



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 №4

Бази Даних

Виконав
студент групи ІТ-03:

Чабан А.Є.

Перевірив:

Марченко О.І

Київ 2021

Мета:

- Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць, групування та сортування даних
- Навчитись використовувати вбудовані функції в запитах
- Вивчити призначення представлень (view) баз даних, синтаксису та семантики команд SQL для їх створення, зміни та видалення, системних збережених процедур для отримання інформації про представлення

Теоретичні основи

Викладені в лекційному матеріалі

Постановка задачі комп'ютерного практикуму № 4

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

1) Створити наступні запити:

- запит з використанням функції COUNT;
- запит з використанням функції SUM;
- запит з використанням функцій UPPER, LOWER;
- запит з використанням функцій для роботи з датами;
- запит з використанням групування по одному стовпцю;
- запит з використанням групування по декільком стовпцям;
- запит з використанням умови відбору груп HAVING;
- запит з використанням HAVING без GROUP BY;
- запит з використанням сортування по стовпцю.

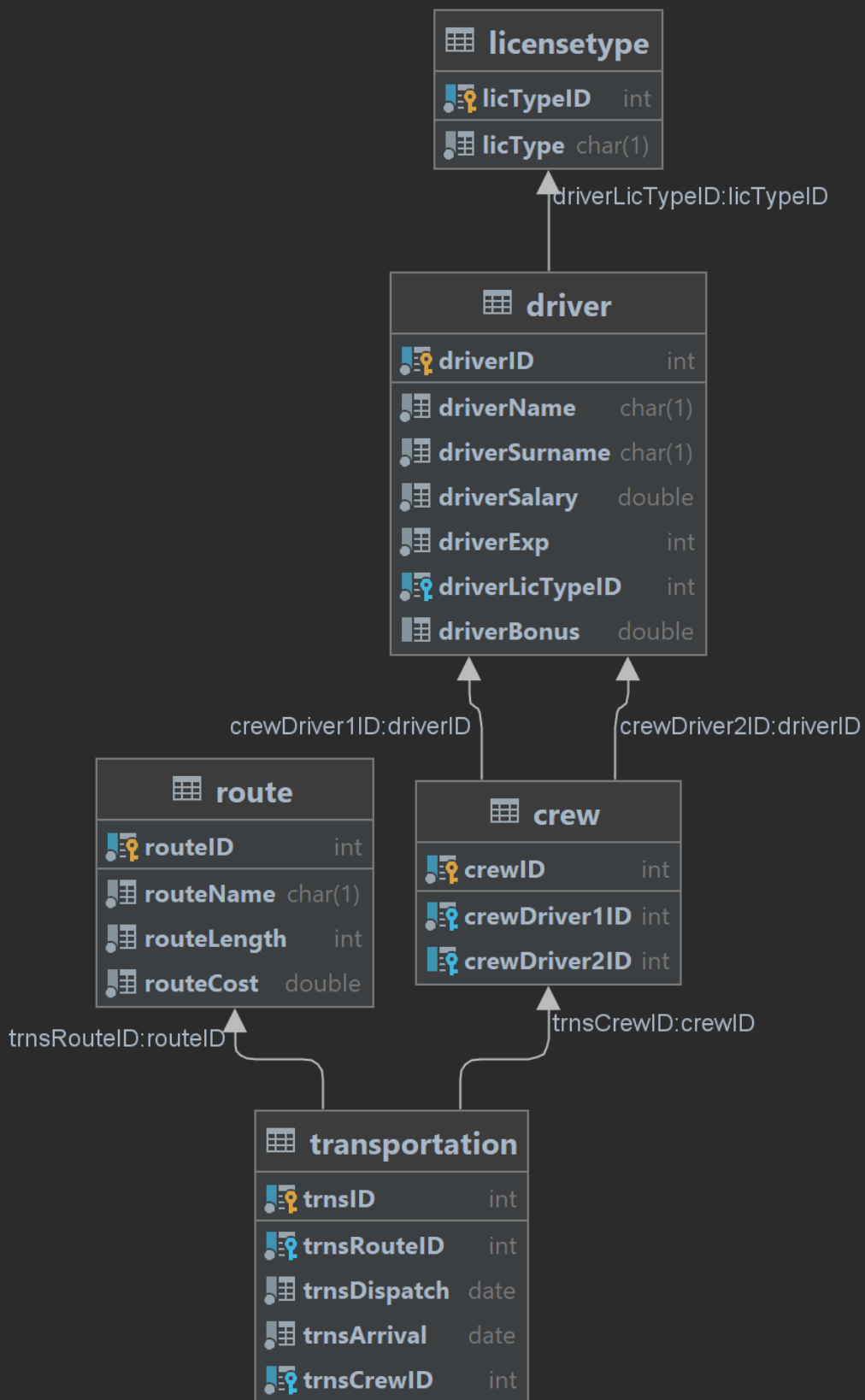
2) Робота з представленнями (view):

- створити представлення, котре містить дані з декількох таблиць;
- створити представлення, котре містить дані з декількох таблиць та посилання, котре створене в п.а;
- модифікувати запити з використанням команди ALTER VIEW;
- отримати довідникову інформацію про ці представлення з використанням вбудованих процедур (наприклад в MsSQL sp_help, sp_helptext и sp_depends)

Завдання:

Вантажні перевезення. Виконується розробка інформаційної системи для компанії, що займається перевезеннями вантажів.. Однією з основних задач є відстеження вартості перевезень з урахуванням заробітної плати водіїв. Компанія здійснює перевезення по різних маршрутах. Для кожного маршруту визначається деяка назва, обчислюється ідеальна відстань (мається на увазі оптимальна відстань між двома пунктами з урахуванням корок, якості покриття, тощо. Пофакту, відстань може бути інша) і встановлюється деяка оплата для водія. Інформація про водіїв включає прізвище, ім'я, по батькові, стаж, наявні категорії прав. Для проведення розрахунків необхідна повна інформація

про перевезення (маршрут, водій, дати відправки і прибуття). За фактом деяких перевезень водіям виплачується премія. Оплата водіям залежить не лише від маршруту, але і від стажу. Крім того, треба врахувати, що перевезення можуть здійснювати два водії.



Використані запити:

```
# a
select count(driverID)
from driver;

# b
select sum(driverSalary) as TotalSalary
from driver;

# c
select upper(driverSurname) as UpperSurname
from driver;

select lower(driverName) as LowerName
from driver;

# d
select now();

select *
from transportation
where trnsArrival = utc_date;

# e

select crewID, count(crewDriver2ID)
from crew
group by crewID;

# f

select driverSalary, driverExp, count(*) as C
from driver
group by driverSalary, driverExp;

# g

select driverExp, count(driverExp)
from driver
group by driverExp
having count(driverExp) >= 3;

# h
select 1 as OldDriver
from driver
having max(driverExp) > 10;

# i

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;

# Part 2

# a

create view view1 as
select trnsDispatch,
       trnsArrival,
       (select concat(d.driverName, ' ', d.driverSurname)) as Dr1FullName,
       (select concat(d2.driverName, ' ', d2.driverSurname)) as Dr2FullName
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;

select *
from view1;

# b

create view view2 as
select trnsID, r.startPoint, r.destPoint, v.trnsDispatch, v.trnsArrival,
       v.Dr1FullName, v.Dr2FullName
from transportation tr
     inner join route r on tr.trnsRouteID = r.routeID
     inner join view1 v on tr.trnsDispatch = v.trnsDispatch and
tr.trnsArrival = v.trnsArrival;

select *
from view2;

# c

alter view view1 as
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;

# backup

```

```
alter view view1 as
    select trnsDispatch,
           trnsArrival,
           (select concat(d.driverName, ' ', d.driverSurname)) as Dr1FullName,
           (select concat(d2.driverName, ' ', d2.driverSurname)) as Dr2FullName
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;

# d

show columns in view1;

show create view view1;
```

Висновок: Під час виконання даної роботи я ознайомився створенням запитів на групування, сортування, використання вбудованих функцій, а також створенням та керуванням представленнями. Та навчився реалізовувати їх на практиці