French Language Database
Databases IST-4-DBM1

Mark J. Thompson
Zahran S. Ghazy
Department of IST

Supervisor: Riccardo Tommasini

Contents

1 Introduction

The objective of this project was to model, design, and implement a database in order to put the theory learned throughout the course into practise. While trying to come up with the domain for the database, the conclusion was made that a database that supports learning and practising the French language would be an intriguing and befitting goal. Through this project, we aimed to showcase our understanding of database concepts such as entity-relationship (ER) modelling, SQL queries, and data normalization.

This report documents the various stages of the project, from conceptual design to implementation. It also reflects the challenges encountered, solutions applied, and future improvements envisioned.

2 Data base design

2.1 Conceptional design

Learning any language requires a lot of practice and the concept for this project is to aid the user to do that. By storing grammatical and vocabulary data. The end goal is to create an application that wil use data from this database in order to quiz users and test their knowledge.

In order to be practical, the following are the natural language queries that the date base should be able to handle: In order to be practical, the databse will be able to perform the following tasks:

- Select a random verb and tense from the database, use this data to provide conjugations for all subjects in french and english.
- Select a random subject, verb, and tense, return the grammatically correct french and english sentences.
- Select a random noun, determinant, and adjective in english, make them agree grammatically in french.

The webapp will use these queries in order to randomly generate sentences in english that the user must translate.

2.2 ER-diagram

In order to create a database based off of the conceptional design, the ER diagram below is used to represent entities, relationships, and cardinalities clearly.

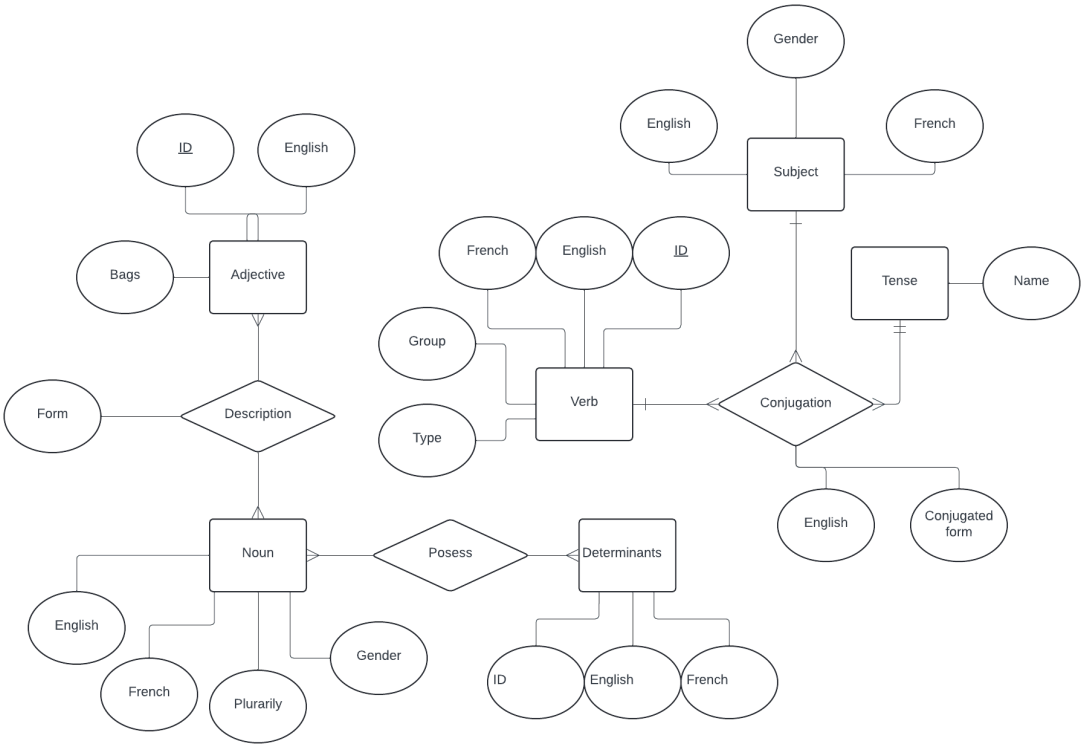

Figure 1: Entity-Relationship Diagram of the French Practice Database.

In this case, the entities are the different word classes, like nouns, verbs, adjectives etc. and they all have attributes that are explained below.

Entities

(-attributes):

- **Verb:** Represents a verb in its infinitive form (e.g., *aller*, *être*).
  - verb_id: Unique numeric identifier for each verb.
  - english_inf: Infinitive form of the verb in english
  - french_inf: Infinitive form of the verb in french
  - group: Whether the verb belongs to the first(-er), second(-ir) or irregular group.
  - type: The type of verb, such as transitive, intransitive, modal, or state.
- **subject:** Represents the subject of a sentence in the French language (e.g., *je*, *il*, *nous*).
  - subject_id: Unique numeric identifier for each subject
  - eng_subj: English translation of subject
  - fr_subj: French translation of subject
  - gender: The gender associated with the subject

- **Tense:** Represents the tense of a verb, such as present, past, or future.
  - tense_id: Unique numeric identifier for each tense
  - tense_name: English name of tense
- **conjugation**
  - verb_id: Foreign key to corresponding verb of conjugation
  - subject_id: Foreign key to corresponding subject of conjugation
  - tense_id: Foreign key to corresponding tense of conjugation
  - eng_conj: English translation
  - fr_conj: French translation
- **Noun:** Represents a noun (e.g., *chaise*, *livre*).
  - noun_id: Unique numeric identifier for each noun
  - noun_fr: French translation
  - noun_english: English translation
  - gender: Whether the french noun is masculine or feminine
  - plural: Binary attribute specifying whether the noun is plural or not
- **English Determinants:** Represents determiners such as articles or possessive adjectives (e.g., *le*, *ma*).
  - det_id: Unique numeric identifier for each determinant
  - det_en: English translation of determinant
- **French determinants**
  - det_id: Foreign key to determinant table
  - gender: Which gender the french determinant corresponds to
  - plural: Which plurality the determinant corresponds to
  - det_fr: The french translation of the adverb
- **English Adjective:** Represents an adjective used to modify a noun (e.g., *grand*, *petit*).
  - adj_id: Unique numeric identifier for each adjective
  - adj_en: English translation of adjective
  - bags: Acronym for beauty, age, goodness and size, indicating if a verb should come before or after a noun.
- French Adjective
  - adj_id: Foreign key to adjective table
  - gender: What gender the adjective form corresponds to
  - plural: What plurality the adjective form corresponds to
  - adj_fr: French translation of adjective

Relations

In this ER diagram, the relations are the grammatical rules that binds together the entities.

- **Conjugation:** Represents the conjugated form of a verb, considering both tense and subject (e.g., *je vais*, *nous allons*).
- **Possess:** Represents the relationship where determinants possess nouns (e.g., *le livre*, *ma chaise*).
- **Agreement:** Represents the agreement between adjectives and nouns in terms of gender and number.

Cardinalities

Cardinalities describe the number of instances of one entity that can relate to instances of another entity through a relationship. The cardinalities for the relationships in the diagram are as follows:

- **Conjugation (Subject ↔ Verb ↔ Tense)**
  - **Subject ↔ Conjugation:** Each subject can have multiple conjugations. For example, *je* can be used in conjugations of every verb and tense , *je suis*, *je mange*, etcetera. On the contrary, one specific conjugation of a verb in a certain tense can only belong to one subject. For example, given the verb "*manger*" in the *present* tense, the conjugation "*mangeons*" can only belong to the subject *nous*. Cardinality: Subject (1) ↔ (M) Conjugation.
  - **Verb ↔ Conjugation:** Each verb can have multiple conjugations. For example, *aller* can be conjugated in several forms: *vais*, *allais*, *irai*. A specific conjugation however, can only belong to one verb. Cardinality: Verb (1) ↔ (M) Conjugation.
  - **Tense ↔ Conjugation:** Each tense can have multiple conjugations, as verbs are conjugated in different tenses (e.g., *je vais* in the present tense, *je suis allé* in the past tense), and as each verb is conjugated differently in specific tenses. One specific conjugation however, belongs to one specific tense. Cardinality: Tense (1) ↔ (M) Conjugation.
- **Agreement (Adjective ↔ Noun)**
  - **Adjective ↔ Noun:** Each adjective can agree with multiple nouns, depending on their gender and number. For instance, *grande* can describe both *pomme* and *table*. Conversely, each noun can agree with different adjectives. Cardinality: Adjective (M) ↔ (M) Noun.
- **Possess (Noun ↔ Determinants)**
  - **Noun ↔ Possess:** Each noun can be "possessed" by multiple determiners. For example, *chaise* can be possessed by *ma* (my), *ta* (your), *la* (the). Conversely, each determinant can posess multiple different nouns. Cardinality: Noun (M) ↔ (M) Determinant.

With this ER-diagram, the database can be used to quizz the user in accordance with the queries described in the conceptional design.

3 Logical Schema

- verb(verb_id(pk), eng_inf, fr_inf, 'group')
- subject(subject_id(pk), eng_subj, fr_subj, gender)
- tense(tense_id(pk), tense_name)
- conjugation(verb_id(fk), subject_id(fk), tense_id, eng_conj, fr_conj)
- noun(noun_id(pk), noun_fr(pk), noun_english, gender, plural)

- determinant(det_id(pk), det_en)
- det_agreement(det_id(fk), gender, plural, det_fr)
- adjectives(adj_id(pk), adj_en)
- adjective(adj_id(pk), adj_en)
- adjective_agreement(adj_id(fk), gender, plural, adj_fr)
- adverb(adv_id(fk), adv_fr, adv_en, type)

4 SQL

4.1 SQL Schema

The database schema was written in SQL. Below is a sample of the schema definition:

```
CREATE TABLE verb (
    verb_id INT PRIMARY KEY AUTO_INCREMENT,
    eng_inf VARCHAR(100) NOT NULL,
    fr_inf VARCHAR(100) NOT NULL,
    `group` TINYINT NOT NULL,
    type VARCHAR(50) NOT NULL
);

CREATE TABLE subj (
    subj_id INT PRIMARY KEY AUTO_INCREMENT,
    eng_subj VARCHAR(50) NOT NULL,
    fr_subj VARCHAR(50) NOT NULL,
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F', 'N'))
);

CREATE TABLE tense (
    tense_id INT PRIMARY KEY AUTO_INCREMENT,
    tense_name VARCHAR(100) NOT NULL
);

CREATE TABLE conj (
    verb_id INT NOT NULL,
    subj_id INT NOT NULL,
    tense_id INT NOT NULL,
    eng_conj VARCHAR(100) NOT NULL,
    fr_conj VARCHAR(100) NOT NULL,
    PRIMARY KEY (verb_id, subj_id, tense_id),
    FOREIGN KEY (verb_id) REFERENCES verb(verb_id),
    FOREIGN KEY (subj_id) REFERENCES subj(subj_id),
    FOREIGN KEY (tense_id) REFERENCES tense(tense_id)
);

CREATE TABLE noun (
    noun_id INT PRIMARY KEY AUTO_INCREMENT,
    noun_fr VARCHAR(100) NOT NULL,
    noun_english VARCHAR(100) NOT NULL,
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F', 'N')),
    plural BOOLEAN NOT NULL
```

```
);

CREATE TABLE determinant (
    det_id INT PRIMARY KEY AUTO_INCREMENT,
    det_en VARCHAR(50) NOT NULL
);

CREATE TABLE det_agr (
    det_id INT NOT NULL,
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F', 'N')),
    plural BOOLEAN NOT NULL,
    det_fr VARCHAR(50) NOT NULL,
    PRIMARY KEY (det_id, gender, plural),
    FOREIGN KEY (det_id) REFERENCES determinant(det_id)
);

CREATE TABLE adj (
    adj_id INT PRIMARY KEY AUTO_INCREMENT,
    adj_en VARCHAR(100) NOT NULL,
    bags TINYINT(1) NOT NULL
);

CREATE TABLE adj_agr (
    adj_id INT NOT NULL,
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F', 'N')),
    plural BOOLEAN NOT NULL,
    adj_fr VARCHAR(100) NOT NULL,
    PRIMARY KEY (adj_id, gender, plural),
    FOREIGN KEY (adj_id) REFERENCES adj(adj_id)
);

CREATE TABLE adv (
    adv_id INT PRIMARY KEY AUTO_INCREMENT,
    adv_fr VARCHAR(100) NOT NULL,
    adv_en VARCHAR(100) NOT NULL,
    `type` ENUM('manner', 'time', 'place', 'degree', 'frequency', 'negation') NOT NULL
);
```

4.2 Queries

To demonstrate functionality, several queries were implemented:

- Retrieve all conjugations of a random verb in a random tense, this view is used for the conjugation part of the web application.

```
SELECT c.fr_conj
FROM
    (SELECT verb_id, fr_inf FROM verb ORDER BY RAND() LIMIT 1) v
CROSS JOIN
    (SELECT tense_id, tense_name FROM tense ORDER BY RAND() LIMIT 1) t
JOIN conj c ON c.verb_id = v.verb_id AND c.tense_id = t.tense_id
        ORDER BY c.subj_id;
```

$$VT \leftarrow Verb \times Tense (VT \; is \; a \; randomly \; selected \; row \; from \; cross \; product)$$

$$C \leftarrow \sigma_{verb\_id=verb\_id \wedge tense\_id=tense\_id}(\text{conj})$$

$$Result \leftarrow \pi_{c.fr\_conj}(VT \bowtie C)$$

- Retrieve a random determinant, noun and adjective which all agree with eachother grammatically. We also retrieve the binary "bags" value from adjective so that the web application will know if an adjective should be placed before or after a noun in french. This is used for the noun part of the web application.

```
SELECT
    n.noun_fr, n.noun_english,
    d.det_fr, dt.det_en,
    a.adj_fr, aj.adj_en,
    aj.bags
FROM (
    SELECT
        noun_english, noun_fr,
        gender, plural
    FROM noun
    ORDER BY RAND()
    LIMIT 1) AS n
JOIN det_agr d ON d.gender = n.gender AND d.plural = n.plural
JOIN determinant dt ON dt.det_id = d.det_id
JOIN adj_agr a ON a.gender = n.gender AND a.plural = n.plural
JOIN adj aj ON aj.adj_id = a.adj_id
ORDER BY RAND()
LIMIT 1;
```

$$N \leftarrow \pi_{noun\_english, noun\_fr, gender, plural}(\text{Noun})$$

$$D \leftarrow Det\_Eng \bowtie_{det_i d = det_i d} Det_F r$$

$$A \leftarrow Adj\_Eng \bowtie_{adj_i d = adj_i d} Adj_F r$$

$$ND \leftarrow (N \bowtie_{gender=gender \wedge plural=plural} DA$$

$$F \leftarrow ND \bowtie_{gender=gender \wedge plural=plural} A$$

$$\pi_{\text{noun\_fr, noun\_english, det\_fr, det\_en, adj\_fr, adj\_en, bags}} F \; (Select \; random \; row)$$

- The following queries are used for the sentence part of the web application. First we must choose a random subject and verb in a random tense:

```
SELECT
    s.eng_sub, s.fr_subj,
    c.en_conj, c.fr_conj,
    v.type, s.gender, s.plurality
FROM
    (SELECT subj_id, eng_sub, fr_sub, gender, plurality
     FROM subj
     ORDER BY RAND()
     LIMIT 1) AS s
CROSS JOIN
    (SELECT verb_id, type
```

```
    FROM verb
    ORDER BY RAND()
    LIMIT 1) AS v
JOIN
    conj AS c ON c.subj_id = s.subj_id AND c.verb_id = v.verb_id
ORDER BY RAND()
LIMIT 1;
```

$$\text{SV} \leftarrow \text{Subject} \times \text{verb}$$

$$\text{SVC} \leftarrow SV \bowtie_{subj_id=subj_id,verb_id=verb_id} \text{Conj}$$

$$\pi_{eng_subj,fr_subj,en_conj,fr_conj,type,gender,plural} SVC \ (Select\ random\ row)$$

Then we use the verb type to decide if we should attach a noun to the end if it is transitive, or an adjective if it is a state noun. If it is transitive, we reuse the statement which retrieves a random determinant, noun and adjective.

– If it is state verb like "être" we use the following to select a suitable adjective to add to the end of the sentence:

```
SELECT
  adj_id, adj_fr
FROM
  adj_agr a
WHERE a.gender=<subj_gender> and a.plural=<subj_plurality>
ORDER BY rand()
LIMIT 1
```

$$\pi_{adj_id,adj_fr} \sigma_{gender=X \wedge plural=Y} \text{French Adjective}$$

Where subj_gender and subj_plurality are passed to the query with a python function.

– If it is a modal verb like "vouloir" we use the following query to select a random verb in the infinitive to add to the sentence:

```
SELECT
    eng_inf, fr_inf, type
FROM
    verb v
WHERE
    v.type!='modal'
ORDER BY rand()
LIMIT 1
```

$$\pi_{eng\_ing,fr\_inf} \sigma_{type=modal} \text{Verb} \ \ (Select\ random\ row)$$

And then we attach appropriate words after the new infinitive verb using the same queries being described now, except our program does not make sentences with 2 modal verbs in a row.

– If it is an intransitive verb we attach an adverb to the end using the following query:

```
SELECT
    adv_en, adv_fr
FROM adv a
WHERE a.type='manner'
ORDER BY rand()
LIMIT1
```

$$\pi_{adv_en,adv_fr}\sigma_{type=manner}\text{adv}$$

  - If it is a transitive verb, meaning it must act on an object, we reuse the function above that returns a determinant, noun, and adjective

5 Challenges and Solutions

5.1 Challenges

When designing the database we were faced with a variety of problems such as:

- Problem: Data redundancy in adjectives and determinants due to most french adjectives and determinants having 4 forms based on gender and plurality.
  Solution: Normalise the tables for adjectives and determinants, so one contains an id and the English translation, then the other contains the id, gender, plurality, and the corresponding French translation.
- Problem: With binary relationship the conjugation table had far too many columns due to each subject requiring a different conjugation.
  Solution: Create a new table for subjects and make conjugation a ternary relationship.
- Problem: Some sentences being output by the web application were nonsensical because the verbs or adverbs didn't work with each other.
  Solution: Create the attribute "type" for verbs, which helps the web application decide what types of words should follow them. Also creating the attribute "type" for adverbs so that the application can choose adverbs in a more sensible way depending on context.

6 Conclusion and Future Work

Conclusion: Finally we managed to create a functional web application using data from the book "Le Petit Prince" to create 3 types of exercises for users to practice. These exercises include:

- Presenting the user with a verb and tense and seeing if they can conjugate it in French correctly.
- Presenting the user with a determinant, noun, and adjective in English and seeing if they can translate it into French correctly.
- Presenting the user with a randomly generated sentence and seeing if they can translate it into French correctly.

Future Work:

- Add support for synonyms so the user will not have their score negatively affected in situations like using the adverb "serieusement" instead of "gravement" when asked to translate the word seriously.
- Add prepositions in order to connect words and allow for sentences with more depth.
- Categorise data by chapter they appear in in the book "Le Petit Prince" so the user can work through the book in order, and know how far they should be able to read comfortably at any point.
- Create user profiles to store progress.