



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Лабораторна робота №5

Бази Даних

Виконав
студент групи ІТ-03:

Чабан А.Є.

Перевірив:

Марченко О.І

Київ 2021

Мета:

- Вивчити правила побудови ідентифікаторів, правила визначення змінних та типів. Визначити правила роботи з циклами та умовними конструкціями, роботу зі змінними типу Table.
- Вивчити синтаксис та семантику функцій та збережених процедур, способів їх ідентифікації, методів визначення та специфікації параметрів та повертаємих значень, виклик функцій та збережених процедур.
- Застосування команд для створення, зміни та видалення як скалярних, так і табличних функцій, збережених процедур.
- Вивчити призначення та типи курсорів, синтаксис та семантику команд мови SQL для створення курсорів, вибірки даних з курсорів, зміни даних із застосуванням курсорів.
- Вивчити призначення та типи тригерів, умов їх активації, синтаксису та семантики для їх створення, модифікації, перейменування, програмування та видалення.

Теоретичні основи

Викладені в лекційному матеріалі

Постановка задачі комп'ютерного практикуму № 5

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

1) Збережені процедури:

- a. запит для створення тимчасової таблиці через змінну типу TABLE;
- b. запит з використанням умовної конструкції IF;
- c. запит з використанням циклу WHILE;
- d. створення процедури без параметрів;
- e. створення процедури з вхідним параметром;
- f. створення процедури з вхідним параметром та RETURN;
- g. створення процедури оновлення даних в деякій таблиці БД;
- h. створення процедури, в котрій робиться вибірка даних.

2) Функції:

- a. створити функцію, котра повертає деяке скалярне значення;
- b. створити функцію, котра повертає таблицю з динамічним набором стовпців;
- c. створити функцію, котра повертає таблицю заданої структури.

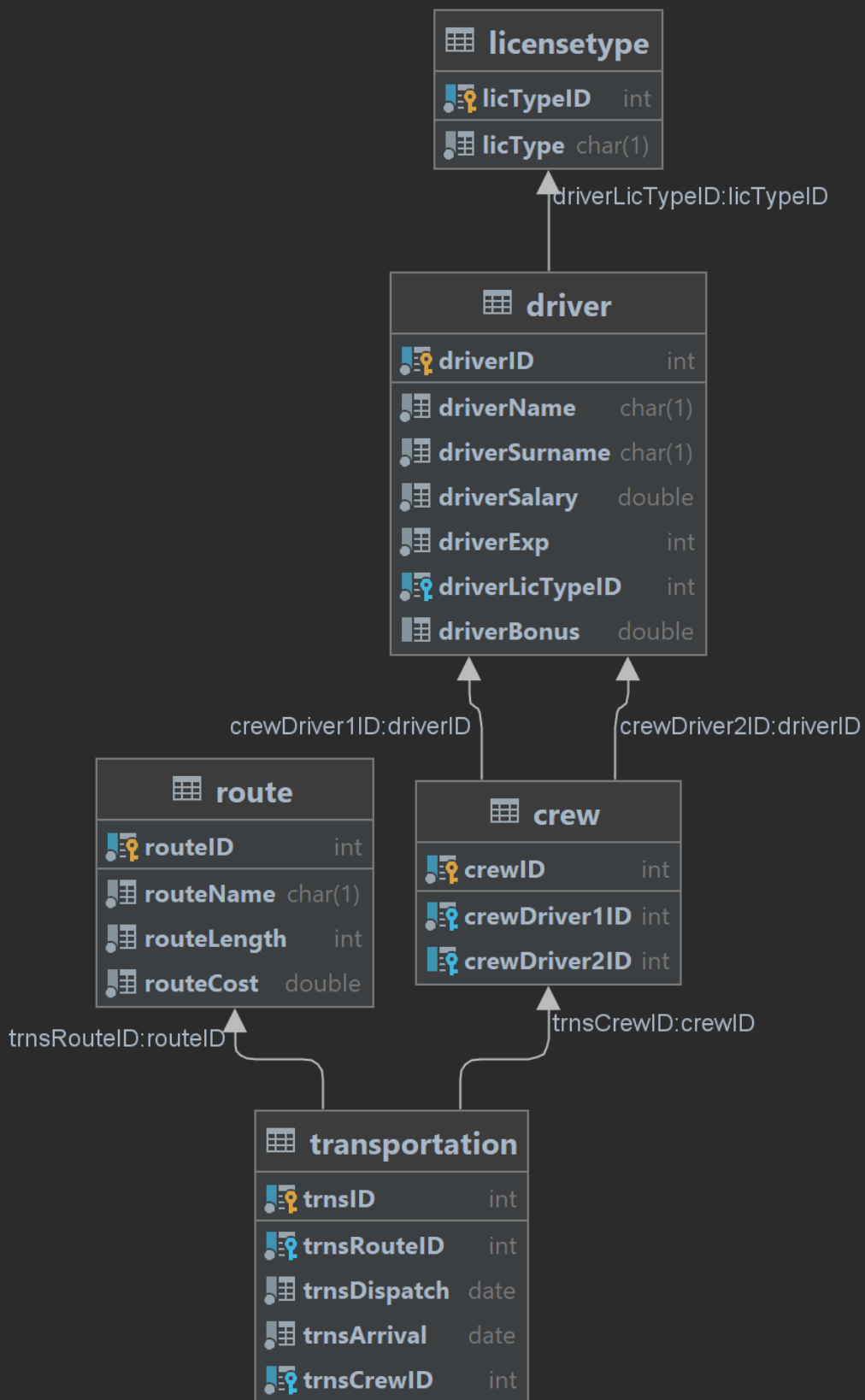
3) Робота з курсорами:

- a. створити курсор;
- b. відкрити курсор;
- c. вибірка даних, робота з курсорами.

4) Робота з тригерами:

- a. створити тригер, котрий буде спрацьовувати при видаленні даних;

- b. створити тригер, котрий буде спрацьовувати при модифікації даних;
- c. створити тригер, котрий буде спрацьовувати при додаванні даних.



Використані запити:

Lab 5

a

```
create procedure driversCount()
begin
    create temporary table TMPTable
    (
        NumOfDrivers int
    );
    insert into TMPTable select count(driverID) from driver;
    select * from TMPTable;
    drop table TMPTable;
end;

drop procedure driversCount;

call driversCount();
```

b

```
create procedure procIf(in var1 int)
begin
    if (var1 > 5) then
        select * from driver;
    else
        select * from route;
    end if;
end;

drop procedure procIf;

call procIf(3);
```

c

```
create procedure procWhile(in var1 int)
begin
    create temporary table TMPTableWhile
    (
        counter int
    );
    while var1 > 0
    do
        insert into TMPTableWhile values (var1);
        set var1 = var1 - 1;
    end while;
    select * from TMPTableWhile;
    drop table TMPTableWhile;
end;
```

```

drop procedure procWhile;

call procWhile(8);

# d

create procedure procNoParam()
begin
    select d.driverSurname as Driver1Surname, d2.driverSurname as
Driver2Surname, c.crewID
    from crew c
        inner join driver d on d.driverID = c.crewDriver1ID
        inner join driver d2 on d2.driverID = c.crewDriver2ID
    where c.crewDriver2ID is not null
    order by crewID;
end;

call procNoParam();

# e

create procedure procWithParam(in var1 int)
begin
    create temporary table TMPTable
    (
        counter int
    );
    insert into TMPTable values (var1);
    select * from TMPTable;
    drop table TMPTable;
end;

call procWithParam(3);

# f

create procedure procRet(in var1 int, var2 int, outerVal int)
begin
    set outerVal = var1 * var2;
    return outerVal;
end;

call procRet(3, 4, @outerVal);

select @outerVal;

# g

create procedure procInsert(in licType char(1))
begin
    INSERT INTO licensetype (licType) VALUES (licType);
    select * from licensetype;
end;

```

```

drop procedure procInsert;

call procInsert('H');

# h

create procedure procSelect()
begin
    select trnsDispatch,
           trnsArrival,
           (select concat(d.driverName, ' ', d.driverSurname)) as
Dr1FullName,
           (select concat(d2.driverName, ' ', d2.driverSurname)) as
Dr2FullName,
           tr.trnsCrewID
    from transportation tr
         inner join crew c on tr.trnsCrewID = c.crewID
         inner join driver d on d.driverID = c.crewDriver1ID
         inner join driver d2 on d2.driverID = c.crewDriver2ID
    where c.crewDriver2ID is not null;
end;

call procSelect();

# Functions

create function func1(var1 int)
returns int
begin
    declare returnVar int;
    set returnVar = var1;
    return returnVar;
end;

select func1(10);

# Cursors

create procedure cursorTest()
begin
    DECLARE done INT DEFAULT FALSE;
    DECLARE a CHAR(30);
    DECLARE b, c INT;
    DECLARE cur1 CURSOR FOR SELECT driverSurname, driverSalary, driverExp
FROM driver;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    create temporary table TMPTable
    (
        surname char(30),
        salary int,
        exp int
    );

```

```

OPEN curl;

read_loop:
LOOP
    FETCH curl INTO a, b, c;
    IF done THEN
        LEAVE read_loop;
    END IF;
    insert into TMPTable values (a,b,c);
END LOOP;

CLOSE curl;
select * from TMPTable;
drop table TMPTable;
end;

call cursorTest();

# Triggers

# On insert

CREATE TRIGGER insertLog AFTER INSERT ON licensetype
FOR EACH ROW BEGIN
    INSERT INTO logtable (logNewText, logevent,logTime) VALUES
(new.licType, 'INSERT',current_timestamp);
END;

# On delete

CREATE TRIGGER deleteLog BEFORE DELETE ON licensetype
FOR EACH ROW BEGIN
    INSERT INTO logtable (logOldText, logevent,logTime) VALUES
(old.licType, 'DELETE',current_timestamp);
END;

delete from licensetype where licType = 'G';

# On update

CREATE TRIGGER updateLog BEFORE UPDATE ON licensetype
FOR EACH ROW BEGIN
    INSERT INTO logtable (logNewText, logOldText, logevent,logTime) VALUES
(NEW.licType, OLD.licType , 'UPDATE',current_timestamp);
END;

drop trigger updateLog;

UPDATE licensetype t SET t.licType = 'F' WHERE t.licTypeID = 6;

show triggers;

```


Висновок: Під час виконання даної лабораторної роботи я навчився основам програмування з використанням мови SQL, зберігати процедури, використовувати курсори та створювати тригери. Та відточив ці навички на практиці.