

コンテナに割り当てられたRUは、各物理パーティションに均等に割り当てられる。

10000のRUを割り当てているのに、最悪、5000程度で性能が頭打ちになる！というケースも。

300万件の  
項目

コンテナ

項目

論理パーティションが  
できるだけたくさん形成されるような  
パーティションキーを選択するとよい。

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

論理P

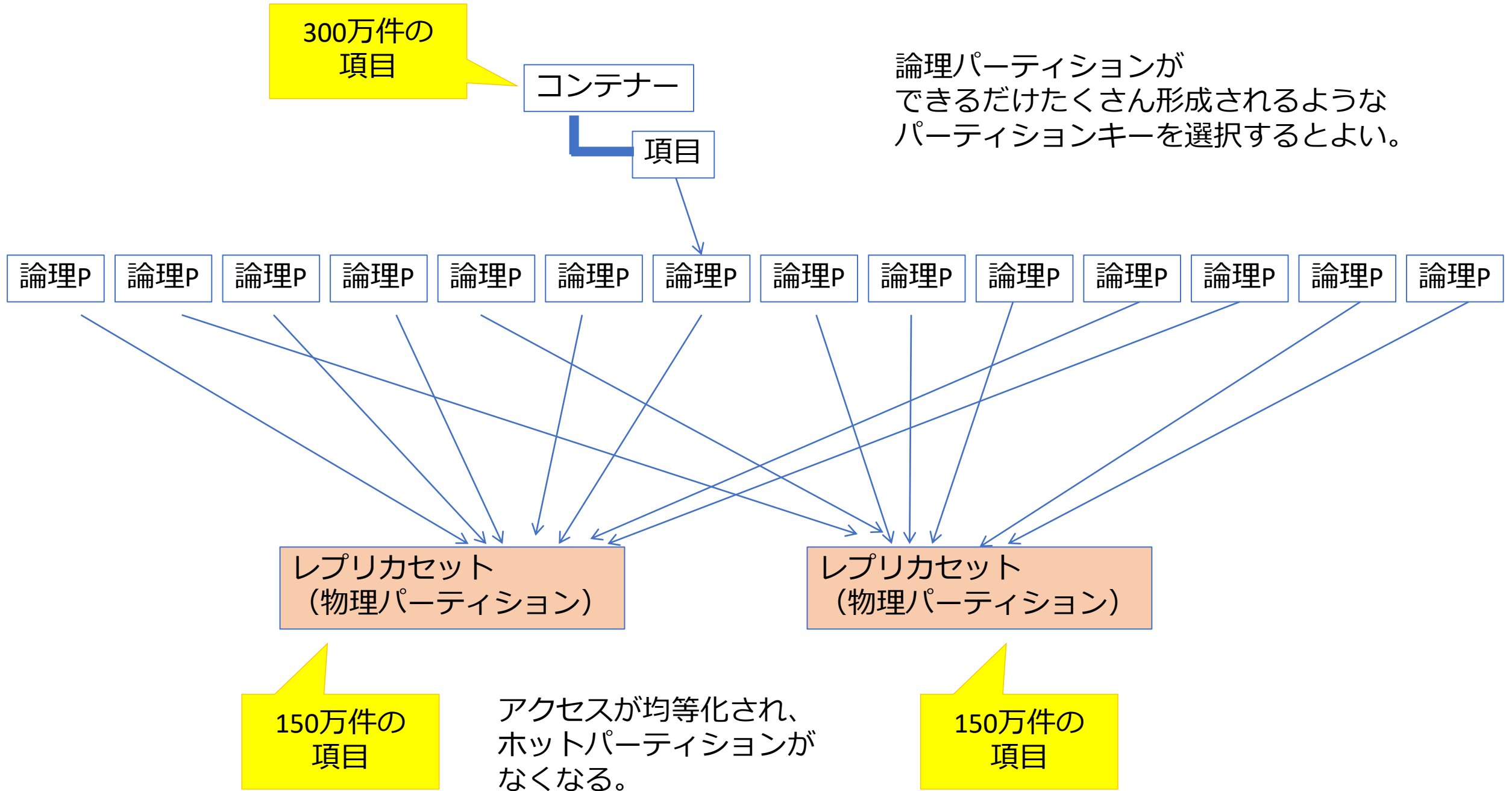
レプリカセット  
(物理パーティション)

レプリカセット  
(物理パーティション)

150万件の  
項目

アクセスが均等化され、  
ホットパーティションが  
なくなる。

150万件の  
項目



# カーディナリティ（値の種類）

ID	氏名	性別	会員種別	利用施設
1	ヤマダ	男	一般会員	上野
2	スズキ	女	プレミアム会員	品川
3	ヨシダ	男	一般会員	東京
4	サトウ	女	一般会員	上野
5	ササキ	男	プレミアム会員	品川

高い

高い

低い

低い

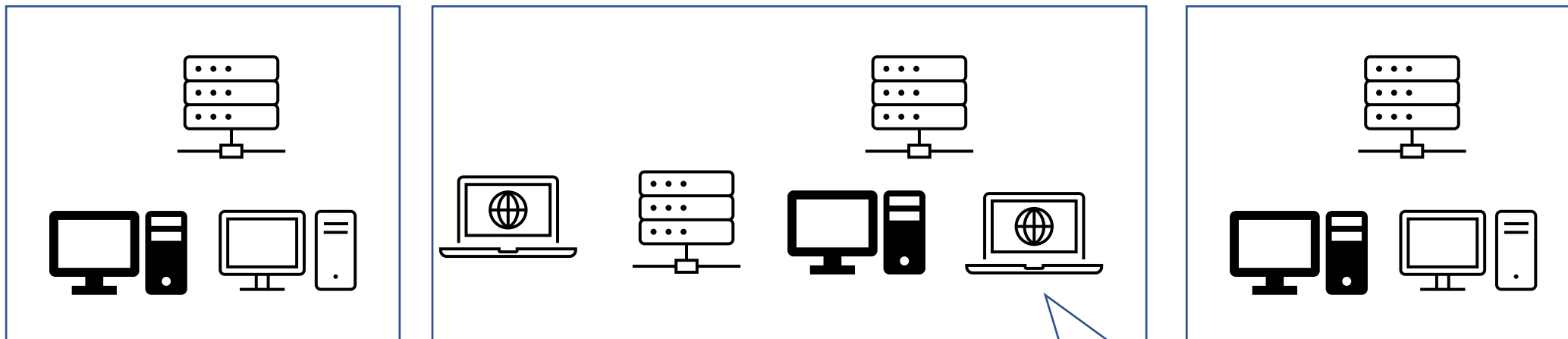
中

基本的には、**ホットパーティション**の発生を避けるため、**カーディナリティ**が高いプロパティ（列）をパーティションキーとして選択する。

ただし、上記のデータ例で「利用施設」をパーティションキーに選択した場合、

「利用施設を指定して該当するデータを取り出す」といった「**インパーティションクエリ**」が利用できるので、必ずしも、最もカーディナリティが高い列をパーティションキーに選択するとは限らない。

例：たくさんのデバイスの温度データをセンサーで測定する



様々なデバイスの  
温度データを  
短い間隔で測定し  
Cosmos DBに蓄積

状況:「場所」をパーティションキーにすることを検討している。

問題:「場所」の値はそれほど多くないので、データがうまく分散しない。  
(特定の物理パーティションにアクセスが集中してしまう)

パーティションキーを  
「作成」する

※もとのデータに  
ないプロパティを  
プログラムで作って  
データに付与し、  
それをパーティション  
キーとして活用する

```
{  
  "location-device-id": "server-room-1.1",  
  "device-id: 1,  
  "location": "server-room-1",  
  "temperature": 45,  
  "created-at": "2021-01-25 12:34:56"  
}
```

合成

```
{  
  "location-rand": "server-room-1.372",  
  "device-id: 1,  
  "location": "server-room-1",  
  "temperature": 45,  
  "created-at": "2021-01-25 12:34:56"  
}
```

乱数

```
{  
  "location-code": "server-room-1.202101010001",  
  "device-id: 1,  
  "location": "server-room-1",  
  "temperature": 45,  
  "created-at": "2021-01-25 12:34:56"  
}
```

他の値の  
何らかの計算結果

※ここでは「id」については考慮しない。