirth

**Q8 [15 pt]:** Consider the following relational algebra expression:

1. To get the correct answer for “Find the book written by author whose last name is King”, which relational algebra expression is correct? Explain the reason.

A is correct. B mistakenly conducts “author.ID=write.ID”.

1. In terms of the correct one, write an equivalent version of the relational algebra expression that is more efficient. Specify why your expression is more efficient.

Though we would need to know the number of tuples in each relation, there might be fewer tuples in the table inner join process for the given expression.

1. Translate the correct expressions into SQL.

SELECT Name FROM bookinfo.writes, bookinfo.book, bookinfo.author

where writes.BookID=book.ID and writes.AuthorID=author.ID and author.LastName ='King'