

Materiały do przedmiotu:

Bazy danych

Laboratorium nr 7:

**Zarządzanie użytkownikami, przydzielanie
uprawnień i zarządzanie dostępem do bazy
danych. Kopia zapasowa, import i eksport danych**

Autor:

Albert Rachwał



Fundusze Europejskie
dla Rozwoju Społecznego



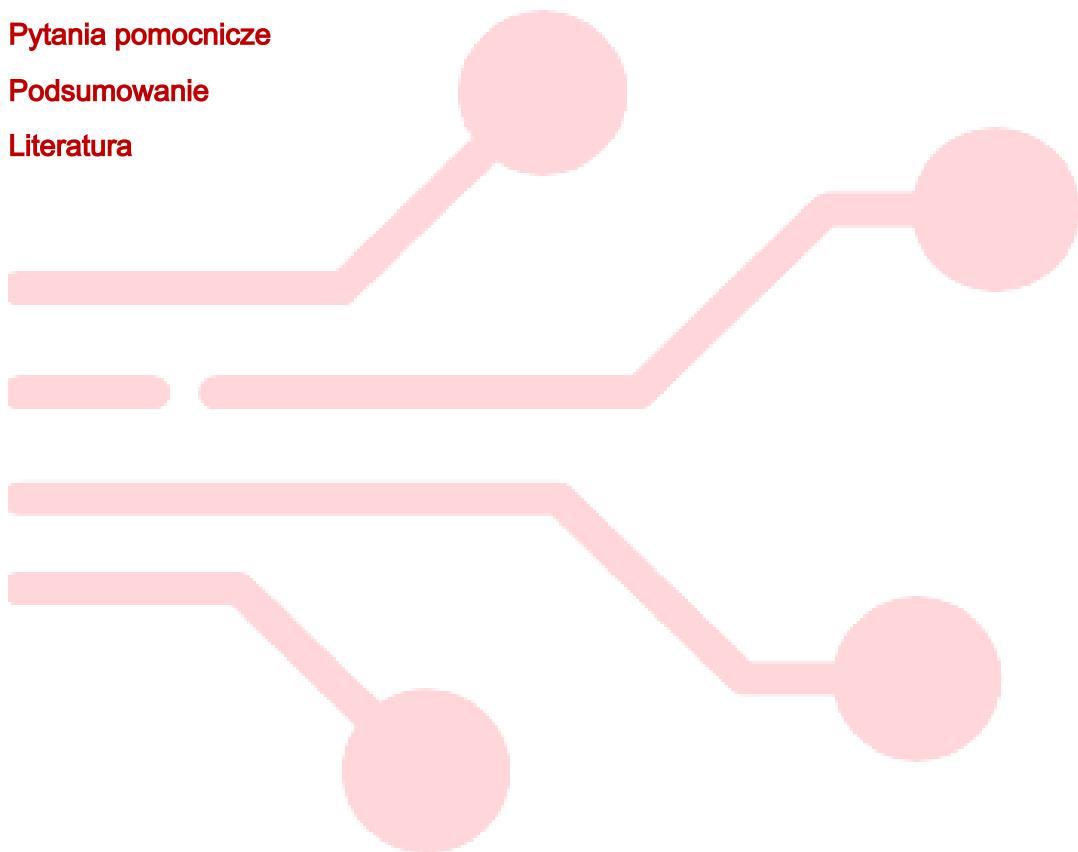
Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Spis treści

Wprowadzenie, tematyka zajęć	3
Cel ćwiczenia/laboratorium	3
Instrukcja laboratoryjna/ćwiczeniowa	3
Ćwiczenia samodzielne	9
Pytania pomocnicze	10
Podsumowanie	10
Literatura	11



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Wprowadzenie, tematyka zajęć

Bezpieczeństwo i stabilność bazy danych zależy nie tylko od jej struktury, ale także od odpowiedniego zarządzania użytkownikami i ich uprawnieniami. Administrator musi kontrolować, kto i w jaki sposób uzyskuje dostęp do danych. Dodatkowo niezwykle istotne jest tworzenie kopii zapasowych oraz umiejętność przywracania danych w razie awarii. Dopełnieniem tych działań są mechanizmy importu i eksportu, które pozwalają przenosić dane między systemami.

Cel ćwiczenia/laboratorium

- Poznanie pojęcia użytkownika bazy danych.
- Tworzenie nowych użytkowników w bazie.
- Nadawanie uprawnień użytkownikom.
- Odbieranie uprawnień użytkownikom.
- Tworzenie ról i grup użytkowników.
- Zrozumienie znaczenia ról w zarządzaniu dostępem.
- Ćwiczenie z nadawania uprawnień SELECT, INSERT, UPDATE, DELETE.
- Poznanie pojęcia kopii zapasowej bazy danych.
- Tworzenie kopii zapasowych.
- Odtwarzanie danych z kopii zapasowej.
- Poznanie metod importu danych.
- Poznanie metod eksportu danych.
- Zrozumienie różnicy między importem a eksportem.
- Ćwiczenie z importu danych CSV do bazy.
- Omówienie dobrych praktyk w zarządzaniu użytkownikami.

Instrukcja laboratoryjna/ćwiczeniowa

Użytkownicy bazy danych – pojęcie, cel i zasady

Użytkownik bazy danych to tożsamość, pod którą serwer DBMS dopuszcza wykonywanie operacji. W MySQL/MariaDB tożsamość tę określa się parą nazwa–host, dzięki czemu „anna@localhost” i „anna@10.0.0.%” to dwa różne konta o potencjalnie zupełnie innych uprawnieniach i sposobach logowania. Sens istnienia użytkowników jest podwójny: po pierwsze rozdzielają oni dostęp do danych i funkcji



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską





systemu między konkretne osoby, aplikacje i usługi; po drugie wymuszają kontrolę nad tym, kto co może zrobić oraz pozwalają później odtworzyć „kto, kiedy i jak” działał na bazie. Konto użytkownika obejmuje mechanizm uwierzytelniania (hasło, czasem certyfikat X.509), reguły bezpieczeństwa (wymóg SSL/TLS, polityka haseł, długość, rotacja, blokada po błędnych logowaniach), ograniczenia kontekstowe (z jakich hostów wolno się łączyć) oraz – co najważniejsze – zestaw przydzielonych uprawnień. W dojrzałych instalacjach rozróżnia się konta osobiste administratorów i analityków, które podlegają pełnej polityce haseł i audytowi, od kont usługowych, pod którymi łączy się kod aplikacji. Te drugie mają zwykle bardzo wąski zakres możliwości: tylko to, czego naprawdę potrzebuje dana usługa w swoim podstawowym scenariuszu, i nic więcej. Całość spina zasada najmniejszych uprawnień: tworząc konto, nadajemy minimalny niezbędnny zestaw praw, ograniczamy miejsca logowania do konkretnych adresów lub podsieci, wymagamy połączeń szyfrowanych i cyklicznej zmiany haseł, a w razie potrzeby konto czasowo blokujemy zamiast usuwać, by zachować historię przypisań. Uprawnienia w MySQL/MariaDB można nadawać na wielu poziomach szczegółowości – globalnie dla całego serwera, dla wybranego schematu, dla pojedynczej tabeli, a nawet dla wskazanych kolumn, a także dla obiektów proceduralnych takich jak procedury i funkcje. Dzięki temu można precyzyjnie dopasować widoczność i zakres modyfikacji do roli danej osoby lub aplikacji: analityk dostaje wyłącznie prawo SELECT do widoków raportowych, operator importu tylko INSERT do wybranych tabel stagingowych, a administracja kadrowa UPDATE wyłącznie do kolumn nietajnych. Cykl życia konta powinien być formalny i przewidywalny: utworzenie z przeglądem ryzyk, bieżące przeglądy i redukcje nadanych praw, okresowa zmiana hasła lub jego unieważnienie, blokada przy odejściu użytkownika z organizacji oraz całkowite usunięcie dopiero po upewnieniu się, że nie ma już zależności technicznych i nie będzie konieczne odtwarzanie działań z przeszłości.

Role i delegacja uprawnień – jak porządkować dostęp

Role to abstrakcja grupująca uprawnienia w logiczne paczki, które następnie przypisuje się użytkownikom. Zamiast nadawać kilkanaście identycznych GRANT-ów każdej nowej osobie w zespole, definiuje się rolę „analytik_raportowy”, do której



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską





trafiają prawa odczytu do właściwych schematów i widoków; przypisanie roli użytkownikowi natychmiast daje mu komplet przygotowanych uprawnień, a odebranie roli – natychmiast je cofa. Role wspierają porządek, skalowalność i zgodność z politykami bezpieczeństwa, a przy tym ułatwiają audyt, bo widać nie tylko „kto co ma”, ale też „z jakiego powodu” to ma. W systemach z rozbudowanym RBAC często buduje się hierarchię ról od ogólnych do wyspecjalizowanych, wyraźnie rozdzielając obowiązki administracyjne (tworzenie struktur, zmiany schematu), operacyjne (importy, wsadowe aktualizacje), analityczne (odczyt hurtowni, widoki) i serwisowe (zadania utrzymywaniowe). Delegacja uprawnień musi być świadoma: w MySQL/MariaDB istnieje rozróżnienie pomiędzy samym posiadaniem uprawnienia a prawem do jego dalszego przekazywania innym. To drugie zapewnia atrybut GRANT OPTION przy klasycznych uprawnieniach i ADMIN OPTION przy rolach; stosuje się je oszczędnie, wyłącznie tam, gdzie faktycznie przewidziana jest decentralizacja administracji, ponieważ nieostrożne użycie prowadzi do cichej eskalacji dostępu i trudnych do uchwycenia łańcuchów dziedziczenia. Mechanizm ról dobrze łączy się z technikami „warstwy dostępu”, takimi jak widoki i procedury o bezpieczeństwie „definer”: użytkownik końcowy często nie musi mieć praw do tabel źródłowych, wystarczy mu rola dająca EXECUTE na procedurach realizujących zatwierdzone ścieżki modyfikacji albo SELECT na widokach maskujących dane wrażliwe. W praktyce wdrożeniowej warto ustalić katalog ról referencyjnych, ich domyślność po zalogowaniu oraz cykl regularnego przeglądu – rola powinna odzwierciedlać stanowisko i zakres obowiązków, a nie „historię projektu”. Jeśli organizacja korzysta z wielu środowisk (dev/test/prod), te same role powinny istnieć w każdym z nich, lecz różnić się zakresem i siłą uprawnień zgodnie z polityką separacji. Dzięki roliom i kontrolowanej delegacji przenosimy zarządzanie dostępem z poziomu pojedynczych, rozproszonych GRANT-ów na poziom zrozumiałych, biznesowo nazwanych pakietów, co czyni system bezpieczniejszym, łatwiejszym w utrzymaniu i bardziej odpornym na błędy ludzkie.

Audit i przegląd użytkowników oraz uprawnień

Audit użytkowników i okresowe przeglądy uprawnień są fundamentem bezpiecznego zarządzania bazą danych. Chodzi w nich o regularną weryfikację, kto ma dostęp do systemu, jakie prawa posiada i czy odpowiadają one jego aktualnym obowiązkom. W praktyce oznacza to nie tylko sprawdzenie listy aktywnych kont, ale także analizę



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską





logów logowania, historii wykonywanych poleceń oraz zestawienie nadanych ról i przywilejów. Audit powinien ujawniać sytuacje niepożądane: konta osierocone po odejściu pracownika, uprawnienia nadane „tymczasowo” i nigdy nieodebrane, czy też nadmiarowe prawa, które wykraczają poza faktyczne potrzeby użytkownika. Przegląd uprawnień musi być procesem cyklicznym, wpisanym w politykę bezpieczeństwa organizacji – wykonywanym np. raz na kwartał lub przy każdej istotnej zmianie personalnej. Warto, aby w ramach audytu stosować zasadę czterech oczu: przeglądu dokonuje inna osoba niż ta, która nadała uprawnienia, co minimalizuje ryzyko błędów i nadużyć. Rezultatem audytu powinny być nie tylko raporty, ale także konkretne działania – odbieranie zbędnych praw, blokowanie nieaktywnych kont, a w skrajnych przypadkach eskalacja incydentów bezpieczeństwa. Systemy DBMS, takie jak MySQL czy MariaDB, umożliwiają prowadzenie logów binarnych i general logs, które w połączeniu z narzędziami monitorującymi mogą służyć jako źródło danych do audytu. Ostatecznym celem audytu jest zapewnienie zgodności z polityką bezpieczeństwa i regulacjami prawnymi, a także zbudowanie przejrzystości w tym, kto ma realny wpływ na dane.

Kopia zapasowa, oraz import i eksport danych.

Kopia zapasowa w MySQL to podstawowy mechanizm zabezpieczenia danych przed ich utratą, uszkodzeniem lub nieautoryzowaną modyfikacją. Jest to proces tworzenia pełnej lub częściowej kopii bazy danych w formie, która pozwala później odtworzyć jej stan w razie awarii sprzętu, błędu administratora czy ataku złośliwego oprogramowania. W świecie MySQL kopie zapasowe można wykonywać na kilka sposobów. Najczęściej stosuje się narzędzie mysqldump, które generuje plik zawierający skrypt SQL odtwarzający strukturę i zawartość bazy. Bardziej zaawansowane mechanizmy, takie jak mysqlpump czy narzędzia zewnętrzne (np. Percona XtraBackup), umożliwiają tworzenie kopii przy minimalnym zatrzymaniu pracy serwera oraz wykonywanie kopii przyrostowych lub różnicowych. Istnieją również fizyczne kopie zapasowe, polegające na skopiowaniu plików danych i logów transakcyjnych przechowywanych w katalogach serwera, co pozwala na wierne odtworzenie całego środowiska. Kopia zapasowa ma charakter systemowy – obejmuje nie tylko tabele i ich zawartość, ale również indeksy, widoki, procedury,



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



wyzwalacze, użytkowników i ich uprawnienia. W praktyce tworzenie backupów odbywa się cyklicznie, w określonych interwałach czasowych, często z wykorzystaniem harmonogramów i automatycznych zadań, a pliki kopii przechowywane są w bezpiecznej lokalizacji, nierzadko poza serwerem produkcyjnym.

Eksport i import danych to procesy, które w MySQL pełnią odmienną funkcję niż backup. Eksport polega na wyciągnięciu danych lub struktury z bazy do pliku w formacie nadającym się do przenoszenia – może to być klasyczny skrypt SQL, ale także formaty takie jak CSV czy JSON, szczególnie gdy dane mają być wykorzystane w innych programach lub systemach. W narzędziach graficznych, takich jak phpMyAdmin czy MySQL Workbench, eksport można wykonać wybierając odpowiednie opcje w menu, a w konsoli poleceniem mysqldump, np. mysqldump -u root -p baza_danych > kopia.sql. Import jest operacją odwrotną – polega na wczytaniu danych z przygotowanego pliku z powrotem do bazy. W przypadku plików SQL najczęściej odbywa się to komendą mysql -u root -p baza_danych < kopia.sql albo poprzez zakładkę „Import” w phpMyAdmin. Import może również dotyczyć plików CSV – wówczas dane ładowane są do konkretnej tabeli, przy czym często wymaga to wcześniejszego przygotowania struktury kolumn.

Różnica pomiędzy eksportem a kopią zapasową jest istotna. Eksport to narzędzie do migracji i przenoszenia danych – pozwala łatwo wyciągnąć tylko wybrane elementy, np. jedną tabelę czy część rekordów, i przenieść je do innego środowiska. Backup natomiast jest kompletnym zabezpieczeniem systemu, tworzonym regularnie i obejmującym całość struktury i danych w taki sposób, by baza mogła zostać w pełni odtworzona po awarii. Eksport zwykle wykonuje się ad hoc, w zależności od potrzeby, natomiast backup powinien być elementem stałej polityki bezpieczeństwa organizacji. Oba mechanizmy się uzupełniają: eksport jest wygodny przy pracy programistycznej i integracji, natomiast backup daje gwarancję bezpieczeństwa i ciągłości działania systemu.

Tworzenie użytkowników w phpMyAdmin

1. Zaloguj się do phpMyAdmin (<http://localhost/phpmyadmin>).



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



2. Wybierz zakładkę Konta użytkowników w menu górnym.
3. Kliknij przycisk Dodaj użytkownika.
4. W formularzu podaj:
 - Nazwę użytkownika (np. student1),
 - Host (najczęściej localhost),
 - Hasło (dwukrotnie).
5. Możesz zaznaczyć opcję Utwórz bazę danych o takiej samej nazwie i przyznaj wszystkie uprawnienia, jeśli chcesz, aby użytkownik od razu miał swoją bazę.
6. W sekcji Globalne uprawnienia wybierz, jakie prawa przydzielić (np. SELECT, INSERT, UPDATE).
7. Zatwierdź przyciskiem Wykonaj – użytkownik zostanie utworzony.

Tworzenie kopii zapasowej (backup)

Metoda graficzna (phpMyAdmin):

1. Zaloguj się do phpMyAdmin i wybierz z menu po lewej bazę danych do skopiowania.
2. Kliknij zakładkę Eksport.
3. Wybierz tryb Szybki i format SQL (najczęściej stosowany).
4. Kliknij Wykonaj – plik .sql zostanie pobrany na komputer i stanowi kopię bazy.

Metoda konsolowa (mysqldump):

```
mysqldump -u root -p nazwa_bazy > kopia.sql
```

Import danych

Metoda graficzna (phpMyAdmin):

1. Wybierz bazę danych, do której chcesz zaimportować dane.
2. Kliknij zakładkę Import.
3. Wskaż plik .sql (lub np. .csv) z danymi do załadowania.
4. Kliknij Wykonaj – dane zostaną wczytane do bazy.

Metoda konsolowa:

```
mysql -u root -p nazwa_bazy < kopia.sql
```



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Eksport danych

Metoda graficzna (phpMyAdmin):

1. Wybierz bazę lub tabelę, którą chcesz wyeksportować.
2. Kliknij zakładkę Eksport.
3. Wybierz tryb Szybki lub Dostosowany (jeśli chcesz wybrać konkretne tabele).
4. Kliknij Wykonaj – dane zostaną zapisane do pliku.

Metoda konsolowa:

```
mysqldump -u root -p nazwa_bazy > eksport.sql
```

Ćwiczenia samodzielne

1. Utwórz nowego użytkownika w bazie danych o nazwie student_lab7, który będzie mógł logować się wyłącznie z localhosta. Ustaw mu hasło i spróbuj zalogować się na jego konto przez phpMyAdmin lub terminal.
2. Nadaj użytkownikowi student_lab7 uprawnienia tylko do odczytu (SELECT) w jednej wybranej tabeli z bazy danych. Przetestuj, czy potrafi wykonywać zapytania SELECT, a następnie sprawdź, że nie może wstawić nowego wiersza ani usunąć istniejącego.
3. Utwórz użytkownika editor_lab7, który będzie miał prawo do wykonywania zapytań SELECT, INSERT i UPDATE w wybranej bazie danych. Przetestuj jego możliwości.
4. Zabierz wybrane uprawnienia użytkownikowi editor_lab7 (np. UPDATE) i ponownie spróbuj wykonać operację modyfikacji danych. Sprawdź, jak reaguje baza.
5. Utwórz rolę raportowanie, przypisz do niej prawo SELECT w całej bazie i nadaj tę rolę użytkownikowi student_lab7. Zaloguj się ponownie i sprawdź, czy dostęp do danych zmienił się zgodnie z oczekiwaniemi.
6. Zablokuj konto jednego z użytkowników i spróbuj się na nie zalogować. Następnie odblokuj je i powtórz test.
7. Wykonaj kopię zapasową swojej bazy danych:
 - najpierw za pomocą narzędzia graficznego (phpMyAdmin – zakładka Eksport),
 - następnie w terminalu przy użyciu polecenia mysqldump.



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



8. Usuń jedną tabelę z bazy danych, a następnie spróbuj ją odtworzyć z przygotowanej kopii zapasowej.
9. Zaimportuj plik CSV do nowej tabeli w bazie danych, wykorzystując narzędzie graficzne (Import w phpMyAdmin).
10. Przeprowadź import i eksport z terminala:
 - eksport: mysqldump -u root -p moja_baza > backup.sql
 - import: mysql -u root -p moja_baza < backup.sql
11. Sprawdź, czy dane odtworzyły się poprawnie.

Pytania pomocnicze

1. Czym jest użytkownik bazy danych?
2. Jak utworzyć nowego użytkownika?
3. Jakie są podstawowe uprawnienia w bazie danych?
4. Jak odebrać uprawnienia użytkownikowi?
5. Czym jest rola w bazie danych?
6. Jakie są zalety stosowania ról?
7. Podaj przykład nadania uprawnienia SELECT.
8. Podaj przykład nadania uprawnienia INSERT.
9. Podaj przykład nadania uprawnienia DELETE.
10. Czym jest kopia zapasowa bazy danych?
11. Jak utworzyć kopię zapasową?
12. Jak odtworzyć bazę z kopii?
13. Jaki format danych są wykorzystywane do importu?
14. Jakie są metody eksportu danych?
15. Jakie są najlepsze praktyki przy zarządzaniu dostępem?

Podsumowanie

Podczas laboratorium studenci zapoznali się z zagadnieniami zarządzania użytkownikami i kontrolą dostępu w systemach bazodanowych. Ćwiczenia pozwoliły zrozumieć, że poprawne definiowanie kont, ról i uprawnień jest fundamentem bezpieczeństwa danych oraz umożliwia wdrożenie zasady najmniejszych uprawnień. Studenci mieli okazję praktycznie utworzyć konta, przypisać im odpowiednie zestawy



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



praw oraz przetestować różnice w możliwościach poszczególnych użytkowników, co pokazało, jak w praktyce realizowana jest polityka bezpieczeństwa i audit działań w bazie.

Drugim ważnym elementem zajęć była praca z kopiami zapasowymi oraz mechanizmami importu i eksportu danych. Studenci poznali różnicę między pełnym backupem bazy a eksportem danych do pliku oraz przećwiczyli obie metody zarówno w narzędziach graficznych (phpMyAdmin), jak i w pracy z terminalem. Wnioskiem płynącym z tych zadań jest świadomość, że backup jest podstawowym narzędziem ochrony danych przed utratą, natomiast import i eksport pełnią głównie rolę w przenoszeniu i wymianie danych między systemami.

Laboratorium miało charakter nie tylko praktyczny, ale i refleksyjny – pokazało, że bezpieczeństwo i ciągłość działania bazy danych zależą w równym stopniu od właściwego zarządzania użytkownikami, jak i od skutecznych procedur tworzenia kopii zapasowych. Zdobyte doświadczenie ma bezpośrednie zastosowanie w praktyce administracyjnej i stanowi fundament pracy w realnych środowiskach, gdzie dane są strategicznym zasobem organizacji.

Literatura

- Access Control and Account Management,
<https://dev.mysql.com/doc/refman/8.4/en/access-control.html>, (data ostatniego dostępu 25.09.25)
- Backup and Recovery,
<https://dev.mysql.com/doc/refman/8.4/en/backup-and-recovery.html>, (data ostatniego dostępu 25.09.25)
- Database Backup from MySQL,
<https://www.geeksforgeeks.org/sql/database-backup-from-mysql/>, (data ostatniego dostępu 25.09.25)



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską

