

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА СИСТЕМОТЕХНІКИ

Звіт

З практичної роботи №3

На тему: «Реалізація програмної логіки й забезпечення цілісності зв'язків за допомогою тригерів для високонавантаженої баз даних на платформі СУБД MySQL»

з дисципліни «Проектування високонавантажених систем зберігання даних»

Виконав:

ст. гр. ІТКНу-19-2

Козорог І.Г.

Перевірів викладач:

Коваленко А.І.

Харків 2020

Мета: Набуття практичних навичок з розробки тригерів для підтримки цілісності зв'язків, модифікації даних і забезпечення основних бізнес-процесів високонавантаженої інформаційної системи. Формування необхідних практичних умінь для аналізу плану виконання SQL-запитів за допомогою оператора EXPLAIN. Формування необхідних практичних умінь для створення тригерів, з урахуванням особливостей реалізації логіки роботи інтерфейсу високонавантаженої інформаційної системи зберігання даних.

Тема індивідуального завдання: Інформаційна система «Надання послуг вантажоперевезень»

Таблиця 1 – Перелік доступних операцій тригерів

Доступність операцій тригера	BEFORE INSERT	AFTER INSERT	BEFORE UPDATE	AFTER UPDATE	BEFORE DELETE	AFTER DELETE
1. NEW доступний – Так/Ні	так	так	так	так	ні	ні
2. OLD доступний – Так/Ні	ні	ні	так	так	так	так
3. Доступ до зміни полів NEW – Так/Ні						
4. Значення автоінкремента (PK) доступно – Так/Ні	ні	так	ні	так	ні	Ні
5. Доступ до значень полів та їх значень (DEFAULT), що явно в інструкціях INSERT, UPDATE, DELETE не фігурували – Так/Ні	ні	так	ні	так	ні	так
6. Можливість скасування операцій INSERT // UPDATE / DELETE – Так/Ні	так	так	так	так	так	так

Таблиця 2 – Переваги використання тригерів

№	Аналізовані параметри	Тригер	SQL-запит
1	Принцип виконання SQL-коду в СУБД MySQL	Збережений код	
2	Вплив на апаратні ресурси	Впливає на апаратні ресурси сервера	Впливає на все
3	Вплив на мережевий трафік	менший	більший
4	Можливість, що-небудь автоматизувати	можна	можна
5	SQL-синтаксис (обмеження)	За допомогою ключового слова WHEN	
6	Принцип (фізичне місце) зберігання SQL-коду	Сервер	Клієнтська частина

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

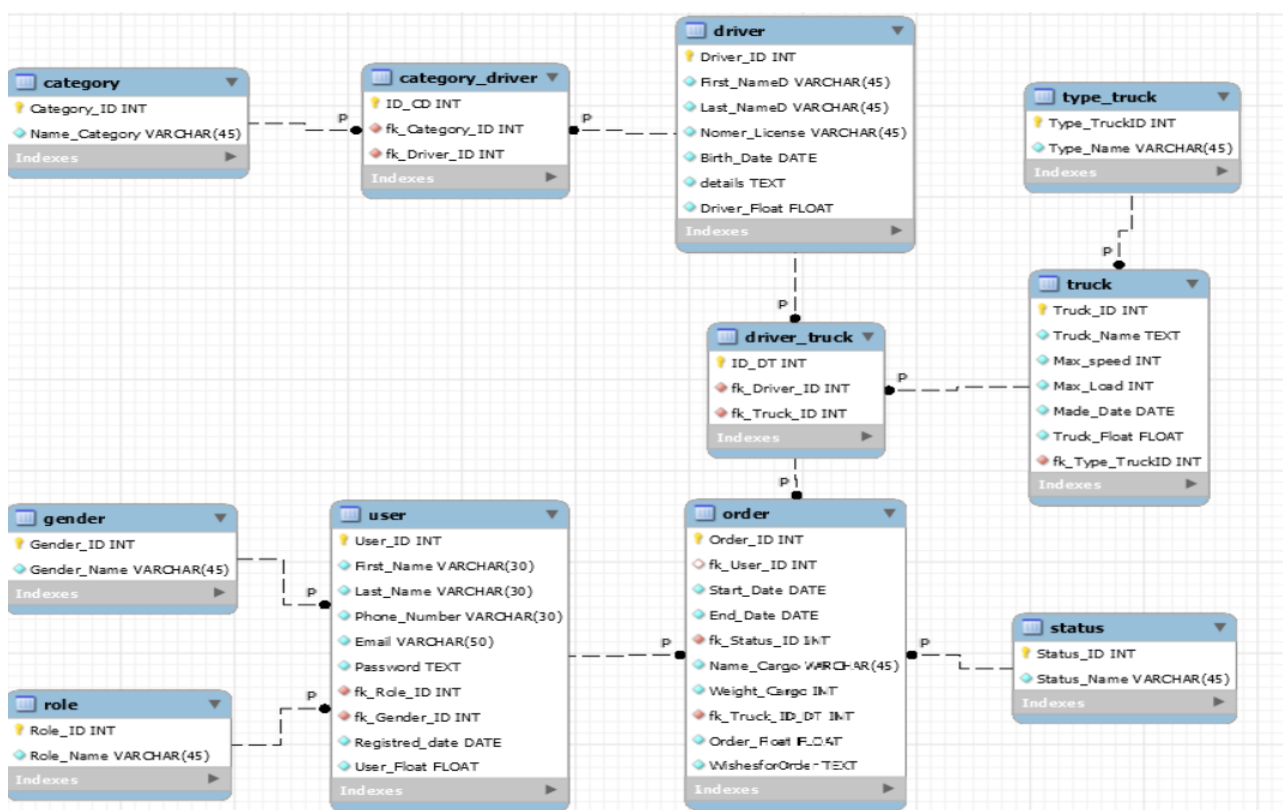


Рисунок 1 – Фізична модель даних

Завдання 3.1 Таблиця з переліком типів посилальної цілісності зв'язків за зовнішнім ключем для всіх таблиць типу MySQL, реалізованих за допомогою тригерів.

Таблиця 3 – Перелік типів посилальної цілісності за допомогою тригерів

№	Ім'я таблиці 1, зовнішній ключ	Ім'я таблиці 2, первинний ключ	Тип посилальної цілісності для таблиці 1	Тип посилальної цілісності для таблиці 2	Тригер
1	User, fk_Role_ID	Role, Role_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
2	User, fk_Role_ID	Role, Role_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
3	User, fk_Gender_ID	Gender, Gender_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
4	User, fk_Gender_ID	Gender, Gender_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
5	User, fk_Role_ID	Role, Role_ID	ON INSERT RESTRICT	ON INSERT RESTRICT	Тригер Before insert
6	User, fk_Gender_ID	Gender, Gender_ID	ON INSERT RESTRICT	ON INSERT RESTRICT	Тригер Before insert
7	Order, fk_User_ID	User, User_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
8	Order, fk_User_ID	User, User_ID	ON DELETE SET NULL	ON DELETE SET NULL	Тригер Before delete
9	Order, fk_User_ID	User, User_ID	ON INSERT RESTRICT	ON INSERT RESTRICT	Тригер Before insert
10	Order, fk_Status_ID	Status, Status_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
11	Order, fk_Status_ID	Status, Status_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
12	Order, fk_Status_ID	Status, Status_ID	ON INSERT RESTRICT	ON INSERT RESTRICT	Тригер Before insert
13	Order, fk_Truck_ID_D T	Driver_Truck, ID_DT	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
14	Order, fk_Truck_ID_D T	Driver_Truck, ID_DT	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete

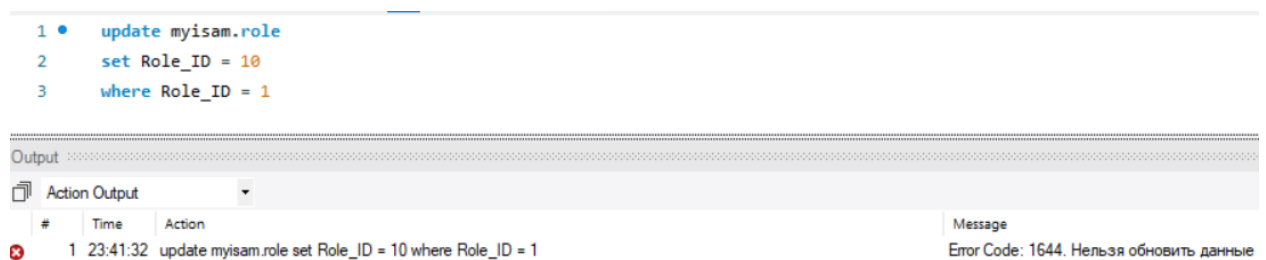
15	Order, fk_Truck_ID_D T	Driver_Truck, ID_DT	ON ISNERT RESTRICT	ON INSERT RESTRICT	Тригер Before insert
16	Truck, fk_Type_TruckI D	Type_Truck, Type_TruckID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
17	Truck, fk_Type_TruckI D	Type_Truck, Type_TruckID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
18	Truck, fk_Type_TruckI D	Type_Truck, Type_TruckID	ON ISNERT RESTRICT	ON ISNERT RESTRICT	Тригер Before insert
19	Driver_Truck, fk_Truck_ID	Truck, Truck_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
20	Driver_Truck, fk_Truck_ID	Truck, Truck_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
21	Driver_Truck, fk_Truck_ID	Truck, Truck_ID	ON ISNERT RESTRICT	ON ISNERT RESTRICT	Тригер Before insert
22	Driver_Truck, fk_Driver_ID	Driver, Driver_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
23	Driver_Truck, fk_Driver_ID	Driver, Driver_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
24	Driver_Truck, fk_Driver_ID	Driver, Driver_ID	ON ISNERT RESTRICT	ON ISNERT RESTRICT	Тригер Before insert
25	Category_Drive r fk_Driver_ID	Driver, Driver_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
26	Category_Drive r fk_Driver_ID	Driver, Driver_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
27	Category_Drive r fk_Driver_ID	Driver, Driver_ID	ON ISNERT RESTRICT	ON ISNERT RESTRICT	Тригер Before insert
28	Category_Drive r fk_Category_ID	Category, Category_ID	ON UPDATE RESTRICT	ON UPDATE RESTRICT	Тригер Before update
29	Category_Drive r fk_Category_ID	Category, Category_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete
30	Category_Drive r fk_Category_ID	Category, Category_ID	ON ISNERT RESTRICT	ON ISNERT RESTRICT	Тригер Before insert
31	User, User_ID	Order, fk_Order_ID	ON DELETE RESTRICT	ON DELETE RESTRICT	Тригер Before delete

1) Тригер для таблиці ролей типу Before Update:

```
CREATE DEFINER='root'@'localhost' TRIGGER `role_BEFORE_UPDATE` BEFORE UPDATE ON `role` FOR EACH ROW BEGIN
    if exists (select * from myisam.user where fk_Role_ID = old.Role_ID)
    and old.Role_ID != new.Role_ID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END
```

Рисунок 2 – Код Sql-тригера

Після виконання запиту update, маємо результат:



The screenshot shows the execution of an SQL update query. The query is: `update myisam.role set Role_ID = 10 where Role_ID = 1`. The output table shows one row with the time 23:41:32 and the action 'update myisam.role set Role_ID = 10 where Role_ID = 1'. The message column displays 'Error Code: 1644. Нельзя обновить данные'.

#	Time	Action	Message
1	23:41:32	update myisam.role set Role_ID = 10 where Role_ID = 1	Error Code: 1644. Нельзя обновить данные

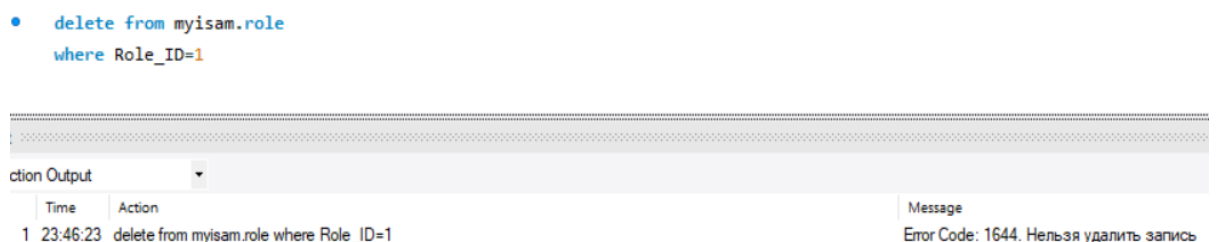
Рисунок 3 – Результат запиту update

2) Тригер для таблиці ролей типу Before Delete

```
CREATE DEFINER='root'@'localhost' TRIGGER `role_BEFORE_DELETE` BEFORE DELETE ON `role` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.user where fk_Role_ID = old.Role_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 4 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:



The screenshot shows the execution of an SQL delete query. The query is: `delete from myisam.role where Role_ID=1`. The output table shows one row with the time 23:46:23 and the action 'delete from myisam.role where Role_ID=1'. The message column displays 'Error Code: 1644. Нельзя удалить запись'.

#	Time	Action	Message
1	23:46:23	delete from myisam.role where Role_ID=1	Error Code: 1644. Нельзя удалить запись

Рисунок 5 – Результат запиту delete

3) Тригер для таблиці ролей типу Before Insert:

```
if not exists (select 1 from myisam.role where Role_ID=new.fk_Role_ID)
then
    SIGNAL SQLSTATE '45000' SET message_text= 'Нельзя вставить запись';
end if;
```

Рисунок 6 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.user
values(51,'Igor','Kozorog',380953334633, 'igor1' , 'q1w23',3,1,"19.12.2020",421.5)
```

ion Output		
Time	Action	Message
1 23:51:41	insert into myisam.user values(51,'Igor','Kozorog',380953334633, 'igor1' , 'q1w23',3,1,"19.12.2020",421.5)	Error Code: 1644. Нельзя вставить запись

Рисунок 7 – Результат запиту insert

4) Тригер для таблиці статі типу Before Update:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `gender_BEFORE_UPDATE` BEFORE UPDATE ON `gender` FOR EACH ROW BEGIN
    if exists(select 1 from myisam.user where fk_Gender_ID=old.Gender_ID)
    and old.Gender_ID != new.Gender_ID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END
```

Рисунок 8 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

```
update myisam.gender
set Gender_ID = 4
where Gender_ID = 1
```

ion Output		
Time	Action	Message
1 23:56:58	update myisam.gender set Gender_ID = 4 where Gender_ID = 1	Error Code: 1644. Нельзя обновить данные

Рисунок 9 – Результат запиту update

5) Тригер для таблиці статі типу Before Delete:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `gender_BEFORE_DELETE` BEFORE DELETE ON `gender` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.user where fk_Gender_ID = old.Gender_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 10 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:

```
delete from myisam.gender
where Gender_ID = 1;
```

ion Output		
Time	Action	Message
1 00:08:00	delete from myisam.gender where Gender_ID = 1	Error Code: 1644. Нельзя удалить запись

Рисунок 11 – Результат запиту delete

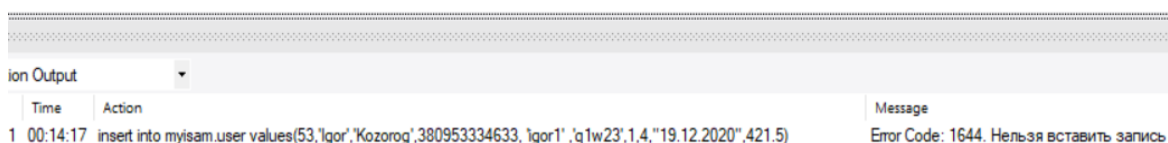
6) Тригер для таблиці статі типу Before Insert:

```
if not exists(select 1 from myisam.gender where Gender_ID=new.fk_Gender_ID)
then
    SIGNAL SQLSTATE '45000' set MESSAGE_TEXT = 'Нельзя вставить запись';
end if;
```

Рисунок 12 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.user
values(53,'Igor','Kozorog',380953334633, 'igor1' , 'q1w23',1,4,"19.12.2020",421.5)
```



ion Output		Message
Time	Action	
1 00:14:17	insert into myisam.user values(53,'Igor','Kozorog',380953334633, 'igor1' , 'q1w23',1,4,"19.12.2020",421.5)	Error Code: 1644. Нельзя вставить запись

Рисунок 13 – Результат роботи insert

7) Тригер для таблиці користувачів типу Update Before:

```
CREATE DEFINER='root'@'localhost' TRIGGER `user_BEFORE_UPDATE` BEFORE UPDATE ON `user` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.order where fk_User_ID = old.User_ID)
    and old.User_ID != new.User_ID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END
```

Рисунок 14 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

```
update myisam.user
set User_ID = 72
where User_ID = 9
```



on Output		Message
Time	Action	
15:46:34	update myisam.user set User_ID = 72 where User_ID = 9	Error Code: 1644. Нельзя обновить данные

Рисунок 15 - Результат запиту update

8) Тригер для таблиці користувачів типу Before Delete:

```
CREATE DEFINER='root'@'localhost' TRIGGER `user_BEFORE_DELETE` BEFORE DELETE ON `user` FOR EACH ROW BEGIN
    if exists(select 2 from myisam.order where fk_User_ID = old.User_ID)
    then
        update myisam.order
        Set fk_User_ID = null
        where fk_User_ID = old.User_ID;
    end if;
END
```

Рисунок 16 - Код Sql-тригера

Після виконання запиту видалення, користувач видалається, а його запис в таблиці замовлень змінюється на null:

Order_ID	fk_User_ID	Start_Date	End_Date	fk_Status_ID	Name_Cargo	Weight_Cargo	fk_Truck_ID_DT	Order_Float	WishesforOrder
47	NULL	2017-12-28	2018-01-23	3	YCHEBD	33283	28	450.2	She gave Denry a key. He went over to the des...
22	3	2012-06-08	2012-06-25	2	LDLXED	14223	19	484.8	As she turned away, Denry saw that she had d...
26	3	2010-04-08	2010-04-25	1	VTBROW	1877	15	447.2	And both Denry and the Countess laughed and ...
38	3	2014-09-04	2014-09-29	2	AINOIW	3683	29	457.7	The next Monday, all the neighbours knew that ...
12	9	2008-04-09	2008-04-27	1	MAANJY	37353	7	469.5	One of the best dancing teachers in Bursley wa...

Рисунок 17 – Запис в таблиці замовлень до видалення користувача

Order_ID	fk_User_ID	Start_Date	End_Date	fk_Status_ID	Name_Cargo	Weight_Cargo	fk_Truck_ID_DT	Order_Float	WishesforOrder
12	NULL	2008-04-09	2008-04-27	1	MAANJY	37353	7	469.5	One of the best dancing teachers in Bursley wa...
47	NULL	2017-12-28	2018-01-23	3	YCHEBD	33283	28	450.2	She gave Denry a key. He went over to the des...

Рисунок 18 – Запис в таблиці замовлень після видалення користувача

9) Тригер для таблиці користувачів типу Before Insert

```
CREATE DEFINER='root'@'localhost' TRIGGER `order_BEFORE_INSERT` BEFORE INSERT ON `order` FOR EACH ROW BEGIN
    if not exists(select 1 from myisam.user where User_ID=new.fk_User_ID)
    then
        signal sqlstate '45000' set message_text='не правильно';
    end if;
```

Рисунок 19 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.order
values(51,70,'10.11.2020','15.10.2020',1,'sand',20000,7,500.5,'i dont have wishes')
```

Time	Action	Message
1 15:59:42	insert into myisam.order values(51,70,'10.11.2020','15.10.2020',1,'sand',20000,7,500.5,'i dont have wishes')	Error Code: 1644. Нельзя вставить запись

Рисунок 20 – Результат запиту Insert

10) Тригер для таблиці статусу типу Before Update:

```
CREATE DEFINER='root'@'localhost' TRIGGER `status_BEFORE_UPDATE` BEFORE UPDATE ON `status` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.order where fk_Status_ID = old.Status_ID)
    and old.Status_ID != new.Status_ID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END
```

Рисунок 21 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

```
update myisam.status
set Status_ID = 10
where Status_ID = 1
```

Time	Action	Message
1 16:06:00	update myisam.status set Status_ID = 10 where Status_ID = 1	Error Code: 1644. Нельзя обновить данные

Рисунок 22 – Результат запиту Update

11) Тригер для таблиці статусу типу Before Delete:

```
CREATE DEFINER='root'@'localhost' TRIGGER `status_BEFORE_DELETE` BEFORE DELETE ON `status` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.order where fk_Status_ID = old.Status_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 23 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:

```
delete from myisam.status
where Status_ID = 1
```



Execution Output		
Time	Action	Message
1 16:07:56	delete from myisam.status where Status_ID = 1	Error Code: 1644. Нельзя удалить запись

Рисунок 24 –Результат запиту Delete

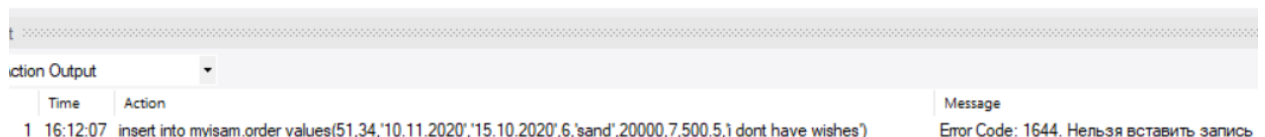
12) Тригер для таблиці статусу типу Before Insert:

```
if not exists(select 1 from myisam.status where Status_ID = new.fk_Status_ID)
then
    signal sqlstate '45000' set message_text = 'Нельзя вставить запись';
end if;
```

Рисунок 25 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.order
values(51,34,'10.11.2020','15.10.2020',6,'sand',20000,7,500.5,'i dont have wishes')
```



Execution Output		
Time	Action	Message
1 16:12:07	insert into myisam.order values(51,34,'10.11.2020','15.10.2020',6,'sand',20000,7,500.5,'i dont have wishes')	Error Code: 1644. Нельзя вставить запись

Рисунок 26 – Результат запиту Insert

13) Тригер для таблиці водій_вантажівка типу Before Update:

```
CREATE DEFINER='root'@'localhost' TRIGGER `driver_truck_BEFORE_UPDATE` BEFORE UPDATE ON `driver_truck` FOR EACH ROW BEGIN
    if exists(select 1 from myisam.order where fk_Truck_ID_DT = old.ID_DT)
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить запись';
    end if;
END
```

Рисунок 27 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

```
update myisam.driver_truck
set ID_DT = 199
where ID_DT = 1
```

Time	Action	Message
1 16:29:09	update myisam.driver_truck set ID_DT = 199 where ID_DT = 1	Error Code: 1644. Нельзя обновить запись

Рисунок 28 – Результат запросу Update

14) Триггер для таблицы водій_вантажівка типу Before Delete:

```
CREATE DEFINER='root'@'localhost' TRIGGER `driver_truck_BEFORE_DELETE` BEFORE DELETE ON `driver_truck` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.order where fk_Truck_ID_DT = old.ID_DT)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 29 - Код Sql-триггера

Після виконання запиту Delete, маємо результат:

```
delete from myisam.driver_truck
where ID_DT = 1
```

Time	Action	Message
1 16:38:45	delete from myisam.driver_truck where ID_DT = 1	Error Code: 1644. Нельзя удалить запись

Рисунок 30 – Результат запросу Delete

15) Триггер для таблицы водій_вантажівка типу Before Insert:

```
if not exists(select 1 from myisam.driver_truck where ID_DT = new.fk_Truck_ID_DT)
then
    signal sqlstate '45000' set message_text = 'Нельзя вставить запись';
end if;
```

Рисунок 31 - Код Sql-триггера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.order
values(51,30,'10.11.2020','15.10.2020',2,'sand',20000,70,500.5,'i dont have wishes')
```

Time	Action	Message
1 16:42:04	insert into myisam.order values(51,30,'10.11.2020','15.10.2020',2,'sand',20000,70,500.5,'i dont have wishes')	Error Code: 1644. Нельзя вставить запись

Рисунок 32 – Результат запросу Insert

16) Триггер для таблицы вантажівок типу Before Update:

```
CREATE DEFINER='root'@'localhost' TRIGGER `truck_BEFORE_UPDATE` BEFORE UPDATE ON `truck` FOR EACH ROW BEGIN
    if exists (select * from myisam.driver_truck where fk_Truck_ID = old.Truck_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END
```

Рисунок 33 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

update myisam.truck		
set Truck_ID = 70		
where Truck_ID = 2		
Action Output		
Time	Action	Message
1 17:01:43	update myisam.truck set Truck_ID = 70 where Truck_ID = 2	Error Code: 1644. Нельзя обновить данные

Рисунок 34 – Результат запиту Update

17) Тригер для таблиці вантажівок типу Delete:

```
CREATE DEFINER='root'@'localhost' TRIGGER `truck_BEFORE_DELETE` BEFORE DELETE ON `truck` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.driver_truck where fk_Truck_ID = old.Truck_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 35 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:

delete from myisam.truck		
where Truck_ID = 2		
ion Output		
Time	Action	Message
1 17:04:47	delete from myisam.truck where Truck_ID = 2	Error Code: 1644. Нельзя удалить запись

Рисунок 36 – Результат запиту Delete

18) Тригер для таблиці вантажівок типу Insert:

```
CREATE DEFINER='root'@'localhost' TRIGGER `driver_truck_BEFORE_INSERT` BEFORE INSERT ON `driver_truck` FOR EACH ROW BEGIN
    if not exists (select 1 from myisam.truck where Truck_ID = new.fk_Truck_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя вставить запись';
    end if;
END
```

Рисунок 37 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:



Рисунок 38 – Результат виконання Insert

19) Тригер для таблиці тип_вантажівки типу Before Update

```

CREATE DEFINER=`root`@`localhost` TRIGGER `type_truck_BEFORE_UPDATE` BEFORE UPDATE ON `type_truck` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.truck where fk_Type_TruckID = old.Type_TruckID)
    and old.Type_TruckID != new.Type_TruckID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END

```

Рисунок 39 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

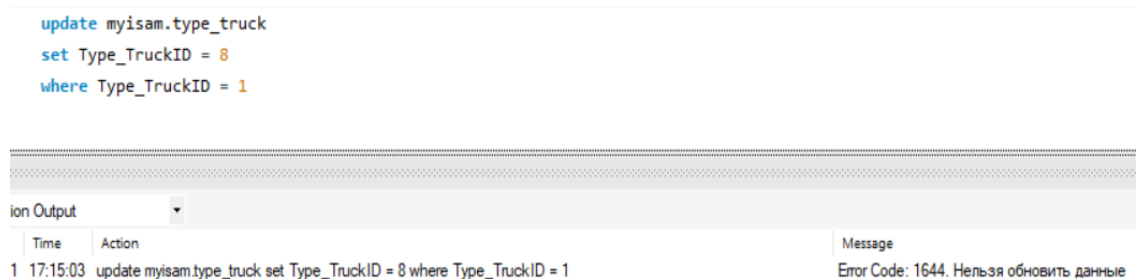


Рисунок 40 – Результат запиту Update

20) Тригер для таблиці тип_вантажівки типу Before Delete:

```

CREATE DEFINER=`root`@`localhost` TRIGGER `type_truck_BEFORE_DELETE` BEFORE DELETE ON `type_truck` FOR EACH ROW BEGIN
    if exists(select * from myisam.truck where fk_Type_TruckID = old.Type_TruckID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END

```

Рисунок 41 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:



Рисунок 42 – Результат запиту Delete

21) Тригер для таблиці тип_вантажівки типу Before Insert:

```
CREATE DEFINER='root'@'localhost' TRIGGER `truck_BEFORE_INSERT` BEFORE INSERT ON `truck` FOR EACH ROW BEGIN
    if not exists(select 1 from myisam.type_truck where Type_TruckID = new.fk_Type_TruckID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя вставить запись';
    end if;
END
```

Рисунок 43 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.truck
values(51,'AQDFET','70','30000','06.03.2020','358.9','18')
```

ion Output			Message
Time	Action		
1 17:35:36	insert into myisam.truck values(51,'AQDFET','70','30000','06.03.2020','358.9','18')		Error Code: 1644. Нельзя вставить запись

Рисунок 44 – Результат запиту Insert

22) Тригер для таблиці водій типу Before Update:

```
CREATE DEFINER='root'@'localhost' TRIGGER `driver_BEFORE_UPDATE` BEFORE UPDATE ON `driver` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.driver_truck where fk_Driver_ID = old.Driver_ID)
    and new.Driver_ID != old.Driver_ID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
end if;
```

Рисунок 45 - Код Sql-тригера

Після виконання запиту Update, маємо результат:

```
update myisam.driver
set Driver_ID = 100
where Driver_ID = 1
```

ion Output			Message
Time	Action		
1 17:43:24	update myisam.driver set Driver_ID = 100 where Driver_ID = 1		Error Code: 1644. Нельзя обновить данные

Рисунок 46 – Результат запиту Update

23) Тригер для таблиці водіїв типу Before Delete:

```
CREATE DEFINER='root'@'localhost' TRIGGER `driver_BEFORE_DELETE` BEFORE DELETE ON `driver` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.driver_truck where fk_Driver_ID = old.Driver_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
end if;
```

Рисунок 47 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:



Рисунок 48 – Результат запиту Delete

24) Тригер для таблиці водіїв типу Before Insert:

```
if not exists (select 1 from myisam.driver where Driver_ID = new.fk_Driver_ID)
then
    signal sqlstate '45000' set message_text = 'Нельзя вставить запись';
end if;
END
```

Рисунок 49 - Код Sql-тригера

Після виконання запити Insert, маємо результат:



Рисунок 50 – Результат запити Insert

25) Тригер для таблиці категорії типу Before Update:

```
CREATE DEFINER='root'@'localhost' TRIGGER `category_BEFORE_UPDATE` BEFORE UPDATE ON `category` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.category_driver where fk_Category_ID = old.Category_ID)
    and new.Category_ID != old.Category_ID
    then
        signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
    end if;
END
```

Рисунок 51 - Код Sql-тригера

Після виконання запити Update, маємо результат:

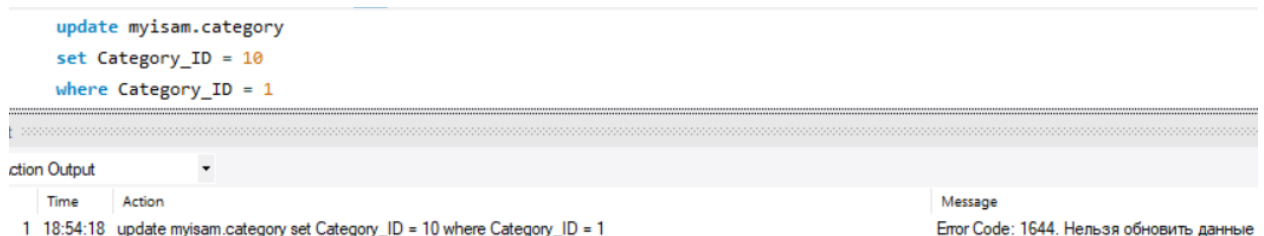


Рисунок 52 – Результат запити Update

26) Тригер для таблиці категорії типу Before Delete:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `category_BEFORE_DELETE` BEFORE DELETE ON `category` FOR EACH ROW BEGIN
    if exists (select 1 from myisam.category_driver where fk_Category_ID = old.Category_ID)
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 53 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:

```
delete from myisam.category
where Category_ID = 1
```



Time	Action	Message
18:57:25	delete from myisam.category where Category_ID = 1	Error Code: 1644. Нельзя удалить запись

Рисунок 54 – Результат запиту Delete

27) Тригер для таблиці категорій типу Before Insert:

```
if not exists(select 1 from myisam.category where Category_ID=new.fk_Category_ID)
then
    signal sqlstate '45000' set message_text = 'не правильно1';
end if;
```

Рисунок 55 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.category_driver
values(52,70,34)
```



Time	Action	Message
19:03:25	insert into myisam.category_driver values(52,70,34)	Error Code: 1644. Нельзя вставить запись

Рисунок 56 – Результат запису Insert

28) Тригер для таблиці водіїв типу Before Update:

```
if exists (select 1 from myisam.category_driver where fk_Driver_ID = old.Driver_ID
and new.Driver_ID != old.Driver_ID
then
    signal sqlstate '45000' set message_text = 'Нельзя обновить данные';
end if;
END
```

Рисунок 57 - Код Sql-тригера

Після виконання запиту Update, маємо результат:


```
Update myisam.driver
set Driver_ID = 100
where Driver_ID = 1
```

Time	Action	Message
1 19:08:10	Update myisam.driver set Driver_ID = 100 where Driver_ID = 1	Error Code: 1644. Нельзя обновить данные

Рисунок 58 – Результат запросу Update

29) Триггер для таблицы водіїв типу Before Delete:

```
if exists (select 1 from myisam.category_driver where fk_Driver_ID = old.Driver_ID)
then
    signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
end if;
END
```

Рисунок 59 - Код Sql-тригера

Після виконання запиту Delete, маємо результат:

```
delete from myisam.driver
where Driver_ID = 1
```

Time	Action	Message
1 19:10:34	delete from myisam.driver where Driver_ID = 1	Error Code: 1644. Нельзя удалить запись

Рисунок 60 – Результат запросу Delete

30) Триггер для таблиці водіїв типу Before Insert:

```
CREATE DEFINER='root'@'localhost' TRIGGER `category_driver_BEFORE_INSERT` BEFORE INSERT ON `category_driver` FOR EACH ROW BEGIN
if not exists(select 1 from myisam.driver where Driver_ID=new.fk_Driver_ID)
then
    signal sqlstate '45000' set message_text = 'Нельзя вставить запись';
end if;
```

Рисунок 61 - Код Sql-тригера

Після виконання запиту Insert, маємо результат:

```
insert into myisam.category_driver
values (52,2,80)
```

Time	Action	Message
1 19:15:09	insert into myisam.category_driver values (52,2,80)	Error Code: 1644. Нельзя вставить запись

Рисунок 62 – Результат запросу Insert

31) Триггер для таблиці замовлень типу Before Delete:

```
CREATE DEFINER='root'@'localhost' TRIGGER `order_BEFORE_DELETE` BEFORE DELETE ON `order` FOR EACH ROW BEGIN
    if exists(select 1 from myisam.`user` u where u.User_ID = old.fk_User_ID) or old.fk_User_ID is null
    then
        signal sqlstate '45000' set message_text = 'Нельзя удалить запись';
    end if;
END
```

Рисунок 63 – Код Sql-тригера

Після виконання запиту Delete, маємо результат:

```
delete from myisam.order
where Order_ID = 1
```

Action Output			
#	Time	Action	Message
1	19:19:02	delete from myisam.order where Order_ID = 1	Error Code: 1644. Нельзя удалить запись

Рисунок 64 – Результат запиту Delete

Висновок:

Набув практичних навичок з розробки тригерів для підтримки цілісності зв'язків, модифікації даних і забезпечення основних бізнес-процесів високонавантаженої інформаційної системи. Сформував необхідні практичні уміння для аналізу плану виконання SQL-запитів за допомогою оператора EXPLAIN. Сформував необхідні практичні уміння для створення тригерів, з урахуванням особливостей реалізації логіки роботи інтерфейсу високонавантаженої інформаційної системи зберігання даних.