ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт математики, информатики и робототехники

КАФЕДРА ПРОГРАММИРОВАНИЯ И ЭКОНОМИЧЕСКОЙ ИНФОРМАТИКИ

**ОТЧЕТ К ЛАБОРАТОРНОЙ РАБОТЕ**

по дисциплине «Базы данных»

|  |  |
| --- | --- |
|  | Выполнила:  Студент(ка) 3 курса очной формы обучения  Направление подготовки (специальность) 09.03.03 Прикладная информатика  Направленность (профиль)  Информационные и вычислительные технологии  Михайлов Александр Олегович |
|  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    Бердникова М. Л.  о           (подпись) |

 «25\_» \_\_\_Декабря\_\_\_\_\_\_\_\_ 2023 г.

УФА – 2023

**1. Обследование предметной области**

**Предметная область: Бюро по трудоустройству**

Вы работаете в бюро по трудоустройству. Вашей задачей является отслеживание финансовой стороны работы компании. Деятельность бюро организована следующим образом: бюро готово искать работников для различных работодателей и вакансии для ищущих работу специалистов различного профиля. При обращении к работодателю его стандартные данные (название, вид деятельности, адрес, телефон) фиксируются в базе данных. При обращении к соискателю его стандартные данные (фамилия, имя, отчество, квалификация, профессия, иные данные) также фиксируются в базе данных.

При обращении к компании его стандартные данные ( Имя, индустрия. Адрес, телефон ) также фиксируются в базе данных. По каждому факту удовлетворения обеих сторон будет проведена сделка. В сделке указываются соискатель, работодатель, вакансия и комиссионные (доход бюро).

**Классы:**

1. Класс "Соискатели" (Applicant)

1. Класс "Вакансия" (Vacancy):
2. Класс "Сделка" (Deal)
3. Класс "Работодатели" (Employers)

**Связи:**

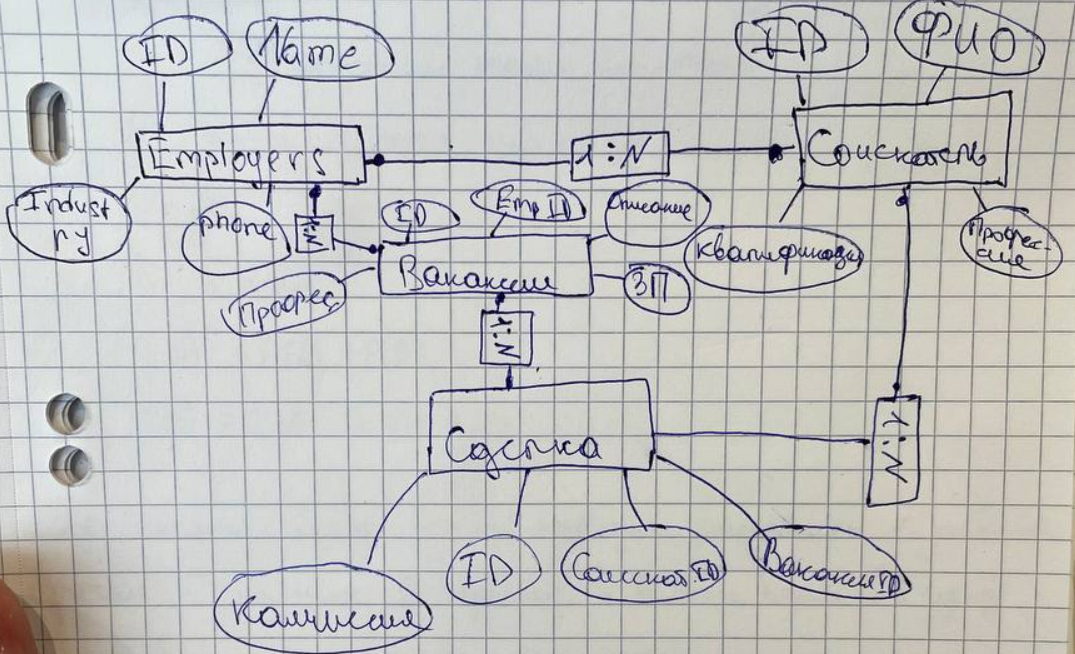
**Работодатель и Вакансии (1 к N): Один работодатель может предложить несколько вакансий.**

**Соискатель и Сделка (1 к N): Один соискатель может быть вовлечён в несколько сделок.**

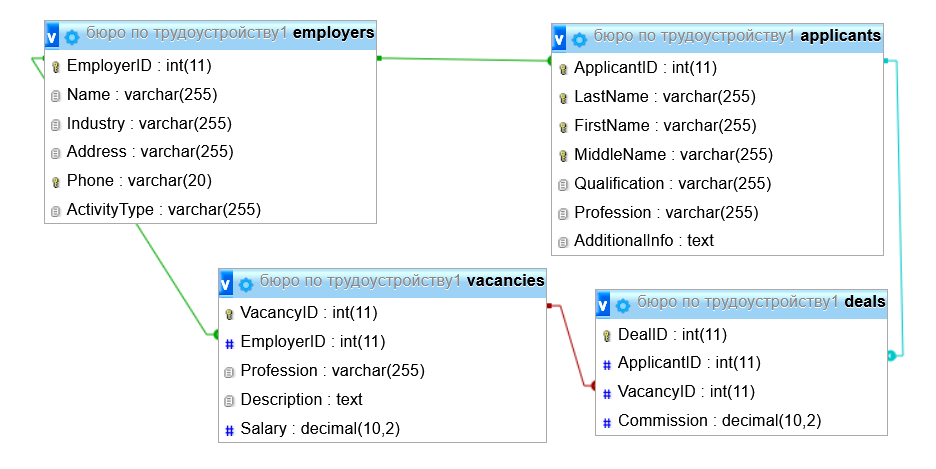
**Вакансия и Сделки (1 к N): Одна вакансия может быть связана с несколькими сделками, если рассматривать несколько кандидатов на одну позицию.**

**Работодатель и Соискатель ( 1 к N ): Один работодатель может быть связан с несколькими соискателями.**

# 2. Модель «сущность-связь»



# 3. Реляционная модель БД



# 4.  Разработка на языке SQL

**1. Запрос на создание базы данных**

CREATE DATABASE Бюро по трудоустройству

**2. Запросы на создание таблиц**

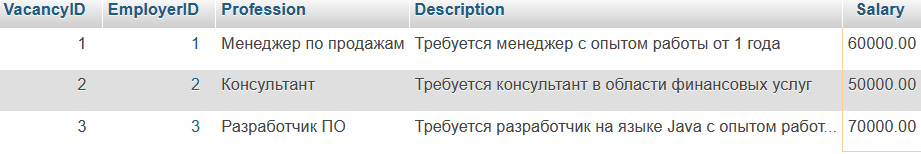
* *-- Структура таблицы `applicants`*
* *--*
* CREATE TABLE `applicants` (
* `ApplicantID` **int**(11) NOT NULL,
* `LastName` **varchar**(255) NOT NULL,
* `FirstName` **varchar**(255) NOT NULL,
* `MiddleName` **varchar**(255) **DEFAULT** NULL,
* `Qualification` **varchar**(255) NOT NULL,
* `Profession` **varchar**(255) NOT NULL,
* `AdditionalInfo` **text**
* ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;
* *--*
* *-- Дамп данных таблицы `applicants`*
* *--*
* INSERT INTO `applicants` (`ApplicantID`, `LastName`, `FirstName`, `MiddleName`, `Qualification`, `Profession`, `AdditionalInfo`) VALUES
* (1, 'Иванов', 'Иван', 'Иванович', 'Высшее', 'Инженер-программист', 'Опыт работы 5 лет'),
* (2, 'Петрова', 'Мария', 'Алексеевна', 'Среднее специальное', 'Бухгалтер', 'Владение 1С: Бухгалтерия'),
* (3, 'Сидоров', 'Николай', 'Петрович', 'Высшее', 'Менеджер по продажам', 'Опыт работы в продажах более 3 лет');
* *-- --------------------------------------------------------*
* *--*
* *-- Структура таблицы `deals`*
* *--*
* CREATE TABLE `deals` (
* `DealID` **int**(11) NOT NULL,
* `ApplicantID` **int**(11) **DEFAULT** NULL,
* `VacancyID` **int**(11) **DEFAULT** NULL,
* `Commission` **decimal**(10,2) NOT NULL
* ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;
* *--*
* *-- Дамп данных таблицы `deals`*
* *--*
* INSERT INTO `deals` (`DealID`, `ApplicantID`, `VacancyID`, `Commission`) VALUES
* (1, 1, 3, '2000.00'),
* (2, 2, 2, '1005.00'),
* (3, 3, 1, '3000.00');
* *-- --------------------------------------------------------*
* *--*
* *-- Структура таблицы `employers`*
* *--*
* CREATE TABLE `employers` (
* `EmployerID` **int**(11) NOT NULL,
* `Name` **varchar**(255) NOT NULL,
* `Industry` **varchar**(255) NOT NULL,
* `Address` **varchar**(255) NOT NULL,
* `Phone` **varchar**(20) NOT NULL,
* `ActivityType` **varchar**(255) NOT NULL
* ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;
* *--*
* *-- Дамп данных таблицы `employers`*
* *--*
* INSERT INTO `employers` (`EmployerID`, `Name`, `Industry`, `Address`, `Phone`, `ActivityType`) VALUES
* (1, 'Завтра лучше чем вчера', 'Розничная торговля', 'Улица Пушкина, дом Колотушкина', '+71234567890', 'Продажи'),
* (2, 'Консалтинг Плюс', 'Консультационные услуги', 'Бульвар Мира, 21', '+70987654321', 'Консалтинг'),
* (3, 'IT Солюшнс', 'Информационные технологии', 'Проспект Науки, 3', '+70223344556', 'IT');
* *-- --------------------------------------------------------*
* *--*
* *-- Структура таблицы `vacancies`*
* *--*
* CREATE TABLE `vacancies` (
* `VacancyID` **int**(11) NOT NULL,
* `EmployerID` **int**(11) **DEFAULT** NULL,
* `Profession` **varchar**(255) NOT NULL,
* `Description` **text**,
* `Salary` **decimal**(10,2) NOT NULL
* ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;
* *--*
* *-- Дамп данных таблицы `vacancies`*
* *--*
* INSERT INTO `vacancies` (`VacancyID`, `EmployerID`, `Profession`, `Description`, `Salary`) VALUES
* (1, 1, 'Менеджер по продажам', 'Требуется менеджер с опытом работы от 1 года', '60000.00'),
* (2, 2, 'Консультант', 'Требуется консультант в области финансовых услуг', '50000.00'),
* (3, 3, 'Разработчик ПО', 'Требуется разработчик на языке Java с опытом работы от 3 лет', '70000.00');
* *--*
* *-- Индексы сохранённых таблиц*
* *--*
* *--*
* *-- Индексы таблицы `applicants`*
* *--*
* ALTER TABLE `applicants`
* ADD **PRIMARY KEY** (`ApplicantID`),
* ADD UNIQUE KEY `LastName` (`LastName`,`FirstName`,`MiddleName`);
* *--*
* *-- Индексы таблицы `deals`*
* *--*
* ALTER TABLE `deals`
* ADD **PRIMARY KEY** (`DealID`),
* ADD KEY `ApplicantID` (`ApplicantID`),
* ADD KEY `VacancyID` (`VacancyID`);
* *--*
* *-- Индексы таблицы `employers`*
* *--*
* ALTER TABLE `employers`
* ADD **PRIMARY KEY** (`EmployerID`),
* ADD UNIQUE KEY `Phone` (`Phone`);
* *--*
* *-- Индексы таблицы `vacancies`*
* *--*
* ALTER TABLE `vacancies`
* ADD **PRIMARY KEY** (`VacancyID`),
* ADD KEY `EmployerID` (`EmployerID`);
* *--*
* *-- AUTO\_INCREMENT для сохранённых таблиц*
* *--*
* *--*
* *-- AUTO\_INCREMENT для таблицы `applicants`*
* *--*
* ALTER TABLE `applicants`
* MODIFY `ApplicantID` **int**(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;
* *--*
* *-- AUTO\_INCREMENT для таблицы `deals`*
* *--*
* ALTER TABLE `deals`
* MODIFY `DealID` **int**(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;
* *--*
* *-- AUTO\_INCREMENT для таблицы `employers`*
* *--*
* ALTER TABLE `employers`
* MODIFY `EmployerID` **int**(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;
* *--*
* *-- AUTO\_INCREMENT для таблицы `vacancies`*
* *--*
* ALTER TABLE `vacancies`
* MODIFY `VacancyID` **int**(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;
* *--*
* *-- Ограничения внешнего ключа сохраненных таблиц*
* *--*
* *--*
* *-- Ограничения внешнего ключа таблицы `deals`*
* *--*
* ALTER TABLE `deals`
* ADD **CONSTRAINT** `deals\_ibfk\_1` **FOREIGN KEY** (`ApplicantID`) **REFERENCES** `applicants` (`ApplicantID`),
* ADD **CONSTRAINT** `deals\_ibfk\_2` **FOREIGN KEY** (`VacancyID`) **REFERENCES** `vacancies` (`VacancyID`);
* *--*
* *-- Ограничения внешнего ключа таблицы `vacancies`*
* *--*
* ALTER TABLE `vacancies`
* ADD **CONSTRAINT** `vacancies\_ibfk\_1` **FOREIGN KEY** (`EmployerID`) **REFERENCES** `employers` (`EmployerID`);

**3. Запросы**

* **Запрос 1:** Получить список всех соискателей, их имена и квалификация:

SELECT \* FROM Vacancies WHERE Salary > 40000;

**Выводит:**

****

* **Запрос 2:** Найти все сделки для определённого соискателя по его ID:

SELECT \* FROM Deals WHERE ApplicantID = 1;**Выводит:**



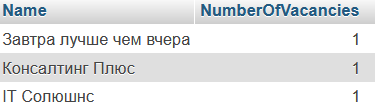
* **Запрос 3:** Получить информацию о работодателях и количестве их вакансий :

SELECT Employers.Name, COUNT(Vacancies.VacancyID) AS NumberOfVacancies

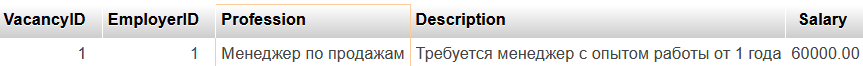
FROM Employers

JOIN Vacancies ON Employers.EmployerID = Vacancies.EmployerID

GROUP BY Employers.EmployerID; **Выводит:**



* **Запрос 4:** Найти вакансии, где требуется определённая профессия: SELECT \* FROM Vacancies WHERE Profession = 'Менеджер по продажам';**Выводит:**



* **Запрос 5:** Получить список всех сделок с комиссией выше определённой суммы: SELECT \* FROM Deals WHERE Commission > 1500;

**Выводит:**



* **Запрос 6:** Вывести среднюю зарплату по всем вакансиям:
* SELECT AVG(Salary) AS AverageSalary FROM Vacancies;

**Выводит:**



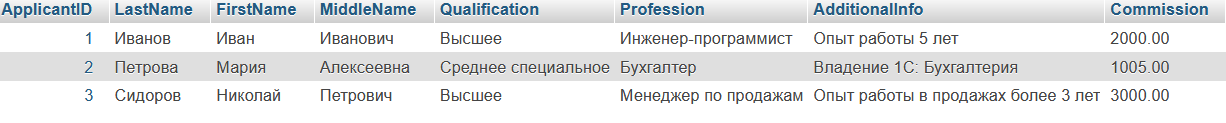
* **Запрос 7:** Вывести список всех соискателей и информацию о соответствующих им сделках:

SELECT Applicants.\*, Deals.Commission

FROM Applicants

JOIN Deals ON Applicants.ApplicantID = Deals.ApplicantID;

**Выводит:**



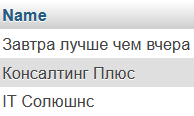
* **Запрос 8:**

SELECT DISTINCT Employers.Name

FROM Employers

JOIN Vacancies ON Employers.EmployerID = Vacancies.EmployerID

WHERE Vacancies.Salary > 1000;

Выводит: 

* **Запрос 9:** Найти общее количество вакансий в определённой индустрии: SELECT COUNT(\*) AS TotalVacancies

FROM Vacancies

JOIN Employers ON Vacancies.EmployerID = Employers.EmployerID

WHERE Employers.Industry = 'Строительство';

Выводит:



* **Запрос 10:** Найти количество вакансий для каждой профессии:

SELECT Profession, COUNT(\*) AS NumberOfVacancies

FROM Vacancies

GROUP BY Profession;

**Выводит:**

