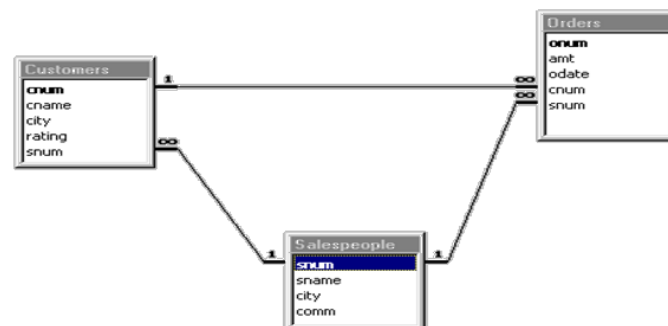


Бази даних

Частина 8

Проектування і перевірка тригерів

Trigger



Навчальний курс
Валько Н.В.

SQL

Мартин Грабер

МАРТИН ГРУБЕР

Понимание SQL

Перевод Лебедева В.Н.

Под редакцией Булычева В.Н.

МОСКВА, 1993

Джеймс Р. Грофф
Пол Н. Вайнберг

SQL

Полное руководство

Второе издание,
переработанное и дополненное

Перевод с английского
под редакцией В.Р. Гинзбурга

Линн Бейли

Изучаем SQL

Приведи
в порядок
свои
отношения
с данными



Прекрати путать
термины



Освой концепцию
и синтаксис SQL
максимально
эффективно

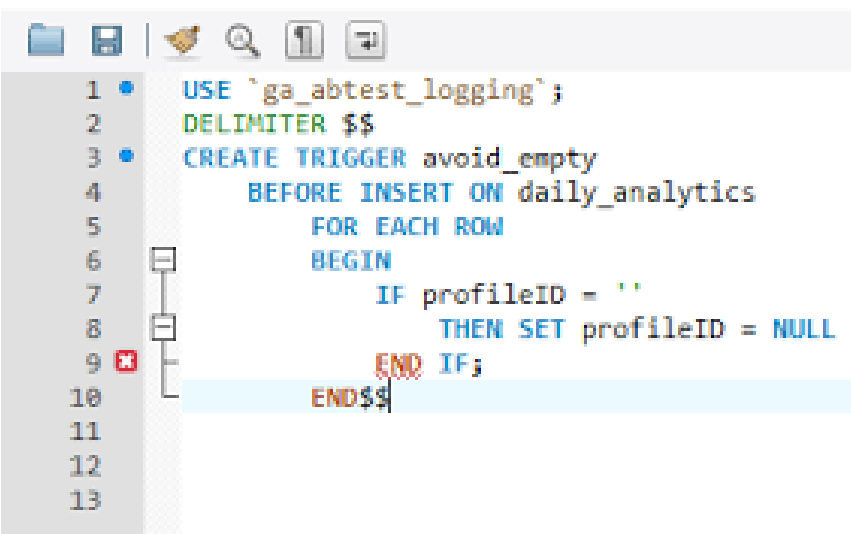


Перестань
смушаться



Зміст

- Визначення і синтаксис
- Приклади



The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, search, and execution. Below the toolbar, a list of lines (1-13) is visible on the left. The main area displays a SQL script for creating a trigger. The script is as follows:

```
1 • USE `ga_abtest_logging`;  
2 DELIMITER $$  
3 • CREATE TRIGGER avoid_empty  
4   BEFORE INSERT ON daily_analytics  
5     FOR EACH ROW  
6     BEGIN  
7       IF profileID = ''  
8         THEN SET profileID = NULL  
9       END IF;  
10  END$$  
11  
12  
13
```



Тригери

- Створюються для таблиць
- Активуються коли відбувається якась подія

Наприклад

- Контроль введення даних
- Контроль оновлення, видалення даних

СИНТАКСИС

CREATE

TRIGGER **trigger_name** **trigger_time** **trigger_event**

ON tbl_name

FOR EACH ROW

trigger_body

{ BEFORE | AFTER }

{ INSERT | UPDATE | DELETE }



Стани тригерів

1. INSERT (BEFORE | AFTER)
2. UPDATE (BEFORE | AFTER)
3. DELETE (BEFORE | AFTER)

Trigger_event

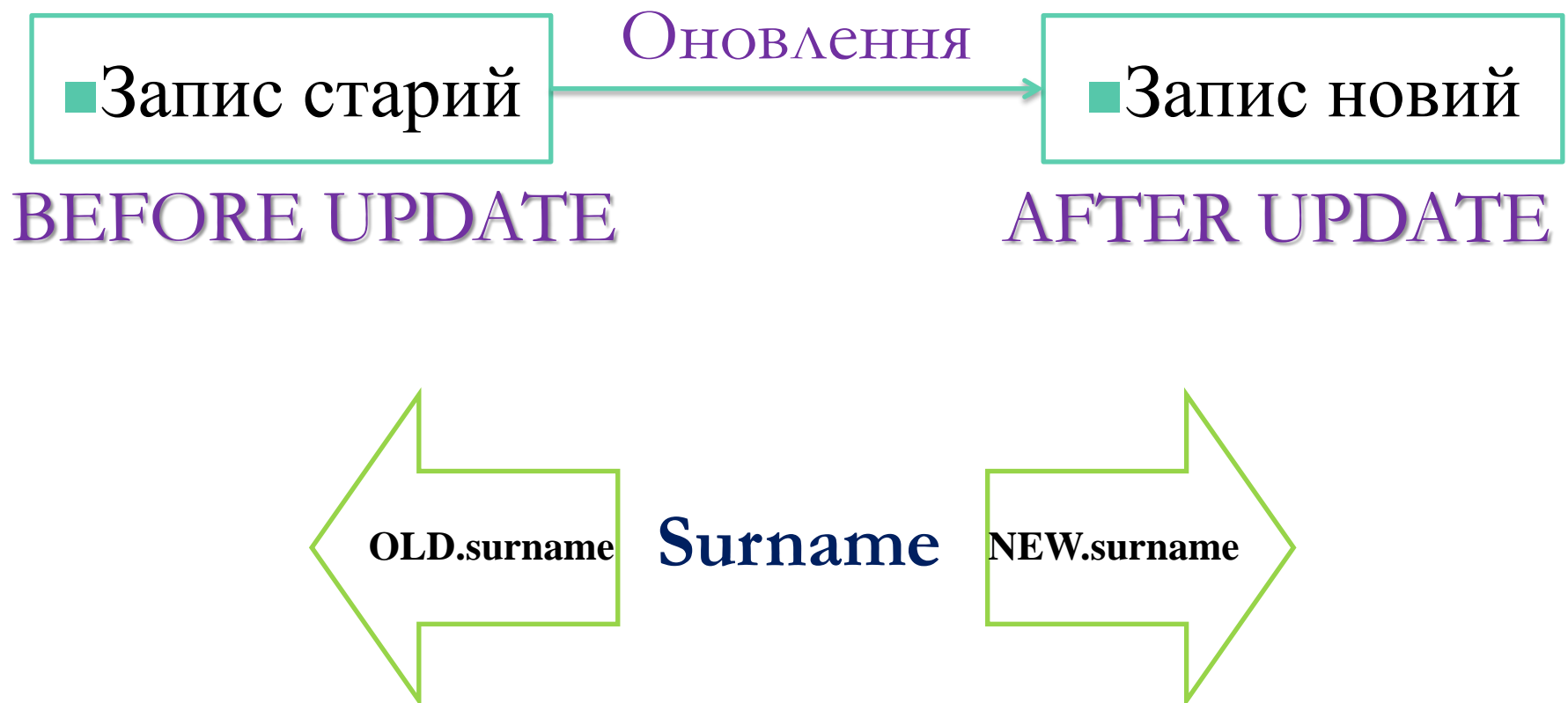
- **INSERT**: активується кожного разу, коли в таблицю вставляється новий рядок; наприклад, через INSERT, LOAD DATA, і REPLACE ;
 - **UPDATE**: активується кожного разу, коли змінюється рядок;
 - **DELETE**: активується кожного разу, коли рядок видаляється з таблиці; наприклад, через DELETE і REPLACE висловлювання.
- DROP TABLE і TRUNCATE TABLE оператори таблиці НЕ активують цей тригер.

Параметри доступу до даних

- **NEW** — для доступу до нових записів
OLD — для доступу до старих записів
- Наприклад, якщо оновили прізвище, то нове значення доступне через **NEW.surname**, а старе **OLD.surname**



Параметри доступу до даних



Приклади

- Створення «журналу» змін в таблиці та аудит

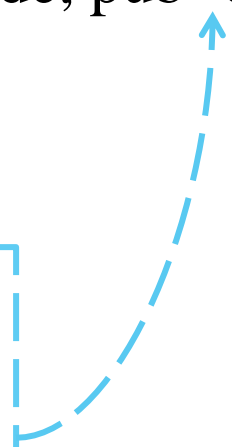


Приклад 1

Фіксація старих записів таблиці

```
CREATE TRIGGER abc BEFORE UPDATE  
ON book FOR EACH ROW  
INSERT INTO book2 SET nu=old.nu, name=old.name,  
neww=old.neww, code=old.code, pub=old.pub
```

Перед оновленням рядка в book –
‘старий’ запис копіюється в book2 –
формується так званий «журнал змін»
зі старих записів

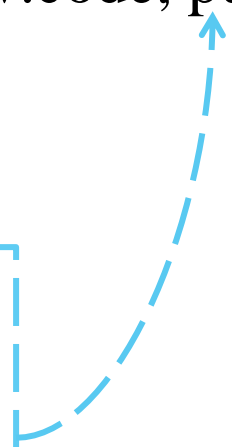


Приклад 2

Фіксація нових записів таблиці

```
CREATE TRIGGER abcd AFTER UPDATE  
ON book FOR EACH ROW  
INSERT INTO book2 SET nu=new.nu, name=new.name,  
neww=new.neww, code=new.code, pub=new.pub
```

Після оновлення рядка в book –
‘новий’ запис копіюється в book2 –
формується так званий «журнал змін»
з нових записів



Приклад 3

Фіксація видалених
з таблиці записів

```
CREATE TRIGGER abc BEFORE DELETE  
ON book FOR EACH ROW  
INSERT INTO book2 SET nu=old.nu, name=old.name,  
neww=old.neww, code=old.code, pub=old.pub
```

АБО

Якщо ми видалимо рядок з book він скопіюється в book2

```
CREATE TRIGGER abc BEFORE DELETE  
ON book FOR EACH ROW  
INSERT INTO book2 VALUES (old.nu, old.name, old.neww,  
old.code, old.pub)
```

Потрібно вказати ВСІ поля таблиці

Приклад 4

Фіксація обраних товарів

```
CREATE TRIGGER cart_items AFTER INSERT
ON trigger_cart_items
FOR EACH ROW
BEGIN
INSERT INTO trigger_cart_log (cart_id, item_id)
VALUES (new.cart_id, new.item_id);
END
```

Приклад 5

Зміна ціни на 30%

```
CREATE TRIGGER update_cost AFTER UPDATE
ON trigger_items_cost FOR EACH ROW
UPDATE trigger_items
SET price = (new.cost * 1.3)
WHERE item_id = new.item_id;
```

Якщо ввести вартість \$ 50,
то ціна повинна бути \$ 65

Приклад

CREATE TRIGGER user_update

BEFORE UPDATE

ON users

FOR EACH ROW SET

NEW.full_name = CONCAT(NEW.first_name, ' ', NEW.last_name);

id	first_name	last_name	full_name
1	Codex	World	Codex World

Тригер додавання імен: робить конкатенацію Прізвища та Ім'я у окремий стовпець

Додає запис тільки якщо менше 10 рядків.
Якщо в таблиці більше 10 рядків – видає помилку.

DELIMITER \$\$

CREATE TRIGGER trig BEFORE INSERT ON `book`

FOR EACH ROW

IF (SELECT COUNT(*) FROM `book`) >10

THEN

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Кількість більша 10';

ELSEIF (select count(*) from `book`) <5

THEN

SET @err2='Кількість менша 5 ';

END IF;

\$\$

DELIMITER ;

Спосіб " повернути " помилку –
SIGNAL надає інформацію про
помилку обробнику



Видалення тригера

DROP TRIGGER IF EXISTS trig;

Питання

- Що буде записано в таблицю Book2, якщо після оновлення назви книги в Book дублювати old.name ? (Приклад 1-2)