

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт до практичної роботи №12
з дисципліни “ ОБДЗ ”
На тему: «Розробка »

Виконав:
ст. гр. КН-211
Качмарик Віктор
Викладач:
Якимишин Х. М.

Мета роботи: Навчитися використовувати механізм транзакцій у СУБД MySQL. Розробити SQL запити, які виконуються як єдине ціле в рамках однієї транзакції.

Короткі теоретичні відомості

Тригер – це спеціальний вид користувацької процедури, який виконується автоматично при певних діях над таблицею, наприклад, при додаванні чи оновленні даних. Кожен тригер асоційований з конкретною таблицею і подією. Найчастіше тригери використовуються для перевірки коректності вводу нових даних та підтримки складних обмежень цілісності. Крім цього їх використовують для автоматичного обчислення значень полів таблиць, організації перевірок для захисту даних, збирання статистики доступу до таблиць баз даних чи реєстрації інших подій.

Для створення тригерів використовують директиву CREATE TRIGGER.

Синтаксис:

```
CREATE [DEFINER = { користувач | CURRENT_USER }]
TRIGGER ім'я_тригера час_виконання подія_виконання
ON назва_таблиці FOR EACH ROW тіло_тригера
```

Аргументи:

DEFINER

Задає автора процедури чи функції. За замовчуванням – це CURRENT_USER.

ім'я_тригера

Ім'я тригера повинно бути унікальним в межах однієї бази даних.

час_виконання

Час виконання тригера відносно події виконання. BEFORE – виконати тіло тригера до виконання події, AFTER – виконати тіло тригера після події.

подія_виконання

Можлива подія – це внесення (INSERT), оновлення (UPDATE), або видалення (DELETE) рядка з таблиці. Один тригер може бути пов'язаний лише з однією подією. Команда AFTER INSERT, AFTER UPDATE, AFTER DELETE визначає виконання тіла тригера відповідно після внесення, оновлення, або видалення даних з таблиці. Команда BEFORE INSERT, BEFORE UPDATE, BEFORE DELETE визначає виконання тіла тригера відповідно до внесення, оновлення, або видалення даних з таблиці.

ON назва_таблиці

Таблиця, або віртуальна таблиця (VIEW), для якої створюється даний тригер. При видаленні таблиці з бази даних, автоматично видаляються всі пов'язані з нею тригери.

FOR EACH ROW тіло_тригера

Задає набір SQL директив, які виконує тригер. Тригер викликається і виконується для кожного зміненого рядка. Директиви можуть об'єднуватись командами BEGIN ... END та містити спеціальні команди OLD та NEW для доступу до попереднього та нового значення поля у зміненому рядку відповідно. В тілі тригера дозволено викликати збережені процедури, але заборонено використовувати транзакції, оскільки тіло тригера автоматично виконується як одна транзакція.

NEW.назва_поля

Повертає нове значення поля для зміненого рядка. Працює лише при подіях INSERT та UPDATE. У тригерах, які виконуються перед (BEFORE) подією можна змінити нове значення поля командою SET NEW.назва_поля = значення.

OLD.назва_поля

Повертає старе значення поля для зміненого рядка. Можна використовувати лише при подіях UPDATE та DELETE. Змінити старе значення поля не можливо.

Щоб видалити створений тригер з бази даних, потрібно виконати команду DROP TRIGGER назва_тригера.

Хід роботи:

Потрібно розробити тригери, які виконуватимуть наступні дії.

1. Каскадне оновлення таблиці `medicine` при видаленні препарату з таблиці `all_medicine`.
2. Шифрування паролю користувача під час внесення в таблицю.
3. Тригер для таблиці `customer_session`, який буде фіксувати у таблиці `customer` дату останнього входу користувача в систему.

1. Каскадне оновлення таблиці `medicine` при видаленні препарату з таблиці `all_medicine`. Діюче обмеження зовнішнього ключа при видаленні препарату встановлює для користувача невизначену роль (значення `NULL`). Натомість, за допомогою тригера, користувачеві потрібно присвоювати певну роль за замовчуванням (роль `Guests` з `roleID=4`).

```
CREATE TRIGGER category_update BEFORE DELETE
ON medicine_category
FOR EACH ROW UPDATE all_medicine
SET medicine_category_id = 1
WHERE medicine_category_id = OLD.medicine_category_id;
```

Перевіримо роботу тригера, видаливши роль з номером 2:

Так виглядає таблиця `medicine` перед видаленням:

medicine_name	medicine_category_id	medicine_description	contraindication	appliance
Adhesive plaster	3	NULL	NULL	NULL
AntiCorona	2	NULL	NULL	NULL
Bandages	3	NULL	NULL	NULL
Card	2	NULL	NULL	NULL
Mezym	1	NULL	NULL	NULL
Mixture	2	NULL	NULL	NULL
Pentalgin	1	NULL	NULL	NULL
Ramtex	1	NULL	NULL	NULL
Sorbex	2	NULL	NULL	NULL

Тепер застосуємо тригер:

```
mysql> DELETE FROM medicine_category WHERE medicine_category_id = 2;
Query OK, 1 row affected (0.01 sec)
```

Перевіримо, чи з'явилися зміни:

```
mysql> SELECT * FROM all_medicine;
```

medicine_name	medicine_category_id	medicine_description	contraindication	appliance
Adhesive plaster	3	NULL	NULL	NULL
AntiCorona	1	NULL	NULL	NULL
Bandages	3	NULL	NULL	NULL
Card	1	NULL	NULL	NULL
Mezym	1	NULL	NULL	NULL
Mixture	1	NULL	NULL	NULL
Pentalgin	1	NULL	NULL	NULL
Ramtex	1	NULL	NULL	NULL
Sorbex	1	NULL	NULL	NULL

2. Створимо тригер, для шифрування паролю користувача під час внесення в таблицю.

```
CREATE TRIGGER customer_password BEFORE
INSERT ON customer FOR EACH ROW
SET NEW.password = HEX(AES_ENCRYPT(NEW.password, 'test'));
```

Такий вигляд має таблиця users:

customer_id	first_name	last_name	phone	email	street	city	state	zip_code	login	password
1	Viktor	Kachmaryk	NULL	vitya.kachmaryk@gmail.com	Sykhivska	Lviv	NULL	NULL	user_vitya	viktor
2	Andrew	Ilkiv	NULL	andrew.ilkiiv@gmail.com	Lybinska	Lviv	NULL	NULL	andreeew	andrew
3	Vlad	Kondratskiy	NULL	vlad.kondratskiyv@gmail.com	Lybinska	Lviv	NULL	NULL	user05_vlad	vlad
4	Max	Kachmaryk	NULL	max.kachmaryk@gmail.com	Sykhivska	Lviv	NULL	NULL	max_kachmaryk	max
5	Olesya	Marko	NULL	olesya.marko@gmail.com	Lybinska	Lviv	NULL	NULL	olesia.marko	jlesya
6	Yuliana	Lavryk	NULL	yuliana.lavryk@gmail.com	Gorodotska	Lviv	NULL	NULL	userq_lavrino	yuliana
7	Roman	Yakymiv	NULL	roman.yakymiv@gmail.com	Gorodotska	Lviv	NULL	NULL	25user_roman	roman
8	Petro	Lagodyk	NULL	petro.lagodyk@gmail.com	Gorodotska	Lviv	NULL	NULL	kllloiak	petro
9	Sasha	Slyusarchik	NULL	sasha.slyusarchik@gmail.com	Gorodotska	Lviv	NULL	NULL	user001_sasha	sasha
10	Yuriy	Oschipko	NULL	yuriy.oschipko@gmail.com	Sykhivska	Lviv	NULL	NULL	superuser_yuriy	yuriy

Тепер вставимо нового користувача:

```
mysql> INSERT INTO customer VALUES (NULL, 'Petroror', 'Petro', NULL, 'pet@gmail.com', 'Zelena', 'Lviv', NULL, NULL, 'petros', 'mypass');
Query OK, 1 row affected (0.00 sec)
```

Перевіримо таблицю юзерів:

customer_id	first_name	last_name	phone	email	street	city	state	zip_code	login	password
1	Viktor	Kachmaryk	NULL	vitya.kachmaryk@gmail.com	Sykhivska	Lviv	NULL	NULL	user_vitya	viktor
2	Andrew	Ilkiv	NULL	andrew.ilkiiv@gmail.com	Lybinska	Lviv	NULL	NULL	andreeew	andrew
3	Vlad	Kondratskiy	NULL	vlad.kondratskiyv@gmail.com	Lybinska	Lviv	NULL	NULL	user05_vlad	vlad
4	Max	Kachmaryk	NULL	max.kachmaryk@gmail.com	Sykhivska	Lviv	NULL	NULL	max_kachmaryk	max
5	Olesya	Marko	NULL	olesya.marko@gmail.com	Lybinska	Lviv	NULL	NULL	olesia.marko	jlesya
6	Yuliana	Lavryk	NULL	yuliana.lavryk@gmail.com	Gorodotska	Lviv	NULL	NULL	userq_lavrino	yuliana
7	Roman	Yakymiv	NULL	roman.yakymiv@gmail.com	Gorodotska	Lviv	NULL	NULL	25user_roman	roman
8	Petro	Lagodyk	NULL	petro.lagodyk@gmail.com	Gorodotska	Lviv	NULL	NULL	kllloiak	petro
9	Sasha	Slyusarchik	NULL	sasha.slyusarchik@gmail.com	Gorodotska	Lviv	NULL	NULL	user001_sasha	sasha
10	Yuriy	Oschipko	NULL	yuriy.oschipko@gmail.com	Sykhivska	Lviv	NULL	NULL	superuser_yuriy	yuriy
16	Petroror	Petro	NULL	pet@gmail.com	Zelena	Lviv	NULL	NULL	petros	30F4A922DAA70D441CCB16129397315B

3. Тригер для таблиці customer_session, який буде фіксувати у таблиці customer дату останнього входу користувача в систему.

```
CREATE TRIGGER customer_lastseen AFTER INSERT
ON customer_session
FOR EACH ROW UPDATE customer
SET customer.lastseen = DATE(NEW.lastseen)
WHERE customer.customer_id = NEW.customer_id;
```

Перевіримо таблицю користувачів:

customer_id	first_name	last_name	login	lastseen
1	Viktor	Kachmaryk	user_vitya	NULL
2	Andrew	Ilkiv	andreeew	NULL
3	Vlad	Kondratskiy	user05_vlad	NULL
4	Max	Kachmaryk	max_kachmaryk	NULL
5	Olesya	Marko	olesia.marko	NULL
6	Yuliana	Lavryk	userq_lavrino	NULL
7	Roman	Yakymiv	25user_roman	NULL
8	Petro	Lagodyk	klloiak	NULL
9	Sasha	Slyusarchik	user001_sasha	NULL
10	Yuriy	Oschipko	superuser_yuriy	NULL
16	Petroror	Petro	petros	NULL

Вставимо в таблицю customer_session нові значення:

```
INSERT INTO customer_session VALUES (NULL, 4, '20200304');
INSERT INTO customer_session VALUES (NULL, 3, '20200101');
```

Тепер подивимось на таблицю користувачів:

customer_id	first_name	last_name	login	lastseen
1	Viktor	Kachmaryk	user_vitya	NULL
2	Andrew	Ilkiv	andreeew	NULL
3	Vlad	Kondratskiy	user05_vlad	2020-01-01
4	Max	Kachmaryk	max_kachmaryk	2020-03-04
5	Olesya	Marko	olesia.marko	NULL
6	Yuliana	Lavryk	userq_lavrino	NULL
7	Roman	Yakymiv	25user_roman	NULL
8	Petro	Lagodyk	klloiak	NULL
9	Sasha	Slyusarchik	user001_sasha	NULL
10	Yuriy	Oschipko	superuser_yuriy	NULL

Висновок: на цій лабораторній роботі було розглянуто тригери, їх призначення, створення та використання. Було розроблено тригери для таблиць `customer_session`, `medicine` та `user`.