

[빅데이터 중간고사 예상문제]

1. Structured Data와 Unstructured Data의 차이점을 5가지 쓰시오.
2. Big Data의 정의를 특성 4가지를 들어 쓰시오.
3. Big Data 분석학의 목표 4가지를 쓰시오.
4. Big Data 프레임워크의 정의와 특징 4가지를 쓰시오.
5. 스트리밍 데이터의 정의와 예시, 빅데이터 시스템의 정의와 2가지 방식에 대해 쓰시오.
6. Centralized Storage와 Decentralized Storage의 차이점을 쓰시오.
7. Relation Model의 특징과 단점 4가지를 쓰시오.
8. NoSQL의 정의 3가지를 쓰시오.
9. NoSQL의 특징 3가지를 쓰시오.
10. NoSQL의 유형 4가지를 그림과 함께 설명하시오.
11. Document Store DB와 Relational DB의 차이점을 쓰시오.
12. 빅데이터 프로세스와 MongoDB의 활용 범위를 쓰시오.
13. MongoDB 구조에 대해 쓰시오.
14. Relational DB와 MongoDB의 차이점을 쓰시오.
15. MongoDB Document의 1 : N 패턴을 구현할 수 있는 2가지 방법을 쓰시오.
16. MongoDB Document의 M : N 패턴을 구현할 수 있는 2가지 방법을 쓰시오.
17. MongoDB Document의 트리 패턴을 구현할 수 있는 2가지 방법과 특성을 쓰시오.
18. MongoDB Document의 Dynamic Field 패턴 정의와 특성을 쓰시오.
19. Embedded Document와 Linked Document의 차이점을 5가지 쓰시오.
20. \$regex를 활용한 정규식 표현 양식 3가지를 쓰시오.

[MongoDB 퀴리문]

1. DB 생성
2. 현재 DB
3. DB 목록
4. DB Collection Document 삽입
5. 현재 DB 삭제
6. Collection 생성
7. Collection 목록
8. Collection 삭제
9. Capped Collection 정의를 쓰시오.
10. Capped Collection 생성
11. Collection의 Document 개수 제한 생성
12. Capped 여부 확인
13. 기존 Collection Capped로 전환
14. Capped Collection 옵션 변경
15. Collation 정의를 쓰시오.
16. Time Series Collection 정의와 특성을 쓰시오.
17. Clustered Collection 정의와 특성을 쓰시오.

18. Document 1개 삽입

19. Document 여러 개 삽입

20. Document에서 _id 필드의 역할 쓰시오.

21. Document 전체 쿼리

22. Document 쿼리 형식에 맞춰 출력

23. 'inventory' Collection에서 'qty' 필드 값이 25인 Document 쿼리

24. 'inventory' Collection에서 'qty' 필드 값이 25 이상인 Document 쿼리

25. 'inventory' Collection에서 'qty' 필드 값이 25가 아닌 Document 쿼리

26. 'inventory' Collection에서 'status' 필드 값이 [A, D] 안에 있는 Document 쿼리

27. 'status'가 D, 'qty'가 75 이하인 Document 쿼리

28. 'status'가 A이거나, 'qty'가 30 미만인 Document 쿼리

29. 'qty'가 75보다 큰 것이 아닌 Document 쿼리

30. Embedded Document 쿼리 시 Dot Notation 사용이 권장되는 이유를 쓰시오.

31. Projection의 정의와 값에 따른 사용법

32. 'inventory' Collection에서 'item' 필드가 paper인 Document의 size만 출력

33. 'item' 필드가 paper인 Document의 값을 paperless로 수정 (첫 번째 Document만)

33. 'item' 필드가 paper인 Document의 값을 paperless로 수정 (Document 전체)

34. Document 삭제

35. Document 중 값이 일치하는 첫 번째 Document만 삭제

36. "restaurants" Collection에서 모든 Document에 대해 restaurant_id, name, borough, cuisine 필드 출력

37. "restaurants" Collection에서 모든 Document에 대해 _id 필드 제외한 ..., zip code(Nested) 필드 출력

38. borough가 "Queens"에 속하는 식당의 restaurant_id, name, borough, cuisine 필드 출력

39. borough가 "Queens"에 속하지 않는 식당의 restaurant_id, name, borough, cuisine 필드 출력

40. "grades.grade" 값이 "A"인 식당의 restaurant_id, name, borough, cuisine 필드 출력

41. "American" cuisine을 제공하지 않고 grades.score 값이 70 초과인 식당 출력

42. borough가 "Staten Island", "Queens", "Bronx", "Brooklyn" 중 하나에 속하는 식당의 ..., cuisine 필드 출력

43. borough가 "Staten Island", "Queens", "Bronx", "Brooklyn"에 속하지 않는 식당의 ..., cuisine 필드 출력

44. "American" cuisine 제공하지 않고 "grades.score" 70 초과하고 borough가 "Brooklyn" 속하지 않는 식당

45. "Hamburgers" cuisine 제공하고 "grades.grade" 값이 "A"이고 borough가 "Manhattan", "Queens", "Staten Island", "Bronx"에 속하지 않는 식당의 _id 필드 제외한 restaurant_id, ..., cuisine 필드 출력

```
- db.inventory2.insert([
  {item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [14, 21]},
  {item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [14, 21]},
  {item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [14, 21]},
  {item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [22.85, 30]},
  {item: "paper", qty: 45, tags: ["blue"], dim_cm: [10, 15.25]}
])
```

45. tags가 red, blank인 Document (Dot Notation 사용 X)

46. tags에 red가 포함되어 있는 Document

47. dim_cm 필드의 두 번째 Element 값이 25보다 큰 Document

```
- db.inventory3.insert([
    {item: "NORWAY", instock: [{warehouse: "A", qty: 5}, {warehouse: "C", qty: 15}]},
    {item: "notebook", instock: [{warehouse: "C", qty: 5}]},
    {item: "paper", instock: [{warehouse: "A", qty: 60}, {warehouse: "B", qty: 15}]},
    {instock: [{warehouse: "A", qty: 40}, {warehouse: "B", qty: 5}]},
    {item: null, instock: [{warehouse: "B", qty: 15}, {warehouse: "C", qty: 35}]}
])
```

45. warehouse가 A, qty가 5인 Document (Dot Notation 사용 X)

46. qty 값이 20 이상인 Document

47. instock 배열 첫 번째 Embedded Document 요소에 'qty' 필드 존재하고 값이 5인 Document 반환

48. instock 배열 첫 번째 Embedded Document 요소에 'qty' 필드 존재하고 값이 20 이하인 Document 반환

49. item 필드가 없거나 null인 Document

50. item 필드의 값이 존재하고 그 값이 null이 아닌 Document

51. item 필드가 존재하지 않는 Document

52. item 필드가 존재하는 Document (null 상관없이)

```
- db.grades.insertMany([
    {"_id": 1, name: "Alice King", classAverage: 87.33333},           // double
    {"_id": 2, name: "Bob Jenkins", classAverage: "83.52"},
    {"_id": 1, name: "Cathy Hart", classAverage: "94.06"},
    {"_id": 1, name: "Drew Williams", classAverage: NumberInt("93")}, // 32-bit Integer
])
```

53. classAverage 필드 자료형이 string인 Document

54. classAverage 필드 자료형이 number인 Document

54. classAverage 필드 자료형이 string 혹은 double인 Document

```
- db.inventory.insertMany([
    {item: "journal", qty: 25, size: {h: 14, w: 21, uom: "cm"}, status: "A"},
    {item: "notebook", qty: 50, size: {h: 8.5, w: 11, uom: "in"}, status: "A"},
    {item: "paper", qty: 100, size: {h: 8.5, w: 11, uom: "in"}, status: "D"},
    {item: "planner", qty: 75, size: {h: 22.85, w: 30, uom: "cm"}, status: "D"},
    {item: "paper", qty: 45, size: {h: 10, w: 15.25, uom: "cm"}, status: "A"}
])
```

55. item 필드 값에 'paper' 키워드 포함한 Document

56. item 필드 값이 'note' 키워드로 시작하는 Document

57. item 필드 값이 'nal' 키워드로 끝나는 Document

58. item 필드 값이 'PAPER' 키워드 포함한 Document, 대소문자 구분 X

59. 'status' 값이 A이고, 'qty' 값이 30보다 작거나 'item' 값이 'p' 키워드로 시작하는 Document

```
"address": {
  "building": "1007",
  "coord": [ -73.856077, 40.848447 ],
  "street": "Morris Park Ave",
  "zipcode": "10462"
},
"borough": "Bronx",
"cuisine": "Bakery",
"grades": [
  { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
  { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
  { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 }
],
{
  "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
  { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
},
"name": "Morris Park Bake Shop",
"restaurant_id": "30075445"
```

60. "restaurants" Collection에서 "grades.grade" 값이 "A"인 식당의 restaurant_id, ..., cuisine 필드를 출력
61. "American" cuisine을 제공하지 않고 grades.score 값이 70 초과인 식당을 출력
62. "address.coord" 필드에서 배열 요소인 latitude = -73.85이고, longitude = 40.84인 모든 식당을 출력
63. "American" 또는 "Chinese" cuisine을 제공하고, "grades.score"이 60보다 크고, latitude가 -74 미만
64. "grades" 배열의 두 번째 요소가 "grade" 필드 값 "A"를 포함하고 "score"이 9, ISODate가 ...인 식당
65. "grades" 배열의 8번째 요소가 "score" 값 30을 초과하는 식당의 restaurant_id, name, score 출력
66. "name" 필드에 "Pizza" 키워드를 포함하는 식당 출력
67. "borough" 필드가 "Staten Island" 또는 "Queens"이며, "name" 필드가 "Wen" 세 글자로 시작하는 식당
68. 모든 Document의 "address" 필드가 "street" 필드를 포함하고 있는지 아닌지 확인
69. 모든 Document의 "address.coord" 필드 값이 배열 형식인지 확인
70. Airbnb DB의 "reviews" Collection에서 "weekly_price" 필드가 존재하고,
"name" 필드가 "Pr" 키워드로 시작하며, "review_scores.review_scores_rating" 값이 80 이상인 Document