**LAPORAN MINI PROJECT DATABASE**

**PENCARIAN KERJA**



**Kelompok 4**

Anggota :

Azarine Aisyah Ramadhani (2206051550)

Juanita Jovana (2206028604)

Maryesta Apriliani S. (2206051531)

Nadira Eka Rahmaharva (2206051525)

Yiesha Reyhani Ghozali (2206828115)

# 

# **DAFTAR ISI**

# 

[**DAFTAR ISI 2**](#_w80o7ro3ye7y)

[**BAB 1 PENDAHULUAN 3**](#_w33yy6h8g53)

[1.1 Latar Belakang Masalah 3](#_5nrjxdnwm6d6)

[1.2 Tujuan 3](#_9ytqbpjxy59e)

[**BAB 2 DATABASE 4**](#_2kul0o68rbao)

[2.1 Rancangan Database 4](#_cqjfh7iahx0r)

[2.2 ER Diagram 7](#_yi8na9gpb7x7)

[2.3 Tabel Relasional 7](#_yhuh8wvr8tdu)

[**BAB 3 IMPLEMENTASI 11**](#_705h3lb6vdl7)

[3.1 Implementasi SQL 11](#_3wc4f3x2zhcs)

[3.2 Implementasi GUI 17](#_9ay2nz7gums9)

[3.2.1 Package yang Digunakan 17](#_ldaufk3yjbgt)

[3.2.2 Fitur Login 18](#_nwhhv5fzjjk6)

[3.2.3 Fitur Register 20](#_f1vcca28q2tw)

[3.2.3 Menu Utama 22](#_m9qal0vgn8hf)

[3.2.3 Fitur Jobseeker 23](#_d2r1xhiqs6hu)

[3.2.3 Fitur Company 27](#_39iqaipprcfs)

[3.3 Testing 29](#_kxfq7kgqehz6)

[3.3.1 Fitur Login 30](#_7i0o9sxza6ru)

[3.3.2 Fitur Registrasi 30](#_dl22t9nw5p2x)

[3.3.3 Menu Utama 32](#_gv57dum716wm)

[3.3.4 Fitur Jobseeker 32](#_ra57z280n1hb)

[3.3.5 Fitur Company 35](#_wointlg7cr0l)

[**BAB 4 PENUTUP 38**](#_i0i0i3ypjvli)

[**DAFTAR PUSTAKA 39**](#_jqfomeq7etfy)

[**LAMPIRAN 40**](#_d6317nfygc7u)

# 

# **BAB 1 PENDAHULUAN**

## 1.1 Latar Belakang Masalah

Proses rekrutmen merupakan proses yang rumit, menghabiskan banyak waktu, uang, dan juga sumber daya bagi perusahaan untuk mencari dan menyeleksi calon karyawan. Oleh karena itu, diperlukan metode pengelolaan yang efisien untuk mengatur proses perekrutan pegawai. Perkembangan teknologi yang ada membuat perusahaan mulai melakukan perubahan dalam proses perekrutan karyawan. Perusahaan mulai beralih dari prosedur manual ke prosedur digital dengan menggunakan bantuan situs atau aplikasi lowongan kerja dalam mencari dan menyeleksi calon karyawannya. Melalui situs tersebut, pelamar dan perusahaan sama-sama dimudahkan karena dapat mendaftar dan menyeleksi calon karyawannya dengan waktu yang lebih fleksibel dan singkat, serta tidak perlu menyiapkan atau menyimpan dokumen secara manual.

Perubahan prosedur tersebut memerlukan perancangan basis data terintegrasi dan mencakup semua informasi yang diperlukan dalam proses pencarian kerja sehingga proses pencarian kerja dapat berjalan dengan lebih efisien

## 1.2 Tujuan

1. Memfasilitasi para *jobseeker* dengan fitur pencarian kerja yang efisien dan dilengkapi filter yang dapat disesuaikan sesuai kebutuhan.
2. Membantu perusahaan untuk mempromosikan pekerjaan mereka secara efektif kepada calon karyawan potensial yang sesuai kriteria.
3. Mengelola informasi tentang deskripsi pekerjaan, gaji, dan penawaran lainnya oleh perusahaan kepada para *jobseeker*.
4. Membantu jalannya proses wawancara dan alur seleksi dengan menyediakan informasi yang lengkap dan terstruktur berkaitan dengan profil *jobseeker* dan pekerjaan yang ditawarkan perusahaan.

# **BAB 2 DATABASE**

## 2.1 Rancangan Database

Entitas:

1. Company: Untuk menyimpan informasi tentang perusahaan yang memposting pekerjaan
2. Jobseeker: Untuk menyimpan informasi dan data diri tentang pencari kerja
3. Job: Untuk menyimpan informasi mengenai pekerjaan yang ditawarkan oleh perusahaan
4. Resume: Untuk menyimpan informasi resume pelamar kerja
5. Interview: Untuk menyimpan tanggal dan waktu untuk melakukan interview
6. Result: Untuk menyimpan informasi mengenai hasil seleksi oleh perusahaan

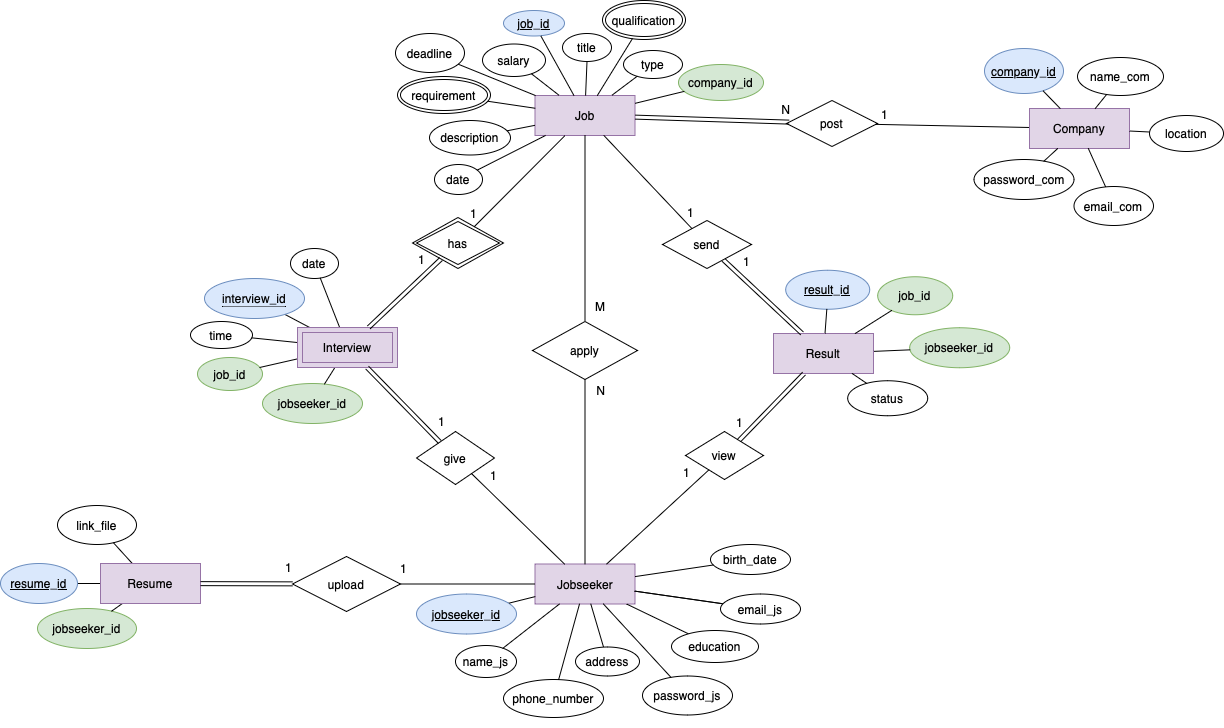
| **Entitas** | **Atribut** | **Key** | **Domain** |
| --- | --- | --- | --- |
| Company | company\_id | Primary Key | Kumpulan digit angka yang merepresentasikan id perusahaan {1, 2, …}. |
| name\_com |  | Kumpulan karakter string yang merepresentasikan nama perusahaan. |
| location |  | Kumpulan karakter string yang merepresentasikan lokasi perusahaan. |
| email\_com |  | Kumpulan karakter yang merepresentasikan email perusahaan. |
| password\_com |  | Kumpulan karakter yang merepresentasikan sandi akun perusahaan. |
| Jobseeker | jobseeker\_id | Primary Key | Kumpulan digit angka yang merepresentasikan id pencari kerja{1, 2, …}. |
| name\_js |  | Kumpulan karakter string yang merepresentasikan nama pencari kerja. |
| phone\_number |  | Kumpulan 10 hingga 12 digit angka yang merepresentasikan nomor telepon pencari kerja. |
| address |  | Kumpulan karakter string yang merepresentasikan alamat tempat tinggal pencari kerja. |
| email\_js |  | Kumpulan karakter yang merepresentasikan email pelamar kerja |
| password\_js |  | Kumpulan karakter yang merepresentasikan sandi akun pencari kerja. |
| education |  | Kumpulan karakter yang merepresentasikan pendidikan terakhir pencari kerja, misalnya Sarjana (*Bachelor’s Degree*). |
| birth\_date |  | Kumpulan tanggal dengan format DD/MM/YYYY yang merepresentasikan tanggal lahir pelamar |
| Job | job\_id | Primary Key | Kumpulan digit angka yang merepresentasikan id pekerjaan {1, 2, …}. |
| title |  | Kumpulan karakter string yang merepresentasikan nama pekerjaan. |
| date |  | Kumpulan tanggal lamaran pekerjaan dibuka dengan format DD/MM/YYYY |
| type |  | Kumpulan karakter string yang merepresentasikan tipe pekerjaan {full time, part time,...} |
| description |  | Kumpulan karakter yang merepresentasikan deskripsi pekerjaan. |
| requirement |  | Kumpulan karakter yang merepresentasikan tanggung jawab pelamar setelah diterima |
| qualification |  | Kumpulan karakter yang merepresentasikan kualifikasi pekerjaan. |
| salary |  | Kumpulan karakter yang merepresentasikan range gaji pekerjaan. |
| company\_id | Foreign Key | Kumpulan digit angka yang merepresentasikan id perusahaan {1, 2, …}. |
| deadline |  | Kumpulan karakter yang merepresentasikan waktu terakhir melamar kerja dengan format DD/MM/YYYY. |
| Resume | resume\_id | Primary Key | Kumpulan digit angka yang merepresentasikan id resume pelamar kerja {1, 2, …}. |
| link\_file |  | Kumpulan karakter yang merepresentasikan link resume pelamar kerja |
| jobseeker\_id | Foreign Key | Kumpulan digit angka yang merepresentasikan id pencari kerja {1, 2, …}. |
| Interview | interview\_id | Primary Key | Kumpulan digit angka yang merepresentasikan id wawancara kerja. |
| job\_id | Foreign Key | Kumpulan digit angka yang merepresentasikan id pekerjaan {1, 2, …}. |
| date |  | Kumpulan tanggal dengan format DD/MM/YYYY yang merepresentasikan tanggal dilakukannya wawancara kerja. |
| time |  | Kumpulan karakter yang merepresentasikan waktu dimulainya wawancara kerja. Contoh: {09.00 WIB, 14.00 WIB, …} |
| jobseeker\_id | Foreign Key | Kumpulan digit angka yang merepresentasikan id pencari kerja {1, 2, …}. |
| Result | result\_id | Primary Key | Kumpulan digit angka yang merepresentasikan id hasil lamaran {1, 2, …}. |
| jobseeker\_id | Foreign Key | Kumpulan digit angka yang merepresentasikan id pencari kerja {1, 2, …}. |
| job\_id | Foreign Key | Kumpulan digit angka yang merepresentasikan id pekerjaan {1, 2, …}. |
| status |  | Kumpulan karakter string yang merepresentasikan status diterima/tidaknya pelamar kerja. |

Rancangan Sistem:

* Suatu *company* dapat mengunggah lebih dari satu pekerjaan ke dalam sistem
* Suatu *job* dapat memiliki beberapa *qualification* dan *requirement*
* *Jobseeker* dapat melamar ke beberapa *job* sekaligus, begitu pula job dapat memiliki beberapa *jobseeker*
* Suatu *job* mengirimkan *result* kepada masing-masing *jobseeker* sesuai dengan diterima atau tidaknya mereka, hanya terdapat satu hasil bagi setiap *jobseeker*
* Suatu *job* mempunyai jadwal *interview* khusus bagi masing-masing *jobseeker*, satu *jobseeker* hanya memiliki satu jadwal *interview* pada sebuah *job*
* *Jobseeker* memberikan *interview* sesuai jadwalnya dan dapat melihat *result* apakah mereka diterima atau tidak
* *Jobseeker* dapat mengunggah *resume* mereka, masing-masing *jobseeker* hanya diperbolehkan mengunggah satu *resume*

## 2.2 ER Diagram

Berdasarkan rancangan database yang telah disampaikan pada bab sebelumnya, maka ER Diagram yang dihasilkan adalah sebagai berikut



Keterangan

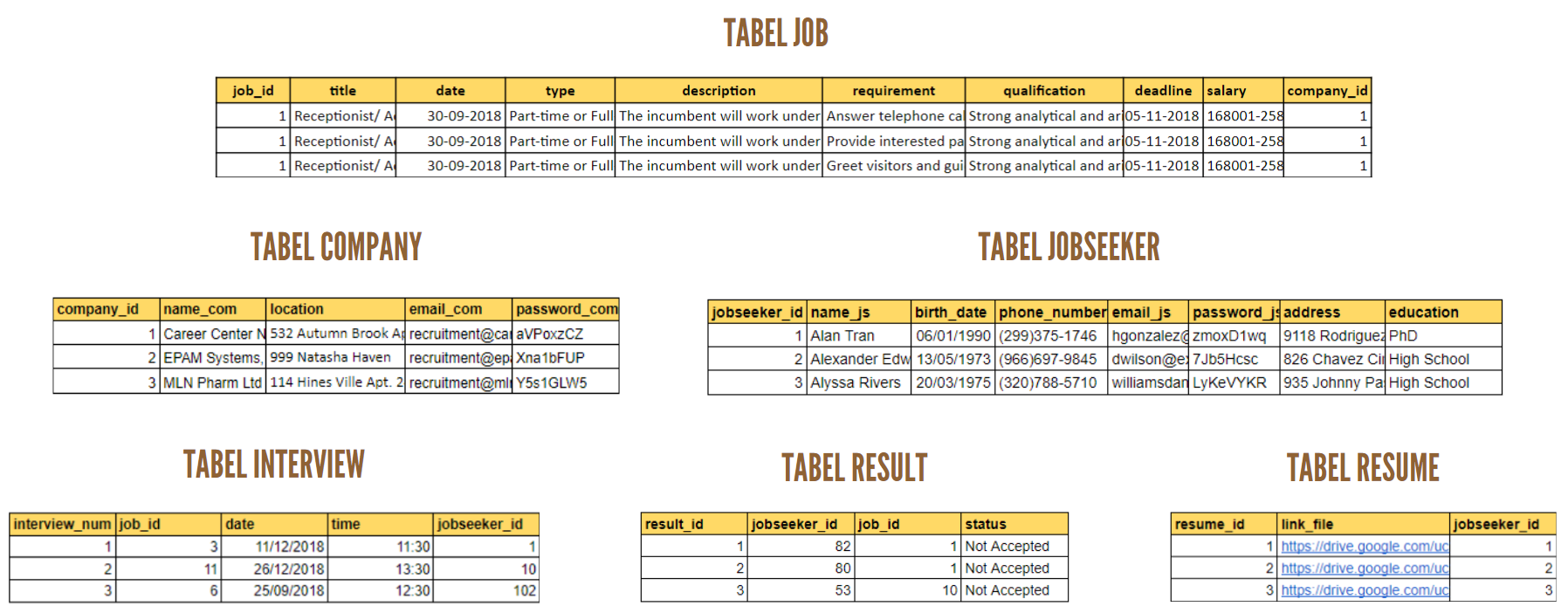
Warna ungu = Entitas

Warna biru = Primary Key

Warna hijau = Foreign Key

## 2.3 Tabel Relasional

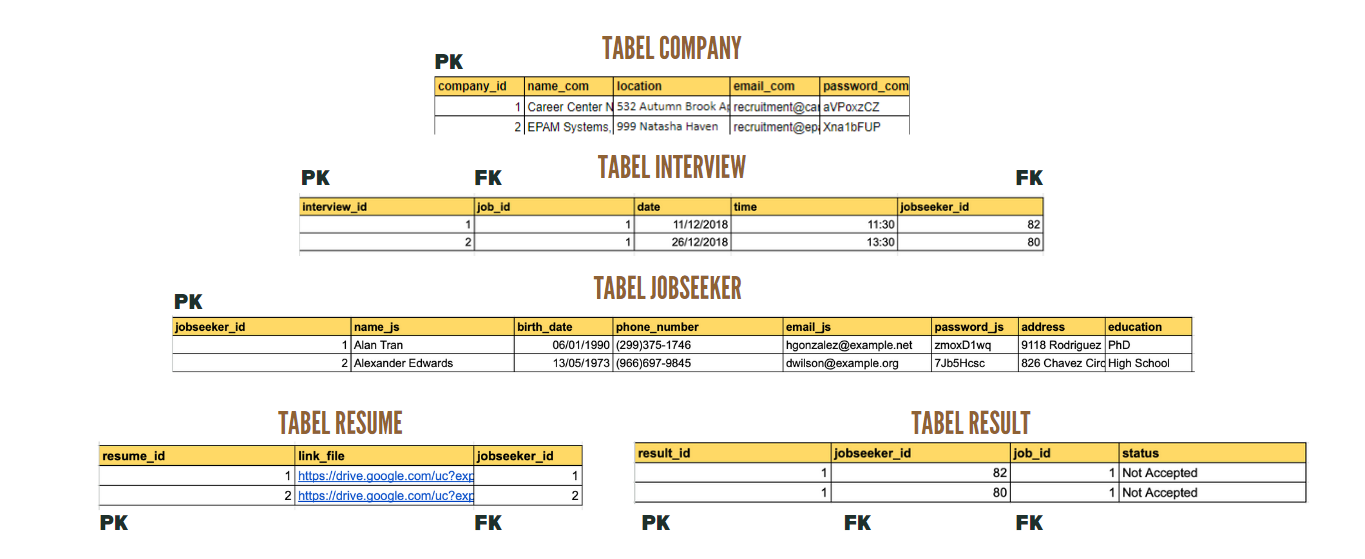
**Tabel 1NF**



Sesuai dengan ER diagram, tabel relasional kami terdiri dari 6 tabel dari tiap entitas. Pada proses pembuatan tabel ini, kami melakukan perubahan pada kolom ‘requirement’ dan ‘qualification’ dari tabel ‘job’ karena kedua kolom tersebut terdiri lebih dari satu nilai (*multivalue*).

**Tabel 2NF**





Tabel 2NF kami sudah sesuai dengan bentuk normal kedua, yakni memenuhi bentuk normal pertama serta atribut yang bukan *key* harus bergantung secara fungsi pada *primary key*. Pada bentuk normal kedua, ketergantungan parsial atribut bukan *key* terhadap *primary key* harus dihilangkan.

Kami mendekomposisi tabel dan menghasilkan beberapa tabel baru untuk mengatasi ketergantungan parsial yang ada. Tabel tersebut adalah tabel ‘qualification’, tabel ‘requirement’, tabel ‘bridge job-qualification’, tabel ‘bridge job-requirement’, dan tabel ‘bridge job-jobseeker’. Hal tersebut dilakukan karena terdapat hubungan *many to many* dari kolom ‘qualification’ dan kolom ‘requirement’ sehingga ditemukan ketergantungan parsial di antara keduanya. Masing-masing dari kolom ‘qualification’ dan kolom ‘requirement’ membentuk tabel baru beserta *primary key* di masing-masing tabel, yakni ‘qualif\_id’ untuk mengidentifikasi kualifikasi *jobseeker* dan ‘req\_id’ untuk mengidentifikasi persyaratan *jobseeker*.

Selain itu, terdapat juga tabel *bridge*. Masih berkaitan dengan sebelumnya terkait mengatasi ketergantungan parsial yang ada, tabel ‘bridge job-qualification’ dibuat untuk menghubungkan antara tabel job dengan tabel qualification dan tabel ‘bridge job-requirement’ dibuat untuk menghubungkan antara tabel job dengan tabel requirement. Sementara itu, tabel ‘bridge job-jobseeker’ dibuat untuk menghubungkan antara tabel job dengan tabel jobseeker. Oleh karena itu, pada tabel bridge hanya berisi *foreign key*.

**Tabel 3NF**

Tabel 2NF kami pada bagian sebelumnya sudah memenuhi bentuk normal ketiga, yakni atribut yang bukan *primary key* tidak memiliki ketergantungan transitif pada *primary key*. Namun pada tabel 3NF dibawah, kami mengintegrasikan kolom ‘email’ dan ‘password’ dari tabel ‘company’ dan ‘jobseeker’ menjadi satu tabel baru, yaitu tabel ‘account’ yang terdiri dari kolom ‘account\_id” sebagai *primary key* serta kolom ‘email’ dan ‘password’. Hal tersebut dilakukan untuk memudahkan implementasi GUI. Secara keseluruhan terdapat 12 tabel, 3 diantaranya merupakan tabel bridge yang menghubungkan antara 3 *primary key.*



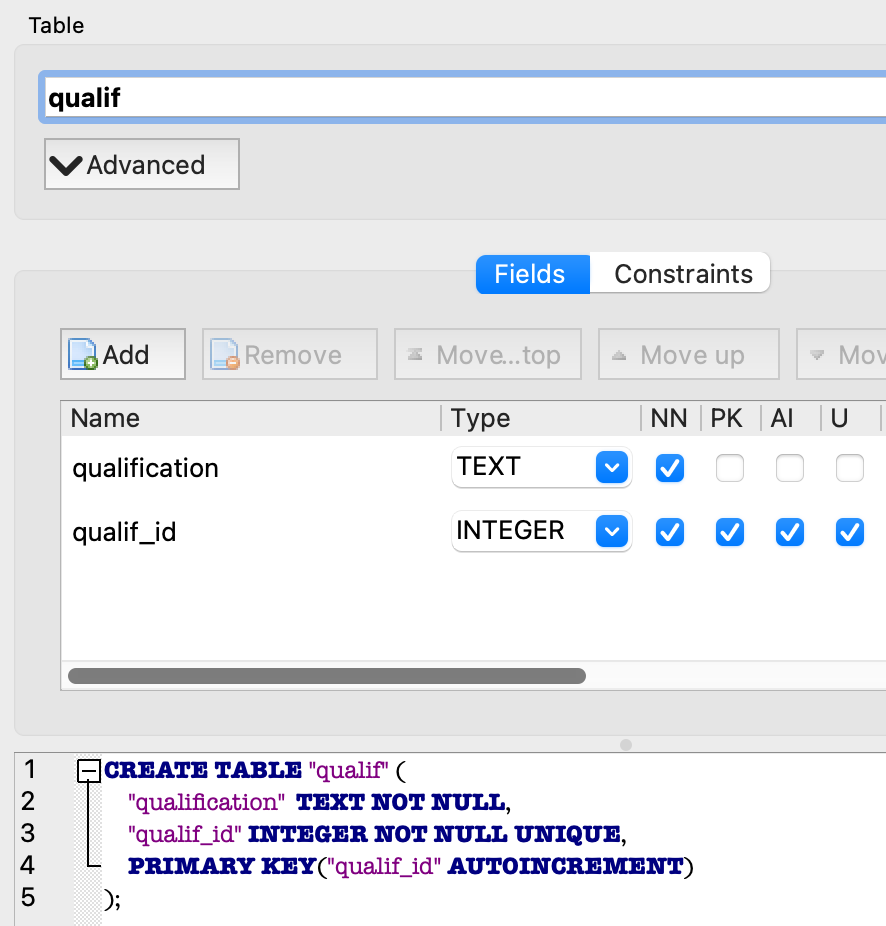
# **BAB 3 IMPLEMENTASI**

## 3.1 Implementasi SQL

**CREATE TABLE**

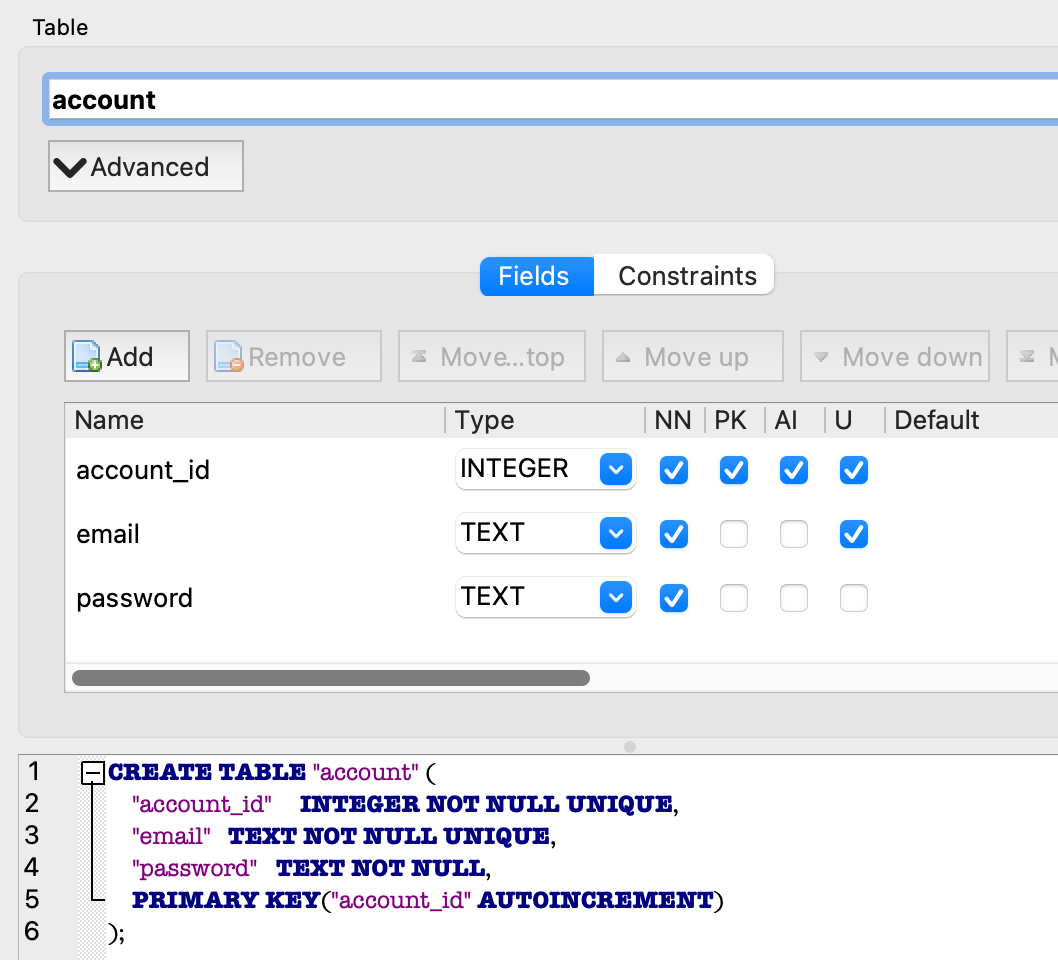
Membuat tabel baru pada database sesuai dengan tabel relasional 3NF

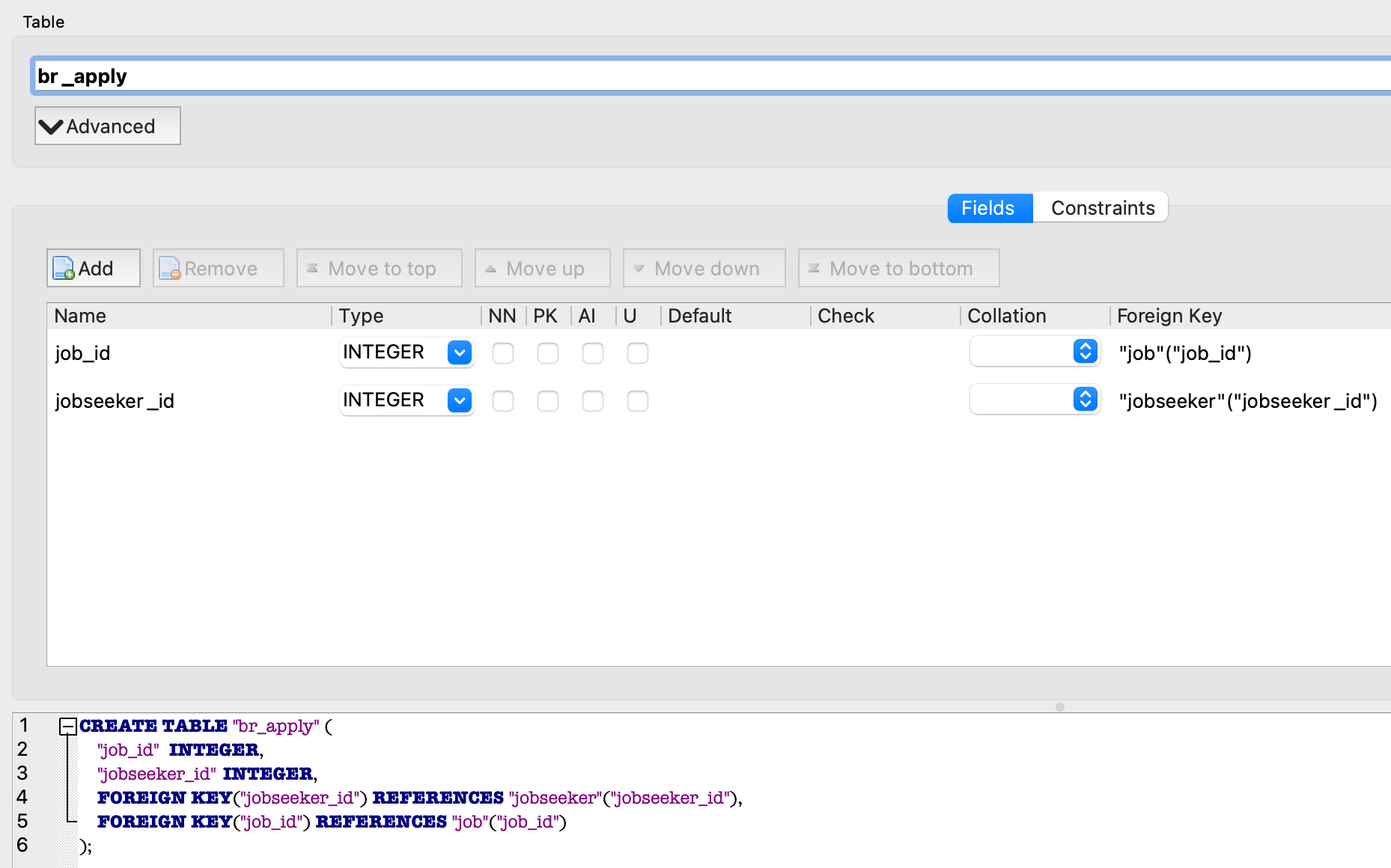
Tabel Qualification

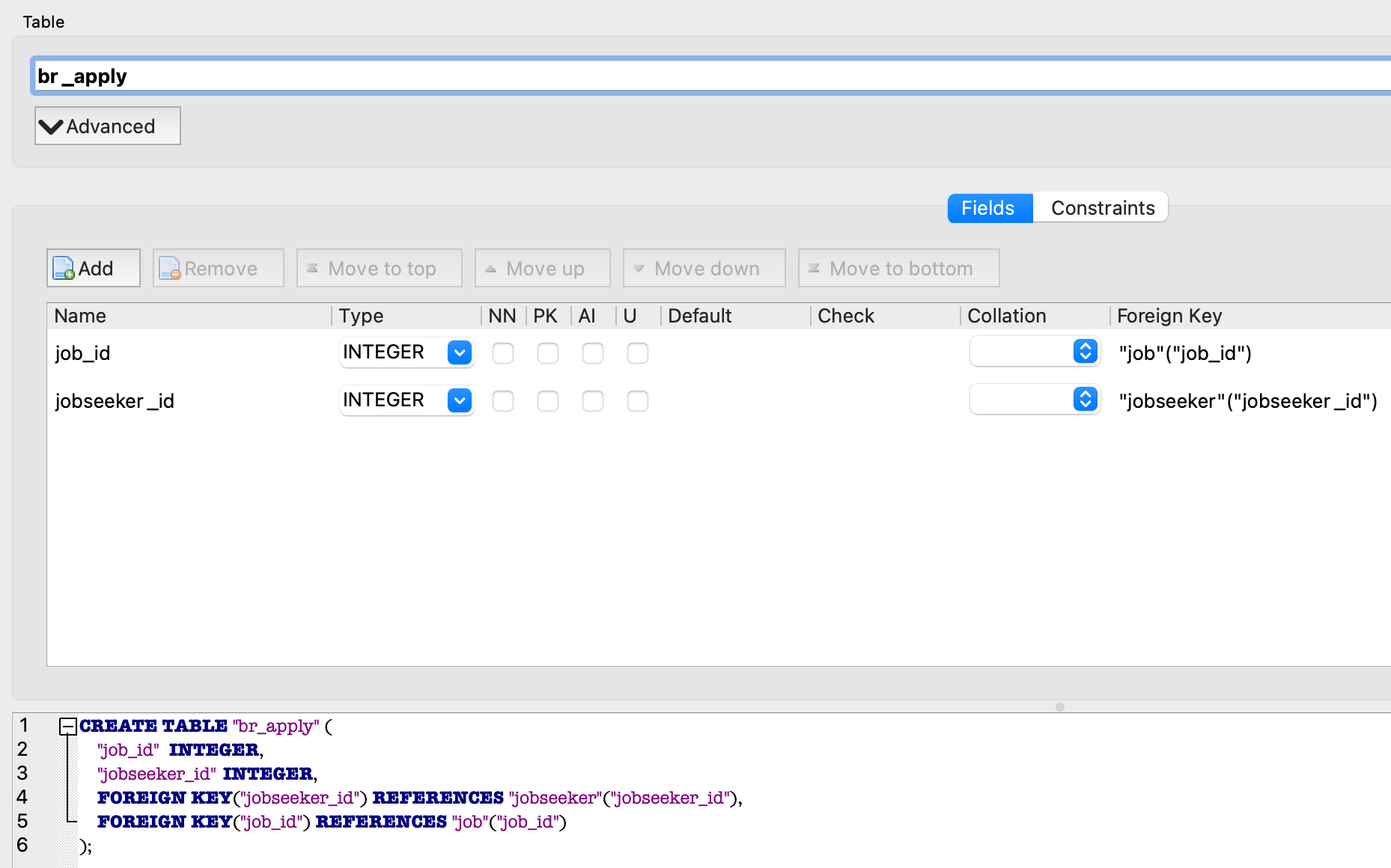


Tabel Bridge Job-Jobseeker

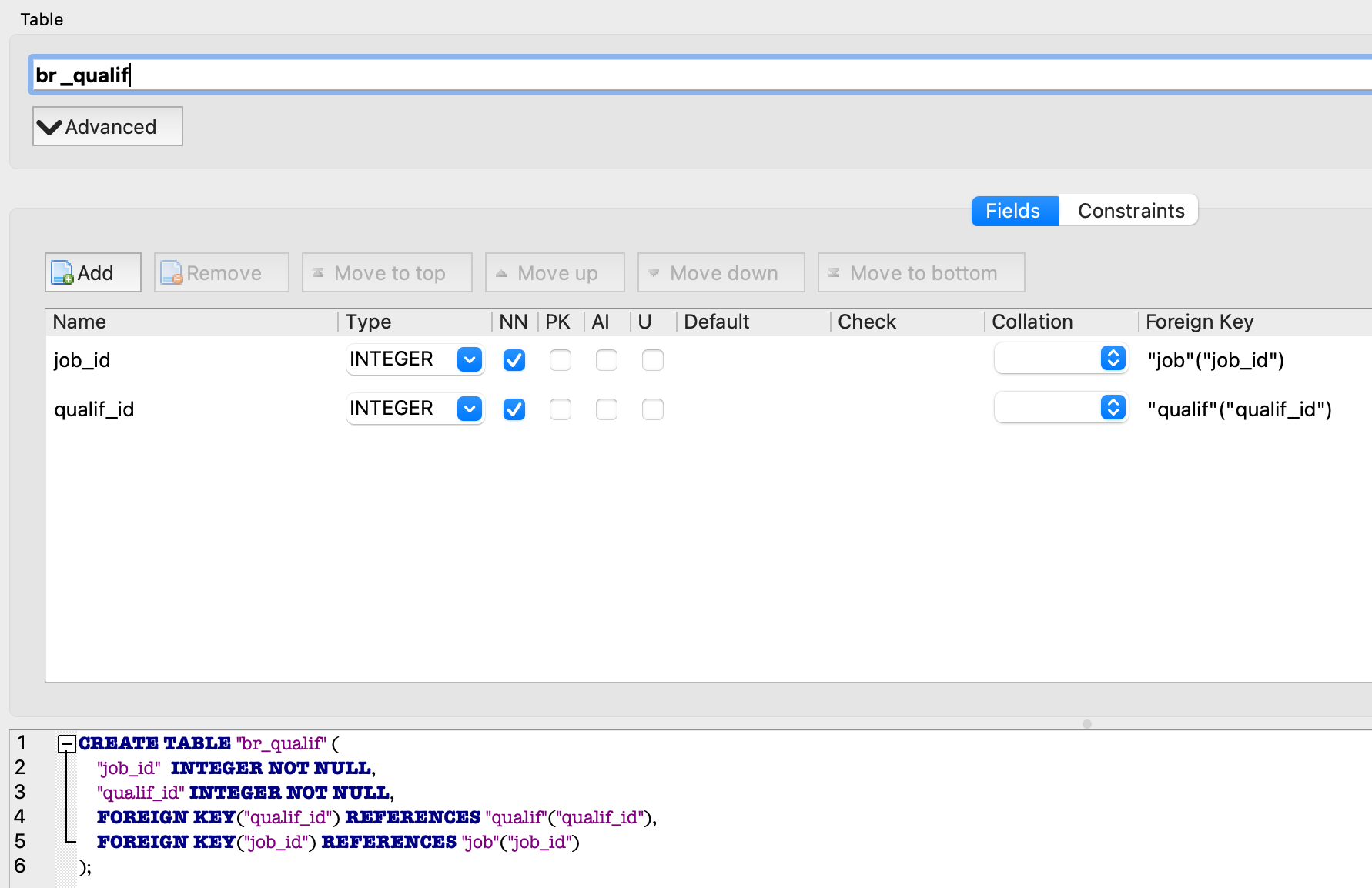
Tabel Account

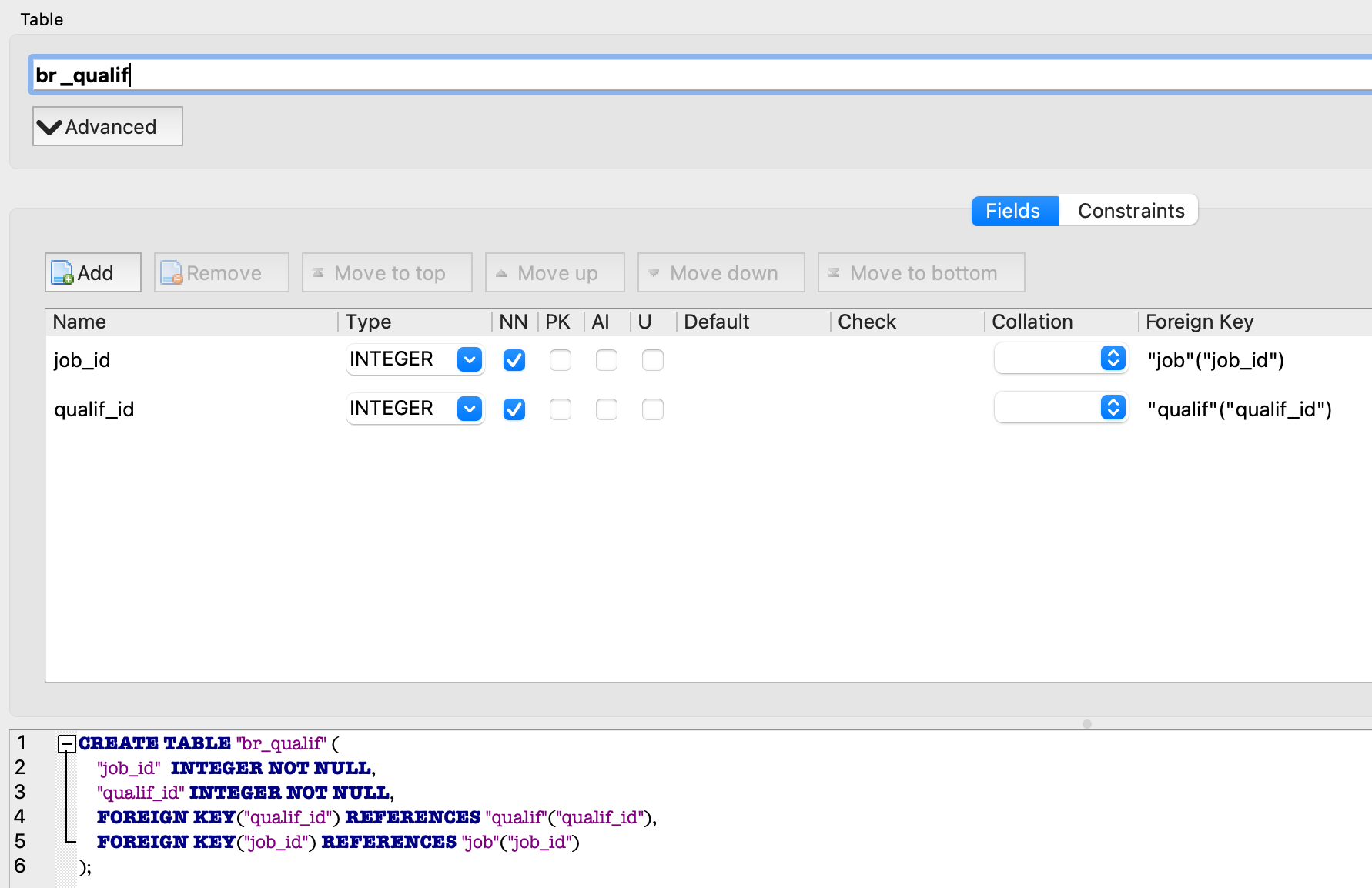




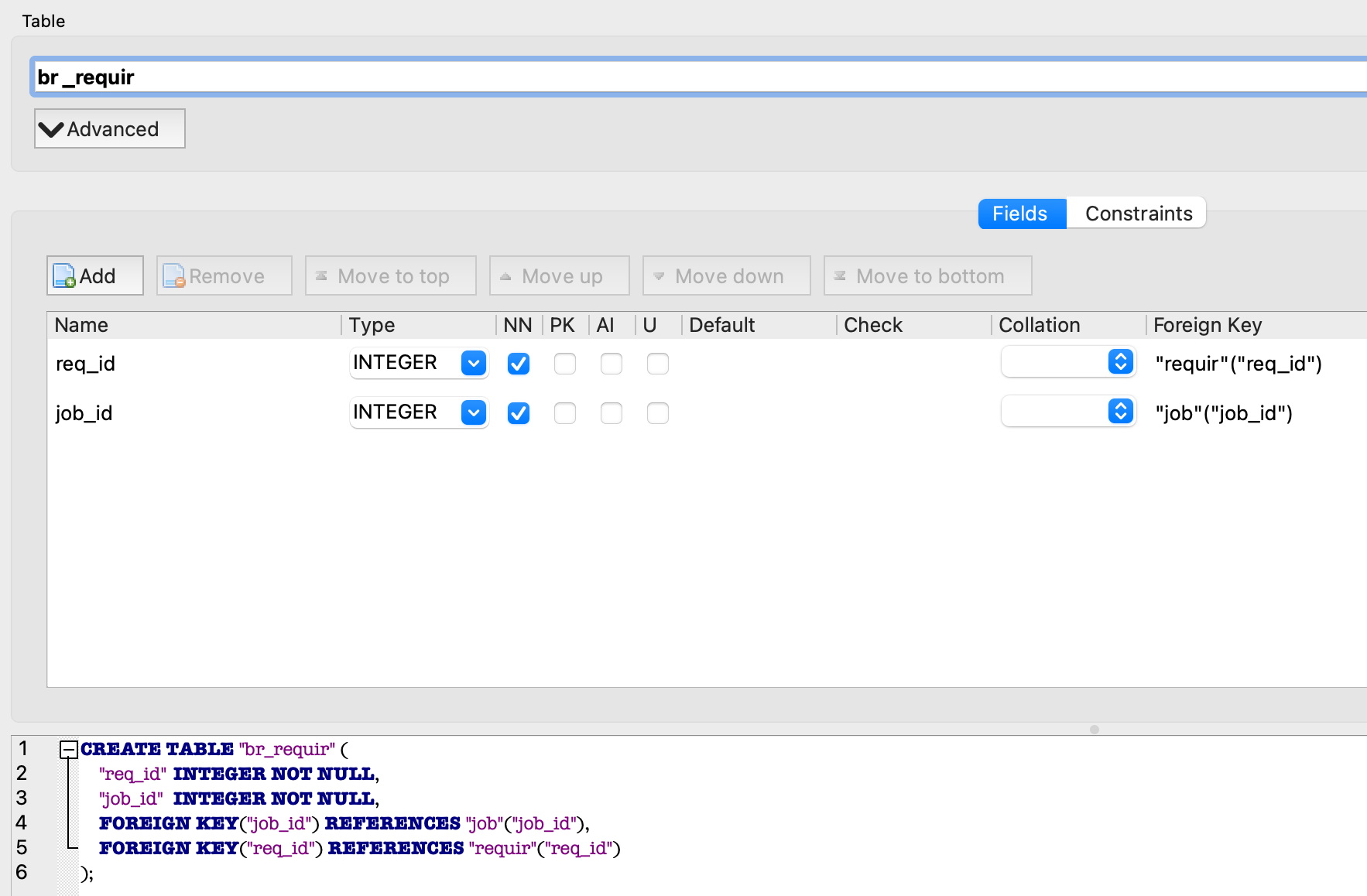


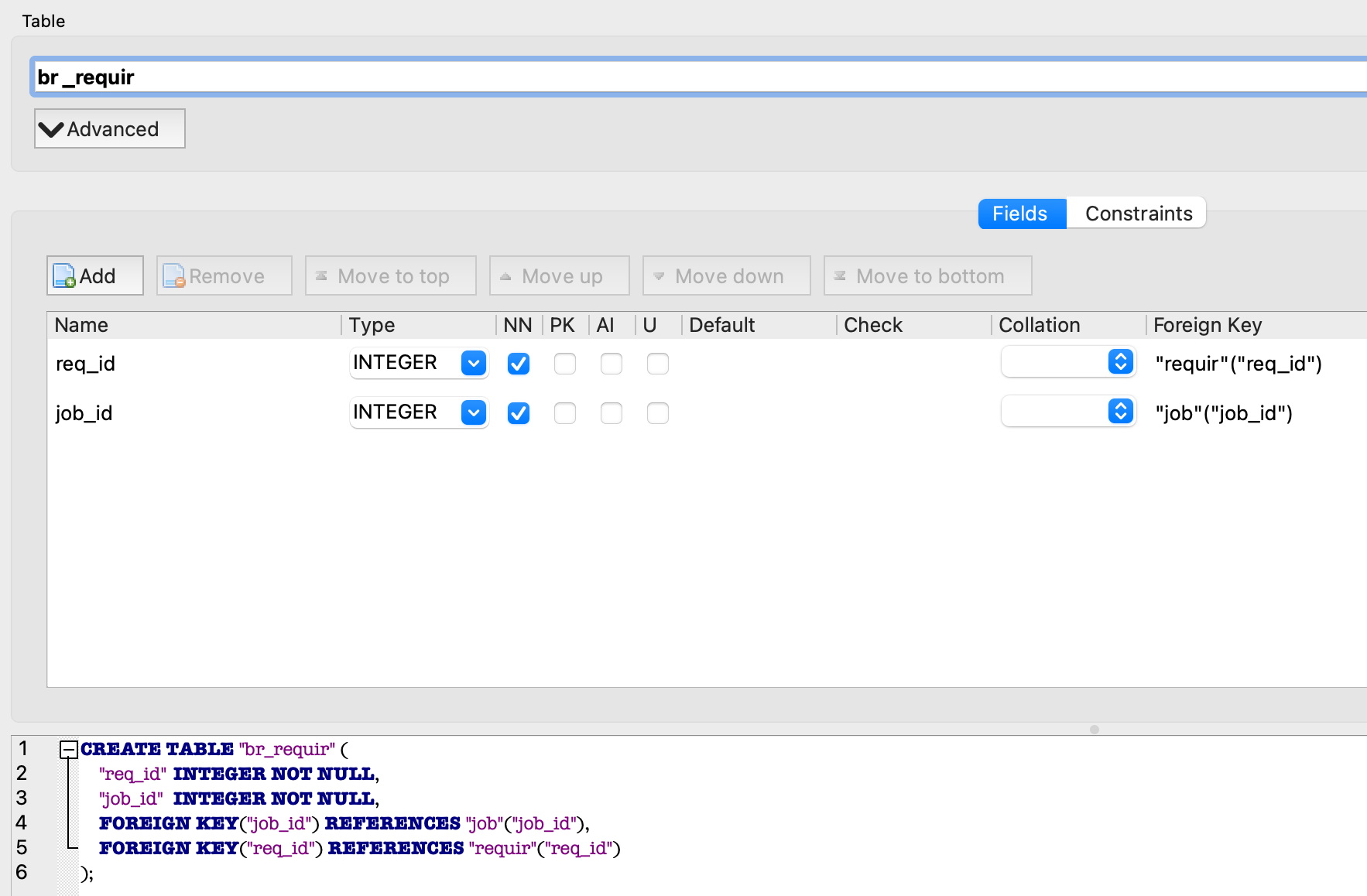
Tabel Bridge Job-Qualification



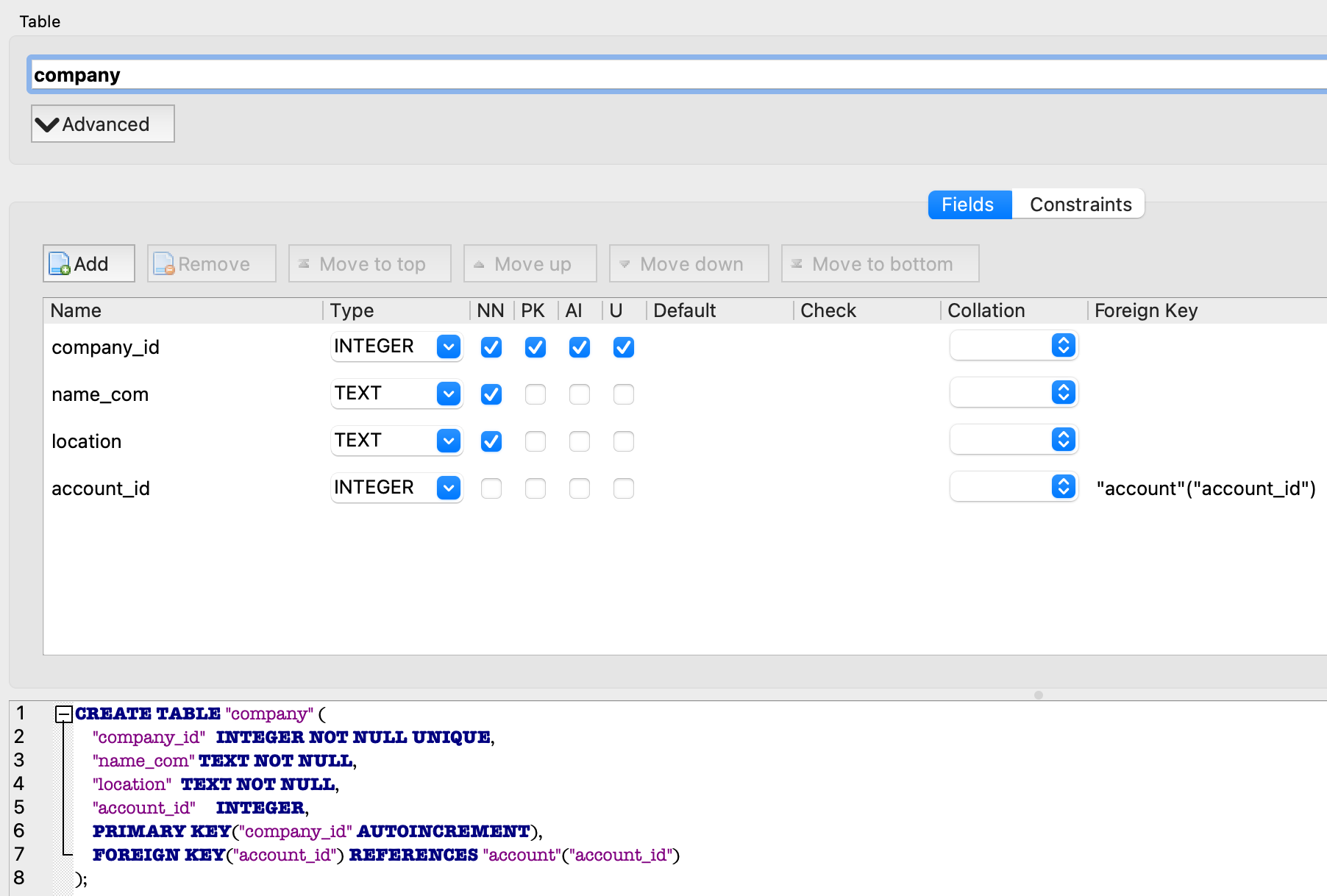


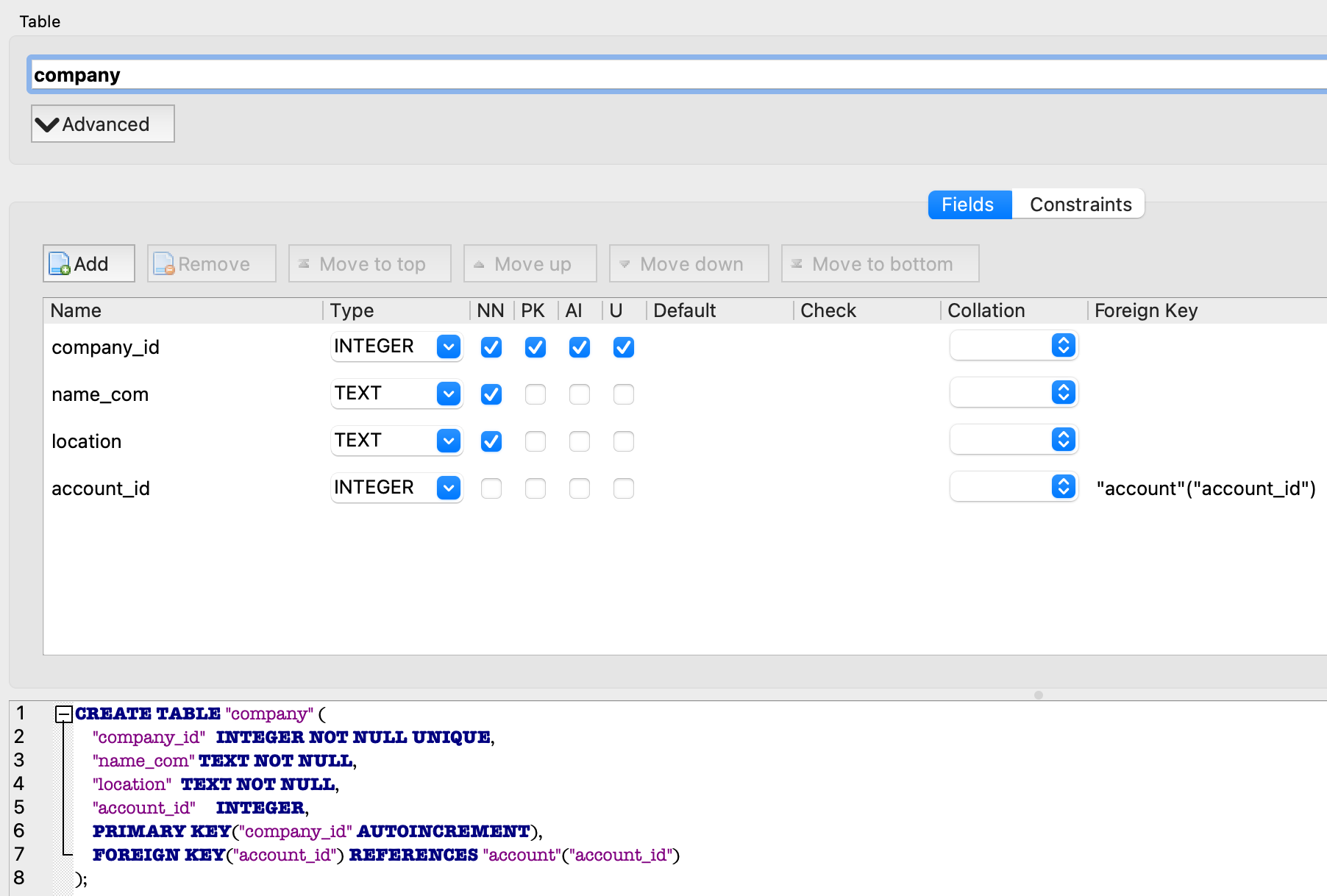
Tabel Bridge Job-Requirement





Tabel Company

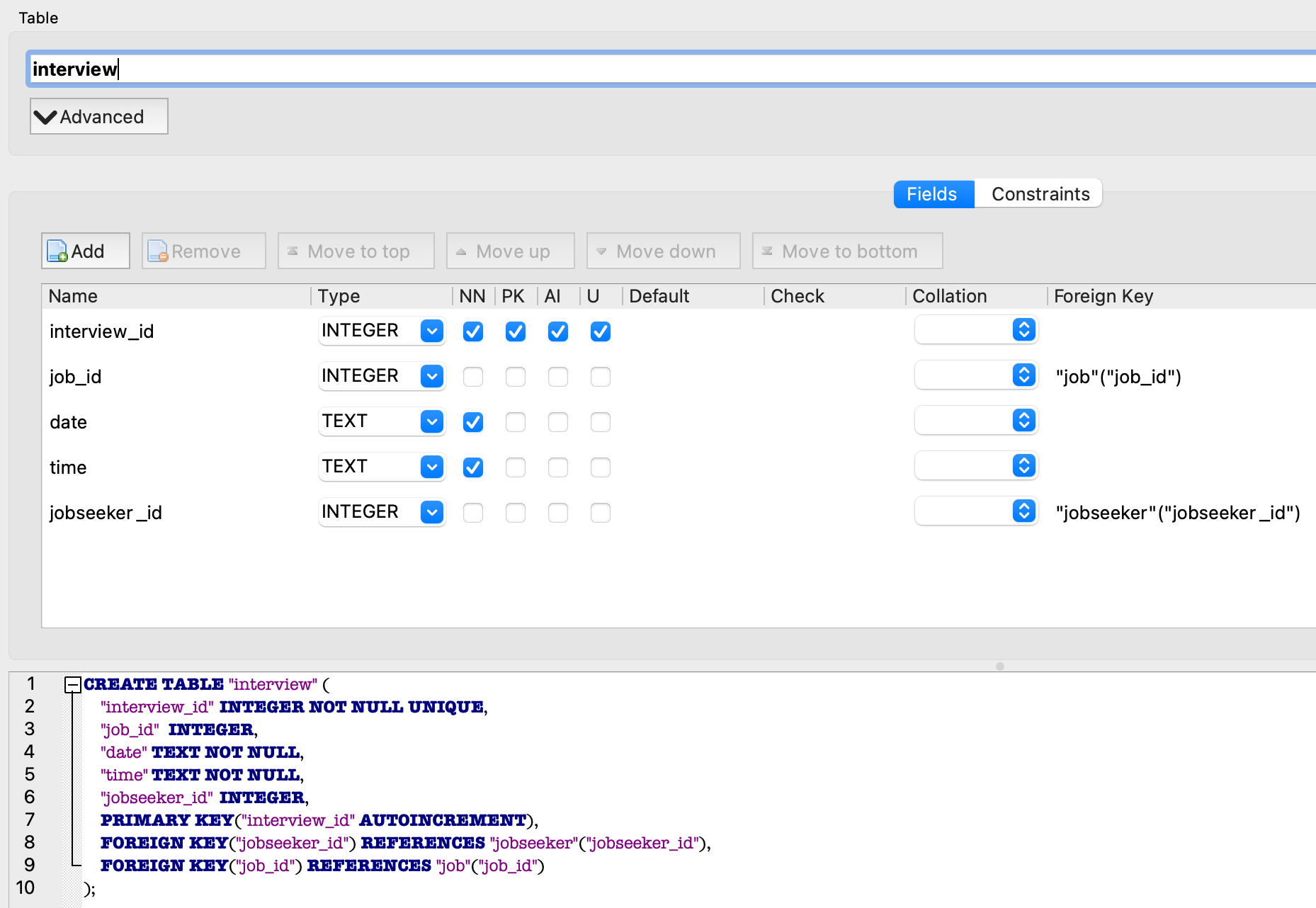


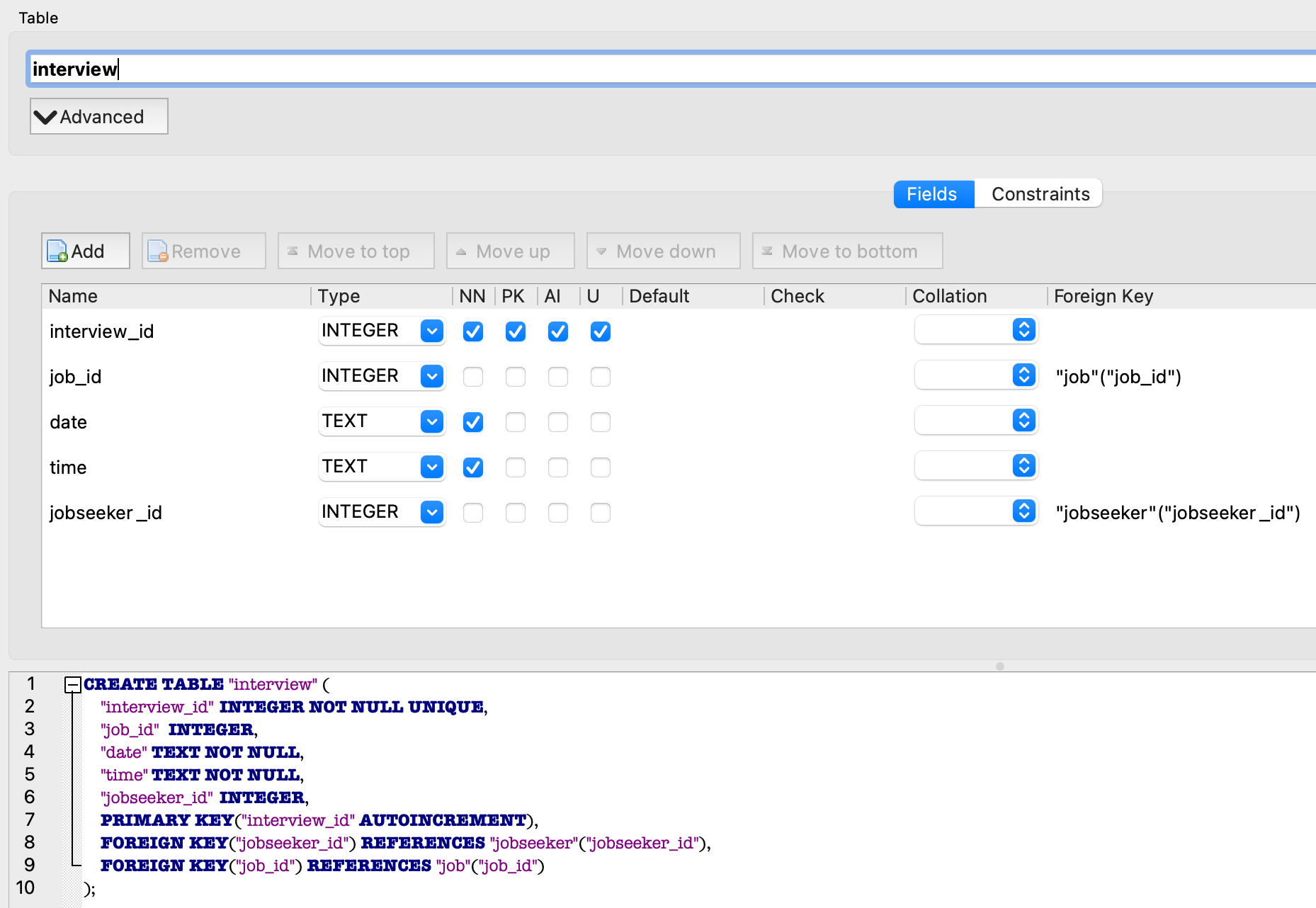


Tabel Job

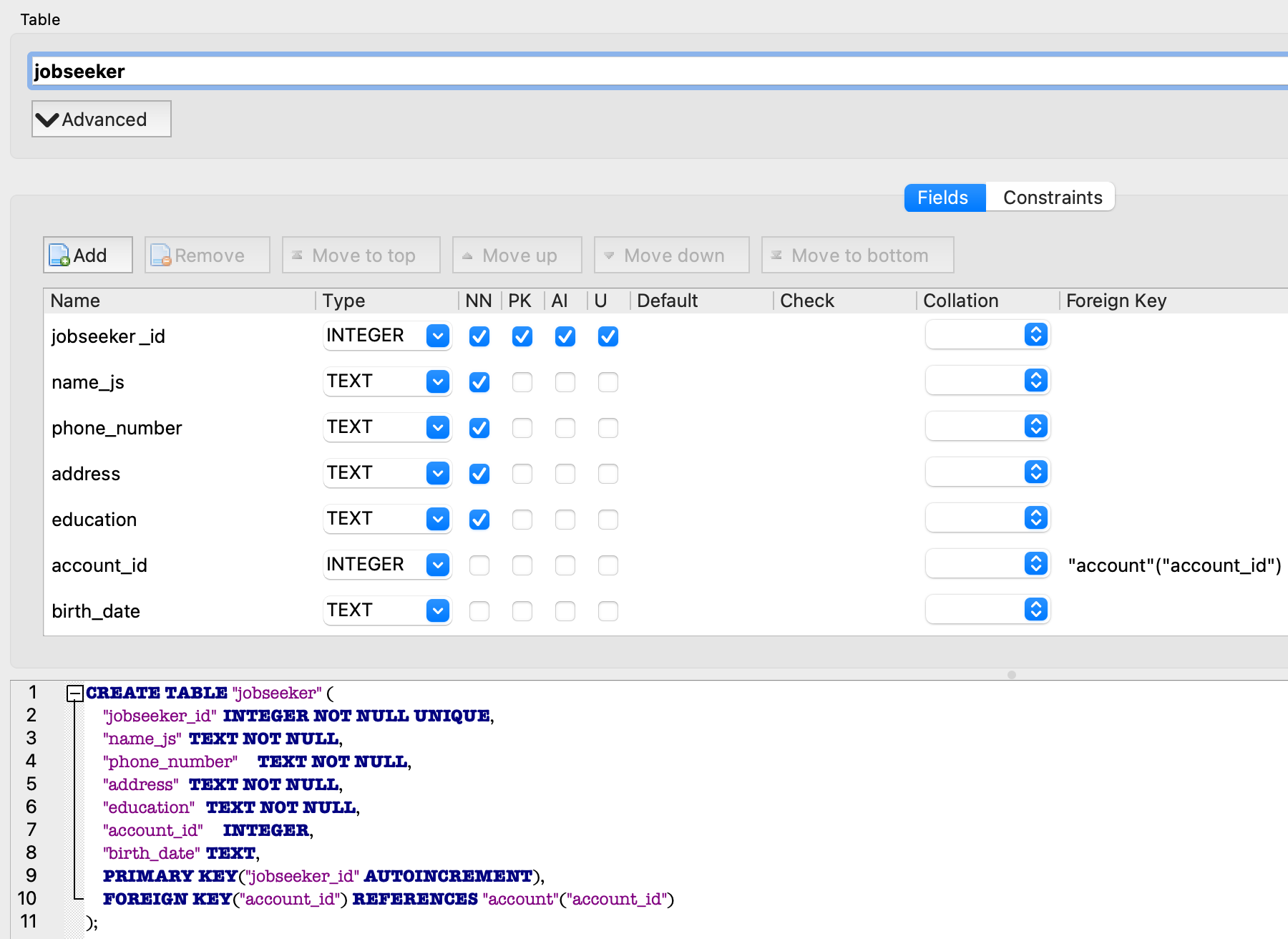


Tabel Interview

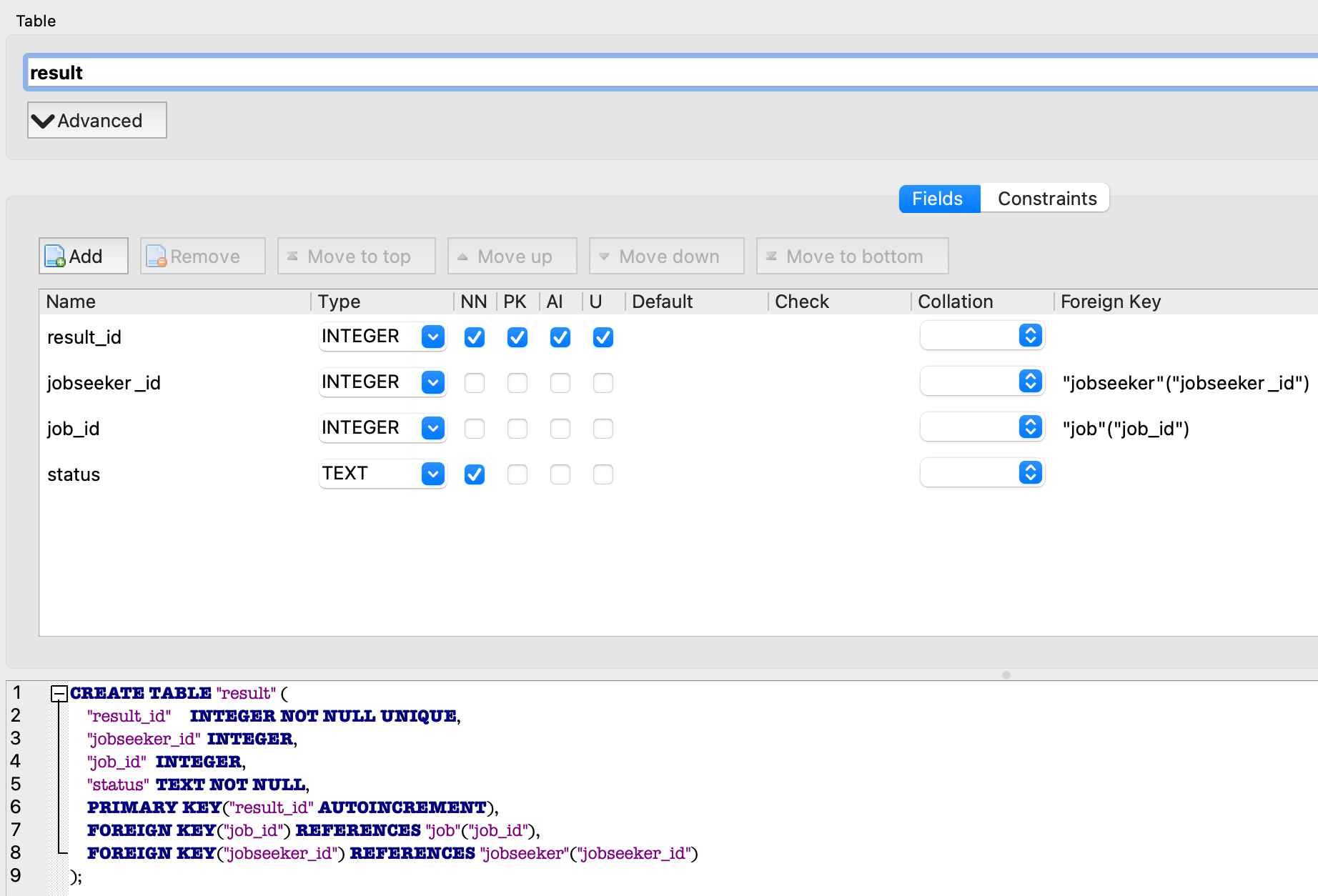


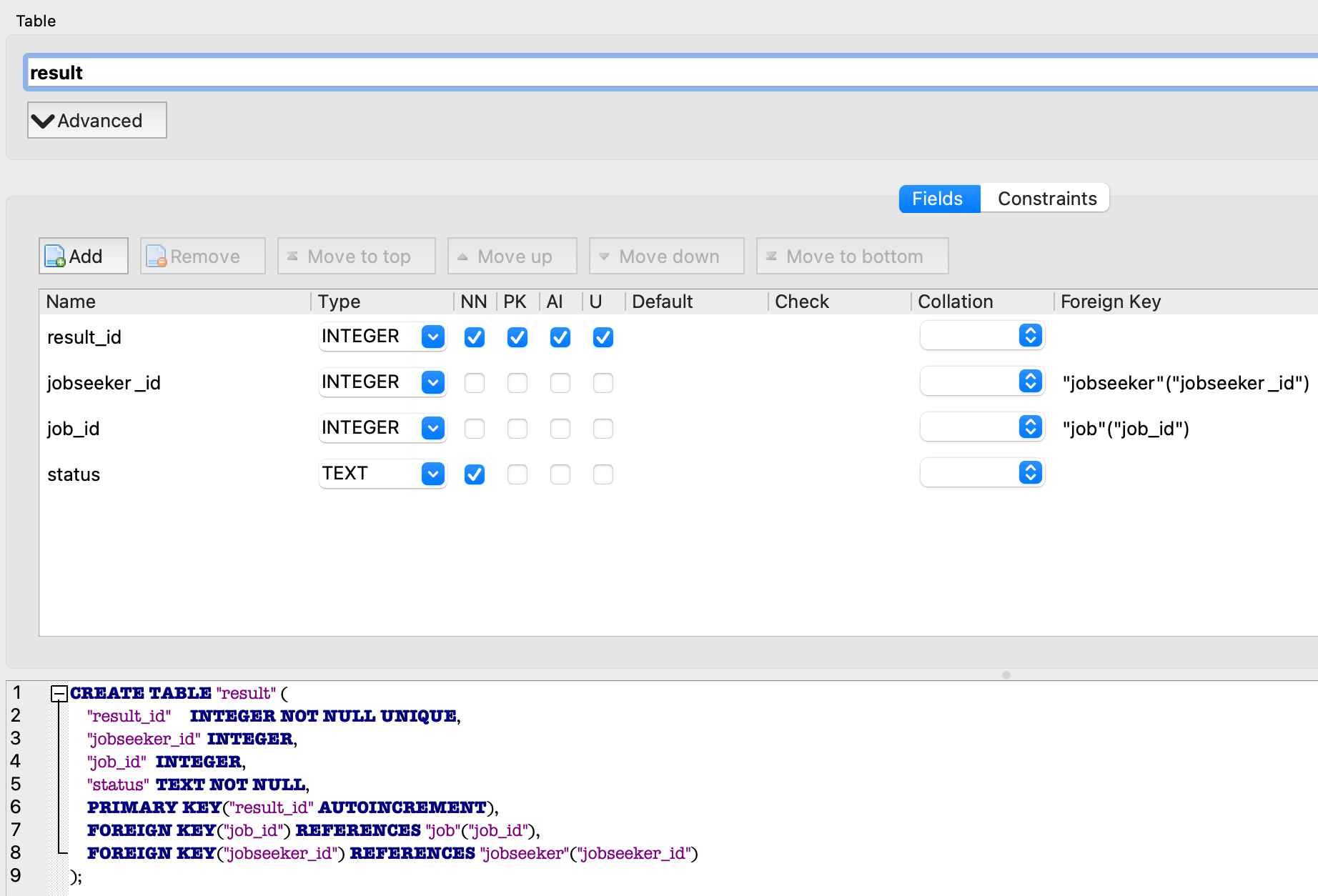


Tabel Jobseeker

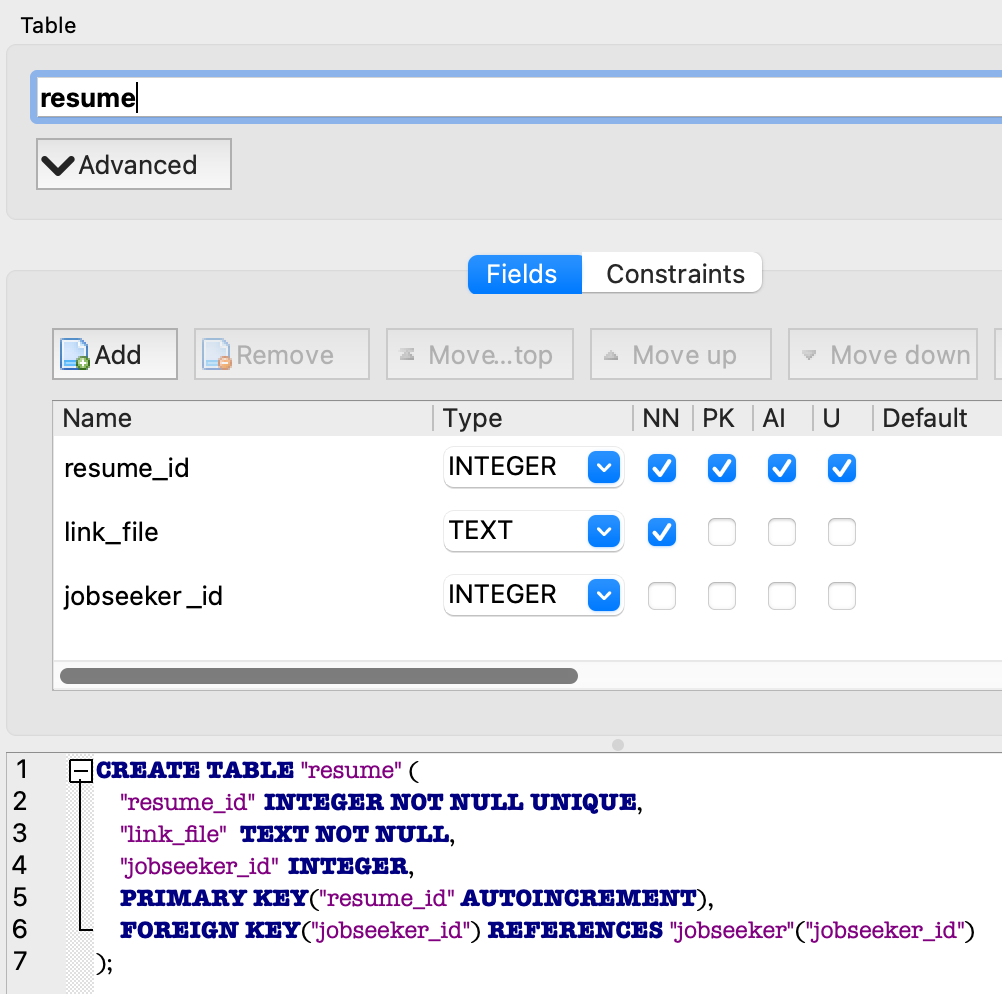


Tabel Result

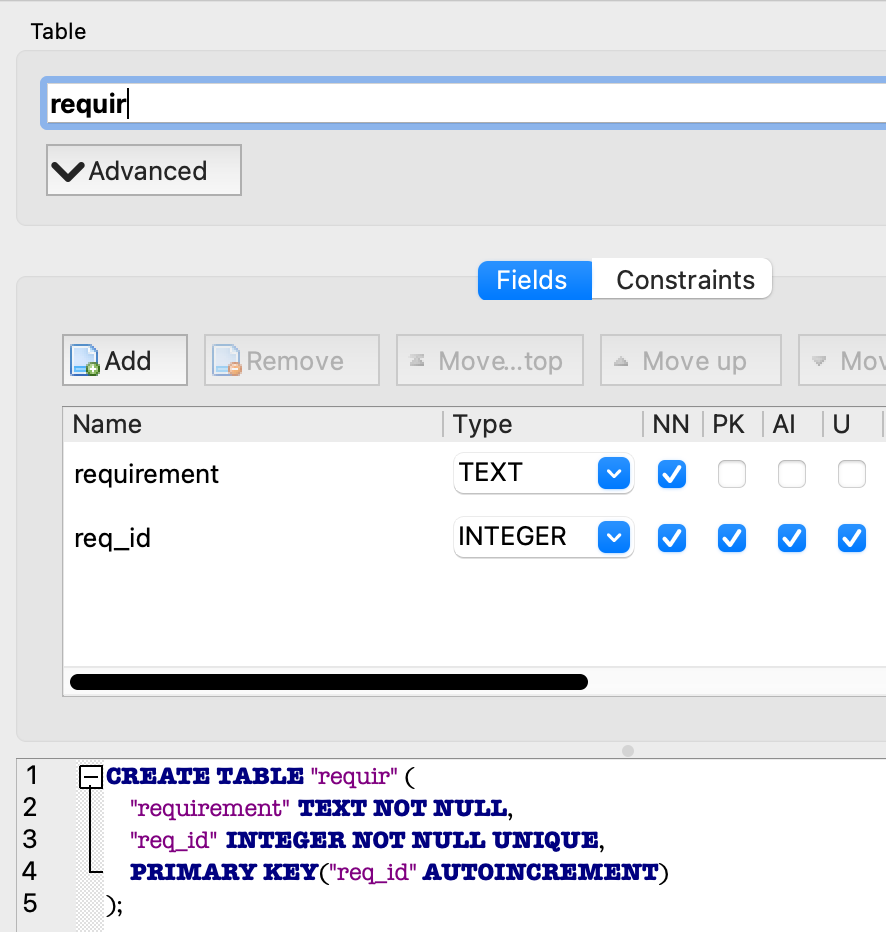




Tabel Resume



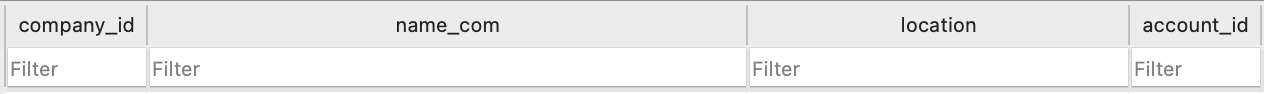
Tabel Requirement

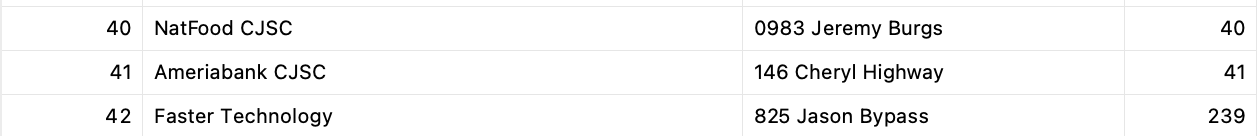


**INSERT INTO**

Memasukkan atau menyisipkan data ke dalam tabel





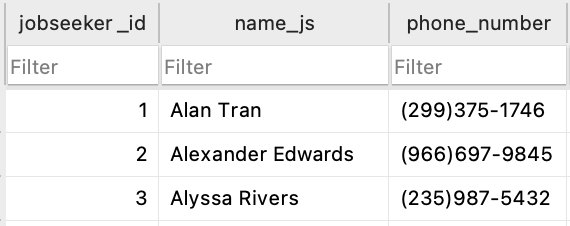


**UPDATE**

Memperbarui atau mengubah data



Sebelum

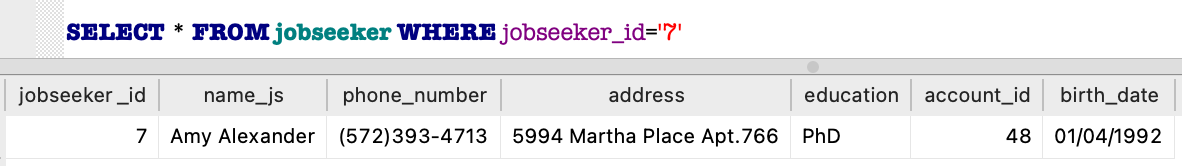


Sesudah



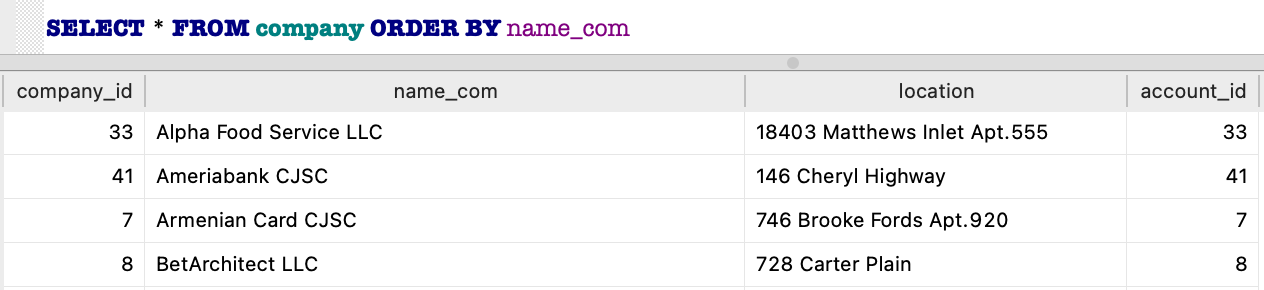
**SELECT WHERE**

Memilih data, hanya record (baris) yang memenuhi kriteria yang diinginkan



**SELECT ORDER BY**

Memilih data dan mengurutkan hasilnya berdasarkan kriteria yang diinginkan



**SELECT COUNT**

Menghitung record (baris) yang memenuhi kriteria yang diminta



**JOIN ON**

Menggabungkan baris pada beberapa tabel dalam database sesuai dengan kolom yang berkaitan antara tabel tersebut





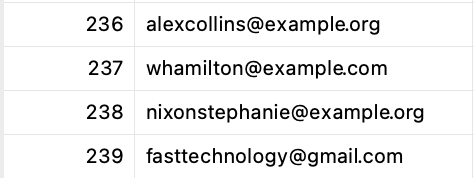
**DELETE**

Menghapus data pada tabel



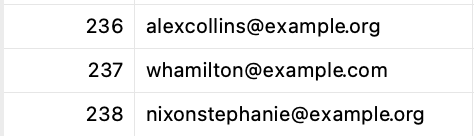
Sebelum





Sesudah





## 

## 3.2 Implementasi GUI

Graphical User Interface (GUI) adalah antarmuka program yang bertindak sebagai media komunikasi antara pengguna dan perangkat lunak (Mandel, 2002). GUI telah menjadi cara penting dan telah diterima dalam berinteraksi dengan perangkat lunak saat ini (Muhtadi, M. M., Friyadi, M. D., & Rahmani, A. (2019)). Graphical User Interface (GUI) adalah suatu antarmuka yang memungkinkan user berinteraksi melalui elemen atau yang biasa disebut *widgets* . Dengan adanya GUI, pengguna dapat mengakses fitur-fitur dengan lebih mudah. Melalui GUI yang dibuat, pengguna dapat melakukan query ke database dan mendapatkan respons secara *real-time* tanpa perlu menulis *query* SQL secara manual. Dengan adanya implementasi GUI, diharapkan database pencarian dan rekrutmen kerja dapat lebih mudah digunakan dan lebih efektif dalam membantu proses pencarian dan rekrutmen kerja.

### 3.2.1 *Package* yang Digunakan

Berikut adalah beberapa package yang digunakan dalam implementasi GUI:

1. **sqlite3**:

sqlite3 adalah modul Python yang menyediakan interface untuk berinteraksi dengan database SQLite. sqlite3 digunakan untuk membuat koneksi ke database SQLite, mengeksekusi query SQL, dan mengelola interaksi database.

1. **pandas.io.sql as pds**:

pandas.io.sql adalah submodul dari library pandas yang mendukung operasi SQL. pandas.io.sq digunakan untuk membaca, mengakses, dan menampilkan data antara Pandas DataFrame dan database SQL.

1. **pandas as pd**:

pandas adalah library *open-source* untuk analisis data dan manipulasi data dalam Python. pandas digunakan untuk berbagai operasi seperti membaca dan menulis data dalam berbagai format (CSV, Excel, SQL, dll.), mengelola dan menganalisis data, dan melakukan operasi data yang kompleks dengan DataFrame dan Series.

1. **Datetime**:

datetime adalah modul bawaan Python yang menyediakan fitur untuk memanipulasi tanggal dan waktu. datetime digunakan untuk membuat, memanipulasi, dan memformat objek tanggal dan waktu.

1. **ipywidgets**:

ipywidgets merupakan library dalam Python, yang memfasilitasi pembuat notebook Jupyter dan Google Colab dalam menciptakan GUI yang interaktif secara langsung di dalam notebook.

* Layout: Untuk menyediakan berbagai opsi tata letak untuk widget, seperti ukuran, margin, dan lainnya.
* widgets: Untuk membuat antarmuka pengguna interaktif, seperti tombol, kotak teks, dropdown, dll.
* Style: Untuk mengatur gaya visual dari widget.
* Button: Untuk membuat tombol interaktif.

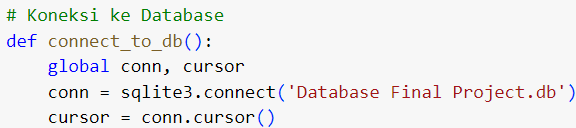
1. **IPython.display**:

* display: Untuk menampilkan objek.
* clear\_output: Untuk menghapus output cell notebook Jupyter.

### 3.2.2 Fitur *Login*

Sebelum *user* dapat mengakses fitur-fitur, *user* diharuskan untuk login terlebih dahulu. Berikut adalah beberapa *function* yang digunakan dalam implementasi GUI untuk fitur *login*

1. **connect\_to\_db()**: Fungsi ini digunakan untuk membuat koneksi ke *database*.



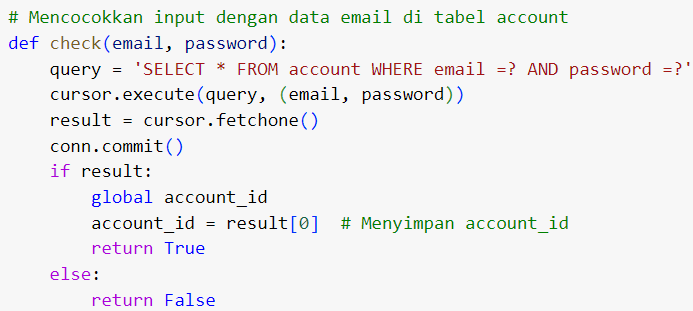
Cuplikan *codes* untuk koneksi ke *database*

1. **enter(email, password)**: Fungsi ini digunakan untuk memeriksa apakah email yang diinput sudah pernah digunakan sebelumnya. Jika email belum pernah digunakan, maka fungsi ini akan menambahkan data ke dalam tabel account.



Cuplikan *codes* untuk memeriksa email yang di-*input* *user*

1. **check(email, password)**: Fungsi ini digunakan untuk memeriksa apakah email dan password yang diinput sesuai dengan data yang ada di tabel account.



Cuplikan *codes* untuk mencocokkan *input* dengan data email di tabel account

1. **login(b)**: Fungsi ini digunakan untuk memeriksa apakah email dan password yang di-*input* sesuai dengan data yang ada di tabel “account”. Jika sesuai, maka fungsi ini akan menampilkan menu sesuai dengan jenis akun yang diinput (jobseeker atau company).

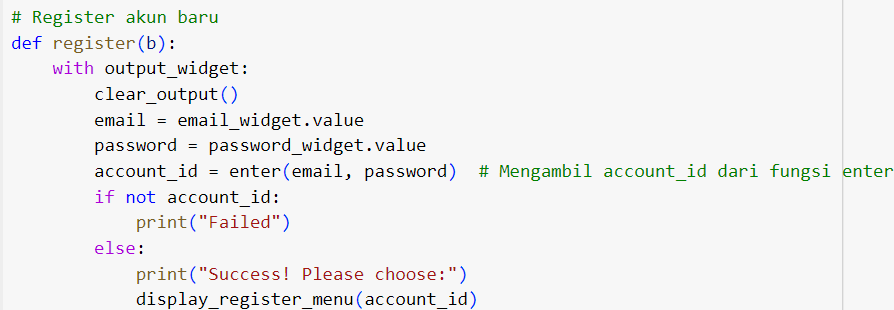


Cuplikan *codes* untuk *login*

### 3.2.3 Fitur *Register*

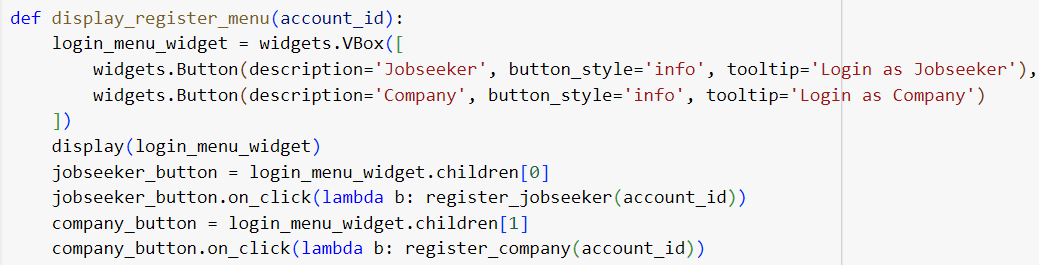
Jika belum mempunyai akun, *user* dapat melakukan *register* terlebih dahulu untuk membuat akun baru. Berikut adalah beberapa *function* yang digunakan dalam implementasi GUI untuk fitur *register*

1. **register(b)**: Fungsi ini digunakan untuk memeriksa apakah email yang di-*input* sudah pernah digunakan sebelumnya. Jika email belum pernah digunakan, maka fungsi ini akan menambahkan data ke dalam tabel account dan menampilkan menu sesuai dengan jenis akun yang diinput (jobseeker atau company).



Cuplikan *codes* untuk *register* akun baru

1. **display\_register\_menu(account\_id)**: Fungsi ini digunakan untuk menampilkan menu pendaftaran akun jobseeker atau company setelah mendaftar akun baru. User dapat memilih jenis akun (sebagai *jobseeker* atau *company*) ketika ingin mendaftar akun baru.



Cuplikan *codes* untuk menampilkan menu pendaftaran

1. **register\_company(account\_id)**: Fungsi ini digunakan untuk mendaftar akun sebagai *company*. Jika mendaftar berhasil, maka akan menampilkan menu pendaftaran akun *company*.
2. **complete\_register\_company(company\_name, location, account\_id)**: Fungsi ini digunakan untuk menyelesaikan pendaftaran akun company. Jika pendaftaran berhasil, maka akan menampilkan konfirmasi pendaftaran dan login lagi.



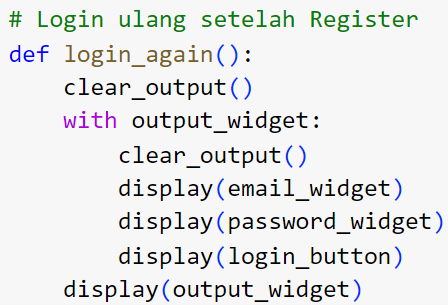
Cuplikan *codes function* jika register sebagai *company,* yakni *function* register\_company(account\_id) dan complete\_register\_company(company\_name, location, account\_id)

1. **register\_jobseeker(account\_id)**: Fungsi ini digunakan untuk mendaftar akun sebagai *jobseeker*. Jika mendaftar berhasil, maka akan menampilkan menu pendaftaran akun *jobseeker*.
2. **complete\_register\_jobseeker(name, address, phone, education, birth\_date, account\_id):** Fungsi ini digunakan untuk menyelesaikan pendaftaran akun *jobseeker*. Jika pendaftaran berhasil, maka akan menampilkan konfirmasi pendaftaran dan login lagi.



Cuplikan *codes function* jika register sebagai *jobseeker,* yakni *function* register\_jobseeker(account\_id) dan complete\_register\_jobseeker(name, address, phone, education, birth\_date, account\_id)

1. **login\_again():** Fungsi ini digunakan untuk *login* ulang setelah mendaftar akun baru.



Cuplikan *codes* untuk *login* kembali setelah berhasil mendaftar akun baru

### 3.2.3 Menu Utama

Terdapat menu utama yang berbeda untuk *jobseeker* dan *company.*

1. **display\_menu\_jobseeker():** Fungsi ini digunakan untuk menampilkan menu utama bagi jobseeker.



Cuplikan *codes* untuk menampilkan menu bagi *jobseeker*

1. **display\_menu\_company():** Fungsi ini digunakan untuk menampilkan menu utama untuk company.

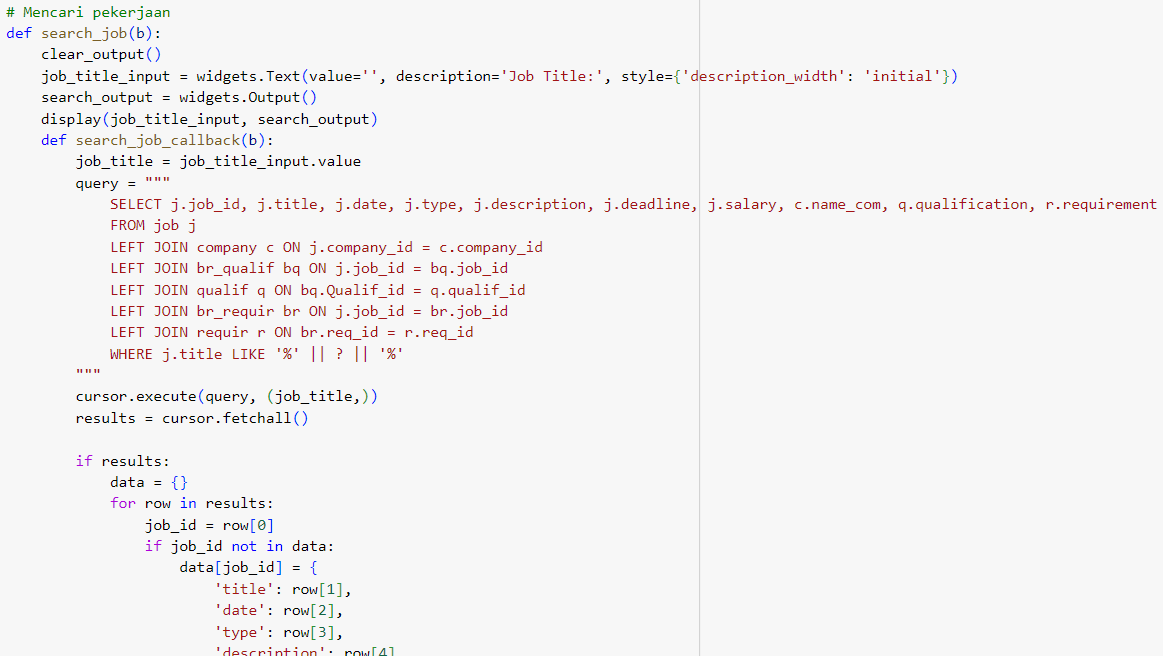


Cuplikan *codes* untuk menampilkan menu *company*

### 3.2.3 Fitur *Jobseeker*

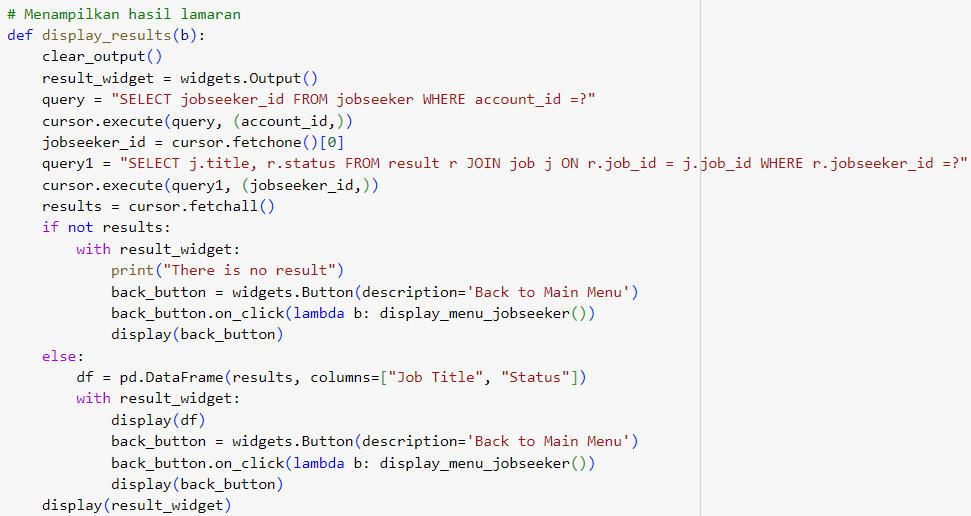
Jika *user* berhasil login sebagai *jobseeker*, akan ditampilkan menu *jobseeker* di mana *user* dapat memilih fitur yang ingin digunakan sebagai *jobseeker*.

1. **search\_job(b):** Fungsi ini digunakan untuk mencari pekerjaan yang sesuai dengan kriteria yang dimasukkan. Jika pencarian berhasil, maka akan menampilkan data pekerjaan yang sesuai.



Cuplikan *codes* untuk fitur mencari pekerjaan bagi *jobseeker*

1. **display\_results(b):** Fungsi ini digunakan untuk menampilkan hasil lamaran pekerjaan yang sudah dilakukan. Jika ada hasil lamaran, maka akan menampilkan data pekerjaan yang dilamar.



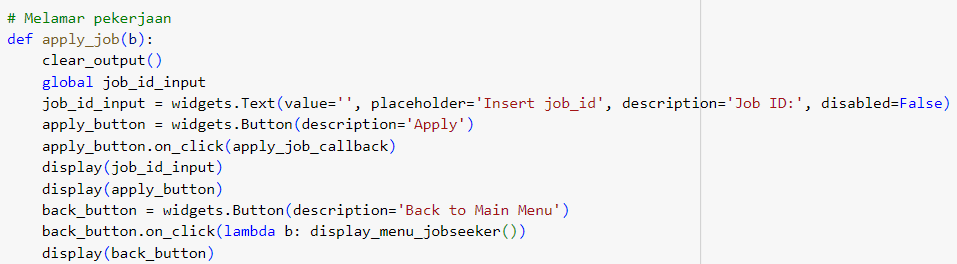
Cuplikan *codes* untuk fitur menampilkan status hasil lamaran pekerjaan

1. **display\_interview(b):** Fungsi ini digunakan untuk menampilkan jadwal *interview* yang sudah dijadwalkan. Jika ada jadwal *interview*, maka akan menampilkan data *interview* yang sudah dijadwalkan.



Cuplikan *codes* untuk fitur menampilkan jadwal *interview* yang sudah dijadwalkan

1. **apply\_job(b)**: Fungsi ini digunakan untuk melamar pekerjaan. Jika melamar berhasil, maka akan menampilkan konfirmasi lamaran pekerjaan.



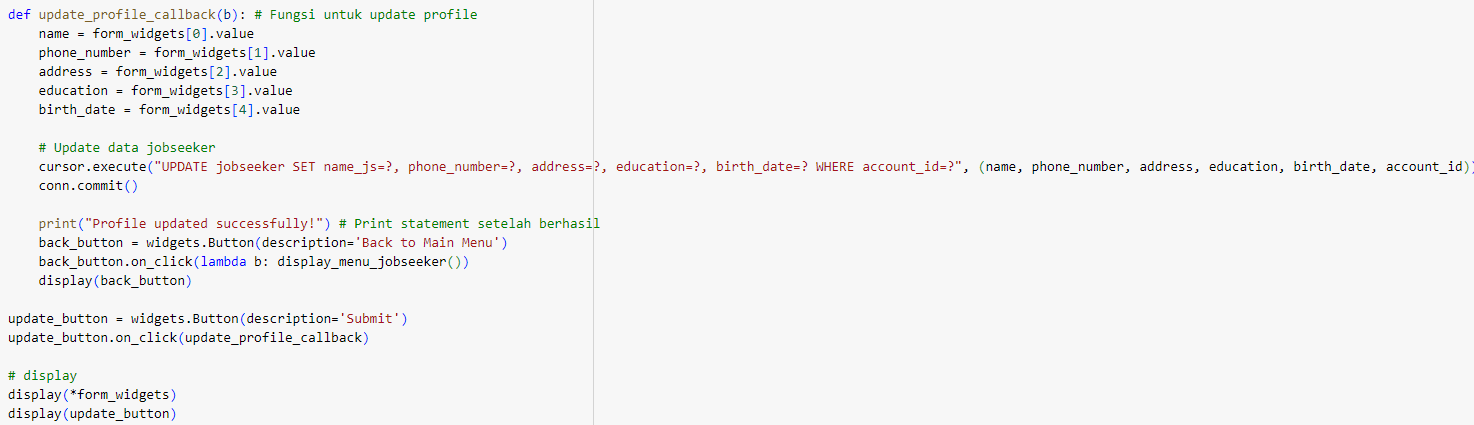
Cuplikan *codes* untuk fitur melamar pekerjaan

1. **display\_jobseeker\_info(account\_id)**: Fungsi ini digunakan untuk menampilkan informasi akun jobseeker. Jika informasi akun jobseeker sudah ada, maka akan menampilkan data akun jobseeker.



Cuplikan *codes* untuk fitur menampilkan informasi akun *jobseeker*

1. **update\_profile(account\_id)**: Fungsi ini digunakan untuk mengupdate informasi akun jobseeker. Jika update berhasil, maka akan menampilkan konfirmasi update.



Cuplikan *codes* untuk fitur mengupdate informasi akun *jobseeker*

1. **update\_resume(b):** Fungsi ini digunakan untuk mengupdate resume akun *jobseeker*. Jika update berhasil, maka akan menampilkan konfirmasi *update*.

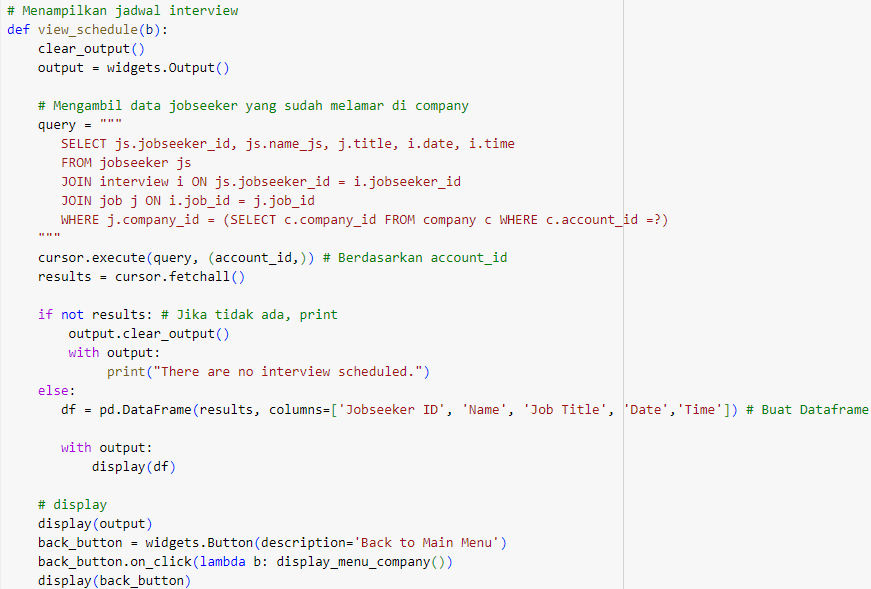


Cuplikan *codes* untuk fitur mengupdate *link resume jobseeker.*

### 3.2.3 Fitur Company

Jika *user* berhasil login sebagai *company*, akan ditampilkan menu *company* di mana *user* dapat memilih fitur yang ingin digunakan sebagai *company*.

1. **view\_schedule(b):** Fungsi ini digunakan untuk melihat jadwal *interview* dari *jobseeker* yang melamar pada *company* tersebut



Cuplikan *codes* untuk fitur melihat jadwal *interview jobseeker* yang melamar

1. **create\_interview\_schedule(account\_id):** Fungsi ini digunakan untuk membuat jadwal interview baru. Jika jadwal interview berhasil akan diberikan pesan konfirmasi.

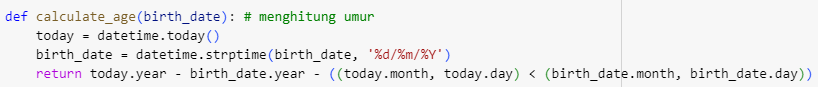


Cuplikan *codes* untuk fitur membuat jadwal interview baru

1. **view\_applicants(b):** Fungsi ini digunakan untuk menampilkan daftar pelamar pekerjaan. Jika ada pelamar, maka akan menampilkan data pelamar pekerjaan.



Cuplikan *codes* untuk fitur menampilkan informasi *jobseeker* yang melamar kerja pada suatu *company*



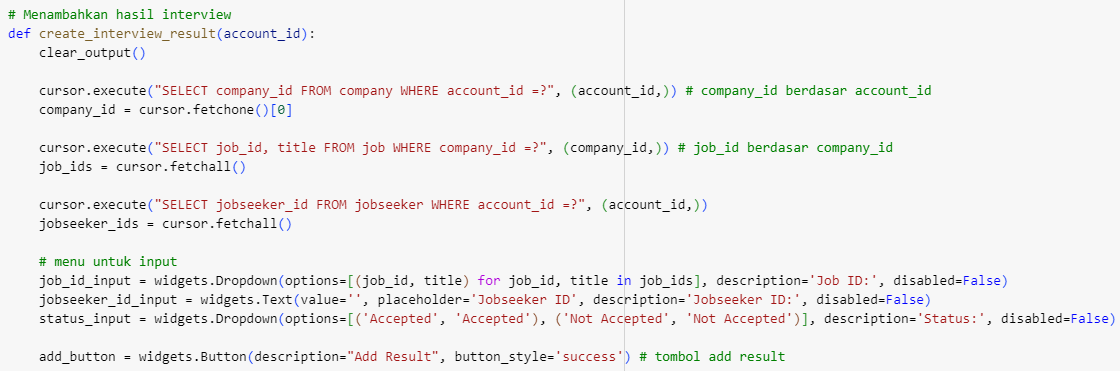
Cuplikan *codes* untuk menghitung umur *jobseeker* berdasarkan tanggal lahir dari *database.*

1. **post\_job(b):** Fungsi ini digunakan untuk mem-*posting* pekerjaan baru. Jika *posting* berhasil, maka akan menampilkan konfirmasi *posting* pekerjaan.



Cuplikan *codes* untuk fitur menambahkan penawaran kerja bagi suatu *company*

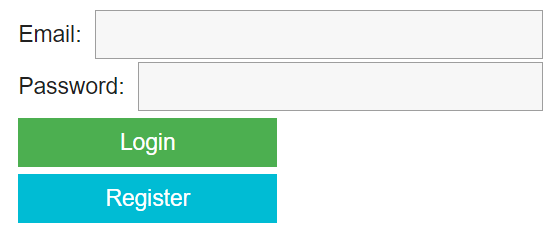
1. **create\_interview\_result(account\_id):** Fungsi ini digunakan untuk menambahkan hasil *interview* ke dalam tabel result.



Cuplikan *codes* untuk fitur menambahkan status diterima atau tidaknya pelamaran kerja

## 3.3 Testing

Ketika pertama kali mengakses GUI, pengguna akan melihat tampilan awal seperti pada gambar dibawah.



Gambar 3.1

### 3.3.1 Fitur Login

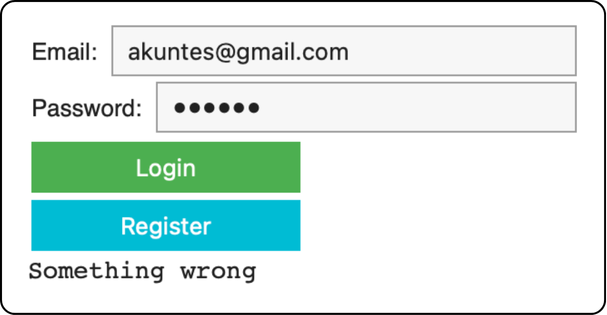
Jika pengguna memilih "Login", pengguna akan diminta memasukkan email dan kata sandinya. Sistem kemudian memeriksa apakah email dan kata sandi cocok dengan akun yang ada pada database.

* Jika login berhasil, pengguna akan diarahkan ke menu utama baik sebagai *jobseeker* (Gambar 3.2) maupun *company* (Gambar 3.3), tergantung jenis akunnya.

## 

Gambar 3.2 Gambar 3.3

* Jika login gagal, pengguna diberitahu dan diberikan pilihan untuk mencoba lagi.

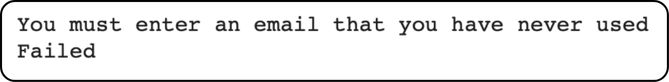


Gambar 3.4

### 3.3.2 Fitur Registrasi

Jika pengguna belum mempunyai akun pada database, maka pengguna harus registrasi terlebih dahulu. Langkah-langkahnya adalah sebagai berikut:

1. Masukkan email dan password pada tampilan awal.
2. Tekan tombol registrasi. Gunakanlah email yang belum pernah digunakan. Jika email yang dimasukkan sudah ada dalam database, maka akan ditampilkan kalimat pada Gambar 3.5.



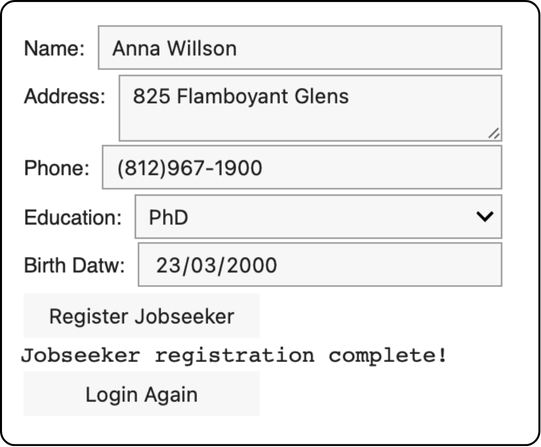
Gambar 3.5

1. Setelah menekan tombol registrasi, pengguna diarahkan untuk memilih mendaftarkan akun sebagai *jobseeker* atau *company*.

## 

Gambar 3.6

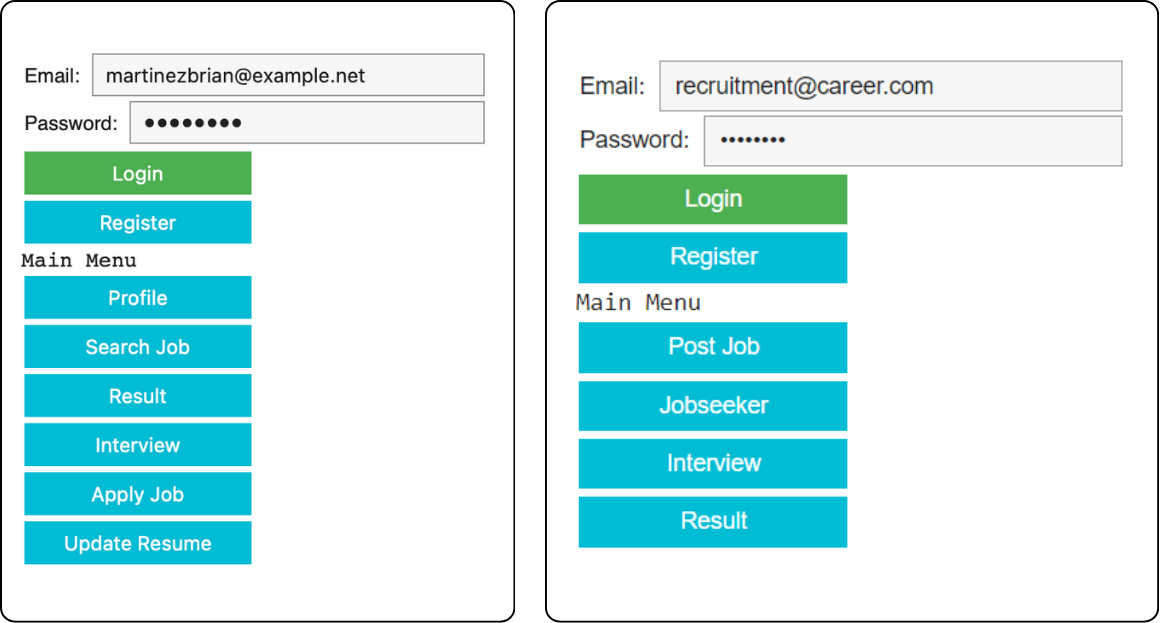
1. Setelah memilih salah satu, pengguna akan diarahkan untuk mengisi data. Gambar 3.7 sebagai *jobseeker* dan Gambar 3.8 sebagai *company*.

 Gambar 3.8

Gambar 3.7

1. Setelah berhasil, klik tombol *login again* dan pengguna akan diarahkan untuk login kembali agar dapat menggunakan fitur fitur selanjutnya.

### 3.3.3 Menu Utama



Gambar 3.9 Gambar 3.10

1. Main Menu Jobseeker (Gambar 3.9)

Menu utama sebagai jobseeker mencakup opsi untuk :

* melihat profil,
* mencari pekerjaan,
* melihat hasil,
* melihat jadwal wawancara,
* melamar pekerjaan, dan
* memperbarui resume.

1. Main Menu Company (Gambar 3.10)

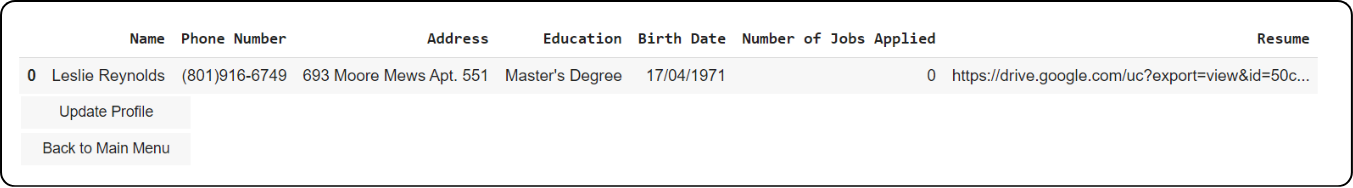
Menu utama sebagai *company* mencakup opsi untuk

* memposting pekerjaan,
* melihat pelamar,
* membuat jadwal wawancara, dan
* menambahkan hasil untuk *jobseeker.*

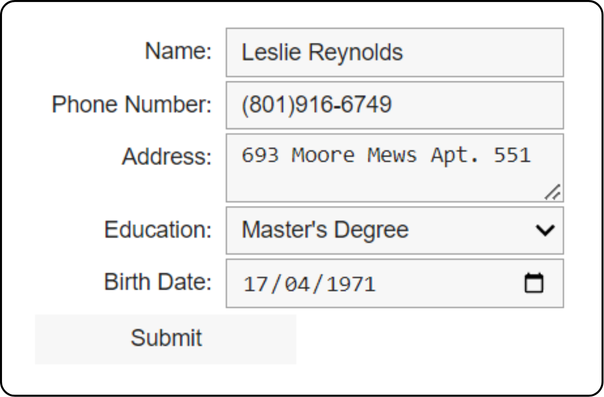
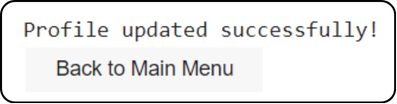
### 3.3.4 Fitur Jobseeker

1. **Profile**

Fitur "Profile" memungkinkan pencari kerja untuk melihat informasi pribadi, seperti nama, nomor telepon, alamat, pendidikan, dan tanggal lahir (Gambar 3.11). Mereka juga dapat melihat jumlah pekerjaan yang telah mereka lamar dan link resume mereka. Untuk memperbarui profil, user dapat mengklik tombol "Update Profile" (Gambar 3.12) dan mengklik tombol “Submit” setelah selesai. Pengguna akan diarahkan mengklik tombol “Back to Main Menu” setelahnya (Gambar 3.13).



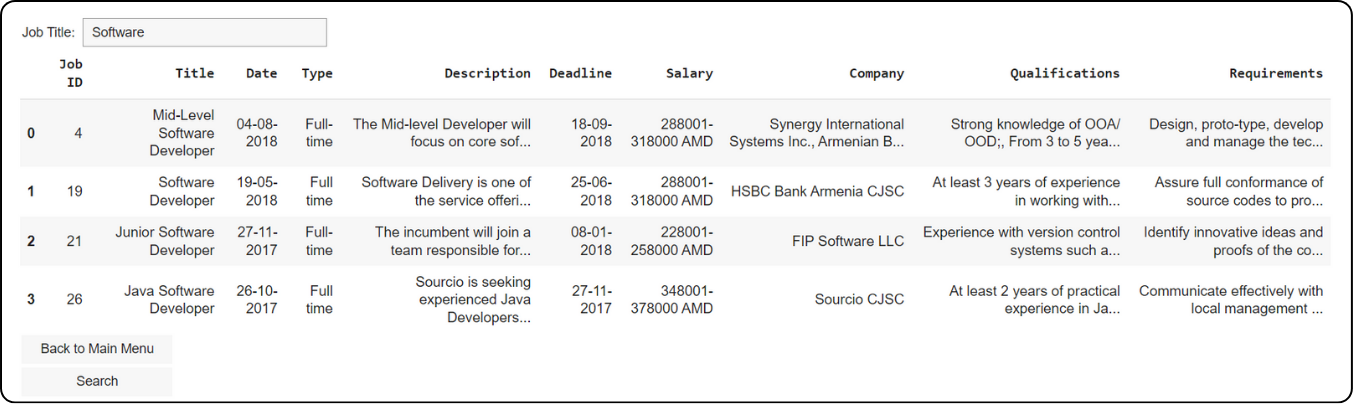
Gambar 3.11

 Gambar 3.13

Gambar 3.12

1. **Search Job**

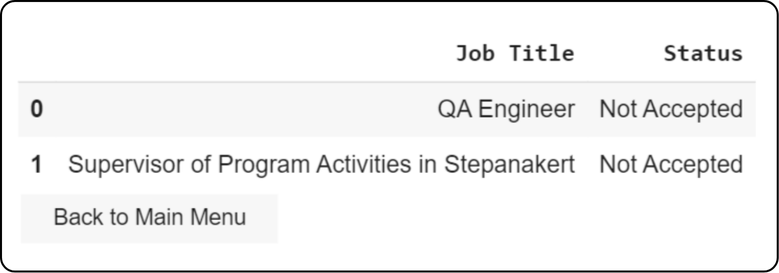
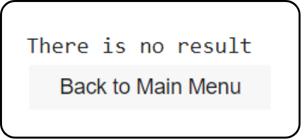
Fitur "Search Job" memungkinkan pencari kerja untuk mencari pekerjaan yang sesuai dengan *Job Title.* Gambar 3.14 menunjukkan contoh hasil pencarian pekerjaan dengan *keyword* software.

**

Gambar 3.14

1. **Result**

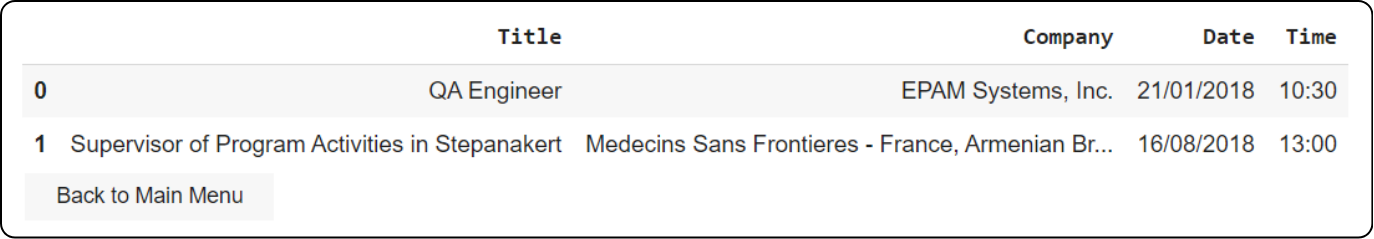
Fitur "Result" memungkinkan jobseeker untuk melihat hasil dari pelamaran kerja (Gambar 3.15). Jika *jobseeker* belum pernah melamar kerja, akan ditampilkan seperti Gambar 3.16.

****

Gambar 3.15 Gambar 3.16

1. **Interview**

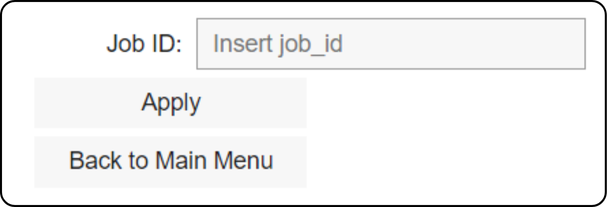
Fitur "Interview" akan menampilkan jadwal interview yang telah dijadwalkan untuk *jobseeker*.



Gambar 3.17

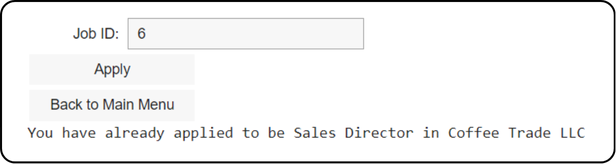
1. **Apply Job**

Fitur "Apply Job" memungkinkan jobseeker untuk melamar kerja dengan meng-*input* Job ID seperti pada Gambar 3.18.

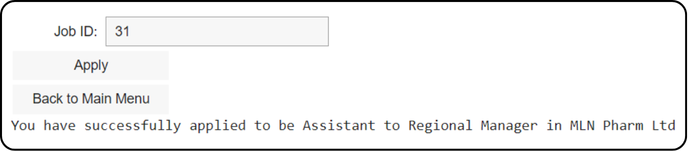


Gambar 3.18

Jika *jobseeker* melamar pada *job* yang sudah pernah dilamar, akan ditampilkan Gambar 3.19. Jika *jobseeker* berhasilmelamar kerja, akan ditampilkan Gambar 3.20.

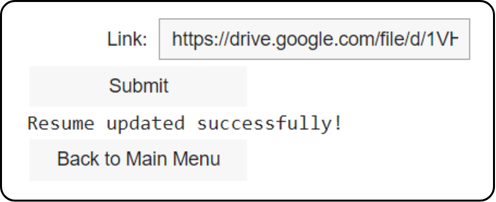


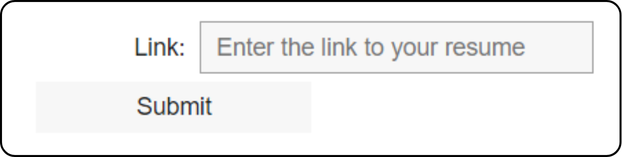
Gambar 3.19



Gambar 3.20

1. **Update Resume**

Fitur "Update Resume" memungkinkan *jobseeker* untuk mengupdate *link* resume (Gambar 3.21). Setelah memasukkan link dan mengklik tombol “Submit”, akan ditampilkan Gambar 3.22.

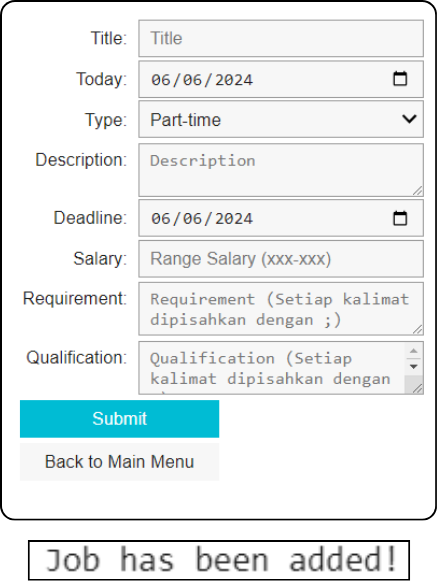


Gambar 3.21 Gambar 3.22

### 3.3.5 Fitur Company

1. **Post Job**

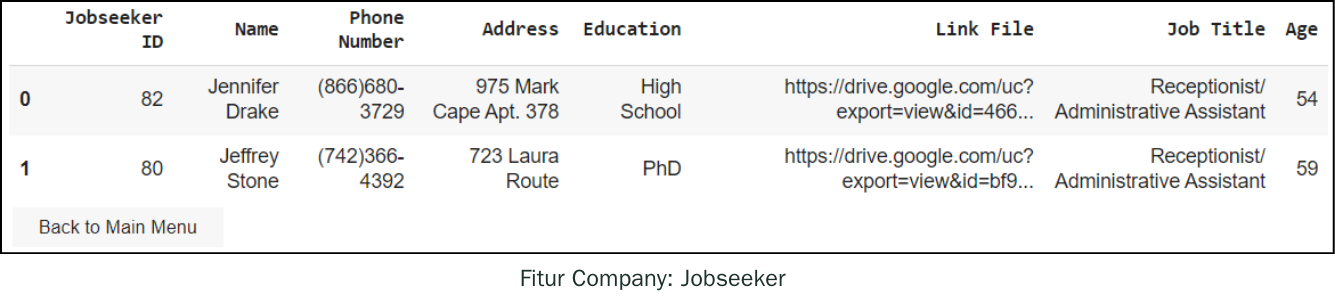
Fitur "Post Job" memungkinkan company untuk menambahkan tawaran pekerjaan baru. Setelah berhasil submit job baru, akan ditampilkan Gambar 3.23.



Gambar 3.23

1. **Jobseeker**

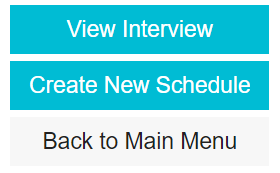
Fitur "Jobseeker" memungkinkan company untuk melihat jobseeker yang sudah melamar ke company mereka dan pekerjaan apa yang mereka lamar (Gambar 3.24).



Gambar 3.24

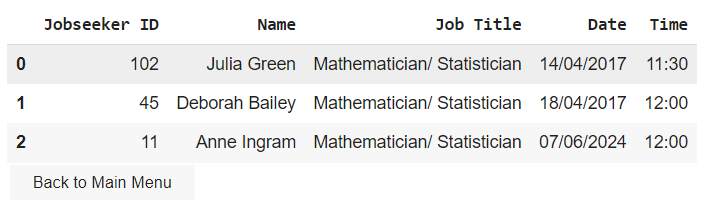
1. **Interview**

Ketika pengguna mengklik tombol “Interview”, akan ditampilkan Gambar 3.25 yang memungkinkan pengguna untuk memilih antara “View Interview” atau “Create New Schedule”.



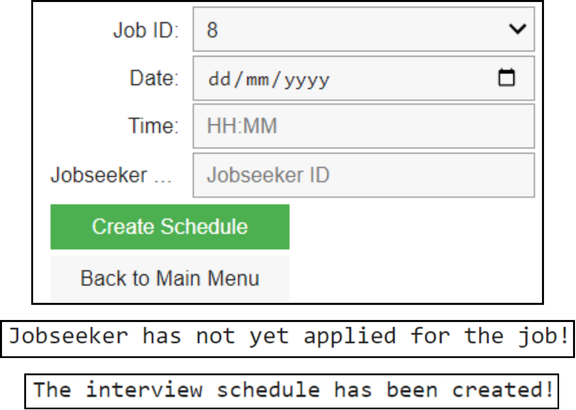
Gambar 3.25

Fitur "View Interview" memungkinkan company untuk melihat interview yang terjadwal (Gambar 3.26).



Gambar 3.26

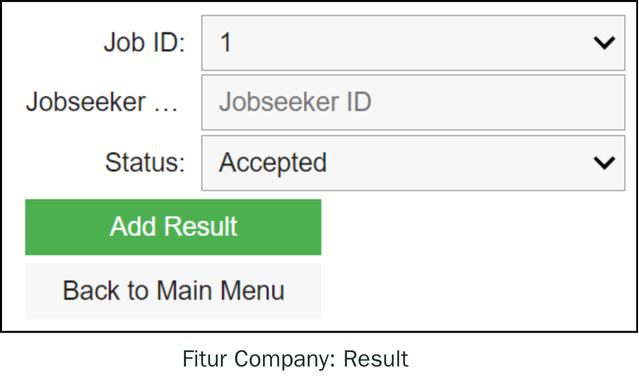
Fitur "Create New Schedule" memungkinkan company untuk menjadwalkan interview dengan jobseeker. Apabila jobseeker ID yang dimasukkan bukanlah pelamar pada company, akan ditampilkan “Jobseeker has not yet applied for the job”. Jika sudah, akan ditampilkan “The interview schedule has been created” (Gambar 3.27).



Gambar 3.27

1. **Result**

Fitur “Result” memungkinkan company untuk menambahkan hasil lamaran kerja dengan pilihan Accepter/Not Accepted (Gambar 3.28).



Gambar 3.28

# 

# **BAB 4 PENUTUP**

Berdasarkan hasil simulasi yang telah dilakukan, dapat disimpulkan bahwa implementasi SQL dan GUI dalam Mini Project Database untuk Sains Data telah berhasil. Database yang digunakan adalah "Database Final Project.db" yang terdiri dari beberapa tabel relasional, yaitu account, sqlite\_sequence, qualif, requir, br\_apply, br\_qualif, br\_requir, interview, jobseeker, result, resume, dan job. Setiap tabel memiliki kolom dan kunci primer yang sesuai dengan kebutuhan data.

Implementasi SQL dalam Mini Project ini meliputi query untuk memasukkan, mengubah, menghapus, dan mengambil data dari database. Query-query ini digunakan untuk berinteraksi dengan database melalui GUI yang telah dibuat. GUI ini terdiri dari beberapa menu, yaitu login, register, menu utama jobseeker, dan menu utama company. Setiap menu memiliki fungsi yang berbeda-beda, seperti mencari pekerjaan, melamar pekerjaan, melihat hasil lamaran, dan lain-lain.

Dari kesimpulan ini, dapat disimpulkan bahwa Mini Project Database untuk Sains Data telah berhasil diimplementasikan menggunakan SQL dan GUI. Database yang digunakan telah terstruktur dengan baik serta dapat digunakan untuk menyimpan dan mengelola data yang diperlukan dalam Mini Project ini.

# **DAFTAR PUSTAKA**

Putra, D. W. T., & Putra, J. J. (2018). Perancangan Sistem Informasi Pencarian Lowongan Pekerjaan. Jurnal Teknoif Teknik Informatika Institut Teknologi Padang, 6(1), 48-54.

Mandel, T. (2002). User/System Interface Design. Encyclopedia of Information Systems, 1, 1-4.

Mathisugan, R. (2021). *Online job portal* (Doctoral dissertation).

Muhtadi, M. M., Friyadi, M. D., & Rahmani, A. (2019, August). Analisis GUI testing pada aplikasi e-commerce menggunakan katalon. In Prosiding Industrial Research Workshop and National Seminar (Vol. 10, No. 1, pp. 1387-1393).

Setiawati, P. (2018). Analisa Dan Perancangan Sistem Informasi Penyedia Lowongan Pekerjaan Yang Direkomendasi Berdasarkan Standar Kompetensi Kerja Nasional Indonesia (Skkni). Ilmu Komput, 3(2), 136-147.

Suwarno, S., Widada, B., & Siswanti, S. (2015). Sistem Informasi Lowongan Pekerjaan Berbasis Web pada Balai Latihan Kerja Boyolali. Jurnal Teknologi Informasi dan Komunikasi (TIKomSiN), 3(1).

# **LAMPIRAN**

**Lampiran 1: *Database* yang digunakan**

[Project Database B Kelompok 4](https://drive.google.com/drive/folders/1i12h6cvZFtUoRDVFPVMhOdECy3D6ebaf?usp=drive_link)

**Lampiran 2: *Codes* untuk implementasi GUI**

[Final Project Kelompok 4](https://colab.research.google.com/drive/1uTkE74dYXu0-uQQXy12AcsP5iGOAkbVq?usp=sharing)