SQL Aggregates

1. Consider the following relational schema:

Reader(RDNR, Surname, Firstname, City, Birthdate)
Book(ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher(PublisherName, PublisherCity)
Category(CategoryName, BelongsTo)
Copy(ISBN, CopyNumber, Shelf, Position)
Loan(ReaderNr, ISBN, Copy, ReturnDate)
BookCategory(ISBN, CategoryName)

BelongsTo refers to which parent categories the current category belongs to. Each book has a specific ISBN, and many copies of a book might be available under the same ISBN. A reader may borrow the same copy for multiple times, and each instance is recorded by its ReturnDate. All the parent categories that a book belongs to are stored in the table BookCategory.

Formulate the following queries in SQL.

(a) Which author has written the maximum number of books?

Solution:

```
SELECT Author, COUNT(ISBN) AS numberbooks
FROM Book
GROUP BY Author
HAVING numberbooks >= ALL(
SELECT COUNT(ISBN) FROM Book
GROUP BY Author);
```

(b) Which readers have borrowed at least one book (by ISBN, not copies) from the author "Philip S. Yu", but have not borrowed all the books (by ISBN, not copies) from the author "Philip S. Yu"?

SQL Aggregates

Solution:

```
SELECT Firstname, Surname
FROM Loan L, Reader R, Book B
WHERE R.RDNR = L.ReaderNr AND
L.ISBN = B.ISBN AND
Author = 'Philip S. Yu'
GROUP BY Firstname, Surname
HAVING COUNT(L.ISBN) <
(SELECT COUNT(ISBN)
FROM Book
WHERE Author = 'Philip S. Yu');
```

2. Suppose we are maintaining a database of articles published in our newspaper, the Straits Times. We have the following schema (where keys are underlined):

Article (issueID, articleID, author, title)

Citation (articleID, issueID, citedArticleID, citedIssueID)

WordAppears (wordID, issueID, articleID, position)

WordIs (wordID, wordText)

Issue (issueID, date, howManyDistributed)

Assume that dates can be compared using comparison operators (<, >, =). Assume that position is an index specifying where the word appears (1 = first word, 2 = second, etc.). Write the following queries in SQL.

(a) Find the most-cited article(s) in the newspaper's history.

Solution:

```
SELECT      citedIssueID, citedArticleID
FROM        Citation
GROUP BY    citedIssueID, citedArticleID
HAVING COUNT(*) >= ALL
                    (SELECT     COUNT(*)
                    FROM        Citation
                    GROUP BY    citedArticleID, citedIssueID)
```

(b) Find the number of citations per author for "senior" authors (i.e., an author who has at least one article that was published 10 or more years ago).

Solution:

```
SELECT      a.author, COUNT(*)
FROM        Article a, Citation c
WHERE       a.issueID = c.citedIssueID AND a.articleID = c.citedArticleID
AND EXISTS
                (SELECT     *
                FROM        Article a2, Issue i
                WHERE       a2.issueID = i.issueID AND
                            a2.author = a.author AND
                            Year(getDate()) – Year(i.date) >= 10)
GROUP BY a.author;
```