

# PanLex 2.5: The Database Design

Jonathan Pool  
Utilika Foundation

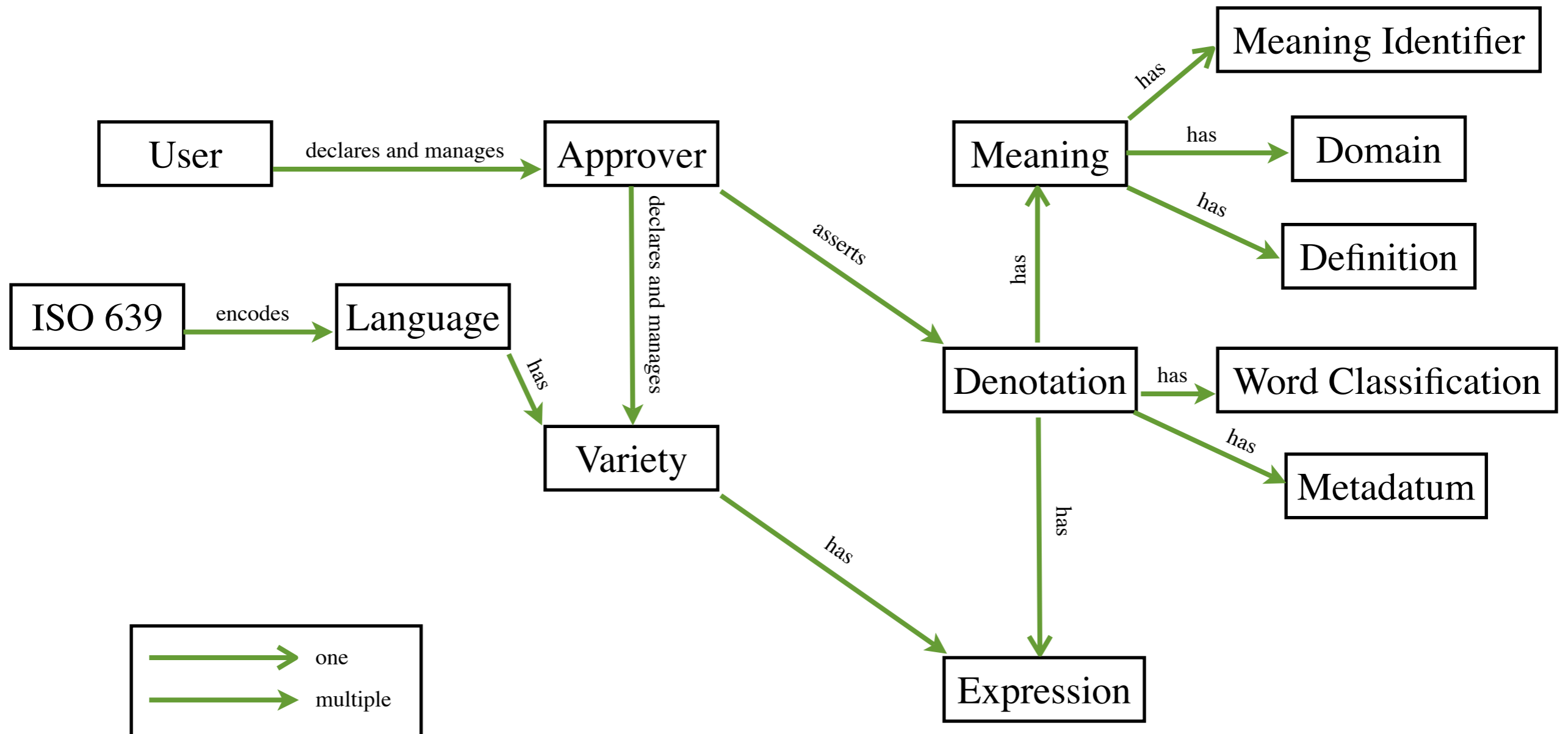
August 2010

# 1. Introduction

- A. PanLex is a database that represents assertions about the meanings of expressions. To be specific:
- B. **Database**: A relational database (PostgreSQL).
- C. **Assertions**: Factual claims, not infallible truths. “A says B = C”, not “B = C”. Assertions can disagree.
- D. **Expressions**: Expressions that are lexemes. A lexeme is an entry (word or phrase) in the lexicon of a language. It is represented in a “citation” or “dictionary” form, i.e. as a “lemma”. E.g., “go” is a lemma, “went” is not. A noncompositional phrase (one not interpretable from its parts) is a lemma: “green thumb” is a lexeme, “green paint” is not.
- E. **Meanings**: Expressions have meanings. Expressions that share a meaning are translations (or, if in the same language, synonyms).

# 1. Introduction

## Entities and Relationships in PanLex



## 2. Constraints

Sources of assertions vary in detail. PanLex is designed to accept even the simplest assertions, plus additional assertion types from richer sources.

Simple

Rich

Basque Russian



**EUSKARA**  
Баскский язык

**Баскско-русский словарь**

ABCDEFGHIJKLMN OPSTUXZ

Aberats богатый  
abertzale националист  
abestu петь  
abiada скорость  
abiadura скорость  
adar ветвь  
adibide пример  
adin возраст

**fouyapin** également "foubap" selon le Dictionnaire de Pouillet et al., 1990), aussi "friyapen", avec la variante "penbwa" d'après R. Confiand ([dictionnaire en ligne](#)) : **arbre à pain** **fruit à pain**.

Il est intéressant de signaler que si, à l'heure actuelle, ne sont répertoriées aux Antilles dans les dictionnaires courants que ces formes calquées sur le mot français, il existe à la Dominique (île voisine indépendante après avoir été colonie britannique au XIXe et XXe siècle) pour désigner la même réalité dans le créole local, le terme de "yanm-pen" (lit. igname-pain) qui tend à faire de "[yanm](#)" un terme générique pour "nourriture". Marcel Fontaine dans son dictionnaire cite également l'usage de "pen-pen" - sans indiquer toutefois s'il faut en rapporter l'usage à un groupe particulier. De fait "penbwa" (pour arbre à pain) est attesté à Sainte-Lucie (île également indépendante après la colonisation britannique).

Ces mots composés créoles sont particulièrement significatifs et intéressants : ne peut-on pas penser que le créole de la Dominique et le créole de Ste-Lucie nous livrent là un usage "non-contaminé" par le français ? Resterait à chercher si ces formes ont été également attestées en créole au XIXe siècle en Guadeloupe et Martinique.

En Haïti semble attestée la forme (un peu étonnante : on s'interroge sur son origine) de "lam"/"lanm" ou "lam véritab" (cf. in Dictionnaire d'Albert Valdman et al., 1996, mais aussi in Wally R. Turnbull, 2003 : *Creole Made Easy*)

# 3. Users

Assertions in PanLex come from **users**.

Table “us”

Column	Type	Modifiers	Storage	Description
us	integer	not null	plain	ID
dt	date	not null default ('now'::text)::date	plain	enrollment date
nm	text		extended	name
al	text	not null	extended	alias (username)
sm	text		extended	SMTP (Internet mail) address
ht	text		extended	HTTP (World Wide Web) address (URL)
pw	character(32)	not null	extended	password
ok	boolean	not null default false	plain	whether approved

Indexes:

"us\_pkey" PRIMARY KEY, btree (us) CLUSTER

"us\_al\_key" UNIQUE, btree (al)

Referenced by:

TABLE "ap" CONSTRAINT "ap\_us\_fkey" FOREIGN KEY (us) REFERENCES us(us)

TABLE "auto" CONSTRAINT "auto\_tsus\_fkey" FOREIGN KEY (tsus) REFERENCES us(us)

Has OIDs: no

# 4. Approvers

Users base assertions on particular sources, such as dictionaries or thesauri (or their personal knowledge). The combination of user + source is an asserted fact's **approver**.

## Table “ap”

Column	Type	Modifiers	Storage	Description
ap	integer	not null	plain	ID
us	integer	not null	plain	managing user
dt	date	not null default ('now'::text)::date	plain	registration date
tt	text	not null	extended	label
ur	text		extended	URI
bn	text		extended	ISBN
au	text		extended	author
ti	text		extended	title
pb	text		extended	monograph publisher or serial title, volume, and page range
yr	smallint		plain	year of publication
uq	smallint		plain	quality measure specified by the user
ui	smallint		plain	numeric ID specified by the user
ul	text		extended	miscellaneous information
li	character(2)		extended	type of offered license
ip	text		extended	summary of intellectual-property claim
co	text		extended	name of apparent intellectual-property claimant
ad	text		extended	SMTP address for licensing correspondence
fp	text		extended	source file or directory path

Indexes:

"ap\_pkey" PRIMARY KEY, btree (ap) CLUSTER

"ap\_tt\_key" UNIQUE, btree (tt)

"ap\_us\_key" btree (us)

Foreign-key constraints:

"ap\_li\_fkey" FOREIGN KEY (li) REFERENCES apli(li)

"ap\_us\_fkey" FOREIGN KEY (us) REFERENCES us(us)

Referenced by:

TABLE "af" CONSTRAINT "af\_ap\_fkey" FOREIGN KEY (ap) REFERENCES ap(ap) ON UPDATE CASCADE ON DELETE CASCADE

TABLE "av" CONSTRAINT "av\_ap\_fkey" FOREIGN KEY (ap) REFERENCES ap(ap) ON UPDATE CASCADE ON DELETE CASCADE

TABLE "mn" CONSTRAINT "mn\_ap\_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)

Has OIDs: no

# 5. Language Varieties

Each expression is in a language **variety**. It may be the standard variety, a dialect, a script-based variety, a controlled technical variety, etc. Its language is specified with a 3-letter ISO 639 code.

Table “lv”

Column	Type	Modifiers	Storage	Description
lv	integer	not null	plain	ID
lc	character(3)	not null	extended	ISO 639 code
vc	smallint	not null	plain	language-specific ID
sy	boolean	not null default true	plain	whether the variety permits synonymy
am	boolean	not null default true	plain	whether the variety permits ambiguity
ap	integer	not null	plain	managing approver
tt	text	not null	extended	label (with no &, ", <, >)

Indexes:

"lv\_pkey" PRIMARY KEY, btree (lv) CLUSTER

"lv\_lc\_key" UNIQUE, btree (lc, vc)

Foreign-key constraints:

"lv\_lc\_fkey" FOREIGN KEY (lc) REFERENCES lc(lc)

Referenced by:

TABLE "av" CONSTRAINT "av\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv) ON UPDATE CASCADE ON DELETE CASCADE

TABLE "cp" CONSTRAINT "cp\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

TABLE "cu" CONSTRAINT "cu\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

TABLE "df" CONSTRAINT "df\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

TABLE "ex" CONSTRAINT "ex\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

TABLE "pl1" CONSTRAINT "pl1\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

Has OIDs: no

# 5. Language Varieties

## Example

PanLex has 10 varieties to which ISO 639 code “cmn” (Mandarin) has been assigned.

lv	lc	vc	sy	am	ap	tt
1627	cmn	0	t	t	6	简体字
1628	cmn	1	t	t	6	繁體中文
128	cmn	2	t	t	6	官話
1835	cmn	3	t	t	6	pīnyīn
2166	cmn	4	t	t	6	Muping
2561	cmn	5	t	t	6	Xi'an
3252	cmn	6	t	t	6	Chengdu
3253	cmn	7	t	t	6	Yangzhou
3254	cmn	8	t	t	6	Nanjing
3255	cmn	9	t	t	6	Ürümqi

(10 rows)

Comment: An lc-vc pair (e.g., “cmn, 4”) uniquely identifies a language variety, equivalently to an lv (e.g., 2166). One application accepts and reports language-variety descriptors in the format “cmn-004”.

## 6. Expressions

An **expression** is distinguished by its variety and a textual representation of its lemma.

Table “ex”

Column	Type	Modifiers	Storage	Description
ex	integer	not null	plain	ID
lv	integer	not null	plain	variety
tt	text	not null	extended	text

Indexes:

```
"ex_pkey" PRIMARY KEY, btree (ex)
"ex_lv_key" UNIQUE, btree (lv, tt)
"ex_lv" btree (lv) CLUSTER
```

Foreign-key constraints:

```
"ex_lv_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)
```

Referenced by:

```
TABLE "dm" CONSTRAINT "dm_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
TABLE "dn" CONSTRAINT "dn_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
TABLE "p10" CONSTRAINT "p10_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
TABLE "p11" CONSTRAINT "p11_ex_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)
```

Has OIDs: no

Comment: A lemma may exist in multiple language varieties. For example, “mata” is an expression in 211 different language varieties in PanLex.

Comment: A lemma may belong to no more than one expression per language variety. For example, “bear” in English is only one expression in PanLex.

# 7. Meanings

**Meanings** are approver-specific.

## Table “mn”

Column	Type	Modifiers	Storage	Description
mn	integer	not null	plain	ID
ap	integer	not null	plain	approver

Indexes:

"mn\_pkey" PRIMARY KEY, btree (mn)

"mn\_ap" btree (ap) CLUSTER

Foreign-key constraints:

"mn\_ap\_fkey" FOREIGN KEY (ap) REFERENCES ap(ap)

Referenced by:

TABLE "df" CONSTRAINT "df\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

TABLE "dm" CONSTRAINT "dm\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

TABLE "dn" CONSTRAINT "dn\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

TABLE "mi" CONSTRAINT "mi\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

TABLE "p10" CONSTRAINT "p10\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

TABLE "p11" CONSTRAINT "p11\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

Has OIDs: no

# 8. Denotations

**Denotations** are expression-meaning pairs, i.e. assertions that particular expressions have particular meanings.

Table “dn”

Column	Type	Modifiers	Storage	Description
dn	integer	not null	plain	ID
mn	integer	not null	plain	meaning
ex	integer	not null	plain	expression

Indexes:

"dn\_pkey" PRIMARY KEY, btree (dn)  
"dn\_mn\_key" UNIQUE, btree (mn, ex) CLUSTER  
"dn\_ex" btree (ex)  
"dn\_mn" btree (mn)

Foreign-key constraints:

"dn\_ex\_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)  
"dn\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

Referenced by:

TABLE "md" CONSTRAINT "md\_dn\_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)  
TABLE "p11" CONSTRAINT "p11\_mnex\_fkey" FOREIGN KEY (mn, ex) REFERENCES dn(mn, ex)  
TABLE "wc" CONSTRAINT "wc\_dn\_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)

Has OIDs: no

# 9. Meaning Identifiers

A meaning may optionally have a (single) textual **meaning identifier**. It may serve to link the data in PanLex to a richer record elsewhere.

Table “mi”

Column	Type	Modifiers	Storage	Description
mn	integer	not null	plain	meaning
tt	text	not null	extended	text

Indexes:

"mi\_pkey" PRIMARY KEY, btree (mn) CLUSTER

Foreign-key constraints:

"mi\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

Has OIDs: no

Example: This meaning identifier allows access to taxonomic data about the meaning in the Arabic WordNet.

mn	tt
3613667	Almmlkp_Almgrbyp_n1AR

```
<item itemid="Almmlkp_Almgrbyp_n1AR"
offset="108413097" lexfile="" name="المملكة المغربية"
type="synset" headword="" POS="n" source="NE file"
gloss="Morocco." authorshipid="1980" />
<authorship author="horacio" date="20070322" score=""
comment="NE import from file couma.out.
#Morocco#Kingdom of Morocco#" covering="0"
authorshipid="1980" />
```

# 10. Domain Descriptors

A meaning may optionally have **domain descriptors**. They are PanLex expressions.

Attaching a domain descriptor to a meaning asserts that the meaning is within the domain described by the expression.

Table “dm”

Column	Type	Modifiers	Storage	Description
dm	integer	not null	plain	ID
mn	integer	not null	plain	meaning
ex	integer	not null	plain	expression

Indexes:

"dm\_pkey" PRIMARY KEY, btree (dm)

"dm\_mn\_key" UNIQUE, btree (mn, ex) CLUSTER

Foreign-key constraints:

"dm\_ex\_fkey" FOREIGN KEY (ex) REFERENCES ex(ex)

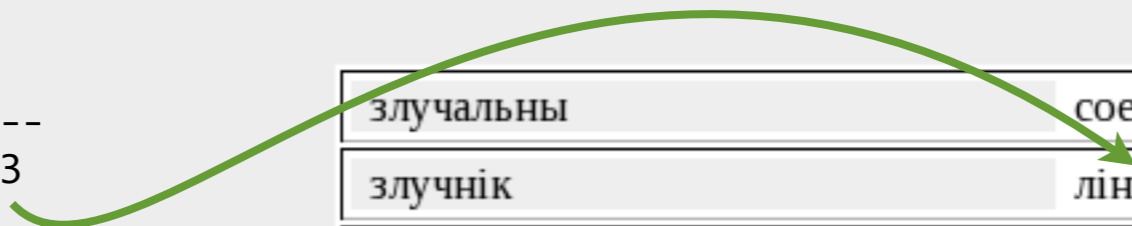
"dm\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

Has OIDs: no

Example: This approver says that “злучнік” in Belorussian and “союз” in Russian share a meaning in the domain (linguistics) described by “лінгв.” in Belorussian, which is expression 57303 in PanLex.)

dm	mn	ex
38611	5149560	57303

злучальны	соединительный
злучнік	лінгв. союз
злучок	лінгв. дефис



# 11. Definitions

A meaning may optionally have **definitions**. A definition, which is a text in some variety, describes a meaning with more than a lexeme, so it is not a PanLex expression.

Table “df”

Column	Type	Modifiers	Storage	Description
df	integer	not null	plain	ID
mn	integer	not null	plain	meaning
lv	integer	not null	plain	variety of the text
tt	text	not null	extended	text

Indexes:

"df\_pkey" PRIMARY KEY, btree (df)

"df\_mn\_key" UNIQUE, btree (mn, lv, tt) CLUSTER

Foreign-key constraints:

"df\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

"df\_mn\_fkey" FOREIGN KEY (mn) REFERENCES mn(mn)

Has OIDs: no

Example: A source translates “couronne” in Cajun French into “crown” and “a wreath of flowers traditionally worn by a bride” in English. It is appropriate to treat “crown” as an expression and “a wreath ... bride” as a definition in PanLex.

# 12. Word Classifications

A denotation may optionally have **word classifications**. These assign grammatical word classes (parts of speech), from a closed set, to denotations.

Table “wc”

Column	Type	Modifiers	Storage	Description
wc	integer	not null	plain	ID
dn	integer	not null	plain	denotation
ex	integer	not null	plain	PanLex word-class expression

Indexes:

- "wc\_pkey" PRIMARY KEY, btree (wc)
- "wc\_dn\_key" UNIQUE, btree (dn, ex) CLUSTER

Foreign-key constraints:

- "wc\_dn\_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)
- "wc\_ex\_fkey" FOREIGN KEY (ex) REFERENCES wcex(ex) ON UPDATE CASCADE

Has OIDs: no

noun	Common noun
name	Proper noun
pron	Pronoun
verb	Verb
vpar	Verb particle
auxv	Auxiliary verb
adjv	Adjective
detr	Determiner
advb	Adverb
prep	Preposition
post	Postposition
conj	Conjunction
ijec	Interjection
affx	Affix
punc	Punctuation
misc	Other

OLIF

VALUE	DESCRIPTION
noun	noun
verb	verb
adj	adjective
adv	adverb
prep	preposition
conj	conjunction
det	determiner
part	verb particle
auxverb	auxiliary verb
pron	pronoun
punc	punctuation
other	other pos to be determined by user

Comment: The PanLex word-class list extends that of the Open Lexicon Interchange Format (OLIF) standard.



# 13. Metadata

A denotation may optionally have **metadata**. These assign variable-value pairs to expressions with particular meanings. Variables and values are arbitrary texts.

Table “md”

Column	Type	Modifiers	Storage	Description
md	integer	not null	plain	ID
dn	integer	not null	plain	denotation
vb	text	not null	extended	variable
vl	text	not null	extended	value

Indexes:

"md\_pkey" PRIMARY KEY, btree (md)

"md\_dn\_key" UNIQUE, btree (dn, vb, vl) CLUSTER

Foreign-key constraints:

"md\_dn\_fkey" FOREIGN KEY (dn) REFERENCES dn(dn)

Has OIDs: no

Example: An English expression “pig” when synonymous with “police officer” could be annotated with a metadatum whose variable is “prag” and whose value is “vulg.”

Comment: A metadatum whose variable is a language-variety descriptor, such as “fra-000”, can thereby optionally contain a PanLex expression.

# 14. Approver Varieties

An approver may optionally have **approver varieties**. These are the language varieties that the approver makes assertions about expressions in.

Table “av”

Column	Type	Modifiers	Storage	Description
ap	integer	not null	plain	approver
lv	integer	not null	plain	variety

Indexes:

"av\_pkey" PRIMARY KEY, btree (ap, lv) CLUSTER

Foreign-key constraints:

"av\_ap\_fkey" FOREIGN KEY (ap) REFERENCES ap(ap) ON UPDATE CASCADE ON DELETE CASCADE

"av\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv) ON UPDATE CASCADE ON DELETE CASCADE

Has OIDs: no

# 15. Exemplar Characters

A language variety may optionally have **exemplar characters**. These are the characters that the Unicode Common Locale Data Repository designates as “exemplar characters” in the variety’s language. These are literal and quotation characters commonly encountered in expressions in the language.

Table “av”

Column	Type	Modifiers	Storage	Description
lv	integer	not null	plain	variety
c0	character(5)	not null	extended	start of character range
c1	character(5)	not null	extended	end of character range
loc	text		extended	locale
vb	text	not null	extended	variable

Indexes:

"cu\_c0\_key" UNIQUE, btree (lv, c0, loc, vb) CLUSTER

"cu\_c1\_key" UNIQUE, btree (lv, c1, loc, vb)

Foreign-key constraints:

"cu\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

Has OIDs: no

Comment: The start or end of a character range is represented with a 5-digit hexadecimal number. The “loc” field is a Unicode locale abbreviation, such as “Cyril”. The “vb” field’s value is “pri” (primary) or “aux” (auxiliary).

# 16. Approved Characters

A language variety may optionally have **approved characters**.

Table “av”

Column	Type	Modifiers	Storage	Description
lv	integer	not null	plain	variety
c0	character(5)	not null	extended	start of character range
c1	character(5)	not null	extended	end of character range

Indexes:  
"cp\_pkey" PRIMARY KEY, btree (lv, c0) CLUSTER  
"cp\_lv\_key" UNIQUE, btree (lv, c1)

Foreign-key constraints:  
"cp\_lv\_fkey" FOREIGN KEY (lv) REFERENCES lv(lv)

Has OIDs: no

Comment: The start or end of a character range is represented with a 5-digit hexadecimal number.

Comment: Exemplar characters do not include digits or any punctuation except quotation marks. Approved characters are not subject to any such restriction.

# 17. Permitted Editors

An approver may have a set of **permitted editors**.

Table “au”

Column	Type	Modifiers	Storage	Description
ap	integer	not null	plain	approver
us	integer	not null	plain	user permitted to edit the approver

Indexes:  
"au\_pkey" PRIMARY KEY, btree (ap, us)  
Has OIDs: no

# 18. Design and Implementation Issues

A. Multiform expressions. Should PanLex recognize the phenomenon of expressions that have sets of forms related by transliteration, register, region, etc.? How would this affect the matching of added expressions with existing ones?

B. Lemmatic objects. Should PanLex include a table of unique lemmata (expression texts) and reference them in the table of expressions? Would this make the database more compact at the price of processing complexity?

C. New attributes. Should PanLex recognize additional attributes, such as pronunciations, inflections, etymologies, word subclasses (gender, aspect, declension, etc.), registers, argument frames, and usage examples? Would managing their complexity conflict with expanding the coverage of low-density languages?

D. Domain control. Should PanLex adopt a universal list of recognized domains? Would doing so prejudge a not yet consensual issue?

E. Attribute generalization. Should PanLex permit word classifications and metadata to be attached not only to denotations, but also to meanings and/or expressions? If to expressions, would this complicate the acquisition of additional data?

# 18. Design and Implementation Issues

F. Metadatum structure. Would metadata be more useful if attribute-value pairs were replaced with PanLex expressions? If so, pairs of expressions (for attribute and value) or single expressions?

G. Valence. Should PanLex represent disapproval as well as approval?. Should it be possible to represent the assertion that expressions A and B share no meaning? Or would ratings of approvers substitute for this?

H. Confidence. Should PanLex represent probabilistic assertions? If so, what assertions should be eligible for this?

I. ID management. Should PanLex use serial generators as default ID values, instead of managing ID assignment? Would this simplification risk integer-range exhaustion in the table of denotations when large approvers are repeatedly refreshed? Could this risk be easily eliminated with periodic daemononic recompaction?

# 19. Philosophical Issues

A. Fundamental assumption. The design of PanLex is based on a fundamental assumption about the assertions made by lexical resources: that saying “B in language Y is a translation of A in language X in sense  $n$ ” is equivalent to saying “There is a meaning  $n$  that expressions A in language X and B in language Y both have”. (In practice, this amounts to treating a bilingual dictionary as if it were bidirectional, and treating a multilingual dictionary as if it had the semantic rigor of a multilingual thesaurus.) Is this a correct, or useful, assumption? An alternative assumption would be that this assertion is equivalent to saying “A in language X has a set of meanings including  $n$ , and in some contexts the best translation of meaning  $n$  of A into language Y is B”. Should PanLex be based on that assumption, instead? If so, how would that version of PanLex be implemented?

# Thanks

The database design of PanLex was inspired by the design of TransGraph, the sister (and original) project created by Kobi Reiter, Marcus Sammer, Michael Schmitz, Stephen Soderland, and others at the Turing Center of the University of Washington.

Revisions from PanLex 1.7 to Panlex 2.0 were based in part on helpful suggestions made by members of the University of Washington Computational Linguistics Laboratory, at a presentation, “PanLex and TransGraph Schema Choices”, on 29 October 2008.

Numerous other valuable suggestions have been received from Susan M. Colowick, Mausam, Oren Etzioni, PostgreSQL Experts, and others.