

## データベース 計算機実習資料 (MongoDB)

### 1 MongoDB

MongoDB はドキュメント指向の NoSQL データベースシステムであり、以下のような特徴を持つ。

- (1) JSON (JavaScript Object Notation) に似た BSON (Binary JSON) の形式でデータを格納する。
- (2) スキーマ定義が不要な NoSQL のデータベースシステムの一つである。
- (3) JavaScript に似た構文でデータ操作ができる。
- (4) スケールアウトにより分散サーバ環境でデータを効率的に処理する機構を持つ。

### 2 MongoDB 実習

#### 実習手順

実習では、Firefox などの Web ブラウザで以下の URL にアクセスして問合せ文を入力することで、データベースへの問合せを行うようにしている。

<http://dbs.exp.ics.es.osaka-u.ac.jp/mongodb/>

注意: 上記のサイトは情報科学科の教育用計算機システムの実験・演習端末のネットワーク上にあるため、学科外からアクセスすることができない。

入力: 上記の URL を入れると、「MongoDB Command Input」と表示が出て、入力可能な Form (テキスト入力) 領域が表示される。

ここに、任意の MongoDB 操作文を自由書式で書き込むことができる。入力終了後、入力領域の下に「submit」ボタンがあるのでこれをクリックすれば、MongoDB サーバに送られる。サーバで実行された後、結果は検索結果表示として入力文の下に表示される。

もし、入力文に何らかの不備があるときは、ブラウザの戻るボタンをクリックして前のページに戻り、入力領域の文を修正して、そのまま「submit」をクリックすることで何度も問合せ文を実行することができる。

#### 入力についての注意

入力文の終了はカッコの対応で見るので、文の終わりに特別な区切り文字を入力する必要はない。2 個以上の入力文を同時に入力することもできる (結果がそれぞれ続けて表示される)。

入力領域は空白が挿入できるところであれば、自由に改行可能である。

この実習環境は、Web サーバを経由して MongoDB サーバにつないでいる。同時に多数のユーザが入力文を実行するとサーバがダウンすることがある。MongoDB の構文エラー以外のエラー (「サーバへの接続が確立できない」など) が出力されたり、しばらく結果が返ってこなくてタイムアウトになるなどサーバの障害が疑われるときは、以下のアドレスまでメールで連絡してください。

[h-akazaw@ist.osaka-u.ac.jp](mailto:h-akazaw@ist.osaka-u.ac.jp), [matsuda@ist.osaka-u.ac.jp](mailto:matsuda@ist.osaka-u.ac.jp)

## MongoDB 問合せの実行例

### 1. データベース

以下のデータベースがあらかじめ登録されている。

```
db.emp.insertMany([
  { empno: "E01", ename: "Smith", deptno: "D01", salary: 100 },
  { empno: "E02", ename: "Morgan", deptno: "D01", salary: 70 },
  { empno: "E03", ename: "Robert", deptno: "D01", salary: 80 },
  { empno: "E04", ename: "Washington", deptno: "D02", salary: 120 },
  (以下略)
])

db.dept.insertMany([
  { deptno: "D01", dname: "Account", manager: "E01" },
  { deptno: "D02", dname: "Personnel", manager: "E04" },
  { deptno: "D03", dname: "Education", manager: "E07" },
  { deptno: "D91", dname: "Database", manager: "E91" }
])
```

### 2. 問合せの考え方

「部門名"Database"の社員の中で最も給与が高い社員の社員名を求めよ」という問合せを次の手順で処理する。

- (1) emp (社員) コレクションで deptno (部門番号) ごとにグループ化して最高給与額を求める。
- (2) (1)の結果を dept (部門) コレクションと deptno (部門番号) で結合することにより dname (部門名) を得る。
- (3) (2)の結果を emp コレクションと salary (給与額) で結合することにより最高給与額を持つ ename (社員名) を得る。
- (4) (3)の結果に対して、dname が Database であるドキュメントを選択する。
- (5) 不要なフィールドを削除する。

### 3. 問合せの記述例

```
db.emp.aggregate([
  {$group: { _id: "$deptno", max_salary: {$max: "$salary"} }},
  {$lookup: {from: "dept", localField: "_id", foreignField: "deptno", as: "max_salary_dept"}},
  {$lookup: {from: "emp", localField: "max_salary", foreignField: "salary", as: "db_max_salary"}},
  {$match: {"max_salary_dept.dname": "Database"}},
  {$project: { _id:0, max_salary_dept: { _id:0, deptno:0, manager:0 },
              db_max_salary: { _id:0, empno:0, deptno:0, salary:0 } }}
])
```

この問合せの実行結果は次のようになる。

```
{ "max_salary" : 140, "max_salary_dept" : [ { "dname" : "Database" } ], "db_max_salary" :
[ { "ename" : "Suzuki" } ] }
```

(参考) 問合せの結果がフィールドが多過ぎて読みにくいときは、問合せ文の後に `.pretty()` とつけると結果を整形して読みやすく表示してくれる。