

• POD MASKĄ

Jak notatki **linkują do siebie** — i co prowadzi z powrotem?

Powiązane notatki to najstarsza obietnica w zarządzaniu wiedzą osobistą i ta najczęściej po cichu łamana: funkcja istnieje, sekcja się renderuje, a żaden odnośnik nigdy nie powstaje. **SumizAI miał dokładnie ten błąd**, a naprawa polegała na zmianie tego, kto jest właścicielem odnośnika.

Aktualizacja: 2026-08-14 8 pytań 3 min czytania

Q. Gdzie żyje odnośnik między dwiema notatkami?

W treści notatki, jako zwykły link Markdown: `[Strategie dzielenia tekstu](strategie-dzielenia-tekstu.md)`.

Nie w tabeli bazy danych i nie w sekcji *Powiązane* doklejonej na dole. Odnośnik to fraza w zdaniu, która akurat gdzieś prowadzi — tak działają odnośniki w tekście i tak działają w każdym narzędziu Markdown, którym możesz otworzyć plik.

To była korekta wcześniejszego projektu. Odnośniki były kiedyś wierszami w tabeli, a sekcja *Powiązane* renderowała się z nich — i w prawdziwych vaultach nie było ani jednego, bo żadna ścieżka zapisu tej tabeli nie wypełniała. Sekcja renderowała pustą listę w nieskończoność. Przeniesienie własności do tekstu sprawiło, że odnośnik albo jest w pliku, albo nie istnieje w ogóle.

Q. Kto decyduje, co zostaje zlinkowane?

Model, który pisze notatkę, pracując z listy, poza którą nie wolno mu wyjść.

Dostaje mapę do ośmiu kandydatów — tytuł do nazwy pliku — wziętych z wyników wyszukiwania dla materiału, o którym pisze. Ma wpleść odnośnik w zdanie, w którym połączenie jest prawdziwe, a nie wyprodukować listę.

Potem przy odbiorze biegnie filtr: **każdy cel `.md` spoza mapy jest usuwany**. Zewnętrzne odnośniki `http` zostają.

Q. Dlaczego wycinać, zamiast zaufać modelowi?

Bo model poproszony o nazwę pliku wyprodukuje prawdopodobną. Poproś o odnośnik do notatki o kosztach retrievalu, a całkiem możliwe, że dostaniesz `koszty-retrievalu.md` — całkowicie sensowną nazwę pliku, który nie istnieje.

Martwy odnośnik jest gorszy niż jego brak, i to w konkretny sposób: wygląda jak trop. Klikasz, nic tam nie ma, i od tej pory nie ufasz żadnemu innemu odnośnikowi na stronie.

Ograniczenie jest więc celowo surowe. Model wybiera, gdzie w tekście należy się połączenie, a aplikacja decyduje, które cele są prawdziwe.

Q. Czym są backlinki i skąd się biorą?

Pod każdą notatką są dwie listy: odnośniki wychodzące z tej notatki i notatki, które prowadzą do niej.

Lista wychodząca jest czytana z pliku, bo w pliku odnośniki żyją. Lista przychodząca pochodzi z kolumny pochodnej w indeksie — to pytanie, na które sama treść nie odpowie bez czytania każdego innego pliku w vaultie przy każdym otwarciu strony.

Ta kolumna jest pochodna, nie autorytatywna. Da się ją w każdej chwili odbudować z plików przez przeindeksowanie, co utrzymuje regułę w mocy: źródłem prawdy jest plik, a indeks tylko odpowiada szybciej.

Q. Co dzieje się z odnośnikami, gdy zmienię tytuł notatki?

Są przepisywane. Nazwa pliku to slug tytułu, więc zmiana tytułu przenosi plik — a potem każde odwołanie `[(nazwa.md)]` do niego w reszcie vaulta zostaje zaktualizowane.

Kolejność jest dobrana tak, żeby awaria nie zostawiła Cię w połowie: zapisz nowy plik, przepisz odwołania, skasuj stary. Jeśli cokolwiek zawiedzie, przepisane odwołania są cofane, a nowy plik usuwany.

Jest tu świadomie przyjęty kompromis. Nazwy były kiedyś stałe, co czyniło zmiany tytułów trywialnymi, a katalogi nieczytelnymi. Teraz katalog czyta się poprawnie z zewnątrz, kosztem tego, że zmiana nazwy jest realną operacją — a archiwa wyeksportowane przed nią zachowują stare nazwy.

Q. Czy mogę przebudować odnośniki w całym vaultie?

Tak — jest akcja *przebuduj odnośniki* na poziomie vaulta, która przechodzi całość.

To właściwe narzędzie po zaimportowaniu większej partii materiału, bo notatki napisane, zanim powstały ich sąsiedzi, nie miały jak do nich linkować. To także sposób na wyjście z okresu intensywnych zmian tytułów.

Obowiązuje to samo zabezpieczenie: model proponuje frazę z istniejącego tekstu i plik docelowy, a podstawienia dokonuje aplikacja. Model nie dostaje przy okazji prawa przepisania Twojej prozy.

Q. Czy mogę pisać odnośniki sam?

Tak, i zachowują się identycznie, bo w wygenerowanym odnośniku nie ma nic szczególnego. To link Markdown w pliku Markdown.

Wskaż prawdziwą nazwę pliku w `notes/` i wszystko działa: lista wychodząca je podchwyci, a notatka po drugiej stronie pokaże Twoją w swoich backlinkach po przeindeksowaniu.

To praktyczna korzyść z nietrzymania odnośników w bazie. Format jest taki, który wszyscy już znają, a Twoje edycje i edycje aplikacji są tym samym rodzajem rzeczy.

Q. Czy odnośniki przeżywają eksport?

Po to właśnie są względne nazwy plików. Vault eksportuje się jako ZIP z `base.md` i katalogiem `notes/`, a odnośniki między plikami są ścieżkami względnymi.

Rozpakuj i otwórz w Obsidianie, VS Code albo repozytorium GitHub — graf działa. Bez wtyczki, bez importu, bez konwersji.

To różnica między posiadaniem swoich notatek a posiadaniem ich kopii. Wyeksportowany vault nie jest zrzutem struktury, która mieszka gdzie indziej — on *jest* tą strukturą.