Autoryzacja
Uwierzytelnianie

Autoryzacja

- Można przypisać użytkownikowi kilka form autoryzacji na częściach bazy danych:
  - **Read** – pozwala czytać, ale nie na modyfikować dane
  - **Insert** – pozwala wstawiać nowe dane, ale nie na modyfikować istniejące.
  - **Update** – pozwala modyfikować, ale nie na usuwać dane.
  - **Delete** – pozwala usuwać dane.

- Każdy z tych rodzajów autoryzacji nazywany jest uprawnieniem. Można przydzielić użytkownikowi wszystkie, żadne lub kombinację tych typów uprawnień na określonych częściach bazy danych, takich jak relacja lub widok

Autoryzacja c.d.

- Formy autoryzacji do modyfikowania schematu bazy danych
  - **Index** – pozwala tworzyć i usuwać indeksy.
  - **Resources** – pozwala tworzyć nowe relacje.
  - **Alteration** – pozwala dodawać lub usuwać atrybuty w relacji.
  - **Drop** – pozwala usuwać relacje.
- Najwyższa forma autoryzacji przekazywana administratorowi bazy danych
  - autoryzacja nowych użytkowników
  - restrukturyzacja bazy danych

Specyfikacja autoryzacji w SQL

- Wyrażenie **grant** do przyznawania uprawnień

  grant <privilege list> **on** <relation name or view name>

  to <user/role list>

- <user list> to:
  - identyfikator użytkownika
  - **public**, pozwala przekazywać uprawnienia wszystkim obecnym i przyszłym użytkownikom systemu
  - rola
- Np:
  - **grant select on** *department* **to** *Amit, Satoshi*
- Przyznanie uprawnienia do widoku nie oznacza przyznania żadnych uprawnień do podstawowych relacji.
- Udzielający uprawnienia musi już posiadać uprawnienia do określonego elementu (lub być administratorem bazy danych).

Uprawnienia w SQL

- **select:** umożliwia dostęp do przeczytania relacji, lub możliwość zapytania za pomocą widoku
  - Np: przekaż użytkownikom U_1, U_2, i U_3 prawnienie **select** do relacji *instructor*:

    grant select on *instructor* **to** U_1, U_2, U_3
- **insert**: możliwość wstawiania krotek (można podać atrybuty, reszta na *null* lub *default*)
- **update**: możliwość aktualizacji za pomocą instrukcji aktualizacji SQL (wszystkich lub wybranych atrybutów)

  grant update *(budget)* **on** *department* **to** *Amit, Satoshi*
- **delete**: możliwość usuwania krotek.
- **all privileges**: krótka forma dla wszystkich dopuszczalnych uprawnień

Odwoływanie uprawnień w SQL

- Wyrażenie **revoke** jest używane do odwoływania autoryzacji.

  revoke <privilege list> **on** <relation name or view name>

  from <user/role list>

- Np:

  revoke select on *student* **from** U_1, U_2, U_3

  revoke select on *department* **from** *Amit, Satoshi*

  revoke update *(budget)* **on** *department* ***from*** *Amit, Satoshi*

- <privilege-list> może być **all** aby odwołać wszystkie posiadane przez użytkownika uprawnienia.
- Jeżeli <revokee-list> zawiera **public,** wszyscy tracą uprawnienia poza tymi, którzy dostali je jawnie.
- Jeśli ten sam przywilej został przyznany dwa razy temu samemu użytkownikowi przez różnych użytkowników, użytkownik ten może zachować przywilej po odwołaniu.
- Wszystkie przywileje, które zależą od odwołanego, są również odwoływane

Role

- **Rola** to sposób na rozróżnienie różnych użytkowników w zakresie, w jakim użytkownicy ci mogą uzyskiwać dostęp/aktualizować bazę danych.
- Aby utworzyć nową rolę:

  create a role <name>
- Np:
  - **create role** instructor
- Po utworzeniu roli, można przypisać użytkowników do roli za pomocą:
  - **grant** <role> **to** <users>

Przykłady ról

Przykładowe role w bazie uniwersyteckiej:

instructor, teaching_assistant, student, dean, department_chair

- **create role** *instructor*;
- **grant** *instructor* **to** *Amit*
- Uprawnienia mogą być przekazywane rolom:
  - **grant select on** *takes* **to** *instructor*;
- Role mogą być przekazywane użytkownikom jak również innym rolom
  - **create role** *teaching_assistant*;
  - **grant** *teaching_assistant* **to** *instructor*;
    - *Instructor* dziedziczy wszystkie uprawnienia od *teaching_assistant*
- Łańcuch ról
  - **create role** *dean*
  - **grant** *instructor* **to** *dean*;
  - **grant** *dean* **to** Satoshi

Autoryzacje na widokach

- **create view** *geo_instructor* **as**
  (**select** *
  from *instructor*
  where *dept_name* = 'Geology');
- **grant select on** *geo_instructor* **to** *geo_staff*
- Twórca widoku nie dostaje wszystkich do niego uprawnień
  - tylko takie, które nie wymagają dodatkowych ponad te, które posiada
  - twórca *geo_instructor* musi mieć uprawnienia **select** na relacji *instructor*
- Widoki mająwszystkie uprawnienia jakie ma twórca.

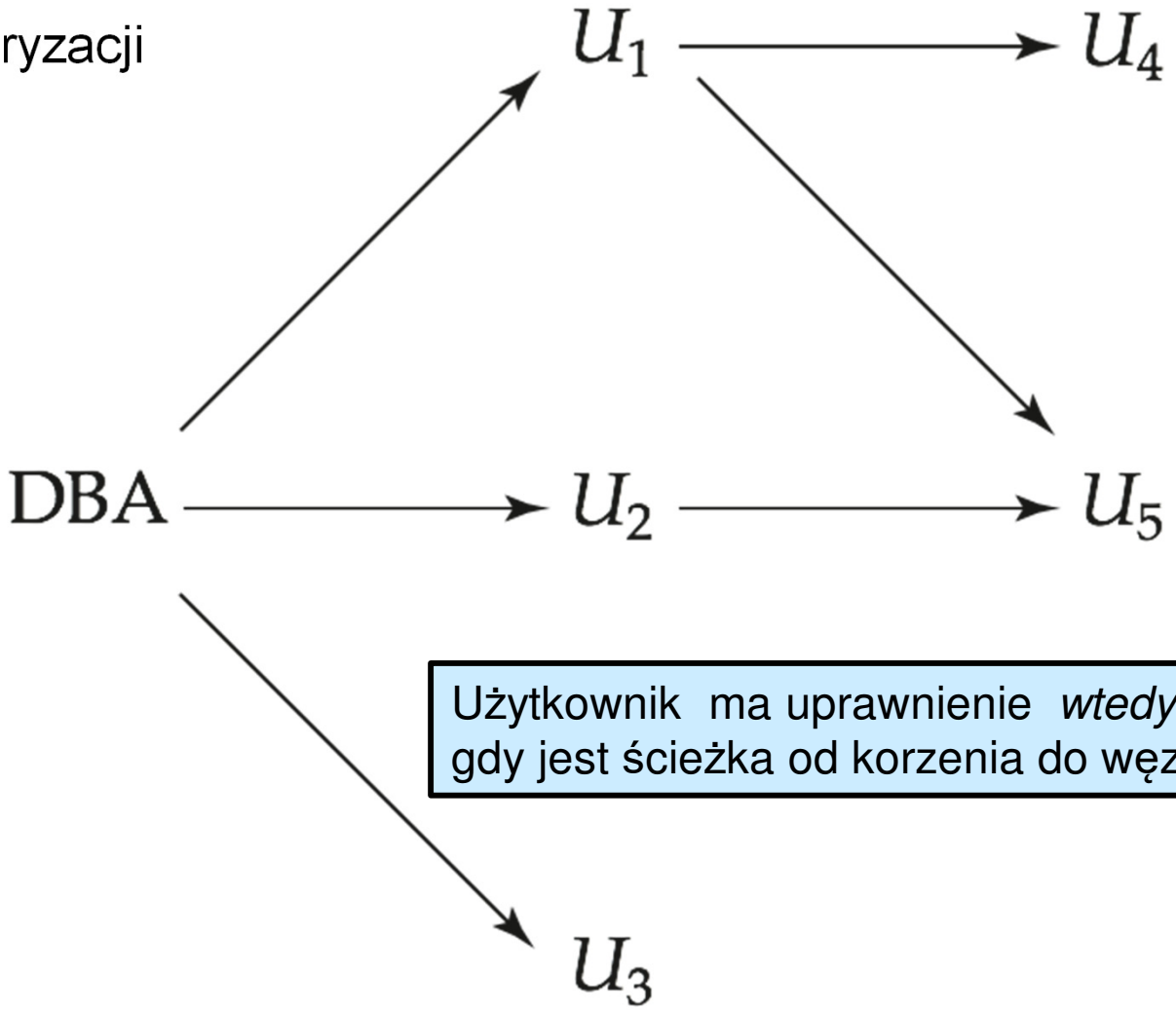
Autoryzacje na schematach

- W standardzie SQL prymitywne mechanizmy autoryzacji schematów
  - tylko właściciel może modyfikować schemat
    - tworzenie/usuwanie relacji, dodawanie/usuwanie atrybutów, dodawanie/usuwanie indeksów
- Uprawnienie **references** do deklarowania kluczy obcych
  - **grant reference** (*dept_name*) **on** *department* **to** Mariano;

- Podobnie uprawnienia r**eferences** na *department* aby utworzyć **check constraint** na relacji **r**, jeżeli constraint ma podzapytanie, które odwołuje się do *department*
  - *check* ogranicza potencjalne modyfikacje relacji

Przekazywanie uprawnień

- domyślnie: użytkownik/rola nie ma uprawnień do przekazywania innym
- przekazywanie uprawnień
  - **grant select on** *department* **to** Amit **with grant option**;
- Graf autoryzacji



Użytkownik ma uprawnienie *wtedy i tylko wtedy*
gdy jest ścieżka od korzenia do węzła go reprezentującego

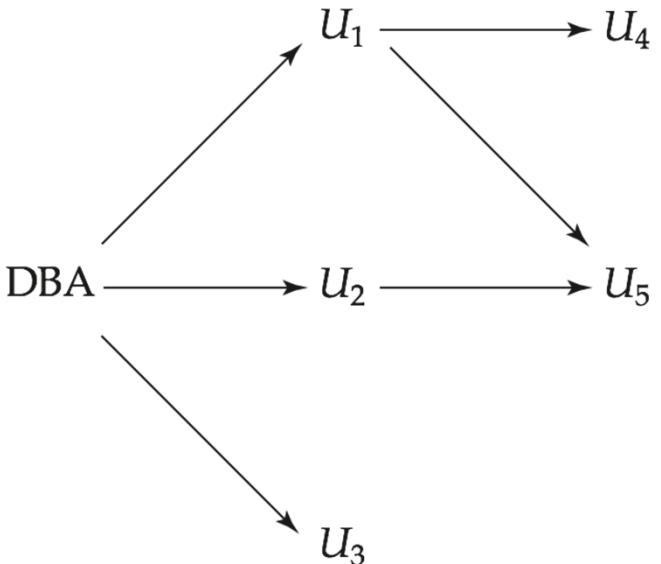
Odwoływanie uprawnień

- Jeżeli to samo uprawnienie zostało przydzielone 2 razy temu samemu użytkownikowi przez różnych użytkowników, może zachować po odwołaniu.
- Wszystkie przywileje, które zależą od cofniętego są również odwoływane.
- Kaskadowe odwoływanie (domyślnie w większości systemów):

  revoke select on *department* **from** Amit, Satoshi **cascade**;
- Blokowanie:

  revoke select on *department* **from** Amit, Satoshi **restrict**;
- Odwoływanie możliwości przekazywania uprawnień:

  revoke grant option for select on *department* **from** Amit;



Odwoływanie uprawnień c.d.

- Kaskadowe odwoływanie niekorzystne w wielu sytuacjach:
  - Satoshi ma rolę *dean*
  - Przekazuje rolę *instructor* do Amit
  - Rola *dean* odwołana dla Satoshi (opuścił uniwersytet)
  - Ale Amit musi mieć w dalszym ciągu rolę *instructor*
- Lepiej przekazywać uprawnienia przez rolę a nie użytkownika