University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: April 1st, 2025

Group Number: 87

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Karli Winkler	69709426	r6y7o	karliswinkler@gmail.com
Mostafa Mostafa	42907808	y3x7z	mxstafa@student.ubc.ca
Arman Thariani	46081311	j5v0v	arman.thariani@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Final Project Description

The project is a fieldwork tool for linguistics researchers, it allows them to view and update language information with an interface and keep track of the speakers of said languages. Our project allows for the following functionality:

- View Languages and the language families
- View total speakers across all dialects to view popularity of language
- Show languages that have ancient writing systems (which may have historical data)
- Add, update, and remove languages
- Search for languages by the age of the writing system, the status and the name of the language to facilitate finding information
- View phonetic data by selecting which distinctive features of phonemes to view to use as reference when making transcriptions
- View speakers of a language and which dialect they speak to determine who to study for research
- View the country with the highest number of speakers on file to determine where to conduct research
- View words in a language that are shared across all dialects for linguistic comparison

Final Schema vs. Submitted Schema

The schema is the same as the last checkpoint which is the tables produced by the BCNF decomposition.

Legend: <u>Primary Key</u>, **Foreign key**

1. Language(<u>Name</u>: VARCHAR, Status: VARCHAR, **FamilyName**: VARCHAR)

2. Family(<u>Name</u>: VARCHAR, Origin: VARCHAR)

3. WritingSystem(<u>Name:</u> VARCHAR, Type: VARCHAR, Age: INTEGER)

4. Uses(**<u>WSName</u>**: VARCHAR, **<u>LanguageName</u>**: VARCHAR)

5. Speaker(<u>ID</u>: VARCHAR, Name: VARCHAR)

6. Country(<u>Name</u>: VARCHAR, Continent: VARCHAR, Population: INTEGER)

7. SpokenBy(**<u>SpeakerID</u>**: VARCHAR, **<u>DialectName</u>**: VARCHAR, **<u>LanguageName</u>**: VARCHAR)

8. SpokenIn(**<u>CountryName</u>**: VARCHAR, **<u>DialectName</u>**: VARCHAR, **<u>LanguageName</u>**: VARCHAR)

9. Defines(**<u>WordID</u>**: INTEGER, **<u>DialectName</u>**: VARCHAR, **<u>LanguageName</u>**: VARCHAR)

10. Contains(**IPANumber**: INTEGER, **WordID:** INTEGER)

11. Phoneme(IPANumber: INTEGER, **Unicode**: INTEGER)

12. Vowel(IPANumber: INTEGER, Height: VARCHAR, Backness: VARCHAR, Rounded: BOOLEAN)

13. Consonant(IPANumber: INTEGER, Voiced: BOOLEAN, **Place**: VARCHAR, Manner: VARCHAR)

14. Dialect(Name: VARCHAR, **LanguageName**: VARCHAR, Population: INTEGER)

15. IsFrom(**SpeakerID**: INTEGER, **CountryName**: VARCHAR)

16. Word(ID: VARCHAR, WrittenForm: NVARCHAR, Meaning: VARCHAR)

17. Unicode(UnicodeID, UnicodeBlock),
18. PlaceInfo(Place, Coronal)

Our setup script (sql_setup_scripts/create_and_populate.sql) produces the following (some of the Unicode in the Word table did not generate properly):

Contains

IPANUMBER	WORDID
101	1
102	1
103	1
104	6
106	3
107	5
108	8
109	9
110	10

Language

NAME	STATUS	FAMILYNAME
English	International	Indo-European
French	International	Indo-European
German	International	Indo-European
Arabic	International	Afro-Asiatic
Swahili	National	Niger-Congo
Halkomelem	Moribund	Salish
Korean	National	Koreanic

Consonant

IPANUMBER	VOICED	PLACE	MANNER
101	1	bilabial	plosive
103	1	alveolar	nasal
104	0	glottal	plosive
106	1	bilabial	nasal
107	1	uvular	trill

Country

NAME	CONTINENT	POPULATION
Canada	North America	38000000
France	Europe	67000000
Kenya	Africa	55000000
South Korea	Asia	52000000
Saudi Arabia	Asia	35000000
Democratic Republic of the Congo	Africa	92000000
Egypt	Africa	107000000
United Kingdom	Europe	68350000

Defines

WORDID	DIALECTNAME	LANGUAGENAME
1	Belgian	French
1	Metropolitan	French
1	Quebecois	French
2	British	English
2	Canadian	English
3	Coastal	Swahili
4	Jeju	Korean
5	Egyptian	Arabic
5	Gulf	Arabic
6	Downriver	Halkomelem
7	Belgian	French
7	Metropolitan	French
7	Quebecois	French
8	Coastal	Swahili
9	Jeju	Korean
10	Egyptian	Arabic
10	Gulf	Arabic
11	Quebecois	French
12	Belgian	French
13	Canadian	English

Dialect

NAME	LANGUAGENAME	POPULATION
Quebecois	French	7700000
Metropolitan	French	64000000
Coastal	Swahili	15000000
Gulf	Arabic	36000000
Egyptian	Arabic	40000000
Jeju	Korean	5000
Pyojuneo	Korean	51000000
Canadian	English	30000000
British	English	60000000
Downriver	Halkomelem	4
Belgian	French	4500000
Belgian	German	77527

Family

NAME	ORIGIN
Indo-European	Pontic-Caspian steppe
Afro-Asiatic	East Africa
Niger-Congo	Savanna belt of West Africa
Salish	Pacific Northwest
Koreanic	Korean Peninsula

Isfrom

SPEAKERID	COUNTRYNAME
1	Canada
2	France
3	Kenya
4	South Korea
5	Kenya
6	Canada
7	France
8	Saudi Arabia

Phoneme

IPANUMBER	UNICODE
101	98
102	111
103	110
104	660
105	629
106	109
107	641
108	616
109	652
110	712

Placeinfo

PLACE	CORONAL
bilabial	0
alveolar	1
glottal	0
uvular	0
palatal	1

Speaker

ID	NAME
1	John Smith
2	Marie Dubois
3	Ali Hassan
4	Ji-hoon Park
5	Amina Mwangi
6	William Johnson
7	Sophie Lefevre
8	Fatima Al-Farsi

Spokenby

SPEAKERID	DIALECTNAME	LANGUAGENAME
1	British	English
1	Canadian	English
1	Quebecois	French
2	British	English
2	Quebecois	French
3	Gulf	Arabic
4	Jeju	Korean
5	Coastal	Swahili
6	Canadian	English
6	Downriver	Halkomelem
7	British	English
7	Coastal	Swahili
7	Metropolitan	French
8	British	English
8	Gulf	Arabic

Spokenin

COUNTRYNAME	DIALECTNAME	LANGUAGENAME
Canada	Canadian	English
Canada	Downriver	Halkomelem
Canada	Quebecois	French
Democratic Republic of the Congo	Coastal	Swahili
France	Quebecois	French
Kenya	Coastal	Swahili
Saudi Arabia	Gulf	Arabic
South Korea	Jeju	Korean
United Kingdom	British	English

Unicode

UNICODEID	UNICODEBLOCK
98	Latin
111	Latin
110	Latin
660	Arabic
629	Arabic
109	Latin
641	Arabic
616	Arabic
652	Latin
712	Arabic

Uses

WSNAME	LANGUAGENAME
Arabic	Arabic
Latin	English
Latin	French
North American Phonetic Alphabet	Halkomelem
Chinese	Korean
Hangul	Korean
Arabic	Swahili
Latin	Swahili

Vowel

IPANUMBER	HEIGHT	BACKNESS	ROUNDED
102	close-mid	back	1
105	close-mid	central	1
108	close	central	0
109	open-mid	back	1
110	high-mid	front	0

Word

ID	WRITTENFORM	MEANING
1	bonjour	hello
2	hello	greeting
3	salama	peace
4	󠀠󠀠	hello
5	ابحرم	hello
6	həṅq̓	hello
7	merci	thank you
8	habari	news
9	󠀠	yes
10	أركش	thank you
11	érablière	maple grove
12	souper	dinner
13	toque	hat

Writingsystem

NAME	TYPE	AGE
Latin	Alphabet	2700
Arabic	Abjad	1800
North American Phonetic Alphabet	Alphabet	160
Hangul	Featural Alphabet	582
Chinese	Logographic	3000

SQL Queries

Query number	File	Line #
1	CPSC304_Node_Project-main/appService.js	111
2	CPSC304_Node_Project-main/appService.js	124
3	CPSC304_Node_Project-main/appService.js	278
4	CPSC304_Node_Project-main/appService.js	173
5	CPSC304_Node_Project-main/appService.js	201
6	CPSC304_Node_Project-mai	152

	n/appService.js	
7	CPSC304_Node_Project-main/appService.js	290
8	CPSC304_Node_Project-main/appService.js	304
9	CPSC304_Node_Project-main/appService.js	224
10	CPSC304_Node_Project-main/appService.js	256

Query 7 - Aggregation with GROUP BY

```
SELECT Language.Name, SUM(Dialect.population)
        FROM Language, Dialect
        WHERE Dialect.LanguageName=Language.name
        GROUP BY Language.Name;
```
This query takes the sum of all the speaker populations for all dialects of a language to get the total number of speakers of the language.

Query 8 - Aggregation with HAVING

```
SELECT max_vals.name, max_vals.age, WS.Name AS WritingSystemName
        FROM (
            SELECT Language.Name AS name, MAX(WritingSystem.Age) AS age
            FROM Language
            JOIN Uses ON Language.Name = Uses.LanguageName
            JOIN WritingSystem ON WritingSystem.Name = Uses.WSName
            GROUP BY Language.Name
            HAVING MAX(WritingSystem.Age) > 1000
        ) max_vals
        JOIN Uses ON Uses.LanguageName = max_vals.name
        JOIN WritingSystem WS ON WS.Name = Uses.WSName
        WHERE WS.Age = max_vals.age;
```
This query finds all languages that have a writing system over 1000 years old (ancient) and displays those languages and the associated writing system.

Query 9 - Nested aggregation with GROUP BY

```
SELECT IsFrom.CountryName, COUNT(*) AS SpeakerCount
    FROM Dialect
```

```
      INNER JOIN SpokenBy ON SpokenBy.LanguageName = Dialect.LanguageName
                        AND SpokenBy.DialectName = Dialect.Name
      INNER JOIN Speaker ON SpokenBy.SpeakerID = Speaker.ID
      INNER JOIN IsFrom ON IsFrom.SpeakerID = Speaker.ID
      WHERE Dialect.LanguageName = :languageName
      GROUP BY IsFrom.CountryName
      HAVING COUNT(*) = (
          SELECT MAX(SpeakerCount)
          FROM (
              SELECT COUNT(*) AS SpeakerCount
              FROM Dialect
              INNER JOIN SpokenBy ON SpokenBy.LanguageName = Dialect.LanguageName
                                 AND SpokenBy.DialectName = Dialect.Name
              INNER JOIN Speaker ON SpokenBy.SpeakerID = Speaker.ID
              INNER JOIN IsFrom ON IsFrom.SpeakerID = Speaker.ID
              WHERE Dialect.LanguageName = :languageName
              GROUP BY IsFrom.CountryName
          )
      );
```

This query finds the country from which the most documented speakers of a language (speakers in the database not population) are from. That is, whichever country has the most speakers of a language in the Speaker table, that gets returned.

Query 10 - Division

```
SELECT Word.WRITTENFORM, Word.Meaning
          FROM Word
          WHERE NOT EXISTS (
              (SELECT Dialect.NAME, Dialect.LANGUAGENAME
               FROM DIALECT
               WHERE Dialect.LANGUAGENAME = :languageName)
             MINUS
              (SELECT Defines.DIALECTNAME, Defines.LANGUAGENAME
               FROM DEFINES
               WHERE Defines.WORDID = Word.ID
               AND Defines.LANGUAGENAME = :languageName)
          );
```

This query finds all the words in a language that are common across all dialects. That is, the query takes all the words that all dialects of a specific language have defined.