ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1999
Seria: INFORMATYKA z. 37 Nr kol. 1421

Stanisław KOZIELSKI
Politechnika Śląska, Instytut Informatyki

BEZPIECZEŃSTWO I INTEGRALNOŚĆ BAZ DANYCH

Streszczenie. W pracy przedstawiono zagrożenia dla danych przechowywanych w bazach danych oraz przegląd mechanizmów zabezpieczających przed tymi zagrożeniami. Oddzielnie rozpatrzono mechanizmy zabezpieczające przed nieuprawnionym dostępem do baz danych oraz mechanizmy ochrony integralności danych.

SECURITY AND INTEGRITY OF DATABASES

Summary. The paper presents the review of mechanisms assuring security and integrity of databases. Database security refers to protection from unauthorized or malicious access. Database integrity refers to protection the database from accidental loss of consistency.

1. Wstęp

Bazy danych są składnikami większości systemów informatycznych wspomagających zarządzanie. Dane gromadzone w bazach danych są więc podstawą funkcjonowania wielu przedsiębiorstw i instytucji, a ich utrata na skutek zdarzeń losowych może mieć katastrofalne skutki dla tych jednostek. Dane takie mogą przy tym być również przedmiotem zainteresowania konkurencyjnych przedsiębiorstw bądź włamywaczy komputerowych. Stąd też ogromna waga ochrony baz danych. Celem pracy jest dokonanie przeglądu i w pewnym zakresie oceny najważniejszych mechanizmów zapewniających bezpieczeństwo i ochronę integralności baz danych.

Rozważając bezpieczeństwo baz danych, na wstępie należy uwzględnić miejsce baz danych w ogólnej strukturze zabezpieczeń systemów komputerowych. Powinny one obejmować kilka poziomów [3]:

- Fizyczne zabezpieczenia dostępu do systemu komputerowego (odpowiednie zabezpieczenie pomieszczeń z systemem komputerowym).
- Personalna odpowiedzialność za dostęp do systemu komputerowego (uczciwość użytkowników dysponujących odpowiednimi uprawnieniami).
- Mechanizmy zabezpieczeń systemu operacyjnego (pierwszy poziom autoryzacji użytkowników).
- Mechanizmy kontroli dostępu w sieciach komputerowych (zabezpieczenia przed włamaniami do systemu sieciowego).
- Mechanizmy zabezpieczeń bazy danych (omówione w dalszych punktach pracy).

2. Ogólna klasyfikacja mechanizmów bezpieczeństwa i ochrony danych w systemach baz danych

Dane przechowywane w bazach danych mogą być narażone na wiele rodzajów zagrożeń [2,3,4]. W szczególności rozważymy:

1) umyślne próby nielegalnego odczytu lub modyfikacji danych przez nieuprawnionych użytkowników,
2) błędy logiczne w operacjach na danych,
3) skutki ewentualnych kolizji przy współbieżnym dostępie do danych wielu użytkowników,
4) awarie oprogramowania i awarie sprzętu komputerowego.

Zagrożenia wymienione w punkcie 1 związane są z działaniami użytkowników wykonywanymi umyślnie i w złej intencji. Stąd też mechanizmy przeciwdziałające takim zagrożeniom określane są jako mechanizmy zapewnienia bezpieczeństwa (ang. security) bazy danych. Ich zadaniem jest ochrona przed umyślnymi próbami niewłaściwego operowania na danych poprzez kontrolę praw do wykonywania operacji na danych.

Zagrożenia wymienione w punktach 2,3,4 mają charakter zdarzeń przypadkowych, które mogą doprowadzić do utraty integralności (spójności) bazy danych. Stąd też mechanizmy przeciwdziałające takim zagrożeniom są określane jako mechanizmy ochrony integralności (ang. integrity) bazy danych.

W kolejnych punktach pracy omówione zostaną oba rodzaje mechanizmów.

3. Mechanizmy zabezpieczania dostępu do danych

Stosowane w bazach danych zabezpieczenia dostępu do danych w głównej mierze dostępne są w serwerach baz danych. Pewien zakres kontroli jest też realizowany

w aplikacjach bazodanowych. Odrębne zagadnienie stanowią: szyfrowanie w bazach danych oraz środki kontroli dostępu w systemach statystycznej analizy danych.

3.1. Mechanizmy kontroli dostępu na poziomie serwera bazy danych

Mechanizmy kontroli dostępu obejmują identyfikację użytkowników, przyznawanie im określonych uprawnień do operacji na bazie danych, a następnie ograniczenie zakresu działań użytkownika do obszaru wyznaczonego przyznanymi uprawnieniami [2,3,4].

Pierwszym krokiem w takich działaniach jest utworzenie identyfikatora (konta) użytkownika przez administratora systemu. Operacji tej towarzyszy określenie przez użytkownika swego tajnego hasła. Istniejącym użytkownikom przyznawane są uprawnienia (zwane też przywilejami). Uprawnienia są zwykle dzielone na dwie kategorie:

1) uprawnienie o charakterze ogólnym odnoszące się do całej bazy danych,
2) uprawnienia szczegółowe, określające prawo do wykonywania określonej operacji na określonym obiekcie bazy danych.

ad 1) Tworzenie kont użytkowników i przyznawanie im uprawnień ogólnych może przebiegać nieco inaczej w różnych Systemach Zarządzania Bazami Danych (SZBD). Przykładowo w systemie Oracle [6,7] konto użytkownika jest tworzone poleceniem (w najprostszej postaci):

CREATE USER <użytkownik> INDENTIFIED BY <hasło>,

zaś uprawnienia ogólne (tu nazywane systemowymi) są przyznawane za pomocą polecenia (w najprostszej postaci):

GRANT <uprawnienie systemowe> TO <użytkownik>.

Przykłady uprawnień systemowych:

CREATE SESSION - prawo do nawiązywania połączeń z bazą danych,
CREATE TABLE - prawo do tworzenia tablic,
UPDATE ANY TABLE - prawo do aktualizacji tablic,
SELECT ANY TABLE - prawo do wyszukiwania danych z tablic.

W systemie Informix operacja tworzenia użytkowników jest powiązana z nadaniem im ogólnych uprawnień (nazywanych tu klasą użytkowników):

GRANT <klasa użytkownika> TO <użytkownik> [IDENTIFIED BY <hasło>]

gdzie klasy użytkowników określają między innymi:

CONNECT - prawo dostępu do tablic,
RESOURCE - prawo tworzenia tablic i udostępniania ich innym użytkownikom,
DBA - wszelkie prawa w bazie danych.

ad 2) Jak wspomnianio wyżej, uprawnienia szczegółowe dotyczą określonych operacji na określonych obiektach bazy danych, stąd uprawnienia te nazywane są obiektowymi. Są one przyznawane poleceniem w języku SQL o postaci:

GRANT <uprawnienie obiektowe> ON <obiekt> TO <użytkownik> [WITH GRANT OPTION]

W dominujących obecnie serwerach baz danych opartych na języku SQL do operacji będących przedmiotem uprawnień należy większość instrukcji tego języka, w tym na przykład:

SELECT
INSERT
DELETE
INDEX
ALTER
UPDATE [<lista kolumn>]

Uprawnienia mogą być również odbierane; umożliwia to instrukcja REVOKE.

Specyficznym aspektem rozważanych zagadnień jest umożliwienie propagacji uprawnień, tzn. przekazanie użytkownikowi prawa do przekazania nadanych mu uprawnień innym użytkownikom (opcja WITH GRANT OPTION). Wykorzystanie tej możliwości może w sposób mało kontrolowany rozszerzać grono użytkowników operujących na pewnych zasobach baz danych. Z drugiej strony natomiast odwołanie przyznanych pewnemu użytkownikowi uprawnień z taką opcją może wywołać kaskadowy efekt odbierania uprawnień innym użytkownikom, którzy dzięki tej opcji je otrzymali.

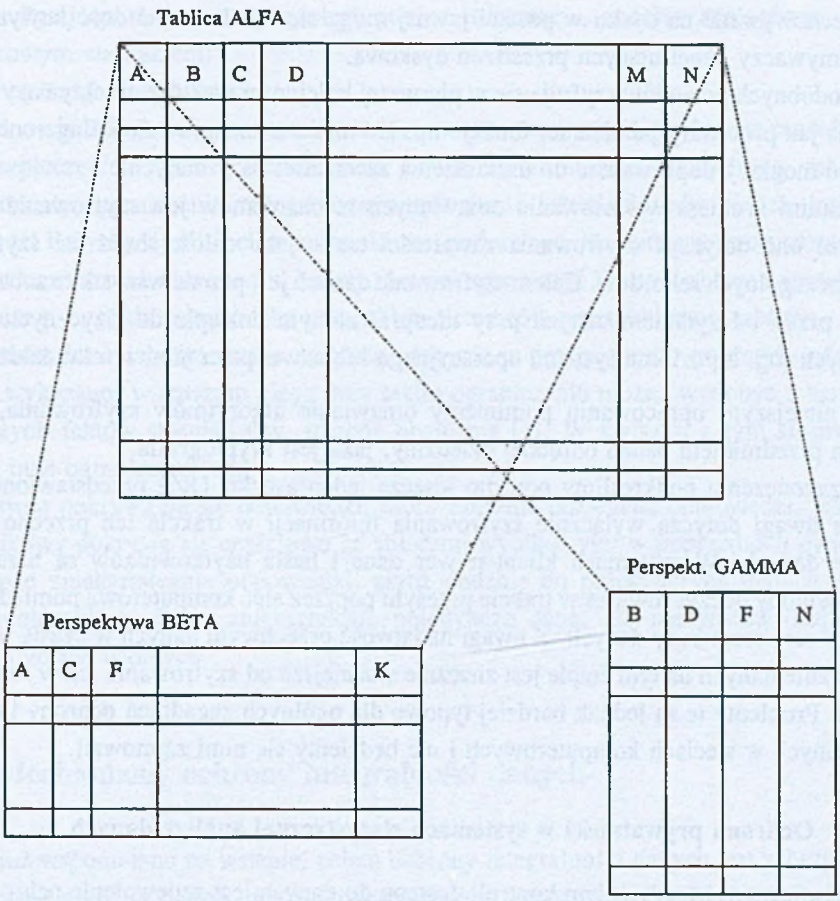
Do szczególnych obiektów, objętych nadawaniem uprawnień, należą perspektywy (VIEWS). Mechanizm perspektyw pozwala tworzyć wirtualne tablice udostępniające użytkownikowi fragment zasobów jednej lub wielu tablic rzeczywistych w formie prostej lub przetworzonej. Na rys. 1 przedstawiono przykład ilustrujący poglądowo możliwość utworzenia dwóch perspektyw w oparciu o jedną tablicę.

Przy wykorzystaniu tego mechanizmu mogą być przyznawane pewnym użytkownikom prawa dostępu nie do tablic rzeczywistych, a tylko do perspektyw. Pozwala to na wprowadzenie bardzo różnorodnych ograniczeń, w tym np. udostępnienie tylko wybranych kolumn bądź też wybranych wierszy.

3.2. Mechanizmy kontroli dostępu na poziomie aplikacji klienta

Przedstawione dotychczas mechanizmy nadawania uprawnień i kontroli za ich pomocą dostępu do bazy danych są realizowane w serwerach baz danych. Ochrona systemów bazodanowych może być również realizowana w tzw. aplikacjach klienta, czyli programach użytkowych wykonywanych w stacjach roboczych, korzystających z bazy danych za pośrednictwem serwera.

Mechanizmy ochrony są w takim przypadku tworzone przez projektanta aplikacji. Projektant dokonuje zawyczaj kategoryzacji użytkowników aplikacji i poszczególnym kategoriom zapewnia dostęp do określonych funkcji aplikacji. Dostępność funkcji jest realizowana przez odpowiednie ukształtowanie menu programu. Funkcje niedostępne dla



Rys. 1. Tworzenie perspektyw w języku SQL
Fig. 1. Views creation in SQL language

danego użytkownika są na ogół niewidoczne, czasami program nie pozwala ich wybrać ("wygaszone" pozycje w menu).

Użytkownicy aplikacji powinni być równocześnie zdefiniowani w systemie jako użytkownicy bazy danych, z której korzysta aplikacja. Bardzo ważne jest w takim przypadku precyzyjne skorelowanie uprawnień definiowanych na poziomie aplikaji z uprawnieniami definiowanymi na poziomie serwera bazy danych.

3.3. Szyfrowanie w bazach danych

Dodatkowym mechanizmem zabezpieczeń dla baz danych może być szyfrowanie [1]. Szyfrowanie może dotyczyć różnych informacji w ramach bazy danych.

- Najczęściej stosowanym zabiegiem jest szyfrowanie haseł użytkowników. Hasła przechowywane na dysku w postaci jawnej mogą się bowiem stać dość łatwym łupem włamywaczy penetrujących przestrzeń dyskową.
- Z podobnych powodów szyfruje się w pierwszej kolejności niektóre obiekty bazy danych, takie jak procedury pamiętane, funkcje itp. Ewentualna nieuprawniona ingerencja w ich treść mogłaby doprowadzić do uszkodzenia zawartości bazy danych.
- Ostatnim krokiem w stosowaniu omawianych mechanizmów jest szyfrowanie danych. Może ono dotyczyć szyfrowania zawartości tablic jako całości bądź też szyfrowanie poszczególnych rekordów. Celem szyfrowania danych jest przede wszystkim zabezpieczenie przed odczytaniem danych przy nieuprawnionym dostępie do fizycznych struktur danych (np. z poziomu systemu operacyjnego lub nawet poza nim, np. kradzież dysku).

W niniejszym opracowaniu pominiemy omawianie algorytmów szyfrowania, są one bowiem przedmiotem badań odrębnej dziedziny, jaką jest kryptografia.

W zakończeniu podkreślimy ponadto jeszcze jeden aspekt. Otóż przedstawione w tym punkcie uwagi dotyczą wyłącznie szyfrowania informacji w trakcie ich przechowywania w bazie danych. W systemach klient-serwer dane i hasła użytkowników są narażone na nieuprawniony odczyt również w trakcie przesyłu poprzez sieć komputerową pomiędzy stacją klienta a serwerem bazy danych. Z uwagi na łatwość przechwytu danych w czasie transmisji szyfrowanie danych na tym etapie jest znacznie ważniejsze od szyfrowania ich w samej bazie danych. Problemy te są jednak bardziej typowe dla ogólnych zagadnień ochrony informacji przesyłanych w sieciach komputerowych i nie będziemy się nimi zajmowali.

3.4. Ochrona prywatności w systemach statystycznej analizy danych

Specyficznym zagadnieniem kontroli dostępu do danych jest zapewnienie ochrony przed dostępem do opisu pojedynczych faktów w systemach stworzonych dla celów statystycznej analizy danych [1,2,3]. Systemy takie mogą działać na zwykłych bazach danych, np. bankowych czy medycznych, przy czym narzędzia analizy danych pozwalają formułować warunki dotyczące większości atrybutów (z wyjątkiem atrybutów identyfikujących osoby, firmy itp.), natomiast odpowiedzi dotyczą jedynie wartości zagregowanych (sum, wartości średnich, liczby wystąpień itp.). W systemach takich można więc zapytać np. o sumę kredytów zaciągniętych przez mężczyzn powyżej 50 roku życia lub liczbę chorych na serce w miejscowości Zawada. Łatwo można jednak pokazać, że znajomość pewnych danych o kliencie banku czy chorym pozwala w kilku pytaniach poznać dalsze dane, których system nie powinien ujawniać. Rozważmy np. zapytanie o liczbę chorych spełniających następujące warunki:

kobieta, wiek 40-43 lata, mężatka, jedno dziecko, wykształcenie wyższe, zawód nauczyciel, miejsce zamieszkania Knurów.

Załóżmy, że w odpowiedzi otrzymano liczbę 1, zaś pytający wie, że warunki te spełnia Pani X. Pytający może wtedy zdobyć poufne informacje, zadając analogiczne pytanie z dodatkowym warunkiem:

leczony na choroby nerwowe.

Jeżeli odpowiedź będzie 1, to sugestia zawarta w powyższym pytaniu będzie prawdziwa.

Zabezpieczeniem przed ujawnianiem poufnych informacji taką drogą może być wprowadzenie ograniczenia zabraniającego systemowi udzielania odpowiedzi, jeśli zbiór odpowiedzi liczy mniej niż k elementów. Biorąc pod uwagę, że pytający może sformułować pytanie z zaprzeczeniem warunku, należy również ograniczyć liczność zbioru odpowiedzi od góry przez n-k, gdzie n jest liczbą wszystkich elementów przeszukiwanej tablicy.

Przedstawione zabezpieczenie utrudnia zdobycie w prosty sposób poufnych informacji, ale, jak wykazano, w dalszym ciągu przy takim ograniczeniu można wydobyć z bazy opisy pojedńczych faktów stosując tzw. metodę obcinania [1]. W związku z tym stosowane są również inne ograniczenia, np.:

- Kontrola pokrywania się odpowiedzi, która zabrania udzielania odpowiedzi, jeśli zbiór wynikowy pokrywa się częściowo ze zbiorami wynikowymi w poprzednich pytaniach.
- Celowe zniekształcanie odpowiedzi, przez dodanie do pojedynczych danych wartości pseudolosowych, które zniekształcają pojedyncze dane, ale nie wprowadzają różnic statystycznie istotnych.

4. Mechanizmy ochrony integralności danych

Jak już wspomniano na wstępie, celem ochrony integralności danych jest zabezpieczenie bazy danych przed skutkami przypadkowych błędów logicznych, konfliktów we współbieżnym dostępie do bazy oraz awarii systemów komputerowych. Obszar zagadnień związanych z pojęciem integralności (innymi stosowanymi określeniami są zgodność lub spójność danych) dzielony jest zazwyczaj na integralność semantyczną i integralność transakcyjną [2,3]. Źródłem naruszeń integralności semantycznej są błędy logiczne w danych, natomiast zagrożeniem integralności transakcyjnej są: współbieżny dostęp do danych oraz awarie sprzętu i oprogramowania.

4.1. Mechanizmy zabezpieczeń integralności semantycznej

Integralność sematyczna jest rozumiana jako zgodność wartości danych w bazie ze znaczeniem atrybutów, których te dane dotyczą lub też z wartościami innych danych logicznie między sobą powiązanych.

Kontrola zgodności prowadzona jest poprzez zdefiniowanie, a następnie sprawdzanie ograniczeń, które muszą być spełniane przez wartości danych. Ograniczenia mają zazwyczaj

postać predykatów, których spełnienie świadczy o zgodnym stanie bazy danych. Sposób definiowania ograniczeń zależy od możliwości serwera bazy danych, języków użytych do: definiowania bazy danych, tworzenia procedur pamiętanych w bazie oraz tworzenia aplikacji użytkownika. Kontrola ograniczeń może się odbywać zarówno na poziomie serwera bazy danych, jak i aplikacji. Jedną z przykładowych technik definiowania ograniczeń na wartości danych jest wykorzystanie mechanizmów aktywnych baz danych, takich jak wyzwalacze.

Warunki integralności semantycznej mogą dotyczyć różnorodnych cech danych. Rozważmy dwa typowe przypadki:

- Ograniczenia nakładane na dziedziny atrybutów

Przykłady ograniczeń:

- wiek osoby powinien być liczbą nieujemną i nie większą od np. 120,
- klasa lub rok studiów mogą być wyrażone liczbą jednocyfrową dodatnią, ograniczoną od góry zależnie od typu szkoły,
- kolor powinien być wyrażony jedną z możliwych wartości z wcześniej zadeklarowanego zbioru,
- ładowność ciężarówki może być określona liczbą dodatnią ograniczoną od góry przez znaną wielkość w danym przedsiębiorstwie.

Ograniczenia tego typu mogą być definiowane i sprawdzane w serwerze bazy danych, np. przy użyciu wyzwalaczy, a także w programie użytkowym.

- Więzy referencyjne

Więzy referencyjne dotyczą zależności między krotkami (wierszami, rekordami), które się ujawniają po zdefiniowaniu tzw. kluczy głównych i kluczy obcych w tablicach.

Przykładowo:

- przypisanie pracownika do zespołu może dotyczyć tylko zespołu, który już istnieje w bazie danych, tzn. identyfikator zespołu w wierszu opisującym pracownika musi mieć swój odpowiednik w identyfikatorze jakiegoś istniejącego zespołu,
- usunięcie danych o pracowniku nie może mieć miejsca, jeśli istnieją informacje o wypłatach, jakie otrzymał ten pracownik, tzn. identyfikatory pracownika w wierszach dotyczących wypłat są równe identyfikatorowi usuwanego pracownika.

Klucze główne są atrybutami (lub podzbiorami atrybutów), których wartości jednoznacznie identyfikują każdy wiersz w tablicy. Taka definicja pociąga za sobą dwa ograniczenia nakładane na atrybuty kluczowe: wartości klucza nie mogą być puste i muszą być unikalne. Ograniczenia te formułowane są zwykle w momencie definiowania struktury tablicy (np. w języku SQL) lub też w trakcie zakładania indeksu (unikalność klucza).

Klucz obcy jest atrybutem pewnej tablicy będąc równocześnie kluczem głównym innej tablicy. Na przykład identyfikator zespołu jest kluczem głównym w tablicy opisującej pracowników (określa w niej przynależność pracownika do zespołu).

Powiązania między kluczami głównymi i obcymi (więzy referencyjne) wymuszają następujące ograniczenia integralnościowe dotyczące wprowadzania i modyfikacji danych:

- definicji klucza obcego musi odpowiadać definicja klucza głównego (w tablicy, która staje się nadrzędną),
- wartość klucza obcego musi odpowiadać istniejącej wartości klucza głównego w jakimś wierszu tablicy nadrzędnej; dopuszcza się przy tym wartości puste kluczy obcych.

Usuwanie wierszy z tablicy nadrzędnej może zostać powiązane (alternatywnie) z dodatkowymi ograniczeniami:

- ograniczenie kaskadowe: usunięcie wiersza z tablicy nadrzędnej powoduje usunięcie wszystkich wierszy logicznie z nim związanych z tablicy podrzednej lub
- ograniczenie restrykcyjne: usunięcie wiersza z tablicy nadrzędnej jest możliwe, jeśli w tablicy podrzędnej brak wierszy logicznie związanych z usuwanym, lub
- ograniczenie z usuwaniem wartości wiążących: usunięcie wiersza z tablicy nadrzędnej powoduje wstawienie wartości pustych w miejsce klucza obcego we wszystkich rekordach logicznie związanych z rekordem usuwanym.

Przykład: definicje dwu tablic Zespoły i Pracownicy z uwzględnieniem więzów restrykcyjnych można zapisać w języku SQL dla systemu Gupta/Centura następująco:

```
CREATE TABLE Zespoły (id_zesp INT NOT NULL, nazwazesp CHAR(30),
id_kier_zesp INT, PRIMARY KEY (id_zesp));

CREATE TABLE Pracownicy (id_prac INT NOT NULL, nazwisko CHAR(15), adres
CHAR(50), id_zesp INT, PRIMARY KEY (id_prac), FOREIGN KEY ko(id_zesp)
REFERENCES Zespoły ON DELETE RESTRICT);
```

W przykładzie tym identyfikator zespołu (id_zesp) jest kluczem głównym w tablicy Zespoły oraz kluczem obcym w tablicy Pracownicy. Kluczem głównym w tablicy Pracownicy jest identyfikator pracownika (id_prac), zaś usuwanie wierszy z nadrzędnej tablicy Zespoły jest ograniczone restrykcyjnie.

4.2. Mechanizmy zabezpieczeń integralności transakcyjnej

Podstawowym pojęciem dla rozważań w tym punkcie jest transakcja. Na potrzeby tej pracy przyjmiemy jedną z definicji transakcji, określającą ją jako ciąg operacji na bazie danych tworzących pewną całość. Klasycznym przykładem transakcji może być przepisanie pewnej kwoty K z konta X na konto Y:

```
T: początek
   odczyt X
   X := X - K
   zapis X
   odczyt Y
   Y := Y + K
   zapis Y
   koniec
```

Naturalnym kryterium integralności bazy danych w tym przykładzie jest zgodność sumy kont przed i po wykonaniu transakcji. Istnieją dwa typowe rodzaje zagrożeń tak zdefiniowanej integralności wymagające odrębnych mechanizmów ochrony. Omówimy je w kolejnych punktach.

4.2.1. Mechanizmy sterowania współbieżnym dostępem do danych

Pierwszym rodzajem zagrożeń jest dopuszczenie do współbieżnej realizacji wielu transakcji operujących na tej samej bazie danych. Załóżmy, np. że współbieżnie są realizowane dwie transakcje T1 i T2 o strukturze takiej jak transakcja T, przy czym w T1 jest przepisywana z konta A na konto B kwota K1, natomiast w T2 jest przepisywana z konta B na konto C kwota K2. Przyjmijmy, że kolejność wykonywania operacji elementarnych obu transakcji (zwana harmonogramem) przedstawia się następująco:

T1	T2
odczyt A	
A:= A - K1	
	odczyt B
zapis A	
	B:= B - K2
odczyt B	
	zapis B
B:= B + K1	
	odczyt C
zapis B	
	C:= C + K2
	zapis C

W prezentowanym przykładzie kolejność wykonanych operacji, a w szczególności odczyt stanu konta B w transakcji T1 przed zapisem stanu tego konta w transakcji T2 spowoduje, że końcowy stan konta B będzie równy B_0 + K1, zamiast spodziewanego B_0 - K2 + K1 (gdzie A_0, B_0 i C_0 reprezentują początkowe stany kont). Tak więc suma stanów kont dla przedstawionego harmonogramu wyniesie $A_0+B_0+C_0+K2$, zamiast spodziewanej wartości $A_0+B_0+C_0$.

Naruszy to integralność bazy danych określoną, jak już wspomniano, zgodnością sumy stanów kont przed i po wykonaniu obu transakcji.

Dla zapobiegania problemom podobnym do przedstawionego stosowane są mechanizmy ochronne. Najczęściej są to mechanizmy blokad, wymagające najpierw uzyskania prawa dostępu do danej (nałożenie blokady), a dopiero potem wykonania odczytu lub zapisu danej.

Nałożenie w powyższym przykładzie w transakcji T2 blokady na dostęp do konta B uniemożliwi transakcji T1 odczyt konta B przed zapisem w transakcji T2. Zapewni to spójność bazy danych.

Kolejność nakładania i zwalniania blokad może być ograniczona dodatkowymi warunkami. Szczególnie ważny jest tzw. dwufazowy protokół blokowania [4] wymagający w ramach transakcji najpierw nałożenia wszystkich blokad, a dopiero potem zezwalający na zwolnienie pierwszej blokady. Protokół taki zapewnia szeregowalność współbieżnie wykonywanych transakcji. Powiązanie blokad z prowadzeniem dziennika transakcji (patrz p.4.2.2) nakazuje zwalnianie blokad nałożonych na dane, które w ramach transakcji zostały zmodyfikowane dopiero po zakończeniu transakcji.

Stosowanie blokad (równoważnych semaforom binarnym - znanym w dziedzinie systemów operacyjnych [5]) wprowadza w proces sterowania współbieżnym dostępem do baz danych niebezpieczeństwo wzajemnej blokady wykonywanych równocześnie transakcji. Znane metody unikania wzajemnej blokady [5] nie są na ogół przydatne w systemach baz danych, m.in. ze względu na ogromną liczbę zasobów, którymi są zwykle pojedyncze rekordy danych. W systemach baz danych na ogół nie stosuje się ochrony przed wzajemną blokadą, a tylko wykrywanie jej wystąpienia. Wykorzystanie dziennika transakcji pozwala wtedy na przerwanie i wycofanie jednej z zakleszczonych transakcji, tzn. na działanie systemu zarządzania transakcjami w sposób podobny jak przy błędach bądź awariach oprogramowania.

Implementacja programowa mechanizmów blokad w systemach baz danych ma różną postać. W wielu systemach dostępne są operacje bezpośredniego nakładania blokady na całe tablice lub wiersze tablic (rekordy). W serwerach baz danych wykorzystujących język SQL stosuje się natomiast pośrednią technikę blokowania przez określanie tzw. poziomów izolacji współbieżnie realizowanych transakcji.

Jak kilkakrotnie podkreślano, mechanizmy blokad są zwykle powiązane z zarządzaniem transakcjami. Ten aspekt ochrony danych zostanie omówiony w kolejnym punkcie.

4.2.2. Mechanizmy zarządzania transakcjami

Drugim rodzajem zagrożeń, które zasygnalizowano na wstępie tego rozdziału, jest przerwanie ciągu operacji tworzących transakcję w wyniku awarii oprogramowania lub sprzętu, bądź też w wyniku wystąpienia sytuacji, która uniemożliwi dokończenie transakcji. Po ponownym uruchomieniu systemu komputerowego stan bazy danych może nie być spójny, ponieważ nie wszystkie operacje w ramach transakcji zostały wykonane. Przykładem może być przerwanie transakcji przenoszenia z konta na konto po zmodyfikowaniu tylko konta X. Dla ochrony integralności bazy danych w takich przypadkach realizowana jest zasada "wszystko albo nic", tzn. albo wszystkie operacje na bazach danych udało się wykonać, albo - w przypadku niepowodzenia, efekty częściowego wykonania transakcji należy wycofać z bazy danych. Najczęściej są wtedy stosowane mechanizmy zarządzania transakcjami oparte na prowadzeniu tzw. dziennika transakcji. Dziennik transakcji (ang. log) jest plikiem,

w którym odnotowywane są wszystkie modyfikacje bazy danych. Ogólną zasadą jest, że każda modyfikacja bazy danych musi zostać poprzedzona zapisem do dziennika informacji o operacji, która ma zostać wykonana. Do informacji tych należą m.in. identyfikator transakcji, adres danej, która ma zostać zmodyfikowana, nowa wartość danej. W najczęściej stosowanej metodzie prowadzenia dziennika i zarządzania transakcjami (tzw. metodzie z opóźnionym zapisem do bazy [3]) wszystkie operacje modyfikacji bazy danych wykonywane w ramach jednej transakcji są odnotowywane tylko w dzienniku. Dopiero po poleceniu sygnalizującym koniec transakcji, a ściślej osiągnięcie tzw. punktu wypełnienia transakcji (ang. commit), program zarządzający transakcjami przepisuje wszystkie modyfikacje do bazy danych i odnotowuje ten fakt (wypełnienie transakcji) w dzienniku. Taka technika pozwala łatwo wycofać transakcję, która nie może zostać dokończona. Ponieważ przed zakończeniem transakcji nie nastąpiła żadna zmiana w bazie danych, więc wycofanie polega tylko na odnotowaniu tego faktu w dzienniku i poinformowaniu użytkownika o takim zdarzeniu. Można wskazać co najmniej trzy rodzaje przyczyn wycofywania transakcji:

1) W samej transakcji wystąpiła sytuacja uniemożliwiająca jej dokończenie, np. brak danych (w przykładzie dotyczącym kont - brak konta Y). W takim przypadku transakcja sama zleca wycofanie samej siebie, służy do tego w języku SQL instrukcja ROLLBACK.

2) W transakcji ujawnił się błąd powodujący jej zawieszenie, bądź też wystąpiła wzajemna blokada kilku transakcji. Wtedy wycofanie transakcji następuje z inicjatywy Systemu Zarządzania Bazą Danych.

3) Nastąpił tzw. upadek systemu w wyniku poważnego błędu na poziomie Systemu Zarządzania Bazą Danych lub systemu operacyjnego, bądź też w wyniku awarii sprzętu (np. zanik zasilania). Po "podniesieniu" systemu operacyjnego system zarządzania transakcjami analizuje zawartość dziennika i wycofuje transakcje nie zakończone.

W przypadku kiedy dziennik transakcji jest umieszczony na innym dysku niż baza danych, a także systematycznie jest prowadzona archiwizacja bazy danych, można odtworzyć bazę danych do stanu spójnego nawet przy zniszczeniu dysku zawierającego bazę danych. Służy do tego instrukcja ROLLFORWARD korzystająca z archiwalnej kopii bazy oraz dziennika transakcji.

Ochrona integralności transakcyjnej jest na ogół zadaniem znacznie trudniejszym w **systemach rozproszonych baz danych.** Problemem, który zwiększa tu skalę trudności, jest replikacja danych, tzn. powielenie części danych (np. tablicy) w kilku węzłach. Takie rozwiązanie, znakomicie przyspieszające odczyt danych, wymaga zagwarantowania ciągłej zgodności zreplikowanych danych. W systemach zarządzania rozproszonymi bazami danych stosuje się z tego względu złożone protokoły blokowania, np. metoda nakładania jednej blokady w przypadku odczytu, a wszystkich blokad w przypadku zapisu, metoda większościowa, metoda węzła centralnego lub węzłów pierwotnych i inne.

W zakresie zarządzania transakcjami stosuje się dwu- lub trójfazowy protokół wypełniania transakcji i wiele ich wariantów (aktualizacje synchroniczne kopii równorzędnych, aktualizacje asynchroniczne w przypadku kopii master/slave i inne). Ze względu na

ograniczony zakres niniejszej pracy problem ochrony integralności w systemach rozproszonych baz danych nie będzie szerzej rozwijany.

5. Podsumowanie

W pracy dokonano przeglądu zagrożeń dla danych przechowywanych w bazach danych oraz mechanizmów chroniących dane przed tymi zagrożeniami.
W problemach tych wyróżniono dwa obszary:

- zabezpieczenie przed umyślnymi próbami nielegalnego odczytu lub modyfikacji danych przez nieuprawnionych użytkowników,
- ochrona danych przed efektami przypadkowych zdarzeń, takich jak błędy logiczne w operacjach na danych, kolizje w trakcie współbieżnego dostępu do danych, awarie oprogramowania i sprzętu komputerowego.

Realizacja takich zadań stanowi znaczną część funkcji wykonywanych przez Systemy Zarządzania Bazami Danych. Ocenia się, że obecnie mechanizmy dotyczące drugiego obszaru są znacznie bardziej złożonymi (i kosztownymi) od mechanizmów pierwszego obszaru. Konsekwencją jest lepsza ochrona danych przed zdarzeniami przypadkowymi (ochrona integralności) od zabezpieczeń przed nielegalnym dostępem do danych.

LITERATURA

1. Denning D.E.: Kryptografia i ochrona danych. WNT, Warszawa 1992.
2. Elmasri R., Nawathe S.: Fundamentals of Database Systems. The Benjamin/Cummings Publishing Company 1989.
3. Silberschatz A., Korth H., Sudarshan S.: Database System Concepts. McGraw-Hill 1996.
4. Ullman J.D., Widom J.: A First Course In Database Systems. Prentice-Hall 1997.
5. Węgrzyn S.: Podstawy informatyki. PWN, Warszawa 1982.
6. Oracle 7 Server Concepts Manual. Oracle Corporation 1992.
7. Oracle 7 Server Administrator's Guide. Oracle Corporation 1992.

Recenzent: Dr hab. inż. Stanisław Wołek, Prof. Pol. Rzeszowskiej

Wpłynęło do Redakcji 31 maja 1999 r.

Abstract

The paper presents the review of mechanisms assuring security and integrity of databases. The databases need to be protected from unauthorized or malicious access and accidental introduction of inconsistency. Database security refers to protection from unauthorized or malicious access. Presented in the paper security mechanisms include: account (user) creation, privileges granting and revoking, view creation and data encryption. Statistical database security mechanisms are also reviewed.

Database integrity refers to protection the database from accidental loss of consistency. Term integrity covers both semantic integrity and transaction integrity. Among semantic integrity constraints are presented: value set (domain) constraints and referential integrity constraints. Presented in the paper transaction integrity mechanisms include the concurrency control and recovery techniques.