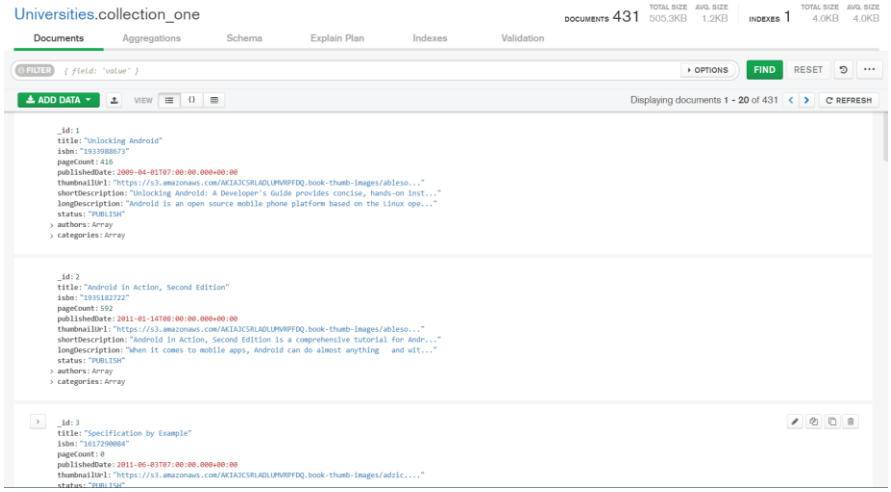
Wydział WIMiIP	Imię i nazwisko 1.Kacper Bielak	Rok 2	Grupa GL01	
Temat: Zadanie nr.2 - MongoDB				
Data wykonania 04.06.2021	Przedmiot: Bazy danych			OCENA

1. Wczytanie danych do nowej bazy

Dane zostały pobrane ze strony https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/datasets/books.json, a następnie wklejone do pliku json, który został zaimportowany do bazy.

Potwierdzenie dodania danych:

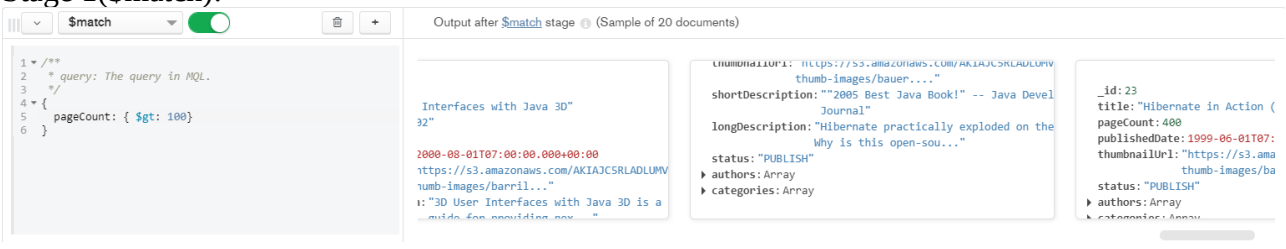


2. Zapytania z różnymi metodami agregacji

1. Wyświetlenie ile jest książek z daną ilością stron

Cel: Wyświetlenie tych książek, które mają więcej niż 100 stron(stage 1), pogrupowanie rekordów względem ilości stron oraz wyświetlenie ile jest książek mających tyle stron(stage 2) oraz posortowanie względem liczby wystąpień od największego do najmniejszego(stage 3).

Stage 1($match):

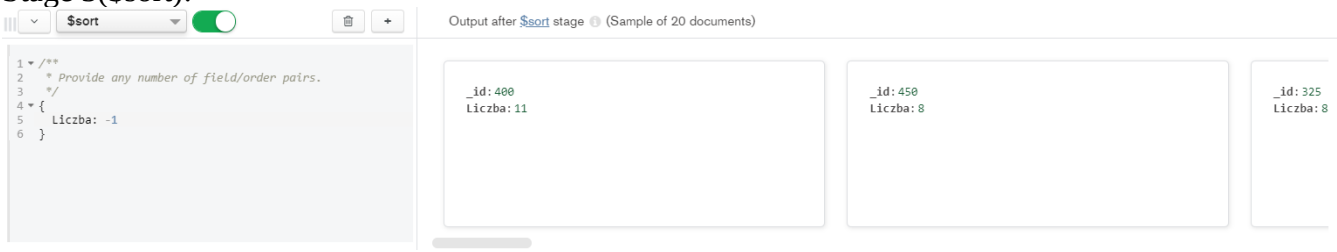


Listing:
{ pageCount: { $gt: 100}}

Stage 2($group):

Listing:
{ _id: "$pageCount", Liczba: {$sum: 1}}

Stage 3($sort):



Listing:
{ Liczba: -1}

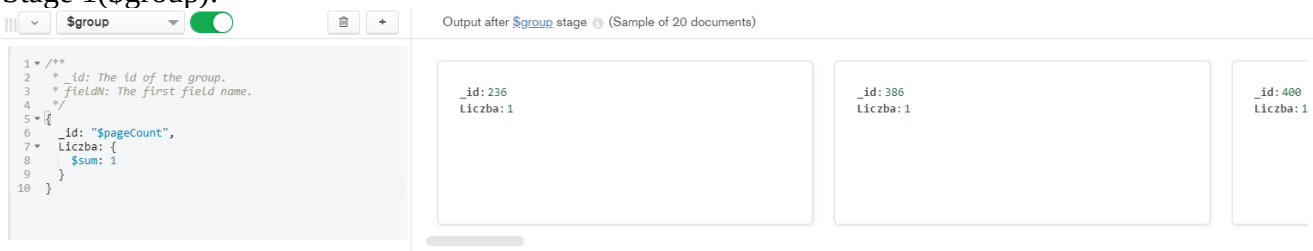
Kod SQL:
select pageCount, count(_id) as liczba from tabela group by pageCount having pageCount>100 order by liczba;

Podsumowanie:
Zapytanie spełniło początkowe założenia. To samo pytanie tylko w SQL również nie jest jakoś bardzo złożone i trudne do zaimplementowania.

2. **Wyświetlenie tej ilości stron książki, którą ma najwięcej książek**

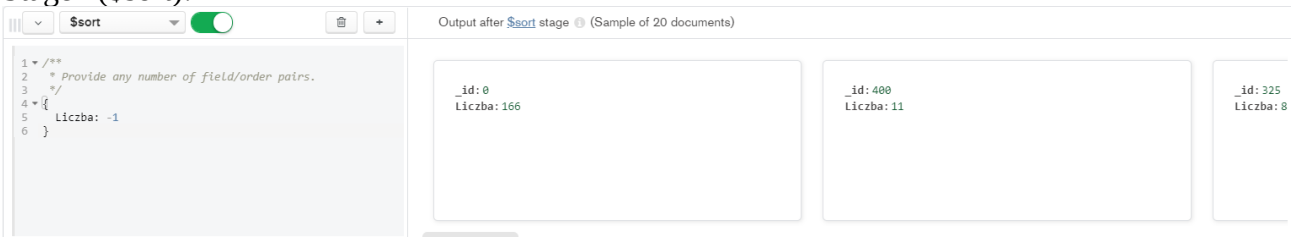
Cel: pogrupowanie rekordów względem ilości stron oraz wyświetlenie ile jest książek mających tyle stron(stage 1), posortowanie względem liczby wystąpień od największego do najmniejszego(stage 2), a następnie za pomocą metody limit wyświetlenie tylko pierwszego rekordu.

Stage 1($group):



Listing:
{ _id: "$pageCount",Liczba: { $sum: 1}}

Stage 2($sort):



Listing:
{ Liczba: -1}

Stage 3($limit):

Listing: 1

Kod SQL:
select pageCount, count(_id) as liczba from tabela group by pageCount order by liczba limit 1;

Podsumowanie:
Początkowe założenia zostały spełnione, kod sql również nie wydaje się być skomplikowany.

3. **Zapytanie do rozbicia tabeli autorów**

Cel: Zapytanie ma wyświetlić każdą książke tyle razy ile ma autorów z zaznaczeniem autora(Stage 1 - $unwind).

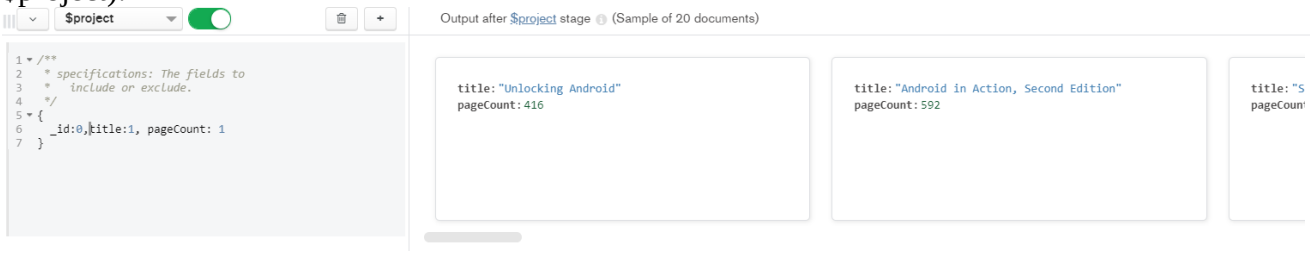
Listing: { path: "$authors", includeArrayIndex: "arrayIndex", preserveNullAndEmptyArrays: false}

Kod SQL:
select * from books inner join authors on books.id=authors.id_ksiazki;

Podsumowanie:
Początkowe założenia zostały spełnione, w kodzie sql trzeba było się zastanowić jakiego joina użyć, wybrano inner join. W celu weryfikacji tego kodu należałoby to przetestować.

4. **Wyświetlenie tylko tytułu książki i ilości stron**

Cel: zapytanie ma zwrócić wszystkie rekordu, ale tylko tytul i ilość stron książki(stage 1 - $project).



Listing: { _id:0,title:1, pageCount: 1}

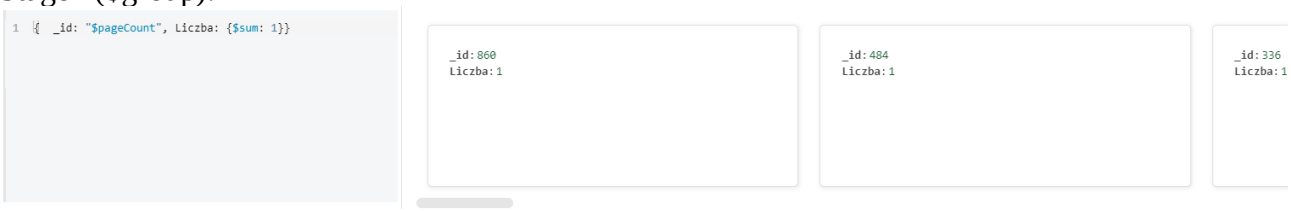
Kod SQL:
select title, pageCount from books;

Podsumowanie:
Początkowe założenia zostały spełnione, kod sql również jest bardzo prosty.

5. **Utworzenie nowej kolekcji z informacjami ile jest książek z daną liczbą stron**

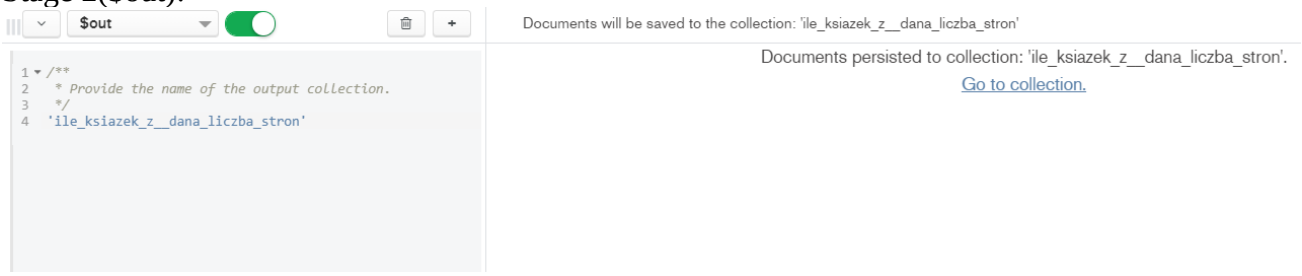
Cel: pogrupowanie rekordów względem ilości stron oraz wyświetlenie ile jest książek mających tyle stron(stage 1), a następnie zapisanie wyników do nowej kolekcji(stage 2).

Stage 1($group):



Listing: { _id: "$pageCount", Liczba: {$sum: 1}}

Stage 2($out):



Listing: 'ile_ksiazek_z__dana_liczba_stron'

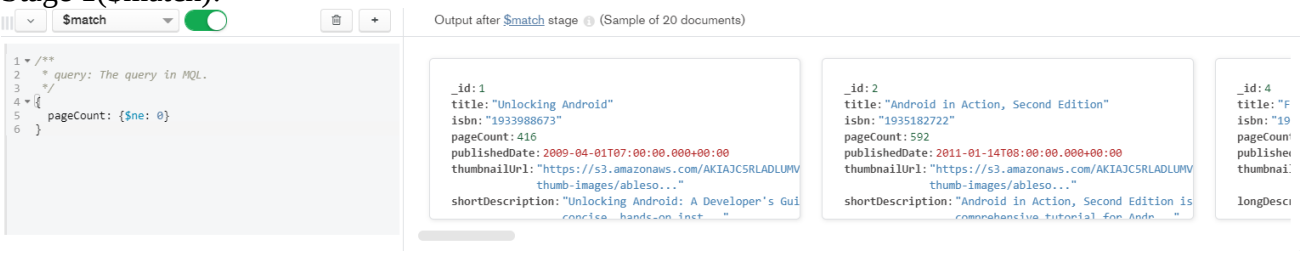
Kod SQL:
create view ile_ksiazek_z__dana_liczba_stron as
select pageCount, count(_id) as liczba from tabela group by pageCount order by liczba;

Podsumowanie:
Początkowe założenia zostały spełnione. Odpowiednikim meotdy $out mogą być po prostu widoki w sql.

6. Wyświetlenie książki z największą ilością stron w danym przedziale stron

Cel: Wyświetlenie książki, która ma ilość stron w przedziale od 0(stage 1) do 300(stage 2) z największą ilością stron pogrupowane według statusu(stage 3).

Stage 1($match):



Listing: { pageCount: {$ne: 0} }

Stage 2($match):

Listing: { pageCount: {$lt: 300} }

Stage 3($group):

Listing: {_id: "$status",max_page: {$max: "$pageCount"}}

Kod SQL:
select status, max(pageCount) as max_page from books group by status having pageCount<300 and pageCount is not null;

Podsumowanie:
Początkowe założenia zostały spełnione. W celu weryfikacji poprawności kodu w SQL należałoby go wykonać.

7. Dodanie do każdego rekordu nowego pola

Cel: dodanie do każdej książki nowego pola, które będzie równe ilości stron książki powiększonej o dwa – np. zapmniano policzyć razem ze stroną tytułową i końcową.

Stage($addFields):

Listing: { dodatkowe_pole:{ $add: ["$pageCount",2]} }

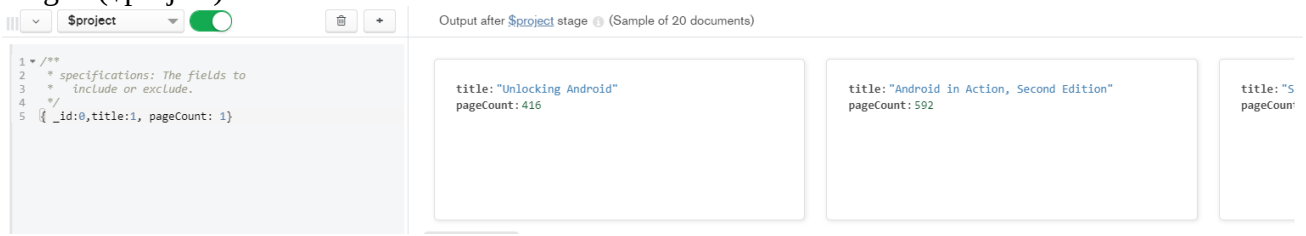
Kod SQL:
alter table books add dodatkowe_pole int;
update books set dodatkowe_pole=pageCount+2 where 1;

Podsumowanie:
Początkowe założenia zostały spełnione. W celu weryfikacji poprawności kodu w SQL należałoby go wykonać.

8. Wypisanie 5 losowych dokumentów

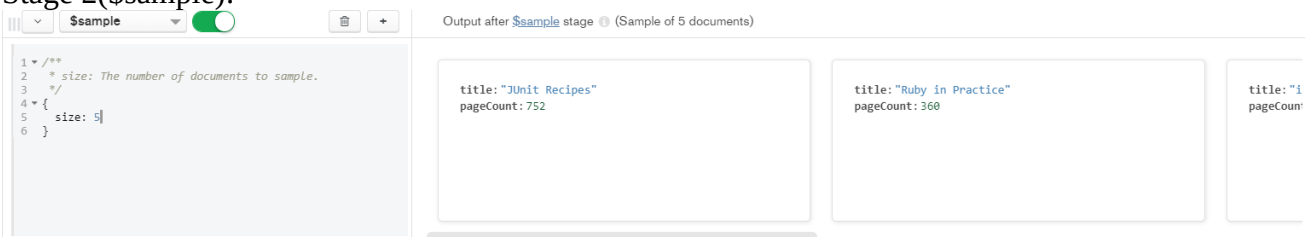
Cel: Wypisanie tytułów i ilości stron książek(Stage 1), po czym wylosowanie z nich 5 przypadkowych rekordów(Stage 2).

Stage 1($project):



Listing: { _id:0,title:1, pageCount: 1}

Stage 2($sample):



Listing: { size: 5 }

Kod SQL:
select title, pageCount from books order by rand() limit 1;

Podsumowanie:
Początkowe założenia zostały spełnione. Kod sql również nienależy do najtrudniejszych.

WNIOSKI

Po wykonaniu tego ćwiczenia można stwierdzić, że metody agregacji bardzo ułatwiają pracę z bazą danych. Metody te są bardzo łatwe do napisania i przejrzyste. Ponadto łącząc je ze sobą można otrzymać niesamowite efekty. Nie trzeba martwić się o stawianie złożonych warunków czy złączeń między tabelami tak jak miało to miejsce w sql. Powyższe przykłady agregacji pozwoliły zrozumieć ich działanie. Jeśli chodzi o ich odpowiedniki w sql to nie można założyć, że są one w pełni poprawne. Aby potwierdzić poprawne ich działanie należałoby je wykońać np. w MySQL. MongoDB Compass bardzo wspomaga pisanie agregacji, a jeszcze bardziej aggregation pipeline dzięki świetnemu interfejsowi w którym można wybrać daną metodę po czym dodać kolejny stage, który operuje już tylko na rekordach zwróconych przez poprzedni.