

Ostatnio namnożyło się postów osób z problemami z budową bazy danych, relacjami, tabelami i.t.d. Więc może komuś pomoże moje przemyślenie, jak warto do tego podejść.

Uprzedzając - będzie to inne niż to, co zaprezentował na filmach

Adam

- bardziej czasochłonne, ale prostsze dla osób, które nie tworzyły nigdy bazy danych.

Przechodząc już do tematu, na sam początek warto zastanowić się, co w ogóle chcemy osiągnąć, jak ma działać nasza aplikacja - ogólnikowo opisać sobie jej model biznesowy i zaznaczyć wszystkie dane, które można na pierwszy rzut oka przechowywać w bazie danych. Przykładowo:

*Sklep internetowy - supermarket*

*Sklep będzie oferował produkty spożywcze i artykuły kosmetyczne. Na stronie internetowej będzie można je przeglądać. Produkty można dodać do koszyka. Aby je zamówić, klient będzie musiał posiadać konto. Po opłaceniu, zamówienie musi zostać potwierdzone przez pracownika, aby mogło zostać zrealizowane. Potwierdzone zamówienie zostanie wysłane do klienta na adres podany w zamówieniu.*

Po wypisaniu zaznaczonych rzeczy, trzeba zastanowić się, jakie informacje będą potrzebne przy konkretnym zagadnieniu:

- *produkty spożywcze - cena, opis, informacje kaloryczne, skład, dostępność*
- *artykuły kosmetyczne - cena, skład, przeznaczenie, opis, dostępność*
- *koszyk - produkty w środku,*
- *klient - złożone zamówienia, aktualne produkty w koszyku, adres(y?), dane kontaktowe, dane osobowe*
- *konto - dane do logowania*
- *zamówienie - co zamówione, kiedy złożone, status, kiedy dostarczone, cena zamówienia, adres, klient*
- *pracownik - dane osobowe, adres, dane kontaktowe, obsługiwane zamówienia*

Skoro to już mamy, to czas uszeregować nasze dane - zastanowić się nad nimi, co tak na prawdę z tego będzie nam potrzebne, co się powtarza, co ma ze sobą związek i postarać się to w miarę zebrać do kupy:

*Na sam początek produkty spożywcze i artykuły kosmetyczne. Czy na prawdę trzeba to tak rozbijać? Oba, to rodzaje oferowanych produktów. Oba mają cenę, opis i dostępność. Za to pozostałe informacje przy nich, to też forma opisu tych produktów. Więc może da rady je połączyć:*

***Produkt - cena, opis, dostępność, rodzaj produktu***

*Klient i pracownik - razem mają dane osobowe, adres, dane kontaktowe. Niestety sporo je też dzieli - klient składa zamówienia, a pracownik je obsługuje. Wobec tego nie damy rady ich połączyć, ale nie musimy też powielać danych, które mogą się powtarzać - przecież pracownik też może być w wolnej chwili klientem, albo pod tym samym adresem może mieszkać kilka osób. Poza tym, nie można zapomnieć, że tak klient, jak pracownik muszą mieć konto.*

**Adres - miasto, kod pocztowy, ulica, numer domu, numer mieszkania**

**Dane kontaktowe - email, numer telefonu**

**Dane osobowe - imię, nazwisko**

**Konto - login, hasło**

**Klient - złożone zamówienia\*, koszyk\*, adres(y)\* (w ten sposób oznaczam powiązanie z inną tabelą), dane kontaktowe\*, dane osobowe\***

**Pracownik - obsługane zamówienia\*, adres\*, , dane kontaktowe\*, dane osobowe\***

*Zostają 2 tabele - zamówienie i koszyk. Ale zastanówmy się, że je też możemy połączyć. W końcu produkty w koszyku, to zamówienie, tylko bez reszty danych i jeszcze nie złożone. Dlatego nie warto rozdzielać się na drobne.*

*Tylko Uwaga - koszyk miał być związany z klientem, a do tego skoro wiemy pod jakie adresy były realizowane zamówienia dla danego klienta, nie musimy ich powielać - więc jego dane, też będziemy musieli zmienić!*

**Zamówienie - produkty\*, data złożenia zamówienia, data dostarczenia, status, cena zamówienia, adres\***

**Klient - złożone zamówienia\*, dane kontaktowe\*, dane osobowe\***

*Wróćmy jeszcze na chwilę do zamówienia i produktów, bo są z nimi 2 problemy:*

*1.1 produkt może być w wielu zamówieniach, a dane zamówienie mieć wiele produktów. Normalnie, połączenie wiele do wielu nie byłoby problemem (bo robi je za nas EntityFramework), ale...*

*2. Produkty mogą zmieniać cenę, a warto by było, żeby w zamówieniu poza znajomością jego całkowitej ceny, znać też cenę każdego produktu z zamówienia.*

*Z tych powodów, ręcznie przygotujemy tabelę łączącą:*

**Zamówienie\_Produkt - produkt\*, zamówienie\*, cena przy zamówieniu**

*A skoro już znamy ceny poszczególnych produktów w zamówieniu na czas jego składania, nie musimy przechowywać dodatkowo ceny całego zamówienia - w końcu każdy (a szczególnie komputery) umie dodawać, więc nie trzeba się powtarzać:*

## ***Zamówienie - produkty\*, data złożenia zamówienia, data dostarczenia, status, adres\****

Zdaję sobie sprawę, że takie ustalanie danych jest trudne i może wydawać się dziwne, ale nie można zapominać o tym, że projektujemy bazę danych - ona ma przechowywać powiązane ze sobą informacje, a nie koniecznie to, jak później chcemy je wyświetlać. Jeżeli będą istniały odpowiednie relacje, będzie można stworzyć dowolne zestawienia danych, dla których często nie ma sensu robić tabel - np. liczba produktów, jakie zostały wysłane na dany adres (na końcu będzie opis).

Wracając do pracy - skoro już poruszony został temat powiązań między tabelami, to teraz się tym zajmujemy. Warto zrobić to graficznie - czy to na kartce, czy w paintcie, czy wykorzystując program taki jak draw.io

Na początek po prostu trzeba narysować tabele z ich zawartością i połączyć kreskami tak jak mniej więcej ustaliło się wcześniej, co które dane będą z czym powiązane. Następnie czas na ustalenie relacji.

*Zacznijmy od tego co już ustaliliśmy - 1 produkt może być w wielu zamówieniach, a dane zamówienie mieć wiele produktów. Ale co z zamówienie\_produkt? Na początku ustalmy w ten sposób relacje, jak opisane były one wyżej:*

*1 produkt może być w wielu Zamówienie\_produktach, 1 zamówienie może zawierać wiele Zamówienie\_produktów. W drugą stronę połączenie będzie zawsze do 1: 1 zamówienie\_produkt zawiera tylko 1 produkt, 1 zamówienie produkt, zawiera tylko 1 zamówienie.*

*Ostatecznie dostaniemy:*

- *1 zamówienie\_produkt może zawierać tylko 1 produkt, 1 produkt może być przypisany do wielu zamówień\_produkt (1 do wielu)*
- *1 zamówienie\_produkt może zawierać tylko 1 zamówienie, 1 zamówienie może być przypisane do wielu zamówienie\_produktów (1 do wielu)*

*Skoro już jesteśmy przy zamówieniach:*

- *Zamówienie może być wysłane tylko na 1 adres, ale na 1 adres można wysłać wiele zamówień (1 do wielu)*
- *Tylko 1 pracownik może zarządzać danym zamówieniem, ale pracownik może zarządzać wieloma zamówieniami (1 do wielu)*
- *Tylko 1 klient może zamówić dane zamówienie, ale klient może zamówić wiele zamówień (1 do wielu)*

*Klient:*

- *dane klienta mogą być przypisane tylko do 1 konta, konto może pozwalać na zalogowanie tylko 1 klienta (1 do 1)*

● klient może posiadać wiele danych kontaktowych, dane kontaktowe mogą być powiązane tylko z 1 klientem (1 do wielu)

● klient może mieć tylko 1 zestaw danych osobowych, dane osobowe mogą być powiązane tylko z 1 klientem (1 do 1)

*Pracownik*

● dane pracownika mogą być przypisane tylko do 1 konta, konto może pozwalać na zalogowanie tylko 1 pracownika (1 do 1)

● pracownik może posiadać wiele danych kontaktowych, dane kontaktowe mogą być powiązane tylko z 1 pracownikiem (1 do wielu)

● pracownik może mieć zapisany tylko 1 adres domowy, adres może być przypisany do wielu pracowników, np. gdy kilka osób mieszka razem (1 do wielu)

● pracownik może mieć tylko 1 zestaw danych osobowych, dane osobowe mogą być powiązane tylko z 1 pracownikiem (1 do 1)

*Pozostałe relacje, tj. od danych kontaktowych, kont, danych osobowych i adresów w sumie już opisaliśmy wyżej, a nie chce mi się ich powtarzać - jak ktoś chce to proszę, ale to już zbędne. Rysowanie 1 i \* na diagramie też już pominę, bo to nie powinien być problem 😊*

Kwestia relacji potrafi sprawić sporo problemów, dlatego polecam pomyśleć o niej globalnie - tak jak opisałem to wyżej. Starać się nie patrzeć z punktu widzenia, np. zamówienia, że skoro klient może realizować na raz tylko 1 zamówienie, to będzie 1:1 - bo jeszcze są zamówienia z przeszłości, a to które teraz jest realizowane, tak samo w końcu zostanie zarchiwizowane, a przypisane będzie w ten sam sposób.

A teraz obiecane powiązanie liczby produktów do adresu i dlaczego nie ma sensu oddzielna tabela, jeżeli taka informacja będzie potrzebna:

*Adresy są powiązane wprost z zamówieniami - wiemy co, dokąd ma dotrzeć.*

*Zamówienia są powiązane tabelą łączącą z produktami - dzięki temu wiemy, jakie produkty zawierało każde z zamówień.*

*Dlatego możemy stworzyć kwerendę, która:*

*Zsumuje produkty, grupując je po nazwach, które były częścią zamówień realizowanych pod dany adres.*

Dobra, aż za bardzo się rozpisałem, ale może komuś to coś pomoże 😊