NORMALIZACJA

Normalizacja bazy

- **Normalizacja bazy danych** jest to proces mający na celu eliminację powtarzających się danych w relacyjnej bazie danych. Główna idea polega na trzymaniu danych w jednym miejscu, a w razie potrzeby linkowania do danych. Taki sposób tworzenia bazy danych zwiększa bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności (w szczególności problemów anomalii).

Postacie normalne

- Istnieją sposoby ustalenia czy dany schemat bazy danych jest „znormalizowany”, a jeżeli jest to jak bardzo. Jednym ze sposobów jest przyrównanie danej bazy do schematów zwanych **postaciami normalnymi** (ang. **normal forms** lub **NF**). Normalizacja bazy danych do konkretnej postaci może wymagać rozbicia dużych tabel na mniejsze i przy każdym wykonywaniu zapytania do bazy danych ponownego ich łączenia. Zmniejsza to wydajność, więc w niektórych przypadkach świadoma denormalizacja (stan bez normalizacji) jest lepsza – zwłaszcza w systemach niekorzystających z **modelu relacyjnego** (np. OLAP).
- **Celem normalizacji baz danych jest unikanie anomalii.**

Anomalie baz danych

- **Redundancja** — ta sama informacja jest niepotrzebnie przechowywana w kilku krotkach.
- **Anomalia modyfikacji** — informacja zostanie zmodyfikowana w pewnych krotkach, a w innych nie. Która informacja jest wówczas prawdziwa?
- **Anomalia usuwania** — usuwanie części informacji powoduje utratę innej informacji, której nie chcielibyśmy stracić.
- **Anomalia dołączania** — wprowadzenie pewnej informacji jest możliwe tylko wtedy, gdy jednocześnie wprowadzamy jakąś inną informację, która może być obecnie niedostępna.

Postacie normalne

- Edgar Frank Codd (twórca normalizacji) początkowo wymyślił 3 postacie normalne: 1NF, 2NF i 3NF. Obecnie istnieją jeszcze inne postacie, ale 3NF jest powszechnie uznawana za wystarczającą do większości projektów. Większość tabel spełniając postać 3NF, spełnia także BCNF (ang. Boyce-Codd normal form). 4NF i 5NF są następnymi rozszerzeniami, a 6NF jest używana do baz uwzględniających w modelu relacyjnym wymiar czasowy.

Pierwsza postać normalna 1NF

Mówimy, że tabela (encja) jest w **pierwszej postaci normalnej, kiedy wiersz przechowuje informacje o pojedynczym obiekcie, nie zawiera kolekcji, posiada klucz główny (kolumnę lub grupę kolumn jednoznacznie identyfikujących go w zbiorze) a dane są atomowe**.

Pierwsza postać normalna

- Wyeliminuj powtarzające się grupy w poszczególnych tabelach.
- Utwórz osobną tabelę dla każdego zestawu powiązanych danych.
- Zidentyfikuj każdy zestaw powiązanych danych za pomocą klucza podstawowego.

Druga postać normalna 2NF

2 NF - tabela powinna przechowywać dane dotyczące tylko konkretnej klasy obiektów.

Druga postać normalna

- Utwórz osobne tabele dla zestawów wartości dotyczących wielu rekordów.
- Powiąż te tabele za pomocą klucza obcego.

Trzecia postać normalna 3NF

Trzecia postać normalna głosi, że **kolumna informacyjna nie należąca do klucza nie zależy też od innej kolumny informacyjnej**, nie należącej do klucza. Czyli każdy niekluczowy argument jest bezpośrednio zależny tylko od klucza głównego a nie od innej kolumny.

Trzecia postać normalna

- Wyeliminuj pola, które nie zależą od klucza.

Normalizowanie przykładowej tabeli

- Tabela nieznormalizowana:

Nr studenta	Opiekun	Pokój opiekuna	Zajęcia 1	Zajęcia 2	Zajęcia 3
1022	Czarnecki	412	101-07	143-01	159-02
4123	Borkowski	216	201-01	211-02	214-01

Pierwsza postać normalna: brak powtarzających się grup

- Tabele powinny mieć tylko dwa wymiary. Ponieważ jeden student może mieć kilka rodzajów zajęć, zajęcia powinny być wymienione w osobnej tabeli. Pola Zajęcia 1, Zajęcia 2 i Zajęcia 3 w powyższych rekordach sygnalizują problemy z projektem.

Nr studenta	Opiekun	Pokój opiekuna	Nr zajęć
1022	Czarnecki	412	101-07
1022	Czarnecki	412	143-01
1022	Czarnecki	412	159-02
4123	Borkowski	216	201-01
4123	Borkowski	216	211-02
4123	Borkowski	216	214-01

Druga postać normalna: wyeliminowanie danych nadmiarowych

- W powyższej tabeli dla każdej wartości Nr studenta występuje wiele wartości Nr zajęć. Wartości Nr zajęć nie są funkcjonalnie zależne od klucza podstawowego Nr studenta, więc ta relacja nie jest w drugiej formie normalnej.
- W poniższych dwóch tabelach pokazano drugą formę normalną:

Studenci:

Nr studenta	Opiekun	Pokój opiekuna
1022	Czarnecki	412
4123	Borkowski	216

Druga postać normalna: wyeliminowanie danych nadmiarowych

Rejestracja:

Nr studenta	Nr zajęć
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Trzecia postać normalna: wyeliminowanie danych, które nie zależą od klucza

- W ostatniej tabeli wartości Pokój opiekuna są funkcjonalnie zależne od atrybutu Opiekun. Rozwiązaniem jest przeniesienie tego atrybutu z tabeli Studenci do tabeli Wykładowcy, jak pokazano poniżej:

Studenci:

Nr studenta	Opiekun
1022	Czarnecki
4123	Borkowski

Trzecia postać normalna: wyeliminowanie danych, które nie zależą od klucza

Wykładowcy:

Nazwisko	Pokój	Wydział
Czarnecki	412	42
Borkowski	216	42

Normalizacja a wydajność

Normalizacja baz danych dostarcza **mechanizmu** pozwalającego unikać anomalii. Ma to jednak swoją cenę: Dostęp do danych w bazie znormalizowanej może być wolniejszy, gdyż RDBMS musi wykonywać złączenia.

Dlatego w wielkich bazach danych zoptymalizowanych na odczyt (na przykład w hurtowniach danych) często rezygnuje się z wyższych postaci normalnych, przechowując dane w tabelach 1PN. *To* także ma swoją cenę: Wprowadzając dane do takich tabel lub modyfikując istniejące dane należy dołożyć szczególnej staranności, aby nie dopuścić do anomalii usuwania lub dołączania, a szczególnie do anomalii modyfikacji (redundancja jest w tego typu bazy niejako wbudowana).

Relational Database Management System, RDBMS

Źródło:

- https://support.microsoft.com/pl-pl/kb/283878