Správa a zabezpečení databáze (tabulky databáze MySQL, přidání uživatele databáze, práva, role, zamykání tabulek, transakce)

Správa a zabezpečení databáze

Databáze jako taková musí být řádně zabezpečena před přístupem neoprávněných osob. K tomu slouží (v databázi MySQL) práva uživatelů, kteří pak v databázi smějí dělat pouze to, co je jim povolené.

Tabulky databáze MySQL

Tabulka je jedním ze základních databázových objektů. Slouží k přímému uložení dat do prostoru relační databáze. Má pevně daný počet a význam jednotlivých sloupců, které určují typ a význam hodnot v takovém sloupci uložených. Není možné, aby dva různé záznamy v tabulce měly odlišný počet položek (sloupců) nebo obsahovaly ve stejné položce dva různé datové typy.

Každý jednotlivý řádek tabulky pak značí jeden konkrétní záznam v databázi. Počet řádků bývá omezen jen technickými možnostmi použité databáze.

Přidání uživatele databáze

Přidání uživatele do databáze MySQL je možné prostřednictvím příkazu CREATE USER. Takový příkaz může pak vypadat například následovně:

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
```

V tento moment nemá nový uživatel žádná oprávnění cokoliv s databázemi dělat. Je tedy nutné mu oprávnění podle potřeby přidělit.

Práva

Práva umožňují upravovat množinu činností, které mohou databázoví uživatelé vykonávat. Pokud chceme práva uživateli přidělit, poslouží příkaz `GRANT`, pokud je chceme uživateli odebrat, pak poslouží příkaz `REVOKE`. Po provedení změn provedeme reload práv příkazem `FLUSH PRIVILEGES;`.

Přidání povolení na SELECT a INSERT pro všechny tabulky v databázi website.
```
GRANT SELECT, INSERT ON website.* TO 'username'@'localhost';
```

Přidání všech oprávnění pro všechny databáze a jejich tabulky.
```
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost';
```

Odebrání oprávnění na DROP pro všechny databáze a jejich tabulky.
```
REVOKE DROP ON *.* TO 'username'@'localhost';
```

Přidání oprávnění měnit oprávnění uživatelů (může přidělovat prává, která sám má).
```
GRANT USAGE ON *.* 'username'@'localhost' WITH GRANT OPTION;
```

Role

Aby nebylo nutné přidělovat oprávnění každému uživateli zvlášť, existují role. Jde o jakési skupiny oprávnění a tuto roli pak je možné snadno přidělit uživatelům, kteří mají disponovat oprávněními takové role. Jako příklad lze uvést role *admin*, která by měla přidělena všechna oprávnění. Nyní je možné uživatelům, kteří mají mít všechna oprávnění (mají být administrátory) přidělit roli *admin*.

Abychom nemuseli přidělovat oprávnění všem uživatelům zvlášť (což by bylo zdlouhavé), existují role. Role jsou takové "šablony na oprávnění". Například můžeme vytvořit roli admin, které povolíme vše. Následně uživatelům, ze kterých chceme udělat administrátory přidělíme tuto roli a ti automaticky získají tato práva. MySQL role nepodporuje. Mnoho databázových systémů jako Oracle, Sybase nebo MS SQL ale ano.

Nastavování rolí může vypadat přibližně následovně.

```
CREATE ROLE 'app_developer', 'app_read', 'app_write';

GRANT ALL ON app_db.* TO 'app_developer';
GRANT SELECT ON app_db.* TO 'app_read';
GRANT INSERT, UPDATE, DELETE ON app_db.* TO 'app_write';

CREATE USER 'dev1'@'localhost' IDENTIFIED BY 'dev1pass';
CREATE USER 'read_user1'@'localhost' IDENTIFIED BY 'read_user1pass';
CREATE USER 'read_user2'@'localhost' IDENTIFIED BY 'read_user2pass';
CREATE USER 'rw_user1'@'localhost' IDENTIFIED BY 'rw_user1pass';

GRANT 'app_developer' TO 'dev1'@'localhost';
GRANT 'app_read' TO 'read_user1'@'localhost', 'read_user2'@'localhost';
GRANT 'app_read', 'app_write' TO 'rw_user1'@'localhost';
```

Zamykání tabulek

Zamčení tabulky je součástí transakcí (viz níže). Zatímco probíhá nějaká transakce, je znemožněno používání tabulky, která byla uzamčena právě pro provedení transakce. Je tím zajištěno, že se data v průběhu provádění transakce nečekaně nezmění.

Transakce

Transakce je uspořádaná skupina databázových operací, která se vnímá a provádí jako jediná jednotka a provádí se buďto celá, nebo vůbec. Nikdy nesmí nastat případ, kdy se vykoná jen její část. Jako příklad lze uvést například převod peněz z jednoho účtu na druhý. Vždy se musí peníze z jednoho účtu odečíst a na druhý přičíst. K provádění transakcí jsou používány tři příkazy.

`START TRANSACTION`	zahájí transakci – veškeré následující příkazy jsou její součástí a navenek se budou jevit jako jediný příkaz
`COMMIT`	aktuální transakce je potvrzena – změny jsou zapsány do databáze a jsou uvolněny systémové prostředky, které si transakce žádala
`ROLLBACK`	aktuální transakce je zamítnuta – všechny provedené změny jsou zrušeny a databáze se vrátí do stavu, v němž byla před zahájením transakce