



**Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики**

Лабораторная работа № 3 SQL-запросы

**Выполнила: Попова Анна Эдуардовна
Группа D41421
Проверил: Говоров Антон Игоревич**

Санкт-Петербург 2020

Цель работы:

Овладеть практическими навыками написания sql-запросов.

Ход работы:

База данных и её описание:

Учёт горюче-мазочных средств.

Описание предметной области:

Предприятие имеет несколько автобаз. На каждой автобазе (код, название, адрес) ведется учет горюче-смазочных средств (ГСМ), заправляемых в автомобили, выполняющих рейсы по соответствующим путевым листам. В путевом листе отражается информация о рейсе: пункты погрузки и разгрузки, пробег общий и с грузом, наименования грузоотправителя и грузополучателя, время в наряде (в днях и/или часах). При заправке автомобилей в гараже формируется раздаточная ведомость, в которой указаны: номер ведомости, дата. В одной ведомости могут быть оформлены данные на нескольких водителей. В каждой позиции ведомости записывается: марка автомобиля, государственный регистрационный номер автомобиля, номер путевого листа, фамилия, инициалы водителя, количество заправленного ГСМ, в литрах и килограммах. ГСМ – это бензин, дизтопливо, дизмасло, автол, солидол, нигрол и т.п. Для каждого автомобиля на одну поездку может быть выделено несколько видов ГСМ. Ведомость подписывает сотрудник, имеющий должность заправщика. Указываются его ФИО. В заголовке ведомости указывается автобаза, которой принадлежит автомобиль и гараж. У каждой автобазы может быть несколько гаражей, расположенных по различным адресам. Ведомости формируются отдельно для каждого гаража заправщиком гаража.

Перечень возможных запросов:

Для каждой автобазы указать количество закрепленных за ней автомобилей.

Вывести список водителей, заправлявшихся в те же дни, что и заданный водитель.

Для каждого рейса вывести общий объем топлива в литрах и килограммах в заданный промежуток времени.

Вывести общий объем отпущенного топлива на предприятии по каждому виду в заданный промежуток времени.

Для заданного водителя вывести информацию о всех его заправках указанием номеров путевых листов и общим объемом отпущенных ГСМ.

Необходимо предусмотреть возможность получения отчета, в котором для каждой автобазы указывается список раздаточных ведомостей с указанием количества автомобилей, количества и стоимости отпущенного топлива по каждому типу топлива и автобазе, общей стоимости топлива по предприятию.

Состав реквизитов сущностей:

Автобаза (код автобазы, имя, адрес);

Гараж (ID_гаража, код (FK), адрес);

Автомобиль (Гос. Рег. Номер, ID_гаража, код автобазы, марка автомобиля);

ГСМ (ID_гсм, тип, стоимость за литр, стоимость за килограмм); Путевой лист (номер, пункт погрузки, пункт разгрузки, общий пробег, пробег с грузом, имя грузоотправителя, имя грузополучателя, время в наряде);

Водитель (ID, ФИО);

Заправщик (ID, ФИО);

Раздаточная ведомость (ID, номер путевого листа, дата, номер ведомости, количество ГСМ в литрах, количество ГСМ в кг, ID_гаража, код автобазы, гос. рег. номер автомобиля, ID_гсм); Водитель_ведомость (ID_водителя, ID_ведомости);

Ведомость_гараж (ID_водителя, ID_ведомости, ID_заправщика, код автобазы).

Схема сущность-связь в нотации Питера Чена:

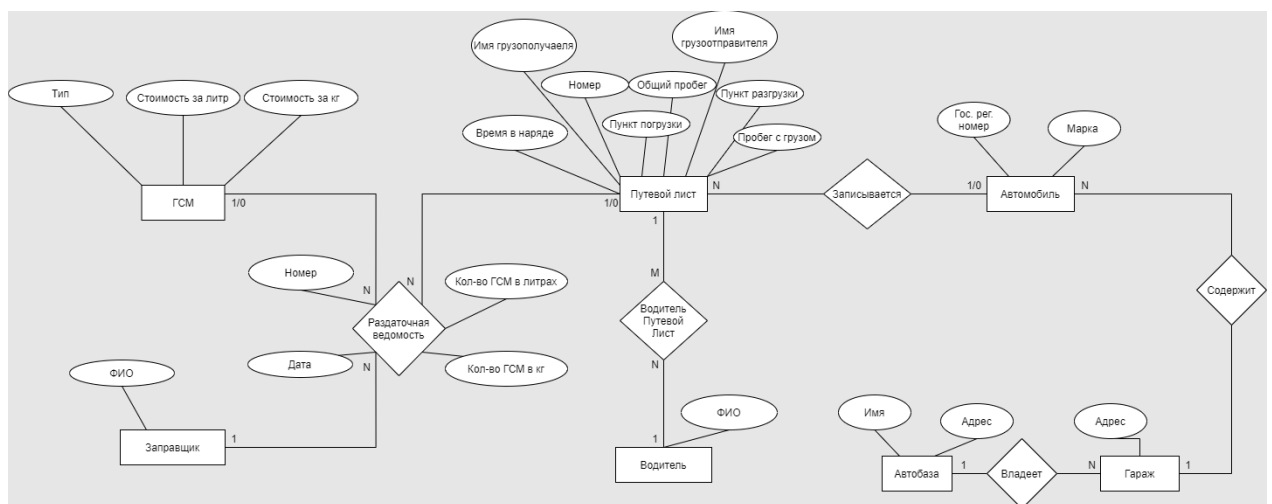


Схема инфологической модели данных БД, выполненная в среде CA ERwin Data Modeler:

Наименование атрибута	Тип	Первичный ключ		Внешний ключ	Обязательность	Ограничения целостности
		Собственный ключ	Внешний ключ			
Автобаза						
код автобазы	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения
Имя	VARCHAR(30)				+	
Адрес	VARCHAR(60)				+	
Гараж						
ID_гаража	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения
Код автобазы	INTEGER		+		+	Значение каскадируется по первичному ключу сущности «Автобаза»

Адрес	VARCHAR(60)				+	
Автомобиль						
Гос.рег.номер	INTEGER	+				Формируется из базы автомобилей
ID_гаража	INTEGER		+		+	Значение каскадируется по первичному ключу сущности «Гараж»
Код автобазы	INTEGER		+		+	Значение каскадируется по первичному ключу сущности «Автобаза»
Марка автомобиля	VARCHAR(30)				+	
ГСМ						
ID_гсм	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения

Тип	VARCHAR(30)				+	Значение должно выбираться из списка типов гсм
Стоимость за литр	INTEGER				+	
Стоимость за килограмм	INTEGER				+	
Путевой лист						
Номер	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения
Пункт погрузки	VARCHAR(30)				+	
Пункт разгрузки	VARCHAR(30)				+	
Общий пробег	INTEGER				+	
Пробег с грузом	INTEGER				+	
Имя грузоотправите ля	VARCHAR(30)				+	
Имя грузополучател я	VARCHAR(30)				+	

Время (в наряде)	DATETIME				+	
ID_гаража	INTEGER			+	+	Значение каскадируется по первичному ключу сущности «Гараж»
Гос. рег. номер автомобиля	INTEGER			+	+	Значение каскадируется по первичному ключу сущности «Автомобиль»
Код автобазы	INTEGER			+	+	Значение каскадируется по первичному ключу сущности «Автобаза»
Водитель						
ID_водителя	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения
ФИО	VARCHAR(60)				+	
Заправщик						
ID_заправщика	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения
ФИО	VARCHAR(60)				+	
Раздаточная ведомость						
ID_ведомости	INTEGER	+			+	Уникален, необходимо обеспечить автоматическую Генерацию значения
Номер путевого листа	INTEGER			+	+	Значение каскадируется по первичному

						ключу сущности «Путевой лист»
Дата	DATETIME				+	
Номер ведомости	INTEGER				+	

Количество ГСМ в литрах	INTEGER				+	
Количество ГСМ в кг	INTEGER				+	
ID_гсм	INTEGER			+	+	Значение каскадируется по первичному ключу сущности «ГСМ»
Водитель_Путевой_лист						
ID_водителя	INTEGER		+		+	Значение каскадируется по первичному ключу сущности «Водитель»
Номер_путевого_листа	INTEGER		+		+	Значение каскадируется по первичному ключу сущности «Путевой лист»

Запросы

1. Для каждой автобазы указать количество закрепленных за ней автомобилей.

```
select autobase.name, count(car_id) from autobases.autobase, autobases.car
where autobase.autobase_id = car.autobase_id
group by autobase.name
```

Data Output		Explain	Messages	Notifications
	name		count	
	character varying (30)		bigint	
1	Southern autobase		1	
2	Western autobase		1	
3	Eastern autobase		1	
4	Northern autobase		1	
5	Great autobase		1	

2. Вывести список водителей, заправлявшихся в те же дни, что и заданный водитель.

```
select driver.full_name from autobases.driver
join autobases.driver_road on driver.driver_id = driver_road.driver_id
join autobases.road_list on driver_road.road_list_id = road_list.road_list_id
join autobases.payroll on road_list.road_list_id = payroll.road_list_id
where payroll.date_payroll = '2020-03-21'
```

	<table><tr><th>full_name</th></tr><tr><td>character varying (60)</td></tr></table>	full_name	character varying (60)
full_name			
character varying (60)			
1	Vaskob Nikolay Alekseevich		

3. Для каждого рейса вывести общий объем топлива в литрах и килограммах в заданный промежуток времени.

```
select payroll.payroll_number, sum(payroll.amount_gsm_liter) as liters,
sum(payroll.amount_gsm_kg) as kgs
from autobases.payroll
where payroll.date_payroll between '2020-02-28' and '2020-03-21'
group by payroll.payroll_number
```


Data Output	Explain	Messages	Notifications
	payroll_number integer	liters bigint	kgs bigint
1	211	13	12
2	777	2	2
3	137	6	6
4	100	19	20

4. Вывести общий объем отпущенного топлива на предприятии по каждому виду в заданный промежуток времени.

```
select gsm.type_gsm, sum(payroll.amount_gsm_liter) + sum(payroll.amount_gsm_kg) as
total_amount
from autobases.gsm
join autobases.payroll on gsm.gsm_id = payroll.gsm_id
where payroll.date_payroll between '2020-02-28' and '2020-03-21'
group by gsm.type_gsm
```

Data Output	Explain	Messages	Notifications
	type_gsm character varying (30)	total_amount bigint	
1	autol		39
2	diesel fuel		12
3	nigrol		25
4	solidol		4

5. Для заданного водителя вывести информацию о всех его заправках указанием номеров путевых листов и общим объемом отпущенных ГСМ.

```
select payroll.payroll_number, sum(payroll.amount_gsm_liter) + sum(payroll.amount_gsm_kg) as
total_amount
from autobases.payroll
join autobases.road_list on payroll.road_list_id = road_list.road_list_id
join autobases.driver_road on road_list.road_list_id = driver_road.road_list_id
join autobases.driver on driver_road.driver_id = driver.driver_id
where driver.full_name = 'Vaskob Nikolay Alekseevich'
group by payroll.payroll_number
```

Data Output	Explain	Messages	Notifications
	payroll_number integer	total_amount bigint	
1	211		25

6. Необходимо предусмотреть возможность получения отчета, в котором для каждой автобазы указывается список раздаточных ведомостей с указанием количества автомобилей, количества и стоимости отпущенного топлива по каждому типу топлива и автобазе, общей стоимости топлива по предприятию.

+ Сортировка по возрастанию

```
select autobase.name, payroll.payroll_number, count(gov_reg_number) as cars, gsm.type_gsm,
sum(payroll.amount_gsm_liter) + sum(payroll.amount_gsm_kg) as gsm_amount,
count(gsm.price_liter) + count(gsm.price_kg) as gsm_price
from autobases.autobase
join autobases.road_list on autobase.autobase_id = road_list.autobase_id
join autobases.payroll on road_list.road_list_id = payroll.road_list_id
join autobases.car on autobase.autobase_id = car.autobase_id
join autobases.gsm on payroll.gsm_id = gsm.gsm_id
group by gsm.type_gsm, payroll.payroll_number, autobase.name
order by gsm_amount asc
```

Data Output

Explain

Messages

Notifications

	name character varying (30)	payroll_number integer	cars bigint	type_gsm character varying (30)	gsm_amount bigint	gsm_price bigint
1	Eastern autobase	777	1	solidol	4	2
2	Southern autobase	7	1	gasolyne	10	2
3	Western autobase	137	1	diesel fuel	12	2
4	Great autobase	211	1	nigrol	25	2
5	Northern autobase	100	1	autol	39	2

7. Посчитать количество водителей в каждом гараже с учетом некомплектованных гаражей без штата водителей.

```
select garage.garage_id, count(driver_road.driver_id) from autobases.garage, autobases.car,
autobases.road_list, autobases.driver_road, autobases.driver
where garage.garage_id = car.garage_id and car.car_id = road_list.car_id
and road_list.road_list_id = driver_road.road_list_id
group by garage.garage_id
union
select garage.garage_id, 0 from autobases.garage
where garage.garage_id not in (select garage.garage_id from autobases.car)
```

	Data Output	Explain	Messages	Notifications
	garage_id integer	count bigint		
1	1	5		
2	5	5		
3	2	5		
4	4	5		
5	3	5		

Примечание: данных было мало, поэтому некомплектованных гаражей нет.

8. Вывести гос. номера и марки автомобилей, на которые были выписаны путевые листы с исключением незарегистрированных машин.

```
select car.gov_reg_number, car.brand from autobases.car
where car.gov_reg_number is not NULL and exists
(select * from autobases.road_list where car.car_id = road_list.car_id)
```

Data Output	Explain	Messages	Notifications
	gov_reg_number integer	brand character varying (30)	
1	1123	Nissan	
2	2121	Skoda	
3	33323	Zhiguli	
4	4111	Lada	
5	523	Nissan 6	

Все машины зарегистрированы.

9. Для каждой автобазы вывести количество закрепленных за ней автомобилей с учетом того, что есть некомплектованные автопарком автобазы.

```
select autobase.name, count(car.gov_reg_number) from autobases.autobase
left join autobases.garage on garage.autobase_id = autobase.autobase_id
left join autobases.car on car.garage_id = garage.garage_id
group by autobase.autobase_id
```

Data Output	Explain	Messages	Notifications
	name character varying (30)	count bigint	
1	Northern autobase	1	
2	Western autobase	1	
3	Great autobase	1	
4	Eastern autobase	1	
5	Southern autobase	1	

Также видим, что все автобазы укомплектованы.

10. Вывести информацию о гаражах и закрепленных за ними машинах. Отсортировать по возрастанию номера гаража.

```
select garage.garage_id, car.gov_reg_number from autobases.garage
inner join autobases.car on garage.garage_id = car.garage_id
order by garage.garage_id asc
```

Data Output	Explain	Messages	Notifications
	garage_id integer	gov_reg_number integer	
1	1	2121	
2	2	4111	
3	3	33323	
4	4	523	
5	5	1123	

11. Вывести номера машин, которые находятся в гаражах по адресам, содержащим букву о. Отсортировать по убыванию гос. рег. номера машин.

```
select car.gov_reg_number from autobases.car
where car.garage_id = some
(select garage.garage_id from autobases.garage
where garage.address like '%o%')
order by car.gov_reg_number desc
```

Data Output	Explain	Messages	Notifications
gov_reg_number integer			
1	4111		
2	2121		
3	1123		
4	523		

12. Вывести номера путевых листов, выданных позже февраля и гос.номера автомобилей, на которые они выданы.

```
select payroll.payroll_number, car.gov_reg_number from autobases.payroll
join autobases.road_list on payroll.road_list_id = road_list.road_list_id
join autobases.car on road_list.car_id = car.car_id
where payroll.date_payroll > (select payroll.date_payroll from autobases.payroll
where payroll.date_payroll = '2020-02-28')
```

Data Output	Explain	Messages	Notifications
payroll_number integer	gov_reg_number integer		
1	211	2121	
2	100	33323	
3	7	4111	
4	777	523	

13. Вывести количество топлива, которое было отпущено по каждой марке, и его тип за каждый день с '2020-03-05' по '2020-04-01'.

```
select payroll.date_payroll, gsm.type_gsm, amount_gsm_liter + amount_gsm_kg as gsm_qty
from autobases.payroll, autobases.gsm
where payroll.gsm_id = gsm.gsm_id and
payroll.date_payroll between '2020-03-05' and '2020-04-01'
```

	Data Output	Explain	Messages	Notifications
	date_payroll date	type_gsm character varying (30)	gsm_qty integer	
1	2020-03-05	solidol	4	
2	2020-03-14	autol	39	
3	2020-03-21	nigrol	25	
4	2020-04-01	gasolyne	10	

14. Вывести общий пробег по каждому путевому листу, где количество машин = 1. Отсортировать по возрастанию пробега.

```
select road_list.total_mileage from autobases.road_list
group by road_list.road_list_id
having count(road_list.car_id) = 1
order by road_list.total_mileage asc
```

	Data Output	Explain	Messa
	total_mileage character varying (10)		
1	0.5		
2	1.09		
3	1.66		
4	10.61		
5	3.66		

15. Вывести тип ГСМ и цену в килограммах, которая будет отражать, дорогой ли тип ГСМ или нет. Если цена в кг > 1.5, то дорогой, в остальных случаях – дешёвый.

```
select gsm.type_gsm,
case
when gsm.price_kg > '1.5' then 'expensive'
else 'cheap'
end as price_kg
from autobases.gsm
```

	Data Output	Explain	Messages	Notifications
	type_gsm character varying (30)	price_kg text		
1	gasolyne	expensive		
2	diesel fuel	cheap		
3	autol	cheap		
4	solidol	cheap		
5	nigrol	expensive		

Вывод:

В данной лабораторной работе были использованы навыки написания sql-запросов на базе PGAdmin.