Problem Statement 6: My SQL Data Aggregation Functions

Prashast Vats- 2566781

```
CREATE TABLE books
        (
                book_id INT NOT NULL AUTO_INCREMENT,
                title VARCHAR(100),
                author_fname VARCHAR(100),
                author_lname VARCHAR(100),
                released_year INT,
                stock_quantity INT,
                pages INT,
                PRIMARY KEY(book_id)
        );

INSERT INTO books (title, author_fname, author_lname, released_year, stock_quantity, pages)
VALUES
('The Namesake', 'Jhumpa', 'Lahiri', 2003, 32, 291),
('Norse Mythology', 'Neil', 'Gaiman',2016, 43, 304),
('American Gods', 'Neil', 'Gaiman', 2001, 12, 465),
('Interpreter of Maladies', 'Jhumpa', 'Lahiri', 1996, 97, 198),
('A Hologram for the King: A Novel', 'Dave', 'Eggers', 2012, 154, 352),
('The Circle', 'Dave', 'Eggers', 2013, 26, 504),
('The Amazing Adventures of Kavalier & Clay', 'Michael', 'Chabon', 2000, 68, 634),
('Just Kids', 'Patti', 'Smith', 2010, 55, 304),
('A Heartbreaking Work of Staggering Genius', 'Dave', 'Eggers', 2001, 104, 437),
('Coraline', 'Neil', 'Gaiman', 2003, 100, 208),
('What We Talk About When We Talk About Love: Stories', 'Raymond', 'Carver', 1981, 23, 176),
("Where I'm Calling From: Selected Stories", 'Raymond', 'Carver', 1989, 12, 526),
('White Noise', 'Don', 'DeLillo', 1985, 49, 320),
('Cannery Row', 'John', 'Steinbeck', 1945, 95, 181),
('Oblivion: Stories', 'David', 'Foster Wallace', 2004, 172, 329),
('Consider the Lobster', 'David', 'Foster Wallace', 2005, 92, 343);

SELECT COUNT(*) AS total_books FROM books;
SELECT released_year, COUNT(*) AS books_released FROM books GROUP BY released_year;
SELECT SUM(stock_quantity) AS book_stock FROM books;
SELECT author_fname, author_lname, AVG(released_year) AS avg_released_year FROM books
GROUP BY author_fname, author_lname;
SELECT CONCAT(author_fname, ' ', author_lname) AS author FROM books
ORDER BY pages DESC LIMIT 1;
SELECT released_year AS year_, COUNT(*) AS `#books`, FORMAT(AVG(pages), 4) AS `avg pages`
FROM books GROUP BY released_year ORDER BY year_;
SELECT author_fname, author_lname, COUNT(*) AS duplicate_count FROM books
GROUP BY author_fname, author_lname HAVING COUNT(*) > 1;
/*duplicating using only 2 records here as duplicationg after selecting all records to duplicate makes
it so none of the records can be seen as no one row is entirely duplicated in one other row so only
2 criteria were chosen to get some duplicates.*/
/* here is code if whole column was duplicated*/
SELECT title, author_fname, author_lname, released_year, stock_quantity, pages,
COUNT(*) AS duplicate_count FROM books
GROUP BY title, author_fname, author_lname, released_year, stock_quantity, pages
HAVING COUNT(*) > 1;
SELECT * FROM ( SELECT *, ROW_NUMBER() OVER (ORDER BY book_id) AS row_number_
FROM books) AS subquery WHERE row_number_ % 2 != 0;
SELECT * FROM ( SELECT *, ROW_NUMBER() OVER (ORDER BY book_id) AS row_number_
FROM books) AS subquery WHERE row_number_ % 2 = 0;
SELECT * FROM books ORDER BY book_id LIMIT 10;
```