University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: February 28th, 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

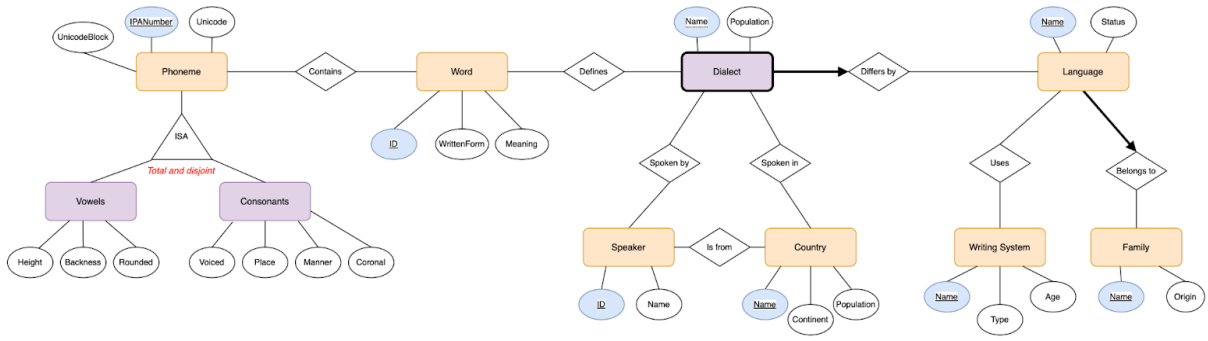
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

University of British Columbia, Vancouver
Department of Computer Science

Project Summary

The database can be used as a tool for linguistic researchers to record data collected in the field when encountering new words, phonemes, and dialects of various languages. Researchers will also be able to record information upon meeting speakers of different languages, including their country of origin and any other relevant personal information. Outside of the field, linguistics professionals can refer back to this database for data on languages and phonetics.

ER Diagram



A minor update has been made to the ER diagram. The entity Consonants now includes a new BOOLEAN attribute, Coronal, which is determined by the attribute Place. This addition allows us to integrate additional functional dependencies into the ER diagram while maintaining logical consistency. Similarly, we have added an attribute called UnicodeBlock to the Phoneme entity. The attribute Unicode determines UnicodeBlock to have more meaningful FDs.

Schema

Legend: Primary Key, **Foreign key**

1. Language(Name: VARCHAR, Status: VARCHAR, **FamilyName**: VARCHAR)
2. Family(Name: VARCHAR, Origin: VARCHAR)
3. WritingSystem(Name: VARCHAR, Type: VARCHAR, Age: INTEGER)
4. Uses(**WSName**: VARCHAR, **LanguageName**: VARCHAR)
5. Speaker(ID: VARCHAR, Name: VARCHAR)
6. Country(Name: VARCHAR, Continent: VARCHAR, Population: INTEGER)

7. SpokenBy(**SpeakerID**: VARCHAR, **DialectName**: VARCHAR, **LanguageName**: VARCHAR)
8. SpokenIn(**CountryName**: VARCHAR, **DialectName**: VARCHAR, **LanguageName**: VARCHAR)
9. Defines(**WordID**: INTEGER, **DialectName**: VARCHAR, **LanguageName**: VARCHAR)
10. Contains(**IPANumber**: INTEGER, **WordID:** INTEGER)
11. Phoneme(IPANumber: INTEGER, Unicode: INTEGER, UnicodeBlock: VARCHAR)
12. Vowel(IPANumber: INTEGER, Height: VARCHAR, Backness: VARCHAR, Rounded: BOOLEAN)
13. Consonant(IPANumber: INTEGER, Voiced: BOOLEAN, Place: VARCHAR, Manner: VARCHAR, Coronal: BOOLEAN)
14. Dialect(Name: VARCHAR, **LanguageName**: VARCHAR, Population: INTEGER)
15. IsFrom(**SpeakerID**: INTEGER, **CountryName**: VARCHAR)
16. Word(ID: VARCHAR, WrittenForm: NVARCHAR, Meaning: VARCHAR)[1]

Attributes in red are introduced to add non-candidate key FDs for q5

Candidate keys:

1. Language: {Name}
2. Family: {Name}
3. WritingSystem: {Name}
4. Uses: {WSName, LanguageName}
5. Speaker: {ID}
6. Country: {Name}
7. SpokenBy: {SpeakerID, DialectName, LanguageName}
8. SpokenIn: {CountryName, DialectName, LanguageName}
9. Defines: {WordID, DialectName, LanguageName}
10. Contains: {IPANumber, WordID}
11. Phoneme: {IPANumber}
12. Vowel: {IPANumber}
13. Consonant: {IPANumber}
14. Dialect: {Name, LanguageName}

[1] NVARCHAR is used instead of VARCHAR since this will include unicode characters as shown below

15. IsFrom: {SpeakerID, CountryName}
16. Word: {ID}

Functional Dependencies

1. Unicode -> UnicodeBlock
2. WordID -> WrittenForm, Meaning
3. LanguageName -> Status
4. FamilyName -> Origin
5. WSName -> Age, Type
6. DialectName, LanguageName -> Population
7. SpeakerID -> SpeakerName
8. CountryName -> Continent, Population
9. IPANumber -> Unicode, Height, Backness, Roundness, Voiced, Place, Manner
10. Place -> Coronal

Normalization

1. Let Unicode -> UnicodeBlock = U -> UB. Since U is not a candidate key, this violates BCNF. We decompose into two relations as follows:

R1(IPANumber, Unicode), R2(Unicode, UnicodeBlock)

Candidate keys:
- In R1: IPANumber
- In R2: Unicode

2. Let Place -> Coronal = P -> C. Since P is not a candidate key, this violates BCNF and we decompose into two relations as follows:

R3(IPANumber, Unicode, Voiced, Place, Manner), R4(Place, Coronal)

Candidate keys:
- In R3: IPANumber
- In R2: Place

This results in two new tables: Unicode(UnicodeID, UnicodeBlock), PlaceInfo(Place, Coronal)

University of British Columbia, Vancouver
Department of Computer Science

SQL DDL Statements

1. Phoneme(IPANumber: INTEGER, Unicode: INTEGER, UnicodeBlock: VARCHAR)

Gets decomposed into Phoeneme(IPANumber, Unicode), Unicode(Unicode, UnicodeBlock)

Phoneme:

```
CREATE TABLE Phoneme (
IPANumber: INTEGER,
Unicode: INTEGER,
PRIMARY KEY (IPANumber),
FOREIGN KEY (IPANumber),
UNIQUE (Unicode)
);
```

Unicode:

```
CREATE TABLE Unicode (
Unicode: INTEGER,
UnicodeBlock: VARCHAR,
PRIMARY KEY (Unicode)
);
```

2. Consonants(IPANumber: INTEGER, Voiced: VARCHAR, Place: VARCHAR, Manner: VARCHAR, Coronal: BOOLEAN)

Gets decomposed into Consonants(IPANumber, Unicode, Voiced, Place, Manner), PlaceInfo(Place, Coronal)

Consonants:

```
CREATE TABLE Consonants (
IPANumber: INTEGER,
Unicode: INTEGER,
Voiced: VARCHAR,
Place: VARCHAR,
```

```
Manner: VARCHAR,
PRIMARY KEY (IPANumber),
FOREIGN KEY (IPANumber) REFERENCES Phoneme ON DELETE CASCADE ON UPDATE
CASCADE,
UNIQUE (Unicode)
);
```

The ON DELETE CASCADE ON UPDATE CASCADE statement is required in the Consonants table to ensure referential integrity. So when a record is deleted or updated in Phoneme, it has to be deleted or updated from Consonants because the ISA relation is total and we don't want to have children in this case that do not have any parent in Phoneme to refer to.

PlaceInfo:

```
CREATE TABLE PlaceInfo (
Place: VARCHAR,
Coronal: BOOLEAN,
PRIMARY KEY (Place)
);
```

INSERT Statements

INSERT INTO Language (Name, Status, FamilyName)
VALUES (‘English’, ‘International’, ’Indo-European’),
(‘French’, ‘International’, ’Indo-European’),
(‘Arabic’, ‘International’, ’Afro-Asiatic’),
(‘Swahili’, ‘National’, ’Niger-Congo’),
(‘Halkomelem’, ‘Moribund’, ’Salish’),
(‘Korean’, ‘National’, ‘Koreanic’);

INSERT INTO Family (Name, Origin)
VALUES (‘Indo-European’, ‘Pontic–Caspian steppe’),
(‘Afro-Asiatic’, ‘East Africa’),
(‘Niger-Congo’, ‘Savanna belt of West Africa’),
(‘Salish’, ‘Pacific Northwest’),
(‘Koreanic’, ‘Korean Peninsula’);

```
INSERT INTO WritingSystem (Name, Type, Age)
VALUES (‘Latin’, ‘Alphabet’, 2700),
        (‘Arabic’, ‘Abjad’, 1800),
        (‘North American Phonetic Alphabet’, ‘Alphabet’, 160),
        (‘Hangul’, ‘Featural Alphabet’, 582),
        (‘Chinese’, ‘Logographic’, 3000);

INSERT INTO Uses (Name, Type, Age)
VALUES (‘English’, ‘Latin’),
        (‘Swahili’, ‘Latin’),
        (‘French’, ‘Latin’),
        (‘Swahili’, ‘Arabic’),
        (‘Arabic’, ‘Arabic’),
        (‘Halkomelem’, ‘North American Phonetic Alphabet’),
        (‘Korean’, ‘Hangul’),
        (‘Korean’, ‘Chinese’);

INSERT INTO Speaker (ID, Name)
VALUES (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’);

INSERT INTO Country (Name, Continent, Population)
VALUES ('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);

INSERT INTO Dialect (Name, LanguageName, Population)
VALUES (‘Canadian’, ‘French’, 7700000),
        (‘Coastal’, ‘Swahili’, 15000000),
        (‘Gulf’, ‘Arabic’, 36000000),
        (‘Jeju’, ‘Korean’, 5000),
        (‘Canadian’, ‘English’, 30000000),
```

('British', 'English', 60000000),
('Downriver', 'Halkomelem', 4);

```
INSERT INTO SpokenBy (SpeakerID, DialectName, LanguageName)
VALUES (1, 'Canadian', 'English'),
        (2, 'Canadian', 'French'),
        (3, 'Gulf', 'Arabic'),
        (4, 'Jeju', 'Korean'),
        (5, 'Coastal', 'Swahili'),
        (6, 'Canadian', 'English'),
        (7, 'British', 'English'),
        (8, 'Gulf', 'Arabic'),
        (8, 'British', 'English'),
        (1, 'British', 'English'),
        (2, 'British', 'English'),
        (6, 'Downriver', 'Halkomelem'),
        (7, 'Coastal', 'Swahili');

INSERT INTO SpokenIn (CountryName, DialectName, LanguageName)
VALUES ('Canada', 'Canadian', 'English'),
        ('Canada', 'Canadian', 'French'),
        ('France', 'Canadian', 'French'),
        ('Kenya', 'Coastal', 'Swahili'),
        ('Democratic Republic of the Congo', 'Coastal', 'Swahili'),
        ('South Korea', 'Jeju', 'Korean'),
        ('Saudi Arabia', 'Gulf', 'Arabic'),
        ('Canada', 'Downriver', 'Halkomelem'),
        ('United Kingdom', 'British', 'English');

INSERT INTO Word (ID, WrittenForm, Meaning)
VALUES (1, 'bonjour', 'hello'),
        (2, 'hello', 'greeting'),
        (3, 'salama', 'peace'),
        (4, '안녕', 'hello'),
        (5, 'مرحبا', 'hello'),
        (6, 'həṅq̓', 'hello'),
        (7, 'merci', 'thank you'),
        (8, 'habari', 'news'),
        (9, '네', 'yes'),
        (10, 'شكرًا', 'thank you');

INSERT INTO Defines (WordID, DialectName, LanguageName)
```

```
VALUES (1, 'Canadian', 'French'),
        (2, 'Canadian', 'English'),
        (3, 'Coastal', 'Swahili'),
        (4, 'Jeju', 'Korean'),
        (5, 'Gulf', 'Arabic'),
        (6, 'Downriver', 'Halkomelem'),
        (7, 'Canadian', 'French'),
        (8, 'Coastal', 'Swahili'),
        (9, 'Jeju', 'Korean'),
        (10, 'Gulf', 'Arabic');

INSERT INTO Phoneme (IPANumber, Unicode)
VALUES (101, 98),
        (102, 111),
        (103, 110),
        (104, 660),
        (105, 629),
        (106, 109),
        (107, 641),
        (108, 616),
        (109, 652),
        (110, 712);

INSERT INTO Vowel (IPANumber, Height, Backness, Rounded)
VALUES (102, 'close-mid', 'back', TRUE),
        (105, 'close-mid', 'central', TRUE),
        (108, 'close', 'central', FALSE),
        (109, 'open-mid', 'back', TRUE),
        (110, 'high-mid', 'front', FALSE);

INSERT INTO Consonant (IPANumber, Voiced, Place, Manner)
VALUES (101, TRUE, 'bilabial', 'plosive'),
        (103, TRUE, 'alveolar', 'nasal'),
        (104, FALSE, 'glottal', 'plosive'),
        (106, TRUE, 'bilabial', 'nasal'),
        (107, TRUE, 'uvular', 'trill');

INSERT INTO Contains (IPANumber, WordID)
VALUES (101, 1),
        (102, 1),
        (103, 1),
        (106, 3),
```

```
        (107, 5),
        (104, 6),
        (108, 8),
        (109, 9),
        (110, 10);

INSERT INTO Unicode (UnicodeID, UnicodeBlock)
VALUES (98, 'Latin'),
        (111, 'Latin'),
        (110, 'Latin'),
        (660, 'IPA Extensions'),
        (629, 'IPA Extensions'),
        (109, 'Latin'),
        (641, 'IPA Extensions'),
        (616, 'IPA Extensions'),
        (652, 'IPA Extensions'),
        (712, 'IPA Extensions');

INSERT INTO PlaceInfo (Place, Coronal)
VALUES ('bilabial', FALSE),
        ('alveolar', TRUE),
        ('glottal', FALSE),
        ('uvular', FALSE),
        ('palatal', TRUE);

INSERT INTO IsFrom (SpeakerID, CountryName)
VALUES (1, 'Canada'),
        (2, 'France'),
        (3, 'Saudi Arabia'),
        (3, 'Egypt'),
        (4, 'South Korea'),
        (5, 'Kenya'),
        (5, 'Democratic Republic of the Congo'),
        (6, 'Canada'),
        (7, 'France'),
        (8, 'Saudi Arabia'),
        (8, 'Egypt');
```

University of British Columbia, Vancouver

Department of Computer Science

AI was used to generate sample data to insert into the tables: so example values like (1, ‘bonjour’, ‘hello’). This was done via ChatGPT using a prompt similar to “Give me N words in language X, and its translation in english”