

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Базы данных»
Тема: Проектирование ER модели и структуры БД по текстовому
описанию предметной области

Студент гр. 3384

Преподаватель

Рудаков А.Л.

Михайлова С.А.

Санкт-Петербург

2025

Цель работы.

Изучение и практическое освоение навыков проектирования ER-диаграмм и БД по заданному текстовому описанию предметной области

Задание.

Вариант 22.

Пусть требуется создать программную систему для поиска вакансий (аналог hh.ru). Такая система должна обеспечивать хранение сведений о работодателях и работниках. Эти сведения включают в себя (для работника) - паспортные работника, данные трудовой книжки, ИНН, дата рождения, информацию о среднем/высшем(их) образованиях, дата поступления на работу, в институт, информация об предыдущей работе(ах) из трудовой книжки. Данные трудовой книжки – это ее номер и дата выдачи, а также даты и номера приказов о зачислении и увольнении, о переходе в другое подразделение или об изменении должности. Кроме того, для работник может создать 1/несколько резюме, с указанием желаемой должности, ЗП, свои умения/навыки. Работодатель имеет возможность создавать/удалять/помещать в архив вакансии. Вакансия имеет название, ЗП, должность, адрес, требования, условия, комментарий, требуемый опыт. У работодателя есть страница с указанием информации о себе - название, фото, описание, файлы презентации, сфера деятельности (ИТ, финансовая и т.п.), количество вакансий (формируется на основе списка вакансий).

Система должна давать ответы на следующие вопросы:

- Какие вакансии есть у данной компании?
- Какая вакансия подходит мне по названию?
- Сколько поблизости вакансий от меня (указание улицы)?
- Какие вакансии были помещены в архив у компании?
- Средняя ЗП каждого работодателя?
- Сколько работников ищут работу, имея высшее образование?
- Сколько работников имело более 3-х мест работы?
- Какие вакансии имеют ЗП более 100 000р и не требуют опыта работы?

Выполнение работы.

При анализе текстового описания были выделены следующие сущности:

“Паспорт” – хранит информацию о паспорте работника. Содержит атрибуты: номер паспорта, серия паспорта. Связан с работником связью Работник 1(1) – 1(1) Паспорт.

“ИНН” – хранит информацию об ИНН работника. Содержит атрибуты: номер ИНН. Связан с работником связью Работник 1(1) – 1(1) ИНН.

“Трудовая книжка” – хранит информацию о трудовой книжке работника. Содержит атрибуты: номер трудовой книжки, дата выдачи трудовой книжки. Связан с работником связью Работник 1(1) – 1(1) Трудовая книжка.

“Работник” – хранит в себе сведения о работниках. Содержит атрибут дата. Также работник имеет сведения об образовании, но в силу того, что у каждого работника может как не быть образования, так и быть их несколько, образование решено вынести в отдельную сущность, связанную с работником связью: Работник M(0) – N(0) Образование. Связи являются необязательными, так как у работника может не быть образования, а у места получения образования, внесенного в бд, может пока не быть работников, закончивших его и зарегистрированных в данном сервисе, а может быть много.

“Образование” – хранит в себе сведения об образовании работников. Содержит атрибуты: дата поступления, название заведения, тип (высшее, среднее).

“Резюме” – хранит в себе информацию о созданных работниками резюме. Содержит атрибут ЗП. Кроме этого, в резюме требуется хранить описание навыков, но их может быть много, поэтому решено вынести их в отдельную сущность “Навыки”, с которой имеется связь: Резюме N(0) – M(0) – Навыки. Необязательность связей обусловлена тем, что в резюме может не быть указано ни одного навыка, а может быть сразу много, при этом умения/навыки хранятся отдельно и пока может не быть ни одного резюме с определенным навыком (например какой-то очень необычный навык), при этом популярные навыки могут быть использованы в многих резюме. Также в резюме должна быть

желаемая должность, которая так же вынесена в отдельную сущность и соединена с помощью связи: Резюме M(0) – 1(1) Должность. Такая связь обусловлена тем, что в каждом резюме обязательно должна быть прописана одна желаемая должность, при этом должность может быть использована в нескольких резюме, а также она в бд может существовать и без резюме (если пока никто ее не использовал). Сущность связана с работниками связью: Работник 1(1) – M(0) Резюме. Связь обусловлена тем, что у одного работника может как не быть пока ни одного резюме, так и быть несколько, а если резюме есть, то оно обязательно относится только к одному работнику, создавшему его.

“Умение/навык” – хранит в себе информацию о навыке. Содержит атрибут описание умения. Вынесен в отдельную сущность для того, чтобы был определенный список навыков, из которых работник мог выбирать, какие добавить в резюме. Проще говоря, сделано для того, чтобы, например, навыки “программирование на python” и “python-программирование” не были разными навыками.

“Должность” – хранит в себе информацию о должностях. Содержит атрибуты: название, описание. Вынесена в отдельную сущность для того, чтобы существовал определенный список должностей, аналогично со списком навыков. Проще говоря, это сделано для того, чтобы, например, должности “python-разработчик” и “разработчик на python” не были разными должностями.

“Вакансия” – хранит в себе сведения о вакансиях, размещаемых работодателями. Содержит атрибуты: название, ЗП, адрес, требуемый опыт, комментарий, статус (открыта или в архиве). Кроме этого, вакансия должна иметь должность, поэтому она соединена с сущностью “Должность” связью: Вакансия M(0) – 1(1) Должность. Такая связь объясняется тем, что в каждой вакансии обязательно должна быть указана ровно одна должность, при этом одна и та же должность может быть указана как в нескольких вакансиях, так и ни в одной. Также в вакансии могут быть требования и условия, но по тем же соображениям, что и умения/навыки в резюме (что их может быть как 0, так и много и чтобы они были однотипными для всех), их решено вынести в отдельные

сущности и соединить связями: Вакансия $M(0) - N(0)$ Требование, Вакансия $M(0) - N(0)$ Условие. Связи обусловлены тем, что в вакансиях может как не быть требований и/или условий, так и быть их много, а требования и/или условия могут встречаться в нескольких вакансиях или пока не встречаются ни в одной. Кроме этого вакансии создает работодатель, поэтому имеется связь: Вакансия $M(0) - 1(1)$ Работодатель. Связь обусловлена тем, что у работодателя может как не быть вакансий, так и быть много, в то время как если вакансия существует, то у нее обязан быть ровно один работодатель.

“Работодатель” - хранит в себе информацию о работодателях. Содержит атрибуты: название, фото(ссылка), описание. Кроме этого, у работодателя должна быть указана сфера деятельности, но для единообразия сфер деятельности их решено вынести в отдельную сущность, поэтому имеется связь: Работодатель $M(0) - 1(1)$ Сфера деятельности. Связь обусловлена тем, что сфера деятельности у работодателя обязательно должна быть одна, при этом одна сфера деятельности из бд может быть как у нескольких работодателей, так и не быть ни у одного. Также у работодателя могут быть файлы презентации. Так как их может быть много, решено их вынести в отдельную сущность, поэтому есть связь: Работодатель $1(1) - M(0)$ Файл презентации. Связь обусловлена тем, что у работодателя может быть как 0, так и несколько файлов презентаций, при этом если файл презентации существует, то у него обязан существовать ровно один работодатель.

“Сфера деятельности” – хранит информацию о сфере деятельности. Содержит атрибут названия. Вынесена в отдельную сущность по тем же причинам, что и умения/навыки, требования, условия.

“Файл презентации” – хранит информацию о файлах презентаций работодателей. Содержит атрибуты: название и ссылка.

“Запись трудовой книжки” – хранит информацию о записях в трудовых книжках от работодателей работникам. Содержит атрибуты: тип (зачисление, увольнение, перевод, изменение должности) и дата. Запись в трудовой книжке пишется конкретным работодателем конкретному работнику, поэтому тут

имеется связь: Трудовая книжка 1(1), Работодатель 1(1) – М(0) запись трудовой книжки. Связь обусловлена тем, что записей от одного работодателя одному работнику может быть много или не быть ни одного, но при этом если запись есть, то она относится к конкретному работодателю и конкретному работнику.

ER-модель представлена на рис.1.

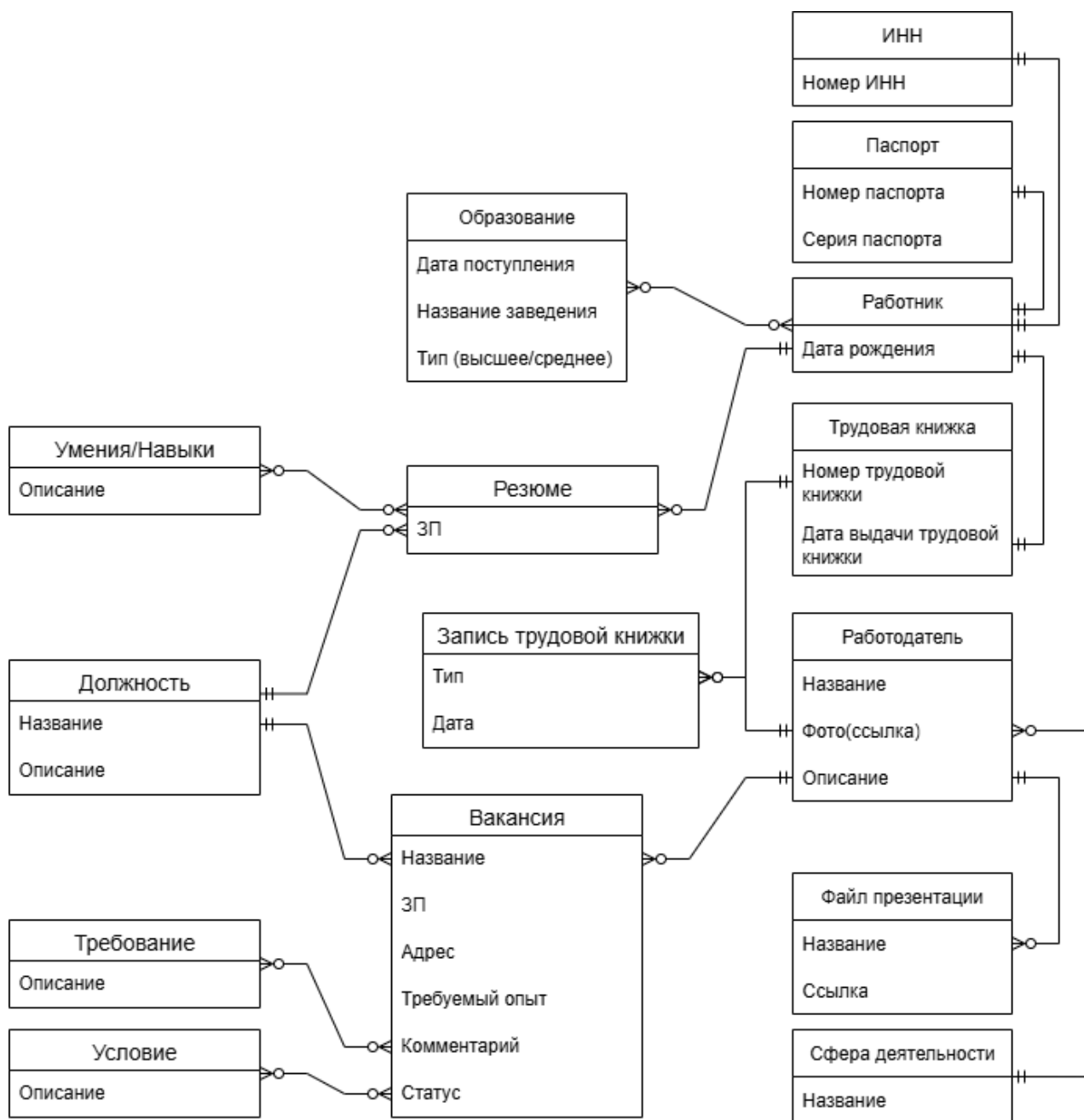


Рисунок 1 – ER-модель

При построении структуры бд, из-за связей типа М - N появились дополнительные таблицы, связывающие сущности между собой.

Если сущность “Работник” объединить с “Паспорт”, “ИНН”, “Трудовая книжка”, в одну таблицу, то она не будет удовлетворять НФБК, поэтому они были разделены на 4 таблицы:

“Workbook” – сведения о трудовой книжке с PK id: INT, и полями workbook_number: INT, issue_date: DATE.

“TIN” – сведения об ИНН с PK id: INT, и полем TIN: INT.

“Passport” – сведения о паспорте с PK id: INT, и полями passport_number: INT, passport_series: INT.

“Worker” – данные о работнике с PK id: INT, а также FK: passport_id: INT, TIN_id: INT, workbook_id: INT, и полем birthday: DATE.

“Workbook”, “TIN”, “Passport” связаны связями 1(1) – 1(1) с “Worker”, так как они уникальны для каждого работника.

“Education” – сведения об образовании с PK id: INT, и полями: name: VARCHAR, type: ENUM.

Так как “Образование” и “Работник” были соединены связью М – N, то добавлена дополнительная таблица для связи:

“WorkerEducation” – информация о связи одного работника с одним образованием, с PK, FK: worker_id: INT, education_id: INT и полем enroll_date: DATE. “Education” и “Worker” связаны с “WorkerEducation” связями 1(1) – M(0). Так как каждое образование может иметь как несколько, так и ноль записей и каждый работник может иметь как несколько, так и ноль записей, при этом если запись существует, то она хранит одно образование и одного работника.

“Resume” – информация о резюме, с PK id: INT, а также FK: worker_id: INT, post_id: INT, и полем salary: INT.

“Skill” – информация о навыке, с PK id: INT, и полем description: VARCHAR.

“Резюме” и “Навык” были соединены связью М – N, поэтому по аналогии с “WorkerEducation” была добавлена:

“ResumeSkill” – информация о связи одного резюме и одного навыка, с PK, FK: resume_id: INT, skill_id: INT. “Resume”, “Skill” связаны с “ResumeSkill”

связями 1(1) – M(0), так как в резюме может быть как 0, так и много записей о навыке и у навыка может быть как 0, так и много записей о резюме, при этом если запись есть, то она относится к одному навыку и одному резюме.

“Post” – информация о должности, с PK id: INT, и поле name: VARCHAR. Связи с “Резюме” и “Вакансия” остались такими же, как и в ER-модели.

“Vacancy” – информация о вакансии, с PK: id: INT, а также FK: post_id: INT, employer_id: INT, и полями: name: VARCHAR, salary: INT, address: VARCHAR, experience: INT, comment: VARCHAR, status: ENUM.

Связь между “Резюме” и “Работодатель” осталась такой же, путем добавления к “Vacancy” внешнего ключа employer_id.

“Requirement” – информация о требованиях, с PK id: INT, и полем description: VARCHAR.

“Condition” – информация об условиях, с PK: id: INT, и полем description: VARCHAR.

Так как “Требование” и “Условие” было связано с “Вакансия” связью M – N, то по аналогии с “WorkerEducation” добавлены таблицы:

“VacancyRequirement” – информация о записях требований в вакансиях, с PK, FK: vacancy_id: INT, requirement_id: INT.

“VacancyCondition” – информация о записях условий в вакансиях, с PK, FK: vacancy_id: INT, condition_id: INT.

Данные таблицы представляют собой записи о связях конкретных вакансий с конкретными требованиями/условиями, при этом для каждой вакансии, требования/условия может быть как 0, так и несколько записей, поэтому связи к таблицам M(0), а от данных таблиц 1(1).

“Field of activity” – информация о сферах деятельности работодателей, с PK id: INT, и полем name: VARCHAR.

“Employer” – информация о работодателях, с PK id: INT, а также FK activity_id: INT, и полями: name: VARCHAR, photo(url): VARCHAR, description: VARCHAR.

Связь между “Сфера активности” и “Работодатель” осталась такой же, путем добавления к “Employer” внешнего ключа activity_id.

“PresentationFile” – информация о файлах презентаций работодателей, с PK id: INT, а также FK employer_id: INT, и полями name: VARCHAR, url: VARCHAR.

Связь между “Файл презентации” и “Работодатель” осталась такой же, путем добавления к “Presentation File” внешнего ключа employer_id.

“Workbook Writing” – информация о записи в трудовую книжку, с PK id: INT, и полями type: ENUM, date: DATE.

“WorkerEmployerWriting” – информация о связи записи в трудовую книжку с работодателем и работником с PK, FK writing_id: INT, workbook_id: INT, employer_id: INT.

“Employer” и “Worker” связаны с “WorkerEmployerWriting” связями 1(1) – M(0). Так как одна запись в трудовую книжку обязательно связана с одним работником и одним работодателем, при этом у работодателя и работника может быть как 0, так и много записей в трудовой книжке.

Проверка на нормальные формы:

1) 1НФ: Все атрибуты таблицы должны быть атомарными, в таблице не должно быть повторяющихся групп.

Workbook: id, workbook_number, issue_date – удовлетворяет;

TIN: id, TIN – удовлетворяет;

Passport: id, passport_number, passport_series – удовлетворяет;

Worker: id, passport_id, TIN_id, workbook_id, birthday – удовлетворяет;

Education: id, name, type – удовлетворяет;

WorkerEducation: worker_id, education_id, enroll_date – удовлетворяет;

Resume: id, worker_id, post_id, salary – удовлетворяет;

Skill: id, description – удовлетворяет;

Post: id, name – удовлетворяет;

Vacancy: id, post_id, employer_id, name, salary, address, experience, comment, status – если учесть, что адрес будет храниться одной строкой (не разделяться на части (например страна, город, улица и т.д.)), то удовлетворяет;

Requirement: id, description – удовлетворяет;

VacancyRequirement: vacancy_id, requirement_id – удовлетворяет;

Condition: id, description – удовлетворяет;

VacancyCondition: vacancy_id, condition_id – удовлетворяет;

Employer: id, activity_id, name, photo(url), description – удовлетворяет;

Field of activity: id, name – удовлетворяет;

Presentation file: id, employer_id, name, url – удовлетворяет;

Workbook Writing: id, type, date – удовлетворяет.

WorkerEmployerWriting: writing_id, worker_id, employer_id, - удовлетворяет.

2) 2НФ Таблица должна находиться в 1НФ, и каждый неключевой атрибут должен полностью зависеть от всего составного первичного ключа.

WorkerEducation: enroll_date -> worker_id, education_id => удовлетворяет

VacancyRequirement, ResumeSkill, WorkerEmployerWriting имеют только РК => удовлетворяют.

Все остальные таблицы имеют не составной первичный ключ => все удовлетворяют.

3) 3НФ Таблица должна находиться во 2НФ, и ни один неключевой атрибут не должен зависеть от другого неключевого атрибута (нет транзитивных зависимостей).

Функциональные зависимости:

Workbook: id -> workbook_number, issue_date – удовлетворяет;

TIN: id -> TIN – удовлетворяет;

Passport: id -> passport_number, passport_series – удовлетворяет;

Worker: id -> passport_id, TIN_id, workbook_id, birthday – удовлетворяет;

Education: id -> name, type – удовлетворяет;

WorkerEducation: worker_id, education_id -> enroll_date – удовлетворяет;

Resume: id -> worker_id, post_id, salary – удовлетворяет;

Skill: id -> description – удовлетворяет;

Post: id -> name – удовлетворяет;

Vacancy: id -> post_id, employer_id, name, salary, address, experience, comment, status – если учесть, что адрес будет храниться одной строкой (не разделяться на части (например страна, город, улица и т.д.)), то удовлетворяет;

Requirement: id -> description – удовлетворяет;

VacancyRequirement: vacancy_id, requirement_id – удовлетворяет;

Condition: id -> description – удовлетворяет;

VacancyCondition: vacancy_id, condition_id – удовлетворяет;

Employer: id -> activity_id, name, photo(url), description – удовлетворяет;

Field of activity: id, name – удовлетворяет;

Presentation file: id -> employer_id, name, url – удовлетворяет;

Workbook Writing: id -> type, date – удовлетворяет.

WorkerEmployerWriting: writing_id, worker_id, employer_id, -
удовлетворяет.

4) НФБК Таблица находится в 3НФ, и любой потенциальный ключ должен однозначно определять все остальные атрибуты.

Возможные нарушения (таблицы, в которых несколько потенциальных ключей) могли бы быть в таблицах:

TIN: id -> TIN; TIN -> id – и id и TIN являются потенциальными ключами
=> удовлетворяет;

Passport: id -> passport_number, passport_series; passport_number, passport_series -> id – и id и passport_number, passport_series являются потенциальными ключами => удовлетворяет;

Workbook: id -> workbook_number; workbook_number -> id – и id и workbook_number являются потенциальными ключами => удовлетворяет;

Skill: id -> description; description -> id – и id и description являются потенциальными ключами => удовлетворяет;

Requirement: id -> description; description -> id – и id и description являются потенциальными ключами => удовлетворяет;

Condition: id -> description; description -> id – и id и description являются потенциальными ключами => удовлетворяет;

Post: id -> name; name -> id – и id и name являются потенциальными ключами => удовлетворяет;

Описанная структура бд изображена на рис.2.

Ссылку на pull-request см. в Приложении А.

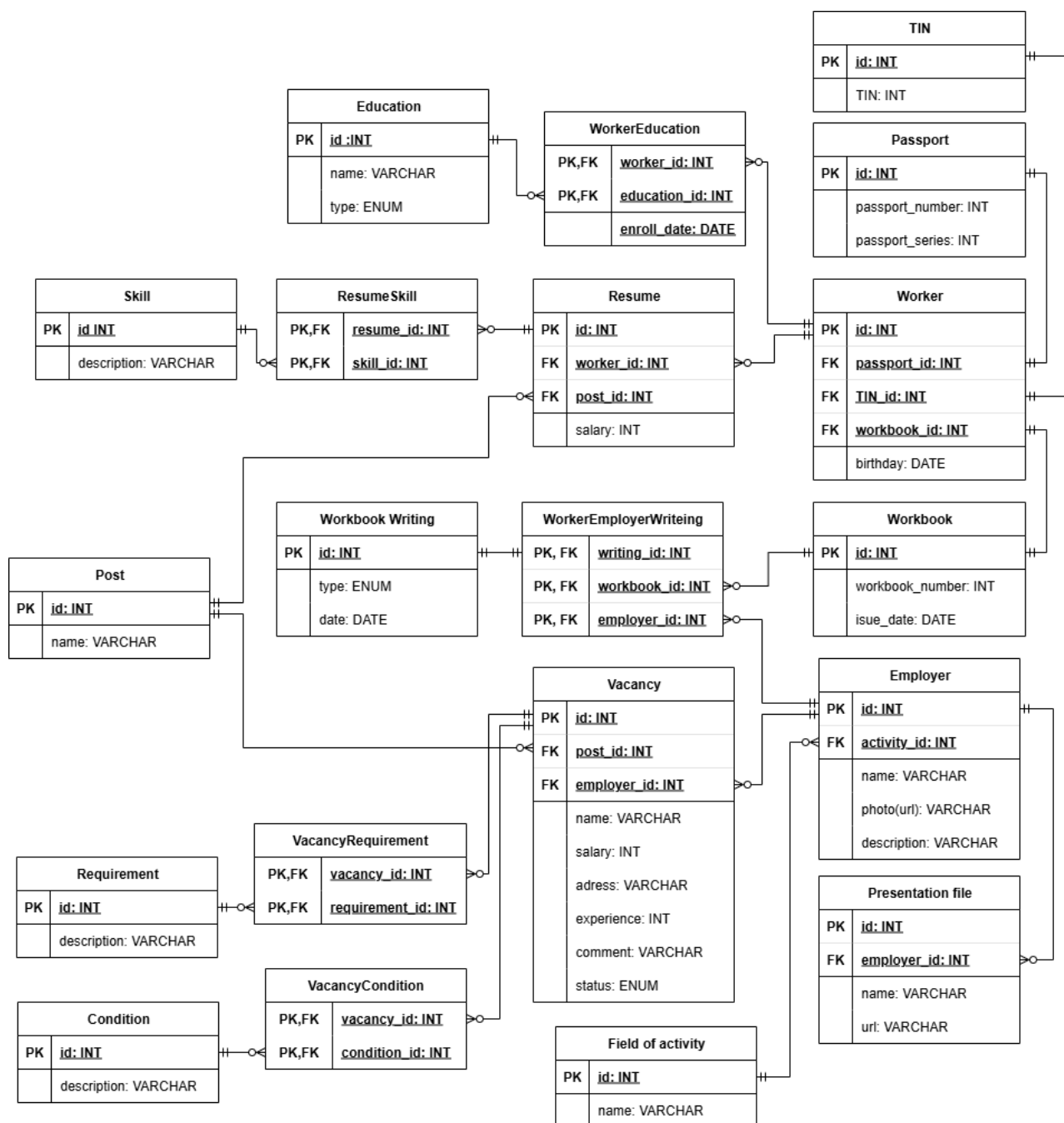


Рисунок 2 – Структура бд

Выводы.

Построена ER-модель и структура бд, по текстовому описанию предметной области, в которых реализованы связи между работниками, работодателями, образованием, документами, резюме, вакансиями, навыками,

требованиями, условиями, файлами презентаций и сферами деятельности. Кроме этого структура проверена на принадлежность к НФБК.

ПРИЛОЖЕНИЕ А
ССЫЛКА НА PULL REQUEST

<https://github.com/moevm/sql-2025-3384/pull/6>