МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА СИСТЕМОТЕХНІКИ

Звіт

З практичної роботи №7

На тему: «Створення транзакцій для високонавантажених баз даних на платформі СУБД MySQL»

з дисципліни «Проектування високонавантажених систем зберігання даних»

Виконав:

ст. гр. ІТКНу-19-2

Марковець Н.С.

Перевірив викладач:

Коваленко А.І.

Харків 2020

Мета: Набуття практичних навичок з розробки транзакцій, що використовуються в збережених процедурах і функціях, для забезпечення основних бізнес процесів високонавантаженої інформаційної системи. Формування необхідних практичних умінь для створення транзакцій, з урахуванням особливостей роботи високонавантаженої інформаційної системи зберігання даних.

Тема індивідуального завдання: ІС "Надання послуг типографії"

Скріншот схеми фізичної моделі бази даних з таблицями типу InnoDB у нотації IDEF1X представлено на рисунку 1.

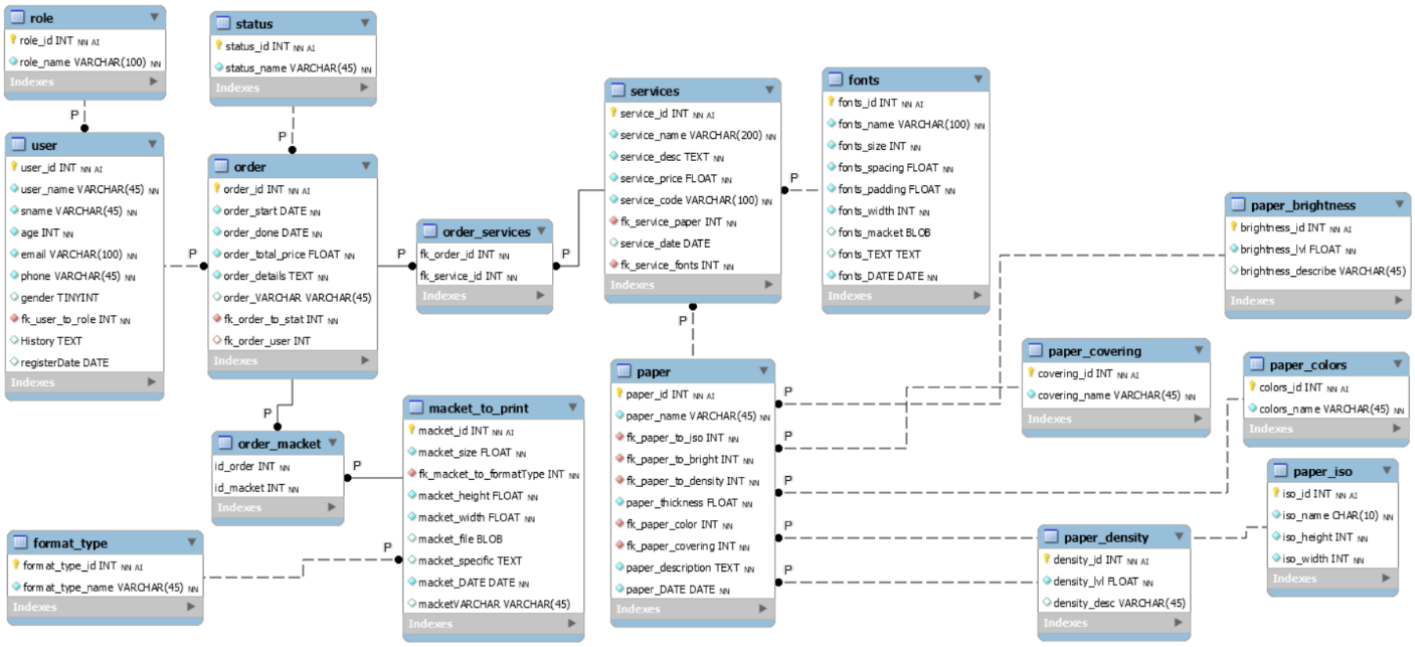

Рисунок 1. Схема фізичної моделі даних типу InnoDb у нотації IDEF1X

Завдання 7.1

– збережена процедура, що забезпечувати виконання SQL-запиту із блокуванням використовуваних таблиць MyISAM.

– скріншоти процесу тестування доступу до заблокованої базової таблиці, за допомогою двох іменованих підключень (потокових сесій) «name1» і «name2» до сервера MySQL (під час тестування в одному з іменованих підключень («name1») використовуються два блокування таблиці – з атрибутами READ і WRITE).

1. Блокування з атрибутом Read

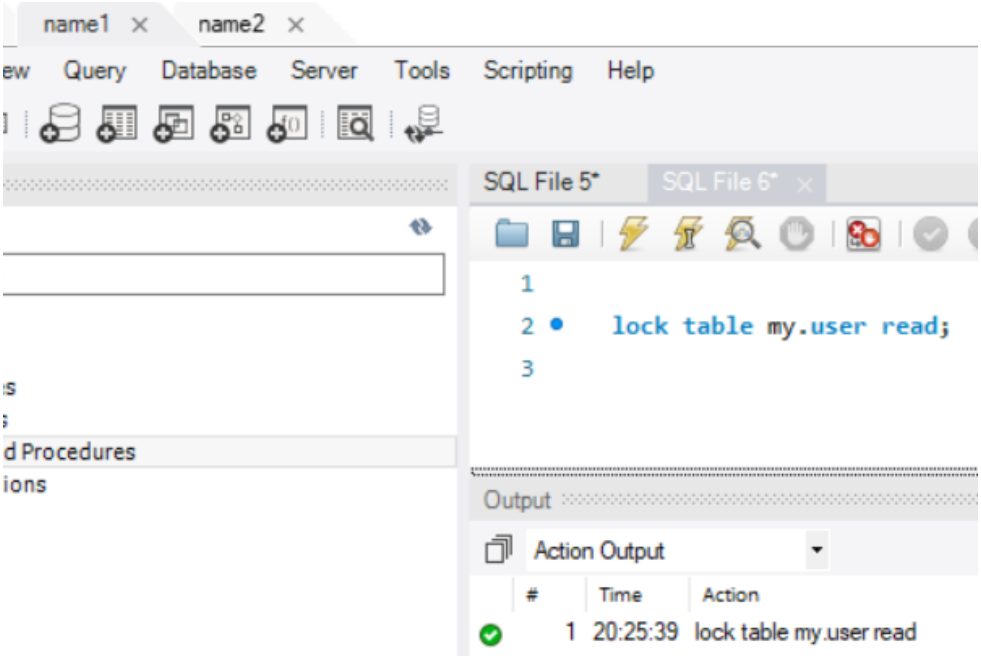

Рисунок 7.2 – Блокування таблиці для підключення name1

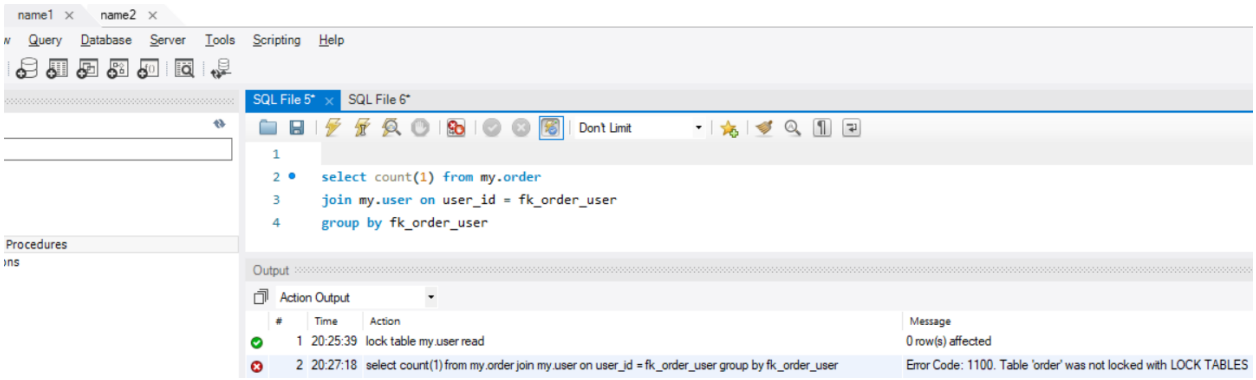

Рисунок 7.3 – Тестування блокування для підключення name1

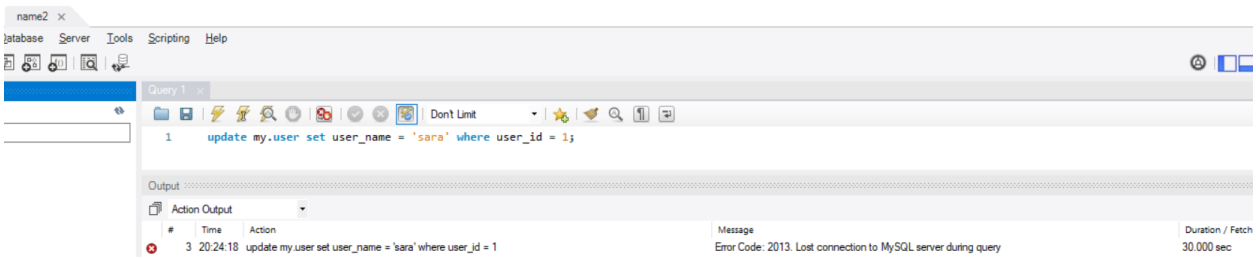

Рисунок 7.3 – Тестування блокування для підключення name2

```
2 •  unlock  tables;
3
```

Output

Action Output

#	Time	Action	Message
1	20:25:39	lock table my.user read	0 row(s) affected
2	20:27:18	select count(1) from my.order join my.user on user_id = fk_order_user group by fk_order_user	Error Code: 1100. Table 'order' was not locked with LOCK TABLES
3	20:29:24	unlock tables	0 row(s) affected

Рисунок 7.4 – Розблокування таблиці user

name1 × name2 ×

Query Database Server Tools Scripting Help

SQL File 5* × SQL File 6*

Don't Limit

```
1
2 •  select count(1) from my.order
3    join my.user on user_id = fk_order_user
4    group by fk_order_user
```

Procedures

ns

Output

Action Output

#	Time	Action	Message
1	20:25:39	lock table my.user read	0 row(s) affected
2	20:27:18	select count(1) from my.order join my.user on user_id = fk_order_user group by fk_order_user	Error Code: 1100. Table 'order' was not locked with LOCK TABLES
3	20:29:24	unlock tables	0 row(s) affected
4	20:29:44	select count(1) from my.order join my.user on user_id = fk_order_user group by fk_order_user	2555 row(s) returned

Рисунок 7.5 – Тестування розблокування при підключенні name1

name2 ×

Database Server Tools Scripting Help

Query 1

Don't Limit

```
1    update my.user set user_name = 'sara' where user_id = 1;
```

Output

Action Output

#	Time	Action	Message
3	20:24:18	update my.user set user_name = 'sara' where user_id = 1	Error Code: 2013. Lost connection to MySQL server during query
1	20:31:05	update my.user set user_name = 'sara' where user_id = 1	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0

Рисунок 7.6 – Тестування розблокування при підключенні name 2

Блокування з атрибутом Write

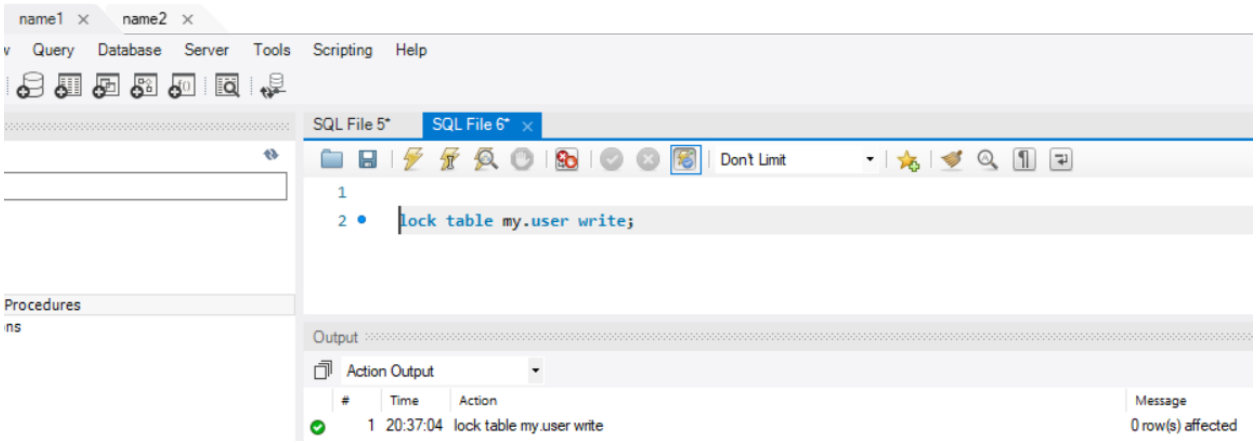

Рисунок 7.7 – Блокування таблиці з атрибутом write для підключення name1

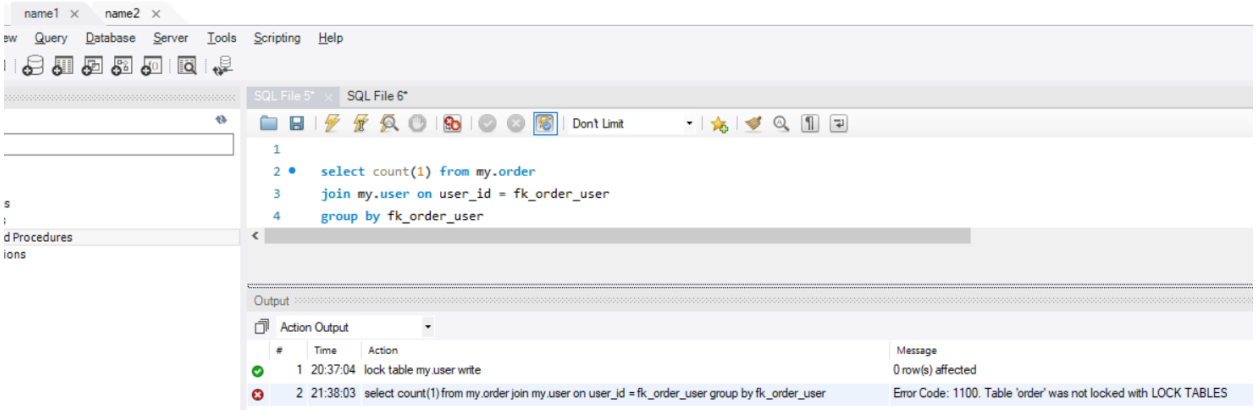

Рисунок 7.8 – Тестування розблокування при підключенні name1

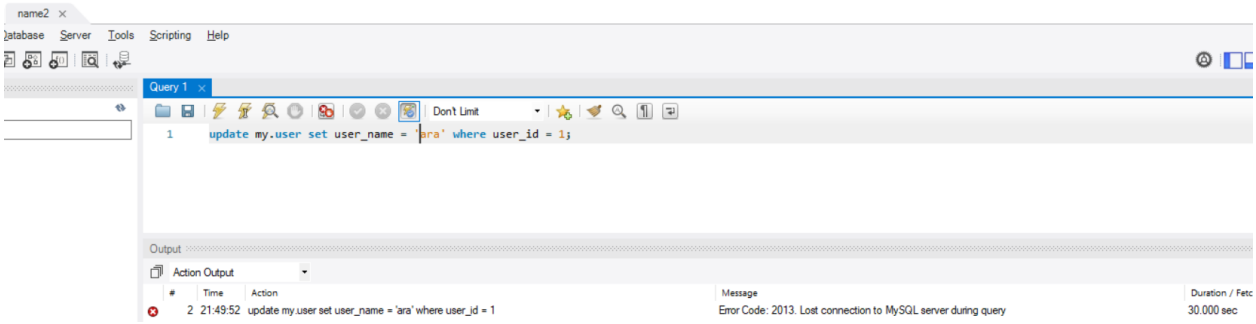

Рисунок 7.9 – Тестування розблокування при підключенні name 2

Завдання 7.2

Перелік 3-5 транзакцій, визначених для бізнес-функцій високонавантаженої системи.

Таблиця 7.1 – Створені транзації

№	Призначення транзакції	Найменування таблиць
1	Оформлення замовлення незареєстрованим клієнтом: - Занесення інформації користувача до таблиці клієнтів - Занесення інформації таблиці замовлень - Занесення інформації до кошику	inno.user, inno.order, order_services
2	Зміна статусу замовлення	inno.order, inno.status
3	Додавання нової послуги	inno.services, inno.paper, inno.fonts

Завдання 7.3

Три збережені процедури, що реалізують транзакції для бази даних з таблицями типу InnoDB (розроблені транзакції мають задовольняти принципи ACID)

1. У разі якщо користувач незареєстрований, система повинна запропонувати ввести контактні дані при оформленні замовлення.

Для тесту введемо невірні дані для кошику. Введено неправильний id послуги в якості 5000. Неправильне число виділено на рисунку 7.10

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_register_user_create_order`()
BEGIN
declare exit handler for sqlexception
        BEGIN
        ROLLBACK;
        END;
    start transaction;

    INSERT INTO `inno`.`user`
    VALUES (4002,'Oleg', 'Majami', '20', 'oleg.jama@outlook.com',
    '(+380)-44-323-3355', '1', '2', 'clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdv',
    '2020-11-19');

    INSERT INTO `inno`.`order`
    VALUES ('4002', '2020-11-19', null, '0', 'test', 'test', '1', '4002');

    insert into `inno`.`order_services` values(4002, 5000);

    update inno.order set order_total_price = check_summ_for_service(4002);
    commit;
END
```

Рисунок 7.10 – Створення процедури для транзакції

```
2 •  call trans_register_user();
```

Output

Action Output

#	Time	Action	Message
1	00:06:31	call trans_register_user()	0 row(s) affected

Рисунок 7.10 – Виклик транзакції

```
1 •  SELECT * FROM inno.order_services where fk_order_id = 4002;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

fk_order_id	fk_service_id
NULL	NULL

Рисунок 7.11 – Результат відкату транзакції

```
1 •  SELECT * FROM inno.user where user_id = 4002;
```

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

user_id	user_name	sname	age	email	phone	gender	fk_user_to_role	History	registerDate
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 7.11 – Результат відкату транзакції

Змінимо дані на вірні та зробимо тест.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_register_user_create_order`()
BEGIN
declare exit handler for sqlexception
        BEGIN
        ROLLBACK;
        END;
    start transaction;

    INSERT INTO `inno`.`user`
    VALUES (4002,'Oleg', 'Majami', '20', 'oleg.jama@outlook.com',
    '(+380)-44-323-3355', '1', '2', 'clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdv',
    '2020-11-19');

    INSERT INTO `inno`.`order`
    VALUES ('4002', '2020-11-19', null, '0', 'test', 'test', '1', '4002');

    insert into `inno`.`order_services` values(4002, 3994);

    update inno.order set order_total_price = check_summ_for_service(4002);
    commit;
END
```

Рисунок 7.12 – Виправлені дані

```
1
2 •  call trans_register_user();
```

Output

Action Output

#	Time	Action	Message
1	00:12:05	Apply changes to trans_register_user_create_order	No changes detected
2	00:12:56	Apply changes to trans_register_user_create_order	No changes detected
3	00:13:03	call trans_register_user()	0 row(s) affected

Рисунок 7.12 – Виклик процедури з виправленими даними

```
1 •  SELECT * FROM inno.order_services where fk_order_id = 4002;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

fk_order_id	fk_service_id
4002	3999
NULL	NULL

Рисунок 7.13 – Результат виконання транзакції для кошику

```
1 •  SELECT * FROM inno.user where user_id = 4002;
```

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

user_id	user_name	sname	age	email	phone	gender	fk_user_to_role	History	registerDate
4002	Oleg	Majami	20	oleg.jama@outlook.com	(+380)-44-323-3355	1	2	clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdv	2020-11-19
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 7.14 – Результат виконання транзакції для користувача

```
1 •  SELECT * FROM inno.order where order_id = 4002;
```

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

order_id	order_start	order_done	order_total_price	order_details	order_VARCHAR	fk_order_to_stat	fk_order_user
4002	2020-11-19	NULL	642	test	test	1	4002
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 7.15 – Результат виконання транзакції для замовлень

2) Зміна статусу

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_change_status`(`ID` int)
BEGIN
declare exit handler for sqlexception
        BEGIN
        ROLLBACK;
        END;
    start transaction;
    select buisness_change_status(`ID`);
    commit;
END
```

Рисунок 7.16 – Створення процедури з транзакцією

```
CREATE DEFINER=`root`@`localhost` FUNCTION `buisness_change_status`(`ID`int) RETURNS varchar(40) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
    declare `current` int;
    select fk_order_to_stat into `current` from inno.order where order_id = `ID`;
    if (`current`>= (select status_id from inno.status order by status_id desc limit 1))
    then
        signal sqlstate '45000' set message_text = 'can not update data ';
    else
    update inno.order set fk_order_to_stat = (`current` +1) where order_id = ID;
    end if;
RETURN 1;
END
```

Рисунок 7.17 – Викликана функція, що змінює статус на наступний

```
1 •  call trans_change_status(4006);
2 •  select * from inno.order where order_id = 4006;
```

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

order_id	order_start	order_done	order_total_price	order_details	order_VARCHAR	fk_order_to_stat	fk_order_user
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result 8 | order 9

Output

Action Output

#	Time	Action	Message
1	01:36:37	call trans_change_status(4006)	1 row(s) returned
2	01:36:37	select * from inno.order where order_id = 4006	0 row(s) returned

Рисунок 7.18 – Тестування невірно обраного замовлення

Демонстрація роботи транзакції:

```
2 •  select * from inno.order where order_id = 4002;
```

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

order_id	order_start	order_done	order_total_price	order_details	order_VARCHAR	fk_order_to_stat	fk_order_user
4002	2020-11-19	NULL	642	test	test	3	4002
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 7.19 – Дані до виконання транзакції

```
1 •  call trans_change_status(4002);
2 •  select * from inno.order where order_id = 4002;
```

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

order_id	order_start	order_done	order_total_price	order_details	order_VARCHAR	fk_order_to_stat	fk_order_user
4002	2020-11-19	NULL	642	test	test	4	4002
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result 15 | order 16

Output

Action Output

#	Time	Action	Message
1	01:39:50	call trans_change_status(4002)	1 row(s) returned
2	01:39:50	select * from inno.order where order_id = 4002	1 row(s) returned

Рисунок 7.20 – Вдале виконання транзакції

У результаті fk_order_to_stat набуває кінцевого значення – «Виконано», якщо користувач захоче змінити його на наступний – спрацює Rollback, через переривання, що визване в функції (див рис. 7.17).

```
call trans_change_status(4002);
select * from inno.order where order_id = 4002;
```

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

order_id	order_start	order_done	order_total_price	order_details	order_VARCHAR	fk_order_to_stat	fk_order_user
4002	2020-11-19	NULL	642	test	test	4	4002
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result 19 | order 20

Output

Action Output

#	Time	Action	Message
1	01:42:15	call trans_change_status(4002)	0 row(s) returned
2	01:42:15	select * from inno.order where order_id = 4002	1 row(s) returned

Рисунок 7.21 – Результат відкату

3) Додавання нової послуги

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_add_service`()
BEGIN
declare exit handler for sqlexception
        BEGIN
        ROLLBACK;
        END;
    start transaction;
    INSERT INTO `inno`.`fonts` VALUES ('4001', 'IIWW3', '14', '2.5', '2.5', '14',null, 'test', '2020-11-19');
    INSERT INTO `inno`.`paper`  VALUES ('4501', 'NN332', '2', '4', '50', '20', '1', '5', 'test', '2020-11-19');
    INSERT INTO `inno`.`services`  VALUES ('4001', 'Light correction', 'test', '250', '#2007574', '4501', '2020-11-19', '4001');
    commit;
END
```

Рисунок 7.22 – Створення процедури для транзакції

```
1 •  select * from inno.services
```

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

service_id	service_name	service_desc	service_price	service_code	fk_service_paper	service_date	fk_service_fonts
3991	_S1474	8QA5SN6N Q6B3QQ28 WOP1DJNM CWJVPEZR ...	685	#2007655	296	2013-05-21	2231
3992	OR1823	JDZY1FI0 VS5AMD0V SQQN1DIQ CD9BDNFF BM...	922	#2007656	1094	2013-05-22	854
3993	XP1580	XVPV0QCR 3LO1NR34 OD6NAF9S 30X3IPZY E1...	626	#2007657	3470	2013-05-23	4
3994	A[934	EKPMVBJ3 SWH9UNTU SS9GP7GE L7VHZWOU J...	576	#2007658	347	2013-05-24	103
3995	JJ1413	4QP48HKA AL8XFNAN 2Q5AIMOR HUZOC2GC 9...	721	#2007659	930	2013-05-25	116
3996	UQ1877	18BHDLFU 1XCPA0KP 5XT2B5GD NWZ7RKG6 TR...	634	#2007660	2839	2013-05-26	3504
3997	EL717	AMLLLTMD NCWL2QFK CWQ4M272 UGWVVYL3 ...	874	#2007661	1705	2013-05-27	2443
3998	IC977	QDJYSSP5 NE7R2HOT 1UEUZX90 B1F1U031 DN...	635	#2007662	3266	2013-05-28	1106
3999	I]1448	YK0DQ2OH QH9358V2 SQ8ALV69 S7E7NG6J IU...	642	#2007663	496	2013-05-29	515
4000	\^1446	NR3K24P5 XEKITWMG NRJY0ZX8 IM25P4N8 LR...	784	#2007664	2071	2013-05-30	3440
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 7.23 – Дані про послуги до транзакції

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_add_service`()
BEGIN
declare exit handler for sqlexception
        BEGIN
        ROLLBACK;
        END;
    start transaction;
    INSERT INTO `inno`.`fonts` VALUES ('4001', 'IIWW3', '14', '2.5', '2.5', '14',null, 'test', '2020-11-19');
    INSERT INTO `inno`.`paper`  VALUES ('4501', 'NN332', '2', '4', '50', '20', '1', '5', 'test', '2020-11-19');
    INSERT INTO `inno`.`services`  VALUES ('4001', 'Light correction', 'test', '250', '#2007574', '4505', '2020-11-19', '4001');
    commit;
END
```

Рисунок 7.24 – Створення процедури з невірними даними

```
1 •  call trans_add_service()
```

Output

Action Output

#	Time	Action	Message
1	02:12:05	Apply changes to trans_add_service	No changes detected
2	02:13:31	Apply changes to trans_add_service	Changes applied
3	02:13:46	call trans_add_service()	0 row(s) affected

Рисунок 7.25 – Виклик процедури

```
1 •  select service_id from inno.services where service_id = 4001;
```

Output

Action Output

#	Time	Action	Message
1	02:14:32	select service_id from inno.services where service_id = 4001	0 row(s) returned

Рисунок 7.26 – Перевірка відкату для таблиці послуг

```
1 •  select paper_id from inno.paper where paper_id = 4501
```

Output

Action Output

#	Time	Action	Message
1	02:14:32	select service_id from inno.services where service_id = 4001	0 row(s) returned
2	02:14:45	select paper_id from inno.paper where paper_id = 4501	0 row(s) returned

Рисунок 7.27 – Перевірка відкату для таблиці паперу

```
1 •  select fonts_id from inno.fonts where fonts_id  = 4001
```

Output

Action Output

#	Time	Action	Message
1	02:14:32	select service_id from inno.services where service_id = 4001	0 row(s) returned
2	02:14:45	select paper_id from inno.paper where paper_id = 4501	0 row(s) returned
3	02:15:49	select fonts_id from inno.fonts where fonts_id = 4001	0 row(s) returned

Рисунок 7.28 – Перевірка відкату для таблиці шрифтів

Виправимо дані в процедурі на вірні:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_add_service`()
BEGIN
declare exit handler for sqlexception
        BEGIN
        ROLLBACK;
        END;
    start transaction;
    INSERT INTO `inno`.`fonts` VALUES ('4001', 'IIWW3', '14', '2.5', '2.5', '14',null, 'test', '2020-11-19');
    INSERT INTO `inno`.`paper`  VALUES ('4501', 'NN332', '2', '4', '50', '20', '1', '5', 'test', '2020-11-19');
    INSERT INTO `inno`.`services`  VALUES ('4001', 'Light correction', 'test', '250', '#2007574', '4501', '2020-11-19', '4001');
    commit;
END
```

Рисунок 7.29 – Виправлення даних

```
1 •  select * from inno.services where service_id = 4001;
```

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

service_id	service_name	service_desc	service_price	service_code	fk_service_paper	service_date	fk_service_fonts
4001	Light correction	test	250	#2007574	4501	2020-11-19	4001
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

services 8

Output

Action Output

#	Time	Action	Message
1	02:19:19	call trans_add_service()	0 row(s) affected
2	02:19:35	select * from inno.services where service_id = 4001	1 row(s) returned

Рисунок 7.30 – Перегляд результату додавання даних до таблиці послуг

```
1 •  select paper_id from inno.paper where paper_id = 4501
```

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

paper_id
4501
NULL

paper 7

Output

Action Output

#	Time	Action	Message
1	02:19:19	call trans_add_service()	0 row(s) affected
2	02:19:35	select * from inno.services where service_id = 4001	1 row(s) returned
3	02:19:54	select paper_id from inno.paper where paper_id = 4501	1 row(s) returned

Рисунок 7.31 – Перегляд результату додавання даних до таблиці паперу

```
1 •  select fonts_id from inno.fonts where fonts_id  = 4001
```

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

fonts_id
4001
NULL

fonts 5

Output

Action Output

#	Time	Action	Message
1	02:19:19	call trans_add_service()	0 row(s) affected
2	02:19:35	select * from inno.services where service_id = 4001	1 row(s) returned
3	02:19:54	select paper_id from inno.paper where paper_id = 4501	1 row(s) returned
4	02:21:36	select fonts_id from inno.fonts where fonts_id = 4001	1 row(s) returned

Рисунок 7.32 – Перегляд результату додавання даних до таблиці шрифтів

Записи бінарного журналу

1. Бінарний журнал для реєстрації користувача та створення замовлення. Зверху вказано невірні дані, нижче – виправлений на правильні дані варіант.

```
declare exit handler for sqlexception
                BEGIN
                ROLLBACK;
                END;
    start transaction;

    INSERT INTO `inno`.`user`
        VALUES (4002,'Oleg', 'Majami', '20', 'oleg.jama@outlook.com',
    '(+380)-44-323-3355', '1', '2', 'clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdv',
    '2020-11-19');

        INSERT INTO `inno`.`order`
    VALUES ('4002', '2020-11-19', null, '0', 'test', 'test', '1', '4002');

    insert into `inno`.`order_services` values(4002, 5000);

    update inno.order set order_total_price = check_summ_for_service(4002);
    commit;
ENDÉg_ë zP_"▯  M   ë4p   ▯                        )"      *"      òÃm,Ø¶D•8▯ ÿàþ▯ zP_▯  ‘   |5p   3       ▯ -      ▯  @    ▯stdÿ ÿ ÿ ♠▯inno ▯Ií#     ▯ÿ inno DROP procedure IF EXISTS `trans_register_user_create_
BEGIN
declare exit handler for sqlexception
                BEGIN
                ROLLBACK;
                END;
    start transaction;

    INSERT INTO `inno`.`user`
        VALUES (4002,'Oleg', 'Majami', '20', 'oleg.jama@outlook.com',
    '(+380)-44-323-3355', '1', '2', 'clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdv',
    '2020-11-19');

        INSERT INTO `inno`.`order`
    VALUES ('4002', '2020-11-19', null, '0', 'test', 'test', '1', '4002');

    insert into `inno`.`order_services` values(4002, 3994);

    update inno.order set order_total_price = check_summ_for_service(4002);
    commit;
ENDFç Iz°zP_"▯   O   Õ9p                          ,"      -"      ▯ŠŠƒØ¶▯:I•8▯ W•D▯°zP_▯   M   ":p ▯ 3       ▯        ▯  @    ▯stdÿ ÿ ÿ ♠▯ÿ inno BEGINj▯$é°zP_▯   A   c:p   ›      ▯ ▯inno ▯order ▯

▯▯▯▯▯ ‡ ▯▯  ▯ÀÏÍÏ[°zP_ ▯  >   ¡:p   ›      ▯  ▯ÿ„¢▯  sÉ▯ € D▯ test▯test▯   ˆZ<▯°zP_▯▯      À:p   cí#     oõ†ÜÏzP_"▯   O   ▯;p                                -"      ."      .Œ^I¢¶▯)▯•8▯ 8HíÏzP_▯   M   \;p ▯ 3       ▯        ▯
▯▯▯▯▯▯▯▯
     ‡ ‡ ,▯‡  @▯▯  ▯À S!WDÈÏzP_▯   •   ;<p   r     ▯
ÿÿ  ¢▯  ▯Oleg▯Majami▯   ▯ oleg.jama@outlook.com▯(+380)-44-323-3355▯    ' clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdvsÉ▯éÅÌÿÏzP_▯▯   A   |<p   ›      ▯ ▯inno ▯order ▯

▯▯▯▯▯ ‡ ▯▯  ▯ÀfX▯Ãõ▯ÏzP_▯   B   %<p   ›      ▯  ▯ÿ▯¢▯  sÉ▯   ▯ test▯test▯   ¢▯  ▯▯▯▯ÏzP_▯▯   =   ù<p   ™     ▯ ▯inno ▯order_services ▯  ▯ 4V¡æÏzP_▯   ,   '=p   ™     ▯   ÿ ¢▯  Ÿ▯  c▯Ò³ÏzP_▯▯   A   h=p   ›      ▯ ▯inno ▯order ▯
```

Рисунок 7.33 – Бінарний журнал для першої процедури

2. Бінарний журнал для зміни статусу замовлення. Зверху вказано невірні дані, нижче – виправлений на правильні дані варіант

```
    DETERMINISTIC
BEGIN
        declare `current` int;
    select fk_order_to_stat into `current` from inno.order where order_id = `ID`;
    if (`current`>= (select status_id from inno.status order by status_id desc limit 1))
    then
            signal sqlstate '45000' set message_text = 'can not update data ';
        else
        update inno.order set fk_order_to_stat = (`current` +1) where order_id = ID;
    end if;
RETURN 1;
END
... inno DROP procedure IF EXISTS `trans_change_status` ...
... root localhost inno ... inno CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_change_status`(`ID` int)
BEGIN
declare exit handler for sqlexception
            BEGIN
            ROLLBACK;
            END;
    start transaction;

    select buisness_change_status(`ID`);
    commit;
END ... inno BEGIN ...
... test test ... test test ...
... test test ... test test ...
... test test ... test test ...
... test test ... test test ...
... inno BEGIN) ... inno order ...
... test test ... test test ...
```

Рисунок 7.34 – Бінарний журнал для другої процедури

3. Бінарний журнал для додавання послуги. Зверху вказано невірні дані, нижче – виправлений на правильні дані варіант

```
BEGIN
declare exit handler for sqlexception
            BEGIN
            ROLLBACK;
            END;
    start transaction;
    INSERT INTO `inno`.`fonts` VALUES ('4001', 'IIWW3', '14', '2.5', '2.5', '14',null, 'test', '2020-11-19');
    INSERT INTO `inno`.`paper`  VALUES ('4501', 'NN332', '2', '4', '50', '20', '1', '5', 'test', '2020-11-19');
    INSERT INTO `inno`.`services`  VALUES ('4001', 'Light correction', 'test', '250', '#2007574', '4505', '2020-11-19', '4001');
        commit;
END ... inno DROP procedure IF EXISTS `trans_add_service` ...
BEGIN
declare exit handler for sqlexception
            BEGIN
            ROLLBACK;
            END;
    start transaction;
    INSERT INTO `inno`.`fonts` VALUES ('4001', 'IIWW3', '14', '2.5', '2.5', '14',null, 'test', '2020-11-19');
    INSERT INTO `inno`.`paper`  VALUES ('4501', 'NN332', '2', '4', '50', '20', '1', '5', 'test', '2020-11-19');
    INSERT INTO `inno`.`services`  VALUES ('4001', 'Light correction', 'test', '250', '#2007574', '4501', '2020-11-19', '4001');
        commit;
END ... inno BEGIN ... inno fonts
... IIWW3 ... tests ...
... NN332 ... tests ...
... Light correction test #2007574 ...
... Light correction test #2007574 ...
... NN332 ... tests ...
... IIWW3 ... tests ...
... IIWW3 ... tests ... inno paper
... NN332 ... tests ... inno services
... Light correction test #2007574 ...
```

Рисунок 7.35 – Бінарний журнал для третьої процедури

Висновок:

В даній практичній роботі було освоєно роботу з транзакціями та log-файлами. Було досліджено впливу рівня ізоляції на зміну даних (деякі записи змогли оновлюватись під час відкату за допомогою виклику функцій).

Було створено три процедури, що формують транзакцію для основних бізнес-функцій: додавання нового користувача та замовлення, зміни статусу замовлення, додавання послуги. Досліджено переваги та недоліки кожного виду ізоляцій для транзакцій.