**Pierwsza postać normalna 1NF**

Wśród najważniejszych czynników określających zasadę działania tej reguły, należy wymienić:

* **Usunięcie powtarzających się grup w ramach pojedynczych tabel.**
* **Tworzenie oddzielnej tabeli dla pojedynczego zestawu danych relacyjnych.**
* **Określenie każdego z zestawów dzięki kluczowi podstawowemu.**

Reguła ta dotyczy atomowości danych, a więc sytuacji, gdy tabela (encja) przechowuje dane atomowo. Każda komórka to konkretna informacja, co zapewnia najbardziej efektywną pracę z zapytaniami. Dzięki tej podstawowej formie pojawia się też klucz główny. To on odpowiedzialny jest za identyfikację każdego wiersza. Nie występują tutaj kolekcje, a struktura bazodanowa opiera się na pojedynczych informacjach w pojedynczych wierszach.

**Druga postać normalna 2NF**

Wprowadzenie drugiej formy normalnej ma na celu przede wszystkim:

* **Budowę osobnych tabel dla zestawów wartości powiązanych z wieloma zapisami.**
* **Ustalenie relacji pomiędzy tabelami przy użyciu klucza obcego.**

Zgodnie z teorią tej reguły, poszczególne zapisy powinny być zależne wyłącznie od klucza głównego konkretnej tabeli. Przykładowo, warto pochylić się nad poruszonym wcześniej adresem klienta. Informacji tej potrzebować mogą tabele z danymi klientów, ale także dotyczące zamówień, dostaw, fakturowania, należności czy kolekcji. Nie musisz przetrzymywać adresu klienta w każdej tabeli z osobna, jeśli znormalizujesz dane do jednej tabeli z adresami.

**Trzecia postać normalna 3NF**

Głównym celem implementacji tej formy w swojej relacyjnej bazie danych jest:

* **Usunięcie pół, które nie zależą od klucza.**

Jeżeli wartości zapisu nie stanowią części klucza rekordu, wówczas nie powinny się tam znajdować. Ogólnie rzecz biorąc, zawsze, gdy zawartość grup lub pól dotyczy więcej niż jednego rekordu w tabeli, warto przenieść je do osobnej encji.

Przykładowo, możesz potrzebować nazwy i adresu uniwersytetu osoby ubiegającej się o pracę w tabeli przeznaczonej kandydatom. Z drugiej strony, lista wszystkich uczelni wyższych może być potrzebna również w innych celach. Jeżeli informacje o uniwersytetach znajdują się w tabeli Kandydaci, wówczas nie da się wyświetlić szkół bez wyświetlania kandydatów. W takim przypadku przydatne jest stworzenie osobnej tabeli Uniwersytety oraz połączenie jej z Kandydatami za pomocą klucza kodu uczelni.

Warto dodać, że zastosowanie się do reguły 3NF może okazać się przydatne wyłącznie w sytuacji częstej zmiany danych. Jeżeli w bazie pozostaną pola zależne, należy tak zaprojektować aplikację, by zmuszała użytkownika do weryfikacji wszystkich powiązanych pól w razie wprowadzenia jakichkolwiek modyfikacji.

