Exercise 1. SELECT

Recall the relational schema for Library from Practice 3/Exercise 1:

**Reader( ID, LastName, FirstName, Address, BirthDate)  
Book ( ISBN, Title, Author, PagesNum, PubYear, PubName)  
Publisher ( PubName, PubKind)  
Category ( CategoryName, ParentCat)  
Copy ( ISBN, CopyNumber, Shelf, Position)  
Borrowing ( ReaderNr, ISBN, CopyNumber, ReturnDate)  
BookCat ( ISBN, CategoryName )**

Write SQL queries for the following questions:

1. Show all book titles together with publisher names.
2. Which book has the largest number of pages?
3. Which authors have written more than 5 books?
4. Which books have more than twice as many pages as the average number of pages for all books?
5. Which categories contain subcategories?
6. \* Which author (assume authors' names to be unique) has the maximum number of books written?
7. \* What readers have borrowed all the books (not copies) authored by "Mark Twain"?
8. \* Which books has more than one copy?
9. \*\* What are the ten oldest books?
10. \*\* Enumerate all categories under “Sports” category (on any distance from it).

## Решение

1. SELECT Book.Title, Book.PubName FROM Book;

2. SELECT Book.ISBN  
FROM Book  
WHERE Book.PagesNum = (SELECT MAX(Book.PagesNum) FROM Book);

3. SELECT Book.Author  
FROM Book  
GROUP BY Book.Author  
HAVING COUNT(\*) > 5;

4. SELECT Book. ISBN  
FROM Book  
WHERE Book.PagesNum > 2 \* (SELECT AVG(Book.PagesNum) FROM Book);

5. SELECT cat.CategoryName  
FROM Category cat  
WHERE EXISTS(SELECT \* FROM Category cat2 WHERE cat2.ParentCat == cat.CategoryName);

6. SELECT Book.Author  
FROM Book  
GROUP BY Book.Author  
HAVING COUNT(\*) = (SELECT MAX(c)  
 FROM (SELECT count(\*) AS c  
 FROM Book  
 GROUP BY Book.Author));

7. SELECT reader.ID  
FROM Reader reader  
WHERE (SELECT COUNT(DISTINCT B.ISBN)  
 FROM Book B  
 JOIN Borrowing B2 on B.ISBN = B2.ISBN  
 WHERE B2.ReaderNr = reader.ID  
 AND B.Author = 'Mark Twain') =  
 (SELECT COUNT(allbooks.ISBN) as num  
 FROM Book allbooks  
 WHERE allbooks.Author == 'Mark Twain');

8. SELECT Book.ISBN  
FROM Book  
 JOIN Copy ON Book.ISBN = Copy.ISBN  
GROUP BY Copy.ISBN  
HAVING COUNT(\*) > 1;

9. SELECT Book.ISBN  
FROM Book  
ORDER BY Book.PubYear  
LIMIT 10;

10. WITH RECURSIVE Categoryunder(x) AS (  
 SELECT Category.CategoryName  
 FROM Category  
 WHERE Category.ParentCat = 'Sport'  
 UNION  
 SELECT Category.CategoryName  
 FROM Category  
 JOIN Categoryunder cat ON Category.ParentCat = cat.x  
)

Exercise 2. DML

Write SQL queries for the following actions:

1. Add a Borrowing record for reader ‘John Johnson’ and the book with ISBN 123456 and the copy number 4.
2. Delete all books with the Year of publish greater than 2000.
3. Change the return date for all the books of the category "Databases" starting from 01.01.2016 so that they will be on borrow for 30 days longer (Assume that it’s possible to add numbers to dates in SQL).

## Решение

1. INSERT INTO Borrowing(ReaderNr, ISBN, CopyNumber)  
SELECT Reader.ID, '123456', 4 FROM Reader WHERE Reader.FirstName = 'John' AND Reader.LastName = 'Johnson' LIMIT 1;

2. DELETE FROM Book WHERE Book.PubYear > '2000';

3. UPDATE Borrowing  
SET ReturnDate = AddDays(ReturnDate, 30)  
WHERE (Borrowing.ISBN, Borrowing.ReaderNr, Borrowing.ReturnDate) IN (SELECT B.ISBN, B.ReaderNr, B.ReturnDate  
 FROM Borrowing B  
 JOIN Book B2 on B2.ISBN = B.ISBN  
 JOIN BookCat BC on B2.ISBN = BC.ISBN  
 WHERE B.ReturnDate >= '2016-01-01'  
 AND BC.CategoryName = 'Databases');

(assuming that “AddDays” function can add int amount of days to date and return the new one)

Exercise 3. Reverse engineering

Consider the following relational schema:

Student( StudID, Name, Semester )

Test( StudID, LectNr, ProfNr, Grade )

Lecture( LectNr, Title, Credit, ProfNr )

Professor( ProfNr, Name, Room )

Describe in English results of the following queries:

a) SELECT s.Name, s.StudID FROM Student s  
WHERE NOT EXISTS (  
SELECT \* FROM Test c WHERE c.StudID = s.StudID AND c.Grade >= 4.0 ) ;

b) ( SELECT p.ProfNr, p.Name, sum(lec.Credit)  
FROM Professor p, Lecture lec  
WHERE p.ProfNr = lec.ProfNr  
GROUP BY p.ProfNr, p.Name)  
UNION  
( SELECT p.ProfNr, p.Name, 0  
FROM Professor p  
WHERE NOT EXISTS (  
SELECT \* FROM Lecture lec WHERE lec.ProfNr = p.ProfNr ));

c) SELECT s.Name, p.Grade  
FROM Student s, Lecture lec, Test c  
WHERE s.StudID = c.StudID AND lec.LectNr = c.LectNr AND c.Grade >= 4  
    AND c.Grade >= ALL (  
      SELECT c1.Grade FROM Test c1 WHERE c1.StudID = c.StudID )

## Решение

a. Show all students (name and ID) who failed all tests they took (the mark for every test was less than 4)

b. Make a list of professors (professor’s number and name) and the sum of credits of the lectures they give (if some professors don’t give any lectures, their credit sum should be 0).

c. For every student (write student’s name) show the highest mark they have that is bigger than 4 (if there is no such mark, don’t include the student in the list).