

Normalizacja bazy danych

Tabele w bazie danych powinny być logicznie uporządkowane i funkcjonalne. W każdym polu i rekordzie powinna być jak najmniejsza porcja informacji tak, aby dane nie były powielane i można było je łatwo znaleźć. **Normalizacja** to proces sprowadzania bazy danych do odpowiedniej postaci. Polega on przede wszystkim na dzieleniu tabeli na kilka połączonych kluczem tabel. Głównym powodem, dla którego normalizuje się bazę danych jest uniknięcie anomalii, które mogą wystąpić przy nieprawidłowo skonstruowanej strukturze bazy. Możliwe anomalie są przedstawione w tabeli poniżej.

Załóżmy, że mamy następującą strukturę bazy książek w bibliotece:

Tytuł książki	Autor	Wypożyczający	Adres wypożyczającego	Data wypożyczenia
---------------	-------	---------------	-----------------------	-------------------

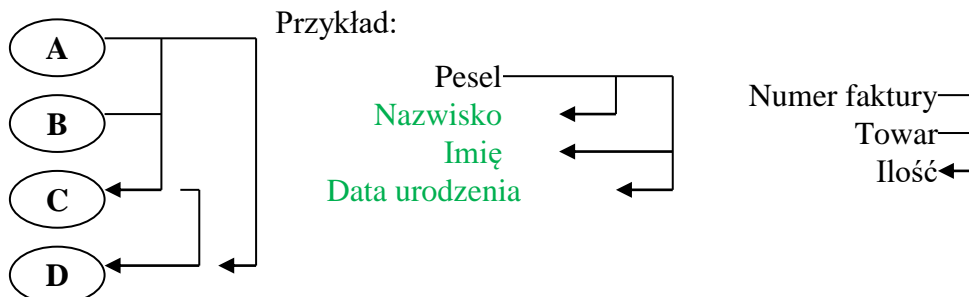
W tej bazie wystąpią następujące anomalie:

Anomalia	Opis
przy aktualizacji	Jeżeli wypożyczający zmienił adres, trzeba przeszukać całą bazę i we wszystkich komórkach, w których występuje, zmienić ten adres
przy usuwaniu	Jeżeli wypożyczający zwróci ostatnią książkę, zostanie utracona informacja na jego temat (adres i inne dane osobowe)
przy wstawianiu	Nowa osoba nie może zapisać się do biblioteki, jeżeli nie wypożyczy książki (a nie musi od razu wypożyczać)
redundacja	Redundacja, czyli powtarzanie tej samej informacji w kilku miejscach w bazie, powoduje niepotrzebne zajmowanie pamięci (wypożyczenie dwóch książek powoduje, że niepotrzebnie adres jest powtarzany dwa razy)

Aby osiągnąć właściwe uporządkowanie danych i zminimalizować ich powtarzalność należy każdą tabelę poddać co najmniej trzem etapom normalizacji (w sumie jest ich sześć). Każda z następných postaci normalnych jest bardziej wymagająca od poprzedniej i jest zgodna z poprzednimi postaciami normalizacji.

Zależność funkcjonalna

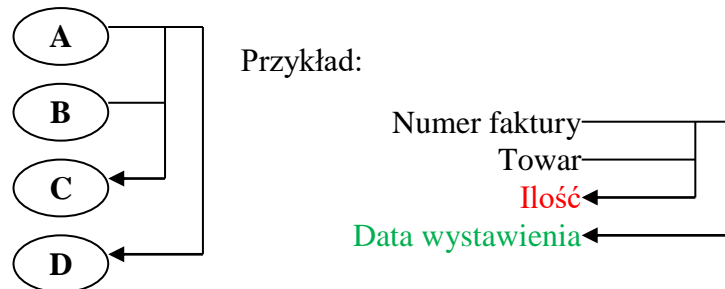
Atrybut B relacji r jest funkcjonalnie zależny od atrybutu A tej relacji (tabeli), jeżeli zawsze każdej wartości a atrybutu A odpowiada zawsze nie więcej niż jedna wartość b atrybutu B; B jest funkcjonalnie zależne od A, A identyfikuje B.



Pełna zależność funkcjonalna

Atrybut B relacji r jest w pełni funkcjonalnie zależny od zbioru atrybutów X, jeżeli jest funkcjonalnie zależny od niego, ale nie jest funkcjonalnie zależny od żadnego podzbioru zbioru X.

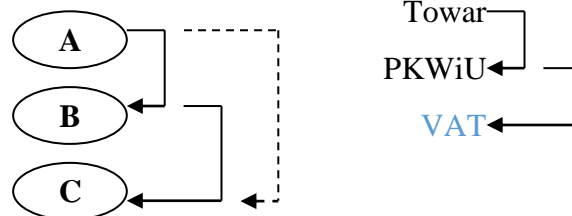
W przykładzie kluczem głównym jest AB. Zależność $A \rightarrow D$ powoduje, że nie ma pełnej zależności funkcjonalnej. Kluczem jest Numer faktury i towar. Identyfikują ilość towaru na fakturze. Jednocześnie Numer faktury określa datę jej wystawienia.



Przechodnia zależność funkcjonalna

Niech X, Y, Z będą trzema rozłącznymi podzbiórami atrybutów. Podzbiór atrybutów Z jest przechodnio funkcjonalnie zależny od podzbioru atrybutów X, jeśli podzbiór atrybutów Z jest funkcjonalnie zależny od podzbioru atrybutów Y, podzbiór atrybutów Y jest funkcjonalnie zależny od podzbioru atrybutów X, natomiast podzbiór atrybutów X nie jest funkcjonalnie zależny od Y lub podzbiór atrybutów Y nie jest funkcjonalnie zależny od Z.

Przykład zależności częściowej (klucz A)



Usunięcie zależności częściowej przez podzielenie tabeli (III postać normalna)

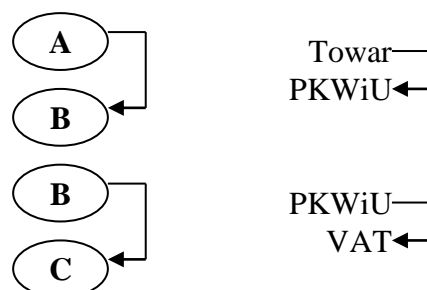


Tabela nieznormalizowana

W tabeli nieznormalizowanej informacje są umieszczone w jednym polu tabeli, są one niejednoznaczne i mają różne wartości. Uniemożliwia to wybranie określonych danych, np. numeru telefonu, nazwy towaru. Tak sporządzona tabela jest nieprzydatna i nie można na jej podstawie tworzyć zapytań oraz raportów.

Pierwsza postać normalna

Schemat tabeli w pierwszej postaci normalnej jest bardzo prosty. Wartości atrybutów (pól) muszą być niepodzielne. Każde pole zawiera tylko jedną wartość, a nie listę wartości lub powtarzającą się grupę danych. Do otrzymanej relacji (tabeli) określa się potencjalne pola – klucze.

Druga postać normalna

Tabela jest w drugiej postaci normalnej, gdy jest w pierwszej postaci normalnej oraz każdy atrybut tej tabeli nie należący do klucza potencjalnego jest w pełni funkcjonalnie zależny od wszystkich kluczy potencjalnych. Usuwa się niepełne zależności funkcjonalne. Oznacza to, że wszystkie dane w tej tabeli zależą tylko od klucza w tej tabeli i nie powtarzają się w innych tabelach (np. nazwisko klienta jest wpisane tylko w jednej tabeli zawierającej dane osobowe wszystkich klientów).

Trzecia postać normalna

Z tabelą w trzeciej postaci normalnej mamy do czynienia, gdy jest w drugiej postaci normalnej i każdy jej atrybut nie wchodzący w skład żadnego klucza potencjalnego nie jest przechodnio funkcjonalnie zależny od żadnego klucza potencjalnego tej tabeli. W bazie nie ma zależności przechodnich.

Przykład normalizacji bazy danych:

a) Tabela nieznormalizowana

Baza nie znajduje się w pierwszej postaci normalnej. Pola, w których znajduje się kilka wartości „atomowych”, należy przesunąć do innej tabeli.

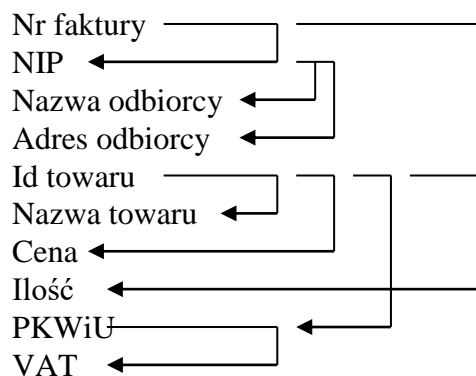
NrF	NIP	Nazwa odbiorcy	Adres odbiorcy	Id Towaru	Nazwa towaru	Ilość	Cena	PKWiU	VAT
1	101203203	WLW	Ul. Róż 2	22	Młotek	3	2,5	1221	7
				21	Wapno	4	3,6	1222	0
				24	Toporek	3	2,5	1221	7
2	123456543	SKK	Ul. Kot 3	22	Młotek	4	2,5	1221	7
				33	Siekiera	5	5,0	2212	22
3	222221445	WSK	Ul. Lód 5	33	Siekiera	2	5,0	2212	22

b) I postać normalna

W każdej komórce tabeli znajduje się jedna wartość atrybutu. W ten sposób otrzymaliśmy pierwszą postać normalną.

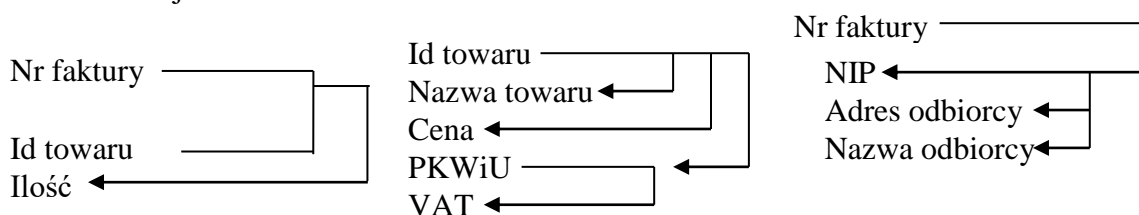
NrF	NIP	Nazwa odbiorcy	Adres odbiorcy	Id Towaru	Nazwa towaru	Ilość	Cena	PKWiU	VAT
1	101203203	WLW	Ul. Róż 2	22	Młotek	3	2,5	1221	7
1	101203203	WLW	Ul. Róż 2	21	Wapno	4	3,6	1222	0
1	101203203	WLW	Ul. Róż 2	24	Toporek	3	2,5	1221	7
2	123456543	SKK	Ul. Kot 3	22	Młotek	4	2,5	1221	7
2	123456543	SKK	Ul. Kot 3	33	Siekiera	5	5,0	2212	22
3	222221445	WSK	Ul. Lód 5	33	Siekiera	2	5,0	2212	22

Kandydat na klucz podstawowy: Nr faktury + Id towaru



c) II postać normalna

Baza znajduje się w drugiej postaci normalnej, jeżeli nie ma zależności częściowych. Oznacza to, że jeśli pole zależy od części klucza wielowartościowego, należy przerzucić ją do oddzielnej tabeli.



III postać normalna

Baza znajduje się w trzeciej postaci normalnej, jeżeli nie ma zależności przechodnich. Oznacza to, że jeżeli pole nie jest wprost zależne od klucza, to należy przenieść je do nowej tabeli.

