
Hướng dẫn: Kết hợp Search local 2000 dòng và chia dữ liệu SharePoint theo lô

Phiên bản 07/01/2026

1. Mục tiêu

Bạn muốn tránh gọi Flow quá nhiều (chậm) nên dự định dùng hàm Search() trong Power Apps. Vì Search() thường chỉ áp dụng trên dữ liệu đã tải về (giới hạn 500/2000 dòng tùy cấu hình), một cách làm là chia danh sách lớn (ví dụ 10.000 items) thành nhiều phần 2.000 items, tải từng phần về, rồi mới Search local và hiển thị ra Gallery.

2. Ý tưởng chia 10.000 dòng thành 5 phần (mỗi phần 2.000)

Nguyên tắc: Filter() theo các điều kiện đơn giản (ví dụ theo ID range) là delegable với SharePoint, nên Power Apps có thể tải đúng từng phần 2.000 items về client. Sau đó bạn chạy Search() trên từng phần (collection) và gộp kết quả để hiển thị.

3. Ví dụ Power Fx - chia theo ID range rồi Search

3.1. Tải dữ liệu theo từng lô 2.000 items (Concurrent)

```
Concurrent(  
    ClearCollect(colP1, Filter(Requests, ID <= 2000)),  
    ClearCollect(colP2, Filter(Requests, ID > 2000 && ID <= 4000)),  
    ClearCollect(colP3, Filter(Requests, ID > 4000 && ID <= 6000)),  
    ClearCollect(colP4, Filter(Requests, ID > 6000 && ID <= 8000)),  
    ClearCollect(colP5, Filter(Requests, ID > 8000 && ID <= 10000))  
);  
  
// Optional: merge collections for a single Search()  
Clear(colAll);  
Collect(colAll, colP1);  
Collect(colAll, colP2);  
Collect(colAll, colP3);  
Collect(colAll, colP4);  
Collect(colAll, colP5);
```

3.2. Search trên dữ liệu đã tải (cách 1: Search trên colAll)

```
ClearCollect(  
    colResult,  
    Search(colAll, txtSearch.Text, "Title", "Description")  
);  
  
// Gallery.Items = colResult
```

3.3. Search từng lô rồi gộp kết quả (cách 2: đỡ RAM hơn)

```
Clear(colResult);  
  
Collect(colResult, Search(colP1, txtSearch.Text, "Title", "Description"));  
Collect(colResult, Search(colP2, txtSearch.Text, "Title", "Description"));  
Collect(colResult, Search(colP3, txtSearch.Text, "Title", "Description"));
```

```
Collect(colResult, Search(colP4, txtSearch.Text, "Title", "Description"));
Collect(colResult, Search(colP5, txtSearch.Text, "Title", "Description"));

// Gallery.Items = colResult
```

4. Lưu ý quan trọng (bẫy thường gặp)

- ID không liên tục (do xóa item). Chia theo ID range vẫn dùng được, nhưng số item mỗi lô có thể lệch.
- Danh sách thay đổi (tăng vượt 10.000) thì bạn phải cập nhật thêm lô/range.
- Tải 10.000 items vẫn tốn thời gian. Bạn tránh Flow, nhưng đổi lại là app phải tải dữ liệu nhiều về client (chậm/mất RAM).
- Search() chạy local nên phụ thuộc vào dữ liệu đã tải; nếu chưa tải đủ lô thì kết quả không đầy đủ.

5. Gợi ý kết hợp để nhanh mà vẫn đúng

Nếu mục tiêu của bạn là tốc độ khi người dùng gõ và vẫn đúng khi dataset lớn, thường dùng 2 tầng:

- Tầng nhanh (local): Khi đã filter đủ hẹp (chắc chắn ≤ 2000 items) thì tải về collection và Search local.
- Tầng chuẩn (server): Khi không filter hoặc dataset rộng thì gọi Flow để search/paging chính xác.

Mẫu rule đơn giản trong Power Apps

```
Set(
    varHasFilter,
    !IsBlank(ddRoot.Selected.Value) ||
    !IsBlank(ddLv1.Selected.Value) ||
    !IsBlank(ddLv2.Selected.Value) ||
    CountRows(colLv1) > 0
);

If(
    /* No filter + has search => use Flow */
    !varHasFilter && !IsBlank(txtSearch.Text),
    /* call Flow ... */,
    /* Has filter => only use local Search when dataset is loaded and small */
    If(
        varLocalReady,
        ClearCollect(colData, Search(colBase, txtSearch.Text, "Title", "Description")),
        Clear(colData)
    )
);
```

Nếu bạn muốn giảm độ trễ khi gọi Flow, bạn có thể áp dụng debounce (Timer 500-800ms) để chỉ gọi Flow sau khi người dùng ngừng gõ.