

NoSQL MongoDB:

Описание сущностей:

1.City - информация о городе.

Содержит:

- «_id» - Object_id, идентификатор, 12 байт.
- «cityName» - string, название города, 50*2 байта.

Итого: 112 байт.

2.Roadwork - информация о дорожной работе. Содержит:

- «_id» - Object_id, идентификатор, 12 байт.
- «workName» - string, название города, 50*2 байта.
- «workAddress» - string, название города, 50*2 байта.
- «workDate» - string, название города, формат: xx\xx\xx = 8*2 байта.
- «type» - string, название города, 50*2 байта.

Итого: 328 байта.

Расчет объема:

Имеется N городов и M дорожных

работ.

Объем городов - $N \cdot 112$ байта.

Объем дорожных работ - $M \cdot 328$ байта.

Итого: $R1 = (N \cdot 112 + M \cdot 328)$ байта

Фактически объем:

$R2 = (N \cdot 112 + M \cdot 352)$ байта

Избыточность модели: $I = (N \cdot 112 + M \cdot 352) / (N \cdot 112 + M \cdot 328)$

Пусть имеется $M = 1000000$. Получаем:

Чистый объем от количества городов:

$N \cdot 112 + 1000000 \cdot 328$ (байт) =

$N \cdot 112 + 328000000$ (байт) = $N \cdot 112$ (байт)

+ 0,32 (гигабайт)

Фактический объем от количества

городов: $N \cdot 112 + 1000000 \cdot 352$ (байт) =

$N \cdot 112 + 352000000$ (байт) = $N \cdot 112$ (байт)

+ 0,33 (гигабайт)

Примеры запросов:

- `db.works.insertOne(`
-
- `)` - добавить работу

SQL

Описание сущностей:

1.City - информация о городе.

Содержит:

- «cityId» - int, идентификатор, 4 байта.
- «cityName» - string, название города, 50*2 байта.

Итого: 104 байта.

2.Roadwork - информация о дорожной работе. Содержит:

- «workId» - int, идентификатор, 4 байта.
- «workName» - string, название города, 50*2 байта.
- «workAddress» - string, название города, 50*2 байта.
- «workDate» - string, название города, формат: xx\xx\xx = 8*2 байта.
- «cityId» - int, идентификатор города, 4 байта.
- «type» - string, название города, 50*2 байта.

Итого: 324 байта.

Расчет объема:

Имеется N городов и M дорожных работ.

Объем городов - $N \cdot 104$ байта.

Объем дорожных работ - $M \cdot 320$ байта.

Итого: $R3 = N \cdot 104 + M \cdot 320$ байта

Фактический объем:

Объем городов - $N \cdot 104$ байта.

Объем дорожных работ - $M \cdot 324$ байта.

Итого: $R4 = N \cdot 104 + M \cdot 324$ байта

Избыточность модели: $I = (N \cdot 104 + M \cdot 324 \text{ байта}) / (N \cdot 104 + M \cdot 320) \text{ байта}$

Пусть имеется $M = 1000000$. Получаем:

Чистый объем от количества городов:

$N \cdot 104 + 1000000 \cdot 320 \text{ (байт)} =$

$N \cdot 104 + 320000000 \text{ (байт)} = N \cdot 104 \text{ (байт)}$

$+ 0,32 \text{ (гигабайт)}$

Фактический объем от количества

городов: $N \cdot 104 + 1000000 \cdot 324 \text{ (байт)} =$

$N \cdot 104 + 324000000 \text{ (байт)} = N \cdot 104 \text{ (байт)}$

+ 0,324 (гигабайт)

Примеры запросов:

- `INSERT INTO Work VALUES(...)` - добавить работу
- `SELECT * FROM City INNER JOIN Work ON Work.workId = City.cityId` - найти название работ по id города

Вывод:

Исходя из сделанного анализа MongoDB занимает меньше памяти по сравнению с MySQL. При этом все запросы на MongoDB выглядят менее емкими и более простыми в сравнении с MySQL. Мы делаем вывод в сторону MongoDB.