

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра «Системи штучного інтелекту»



Звіт до лабораторної роботи №11
З дисципліни «Організація Баз Даних»

Виконав:
студент групи КН-208
Дерев'янний Андрій

Прийняла:
Мельникова Н.І

Львів-2020

Тема: Розробка та застосування транзакцій

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

Завдання:

Використати механізм транзакцій у СУБД MySQL. Розробити SQL запити, які виконуються як єдине ціле в рамках однієї транзакції

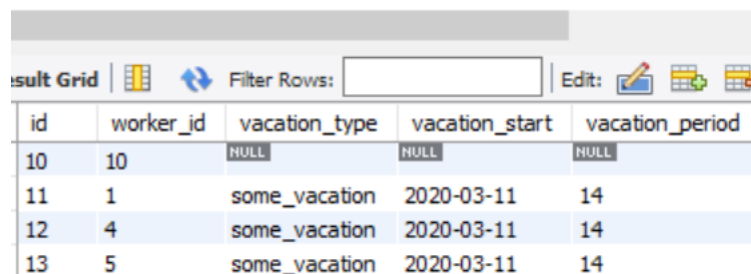
Виконання завдання:

1) Перевіримо роботу rollback. Для цього заповнимо таблицю vacation, використовуючи insert into:

```
1 • use db;
2
3 • start transaction;
4
5 • insert into db.vacation values (11, 1, 'some_vacation', '2020-03-11', 14);
6 • insert into db.vacation values (12, 4, 'some_vacation', '2020-03-11', 14);
7 • insert into db.vacation values (13, 5, 'some_vacation', '2020-03-11', 14);
```

Перевіримо заповнення таблиці:

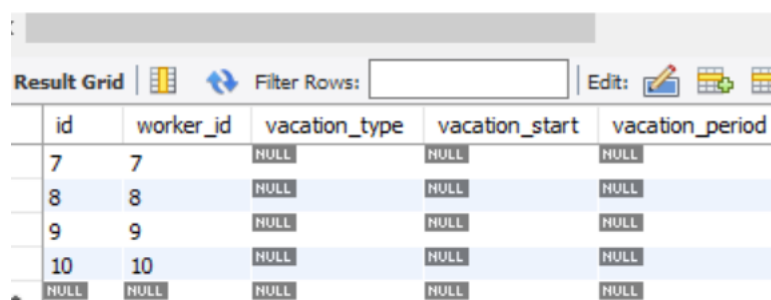
```
9 • select * from vacation;
```



id	worker_id	vacation_type	vacation_start	vacation_period
10	10	NULL	NULL	NULL
11	1	some_vacation	2020-03-11	14
12	4	some_vacation	2020-03-11	14
13	5	some_vacation	2020-03-11	14

Тепер зробимо rollback і порівняємо результати:

```
11 • rollback;
12
13 • select * from vacation;
```



id	worker_id	vacation_type	vacation_start	vacation_period
7	7	NULL	NULL	NULL
8	8	NULL	NULL	NULL
9	9	NULL	NULL	NULL
10	10	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

2) Тепер перевіримо роботу savepoint. Для цього знову заповнимо таблицю vacation, використовуючи insert into, але цього разу також додамо savepoint sv0 і sv1:

```
3 • start transaction;
4
5 • insert into db.vacation values (11, 1, 'some_vacation', '2020-03-11', 14);
6 • insert into db.vacation values (12, 4, 'some_vacation', '2020-03-11', 14);
7 • savepoint sv0;
8 • insert into db.vacation values (13, 5, 'some_vacation', '2020-03-11', 14);
9 • insert into db.vacation values (14, 2, 'some_vacation', '2020-03-11', 14);
10 • insert into db.vacation values (15, 7, 'some_vacation', '2020-03-11', 14);
11 • savepoint sv1;
12 • insert into db.vacation values (16, 10, 'some_vacation', '2020-03-11', 14);
```

Перевіримо заповнення таблиці:

```
14 • select * from vacation;
```

id	worker_id	vacation_type	vacation_start	vacation_period
13	5	some_vacation	2020-03-11	14
14	2	some_vacation	2020-03-11	14
15	7	some_vacation	2020-03-11	14
16	10	some_vacation	2020-03-11	14

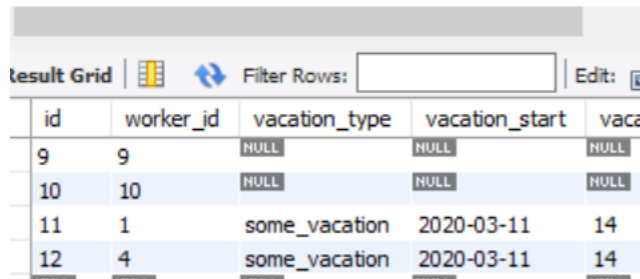
Тепер зробимо rollback до savepoint sv1 і перевіримо таблицю:

```
14 rollback to sv1;
15 • select * from vacation;
16
```

id	worker_id	vacation_type	vacation_start	vacation_period
12	4	some_vacation	2020-03-11	14
13	5	some_vacation	2020-03-11	14
14	2	some_vacation	2020-03-11	14
15	7	some_vacation	2020-03-11	14

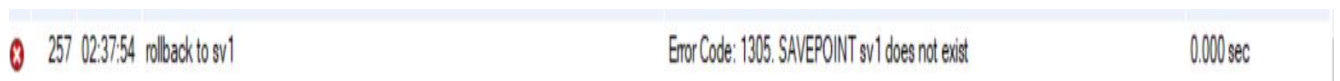
Повторимо дії, тільки цього робимо rollback до savepoint sv0:

```
14 rollback to sv0;  
15 • select * from vacation;
```



id	worker_id	vacation_type	vacation_start	vacation_end
9	9	NULL	NULL	NULL
10	10	NULL	NULL	NULL
11	1	some_vacation	2020-03-11	14
12	4	some_vacation	2020-03-11	14

Варто зазначити, що якщо з цього стану ще раз спробувати зробити rollback до savepoint sv1 – MySQL Workbench видасть нам помилку, що такого savepoint не існує:



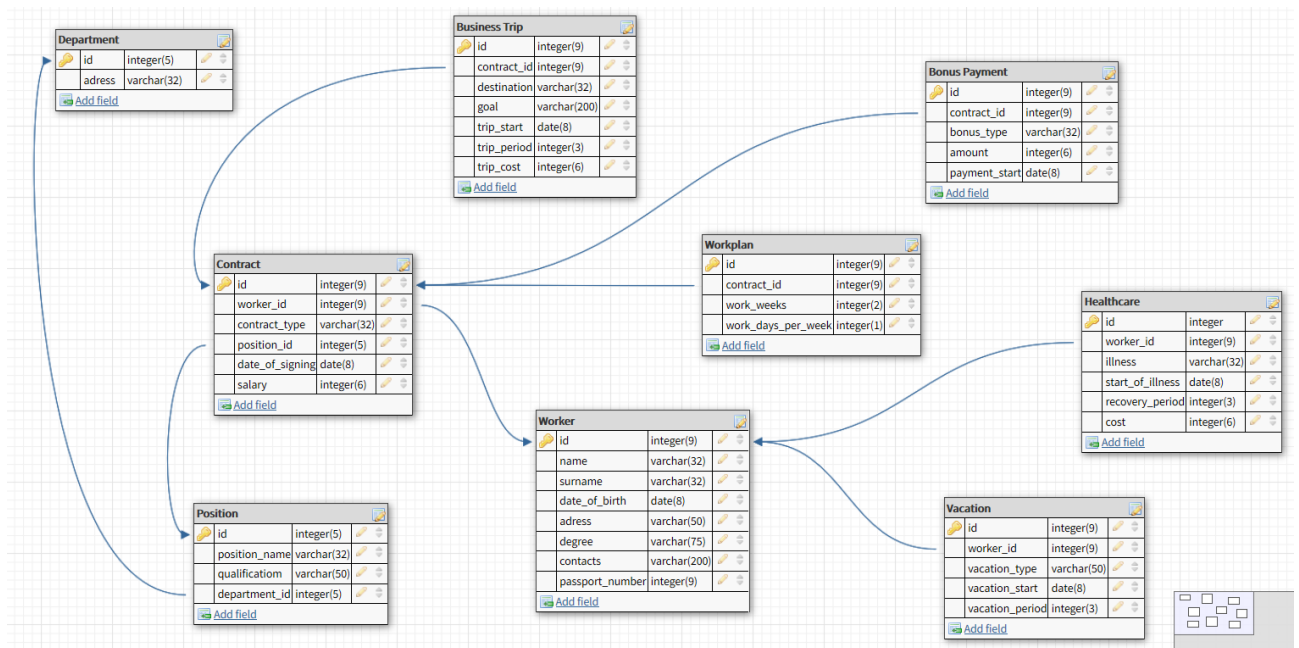
✖	257	02:37:54	rollback to sv1	Error Code: 1305. SAVEPOINT sv1 does not exist	0.000 sec
---	-----	----------	-----------------	--	-----------

Висновок:

За час виконання лабораторної роботи я навчився використовувати механізм транзакцій у СУБД MySQL, розробляти SQL запити, які виконуються як єдине ціле в рамках однієї транзакції

Додаток:

Діаграма бази даних(відділ кадрів підприємства):



SQL-скрипт бази даних:

```
create database db;
```

```
use db;
```

```
CREATE TABLE `Department` (
    `id` INT(5) NOT NULL AUTO_INCREMENT,
    `adress` varchar(32) NOT NULL,
    PRIMARY KEY (`id`)
);
```

```
CREATE TABLE `Worker` (
    `id` INT(9) NOT NULL AUTO_INCREMENT,
    `name` varchar(32) NOT NULL,
    `surname` varchar(32) NOT NULL,
    `date_of_birth` DATE NOT NULL,
    `adress` varchar(50) NOT NULL,
    `degree` varchar(75) NOT NULL,
    `contacts` varchar(200) NOT NULL,
    `passport_number` INT(9) NOT NULL,
    PRIMARY KEY (`id`)
```

);

```
CREATE TABLE `Vacation` (  
    `id` INT(9) NOT NULL AUTO_INCREMENT,  
    `worker_id` INT(9) NOT NULL,  
    `vacation_type` varchar(50),  
    `vacation_start` DATE,  
    `vacation_period` INT(3),  
    PRIMARY KEY (`id`),  
  
    CONSTRAINT fk0 FOREIGN KEY (worker_id)  
        REFERENCES Worker(id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `Position` (  
    `id` INT(5) NOT NULL AUTO_INCREMENT,  
    `position_name` varchar(32) NOT NULL,  
    `qualification` varchar(50) NOT NULL,  
    `department_id` INT(5) NOT NULL,  
    PRIMARY KEY (`id`),  
  
    CONSTRAINT fk1 FOREIGN KEY (department_id)  
        REFERENCES Department(id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `Contract` (  
    `id` INT(9) NOT NULL AUTO_INCREMENT,  
    `worker_id` INT(9) NOT NULL,  
    `contract_type` varchar(32) NOT NULL,  
    `position_id` INT(5) NOT NULL,  
    `date_of_signing` DATE NOT NULL,  
    `salary` INT(6) NOT NULL,  
    PRIMARY KEY (`id`),
```

```
CONSTRAINT fk2 FOREIGN KEY (worker_id)
REFERENCES Worker(id)
ON DELETE CASCADE ON UPDATE CASCADE,
```

```
CONSTRAINT fk3 FOREIGN KEY (position_id)
REFERENCES `Position`(id)
ON DELETE CASCADE ON UPDATE CASCADE
```

```
);
```

```
CREATE TABLE `Workplan` (
  `id` INT(9) NOT NULL AUTO_INCREMENT,
  `contract_id` INT(9) NOT NULL,
  `work_weeks` INT(2),
  `work_days_per_week` INT(1),
  PRIMARY KEY (`id`),
```

```
CONSTRAINT fk4 FOREIGN KEY (contract_id)
REFERENCES Contract(id)
ON DELETE CASCADE ON UPDATE CASCADE
```

```
);
```

```
CREATE TABLE `Bonus_Payment` (
  `id` INT(9) NOT NULL AUTO_INCREMENT,
  `contract_id` INT(9) NOT NULL,
  `bonus_type` varchar(32),
  `amount` INT(6),
  `payment_start` DATE,
  PRIMARY KEY (`id`),
```

```
CONSTRAINT fk5 FOREIGN KEY (contract_id)
REFERENCES Contract(id)
ON DELETE CASCADE ON UPDATE CASCADE
```

```
);
```

```
CREATE TABLE `Healthcare` (
  `id` INT NOT NULL AUTO_INCREMENT,
```



```
`worker_id` INT(9) NOT NULL,  
`illness` varchar(32),  
`start_of_illness` DATE,  
`recovery_period` INT(3),  
`cost` INT(6),  
PRIMARY KEY (`id`),
```

```
CONSTRAINT fk6 FOREIGN KEY (worker_id)  
REFERENCES Worker(id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `Business_Trip` (  
  `id` INT(9) NOT NULL AUTO_INCREMENT,  
  `contract_id` INT(9) NOT NULL,  
  `destination` varchar(32),  
  `goal` varchar(200),  
  `trip_start` DATE,  
  `trip_period` INT(3),  
  `trip_cost` INT(6),  
  PRIMARY KEY (`id`),  
  
  CONSTRAINT fk7 FOREIGN KEY (contract_id)  
  REFERENCES Contract(id)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```