

Documentation du système Verba

Ce document a pour objectif de fournir un mode d'emploi du système informatique *Verba*, tel qu'il est actuellement utilisé dans le cadre du projet ANR APPI sous le nom *VerbaPicardia* (<https://anr-appi.univ-lille.fr>). Le système Verba est commun au projet APPI et au projet *VerbaAlpina* (<https://www.verba-alpina.gwi.uni-muenchen.de>), qui l'a conçu et mis à la disposition de la communauté scientifique. Une documentation du système tel qu'il est utilisé dans le cadre du projet VerbaAlpina est disponible à l'adresse https://www.verba-alpina.gwi.uni-muenchen.de/?page_id=493&db=192.

Le présent document est destiné aux rédacteurs qui utilisent le système Verba pour y encoder des données et les publier en ligne. Dans les pages suivantes sont décrits l'architecture générale du système (A), le frontend du site Web (B), la base de données (C) et le backend du site Web (D). A l'intention des utilisateurs non initiés au langage SQL, quelques requêtes utiles pour l'exploitation de la base de données sont fournies ensuite (E). Les figures qui illustrent ce document sont regroupées dans une annexe (F).

A. Architecture générale du système

Le système Verba est divisé en trois composants : un site Web WordPress, un ensemble de bases de données MySQL et un espace de stockage de fichiers. Les trois composants sont hébergés sur un même serveur (à l'université de Lille pour le projet VerbaPicardia).

- Le site Web est lui-même divisé en deux parties : le *frontend*, composé des pages accessibles publiquement, et le *backend*, accessible uniquement aux personnes autorisées. C'est dans le backend que s'effectuent les tâches d'encodage des données.
- La base de données Verba contient les données qui ont été encodées dans le backend, ainsi que toutes les données nécessaires pour l'encodage et pour l'affichage de ces données : liste des cartes, des concepts, des points d'enquête, des caractères avec leur codage, des transcriptions en alphabet phonétique international etc.
- L'espace de stockage de fichiers est utilisé pour les scans des cartes originales des atlas.

B. Frontend du site Web

L'espace public du site Web est spécifique à chaque projet, mais il utilise un plugin Verba dans une page intitulée *plan interactif* (et que nous désignons aussi par *carte interactive*). Ce plugin permet l'affichage des données encodées sous une forme cartographique. La page se divise en deux zones : la carte proprement dite et l'interface d'interrogation.

L'interface d'interrogation permet de choisir les critères de regroupement des données (conceptuel, étymologique ou morpholexical) et permet également d'afficher des données non langagières, par exemple les points d'enquête de chaque atlas (voir illustration sous F.1). Lors de l'affichage d'une carte, une légende décrit les regroupements et permet de (dé)sélectionner certains groupes.

Le système Verba est versionné. Il est possible de choisir le numéro de version en ouvrant le menu déroulant situé en haut à droite de la carte interactive. La version « xxx » désigne la version de travail (en cours d'encodage). Les versions numérotées correspondent à un état figé de la base à un moment donné. Par exemple, la version 2 de VerbaPicardia correspond à l'état de la base au 31 décembre 2019.

C. Base de données

C.1. Contenu initial de la base de données Verba

La base de données se compose de diverses tables nécessaires pour l'encodage et l'affichage des données. Ces tables sont citées dans la section D ci-dessous au fur et à mesure de l'explication du fonctionnement du backend. Un schéma général de la base de données est accessible sur le site de VerbaAlpina, à l'adresse suivante: https://www.verba-alpina.gwi.uni-muenchen.de/?page_id=493&db=192&tag=3#16. La base de données du projet VerbaPicardia est accessible via une interface PHPMyAdmin à l'adresse <https://anr-appi.univ-lille.fr/db/>.

- Deux tables doivent être remplies dès le début d'un nouveau projet, préalablement à tout encodage : la table *stimuli*, contenant la liste des cartes des atlas (voir F.2), et la table *informanten*, contenant la liste des points d'enquête de chaque atlas avec leur géoréférencement.
- La table *konzepte*, contenant la liste des concepts, et la table *konzepte_kategorien*, contenant les catégories conceptuelles, peuvent être remplies préalablement à tout encodage. Elle peuvent également être complétées via le backend au moment de l'encodage (voir F.3 et D.1.1).
- La table *codepage_original* contient l'ensemble des données nécessaires pour l'encodage des caractères et pour leur affichage sur la carte interactive. Chacun des caractères y est associé à un codage en *betacode* (voir D.1.2), qui est le système de codage utilisé par Verba (voir F.4). Cette table peut être agrandie de nouveaux caractères au fur et à mesure des besoins qui apparaissent dans l'encodage des données. Le contenu de cette table étant commun aux projets VerbaPicardia et VerbaAlpina, il est vivement conseillé de synchroniser sa mise à jour avec le projet VerbaAlpina.
- La table *aeusserungen* contient les données langagières proprement dites, à savoir les formes qui sont affichées sur la carte pour chaque point d'enquête. Cette table reçoit les données encodées via le backend (voir F.5 et D.1.1).

Remarque : si des données sont déjà disponibles dans un fichier structuré (xls, csv ou xml par exemple), il est évidemment possible, via un script SQL à créer en fonction du format de départ des données, de les verser directement dans la table *aeusserungen* sans devoir les encoder à nouveau dans le backend. Les données doivent préalablement être transcodées en betacode, à l'aide d'un script spécifique à créer en fonction du format de départ des données.

C.2. Mise à jour de la base de données

L'opération d'encodage des données via le backend (voir ci-dessous D) nécessite parfois de compiler les données. Dans VerbaPicardia, il est possible d'effectuer cette opération directement depuis le backend, avec l'outil *Build Tables*, accessible via le menu *Verba Tools* : un clic sur le bouton *Build Tables* met automatiquement à jour les tables de la base de données. Cette opération peut aussi s'effectuer en entrant dans la base de données et en cliquant sur l'action « Exécuter » de la procédure *build tables* (voir F.6).

C.3. Export du contenu de la base de données

La base de données Verba se consulte comme toute base de données relationnelle. Son contenu peut donc être interrogé à l'aide de requêtes SQL et il peut être exporté sous d'autres formats (ods, xls, csv etc.). A la fin de ce document sont fournies quelques requêtes toutes faites qui peuvent être utiles durant le processus d'encodage (voir E).

D. Backend du site Web

Cette section décrit les opérations successives qui sont effectuées lors de l’insertion des données, depuis leur encodage jusqu’à leur mise en ligne cartographiée en alphabet phonétique international. Toutes ces opérations s’effectuent dans le backend du site Web, qui est accessible aux personnes autorisées après identification de l’utilisateur (voir F.7).

En-dessous des outils dédiés à la rédaction des pages du site WordPress et d’outils spécifiques au projet VerbaPicardia, le menu du backend présente deux entrées liées au système Verba et repérables par l’icône ✖ : *Verba* et *Verba Tools*. Sous ces deux boutons se trouvent tous les outils décrits ci-dessous.

Deux opérations seulement sont à effectuer pour permettre la mise en ligne des données sous la forme de cartes interrogeables par concept : l’encodage proprement dit (D.1) et la tokenisation (D.2). Il est ensuite possible d’enrichir les données en y ajoutant une typisation (D.3), qui permet de représenter et d’interroger les données selon leur type morpholexical et leur étymon. Le système Verba offre également la possibilité d’afficher les données en alphabet phonétique international (D.4). Enfin, le menu du backend présente quelques outils permettant d’avoir une vue d’ensemble, par concept et par carte, de l’avancement de l’encodage (D.5).

D.1. Encodage des données

D.1.1. Transcription tool

L’outil d’encodage, appelé *Transcription tool*, est accessible via le menu *Verba* du backend. Il contient trois parties : une zone affichant les scans des atlas à encoder (scans stockés sur le serveur du projet, dans l’espace de stockage de fichiers), une zone affichant les règles de transcription et une zone destinée à l’encodage proprement dit (voir F.8).

L’encodage s’effectue carte par carte. Il faut choisir l’atlas que l’on veut encoder, puis la carte, dans un menu déroulant qui affiche le contenu de la table stimuli. Les cartes qui sont en cours d’encodage par d’autres rédacteurs ne peuvent être sélectionnées. Une fois la carte choisie, elle s’affiche dans la zone d’affichage des scans.

Remarque. Cet affichage est possible à condition que le nom de fichier des scans stockés sur le serveur corresponde au nom d’atlas et au numéro de carte stockés dans la table stimuli. Par exemple, la carte ALF *abeille*, identifiée dans la table *stimuli* avec la source *ALF* et le numéro de carte *1*, s’affiche car le scan correspondant est intitulé *ALF#1.pdf* sur le serveur. Si la correspondance n’est pas assurée (parce que le scan n’est pas disponible sur le serveur, ou parce que le nom de fichier du scan ne correspond pas), le menu déroulant affiche le nom de la carte en rouge au lieu de vert (voir F.9).

Dans chaque carte, l’encodage s’effectue point d’enquête par point d’enquête. Le système propose des points d’enquête de façon aléatoire, mais il est possible également de choisir le point d’enquête que l’on veut encoder en insérant manuellement le numéro de ce point dans la zone marquée *Informant number(s)* (voir F.8). Pour revenir au mode aléatoire, il suffit de laisser cette zone de texte vide et d’appuyer sur *Enter*.

Pour chaque point d’enquête, il est possible d’encoder une ou plusieurs formes. L’ajout de lignes supplémentaires s’effectue via le bouton *Add row*. Si le point d’enquête ne contient aucune forme à encoder, il faut appuyer sur le bouton *vacat*.

Pour chaque forme à encoder, il faut indiquer s’il s’agit d’une attestation (par défaut), d’un type phonétique ou d’un type morphologique. Dans le cadre de VerbaPicardia, il s’agit toujours d’attestations. Chaque forme encodée doit également être associée à un concept, accessible via un

menu déroulant qui correspond au contenu de la table *konzepte* (voir F.10). Il est possible d'ajouter un nouveau concept, qui sera alors automatiquement inséré dans la table *konzepte* de la base de données (voir F.11). Les nouveaux concepts doivent être associés à un numéro de catégorie, correspondant au contenu de la colonne *id_kategorie* de la table *konzepte_kategorien* de la base de données (voir C.1). La création d'une nouvelle catégorie peut s'effectuer à l'aide de l'outil *Concept Tree*, accessible via le menu Verba.

Après encodage d'un point d'enquête, il faut appuyer sur le bouton *Insert*, qui insère automatiquement la donnée encodée dans la table *aeusserungen*.

D.1.2. Codage des caractères

L'encodage des formes doit suivre le système de codage Verba, appelé *betacode*. Ce codage est expliqué et documenté dans la zone en haut à droite de la fenêtre. Un clic sur le caractère désiré permet son insertion directe dans la zone d'encodage, à la droite des caractères déjà insérés.

Au fur et à mesure de l'encodage d'une forme, sa visualisation s'affiche à droite de la zone de texte (voir F.10). Si la visualisation contient des caractères en rouge, cela signifie que le codage en betacode est valide, mais que ce codage n'est pas encore associé, dans la table *codepage_original*, à un caractère permettant son affichage. Cette absence de visualisation n'empêche pas l'encodage. En revanche, si le message *Not valid* apparaît en rouge, cela signifie que le codage inséré par le rédacteur n'est pas valide. Il faut alors soit corriger l'encodage, soit ajouter ce codage dans la table *codepage_original* de la base de données, en concertation avec l'équipe de VerbaAlpina (voir C.1).

Il est toujours possible de postposer l'encodage d'une forme problématique (parce que l'encodage n'est pas valide ou parce que le rédacteur hésite sur la façon dont il doit encoder une forme, par exemple) en utilisant le bouton *Problem* (voir F.10), qui insère la donnée dans la table *aeusserungen* avec comme contenu <problem> (voir F.12). Les données ainsi marquées seront accessibles par la suite pour correction.

D.1.3. Correction de l'encodage

Si des points d'enquête déjà encodés doivent être modifiés, il est nécessaire de passer en mode de correction. Cela s'effectue en choisissant l'option *Correction* au lieu de *Initial recording* dans le menu déroulant. Il faut insérer le numéro du point d'enquête à corriger. Il est alors possible de modifier l'encodage de ce point, puis d'appuyer sur le bouton *Update* pour effectuer la mise à jour (voir F.13).

L'option *Problems* du même menu déroulant permet de revoir les points d'enquête qui ont été encodés en tant que <problem> lors de l'encodage initial et de les encoder à nouveau.

D.2. Tokenisation

D.2.1. Objectif

Pour que les données qui ont été encodées lors de l'étape précédente (voir C) s'affichent sur la page *Plan interactif* du site Web, il est nécessaire de les « tokeniser ». Cette étape permet notamment de diviser en plusieurs unités (*tokens*) les données qui se présentent sous la forme d'un syntagme, de façon à permettre plus tard leur typisation (voir D.3). Par ailleurs, les données encodées dans la table *aeusserungen* le sont en betacode uniquement ; ce sont les tokens créés durant cette étape de tokenisation qui seront pourvus d'une transcription permettant leur affichage en ligne, sous leur

forme originale ou sous une forme en API. Ces tokens, ainsi que leurs transcriptions, se trouvent dans la table *Tokens* de la base de données.

D.2.2. Fonctionnement

L'opération de tokenisation s'effectue via le menu *Verba Tools* du backend, sous l'onglet *Tokenization* (voir F.14). Il faut choisir la carte que l'on désire tokeniser en insérant dans la zone de texte le numéro d'identifiant de la carte, précédé du signe *égal*. Ce numéro d'identifiant se trouve dans la colonne *ID_Stimulus* de la table *stimuli* de la base de données (voir F.2) ; dans VerbaPicardia, il peut aussi être recherché via l'outil *Overview* du backend (voir D.5.1).

Une fois l'identifiant inséré, il faut appuyer sur *Aeusserungen suchen* pour voir s'afficher le nombre de formes de cette carte qui ne sont pas encore tokenisées. Il faut ensuite appuyer sur le bouton *Vorschau* pour prévisualiser le résultat de la tokenisation et voir s'il y a des problèmes. L'outil fournit notamment la liste des caractères qui ne sont pas encore pourvus d'une transcription phonétique en API : ce problème peut se régler plus tard (voir D.4).

Si tout est correct, il faut appuyer sur le bouton *Tokenisieren*. Les unités sont alors tokenisées et insérées dans la table *tokens* de la base de données. Pour que les données tokenisées s'affichent en ligne, il faut ensuite compiler les données (voir ci-dessus C.2).

D.2.3. Affichage des caractères

Sur la carte interactive, il se peut que certains caractères s'affichent en betacode et non sous leur forme correcte. Pour résoudre ce problème, il faut tout d'abord mettre à jour la table *codepage_original*, en concertation avec l'équipe de VerbaAlpina, afin d'y ajouter les caractères manquants. Il faut ensuite aller dans le backend, dans l'outil *Beta -> Original* (accessible via le menu *Verba Tools*), et appuyer sur le bouton *Fix Tokens*. Il ne faut pas oublier, ensuite, de compiler les données (voir C.2).

Le bouton *Fix Tokens* met à jour la colonne *Original* de la table *Tokens* en fonction du contenu de la table *codepage_original*. Les codes en betacode qui n'ont pas pu être transcrits sont listés dans la fenêtre *Results*. Cet outil peut donc être utilisé à tout moment pour connaître la liste des caractères pour lesquels un problème d'affichage persiste (voir F.15).

D.3. Typisation

D.3.1. Objectif

Une fois la base de données compilée après tokenisation, toutes les données qui ont été tokenisées s'affichent sur la carte interactive du site Web. Une recherche par concept permet dès lors l'affichage d'une carte correspondant à ce concept. Il est possible d'améliorer la représentation cartographique de ces données en les typisant, de façon à permettre d'une part l'interrogation des données selon leur type étymologique ou morpholexical et, d'autre part, une représentation cartographique qui mette en évidence la distribution des données selon leur type (voir F.16).

D.3.2. Fonctionnement

La typisation des données s'effectue via l'outil *Typification* disponible dans le menu Verba du backend. Il faut choisir un atlas, puis une carte de cet atlas, qui s'affiche alors dans la fenêtre. Il est possible de sélectionner soit l'ensemble des données, soit uniquement les données non encore typisées (en cochant l'option *Nur Belege ohne morph.-lex. typisierung anzeigen*), soit uniquement

les données non encore associées à un concept (en cochant l'option *Nur Belege ohne Konzeptzuweisung anzeigen*). Les données apparaissent dans le menu déroulant sous *Belege*, regroupées selon leur forme (voir F.17).

Pour typer un ensemble de formes, il faut d'abord les sélectionner une par une pour qu'elles apparaissent dans le tableau (voir F.18). Il faut ensuite sélectionner un type morpholexical dans le menu déroulant sous *VA-Typ zuweisen*, puis appuyer sur le bouton *Assign type*. Les données du tableau sont alors typisées : cette opération met à jour la table *vtbl_token_morph_typ* de la base de données.

Remarque. Si l'option *Nur Belege ohne morph.-lex. Typisierung anzeigen* a été cochée, les données typisées n'apparaissent plus dans le menu déroulant. Si l'option n'a pas été cochée, elles apparaissent encore dans le menu déroulant, mais en caractères non grasses.

Il se peut que certaines données ne doivent pas être typisées, par exemple dans le cas d'une préposition au sein d'un syntagme (par exemple *de* dans *noix de moulin*). Il faut alors sélectionner ces données dans le menu déroulant pour les ajouter au tableau, puis appuyer sur le bouton *Confirm* sous *Typification not necessary* (voir F.19).

Une fois que les données d'une carte ont été typisées, il est nécessaire de compiler les données (voir C.2). Les données sont alors interrogeables selon leur type de base et leur type morpholexical, et il est possible de visualiser leur distribution cartographique selon ces critères (voir D.3.1).

D.3.3. Création d'un type morpholexical

Si le type morpholexical que l'on veut attribuer à une donnée n'apparaît pas dans le menu déroulant sous *VA-Typ zuweisen*, il faut le créer. La création d'un nouveau type s'effectue via le bouton *Create new type*, qui ouvre une fenêtre intitulée *Morpho-Lexicalischer Typ* (voir F.20). Le type doit être encodé dans le champ *Orth*. L'appartenance linguistique, sous *Sprache*, est à choisir entre *rom* (langue romane), *germ* (langue germanique) ou *sla* (langue slave). Il suffit ensuite d'appuyer sur *Bestätigen* en bas de la fenêtre pour ajouter le nouveau type morpholexical au menu déroulant et, par la même occasion, à la table *morph_typen* de la base de données.

Notons que cette fenêtre *Morpho-Lexicalischer Typ* est accessible également pour les types déjà créés, si l'on veut en modifier ou en compléter les informations. Il suffit alors de choisir le type morpholexical dans le menu déroulant sous *VA-Typ zuweisen*, ensuite d'appuyer sur le bouton *Typ bearbeiten*.

D.3.4. Création d'un type de base (étymon) et liens vers le FEW

Il est possible (mais non obligatoire) d'attribuer un étymon à chaque type morpholexical. Cela s'effectue au moyen de la fenêtre *Morpho-Lexicalischer Typ* (voir ci-dessus D.3.3). Dans cette fenêtre se trouve un menu déroulant sous *Zugeordnete Basistypen*, qui affiche la liste des étymons déjà encodés dans la table *basistypen* de la base de données. Il est alors possible de sélectionner l'étymon voulu (voir F.21).

Si l'étymon n'apparaît pas dans la liste des étymons, il faut le créer en cliquant sur *Neuen Basistyp anlegen*. Une fenêtre intitulée *Basistyp* s'ouvre alors. On peut y encoder la forme de l'étymon (sous *Orth*), son appartenance linguistique (sous *Sprache*) et également une source : typiquement le FEW pour les données de VerbaPicardia (voir F.22).

Les références au FEW déjà encodées apparaissent dans le menu déroulant sous *Zugeordnete Referenzen*. Il est possible d'ajouter une nouvelle référence en cliquant sur *Neue Referenz anlegen*. Cette opération ouvre une fenêtre intitulée *Lemmata_Basistypen* (voir F.23), dans laquelle on peut

indiquer la source lexicographique (typiquement le FEW), le lemme (sous *Subvozem*) et également l'URL de la page en ligne de cette source, sous *Link*. Il faut appuyer ensuite sur *Insert*, qui ajoute les informations ainsi encodées dans la table *lemmata_basistypen* de la base de données.

Les étymons et les liens vers le FEW qui ont été encodés de cette façon sont accessibles en ligne sur la carte interactive pour chacune des données typisées. L'étymon s'affiche en tant que « type de base ». Si le lien vers le FEW a été indiqué pour cet étymon, il est accessible via une icône avec la lettre *F* située à côté de la forme de l'étymon (voir F.16). Un clic sur cette icône renvoie directement à la page du FEW qui a été indiquée dans la fenêtre *Lemmata_Basistypen*. Ces informations peuvent être modifiées à tout moment via le menu Verba du backend, sous l'onglet *Base Types*.

D.4. Transcription en API

D.4.1. Objectif

Le système Verba offre la possibilité d'afficher les données selon leur transcription phonétique en API. Par exemple, le projet VerbaPicardia affiche les données de l'ALF selon la transcription phonétique qui a été établie dans le cadre du projet Symila (voir http://symila.univ-tlse2.fr/alf/notation_phonetique).

Sur la carte interactive, chaque forme s'affiche par défaut en API lorsque cette transcription API est disponible. La forme originale de l'atlas (non API) reste accessible : elle s'affiche lorsqu'on passe la souris sur la forme. Si la transcription API n'existe pas pour cette forme, elle s'affiche uniquement en mode « original ».

D.4.2. Fonctionnement

Pour que les données des atlas s'affichent en API, il est tout d'abord nécessaire d'associer à chaque caractère sa transcription phonétique. Cette transcription est propre à chaque atlas. La liste des caractères et de leur transcription en API se trouve dans la table *codepage_ipa* de la base de données (voir F.24). Cette table peut être mise à jour à tout moment, en ajoutant des caractères ou en modifiant leur transcription phonétique.

Pour qu'une donnée s'affiche en API sur la carte interactive, il faut en outre que la table *Tokens* contienne, dans la colonne *IPA*, la transcription en API de cette donnée. Cette transcription est générée automatiquement à partir des données de la table *codepage_ipa*. Pour effectuer cette opération automatique (par exemple après une modification de la table *codepage_ipa*), il faut aller dans le menu *Verba Tools* du backend, où se trouve un outil intitulé *Beta -> IPA*.

L'outil *Beta->IPA* du backend affiche pour chaque atlas (à choisir dans le menu déroulant) le nombre de tokens qui ne sont pas encore pourvus d'une transcription phonétique. Un clic sur le bouton *IPA berechnen* permet de générer automatiquement une transcription phonétique pour chaque token qui n'en est pas encore pourvu. L'outil fournit dans la fenêtre intitulée *Ergebnis* la liste des formes qui ont pu être transcrites durant cette opération. La fenêtre intitulée *Fehler* indique la liste des caractères qui n'ont pas pu être transcrits (voir F.25). Cette fenêtre *Fehler* est donc très utile pour connaître la liste des caractères qui, dans chaque atlas, ne disposent pas encore d'une transcription en API.

Remarque. En cas de modification de transcriptions existantes dans la table *codepage_ipa*, il est nécessaire de décocher l'option *Nur Tokens ohne IPA-Darstellung*, afin de régénérer une nouvelle transcription API pour l'ensemble des tokens existants.

D.5. Vue d'ensemble

Dans le projet VerbaPicardia, le menu du backend présente un outil intitulé *Overview*. Cet outil offre trois onglets présentant chacun une vue d'ensemble de l'avancement de l'encodage.

D.5.1. État d'avancement par carte

L'onglet *Transkription* présente l'état d'avancement de l'encodage carte par carte. Chaque carte est identifiée par un numéro unique (correspondant au contenu de la colonne *ID Stimulus* dans la table *stimuli* de la base de données), suivi de l'atlas, du numéro de carte et de son intitulé. Pour chacune des cartes est indiqué le nombre de points d'enquête encodés par rapport au nombre total de points d'enquête. La ligne s'affiche en vert si tout est encodé, en bleu si l'encodage n'est pas terminé et en jaune s'il y a des points d'enquête marqués comme problématiques (voir F.26 et D.1.2).

D.5.2. État d'avancement par concept

L'onglet *Stimuli/Konzepte* présente l'état d'avancement de l'encodage concept par concept. Pour chaque concept sont indiquées les cartes qui ont été encodées, le nombre d'attestations encodées (correspondant au contenu de la table *aeusserungen* de la base de données), le nombre d'attestations tokenisées et le nombre de tokens (voir F.27).

L'onglet *Atlanten/Konzepte* présente quant à lui un résumé succinct des atlas encodés pour chaque concept (voir F.28).

E. Requêtes SQL

Nous fournissons ici quelques requêtes SQL utiles pour les rédacteurs qui désirent extraire des informations hors de la base de données.

E.1. Effectuer une requête et exporter le résultat

Pour effectuer une requête dans la base de données, il faut entrer dans la base de données Verba, dans la version souhaitée (désignée par *va_1*, *va_2* etc. ou par *va_xxx* pour la version de travail), et cliquer sur l'onglet SQL. Il faut insérer l'une des requêtes ci-dessous dans la fenêtre qui s'ouvre (voir F.29).

Pour exporter les résultats, il faut appuyer sur le bouton *Exporter* (voir F.30) et choisir, dans la fenêtre qui s'ouvre, le format d'exportation désiré ainsi que le nom de fichier (voir F.31).

E.2. Générer la liste des points d'enquête pour un atlas

```
SELECT
  Erhebung, Nummer, Ortsname
FROM
  informanten
WHERE
  Erhebung = 'ALW'
ORDER BY
  Erhebung ASC
;
```

E.3. Générer la liste des formes encodées pour une carte, avec leur numéro de point

Dans la requête ci-dessous, remplacer X par le numéro d'identifiant de la carte (voir D.5.1).

```
SELECT
  s.Erhebung, s.Karte, s.Stimulus,
  i.Nummer,
  a.Aeusserung, a.Erfasst_Von
FROM
  aeusserungen a
  JOIN stimuli s ON a.Id_Stimulus = s.ID_Stimulus
  JOIN informanten i ON a.Id_Informant = i.Id_Informant
WHERE
  a.Id_Stimulus = X
ORDER BY
  s.Erhebung, s.Karte, CONVERT(i.Nummer, UNSIGNED INTEGER)
;
```

E.4. Générer la liste des formes encodées pour une carte, avec leur numéro de point : cas particulier (ALW)

Dans la requête ci-dessous, remplacer X par le numéro d'identifiant de la carte (voir D.5.1).

```
SELECT
  s.Erhebung, s.Karte, s.Stimulus,
  i.Nummer,
  a.Aeusserung, a.Erfasst_Von
FROM
  aeusserungen a
  JOIN stimuli s ON a.Id_Stimulus = s.ID_Stimulus
  JOIN informanten i ON a.Id_Informant = i.Id_Informant
WHERE
  a.Id_Stimulus = X
ORDER BY
  s.Erhebung, s.Karte, i.Nummer
;
```

E.5. Générer la liste des transcriptions API pour un atlas

```
SELECT
  *
FROM
  codepage_ipa
WHERE
  Erhebung = 'ALF'
;
```

F. Annexe : illustrations (dans un document à part)