Popis normalizace

Normalizace je proces uspořádání dat v databázi. Patří sem vytváření tabulek a navazování relací mezi těmito tabulkami podle pravidel určených k ochraně dat a k tomu, aby byla databáze flexibilnější tím, že eliminuje redundanci a nekonzistentní závislost.

Redundantní data ztrácí místo na disku a problémy s údržbou. Pokud je nutné změnit data, která existují na více než jednom místě, musí být data ve všech umístěních změněna přesně stejným způsobem. Příklad: Změna adresy zákazníka je mnohem jednodušší implementovat, pokud jsou tato data uložená jenom v tabulce Zákazníci a nikde jinde v databázi.

Co je nekonzistentní závislost? I když je intuitivní, když se uživatel podívá do tabulky Zákazníci na adresu konkrétního zákazníka, nemusí mít smysl hledat tam plat zaměstnance, který volá zákazníkovi. Mzda zaměstnance souvisí se zaměstnancem nebo je na zaměstnanci závislá, a proto by měla být přesunuta do tabulky Zaměstnanci. Nekonzistentní závislosti mohou ztěžovat přístup k datům, protože cesta k vyhledání dat může být chybějící nebo nefunkční.

Existuje několik pravidel pro normalizaci databáze. Každé pravidlo se nazývá "normální forma". Pokud je dodrženo první pravidlo, je databáze v "první normální formě". Pokud jsou dodržena první tři pravidla, považuje se databáze za "třetí normální formu". I když jsou možné další úrovně normalizace, třetí normální forma se považuje za nejvyšší úroveň potřebnou pro většinu aplikací.

Stejně jako u mnoha formálních pravidel a specifikací neumožňují scénáře v reálném světě vždy dokonalé dodržování předpisů. Obecně platí, že normalizace vyžaduje další tabulky a někteří zákazníci to stěžují. Pokud se rozhodnete porušovat jedno z prvních tří pravidel normalizace, ujistěte se, že vaše aplikace předvídá případné problémy, jako jsou redundantní data a nekonzistentní závislosti.

Následující popisy zahrnují příklady.

První normální forma

- Odstraňte opakující se skupiny v jednotlivých tabulkách.
- Vytvořte samostatnou tabulku pro každou sadu souvisejících dat.
- Identifikujte každou sadu souvisejících dat pomocí primárního klíče.

K ukládání podobných dat nepoužívejte více polí v jedné tabulce. Pokud třeba chcete sledovat skladovou položku, která může pochovat ze dvou možných zdrojů, může záznam zásob obsahovat pole pro Kód dodavatele 1 a Kód dodavatele 2.

Co se stane, když přidáte třetího dodavatele? Přidání pole není odpověď; vyžaduje úpravy programů a tabulek a nepasuje plynule dynamickému počtu dodavatelů. Místo toho umístěte všechny informace o dodavateli do samostatné tabulky s názvem Dodavatelé a pak propojte zásoby s dodavateli pomocí číselného klíče položky nebo dodavatelů do inventáře pomocí kódu dodavatele.

Druhá normální forma

- Vytvořte samostatné tabulky pro sady hodnot, které platí pro více záznamů.
- Spojte tyto tabulky cizím klíčem.

Záznamy by neměly záviset na ničem jiném než na primárním klíči tabulky (v případě potřeby na složeném klíči). Zvažte třeba adresu zákazníka v účetním systému. Adresu je potřeba v tabulce Zákazníci, ale také v tabulkách Objednávky, Expedice, Faktury, Pohledávky a Kolekce. Místo uložení adresy zákazníka jako samostatné položky v každé z těchto tabulek ji uložte na jednom místě, a to buď v tabulce Zákazníci, nebo v samostatné tabulce Adresy.

Třetí normální forma

- Odstraňte pole, která nezávisí na klíči.

Hodnoty v záznamu, který není součástí klíče tohoto záznamu, nepatří do tabulky. Obecně platí, že kdykoli se obsah skupiny polí může vztahovat na víc než jeden záznam v tabulce, zvažte umístění těchto polí do samostatné tabulky.

Například v tabulce Nábor zaměstnanců může být zahrnutý název a adresa školy kandidáta. Potřebujete ale úplný seznam vysokých škol pro skupinové korespondence. Pokud jsou informace pro vysokoškoláky uložené v tabulce Kandidáti, neexistuje žádný způsob, jak vypište univerzity bez současných kandidátů. Vytvořte samostatnou tabulku Univerzity a propojte ji s tabulkou Kandidáti pomocí kódu pro vysokoškoláky.

VÝJIMKA: Dodržování třetí normální formy, i když je teoreticky žádoucí, není vždy praktické. Pokud máte tabulku Zákazníci a chcete vyloučit všechny možné závislosti mezi poli, musíte vytvořit samostatné tabulky pro města, PSČ, obchodní zástupce, třídy zákazníků a jakýkoli jiný faktor, který může být duplikován ve více záznamech. Normalizace se teoreticky vyplatí vyčistit. Mnoho malých tabulek ale může snížit výkon nebo překročit kapacitu otevřených souborů a paměti.

Může být nasnadě použít třetí normální formulář jenom u dat, která se často mění. Pokud některá závislá pole zůstanou, navrhněte aplikaci tak, aby vyžadovala, aby uživatel ověřil všechna související pole při změně libovolného pole.

Další normalizační formy

Čtvrtá normální forma, nazývaná také Boyce Codd Normal Form (BCNF), a pátá normální forma existuje, ale v praktickém designu se málokdy zváží. Ignorování těchto pravidel může vést k menšímu než dokonalému návrhu databáze, ale nemělo by to mít vliv na funkčnost.

Normalizace příkladové tabulky

Tento postup ukazuje proces normalizace fiktivní tabulky studentů.

1. Nenormalizovaná tabulka:

TABLE 1

Student #	Poradce	Adv-Room	Třída 1	Třída 2	Class3
1022	Jones	412	101-07	143-01	159-02
4123	Novák	216	101-07	143-01	179-04

2. První normální forma: Žádné opakující se skupiny

   Tabulky by měly mít jenom dva rozměry. Vzhledem k tomu, že jeden student má několik kurzů, měly by být uvedené v samostatné tabulce. Pole Class1, Class2 a Class3 ve výše uvedených záznamech jsou indikací problémů s návrhem.

   Tabulky často používají třetí rozměr, ale tabulky by neměly. Dalším způsobem, jak se na tento problém podívat, je vztah 1:N, neumístíte jednu stranu a stranu N do stejné tabulky. Místo toho vytvořte další tabulku v první normální podobě odstraněním opakující se skupiny (Třída#), jak je znázorněno níže:

TABLE 2

Student #	Poradce	Adv-Room	Třída #
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Novák	216	101-07
4123	Novák	216	143-01

TABLE 2

Student #	Poradce	Adv-Room	Třída #
4123	Novák	216	179-04

3. Druhá normální forma: Odstranění nadbytečných dat

   Všimněte si více hodnot Class# pro každou hodnotu Student# ve výše uvedené tabulce. Třída# není funkčně závislá na studentovi # (primární klíč), takže tato relace není v druhé normální formě.

   Následující tabulky znázorňují druhý normální formulář:

   Studenti:

TABLE 3

Student #	Poradce	Adv-Room
1022	Jones	412
4123	Novák	216

   Registrace:

TABLE 4

Student #	Třída #
1022	101-07
1022	143-01
1022	159-02
4123	101-07
4123	143-01
4123	179-04

4. Třetí normální forma: Odstranění dat, která nejsou závislá na klíči

   V posledním příkladu je Adv-Room (číslo kanceláře poradce) funkčně závislé na atributu Poradce. Řešením je přesunout tento atribut z tabulky Studenti do tabulky Fakulta, jak je vidět níže:

   Studenti:

TABLE 5

Student #	Poradce
1022	Jones
4123	Novák

Fakulta:

TABLE 6

Name (Název)	Místnost	Oddělení
Jones	412	42
Novák	216	42