МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Лабораторна робота №12

З дисципліни «Організація баз даних та знань»

Виконав:
студент групи КН-210
Дойков Вадим

Викладач:
Мельникова Н. І.

Львів – 2020

Тема: Розробка та застосування тригерів

Мета: Розробити SQL запити, які моделюють роботу тригерів: каскадне знищення, зміна та доповнення записів у зв’язаних таблицях.

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. КомандаAFTER INSERT,AFTER UPDATE,AFTER DELETE визначаєвиконаннятіла 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 видалені 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

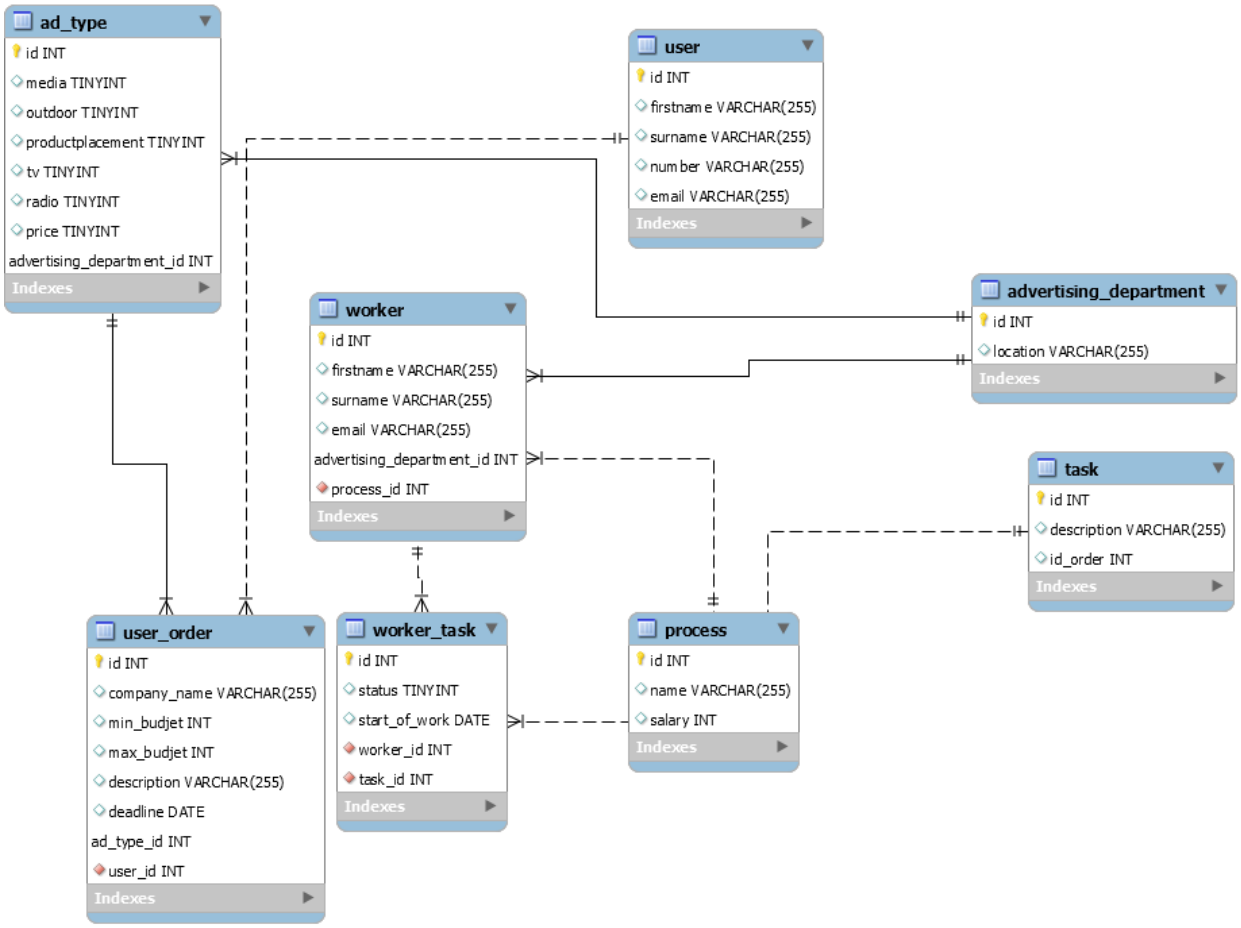


Рис 1. ER-діаграма

Потрібно розробити тригери які будуть виконувати наступні дії:

1. Каскадне оновлення таблиці користувачів при видаленні типу реклами з таблиці user_order.
2. Шифрування паролю користувача під час внесення в таблицю.
3. Тригер для таблиці Session, який буде фіксувати у таблиці Author дату останнього входу користувача в систему.

1. Каскадне оновлення таблиці замовлень при видаленні ролі з таблиці ad_type. Діюче обмеження зовнішнього ключа при видалені типу реклами встановлює для замовлення невизначену роль (значення NULL). Натомість, за допомогою тригера, замовленню потрібно присвоювати певний тип реклами за замовчуванням (ad_type_id=4).

```
4 •    CREATE
5      TRIGGER ad_type_delete BEFORE DELETE
6      ON ad_type FOR EACH ROW
7      UPDATE user_order SET ad_type_id=4 WHERE ad_type_id=OLD.id;
8
```

Таблиця до спрацювання тригера:

id	company_name	min_budjet	max_budjet	description	deadline	ad_type_id	user_id
1	Ac Ipsum Phasellus LLC	84	26	amet ultricies sem magna nec quam. Curabitur v...	2020-07-01	3	3
2	Sollicitudin Adipiscing Corp.	96	17	lacinia mattis. Integer eu lacus. Quisque imperdi...	2019-07-09	1	9
3	A Sollicitudin Orci Industries	307	18	malesuada. Integer id magna et ipsum cursus v...	2020-09-11	2	4
4	Et Eros PC	148	21	vulputate, nisi sem semper erat, in consectetue...	2020-02-06	4	8
5	Mattis Company	35	12	Phasellus libero mauris, aliquam eu, accumsan s...	2020-06-27	6	6
6	Enim Mauris PC	179	18	Phasellus dolor elit, pellentesque a, facilisis non,...	2020-11-05	5	9
7	Ac Corp.	390	28	erat vel pede blandit congue. In scelerisque sce...	2019-06-20	7	7
8	Parturient Montes Nascetur Industries	421	25	dignissim tempor arcu. Vestibulum ut eros non e...	2021-02-23	9	10
9	Nullam LLC	492	21	fermentum metus. Aenean sed pede nec ante b...	2019-07-08	8	3
10	Dis Parturient Company	268	17	lacus. Nulla tincidunt, neque vitae semper egest...	2020-10-17	10	4
11	faceboo	123	200	order description "faceboo"	2021-07-01	3	16
12	gaagle	400	600	order description "gaagle"	2020-03-10	6	20
13	amazonka	20	100	order description "amazonka"	2020-12-07	1	25

Після спрацювання:

```
5 •  delete from ad_type where id=3;
6 •  select *
7    from user_order;
```

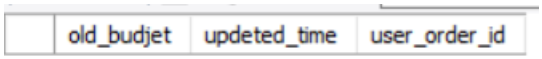
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	company_name	min_budjet	max_budjet	description	deadline	ad_type_id	user_id
1	Ac Ipsum Phasellus LLC	84	26	amet ultricies sem magna nec quam. Curabitur v...	2020-07-01	4	3
2	Sollicitudin Adipiscing Corp.	96	17	lacinia mattis. Integer eu lacus. Quisque imperdi...	2019-07-09	1	9
3	A Sollicitudin Orci Industries	307	18	malesuada. Integer id magna et ipsum cursus v...	2020-09-11	2	4
4	Et Eros PC	148	21	vulputate, nisi sem semper erat, in consectetue...	2020-02-06	4	8
5	Mattis Company	35	12	Phasellus libero mauris, aliquam eu, accumsan s...	2020-06-27	6	6
6	Enim Mauris PC	179	18	Phasellus dolor elit, pellentesque a, facilisis non,...	2020-11-05	5	9
7	Ac Corp.	390	28	erat vel pede blandit congue. In scelerisque sce...	2019-06-20	7	7
8	Parturient Montes Nascetur Industries	421	25	dignissim tempor arcu. Vestibulum ut eros non e...	2021-02-23	9	10
9	Nullam LLC	492	21	fermentum metus. Aenean sed pede nec ante b...	2019-07-08	8	3
10	Dis Parturient Company	268	17	lacus. Nulla tincidunt, neque vitae semper egest...	2020-10-17	10	4
11	faceboo	123	200	order description "faceboo"	2021-07-01	4	16
12	gaagle	400	600	order description "gaagle"	2020-03-10	6	20
13	amazonka	20	100	order description "amazonka"	2020-12-07	1	25

2.При оновленні даних бюджету замовлення, старі дані будуть переміщуватись у нову таблицю.

```
3 •   CREATE
4     TRIGGER checkUpdate BEFORE
5     update ON user_order
6     FOR EACH ROW
7     insert into updated_user_order(old_budjet, updeted_time, user_order_id)
8         values (old.min_budjet, now(), old.id)
```

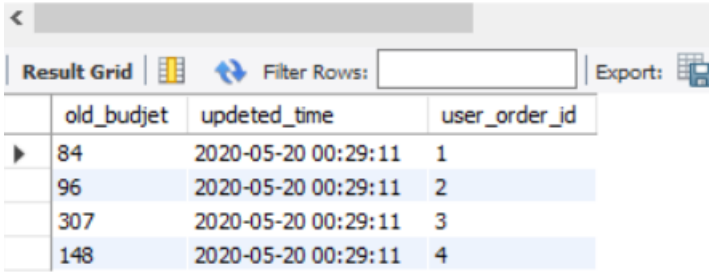
Таблиця до спрацювання тригера:



	old_budjet	updeted_time	user_order_id

Після спрацювання:

```
3 •   update user_order set min_budjet = 123
4     where id  between 1 and 4;
5
6 •   select *
7     from updated_user_order;
```



Result Grid | Filter Rows: | Export:

	old_budjet	updeted_time	user_order_id
▶	84	2020-05-20 00:29:11	1
	96	2020-05-20 00:29:11	2
	307	2020-05-20 00:29:11	3
	148	2020-05-20 00:29:11	4

3. При вставці нового працівника працівнику, додавати в таблицю worker_task процес який він виконує та департамент, де працює.

```
create  trigger  after_insert2
after insert on worker
for each row
insert into worker_task( worker_id, task_id)
values(new.id, 5);
```

Таблиця до спрацювання тригера:

id	status	start_of_work	worker_id	task_id
1	1	2020-05-20	3	5
2	1	2020-07-19	5	4
3	0	2021-03-18	3	2
4	0	2020-05-26	2	3
5	1	2021-01-22	6	2
6	1	2021-01-18	5	3

Після спрацювання:

id	status	start_of_work	worker_id	task_id
1	1	2020-05-20	3	5
2	1	2020-07-19	5	4
3	0	2021-03-18	3	2
4	0	2020-05-26	2	3
5	1	2021-01-22	6	2
6	1	2021-01-18	5	3
7	NULL	NULL	29	5

Висновок: в даній лабораторній роботі я навчився розробляти та використовувати різні види тригерів.