

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет «МРИП»  
Направление подготовки «Data, Culture, Visualization»

О Т Ч Е Т

Тема задания: SQL запросы

---

Выполнил:

Студент Селина М. А. D41423

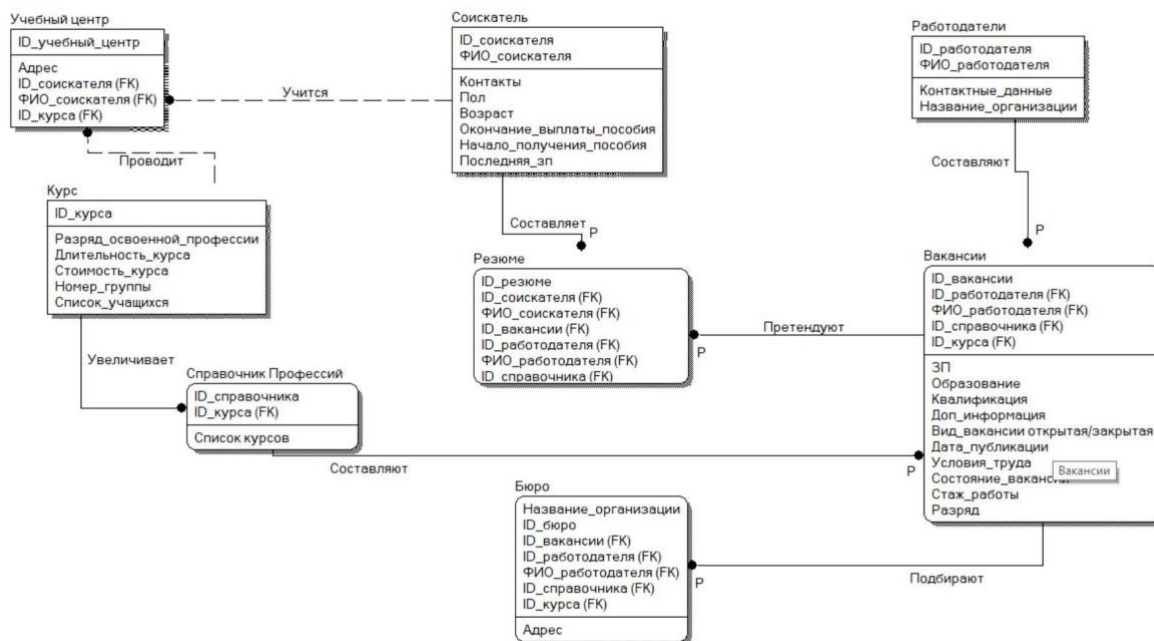
Проверил:

Преподаватель: Говоров А.И.

Санкт-Петербург  
2020

Цель: овладеть практическими навыками написания запросов к базе данных

Индивидуальное задание:



Для биржи труда создается ИС, главной целью которой является хранение информации о соискателях и вакансиях, а также формирование резюме соискателя, вывод данных о работодателях, состоянии вакансии, проходящих курсах и т.д. В отчетах фигурируют данные о соискателях, закрытых вакансиях, проводимых курсах, работодателях.

Для каждого соискателя составляется резюме, осуществляется подбор вакансии по профессии и образованию, начисляется пособие. Размер пособия зависит размер последней заработной платы соискателя. При начислении пособия необходима информация о датах начала и окончания выплаты пособия.

Соискателям предлагается пройти курсы. После прохождения курсов соискатели получают разряд по освоенной профессии.

Для работодателей фиксируются их контактные данные (название, адрес, контактное лицо, телефон, электронный адрес), дата подачи вакансии, состояние вакансии. Работодатель при размещении вакансии должен указать в заявке перечень требуемых профессий, образование соискателя, требуемый стаж, разряд, заработную плату. Возможна дополнительная информация

## Список запросов:

1. Выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой

- 1.1 Вывод списка студентов и списка курсов отсортированного по стоимости

```
SELECT "Course"."Students_list", "Directory_professions"."Course_list" FROM l_e."Course", l_e."Directory_professions" WHERE "Course"."ID_course" = "Directory_professions"."ID_course" ORDER BY "Price";
```

	Students_list text	Course_list text
1	В.А. Попов, Л.С. ...	Пилить
2	В.А. Попов, Л.С. ...	Красить
3	В.А. Попов, К.К. И...	Красить
4	В.А. Попов, К.К. И...	Быстро ездить
5	В.А. Попов, Л.С. ...	Быстро ездить

- 1.2 Вывод ID и ФИО работодателей отсортированных по названию организации

```
SELECT "ID_employer", "FCS_employer" FROM l_e."Employers" ORDER BY "Organization_name";
```

	ID_employer [PK] integer	FCS_employer [PK] text
1	2	Е.Ф. Сидоров
2	4	Л.Т. Лебедев
3	3	С.В. Иванов
4	5	О.Н. Желтова
5	1	А.О. Попов

2. Использование условий WHERE, состоящих из более, чем одного условия

- 2.1 Вывод заявителей мужчин младше 30 лет

```
SELECT * FROM l_e."Applicant" WHERE "Sex"='M' AND "Age"<30;
```

	ID_applicant [PK] integer	FCS_applicant [PK] text	Age integer	Sex text	Contacts text	Last_salary integer	Start_receiving_benefits date	End_benefit_receipt date
1	1	В.А. Попов	19	M	44-44-44	30000	2018-01-01	2019-01-01
2	3	К.К. Иванов	23	M	23-23-23	12000	2018-10-01	2018-11-01

- 2.2 Вывод контактного номера работодателя А.О. Попова, который работает в Яндексе

```
SELECT "Contact_details" FROM l_e."Employers" WHERE "Organization_name"='Яндекс' AND "FCS_employer"='А.О. Попов';
```

	Contact_details text
1	11-11-11

### 3. Использование функций для работы с датами

#### 3.1 Вывод ФИО и дат начала выплаты пособия для заявителей

```
SELECT "ID_applicant", "FCS_applicant", "Start_receiving_benefits" FROM I_e."Applicant";
```

	ID_applicant [PK] integer	FCS_applicant [PK] text	Start_receiving_benefits date
1	1	В.А. Попов	2018-01-01
2	2	П.К. Сидоров	2019-03-01
3	3	К.К. Иванов	2018-10-01
4	4	Н.У. Лебедев	2010-01-01
5	5	Л.С. Желтова	2014-07-01

#### 3.2 Вывод ID заявителей и ФИО, которые уже закончили их получать в сентябре 2019

```
SELECT "ID_applicant", "FCS_applicant" FROM I_e."Applicant" WHERE "End_benefit_receipt"<'2019-09-01';
```

	ID_applicant [PK] integer	FCS_applicant [PK] text
1	1	В.А. Попов
2	3	К.К. Иванов
3	5	Л.С. Желтова

### 4. Использование строковых функций

#### 4.1 Вывод контактной информации от организаций, в которых название организации меньше 7 символов

```
SELECT "Contact_details", "Organization_name" FROM I_e."Employers" WHERE char_length("Organization_name")<7;
```

	Contact_details text	Organization_name text
1	11-11-11	Яндекс
2	22-22-22	Apple
3	55-55-55	Xiaomi

#### 4.2 Выводим только фамилии заявителей (без И.О.)

```
SELECT SUBSTR("FCS_applicant",6) FROM I_e."Applicant";
```

1	Попов
2	Сидоров
3	Иванов
4	Лебедев
5	Желтова

## 5. Запрос с использованием подзапросов

### 5.1 Выводим ФИО заявителей женского пола

```
SELECT "FCS_applicant" FROM I_e."Applicant" WHERE "Sex" = (SELECT "Sex" FROM I_e."Applicant" WHERE "Applicant"."Sex" = 'F');
```

	<b>FCS_applicant</b> text	
1	Л.С. Желтова	

### 5.2 Вывод ФИО заявителей с предыдущей зарплатой больше 30000

```
SELECT "FCS_applicant" FROM I_e."Applicant" WHERE "Last_salary" = (SELECT "Last_salary" FROM I_e."Applicant" WHERE "Applicant"."Last_salary" > 30000);
```

	<b>FCS_applicant</b> text	
1	П.К. Сидоров	

## 6. Вычисление групповой (агрегатной) функции

### 6.1 Вывод количества учащихся на втором курсе

```
SELECT count("ID_course") FROM I_e."Course" WHERE "Group_number" LIKE '_3%';
```

	<b>count</b> bigint	
1	2	

### 6.2 Вычисление средней прошлой зарплаты заявителей

```
SELECT AVG("Last_salary") FROM I_e."Applicant";
```

	<b>avg</b> numeric	
1	24800.000000000000	

## 7. Вычисление групповой (агрегатной) функции с условием HAVING

### 7.1 Вывод заявителей младше 30.

```
SELECT "ID_applicant", "FCS_applicant" FROM I_e."Applicant" GROUP BY
"Age", "ID_applicant", "FCS_applicant" HAVING "Age"<30;
```

	<b>ID_applicant</b> [PK] integer	<b>FCS_applicant</b> [PK] text
1	1	В.А. Попов
2	3	К.К. Иванов

7.2 Вывод ФИО и название организаций, в которых фамилия короче 7 букв.

```
SELECT "FCS_employer", "Organization_name" FROM I_e."Employers"
GROUP BY "FCS_employer", "Organization_name" HAVING
LENGTH("FCS_employer")< 12;
```

	<b>FCS_employer</b> text	<b>Organization_name</b> text
1	С.В. Иванов	Facebook
2	А.О. Попов	Яндекс

8. Использование предикатов EXISTS, ALL, SOME и ANY

8.1 Выведем курсы, стоимость которых меньше 5000

```
SELECT "ID_course", "Price" FROM I_e."Course" WHERE "Price" =
ANY(SELECT "Price" FROM I_e."Course" WHERE "Price" < 5000 );
```

	<b>ID_course</b> [PK] integer	<b>Price</b> integer
1	1	3000
2	2	4000
3	3	1000

9. Использование запросов с операциями реляционной алгебры  
(объединение, пересечение и т.д.)

9.1 Вывод ФИО заявителя и количество курсов, которые он может оплатить  
на предыдущую зарплату

```
SELECT "Applicant"."FCS_applicant", CAST("Applicant"."Last_salary" as
FLOAT) / CAST("Course"."Price" as FLOAT) FROM I_e."Applicant" INNER JOIN
I_e."Course" ON "Applicant"."ID_applicant"="Course"."ID_course";
```

	<b>FCS_applicant</b> text	<b>?column?</b> double precision
1	В.А. Попов	10
2	П.К. Сидоров	8
3	К.К. Иванов	12
4	Н.У. Лебедев	3
5	Л.С. Желтова	4.83333333333333

## 10. Использование объединений запросов (inner join и т.д.)

### 10.1 Вывод ФИО всех работодателей и заявителей

```
SELECT "FCS_employer" FROM I_e."Employers" UNION ALL SELECT
"FCS_applicant" FROM I_e."Applicant";
```

	<b>FCS_employer</b> text
1	А.О. Попов
2	Е.Ф. Сидоров
3	С.В. Иванов
4	Л.Т. Лебедев
5	О.Н. Желтова
6	В.А. Попов
7	П.К. Сидоров
8	К.К. Иванов
9	Н.У. Лебедев
10	Л.С. Желтова

### 10.2 Вывести все данные о курсах, которые находятся в справочнике

```
SELECT "Course"."ID_course", "New_discharge", "Duration", "Price",
"Group_number", "Students_list" FROM I_e."Course" INNER JOIN
I_e."Directory_professions" ON
"Course"."ID_course"="Directory_professions"."ID_course" WHERE
"Course_list"="New_discharge";
```

	<b>ID_course</b> [PK] integer	<b>New_discharge</b> text	<b>Duration</b> time with time zone	<b>Price</b> integer	<b>Group_number</b> text	<b>Students_list</b> text
1	1	Красить	04:05:06+03:00	3000	K3240	В.А. Попов, Л.С. ...
2	5	Быстро ездить	19:45:22+03:00	6000	K4441	В.А. Попов, К.К. И...