

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет «Институт Международного Развития и Партнёрства»

Направление подготовки «45.03.04 Интеллектуальные системы в гуманитарной сфере»

ОТЧЁТ

ЛАБОРАТОРНАЯ РАБОТА №3

Тема задания: РЕАЛИЗАЦИЯ SQL- ЗАПРОСОВ К БАЗЕ ДАННЫХ

Выполнил:

Студент Муркин А. А. D41421

Проверил:

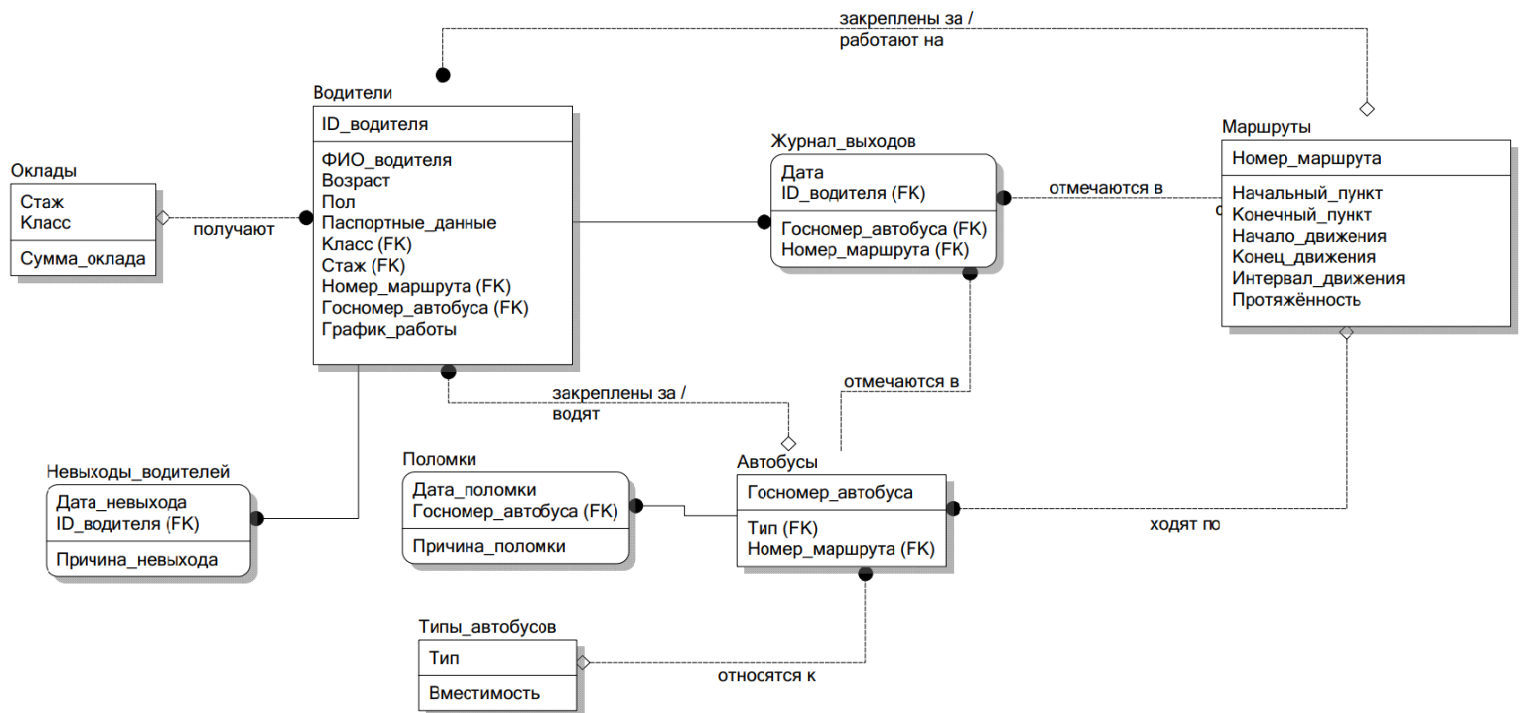
Преподаватель Говоров А.И.

Санкт-Петербург

2020

1. Скриншот модели БД и индивидуальное задание

1.1. Скриншот модели БД



1.2. Индивидуальное задание:

Создать программную систему, предназначенную для диспетчера автобусного парка частной транспортной фирмы. Фирма обслуживает несколько коммерческих маршрутов. Такая система должна обеспечивать хранение сведений о водителях, о маршрутах и характеристиках автобусов.

Каждый водитель характеризуется паспортными данными, классом, стажем работы и окладом, причем оклад зависит от класса и стажа работы. Маршрут автобуса характеризуется номером маршрута, названием начального и конечного пункта движения, временем начала и конца движения, интервалом движения и протяженностью в минутах (время движения от кольца до кольца). Характеристиками автобуса являются: номер государственной регистрации автобуса, его тип и вместимость, причем вместимость автобуса зависит от его типа.

Каждый водитель закреплен за определенным автобусом и работает на определенном маршруте, но в случае поломки своего автобуса или болезни другого водителя может пересесть на другую машину.

В базе должен храниться график работы водителей.

Необходимо предусмотреть возможность корректировки БД в случаях поступления на работу нового водителя, списания старого автобуса, введения нового маршрута или изменения старого и т.п.

2. Список запросов

2.1. Запрос выводит суммарную протяжённость всех маршрутов, обслуживаемых каждой водительницей (то есть выведены только женщины)

```
select driver_name, sum(route_length) sum_length
from bus_bd_schema.routes
```

```

inner join bus_bd_schema.drivers on bus_bd_schema.routes.route_number =
bus_bd_schema.drivers.route_number
where bus_bd_schema.drivers.driver_sex = 'f'
group by bus_bd_schema.drivers.driver_id

```

bus_park/postgres@PostgreSQL 12

Query Editor

Query History

```
1 select driver_name, sum(route_length) sum_length
2 from bus_bd_schema.routes
3 inner join bus_bd_schema.drivers on bus_bd_schema.routes.route_number = bus_bd_schema.drivers.route_number
4 where bus_bd_schema.drivers.driver_sex = 'f'
5 group by bus_bd_schema.drivers.driver_id
```

Data Output

Explain

Messages

Notifications


	driver_name character varying (60)	sum_length bigint
1	Трофимова Ольга Игоревна	31

2.2. Запрос выводит госномера автобусов, которые не ломались ни разу.

```

select license_plate
from bus_bd_schema.buses
where not exists
(select * from bus_bd_schema.malfunctions
where bus_bd_schema.buses.license_plate = bus_bd_schema.malfunctions.license_plate)

```

 bus_park/postgres@PostgreSQL 12

Query Editor

Query History

1

select license_plate

2

from bus_bd_schema.buses

3

where not exists

4

(select * from bus_bd_schema.malfunctions

5

where bus_bd_schema.buses.license_plate = bus_bd_schema.malfunctions.license_plate)

Data Output


Explain

Messages

Notifications

license_plate

[PK] character (8)



1

о741ии78

2.3. Запрос выводит информацию о водителях, которые не выходили на работу по личным причинам (то есть не из-за поломки автобуса) больше одного раза и тех, кто ни разу не пропускал смены

```
select driver_name, count(absence_date) from bus_bd_schema.drivers
left join bus_bd_schema.absences on bus_bd_schema.drivers.driver_id =
bus_bd_schema.absences.driver_id
group by bus_bd_schema.drivers.driver_id
having count(absence_date) > 1
union
select driver_name, count(absence_date) from bus_bd_schema.drivers
left join bus_bd_schema.absences on bus_bd_schema.drivers.driver_id =
bus_bd_schema.absences.driver_id
group by bus_bd_schema.drivers.driver_id
having count(absence_date) < 1
```

bus_park/postgres@PostgreSQL 12

Query Editor Query History


```
1 select driver_name, count(absence_date) from bus_bd_schema.drivers
2 left join bus_bd_schema.absences on bus_bd_schema.drivers.driver_id = bus_bd_schema.absences.driver_id
3 group by bus_bd_schema.drivers.driver_id
4 having count(absence_date) > 1
5 union
6 select driver_name, count(absence_date) from bus_bd_schema.drivers
7 left join bus_bd_schema.absences on bus_bd_schema.drivers.driver_id = bus_bd_schema.absences.driver_id
8 group by bus_bd_schema.drivers.driver_id
9 having count(absence_date) < 1
```

Data Output Explain Messages Notifications

	driver_name character varying (60)	count bigint
1	Трофимова Ольга Игоревна	0
2	Сергеев Иван Владимиров...	2

2.4. Запрос выводит имена водителей, чья фамилия начинается на «И»

```
select driver_name from bus_bd_schema.drivers where driver_name like 'И%'
```


bus_park/postgres@PostgreSQL 12

Query Editor
Query History

1
select driver_name from bus_bd_schema.drivers where driver_name like 'И%'

Data Output
Explain
Messages
Notifications

	driver_name character varying (60)	
1	Иванов Максим Петрович	

2.5. Этот запрос выводит информацию обо всех невыходах водителей в смену по личным причинам и из-за поломок с 28.10 по 10.11

```

select bus_bd_schema.drivers.driver_name, malfunction_date D, malfunction_cause from
bus_bd_schema.malfunctions
inner join bus_bd_schema.buses on bus_bd_schema.malfunctions.license_plate =
bus_bd_schema.buses.license_plate
inner join bus_bd_schema.drivers on bus_bd_schema.buses.license_plate =
bus_bd_schema.drivers.license_plate
where malfunction_date between '2019-10-28' and '2019-11-10'
union
select bus_bd_schema.drivers.driver_name, absence_date D, absence_cause from
bus_bd_schema.absences
inner join bus_bd_schema.drivers on bus_bd_schema.absences.driver_id =
bus_bd_schema.drivers.driver_id
where absence_date between '2019-10-28' and '2019-11-10'
order by D desc

```

```

1 select bus_bd_schema.drivers.driver_name, malfunction_date D, malfunction_cause from bus_bd_schema.malfunctions
2 inner join bus_bd_schema.buses on bus_bd_schema.malfunctions.license_plate = bus_bd_schema.buses.license_plate
3 inner join bus_bd_schema.drivers on bus_bd_schema.buses.license_plate = bus_bd_schema.drivers.license_plate
4 where malfunction_date between '2019-10-28' and '2019-11-10'
5 union
6 select bus_bd_schema.drivers.driver_name, absence_date D, absence_cause from bus_bd_schema.absences
7 inner join bus_bd_schema.drivers on bus_bd_schema.absences.driver_id = bus_bd_schema.drivers.driver_id
8 where absence_date between '2019-10-28' and '2019-11-10'
9 order by D desc

```

	driver_name character varying (60)	d date	malfunction_cause character (20)
1	Сергеев Иван Владимирович	2019-11-10	поломка генератора
2	Петров Николай Николаевич	2019-11-06	прогул
3	Сергеев Иван Владимирович	2019-11-04	болезнь
4	Иванов Максим Петрович	2019-10-28	поломка аккумулятора

2.6. Этот запрос выводит информацию о водителях и их зарплатах для водителей которым платят больше 30000

```

select driver_name, bus_bd_schema.wages.wage from bus_bd_schema.drivers
left join bus_bd_schema.wages on bus_bd_schema.drivers.rank = bus_bd_schema.wages.rank
and bus_bd_schema.drivers.experience = bus_bd_schema.wages.experience
where bus_bd_schema.wages.wage = some (select wage from bus_bd_schema.wages where
wage > 30000)
order by wage desc

```

```

1 select driver_name, bus_bd_schema.wages.wage from bus_bd_schema.drivers
2 left join bus_bd_schema.wages on bus_bd_schema.drivers.rank = bus_bd_schema.wages.rank and bus_bd_schema.drivers.experience = bus_bd_schema.wages.experience
3 where bus_bd_schema.wages.wage = some (select wage from bus_bd_schema.wages where wage > 30000)
4 order by wage desc

```

	driver_name character varying (60)	wage integer
1	Петров Николай Николаев...	50000
2	Сергеев Иван Владимиров...	40000
3	Фёдоров Роман Алексеевич	40000

2.7. Этот запрос выводит информацию о доходах водителей в следующем виде: доход выше 40000 считается высоким, ниже 40000 – средним.

```
select driver_name,  
case  
when bus_bd_schema.wages.wage > 40000 then 'Высокий доход' else 'Средний доход'  
end as wage  
from bus_bd_schema.drivers  
left join bus_bd_schema.wages on bus_bd_schema.drivers.rank = bus_bd_schema.wages.rank  
and bus_bd_schema.drivers.experience = bus_bd_schema.wages.experience
```

bus_park/postgres@PostgreSQL 12

Query EditorQuery History

```
1 select driver_name,  
2 case  
3 when bus_bd_schema.wages.wage > 40000 then 'Высокий доход' else 'Средний доход' end as wage  
4 from bus_bd_schema.drivers  
5 left join bus_bd_schema.wages on bus_bd_schema.drivers.rank = bus_bd_schema.wages.rank and bus_bd_schema.drivers.experience = bus_bd_schema.wages.experience
```

Data OutputExplainMessagesNotifications

	driver_name character varying (60)	wage text
1	Сергеев Иван Владимирович	Средний доход
2	Фёдоров Роман Алексеевич	Средний доход
3	Петров Николай Николаевич	Высокий доход
4	Иванов Максим Петрович	Средний доход
5	Трофимова Ольга Игоревна	Средний доход

2.8. Этот запрос выводит информацию о средней протяжённости маршрутов, по которым ходят двухэтажные автобусы и автобусы «с гармошкой»

```
select bus_bd_schema.bus_types.bus_type, round(avg(route_length))  
from bus_bd_schema.bus_types left join bus_bd_schema.buses on  
bus_bd_schema.bus_types.bus_type = bus_bd_schema.buses.bus_type  
left join bus_bd_schema.routes on bus_bd_schema.routes.route_number =  
bus_bd_schema.buses.route_number  
where bus_bd_schema.bus_types.bus_type in ('doubledeck', 'middlebend')  
group by bus_bd_schema.bus_types.bus_type
```

bus_park/postgres@PostgreSQL 12

Query Editor
Query History

```

1 select bus_bd_schema.bus_types.bus_type, round(avg(route_length))
2 from bus_bd_schema.bus_types left join bus_bd_schema.buses on bus_bd_schema.bus_types.bus_type = bus_bd_schema.buses.bus_type
3 left join bus_bd_schema.routes on bus_bd_schema.routes.route_number = bus_bd_schema.buses.route_number
4 where bus_bd_schema.bus_types.bus_type in ('doubledeck', 'middlebend')
5 group by bus_bd_schema.bus_types.bus_type

```

Data Output
Explain
Messages
Notifications

	bus_type [PK] character (10)	round numeric
1	doubledeck	20
2	middlebend	15

2.9. Этот запрос выводит информацию о средней вместимости автобусов, ходящих по каждому маршруту

```

select route_number, round(avg(capacity)) avg_cap
from bus_bd_schema.buses inner join bus_bd_schema.bus_types
on bus_bd_schema.buses.bus_type = bus_bd_schema.bus_types.bus_type
group by route_number
order by avg_cap desc

```

bus_park/postgres@PostgreSQL 12

Query Editor
Query History

```

1 select route_number, round(avg(capacity)) avg_cap
2 from bus_bd_schema.buses inner join bus_bd_schema.bus_types
3 on bus_bd_schema.buses.bus_type = bus_bd_schema.bus_types.bus_type
4 group by route_number
5 order by avg_cap desc

```

Data Output
Explain
Messages
Notifications

	route_number integer	avg_cap numeric
1	7	80
2	12	60
3	15	40
4	31	40
5	124	20

2.10. Этот запрос выводит число случаев каждого типа поломок, случившееся с каждым типом автобуса.

```
select bus_bd_schema.buses.bus_type, bus_bd_schema.malfunctions.malfunction_cause,  
count(bus_bd_schema.malfunctions.malfunction_cause) C  
from bus_bd_schema.buses  
left join bus_bd_schema.malfunctions on bus_bd_schema.buses.bus_type =  
bus_bd_schema.malfunctions.malfunction_cause  
left join bus_bd_schema.malfunctions on bus_bd_schema.malfunctions.license_plate =  
bus_bd_schema.buses.license_plate  
group by bus_bd_schema.buses.bus_type,  
bus_bd_schema.malfunctions.malfunction_cause  
order by C desc
```

bus_park/postgres@PostgreSQL 12			
Query Editor Query History			
<pre>1 select bus_bd_schema.buses.bus_type, bus_bd_schema.malfunctions.malfunction_cause, count(bus_bd_schema.malfunctions.malfunction_cause) C 2 from bus_bd_schema.buses 3 left join bus_bd_schema.malfunctions on bus_bd_schema.buses.bus_type = bus_bd_schema.malfunctions.malfunction_cause 4 left join bus_bd_schema.malfunctions on bus_bd_schema.malfunctions.license_plate = bus_bd_schema.buses.license_plate 5 group by bus_bd_schema.buses.bus_type, bus_bd_schema.malfunctions.malfunction_cause 6 order by C desc</pre>			
Data Output Explain Messages Notifications			
bus_type	malfunction_cause	c	
character (10)	character (20)	bigint	
1 minibus	поломка аккумулятора	2	
2 trolleybus	поломка стартера	1	
3 doubledeck	поломка генератора	1	
4 singledeck	поломка аккумулятора	1	
5 middlebend	[null]	0	