



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 №3

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
- Вивчити команди SQL для створення запитів з використанням підзапитів та з'єднань

Теоретичні основи

Викладені в лекційному матеріалі

Постановка задачі комп'ютерного практикуму № 3

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

1) Створити запити для вибірки даних з використанням (разом 15 запитів):

- Найпростіших умов
- Операторів порівняння
- Умов з використанням логічних операторів AND, OR та NOT.
- Умов з використанням комбінацій логічних операторів
- З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах
- Використання операторів:
 - Приналежності множині
 - Приналежності діапазону
 - Відповідності шаблону
 - Відповідності регулярному виразу
 - Перевірка на невизначене значення

2) Створити запити з використанням підзапитів та з'єднань (разом 15 запитів) (в запити повинні реалізовуватись до 3 та більше таблиць):

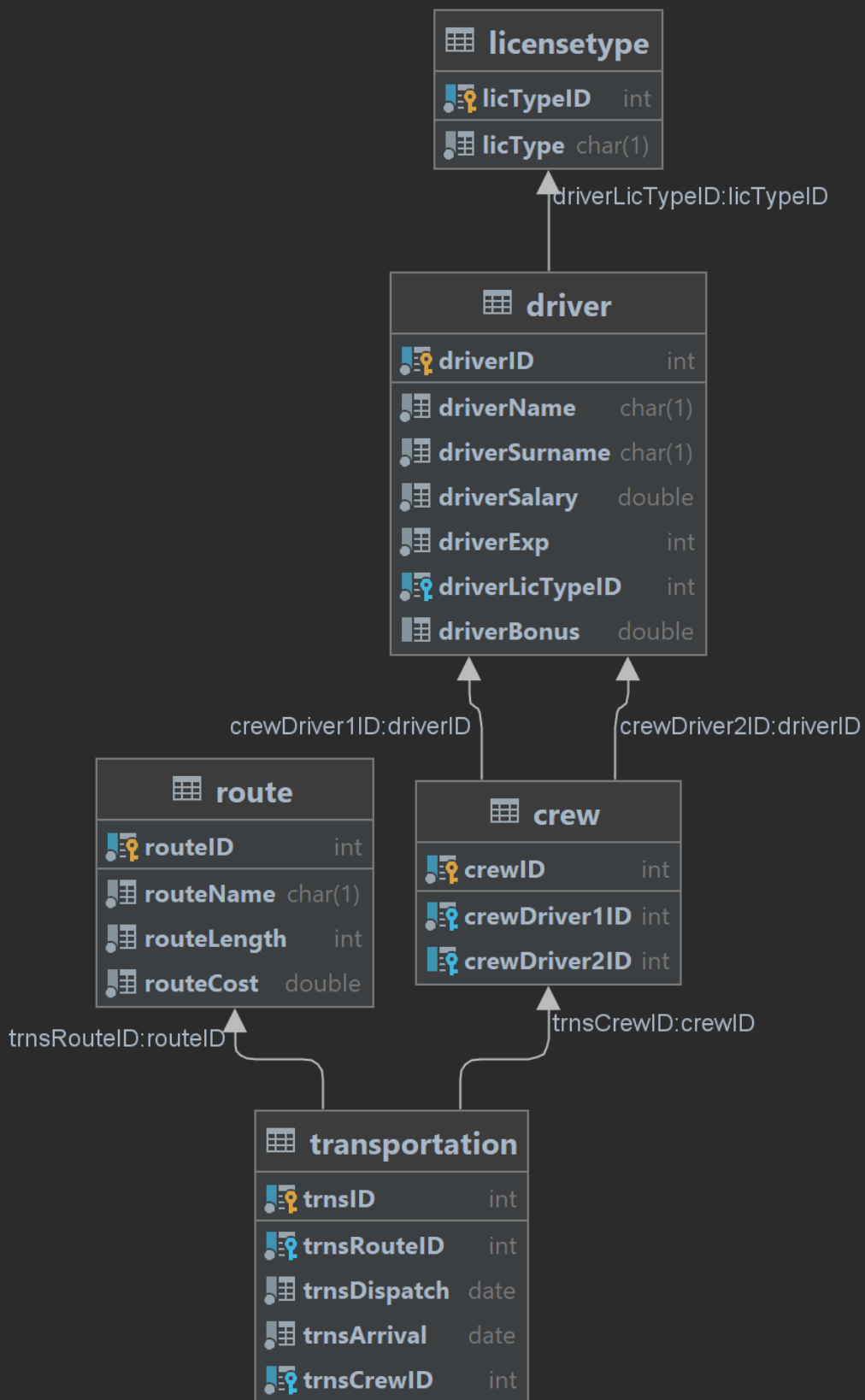
- Використання підзапитів в рядку вибірки полів та вибірки з таблиць
- Використання підзапитів в умовах з конструкціями EXISTS, IN
- Декартовий добуток
- З'єднання декількох таблиць (більше 2) за рівністю
- З'єднання декількох таблиць (більше 2) за рівністю та умовою відбору
- Внутрішнього з'єднання
- Лівого зовнішнього з'єднання
- Правого зовнішнього з'єднання
- Об'єднання таблиць

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:

1

```
select driverName,driverSurname
from driver
where driverSurname = 'Chaban';
```

2

```
select driverName,driverSurname,driverSalary
from driver
where driverSalary > 10000;
```

3

```
with seniors as (
    select avg(driverExp) as avgexp
    from driver
)
select driverName, driverSurname, driverExp
from driver
where driverExp > (select avgexp from seniors);
```

4

```
select driverName,driverSurname,driverExp
from driver
where driverExp < 5;
```

5

```
select driverName,driverSurname
from driver
where driverName LIKE 'A%';
```

6

```
select startPoint, destPoint
from route
where routeLength > 2000
and routeCost > 6000
and not startPoint like '%i%';
```

7

```
select *
from crew
where crewDriver1ID && crew.crewDriver2ID is not null;
```

8

```
select *
from crew
where crewDriver1ID || crew.crewDriver2ID is null;
```

9

```
select avg(driverSalary) as AverageSalary, max(driverExp) as Oldest
from driver;
```

10

```
select concat(driverName, ' ', driverSurname) as FullName
from driver;
```

11

```
select dayofweek(trnsDispatch) as ArriveDay,dayofweek(trnsArrival) as
ArriveDay,r.startPoint,r.destPoint
```

```

from transportation
inner join route r on transportation.trnsRouteID = r.routeID;
# 12
select trnsDispatch, trnsArrival
from transportation
where trnsDispatch between '2021-10-23' and '2021-11-23';
# 13
select startPoint, destPoint, routeLength
from route
where startPoint in ('Kyiv', 'Kherson', 'Krakow');
# 14
select startPoint, destPoint
from route
where destPoint like 'M%';
# 15
select driverName, driverSurname
from driver
where driverSurname regexp '^C|a';
# 16
select crewDriver1ID, crewDriver2ID
from crew
where crewDriver2ID is not null;

# Part 2
# 1
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;

# 2
select driverName, driverSurname, l.licType
from driver d
inner join licensetype l on d.driverLicTypeID = l.licTypeID
where licType = 'A' or licType = 'C';

# 3
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;

# 4
select *
from driver d

```

```

where not exists(select * from crew where crewDriver1ID = d.driverID or
crewDriver2ID = d.driverID);

# 5
select *
from driver d
where driverID not in (select crewDriver1ID from crew union select
crewDriver2ID from crew where crewDriver2ID is not null );

# 6
select driverSurname, licType
from driver d cross join licensetype lt;

# 7
select trnsID,tr.trnsCrewID
from transportation tr, crew c
where (tr.trnsCrewID = c.crewID) and (crewDriver2ID is not null );

# 8
select trnsID,tr.trnsCrewID,d.driverSurname as Driver1Sur, d2.driverSurname
as D2Sur
from transportation tr, crew c
inner join driver d on d.driverID = c.crewDriver1ID
inner join driver d2 on d2.driverID = c.crewDriver2ID
where (tr.trnsCrewID = c.crewID) and (crewDriver2ID is not null ) and
d.driverSurname like '%a%';

# 9
select
r.startPoint,r.destPoint,c.crewID,d.driverSurname,d2.driverSurname,tr.trnsDis
patch,trnsArrival
from transportation tr
inner join route r on tr.trnsRouteID = r.routeID
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;

# 10
select
r.startPoint,r.destPoint,c.crewID,d.driverSurname,d2.driverSurname,tr.trnsDis
patch,trnsArrival
from transportation tr
left join route r on tr.trnsRouteID = r.routeID
left join crew c on tr.trnsCrewID = c.crewID
left join driver d on d.driverID = c.crewDriver1ID
left join driver d2 on d2.driverID = c.crewDriver2ID;

# 11
select
r.startPoint,r.destPoint,c.crewID,d.driverSurname,d2.driverSurname,tr.trnsDis
patch,trnsArrival
from transportation tr
right join route r on tr.trnsRouteID = r.routeID

```

```
right join crew c on tr.trnsCrewID = c.crewID
right join driver d on d.driverID = c.crewDriver1ID
right join driver d2 on d2.driverID = c.crewDriver2ID;
```

```
# 12
select driverSurname
from driver
union
select licType
from licensetype;
```

Висновок: Під час виконання даної роботи я ознайомився з основними поняттями баз даних, попрактикувався в написанні простих та складних запитів та об'єднаннях таблиць.