TEMAT 19

GRANT, REVOKE i zasada najmniejszych uprawnień

Jak nadawać dokładnie taki dostęp, jaki jest potrzebny

Teoria • przykład • schemat • pytanie • ćwiczenie

Bazy danych • 45 minut • bez komputerów

Cel lekcji i powrót do tematu 18

PRZYPOMNIENIE	Użytkownik ma konto, a rola grupuje uprawnienia.
GRANT	Nadaje wskazane uprawnienie użytkownikowi albo roli.
REVOKE	Odbiera wcześniej nadane uprawnienie.
ZASADA	Dajemy tylko prawa konieczne do wykonania zadania.

Pytanie: czy nauczyciel potrzebuje prawa DELETE do tabeli UCZNIOWIE?

Najważniejsze pojęcia

POJĘCIE	PROSTE WYJAŚNIENIE	PRZYKŁAD
uprawnienie	pozwolenie na określoną operację	SELECT na tabeli uczniowie
GRANT	nadanie prawa	rola może czytać rekordy
REVOKE	odebranie prawa	rola traci możliwość zmiany
najmniejsze uprawnienia	tylko prawa potrzebne do zadania	raporty mają wyłącznie SELECT

Uprawnienia dotyczą konkretnych operacji i konkretnych obiektów bazy.

Operacje na danych i ich skutki

UPRAWNIENIE	CO POZWALA ZROBIĆ	PRZYKŁADOWY SKUTEK
SELECT	odczytać dane	wyświetlić nazwiska uczniów
INSERT	dodać rekord	dopisać nową ocenę
UPDATE	zmienić rekord	poprawić adres ucznia
DELETE	usunąć rekord	usunąć wpis z tabeli
REFERENCES	utworzyć odwołanie FK	połączyć tabelę z kluczem

Im większy skutek operacji, tym ostrożniej nadajemy uprawnienie.

Prosty przykład szkolny

ROLA	ZADANIE	MINIMALNE PRAWA	ZBĘDNE PRAWO
sekretariat	poprawia dane uczniów	SELECT, UPDATE	DROP
nauczyciel	wpisuje oceny	SELECT, INSERT, UPDATE	DELETE uczniów
dyrektor	czyta raporty	SELECT	INSERT
aplikacja WWW	wyświetla plan lekcji	SELECT plan_lekcji	ALTER

Rola otrzymuje prawa wynikające z obowiązków, a nie z wygody administratora.

Schemat: od potrzeby do uprawnienia

1. ZADANIE	2. OBIEKT	3. OPERACJA	4. GRANT
nauczyciel wpisuje oceny	tabela OCENY	INSERT i UPDATE	nadanie praw roli

Najpierw ustalamy zadanie. Dopiero potem wybieramy tabelę i potrzebną operację.

Nie zaczynamy od listy wszystkich możliwych praw.

GRANT: nadanie prawa do odczytu

```
-- Rola raporty może odczytać tabelę uczniowie
GRANT SELECT ON uczniowie TO raporty;
```

GRANT wskazuje operację SELECT, obiekt UCZNIOWIE i odbiorcę RAPORTY.

Po tym poleceniu użytkownik należący do roli raporty może wykonać SELECT.

GRANT: kilka praw do jednej tabeli

```
-- Nauczyciel odczytuje i dopisuje oceny
GRANT SELECT, INSERT
ON oceny
TO nauczyciel;
```

Przecinek rozdziela dwa uprawnienia. Polecenie nie nadaje UPDATE ani DELETE.

Pytanie: czy nauczyciel może poprawić błędnie wpisaną ocenę?

REVOKE: odebranie prawa

```
-- Odbieramy roli prawo usuwania rekordów
REVOKE DELETE ON uczniowie FROM nauczyciel;
```

REVOKE wskazuje prawo DELETE, tabelę UCZNIOWIE i rolę NAUCZYCIEL.

Po wykonaniu polecenia próba DELETE zostanie odrzucona.

GRANT nie zmienia rekordów w tabeli

UPRAWNIENIA

GRANT i REVOKE zmieniają zasady dostępu.

DANE

INSERT, UPDATE i DELETE zmieniają rekordy.

```
GRANT UPDATE ON uczniowie TO sekretariat;
```

To polecenie pozwala zmieniać dane, ale samo nie zmienia żadnego ucznia.

Przewidujemy wynik polecenia

ROLA	POSIADANE PRAWA	POLECENIE	WYNIK
raporty	SELECT uczniowie	SELECT * FROM uczniowie	wykonane
raporty	SELECT uczniowie	UPDATE uczniowie …	odrzucone
nauczyciel	SELECT, INSERT oceny	INSERT INTO oceny …	wykonane
nauczyciel	SELECT, INSERT oceny	DELETE FROM oceny …	odrzucone

Poprawna składnia SQL nie wystarcza. Serwer sprawdza także uprawnienia roli.

Uprawnienia a relacje między tabelami

TABELA	KLUCZ	ROLA NAUCZYCIEL
UCZNIOWIE	uczen_id PK	SELECT
OCENY	ocena_id PK, uczen_id FK	SELECT, INSERT, UPDATE
KLASY	klasa_id PK	SELECT

```
INSERT INTO oceny (uczen_id, wartosc) VALUES (17, 5);
```

Nawet przy prawie INSERT klucz obcy uczen_id musi wskazywać istniejącego ucznia.

Dwa różne rodzaje błędów

SYTUACJA	KOMUNIKAT / SKUTEK	PRZYCZYNA
brak INSERT	permission denied	brak uprawnienia
uczen_id = 999 nie istnieje	foreign key violation	naruszenie relacji
błędne słowo SELCT	syntax error	błąd składni SQL
tabela oceny nie istnieje	relation does not exist	błędna nazwa obiektu

Najpierw rozpoznajemy rodzaj błędu, a dopiero potem szukamy rozwiązania.

Dlaczego ALL PRIVILEGES bywa ryzykowne

```
-- Zbyt szerokie prawo do zwykłej pracy
GRANT ALL PRIVILEGES ON uczniowie TO nauczyciel;
```

Rola może otrzymać więcej praw, niż potrzebuje. Pomyłka lub przejęcie konta wywoła wtedy większe szkody.

Lepszy wybór: nadać osobno tylko SELECT, INSERT lub UPDATE na właściwych tabelach.

Zasada najmniejszych uprawnień

KROK 1	Ustal rzeczywiste zadanie użytkownika.
KROK 2	Wskaż potrzebne tabele, widoki albo procedury.
KROK 3	Nadaj tylko konieczne operacje.
KROK 4	Regularnie odbieraj prawa, które przestały być potrzebne.

Minimalny dostęp zmniejsza ryzyko błędu, nadużycia i skutki przejęcia konta.

Ćwiczenie na kartce: zaprojektuj dostęp

Sytuacja

Dobierz minimalne prawa do tabel UCZNIOWIE, OCENY i RAPORTY.

ROLA	ZADANIE	GRANT	CZEGO NIE NADAWAĆ
sekretariat	poprawia adres ucznia	.....	.....
nauczyciel	wpisuje i poprawia oceny	.....	.....
dyrektor	czyta raporty	.....	.....
aplikacja	wyświetla listę klas	.....	.....

Na końcu zapisz jedno polecenie REVOKE dla prawa nadanego omyłkowo.

Omówienie ćwiczenia

SEKRETARIAT	GRANT SELECT, UPDATE ON uczniowie. Bez DELETE i DROP.
NAUCZYCIEL	GRANT SELECT, INSERT, UPDATE ON oceny. Bez DELETE uczniów.
DYREKTOR	GRANT SELECT ON raporty. Bez praw zmieniających dane.
APLIKACJA	GRANT SELECT ON klasy. Bez ALTER i bez dostępu administracyjnego.

```
REVOKE DELETE ON uczniowie FROM nauczyciel;
```

Uzasadnienie powinno łączyć zadanie użytkownika z dokładnym zakresem dostępu.

Podsumowanie i bilet wyjścia

GRANT	Nadaje wskazane uprawnienie do obiektu bazy.
REVOKE	Odbiera wcześniej nadane uprawnienie.
MINIMUM	Rola otrzymuje tylko prawa potrzebne do zadania.
KONTROLA	Poprawne SQL może zostać odrzucone z powodu braku prawa.

BILET WYJŚCIA: napisz jedno polecenie GRANT dla roli raporty i wyjaśnij, dlaczego nie dodajesz UPDATE ani DELETE.