Stefan Daschek / @noniq

Cologne Phonetics

Phonetic Search for German

Background

WIENER STAATSOPER

SPIELPLAN & TICKETS LIVE ENSEMBLE & GÄSTE IHR BESUCH DIE STAATSOPER OPERNBALL

LISTE KALENDER WERKE ARCHIV

Spielplanarchiv

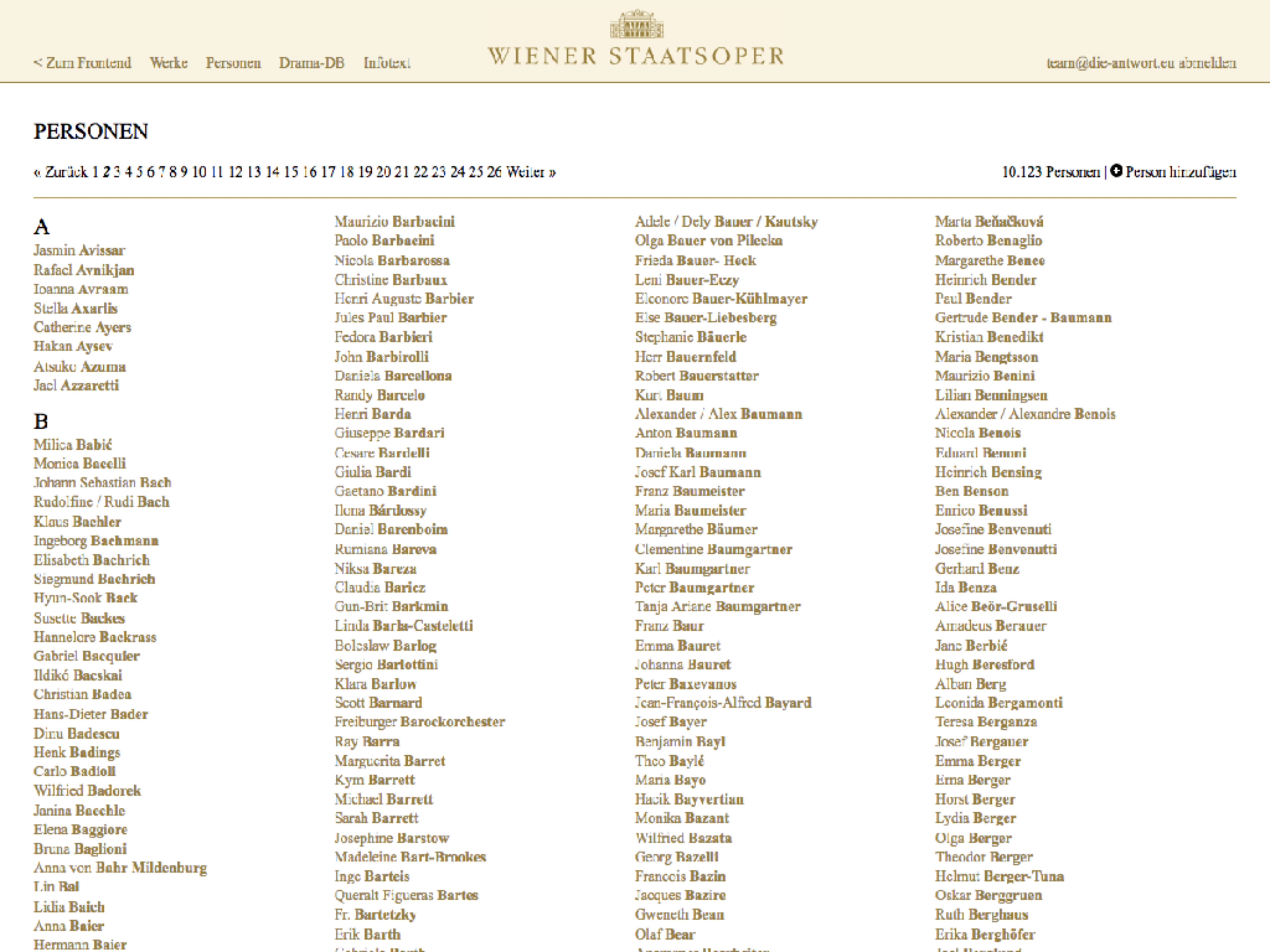
Personenverzeichnis | Werkverzeichnis | Rollenverzeichnis | Vorstellungsverzeichnis

SUCHE

Sie können mehrere Suchkriterien gleichzeitig angeben. Angezeigt werden dann jene Ergebnisse, bei denen alle angegebenen Kriterien gleichzeitig zutreffen.

Personen

Werk



PERSONEN

« Zurück 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 Weiter »

10.123 Personen | ● Person hinzufügen

A

Jasmin **Avissar**
Rafael **Avnikjan**
Ioanna **Avraam**
Stella **Axarlis**
Catherine **Ayers**
Hakan **Aysev**
Atsuko **Azuma**
Jael **Azzaretti**

B

Milica **Babić**
Monica **Bacelli**
Johann Sebastian **Bach**
Rudolfine / Rudi **Bach**
Klaus **Bachler**
Ingeborg **Bachmann**
Elisabeth **Bachrich**
Siegmund **Bachrich**
Hyun-Sook **Back**
Susette **Backes**
Hannelore **Backrass**
Gabriel **Bacquier**
Ildikó **Bacskai**
Christian **Badea**
Hans-Dieter **Bader**
Dinu **Badescu**
Henk **Badings**
Carlo **Badioli**
Wilfried **Badorek**
Janina **Baechle**
Elena **Baggiore**
Bruna **Baglioni**
Anna von **Bahr Mildenburg**
Lin **Bai**
Lidia **Baich**
Anna **Baier**
Hermann **Baier**
Maurizio **Barbacini**
Paolo **Barbacini**
Nicola **Barbarossa**
Christine **Barbaux**
Henri Auguste **Barbier**
Jules Paul **Barbier**
Fedora **Barbieri**
John **Barbirolli**
Daniela **Barcellona**
Randy **Barcelo**
Henri **Barda**
Giuseppe **Bardari**
Cesare **Bardelli**
Giulia **Bardi**
Gaetano **Bardini**
Ilona **Bárdossy**
Daniel **Barenboim**
Rumiana **Bareva**
Niksa **Bareza**
Claudia **Baricz**
Gun-Brit **Barkmin**
Linda **Barla-Casteletti**
Boleslaw **Barlog**
Sergio **Barlottini**
Klara **Barlow**
Scott **Barnard**
Freiburger **Barockorchester**
Ray **Barra**
Marguerita **Barret**
Kym **Barrett**
Michael **Barrett**
Sarah **Barrett**
Josephine **Barstow**
Madeleine **Bart-Brookes**
Inge **Barteis**
Queralt Figueras **Bartes**
Fr. **Bartetzky**
Erik **Barth**
Adele / Dely **Bauer / Kautsky**
Olga **Bauer von Pilecka**
Frieda **Bauer- Heck**
Leni **Bauer-Eczy**
Eleonore **Bauer-Kühlmayer**
Else **Bauer-Liebesberg**
Stephanie **Bäuerle**
Herr **Bauernfeld**
Robert **Bauerstatter**
Kurt **Baum**
Alexander / Alex **Baumann**
Anton **Baumann**
Daniela **Baumann**
Josef Karl **Baumann**
Franz **Baumeister**
Maria **Baumeister**
Margarethe **Bäumer**
Clementine **Baumgartner**
Karl **Baumgartner**
Peter **Baumgartner**
Tanja Ariane **Baumgartner**
Franz **Baur**
Emma **Bauret**
Johanna **Bauret**
Peter **Baxevanos**
Jean-François-Alfred **Bayard**
Josef **Bayer**
Benjamin **Bayl**
Theo **Baylé**
Maria **Bayo**
Hacik **Bayvertian**
Monika **Bazant**
Wilfried **Bazata**
Georg **Bazelli**
Francois **Bazin**
Jacques **Bazire**
Gweneth **Bean**
Olaf **Bear**
Marta **Beňačková**
Roberto **Benaglio**
Margarethe **Bence**
Heinrich **Bender**
Paul **Bender**
Gertrude **Bender - Baumann**
Kristian **Benedikt**
Maria **Bengtsson**
Maurizio **Benini**
Lilian **Benningsen**
Alexander / Alexandre **Benois**
Nicola **Benois**
Eduard **Benoni**
Heinrich **Bensing**
Ben **Benson**
Enrico **Benussi**
Josefine **Benvenuti**
Josefine **Benvenutti**
Gerhard **Benz**
Ida **Benza**
Alice **Beör-Gruselli**
Amadeus **Berauer**
Jane **Berbié**
Hugh **Beresford**
Alban **Berg**
Leonida **Bergamonti**
Teresa **Berganza**
Josef **Bergauer**
Emma **Berger**
Erna **Berger**
Horst **Berger**
Lydia **Berger**
Olga **Berger**
Theodor **Berger**
Helmut **Berger-Tuna**
Oskar **Berggruen**
Ruth **Berghaus**
Erika **Berghöfer**

< Zum Frontend Werke Personen Drama-DB Infotext

WIENER STAATSOPER

team@die-antwort.eu abmelden

PERSONEN

« Zurück 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 Weiter »

10.123 Personen | ⊕ Person hinzufügen

A

Jasmin Avissar
Rafael Avnikjan
Ioanna Avraam
Stella Axarlis
Catherine Ayers
Hakan Aysev
Atsuko Azuma
Jael Azzaretti

B

Milica Babić
Monica Bacelli
Johann Sebastian Bach
Rudolfine / Rudi Bach
Klaus Bachler
Ingeborg Bachmann
Elis
Sieg
Hy
Sus
Han
Gab
Ildi
Chr
Han
Din
Hen
Car
Wil
Jani
Ele
Bru
Ann
Lin
Lidia Baich
Anna Baier
Hermann Baier

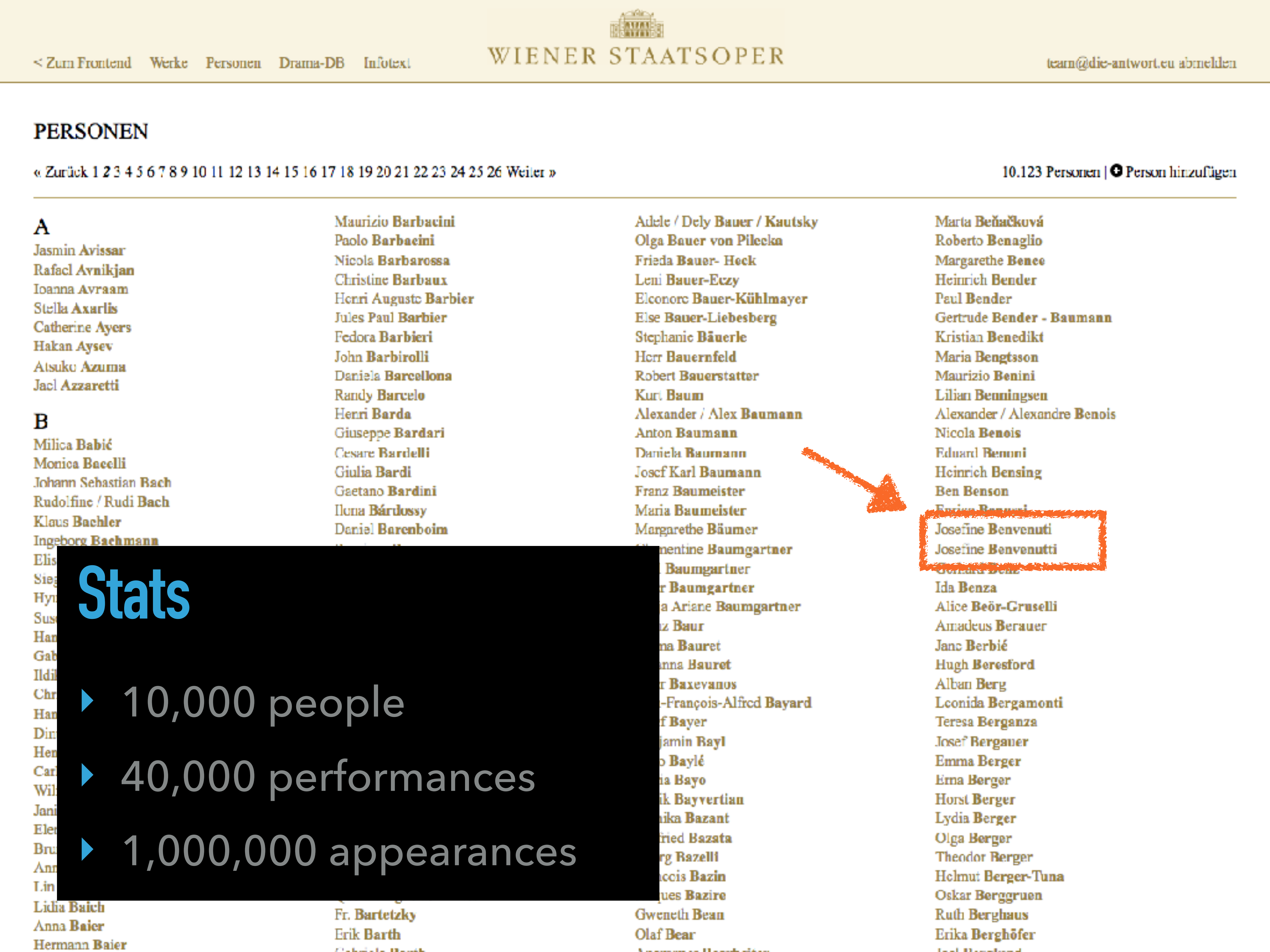
Maurizio Barbacini
Paolo Barbacini
Nicola Barbarossa
Christine Barbaux
Henri Auguste Barbier
Jules Paul Barbier
Fedora Barbieri
John Barbirolli
Daniela Barcellona
Randy Barcelo
Henri Barda
Giuseppe Bardari
Cesare Bardelli
Giulia Bardi
Gaetano Bardini
Ilona Bárdossy
Daniel Barenboim
Fr. Bartetzky
Erik Barth

Adele / Dely Bauer / Kautsky
Olga Bauer von Pilecka
Frieda Bauer- Heck
Leni Bauer-Eczy
Eleonore Bauer-Kühlmayer
Else Bauer-Liebesberg
Stephanie Bäuerle
Herr Bauernfeld
Robert Bauerstatter
Kurt Baum
Alexander / Alex Baumann
Anton Baumann
Daniela Baumann
Josef Karl Baumann
Franz Baumeister
Maria Baumeister
Margarethe Bäumer
mentine Baumgartner
Baumgartner
r Baumgartner
a Ariane Baumgartner
z Baur
na Bauret
nna Bauret
r Baxevanos
-François-Alfred Bayard
f Bayer
jamin Bayl
o Baylé
ia Bayo
ik Bayvertian
nika Bazant
ried Bazata
rg Bazelli
ccis Bazin
ues Bazire
Gweneth Bean
Olaf Bear

Marta Beňačková
Roberto Benaglio
Margarethe Bence
Heinrich Bender
Paul Bender
Gertrude Bender - Baumann
Kristian Benedikt
Maria Bengtsson
Maurizio Benini
Lilian Benningsen
Alexander / Alexandre Benois
Nicola Benois
Eduard Benoni
Heinrich Bensing
Ben Benson
Enrico Benussi
Josefine Benvenuti
Josefine Benvenutti
Gerhard Benz
Ida Benza
Alice Beör-Gruselli
Amadeus Berauer
Jane Berbié
Hugh Beresford
Alban Berg
Leonida Bergamonti
Teresa Berganza
Josef Bergauer
Emma Berger
Erna Berger
Horst Berger
Lydia Berger
Olga Berger
Theodor Berger
Helmut Berger-Tuna
Oskar Berggruen
Ruth Berghaus
Erika Berghöfer

Stats

- 10,000 people
- 40,000 performances
- 1,000,000 appearances



< Zum Frontend Werke Personen Drama-DB Infotext

WIENER STAATSOPER

team@die-antwort.eu abmelden

PERSONEN

« Zurück 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 Weiter »

10.123 Personen | ● Person hinzufügen

A

Jasmin Avissar
Rafael Avnikjan
Ioanna Avraam
Stella Axarlis
Catherine Ayers
Hakan Aysev
Atsuko Azuma
Jael Azzaretti

B

Milica Babić
Monica Bacelli
Johann Sebastian Bach
Rudolfine / Rudi Bach
Klaus Bachler
Ingeborg Bachmann
Elis
Sieg
Hy
Sus
Han
Gab
Ildi
Chr
Han
Dir
Hen
Car
Wil
Jani
Ele
Bru
Ann
Lin
Lidia Baich
Anna Baier
Hermann Baier

Maurizio Barbacini
Paolo Barbacini
Nicola Barbarossa
Christine Barbaux
Henri Auguste Barbier
Jules Paul Barbier
Fedora Barbieri
John Barbirolli
Daniela Barcellona
Randy Barcelo
Henri Barda
Giuseppe Bardari
Cesare Bardelli
Giulia Bardi
Gaetano Bardini
Ilona Bárdossy
Daniel Barenboim
Fr. Bartetzky
Erik Barth

Adele / Dely Bauer / Kautsky
Olga Bauer von Pilecka
Frieda Bauer- Heck
Leni Bauer-Eczy
Eleonore Bauer-Kühlmayer
Else Bauer-Liebesberg
Stephanie Bäuerle
Herr Bauernfeld
Robert Bauerstatter
Kurt Baum
Alexander / Alex Baumann
Anton Baumann
Daniela Baumann
Josef Karl Baumann
Franz Baumeister
Maria Baumeister
Margarethe Bäumer
mentine Baumgartner
Baumgartner
r Baumgartner
a Ariane Baumgartner
z Baur
na Bauret
nna Bauret
r Baxevanos
-François-Alfred Bayard
f Bayer
jamin Bayl
Baylé
ia Bayo
ik Bayvertian
ika Bazant
ried Bazata
rg Bazelli
ccis Bazin
ues Bazire
Gweneth Bean
Olaf Bear

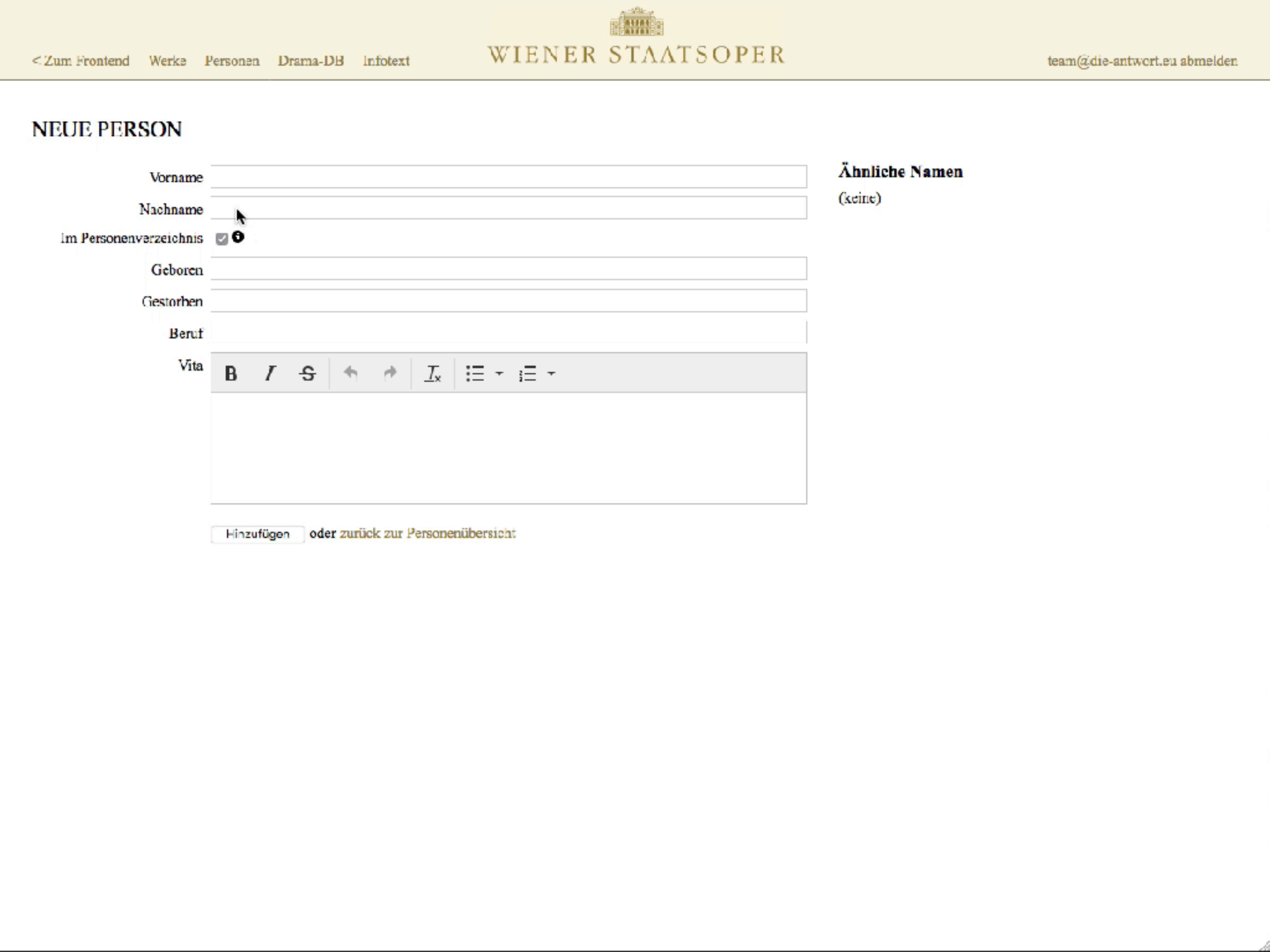
Marta Beňačková
Roberto Benaglio
Margarethe Bence
Heinrich Bender
Paul Bender
Gertrude Bender - Baumann
Kristian Benedikt
Maria Bengtsson
Maurizio Benini
Lilian Benningsen
Alexander / Alexandre Benois
Nicola Benois
Eduard Benoni
Heinrich Bensing
Ben Benson
Enrico Benvenuti
Josefine Benvenuti
Josefine Benvenutti
Gerhard Benz
Ida Benza
Alice Beör-Gruselli
Amadeus Berauer
Jane Berbié
Hugh Beresford
Alban Berg
Leonida Bergamonti
Teresa Berganza
Josef Bergauer
Emma Berger
Erna Berger
Horst Berger
Lydia Berger
Olga Berger
Theodor Berger
Helmut Berger-Tuna
Oskar Berggruen
Ruth Berghaus
Erika Berghöfer

Stats

- 10,000 people
- 40,000 performances
- 1,000,000 appearances

User Story

“When adding a new person, we need to see all existing people with similar names.”



NEUE PERSON

Vorname

Nachname

Im Personenverzeichnis ☑

Geboren

Gestorben

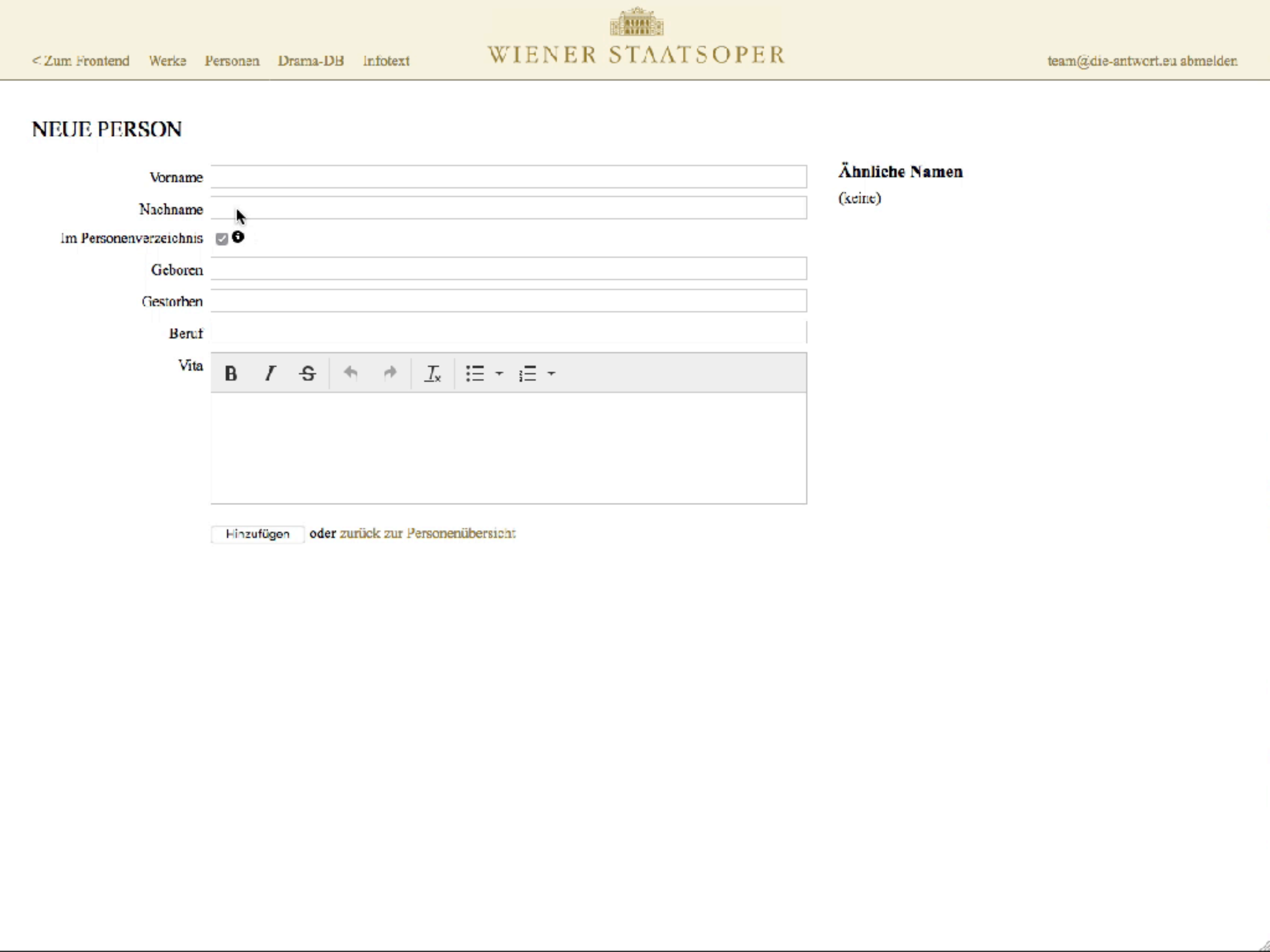
Beruf

Vita

oder zurück zur Personenübersicht

Ähnliche Namen

(keine)



NEUE PERSON

Vorname

Nachname

Im Personenverzeichnis ☑

Geboren

Gestorben

Beruf

Vita

Ähnliche Namen

(keine)

Phonetic Search

The Idea

“Assign a code to each word, such that words with similar pronunciation get identical codes.”

The Idea

“Assign a code to each word, such that words with similar pronunciation get identical codes.”

Meier	'maɪ̯ɐ

The Idea

“Assign a code to each word, such that words with similar pronunciation get identical codes.”

Meier	'maɪ̯ɐ
Mair	'maɪ̯ɐ

The Idea

“Assign a code to each word, such that words with similar pronunciation get identical codes.”

Meier	'maɪ̯ɐ
Mair	'maɪ̯ɐ
Mayer	'maɪ̯ɐ

The Idea

“Assign a code to each word, such that words with similar pronunciation get identical codes.”

Meier	'maɪ̯ɐ
Mair	'maɪ̯ɐ
Mayer	'maɪ̯ɐ
Müller	'mʏlɐ

Common Phonetic Algorithms

Soundex

- developed in the 1920s to encode surnames for use in censuses
- available in most programming languages and databases
- Rubygems: Phonetic, Soundex, Text
- simple

Soundex

⚠ **Description simplified!**

1. keep first letter
2. drop vowels
3. encode consonants as digits
4. stop after three digits (or add zeros)

Soundex

⚠ Description simplified!

1. keep first letter
2. drop vowels
3. encode consonants as digits
4. stop after three digits (or add zeros)

Meier	M600
Mair	M600
Mayer	M600
Müller	M460

Metaphone

- published in 1990
- does a better job (than Soundex) of matching similar sounding words and names
- superseded by Double Metaphone, and later Metaphone 3 (commercial closed source)
- used in many spellcheckers
- Rubygems: Phonetic, Text
- complex, no description except original implementation

```
////////////////////////////////////////////////////////////////////////////////
// Double Metaphone (c) 1998, 1999 by Lawrence Philips
//
// Slightly modified by Kevin Atkinson to fix several bugs and
// to allow it to give back more than 4 characters.
//
//
////////////////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "dmetaph.h"

#define AND &&
#define OR ||

////////////////////////////////////////////////////////////////////////////////
//
////////////////////////////////////////////////////////////////////////////////
MString::MString()
{
}

////////////////////////////////////////////////////////////////////////////////
//
////////////////////////////////////////////////////////////////////////////////
MString::MString(const char* in) : CString(in)
{
}

////////////////////////////////////////////////////////////////////////////////
//
////////////////////////////////////////////////////////////////////////////////
MString::MString(const CString& in) : CString(in)
{
}

////////////////////////////////////////////////////////////////////////////////
//
////////////////////////////////////////////////////////////////////////////////
bool MString::SlavoGermanic()
{
        if((Find('W') > -1) OR (Find('K') > -1) OR (Find("CZ") > -1) OR (Find("WITZ") > -1))
                return TRUE;

        return FALSE;
}

////////////////////////////////////////////////////////////////////////////////
//
////////////////////////////////////////////////////////////////////////////////
inline void MString::MetaphAdd(const char* main)
{
        if(*main)
        {
                primary += main;
                secondary += main;
        }
}

////////////////////////////////////////////////////////////////////////////////
//
////////////////////////////////////////////////////////////////////////////////
inline void MString::MetaphAdd(const char* main, const char* alt)
{
        if(*main)
                primary += main;
        if(*alt)
        {
```

```
                        break;
                }

                //polish e.g. 'filipowicz'
                if(StringAt(current, 4, "WICZ", "WITZ", ""))
                {
                        MetaphAdd("TS", "FX");
                        current +=4;
                        break;
                }

                //else skip it
                current +=1;
                break;

            case 'X':
                //french e.g. breaux
                if(!((current == last)
                        AND (StringAt((current - 3), 3, "IAU", "EAU", "")
                                OR StringAt((current - 2), 2, "AU", "OU", ""))) )
                        MetaphAdd("KS");

                if(StringAt((current + 1), 1, "C", "X", ""))
                        current += 2;
                else
                        current += 1;
                break;

            case 'Z':
                //chinese pinyin e.g. 'zhao'
                if(GetAt(current + 1) == 'H')
                {
                        MetaphAdd("J");
                        current += 2;
                        break;
                }else
                        if(StringAt((current + 1), 2, "ZO", "ZI", "ZA", "")
                            OR (SlavoGermanic() AND ((current > 0) AND GetAt(current - 1) != 'T')))
                        {
                            MetaphAdd("S", "TS");
                        }
                        else
                            MetaphAdd("S");

                if(GetAt(current + 1) == 'Z')
                        current += 2;
                else
                        current += 1;
                break;

            default:
                current += 1;
        }
    }

    metaph  = primary;
    //only give back 4 char metaph
    //if(metaph.GetLength() > 4)
    //      metaph.SetAt(4,'\0');
    if(alternate)
    {
        metaph2 = secondary;
        //if(metaph2.GetLength() > 4)
        //      metaph2.SetAt(4,'\0');
    }

}
```

Cologne Phonetics (Kölner Phonetik)

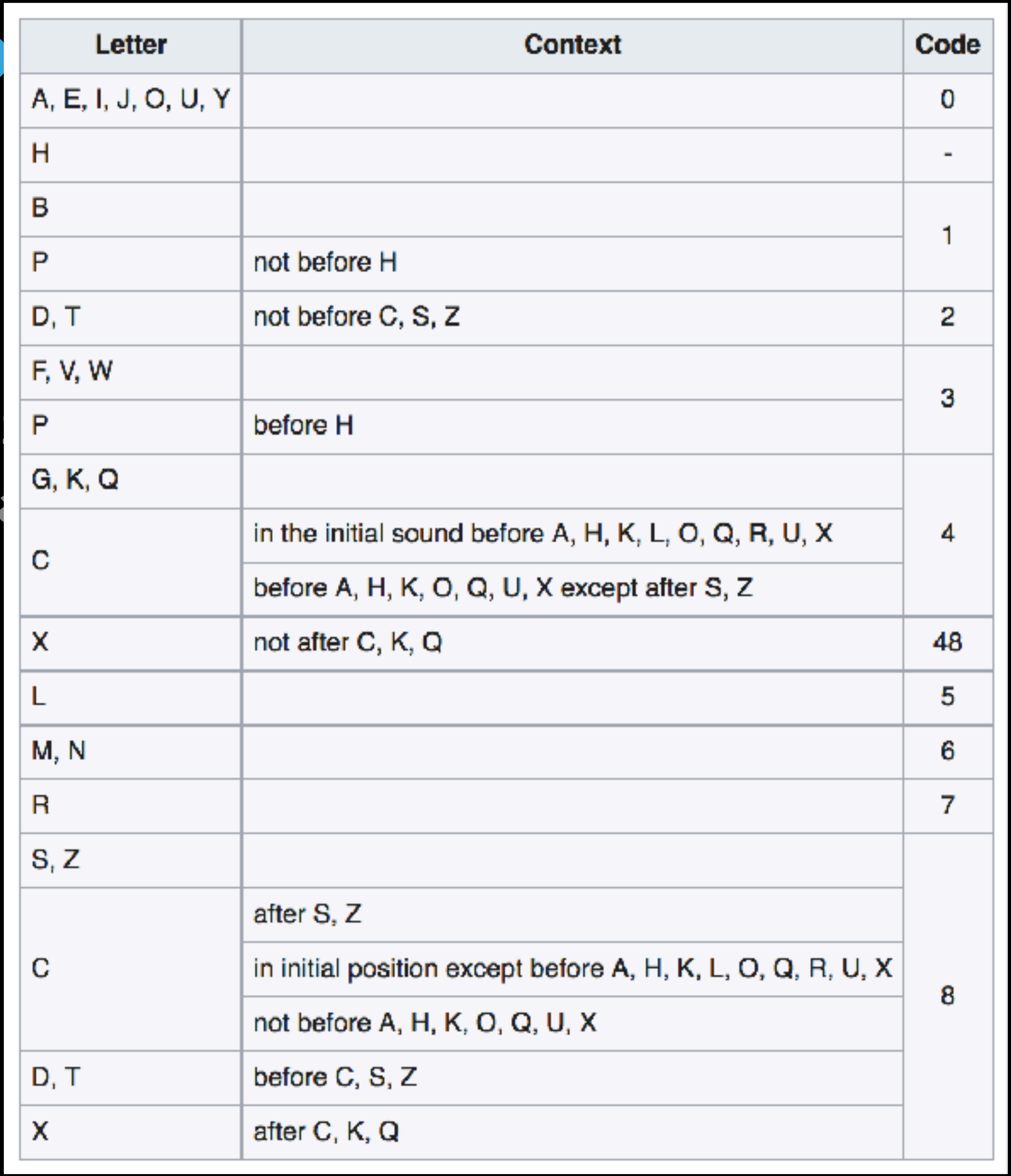
- published in 1968
- optimized for German
- not as complex as Metaphone
- few implementations available (Perl, PHP)
- Rubygems: –

Cologne Phonetics

1. encode letters as digits
2. remove repeated digits
3. remove zeros
   (except at the beginning)

Cologne P

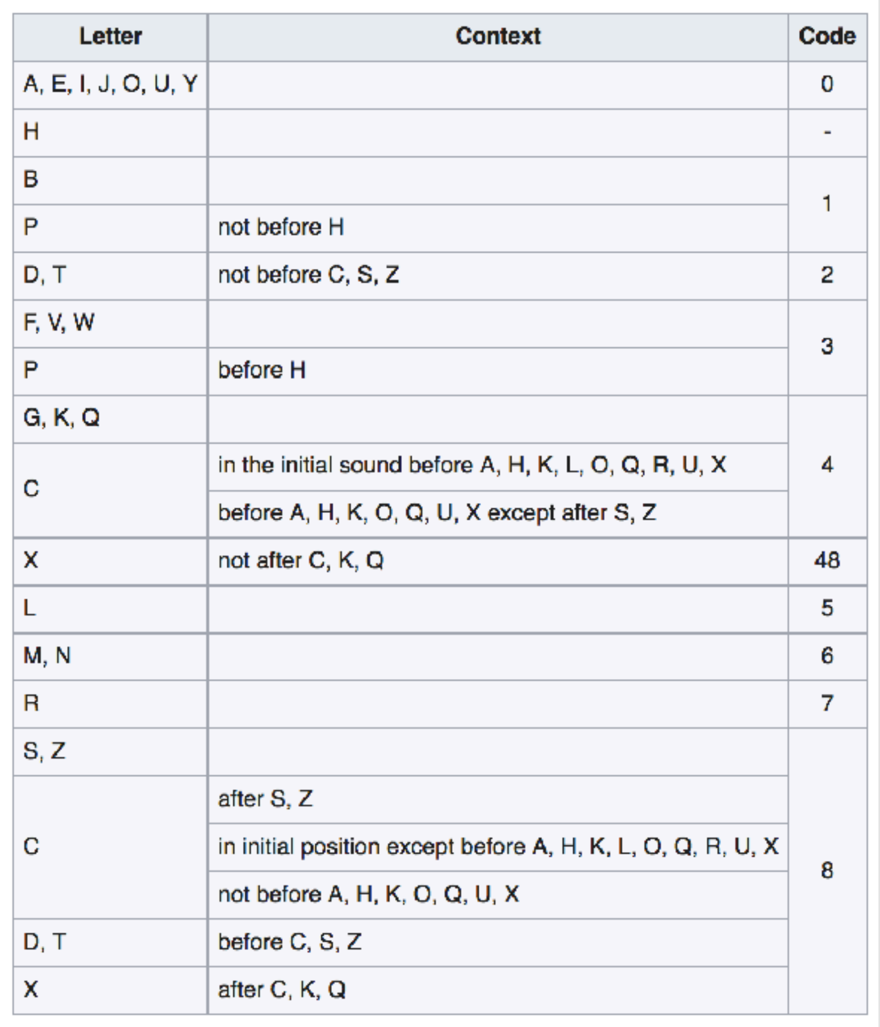
1. encode
2. remove
3. remove
   (except



Letter	Context	Code
A, E, I, J, O, U, Y		0
H		-
B		1
P	not before H	
D, T	not before C, S, Z	2
F, V, W		3
P	before H	
G, K, Q		4
C	in the initial sound before A, H, K, L, O, Q, R, U, X	
	before A, H, K, O, Q, U, X except after S, Z	
X	not after C, K, Q	48
L		5
M, N		6
R		7
S, Z		8
C	after S, Z	
	in initial position except before A, H, K, L, O, Q, R, U, X	
	not before A, H, K, O, Q, U, X	
D, T	before C, S, Z	
X	after C, K, Q	

Cologne Phonetics

1. encode letters as digits
2. remove repeated digits
3. remove zeros
   (except at the beginning)



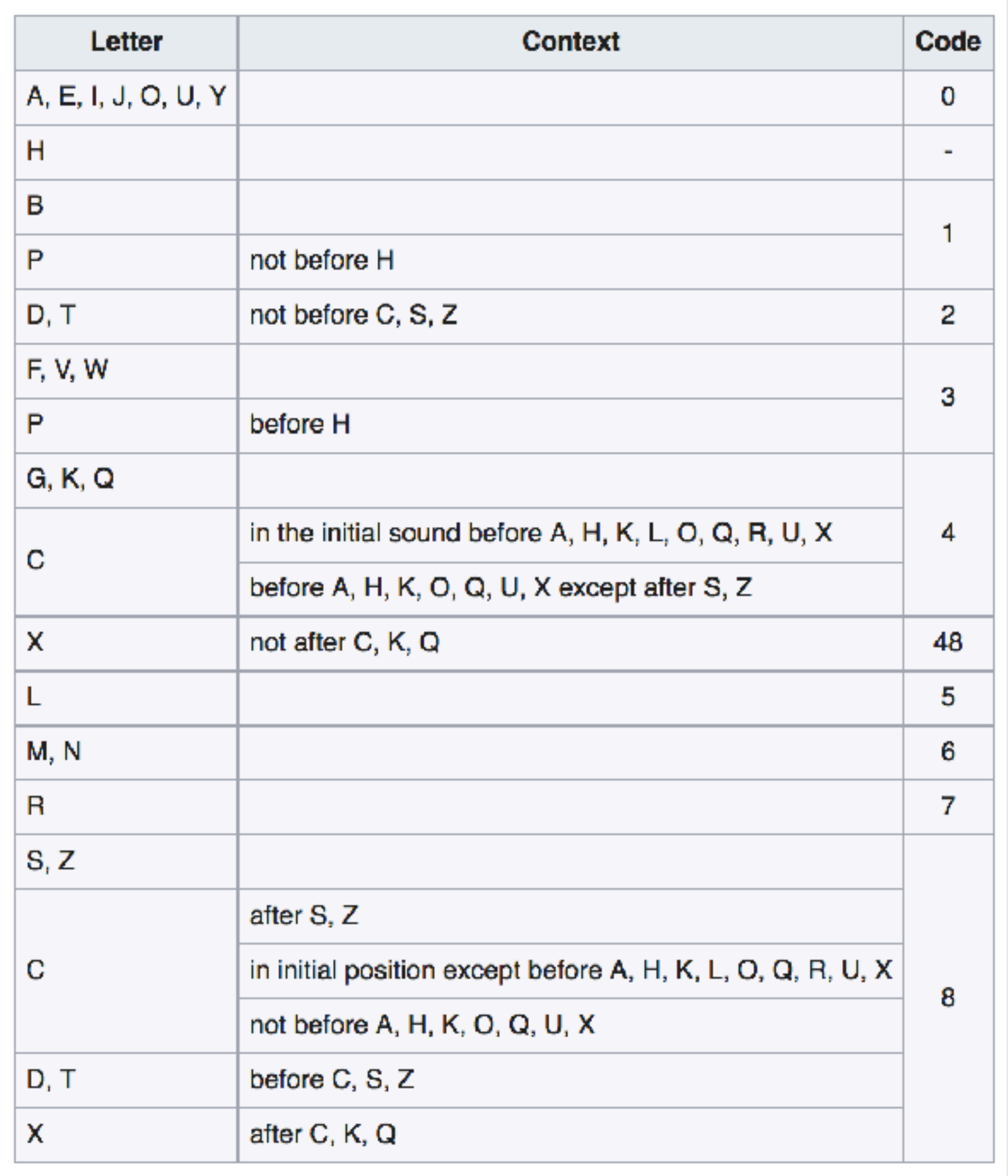
Letter	Context	Code
A, E, I, J, O, U, Y		0
H		-
B		1
P	not before H	
D, T	not before C, S, Z	2
F, V, W		3
P	before H	
G, K, Q		4
C	in the initial sound before A, H, K, L, O, Q, R, U, X	
	before A, H, K, O, Q, U, X except after S, Z	
X	not after C, K, Q	48
L		5
M, N		6
R		7
S, Z		8
C	after S, Z	
	in initial position except before A, H, K, L, O, Q, R, U, X	
	not before A, H, K, O, Q, U, X	
D, T	before C, S, Z	
X	after C, K, Q	

Meier	67
Mair	67
Mayer	67
Müller	657

Let's make a Rubygem!

```ruby
module ColognePhonetics
  module Rules
    def self.define(&block)
      @rules = DSL.new(&block).rules
    end

    def self.apply_to(string)
      string = string.downcase.tr('ÄÖÜ', 'äöü') # Ruby < 2.3 downcases ASCII characters only
      chars = [nil] + string.chars + [nil]
      chars.each_cons(3).map{ |prev_char, char, next_char|
        code_for(prev_char, char, next_char)
      }.join
    end

    def self.code_for(prev_char, char, next_char)
      @rules.each do |matcher, code|
        return code if matcher.call(prev_char, char, next_char)
      end
      nil
    end

   class DSL
      attr_reader :rules

      def initialize(&block)
        @rules = []
        instance_exec(&block)
      end

      def change(chars, to:, before: nil, not_before: nil, after: nil, not_after: nil, initial: nil)
        matcher = ->(prev_char, char, next_char){
          return unless chars.include?(char)
          return if initial && prev_char
          return if before && (!next_char || !before.include?(next_char))
          return if not_before && next_char && not_before.include?(next_char)
          return if after && (!prev_char || !after.include?(prev_char))
          return if not_after && prev_char && not_after.include?(prev_char)
          true
        }
        @rules << [matcher, to]
      end
    end
  end
end
```

```ruby
Rules.define do
  change 'aeijouy', to: '0'
  change 'äöü'    , to: '0'
  change 'h'      , to: ''
  change 'b'      , to: '1'
  change 'p'      , to: '1', not_before: 'h'
  change 'dt'     , to: '2', not_before: 'csz'
  change 'fvw'    , to: '3'
  change 'p'      , to: '3', before: 'h'
  change 'gkq'    , to: '4'
  change 'c'      , to: '4', initial: true, before: 'ahkloqrux'
  change 'c'      , to: '4', before: 'ahkoqux', not_after: 'sz'
  change 'x'      , to: '48', not_after: 'ckq'
  change 'l'      , to: '5'
  change 'mn'     , to: '6'
  change 'r'      , to: '7'
  change 'sz'     , to: '8'
  change 'ß'      , to: '8'
  change 'c'      , to: '8', after: 'sz'
  change 'c'      , to: '8', initial: true, not_before: 'ahkloqrux'
  change 'c'      , to: '8', not_before: 'ahkoqux'
  change 'dt'     , to: '8', before: 'csz'
  change 'x'      , to: '8', after: 'ckq'
end
```



Letter	Context	Code
A, E, I, J, O, U, Y		0
H		-
B		1
P	not before H	
D, T	not before C, S, Z	2
F, V, W		3
P	before H	
G, K, Q		4
C	in the initial sound before A, H, K, L, O, Q, R, U, X	
	before A, H, K, O, Q, U, X except after S, Z	
X	not after C, K, Q	48
L		5
M, N		6
R		7
S, Z		8
C	after S, Z	
	in initial position except before A, H, K, L, O, Q, R, U, X	
	not before A, H, K, O, Q, U, X	
D, T	before C, S, Z	
X	after C, K, Q	

```ruby
Rules.define do
  change 'aeijouy', to: '0'
  change 'äöü'    , to: '0'
  change 'h'      , to: ''
  change 'b'      , to: '1'
  change 'p'      , to: '1', not_before: 'h'
  change 'dt'     , to: '2', not_before: 'csz'
  change 'fvw'    , to: '3'
  change 'p'      , to: '3', before: 'h'
  change 'gkq'    , to: '4'
  change 'c'      , to: '4', initial: true, before: 'ahkloqrux'
  change 'c'      , to: '4', before: 'ahkoqux', not_after: 'sz'
  change 'x'      , to: '48', not_after: 'ckq'
  change 'l'      , to: '5'
  change 'mn'     , to: '6'
  change 'r'      , to: '7'
  change 'sz'     , to: '8'
  change 'ß'      , to: '8'
  change 'c'      , to: '8', after: 'sz'
  change 'c'      , to: '8', initial: true, not_before: 'ahkloqrux'
  change 'c'      , to: '8', not_before: 'ahkoqux'
  change 'dt'     , to: '8', before: 'csz'
  change 'x'      , to: '8', after: 'ckq'
end
```

Cologne Phonetics (Kölner Phonetik)

- published in 1968
- optimized for German
- not as complex as Metaphone
- few implementations available (Perl, PHP)
- Rubygems: –

Cologne Phonetics (Kölner Phonetik)

- published in 1968
- optimized for German
- not as complex as Metaphone
- few implementations available (Perl, PHP)
- Rubygems: **Cologneg Phonetics** 😎

```ruby
require 'cologne_phonetics'

ColognePhonetics.encode('Meier') # => "67"
ColognePhonetics.encode('Müller-Lüdenscheid') # => "65752682"
```

Back to the Opera . . .

Problem 1: Complex Last Names

Problem 1: Complex Last Names

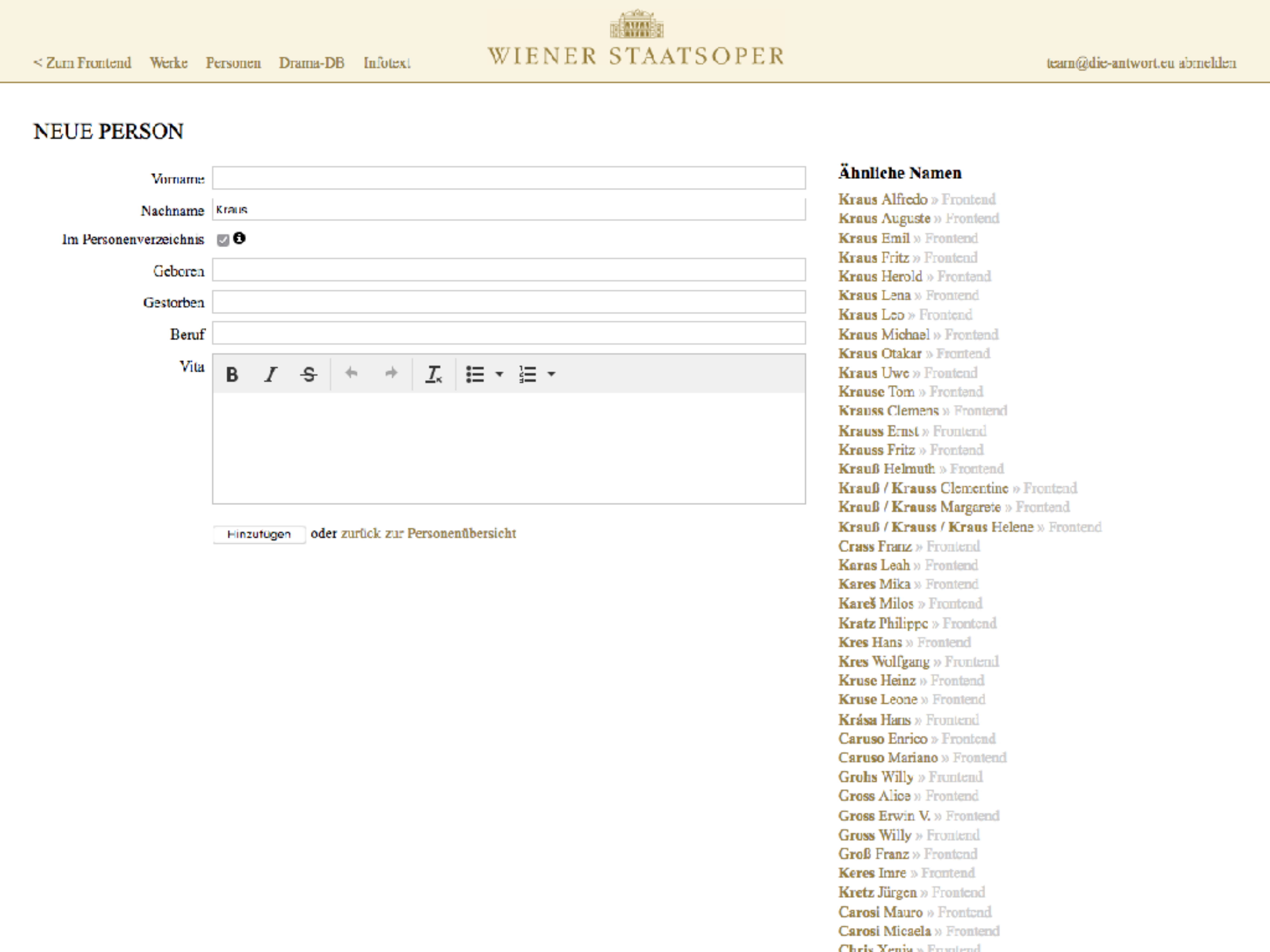
- da Ponte
- de la Mora
- del Grande
- du Roullet
- Garcia-Torres
- Gullberg Jensen
- Hellmesberger jun.
- La Fosse
- Rabl von Kriesten
- Saint Laurent
- Strauß (Vater)
- Te Kanawa
- van Beethoven
- Vernoy de Saint-Georges
- von der Damerau

Solution: Normalize!

```ruby
def normalize(name)
  normalized_name = name.split(%r{\s*/\s*})[0] || ''
  normalized_name.delete!("'’‘")  # Encode "O'Hara" as "Ohara"
  normalized_name.gsub!(/(.)\(.+$/, '\1')
  normalized_name.gsub!(/^(de (la|los)|dal?|del?|della|di|du|la|le|van der|von der) /i){ |m|
    m.delete(' ')
  }
  normalized_name.gsub!(/^(van|von) /i, '')
  normalized_name.gsub!(/[^[[:alpha:]]]+/, ' ')
  I18n.transliterate(normalized_name)
end
```

(… and treat name parts seperately)

Problem 2: How to Sort Similar Names?



NEUE PERSON

Vorname

Nachname Kraus

Im Personenverzeichnis ☑

Geboren

Gestorben

Beruf

Vita

oder zurück zur Personenübersicht

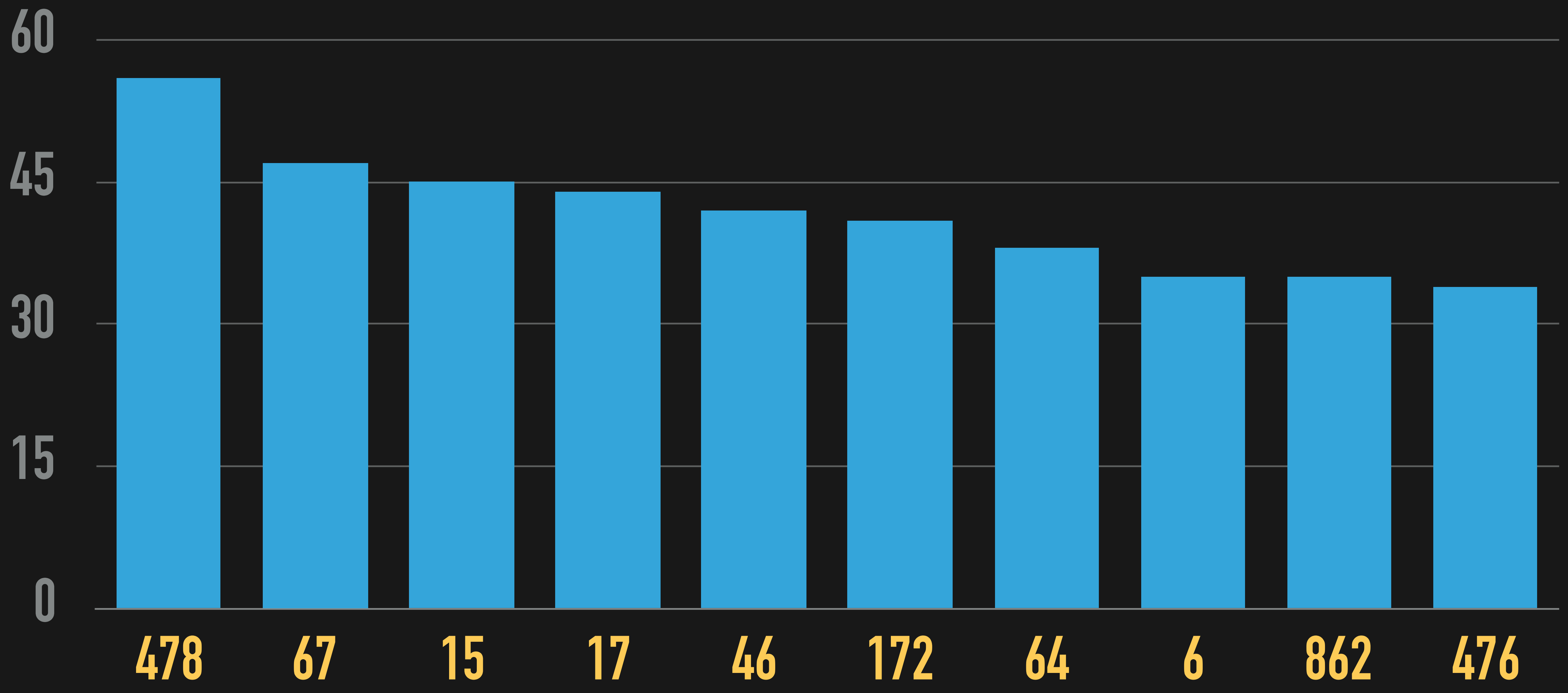
Ähnliche Namen

Kraus Alfredo » Frontend
Kraus Auguste » Frontend
Kraus Emil » Frontend
Kraus Fritz » Frontend
Kraus Herold » Frontend
Kraus Lena » Frontend
Kraus Leo » Frontend
Kraus Michael » Frontend
Kraus Otakar » Frontend
Kraus Uwe » Frontend
Krause Tom » Frontend
Krauss Clemens » Frontend
Krauss Ernst » Frontend
Krauss Fritz » Frontend
Krauß Helmuth » Frontend
Krauß / Krauss Clementine » Frontend
Krauß / Krauss Margarete » Frontend
Krauß / Krauss / Kraus Helene » Frontend
Crass Franz » Frontend
Karas Leah » Frontend
Kares Mika » Frontend
Kareš Milos » Frontend
Kratz Philippe » Frontend
Kres Hans » Frontend
Kres Wolfgang » Frontend
Kruse Heinz » Frontend
Kruse Leone » Frontend
Krása Hans » Frontend
Caruso Enrico » Frontend
Caruso Mariano » Frontend
Grohs Willy » Frontend
Gross Alice » Frontend
Gross Erwin V. » Frontend
Gross Willy » Frontend
Groß Franz » Frontend
Keres Imre » Frontend
Kretz Jürgen » Frontend
Carosi Mauro » Frontend
Carosi Micaela » Frontend
Chris Xenia » Frontend

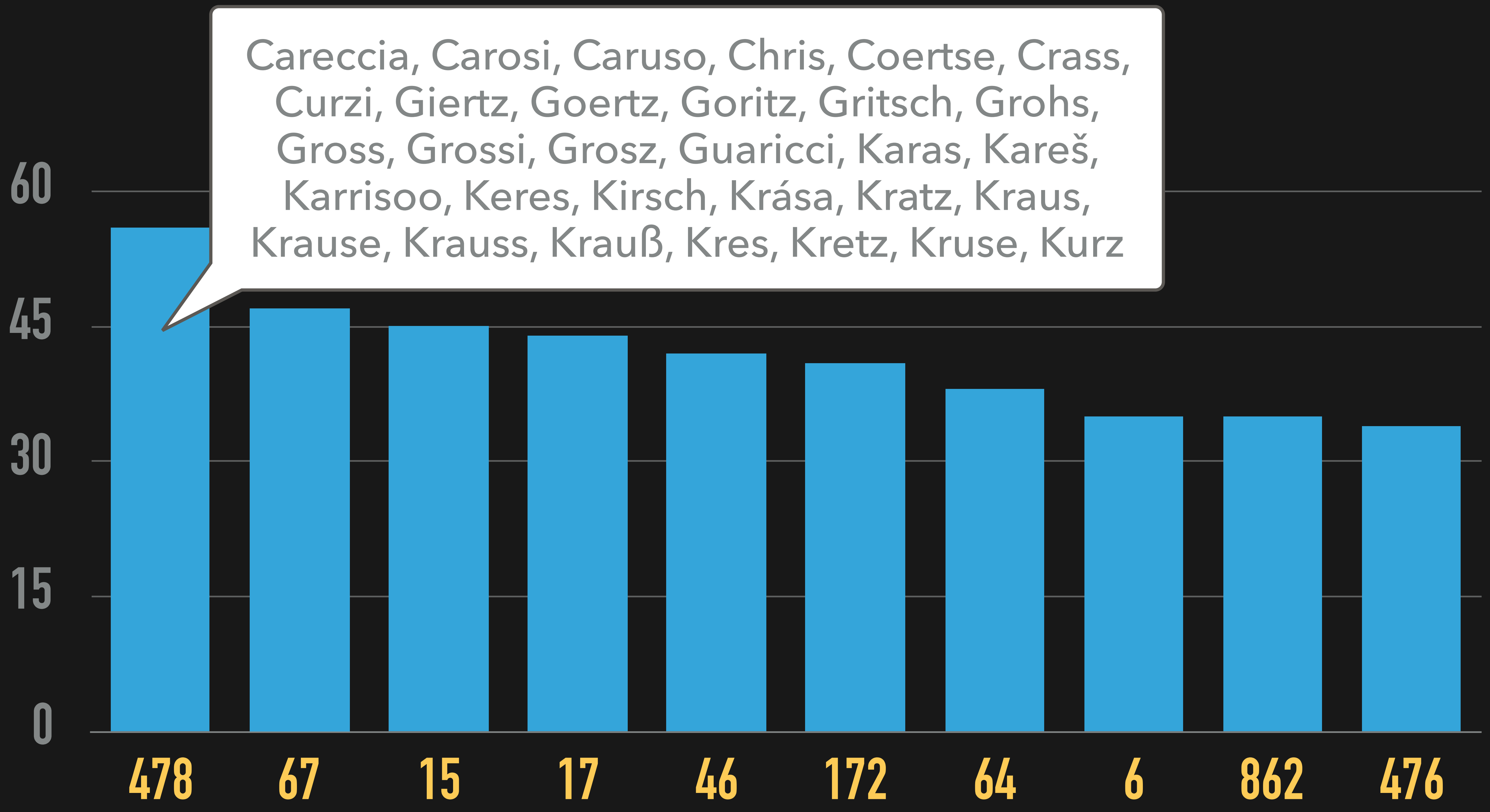
Solution: Levenshtein-Distance!

```
require 'damerau-levenshtein'

DamerauLevenshtein.distance('Kraus', 'Kraus') # => 0
DamerauLevenshtein.distance('Kraus', 'Krauß') # => 1
DamerauLevenshtein.distance('Kraus', 'Karas') # => 2
DamerauLevenshtein.distance('Kraus', 'Caruso') # => 3
DamerauLevenshtein.distance('Kraus', 'Grosz') # => 4
```

Top Ten Encoded Name Clusters

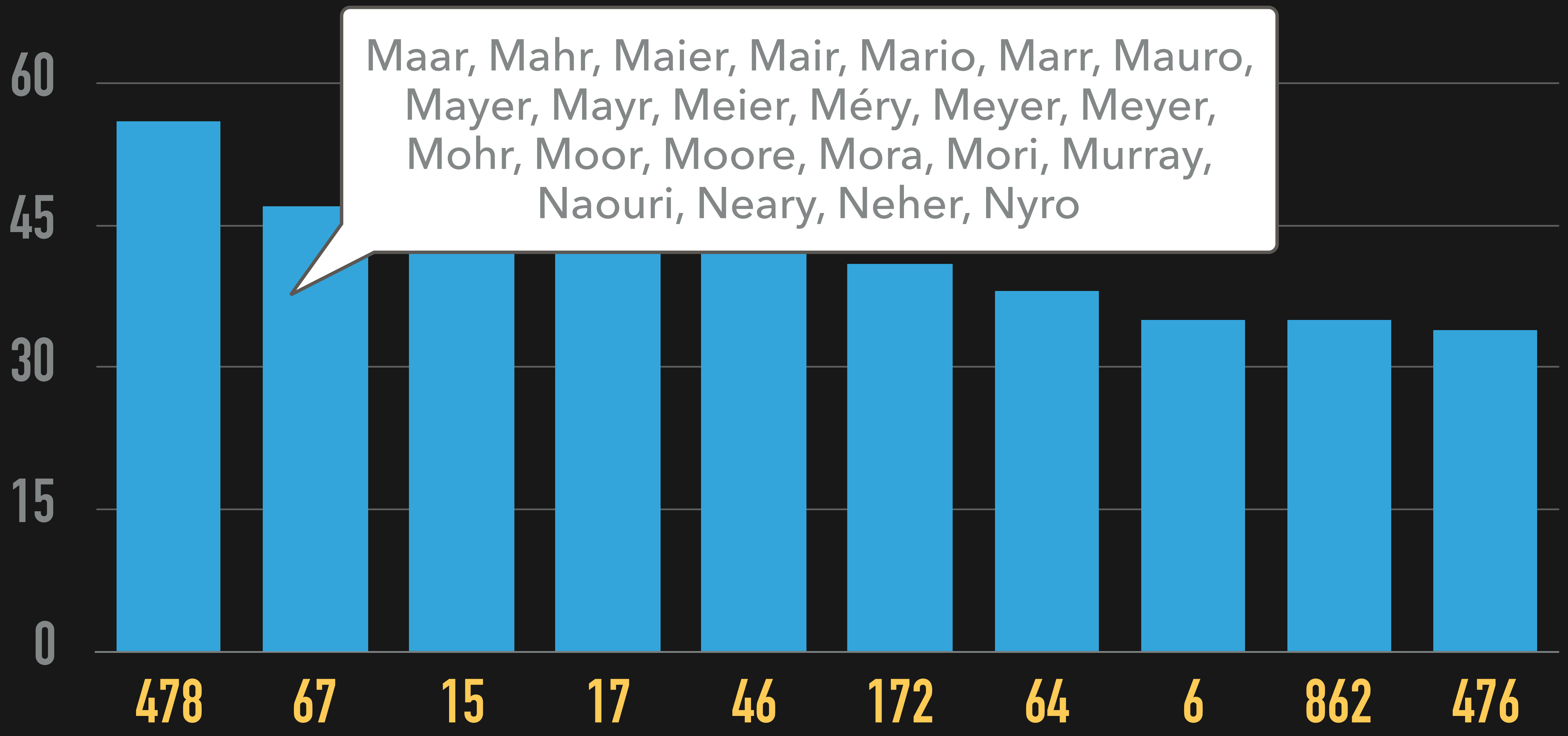


Top Ten Encoded Name Clusters



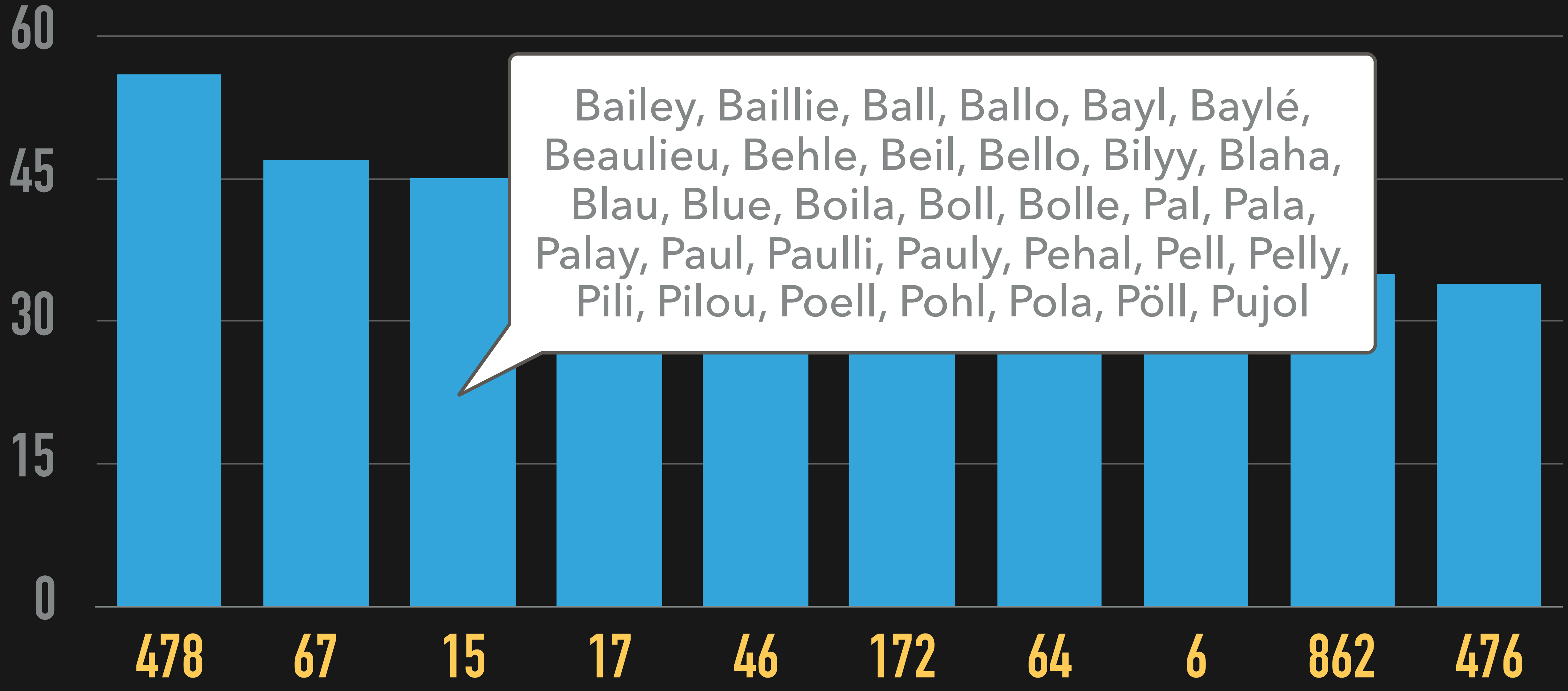
Careccia, Carosi, Caruso, Chris, Coertse, Crass, Curzi, Giertz, Goertz, Goritz, Gritsch, Grohs, Gross, Grossi, Grosz, Guaricci, Karas, Kareš, Karrisoo, Keres, Kirsch, Krása, Kratz, Kraus, Krause, Krauss, Krauß, Kres, Kretz, Kruse, Kurz

Top Ten Encoded Name Clusters



Maar, Mahr, Maier, Mair, Mario, Marr, Mauro, Mayer, Mayr, Meier, Méry, Meyer, Meyer, Mohr, Moor, Moore, Mora, Mori, Murray, Naouri, Neary, Neher, Nyro

Top Ten Encoded Name Clusters



Bailey, Baillie, Ball, Ballo, Bayl, Baylé, Beaulieu, Behle, Beil, Bello, Bilyy, Blaha, Blau, Blue, Boila, Boll, Bolle, Pal, Pala, Palay, Paul, Paulli, Pauly, Pehal, Pell, Pelly, Pili, Pilou, Poell, Pohl, Pola, Pöll, Pujol

Thank you!

Stefan Daschek / @noniq