

**Міністерство освіти і науки України**  
**Національний технічний університет України «Київський політехнічний**  
**інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
  
**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 2 з дисципліни  
«Бази даних»

**„Створення бази даних. Користувачі, ролі, права ”**

**Виконав(ла)**

*ІП-15 Мешков Андрій Ігорович*  
(шифр, прізвище, ім'я, по батькові)

**Перевірив**

*Ліщук Олександр*  
(прізвище, ім'я, по батькові)

Київ 2022

## **Лабораторна робота № 2**

### **Створення бази даних. Користувачі, ролі, права**

#### **Мета:**

- Створення бази даних шляхом визначення схеми БД та заповнення її тестовими даними
- Навчитися проектувати бази даних, вводити і редагувати структуру таблиць та дані в таблицях
- Вивчити DDL-команди SQL для роботи з таблицями (створення, модифікації та видалення таблиць)
- Вивчити використовувані в SQL засоби для підтримки цілісності даних та їх практичне застосування
- Вивчити основні принципи керування обліковими записами та ролями

#### **Теоретичні основи**

Викладені в лекційному матеріалі

#### **Постановка задачі лабораторної роботи №2**

При виконанні лабораторної роботи необхідно виконати наступні дії:

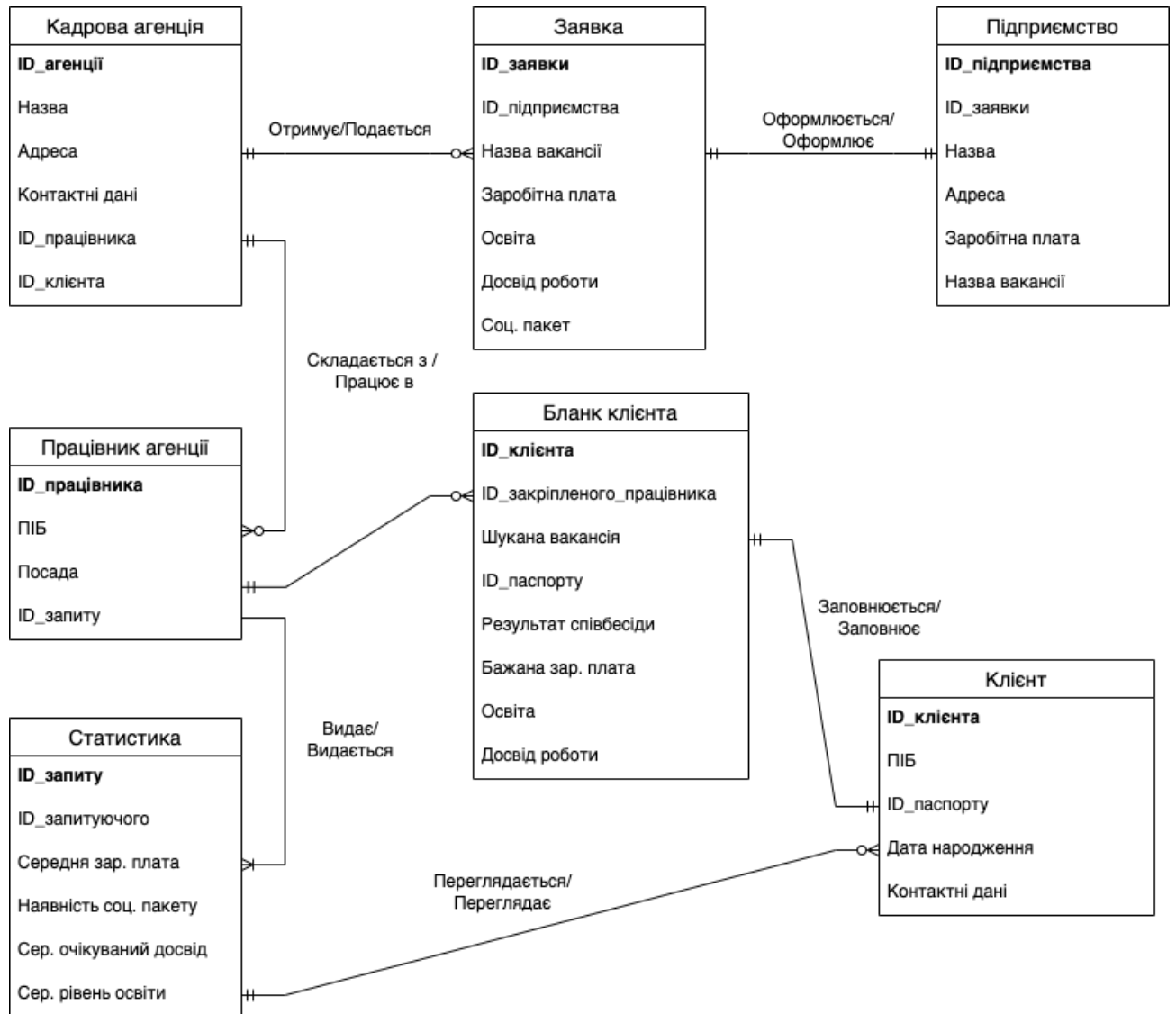
- 1) Створити схему БД згідно з розробленою в роботі No1 ER-моделлю
- 2) Розробити SQL-скрипти для:
  - a. створення таблиць в БД засобами мови SQL. Передбачити обмеження для підтримки цілісності та коректності даних;
  - b. зміни в структурах таблиць, обмежень засобами мови SQL (до 10 різних за суттю запитів для декількох таблиць);
  - c. видалення окремих елементів таблиць/обмежень або самих таблиць засобами мови SQL (до 10 різних за суттю команд);
  - d. встановлення зв'язків між таблицями засобами мови SQL;

- e. визначити декілька (2-3) типів користувачів, котрі будуть працювати з базою даних. Для кожного користувача визначити набір привілеїв, котрі він буде мати;
  - f. для типу користувача створити відповідну роль та наділити її необхідними привілеями;
  - g. створити по одному користувачу в базі даних для кожного типу та присвоїти їм відповідні ролі.
- 3) Згенерувати схему даних засобами СУБД
  - 4) Імпортувати дані в створену БД з використанням засобів СУБД

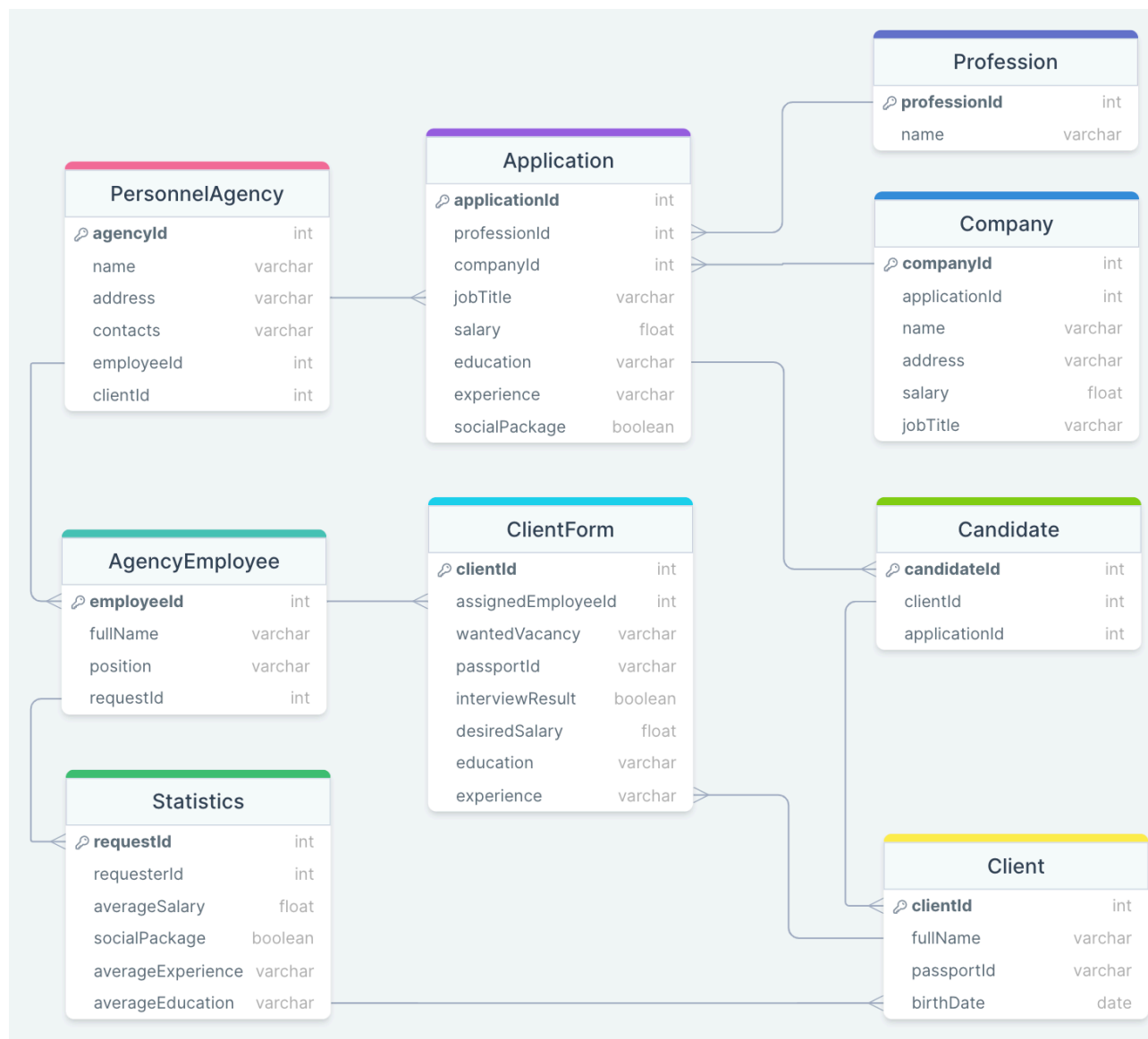
### Варіант 18

**Кадрова агенція.** Кадрова агенція займається підбором персоналу для підприємств і пошуком вакансій для людей. При здійсненні підбору персоналу підприємство звертається в кадрову агенцію і оформляє заявку на пошук працівника. У заявці вказується назва вакансії, професійні вимоги до претендента, пропонований рівень заробітної плати, наявність соціального пакету та інше. Працівники агенції розміщують заяву на сайті агенції, подають оголошення у відповідні друковані видання, вказуючи для контактів свої робочий телефон і e-mail. Претенденти на вакансію звертаються в кадрову агенцію, де проходять початкову співбесіду. Якщо працівник кадрової агенції вважає, що претендент задовольняє виставленим вимогам, то він реєструє його дані (ПІБ, контактний телефон або e-mail) в базі даних і повідомляє їх підприємству. При здійсненні підбору вакансій клієнт заповнює бланк, в якому вказує свої професійні можливості з переліку професійних вимог працедавців, паспортні дані, бажаний рівень заробітної плати. Працівник агенції виконує пошук вакансій по вказаних параметрах, при знаходженні вакансії він повідомляє клієнту контактні дані працедавця. Серед іншого адміністрація кадрової агенції може видавати статистику про середній рівень пропонованої заробітної плати по професіях, по конкретному підприємству і так далі.

## ER-діаграма предметної області



## Схема бази даних



## SQL-скрипти

а. створення таблиць в БД засобами мови SQL. Передбачити обмеження для підтримки цілісності та коректності даних;

```
CREATE TABLE IF NOT EXISTS PersonalAgency(  
    agencyId INT AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    contacts VARCHAR(255) NOT NULL,  
    employeeId INT NOT NULL,  
    clientId INT NOT NULL  
) ENGINE=INNODB;  
  
CREATE TABLE IF NOT EXISTS Application(  
    applicationId INT AUTO_INCREMENT PRIMARY KEY,  
    professionId INT NOT NULL,  
    companyId INT NOT NULL,  
    jobTitle VARCHAR(255) NOT NULL,  
    salary FLOAT NOT NULL,  
    education VARCHAR(255) NOT NULL,  
    experience VARCHAR(255) NOT NULL,  
    socialPackage BOOLEAN NOT NULL  
) ENGINE=INNODB;  
  
CREATE TABLE IF NOT EXISTS Profession(  
    professionId INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL  
) ENGINE=INNODB;  
  
CREATE TABLE IF NOT EXISTS Company(  
    companyId INT AUTO_INCREMENT PRIMARY KEY,  
    applicationId INT NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    salary FLOAT NOT NULL,  
    jobTitle VARCHAR(255) NOT NULL  
) ENGINE=INNODB;  
  
CREATE TABLE IF NOT EXISTS AgencyEmployee(  
    employeeId INT AUTO_INCREMENT PRIMARY KEY,  
    fullName VARCHAR(255) NOT NULL,  
    position VARCHAR(255) NOT NULL,  
    requestId INT NOT NULL  
) ENGINE=INNODB;
```

```

CREATE TABLE IF NOT EXISTS ClientForm(
    clientId INT AUTO_INCREMENT PRIMARY KEY,
    assignedEmployeeId INT NOT NULL,
    wantedVacancy VARCHAR(255) NOT NULL,
    passportId VARCHAR(255) NOT NULL,
    interviewResult BOOLEAN NOT NULL,
    desiredSalary FLOAT NOT NULL,
    education VARCHAR(255) NOT NULL,
    experience VARCHAR(255) NOT NULL
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Candidate(
    candidateId INT AUTO_INCREMENT PRIMARY KEY,
    clientId INT NOT NULL,
    applicationId INT NOT NULL
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Statistic(
    requestId INT AUTO_INCREMENT PRIMARY KEY,
    requesterId INT NOT NULL,
    averageSalary FLOAT NOT NULL,
    socialPackage BOOLEAN NOT NULL,
    averageExperience VARCHAR(255) NOT NULL,
    averageEducation VARCHAR(255) NOT NULL
) ENGINE=INNODB;

CREATE TABLE IF NOT EXISTS Client(
    clientId INT AUTO_INCREMENT PRIMARY KEY,
    fullName VARCHAR(255) NOT NULL,
    passportId VARCHAR(255) NOT NULL,
    birthDate DATE NOT NULL
) ENGINE=INNODB;

```

б. зміни в структурах таблиць, обмежень засобами мови SQL (до 10 різних за суттю запитів для декількох таблиць);

Додаймо новий стовпець та обмеження у таблицю PersonalAgency:

```
ALTER TABLE PersonalAgency
ADD documents INT NOT NULL,
ADD CONSTRAINT documents_unique UNIQUE (documents);
```

	Field	Type	Null	Key	Default	Extra	
►	agencyId	int	NO	PRI	NULL	auto_increment	
■	title	varchar(255)	NO		NULL		
■	address	varchar(255)	NO		NULL		
■	contacts	varchar(255)	NO		NULL		
■	employeeId	int	NO		NULL		
■	clientId	int	NO		NULL		
■	documents	int	NO	UNI	NULL		

Видаємо обмеження та стовпець:

```
ALTER TABLE PersonalAgency
DROP CONSTRAINT documents_unique,
DROP COLUMN documents;
```

	Field	Type	Null	Key	Default	Extra	
►	agencyId	int	NO	PRI	NULL	auto_increment	
■	title	varchar(255)	NO		NULL		
■	address	varchar(255)	NO		NULL		
■	contacts	varchar(255)	NO		NULL		
■	employeeId	int	NO		NULL		
■	clientId	int	NO		NULL		

Створимо нову таблицю та перейменуємо її:



Tables

- > AgencyEmployee
- > Aplication
- > Candidate
- > Client
- > ClientForm
- > Company
- > Name1
- > PersonalAgency
- > Profession
- > Statistic

Tables

- > AgencyEmployee
- > Aplication
- > Candidate
- > Client
- > ClientForm
- > Company
- > Name2
- > PersonalAgency
- > Profession
- > Statistic

```
CREATE TABLE IF NOT EXISTS Name1(
  nameId INT AUTO_INCREMENT PRIMARY KEY
) ENGINE=INNODB;

ALTER TABLE Name1
RENAME Name2;
```

Змінимо тип даних одного стовпця та тип даних з ім'ям другого:

	Field	Type	Null	Key	Default		Field	Type	Null	Key	Default
▶	nameId	int	NO	PRI	NULL	▶	nameId	int	NO	PRI	NULL
	dig	int	NO		NULL		dig	tinyint	YES		NULL
	let	char(4)	YES		NULL		letter	char(20)	YES		NULL

```
CREATE TABLE IF NOT EXISTS Name2(
  nameId INT AUTO_INCREMENT PRIMARY KEY,
  dig INT NOT NULL,
  let CHAR(4)
) ENGINE=INNODB;

ALTER TABLE Name2
MODIFY dig TINYINT,
CHANGE let letter CHAR(20);
```

с. видалення окремих елементів таблиць/обмежень або самих таблиць засобами мови SQL (до 10 різних за суттю команд);

Створимо нову таблицю з даними та видалимо рядок з потрібними даними:

	id	cDig	
▶	1	123	
	NULL	NULL	

	id	cDig	
▶	NULL	NULL	

```
CREATE TABLE IF NOT EXISTS C(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    cDig INT NOT NULL  
) ENGINE=INNODB;  
  
INSERT INTO C(cDig)  
VALUES(123);  
  
DELETE FROM C  
WHERE id = 1;
```

Видалимо таблицю:

```
DROP TABLE C;
```

▼	Tables
>	AgencyEmployee
>	Aplication
>	Candidate
>	Client
>	ClientForm
>	Company
>	PersonalAgency
>	Profession
>	Statistic

д. встановлення зв'язків між таблицями засобами мови SQL;

```

ALTER Table PersonalAgency
ADD CONSTRAINT FK_PersonalAgency_Application FOREIGN KEY(applicationId)
REFERENCES Application(applicationId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table Application
ADD CONSTRAINT FK_Application_Profession FOREIGN KEY(professionId)
REFERENCES Profession(professionId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table Application
ADD CONSTRAINT FK_Application_Company FOREIGN KEY(companyId)
REFERENCES Company(companyId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table PersonalAgency
ADD CONSTRAINT FK_PersonalAgency_AgencyEmployee FOREIGN KEY(employeeId)
REFERENCES AgencyEmployee(employeeId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table ClientForm
ADD CONSTRAINT FK_ClientForm_gencyEmployee FOREIGN KEY(assignedEmployeeId)
REFERENCES AgencyEmployee(employeeId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table Candidate
ADD CONSTRAINT FK_Candidate_Application FOREIGN KEY(applicationId)
REFERENCES Application(applicationId)
ON UPDATE CASCADE
ON DELETE CASCADE;

```

```

ALTER Table AgencyEmployee
ADD CONSTRAINT FK_AgencyEmployee_Statistics FOREIGN KEY(requestId)
REFERENCES Statistic(requestId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table Statistic
ADD CONSTRAINT FK_Statistic_Client FOREIGN KEY(requesterId)
REFERENCES Client(clientId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table ClientForm
ADD CONSTRAINT FK_ClientForm_Client FOREIGN KEY(clientId)
REFERENCES Client(clientId)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER Table Candidate
ADD CONSTRAINT FK_Candidate_Client FOREIGN KEY(clientId)
REFERENCES Client(clientId)
ON UPDATE CASCADE
ON DELETE CASCADE;

```

е. визначити декілька (2-3) типів користувачів, котрі будуть працювати з базою даних. Для кожного користувача визначити набір привілеїв, котрі він буде мати;

1) Client – має змогу тільки на перегляд.

2) AEmployee – має змогу переглядати та редагувати.

ф. для типу користувача створити відповідну роль та наділити її необхідними привілеями;

```

CREATE ROLE Reader, Administrator;

GRANT SELECT
ON PersonalAgency.*
TO Reader;

GRANT SELECT, INSERT, UPDATE, DELETE
ON PersonalAgency.*
TO Administrator;

```

g. створити по одному користувачу в базі даних для кожного типу та присвоїти їм відповідні ролі.

Створюємо користувача-клієнта.

```
CREATE USER 'Client'@'localhost' IDENTIFIED BY '111';  
GRANT 'Reader' TO 'Client'@'localhost';
```

Бачимо, що не маємо дозволу на деякі команди.

Error Code: 1142. DROP command denied to user 'Client'@'localhost' for table 'profession'

Error Code: 1142. CREATE command denied to user 'Client'@'localhost' for table 'new'

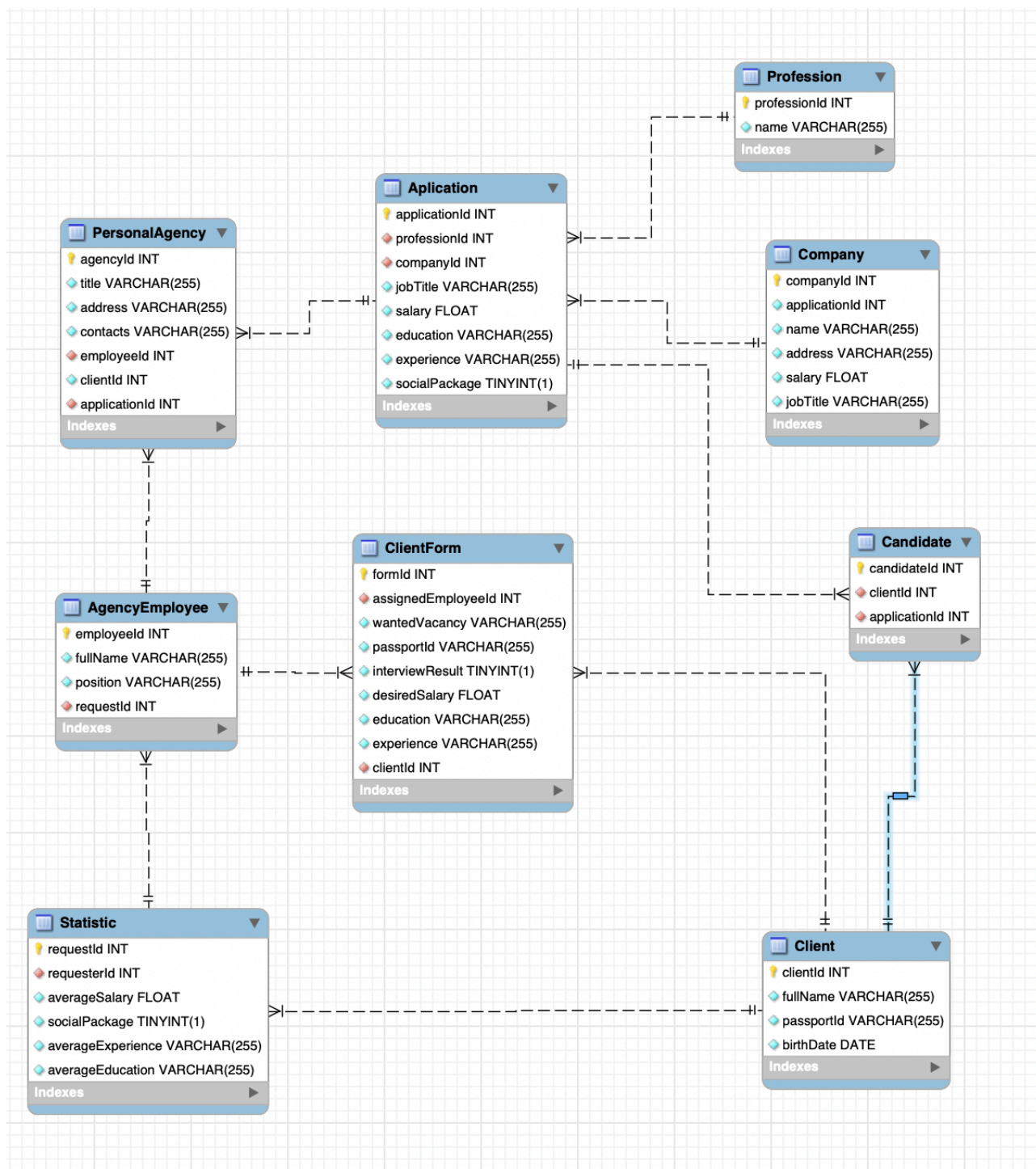
```
CREATE USER 'AEmployee'@'localhost' IDENTIFIED BY '111';  
GRANT 'Administrator' TO 'AEmployee'@'localhost';
```

Бачимо, що не маємо дозволу на деякі команди, але маємо дозвіл на INSERT, DELETE, UPDATE.

	professionId	name	
▶	1	a	
■	NULL	NULL	

Error Code: 1142. CREATE command denied to user 'AEmployee'@'localhost' for table 'test'

## Згенерувати схему даних засобами СУБД



## Імпортувати дані в створену БД з використанням засобів СУБД

Створюємо Excel таблицю з даними та імпортуємо.

A	B	C	D	E	F	G	H
applicationId	professionId	companyId	jobTitle	salary	education	experience	socialPackage
1	12	222	Specialist of technical communication	24000	Higher	3 years	0
2	14	222	Service Engineer of Sony Power Plants	12500	Higher	not	0
3	23	12	Head of the development department 1C	100000	Higher	10 years	1
4	55	20	PPC specialist	40000	Higher	5 years	1

### Table Data Import

**Select File to Import**

Table Data Import allows you to easily import CSV, JSON datafiles.  
You can also create destination table on the fly.

File Path:  Browse...

**BOOTCAMP**

< Back Next > Cancel



