

Лабораторна робота 6

ОСНОВИ ВИКОРИСТАННЯ ЗАСОБІВ КОНТРОЛЮ ЦІЛІСНОСТІ ДАНИХ

Мета роботи: вивчити основи роботи із засобами контролю посилкової цілісності даних на прикладі СУБД MySQL.

Хід роботи

Увага! Перш ніж перейти до виконання лабораторної роботи, необхідно створити тимчасову базу даних, використовуючи запити, використані у лабораторній роботі 2. В усіх подальших пунктах даної лабораторної роботи передбачається використання тимчасової бази даних.

1. Вивчити особливості роботи механізму посилкової цілісності NO ACTION

Особливості роботи механізму посилкової цілісності NO ACTION розглянемо на прикладі відношень між таблицями `supplier` та `contract`, `supplier` та `supplier_person`, `supplier` та `supplier_org`. Дані таблиці пов'язані між собою за полем `supplier_id`. У цьому зв'язку таблиця `supplier` є батьківською, а таблиці `contract`, `supplier_org`, `supplier_person` – дочірніми. Для вивчення особливостей роботи механізму посилкової цілісності необхідно виконати наступну послідовність дій.

Встановити параметри ON DELETE та ON UPDATE, що визначають поведінку під час видалення та оновлення записів з таблиці-предка.

```
ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE supplier_org
DROP FOREIGN KEY supplier_org_ibfk_1;

ALTER TABLE supplier_org
ADD CONSTRAINT supplier_org_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE supplier_person
DROP FOREIGN KEY supplier_person_ibfk_1;

ALTER TABLE supplier_person
ADD CONSTRAINT supplier_person_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

Припустимо, що в силу певних причин необхідно видалили постачальника з кодом 4 (рисунок 6.1).

```
DELETE FROM supplier WHERE supplier_id = 4;
```

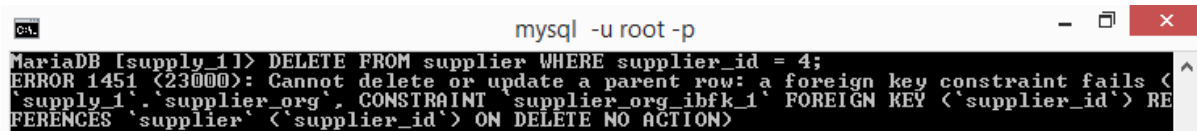


Рисунок 6.1

Таким чином, для того, щоб видалити даного постачальника, необхідно попередньо видалити усі пов'язані з ним дані. Для цього потрібно видалити відповідний запис з таблиці `supplier_org` та перевірити наявність договорів з даним постачальником у таблиці `contract`. Якщо такі договори є, їх також потрібно видалити (при цьому необхідно мати на увазі, що може виникнути потреба видалення й вмісту даних договорів). Після цього необхідно спробувати видалити постачальника з кодом 4 знову. Якщо зв'язаних з ним даних немає, постачальник буде видалений.

Припустимо, що в силу певних причин виникла необхідність для постачальника з кодом 5 змінити код на 7 (рисунок 6.2).

```
UPDATE supplier SET supplier_id = 7 WHERE supplier_id = 5;
```

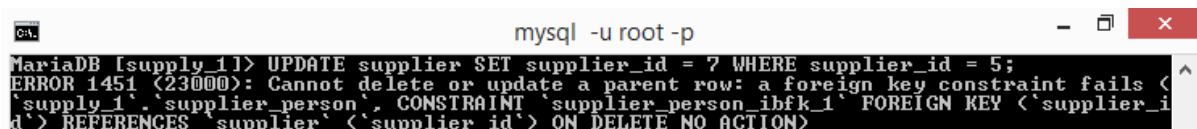


Рисунок 6.2

Оскільки договори з даним постачальником відсутні, посилання на нього є лише в таблиці `supplier_person`. Видаливши цей запис, необхідно повторити зміну коду постачальника з 5 на 7. Тепер ця операція повинна пройти успішно. Після цього необхідно перевірити вміст таблиць.

2. Вивчити особливості роботи механізму посилкової цілісності CASCADE

Змінимо механізми посилкової цілісності для зв'язків між усіма розглянутими вище таблицями на CASCADE.

```
ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE supplier_org
DROP FOREIGN KEY supplier_org_ibfk_1;

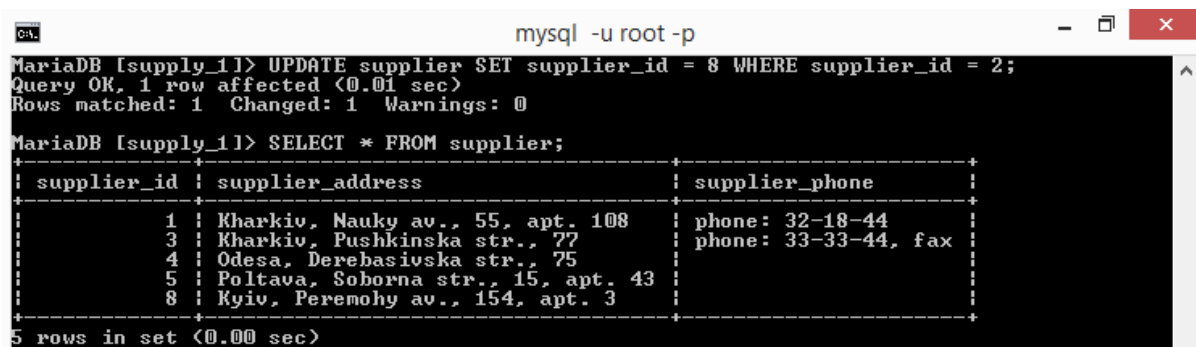
ALTER TABLE supplier_org
ADD CONSTRAINT supplier_org_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE supplier_person
DROP FOREIGN KEY supplier_person_ibfk_1;

ALTER TABLE supplier_person
ADD CONSTRAINT supplier_person_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;
```

Припустимо, що в силу певних причин виникла необхідність для постачальника з кодом 2 змінити код на 8 (рисунок 6.3).

```
UPDATE supplier SET supplier_id = 8 WHERE supplier_id = 2;
```



```
mysql -u root -p
MariaDB [supply_1]> UPDATE supplier SET supplier_id = 8 WHERE supplier_id = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [supply_1]> SELECT * FROM supplier;
```

supplier_id	supplier_address	supplier_phone
1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44
3	Kharkiv, Pushkinska str., 77	phone: 33-33-44, fax
4	Odesa, Derebasivska str., 75	
5	Poltava, Soborna str., 15, apt. 43	
8	Kyiv, Peremohy av., 154, apt. 3	

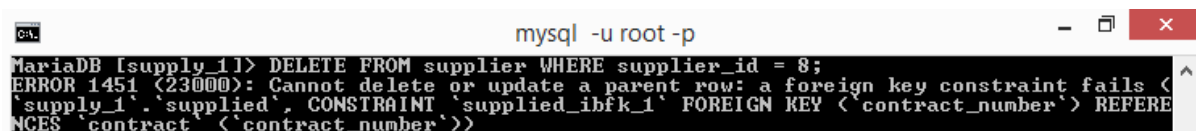
```
5 rows in set (0.00 sec)
```

Рисунок 6.3

Перевірити наявність відповідних змін у таблиці supplier_org.

Тепер припустимо, що даного постачальника (який зараз має код 8) треба видалити (рисунок 6.4).

```
DELETE FROM supplier WHERE supplier_id = 8;
```



```
mysql -u root -p
MariaDB [supply_1]> DELETE FROM supplier WHERE supplier_id = 8;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (<supply_1>.`supplied`, CONSTRAINT `supplied_ibfk_1` FOREIGN KEY (<contract_number>) REFERENCES <contract> (<contract_number>))
```

Рисунок 6.4

Визначити причину, через яку записи не були видалені. Внести необхідні зміни у механізми посилкової цілісності необхідних таблиць для того, щоб необхідні дані все ж були видалені.

3. Вивчити особливості роботи механізму посилкової цілісності SET NULL

Особливості механізму посилкової цілісності SET NULL розглянемо на прикладі таблиць supplier та contract.

Змінимо механізми посилкової цілісності для зв'язків між усіма розглянутими вище таблицями на SET NULL.

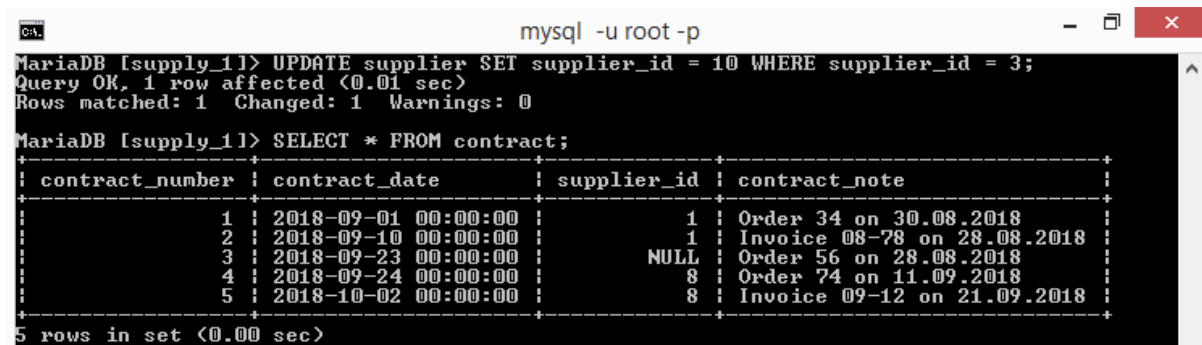
```
ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
MODIFY supplier_id INT NULL;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE SET NULL ON UPDATE SET NULL;
```

В таблиці supplier змінити код постачальника 3 на 10. Перевірити дані в таблиці contract (рисунок 6.5).

```
UPDATE supplier SET supplier_id = 10 WHERE supplier_id = 3;
```



The screenshot shows a MySQL terminal window with the following content:

```
mysql -u root -p
MariaDB [supply_1]> UPDATE supplier SET supplier_id = 10 WHERE supplier_id = 3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [supply_1]> SELECT * FROM contract;
```

contract_number	contract_date	supplier_id	contract_note
1	2018-09-01 00:00:00	1	Order 34 on 30.08.2018
2	2018-09-10 00:00:00	1	Invoice 08-78 on 28.08.2018
3	2018-09-23 00:00:00	NULL	Order 56 on 28.08.2018
4	2018-09-24 00:00:00	8	Order 74 on 11.09.2018
5	2018-10-02 00:00:00	8	Invoice 09-12 on 21.09.2018

5 rows in set (0.00 sec)

Рисунок 6.5

Замість NULL встановити значення коду постачальника 10 для договору з номером 3.

4. Оформити звіт з лабораторної роботи

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

5. Питання для самоконтролю

1. Чи є конструкції ON DELETE та ON UPDATE обов'язковими при формуванні команди CREATE TABLE або ALTER TABLE?
2. Яку поведінку СУБД задає конструкція ON DELETE?
3. Яку поведінку СУБД задає конструкція ON UPDATE?
4. Які параметри можна встановити після конструкцій ON DELETE та ON UPDATE?
5. Назвати особливості механізму посилкової цілісності CASCADE.
6. Назвати особливості механізму посилкової цілісності SET NULL.
7. Назвати особливості механізму посилкової цілісності NO ACTION.
8. Назвати особливості механізму посилкової цілісності SET DEFAULT.
9. Назвати особливості механізму посилкової цілісності RESTRICT.
10. Чому у даній лабораторній роботі не було розглянуто роботу з механізмом посилкової цілісності SET DEFAULT?
11. Яким чином можна встановити той чи інший механізм посилкової цілісності для зовнішнього ключа таблиці?
12. Навіщо перед тим, як встановити механізм SET NULL, була виконана модифікація поля supplier_id таблиці contract?
13. В яких випадках не рекомендується використання механізму посилкової цілісності CASCADE?
14. Який механізм посилкової цілісності завжди використовується за замовченням в СУБД MySQL у випадку, якщо конструкції ON DELETE та ON UPDATE не були визначені?