『日本語話し言葉コーパス』
RDB講習会 中級編

主催

国語研共同研究P「会話の韻律機能に関する実証的研究」（小磯花絵）
国語研共同研究P「多様な様式を網羅した会話コーパスの共有化」（伝康晴）
国語研共同研究P「コーパス日本語学の創成」（前川喜久雄）

共催

国立国語研究所コーパス開発センター

講習会の内容

- 『日本語話し言葉コーパス』RDB（CSJ-RDB）を研究に利用できるようになるために、次のことを学ぶ

  - CSJ-RDBの構成
  - SQL（RDBを操作するための言語）
    - 初級（SELECT文，WHERE句，JOIN句のうち内部結合）の復習
    - JOIN句（内部結合中級編，外部結合）
    - GROUP BY句
    - ORDER BY句
    - HAVING句
    - CASE式

第1部
CSJ-RDBの概要

CSJ-RDBについて

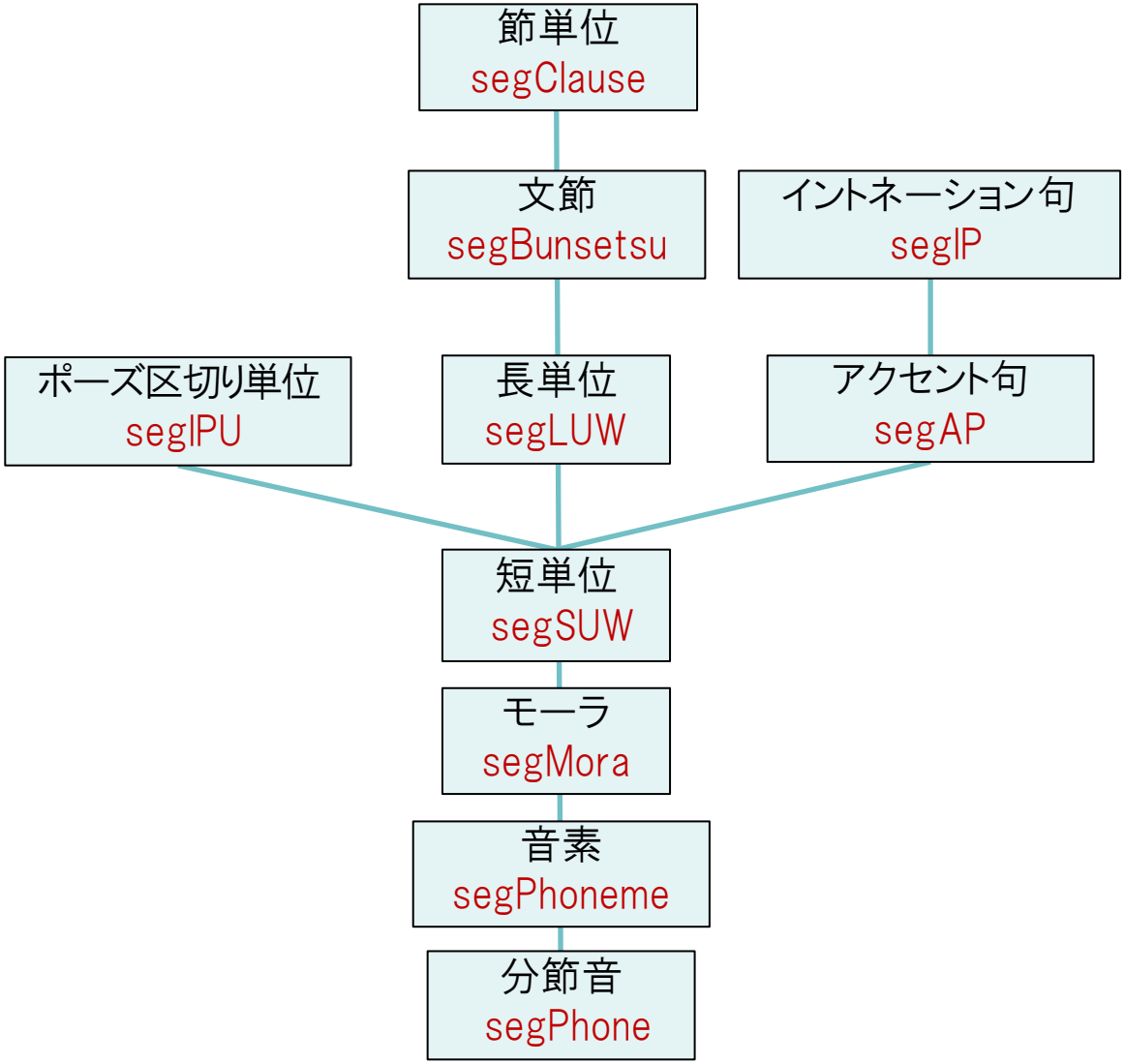
- CSJ-RDB とは
  - ✓ 『日本語話し言葉コーパス』のうちコア(201講演、約45時間)を対象
  - ✓ 第3刷のXML文書に含まれる情報をほぼ反映(若干、追加・修正)
- 参考資料
  - ✓ CSJ-RDBの概要・構成

    `http://www.ninjal.ac.jp/corpus_center/csj/data/` *
  - ✓ CSJ各種マニュアル

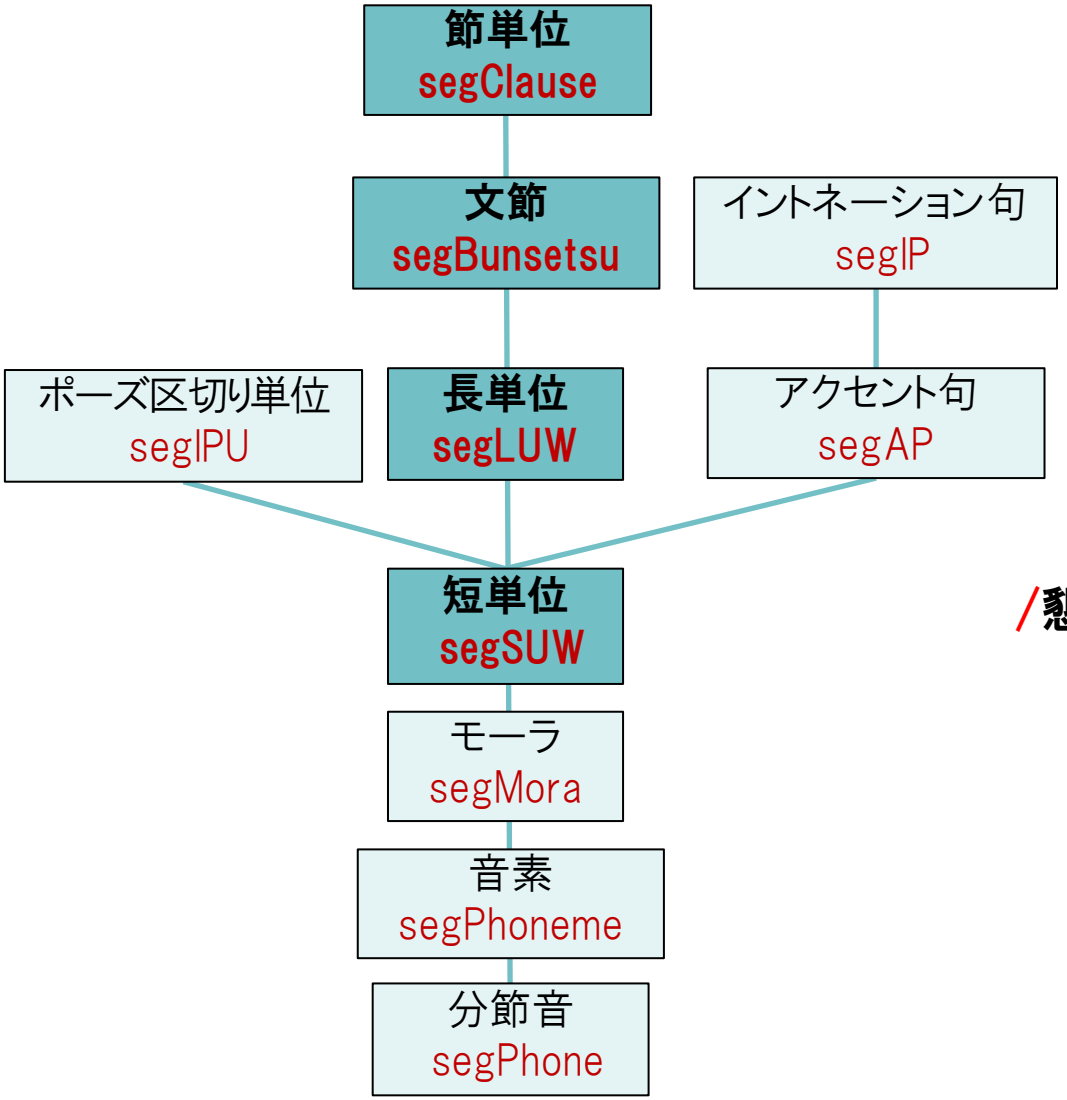
    `http://www.ninjal.ac.jp/corpus_center/csj/doc/` *

  * 近日中にCSJのURLが上記の通り変更(現在は"corpus_center/"の部分が不要)
- 今回配布した`csj_mini.db`(12講演分)についての注意事項
  - ✓ 二次配布は禁止
  - ✓ CSJ購入者以外は本データを使った研究発表は不可

CSJ-RDBの構成


節単位
segClause
文節
segBunsetsu
イントネーション句
segIP
ポーズ区切り単位
segIPU
長単位
segLUW
アクセント句
segAP
短単位
segSUW
モーラ
segMora
音素
segPhoneme
分節音
segPhone

CSJ-RDBの構成 ～統語・形態論の階層～



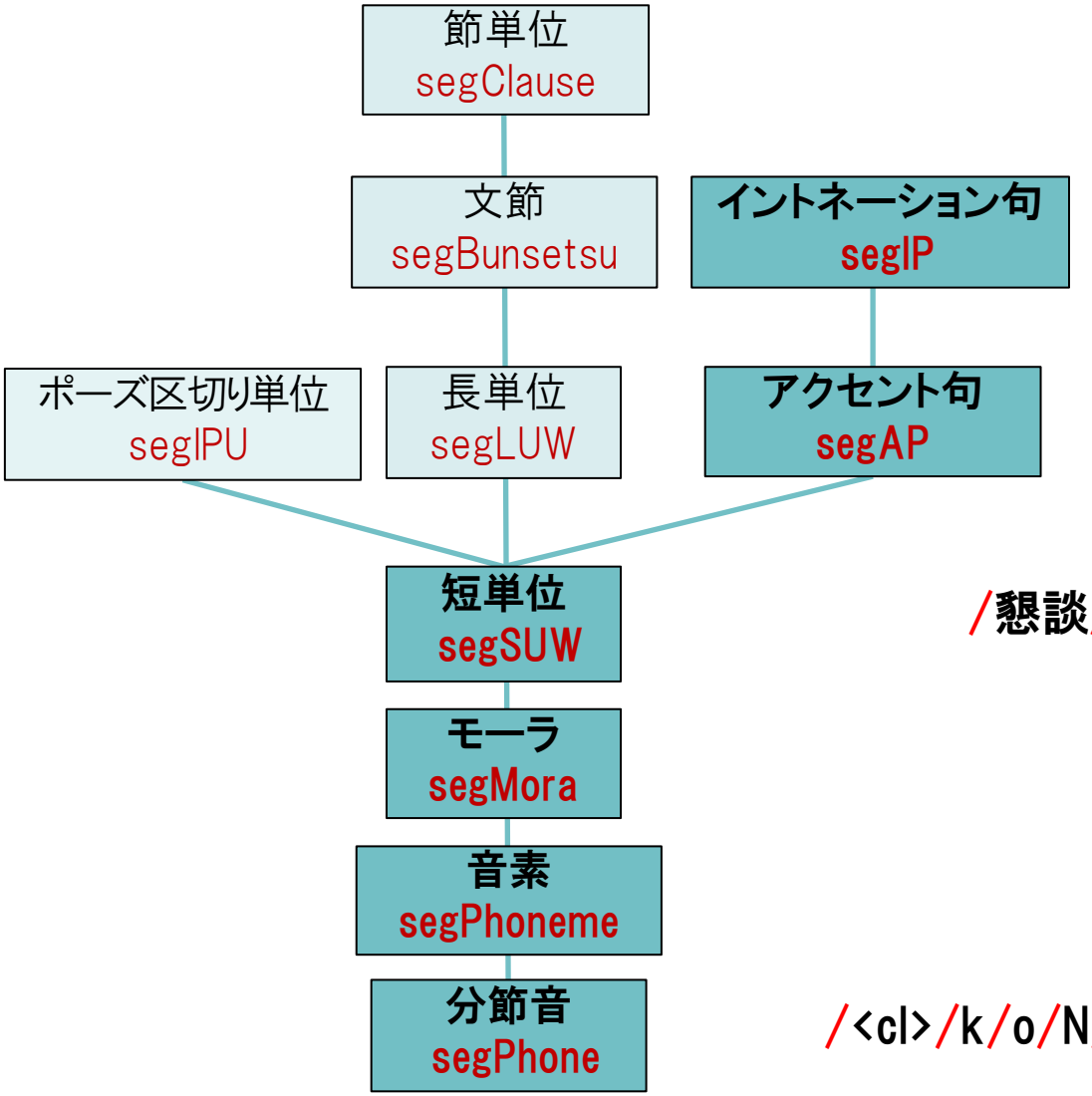
/懇談会に行ってもいいけど/

/懇談会に/行ってもいいけど/

/懇談会/に/行っ/てもいい/けど/

/懇談/会/に/行っ/て/も/いい/けど/

CSJ-RDBの構成 ～韻律・音韻の階層～



/懇談会に行ってもいいけど/

/懇談会に/行っても/いいけど/

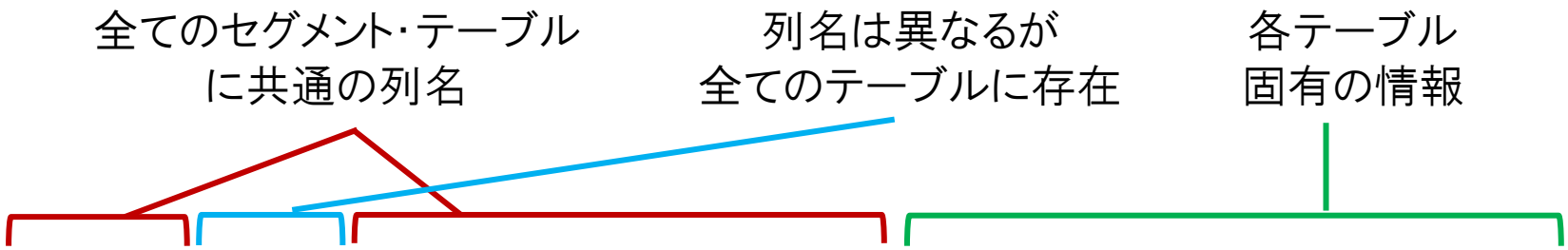
/懇談/会/に/行っ/て/も/いい/けど/

/コ/ン/ダ/ン/カ/イ/…/

/k/o/N/d/a/N/k/a/i/…/

/<cl>/k/o/N/<cl>/d/a/N/<cl>/k/a/i/…/

テーブルのサンプル ―segPhone―



TalkID	PhoneID	StartTime	EndTime	Channel	PhoneEntity	PhoneClass	Devoiced	…
A01F0055	05551385L	5.551385	5.632454	L	h	consonant	0	
A01F0055	05632454L	5.632454	5.698874	L	a	vowel	0	
A01F0055	05698874L	5.698874	5.760029	L	Q	special	0	
A01F0055	05760029L	5.760029	5.821184	L	SclS	others	0	
A01F0055	05821184L	5.821184	5.837566	L	py	consonant	0	
A01F0055	05837566L	5.837566	5.907904	L	o	vowel	0	
A01F0055	05907903L	5.907903	5.978241	L	H	special	0	
A01F0055	05978241L	5.978241	6.028152	L	sj	consonant	0	
A01F0055	06028153L	6.028153	6.078064	L	i	vowel	1	
A01F0055	06078064L	6.078064	6.16653	L	m	consonant	0	
A01F0055	06166530L	6.16653	6.277931	L	a	vowel	0	
A01F0055	06277931L	6.277931	6.382126	L	s	consonant	0	
A01F0055	06382126L	6.382126	6.532801	L	u	vowel	0	

第2部
講習会初級編の復習

RDBとは

- 相互に関係づけ可能な複数のテーブルの集合

table1　談話情報

TalkID	TalkType	SpeakerID
A01F0055	monolog	459
D01F0023	dialog	514

table2　話者情報

SpeakerID	性別	出身地
459	男	東京都
514	男	神奈川県

table3　単語情報

TalkID	単語	品詞
A01F0055	発表	名詞
A01F0055	し	動詞
A01F0055	まし	助動詞
A01F0055	た	助動詞
D01F0023	報告	
D01F0023	し	
D01F0023	ます	

三つのテーブルを共通キーで
関係づけて結合すると…

TalkID	単語	品詞	TalkType	SpeakerID	性別	出身地
A01F0055	発表	名詞	monolog	459	男	東京都
A01F0055	し	動詞	monolog	459	男	東京都
A01F0055	まし	助動詞	monolog	459	男	東京都
A01F0055	た	助動詞	monolog	459	男	東京都
D01F0023	報告	名詞	dialog	514	男	神奈川県
D01F0023	し	動詞	dialog	514	男	神奈川県
D01F0023	ます	助動詞	dialog	514	男	神奈川県

SQLクエリ初級編の復習

- SELECT文　列の選択

  テーブルから指定した列を選択して新しいテーブルを作成

- WHERE句　行の抽出

  テーブルから条件に合致した行を抽出して新しいテーブルを作成

- JOIN句　結合

  複数のテーブルを共通するキーにより関係づけて結合し，
  新しい一つのテーブルを作成

SELECT文　～列の選択～

- テーブルから指定した**列**を選択

テーブル名：infoTalk

TalkID	TalkType	SpeakerID
A01F0055	monolog	459
A02M0098	monolog	130
D01F0023	dialog	514
D01M0009	dialog	471

新しいテーブル

TalkID	TalkType
A01F0055	monolog
A02M0098	monolog
D01F0023	dialog
D01M0009	dialog

テーブルから「TalkID」と「TalkType」の列を選択

※SQLで書くと

```
SELECT infoTalk.TalkID, infoTalk.TalkType FROM infoTalk
```

選択する列名（テーブル名.列名の形式）　テーブル名

WHERE 句 ①　～行の抽出～

- テーブルから条件に合致した行を抽出

テーブル名：infoTalk

TalkID	TalkType	SpeakerID
A01F0055	monolog	459
A02M0098	monolog	130
D01F0023	dialog	514
D01M0009	dialog	471

新しいテーブル

TalkID	TalkType	SpeakerID
D01F0023	dialog	514
D01M0009	dialog	471

テーブルから、列「TalkType」が「dialog」である、という条件に合致した行を抽出

※SQLで書くと

```
SELECT infoTalk.*  FROM  infoTalk   WHERE  infoTalk.TalkType = “dialog”
```

抽出する列名（「*」は全ての列）　テーブル名　抽出条件（列「TalkType」が「dialog」である行）

WHERE 句 ② ～比較演算子～

- **基本操作**

```
SELECT 列名1,列名2 …
FROM   テーブル名
WHERE  条件式
```

- **条件式で用いる演算子1**　比較演算子

演算子	使用例	説明	補足
=	a = b	a と b は等しい	「==」でも可
<>	a <> b	a と b は等しくない	「!=」でも可
>	a > b	a は b より大きい	
>=	a >= b	a は b 以上	
<	a < b	a は b より小さい	
<=	a <= b	a は b 以下	

例）
```
WHERE   StartTime  =  60          # 開始時間が60秒に一致
WHERE   TalkID  =  “D01F0023”     # 談話IDが“D01F0023”に一致
WHERE   StartTime  >=  400        # 開始時間が400秒以降
```

WHERE 句 ④ ～論理演算子～

- **条件式で用いる演算子3** 論理演算子

演算子	使用例	説明
AND	p AND q	p と q が共に真の場合
OR	p OR q	p か q の少なくとも一つが真の場合
NOT	NOT p	p が真ではない（偽である）場合

pとqは条件式

例）WHERE TalkID = ″D01M0009″ AND OrthographicTranscription LIKE ″%です”

WHERE MoraEntity = “ア” OR MoraEntity = “ウ″

WHERE MoraEntity = “ア” OR MoraEntity = “ウ” OR MoraEntity = “オ″

INNER JOIN句

■ infoTalk

TalkID	TalkType	SpeakerID
A01F0055	monolog	459
A02M0098	monolog	130
D01M0009	dialog	471
R00M0187	monolog	423

対応させるキー

対応

■ infoSpeaker

SpeakerID	SpeakerSex	SpeakerBirthPlace
459	男	東京都
130	女	東京都
471	女	神奈川県
131	男	群馬県

■ 新しいテーブル

TalkID	TalkType	SpeakerID	SpeakerSex	SpeakerBirthPlace
A01F0055	monolog	459	男	東京都
A02M0098	monolog	130	女	東京都
D01M0009	dialog	471	女	神奈川県

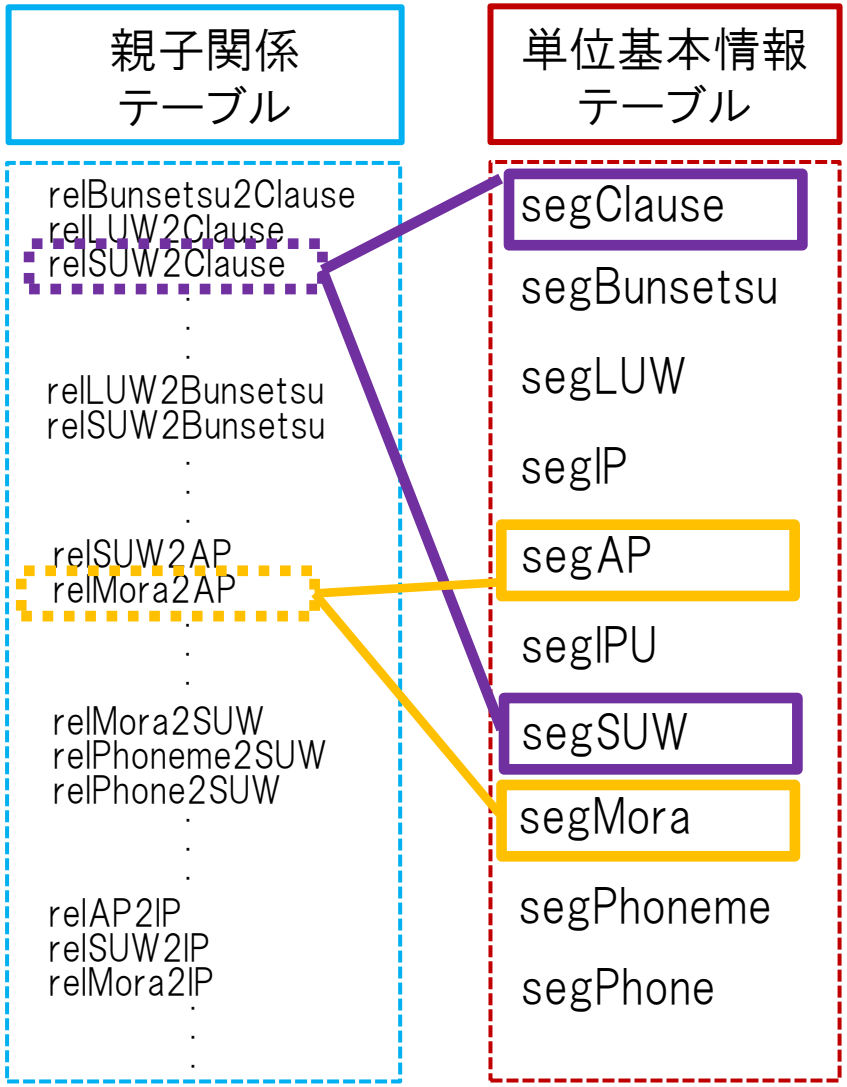
```
SELECT  infoTalk.*,  infoSpeaker.SpeakerSex,  infoSpeaker.SpeakerBirthPlace
FROM  infoTalk
INNER JOIN  infoSpeaker  ON  InfoTalk.SpeakerID  =  infoSpeaker.SpeakerID
```

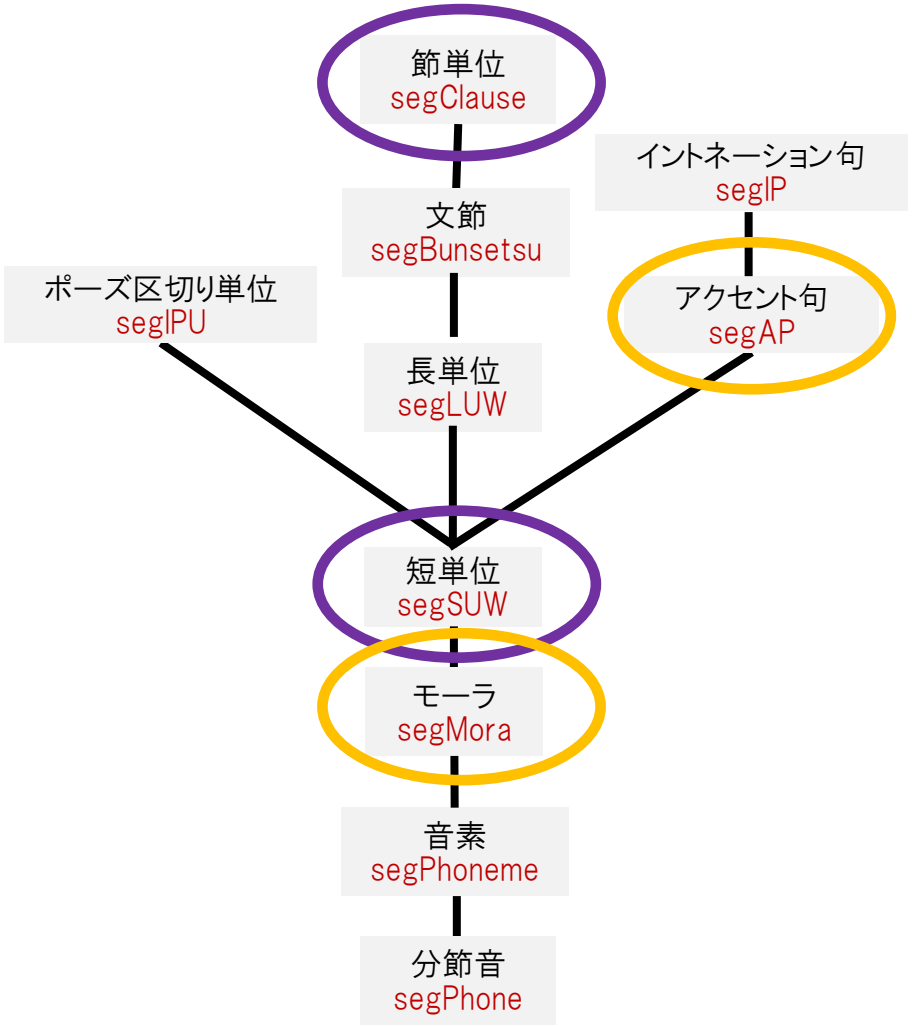
結合元のテーブル名

結合先のテーブル名

対応させるキー

CSJ-RDBの構成 ―親子関係テーブル―





親子関係テーブルの具体例

■ segLUW （子供のテーブル）

TalkID	LUWID	OrthographicTranscription
A01F0055	00053773L	いつ頃
A01F0055	00054186L	から
A01F0055	00054505L	可能
A01F0055	00054862L	な
A01F0055	00054980L	のでしょ
A01F0055	00055415L	う
A01F0055	00055478L	か

子供の単位ID

■ relLUW2Bunsetsu （親子関係テーブル）

親単位中の子単位の数と位置（何番目か）

TalkID	LUWID	BunsetsuID	nth	len
A01F0055	00053773L	00053773L	1	2
A01F0055	00054186L	00053773L	2	2
A01F0055	00054505L	00054505L	1	5
A01F0055	00054862L	00054505L	2	5
A01F0055	00054980L	00054505L	3	5
A01F0055	00055415L	00054505L	4	5
A01F0055	00055478L	00054505L	5	5

親の単位ID

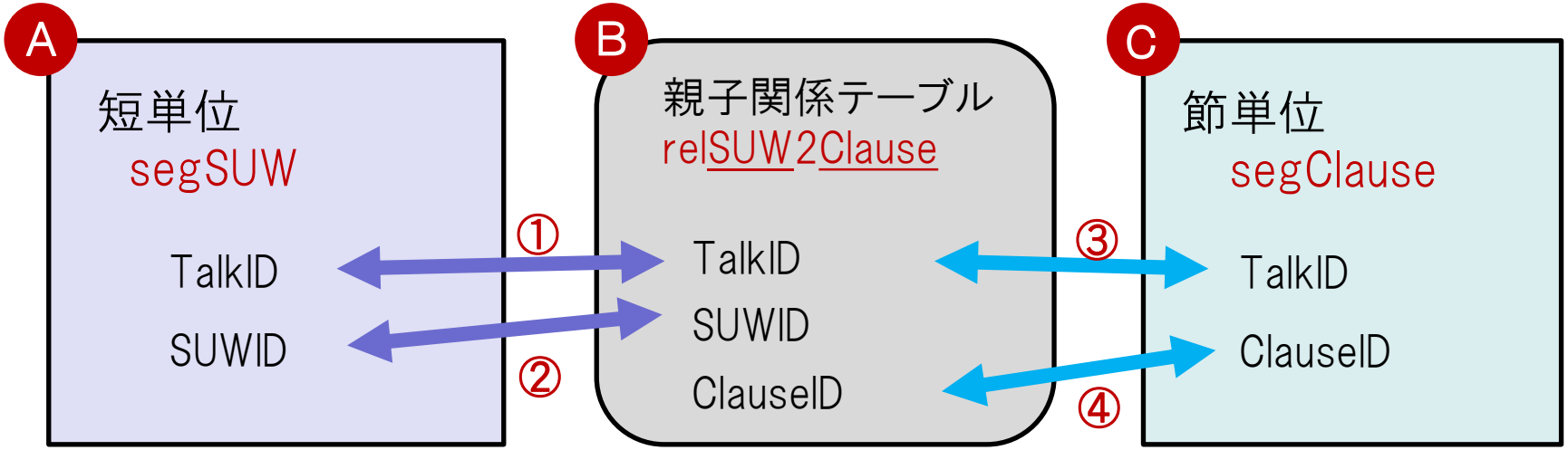
■ segBunsetsu （親のテーブル）

TalkID	BunsetsuID	OrthographicTranscription
A01F0055	00053773L	いつ頃から
A01F0055	00054505L	可能なのでしょうか

■ 内部結合後

TalkID	nth	len	LUWID	OrthographicTranscription	BunsetsuID	OrthographicTranscription
A01F0055	1	2	00053773L	いつ頃	00053773L	いつ頃から
A01F0055	2	2	00054186L	から	00053773L	いつ頃から
A01F0055	1	5	00054505L	可能	00054505L	可能なのでしょうか
A01F0055	2	5	00054862L	な	00054505L	可能なのでしょうか
A01F0055	3	5	00054980L	のでしょ	00054505L	可能なのでしょうか
A01F0055	4	5	00055415L	う	00054505L	可能なのでしょうか
A01F0055	5	5	00055478L	か	00054505L	可能なのでしょうか

SQL文　～短単位と節単位の結合の場合～

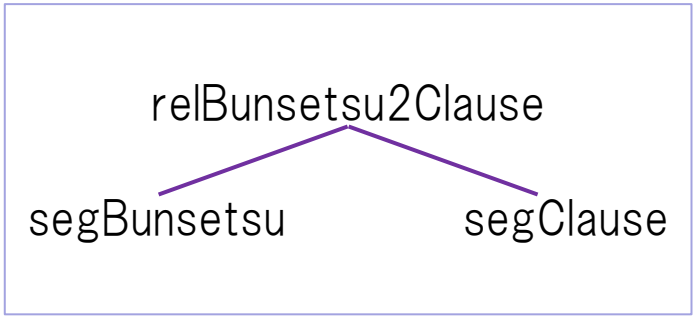



```
SELECT  segSUW.*,  segClause.*
FROM  segSUW  A
INNER JOIN  relSUW2Clause  B
  ON  segSUW.TalkID = relSUW2Clause.TalkID  AND        ①
      segSUW.SUWID = relSUW2Clause.SUWID               ②
INNER JOIN  segClause  C
  ON  relSUW2Clause.TalkID  =  segClause.TalkID  AND   ③
      relSUW2Clause.ClauseID  =  segClause.ClauseID    ④
```

例題1

- **例題1**　二つのセグメント･テーブルの結合

文節テーブル(segBunsetsu)と節単位テーブル(segClause)を結合し、
TalkID、文節の基本形(OrthographicTranscription)、
節単位の基本形(OrthographicTranscription)の列を選択



例題2

- **例題2**　二つのセグメント・テーブルの結合

文節テーブルと節単位テーブルを結合し、節単位末尾の文節を抽出し、
TalkID、文節の基本形、節単位の基本形の列を選択

ヒント：節単位の末尾の文節を指定するにはどのような条件を付ければよいか、
以下のテーブルを参考に考えてみましょう。

clauseID=00176804L中の
文節の位置（何番目か）

clauseID=00176804Lに
含まれる文節の数

TalkID	BunsetsuID	ClauseID	nth	len
A01F0055	00176804L	00176804L	1	6
A01F0055	00176882L	00176804L	2	6
A01F0055	00177374L	00176804L	3	6
A01F0055	00177569L	00176804L	4	6
A01F0055	00178510L	00176804L	5	6
A01F0055	00179521L	00176804L	6	6

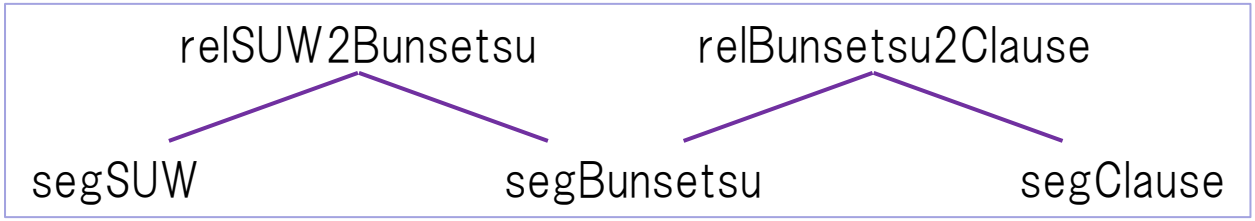
例題3～4

- **例題3　二つのセグメント・テーブルの結合**

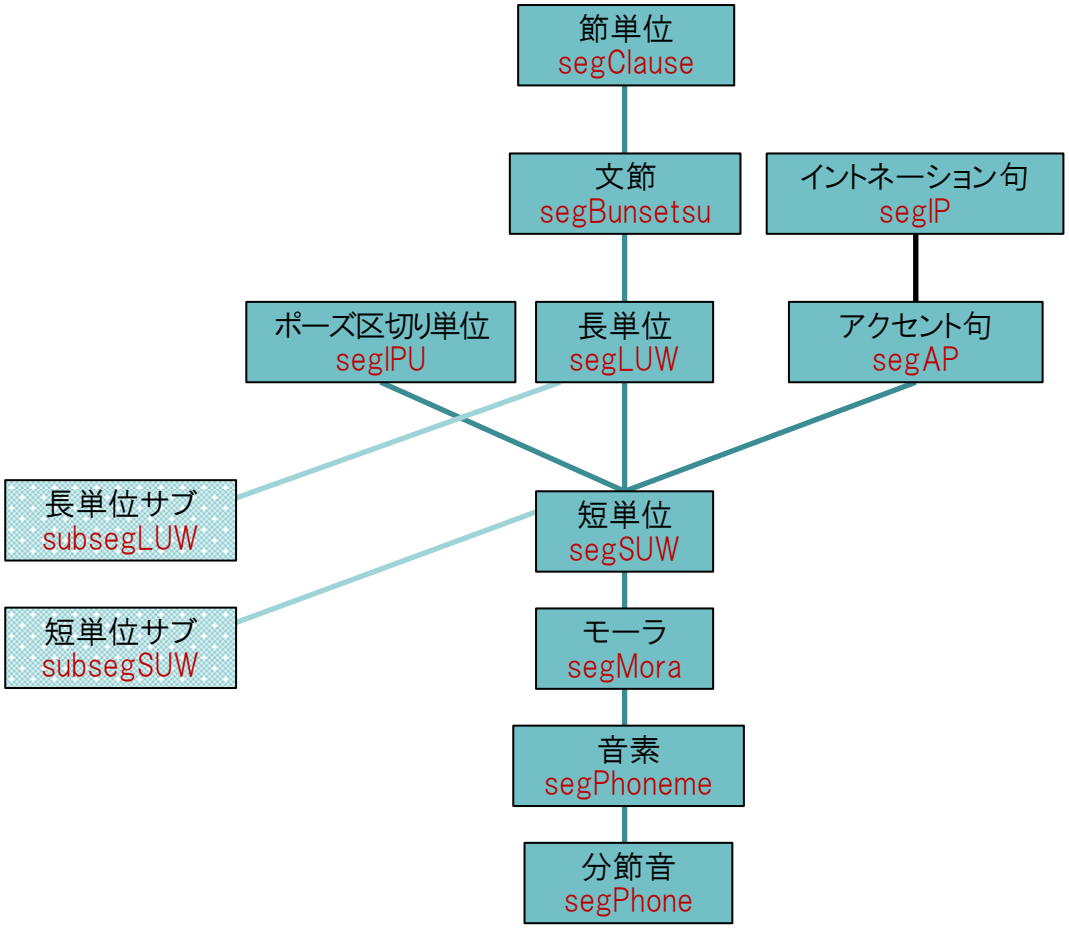
文節テーブルと節単位テーブルを結合し、節単位末尾の文節が「思」で始まる行を抽出し、TalkID、文節の基本形、節単位の基本形の列を選択

- **例題4　三つのセグメント・テーブルの結合**

短単位テーブル（segSUW）と文節テーブルと節単位テーブルを結合し、
節単位末尾の文節の先頭の短単位の基本形（OrthographicTranscription）が
「思」で始まる行を抽出した上で、
当該短単位の継続長を計算して”Duration”という別名を付け、
TalkID、短単位の基本形、文節の基本形、節単位の基本形と合わせて表示

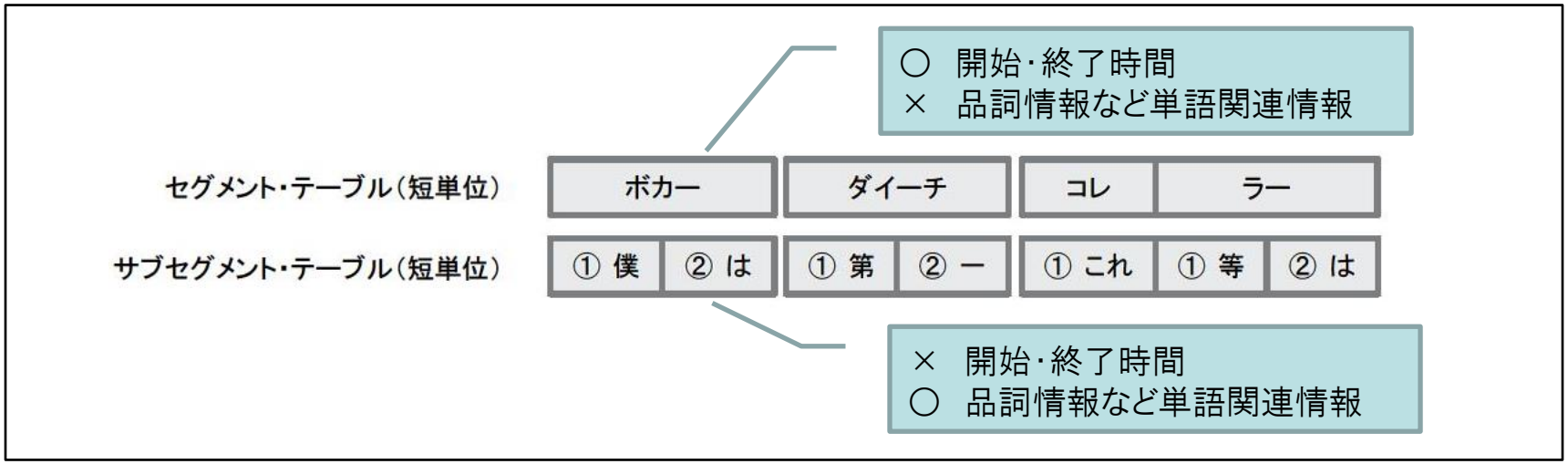


CSJ-RDBの構成 ーサブセグメント・テーブルー



サブセグメント・テーブル(短単位・長単位)

複数の語が融合して、分割できない一つの要素を形成する場合



■ セグメント・テーブル segSUW

TalkID	SUWID	StartTime	EndTime	OrthographicTranscription	word
A01F0122	00513758L	513.757614	514.114756	第ー	(W daici)
A01F0122	00514115L	514.114756	514.483693	母音	bo'iN

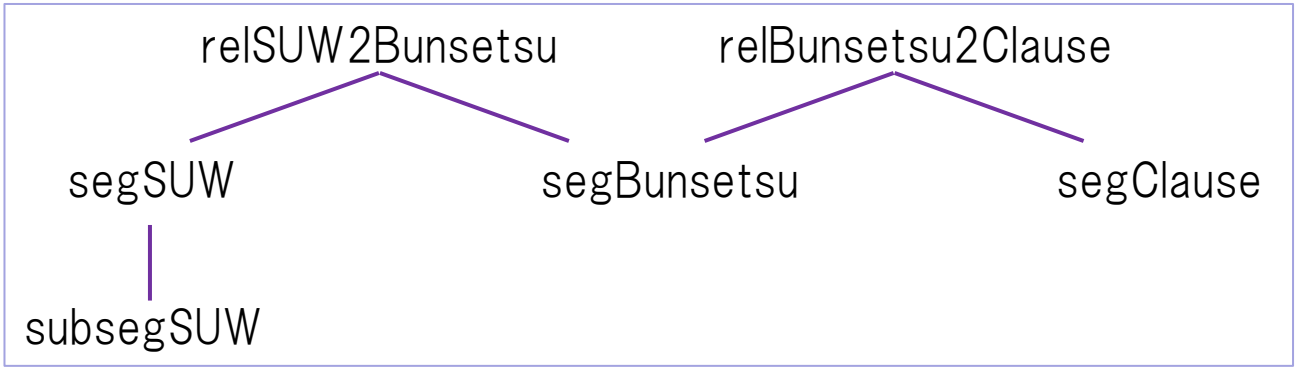
■ サブセグメント・テーブル subsegSUW （一部抜粋）

TalkID	SUWID	nth	len	PlainOrthographicTranscription	SUWLemma	SUWPOS
A01F0122	00513758L	1	2	第	第	接頭辞
A01F0122	00513758L	2	2	ー	ー	名詞
A01F0122	00514115L	1	1	母音	母音	名詞

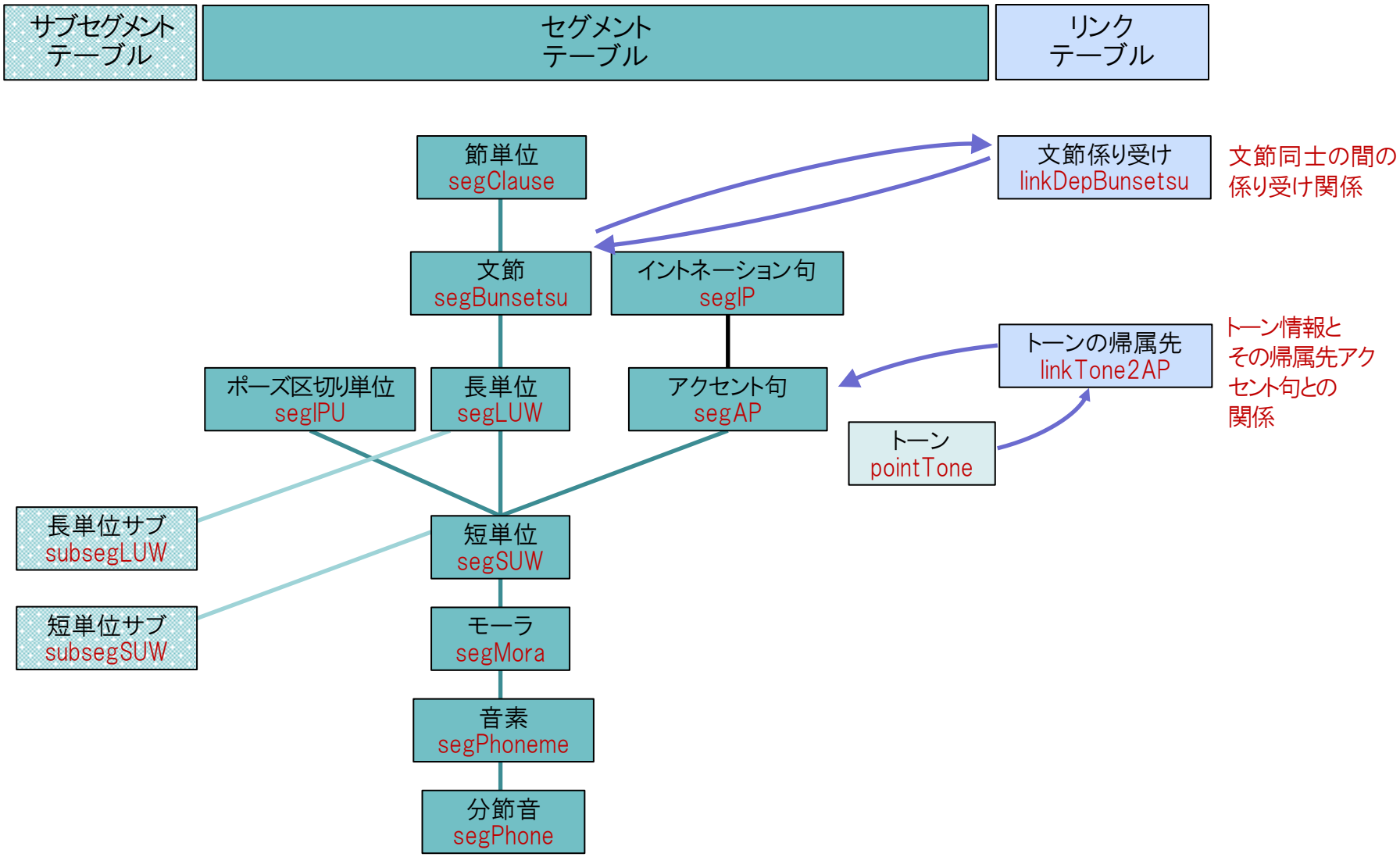
例題5

- **例題5**　セグメント・テーブルとサブセグメント・テーブルの結合

短単位テーブル、サブ短単位テーブル(subsegSUW)、文節テーブル、節単位テーブルを結合し、
節単位末尾の文節の先頭の短単位の品詞(SUWPOS)が「名詞」である行を抽出した上で、
当該短単位の継続長を計算して “Duration” という別名を付け、
TalkID、短単位の基本形、文節の基本形、節単位の基本形と合わせて表示



CSJ-RDBの構成 ーリンク・テーブルー



リンク・テーブル　linkDepBunsetsu

■ linkDepBunsetsu テーブル

TalkID	BunsetsuID	ModifieeBunsetsuID
A01F0055	00395174L	00395851L

係り元の文節　　係り先の文節

■ segBunsetsu テーブル

TalkID	BunsetsuID	StartTime	EndTime	OrthographicTranscription
A01F0055	00395174L	395.173616	395.850548	こちらだけが
A01F0055	00395851L	395.850548	396.642817	点滅します

リンク・テーブル　linkTone2AP

■ pointTone テーブル

TalkID	ToneID	Time	tone
A01F0055	28	11.77875	%L
A01F0055	29	11.89875	A
A01F0055	30	12.13375	L%
A01F0055	31	12.25875	H%

■ linkTone2AP テーブル

TalkID	ToneID	APID
A01F0055	28	00011679L
A01F0055	29	00011679L
A01F0055	30	00011679L
A01F0055	31	00011679L

帰属元のトーン　　帰属先のAP

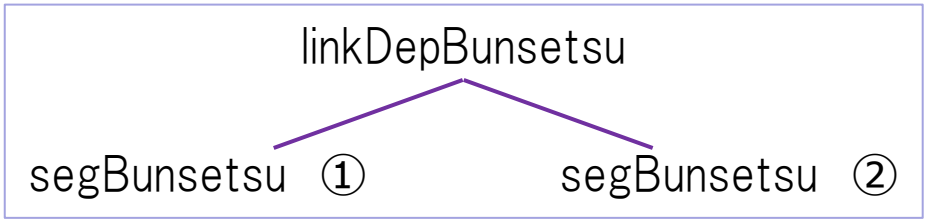
■ segAP テーブル

TalkID	APID	StartTime	EndTime	OrthographicTranscription
A01F0055	00011679L	11.67941	12.362888	聴取に

例題6

- **例題6**　係り元と係り先の文節を結合

linkDepBunsetsu を用いて係り元と係り先の文節テーブル(いずれもsegBunsetsu)を結合し、
係り元の文節の基本形が「が」で終わる行を抽出した上で、
TalkID、係り元と係り先の文節の基本形を表示



注意：segBunsetsu ① とsegBunsetsu ② のテーブル名が同じため問題が生じます。
各テーブルに別名(例えば “segB1”,“segB2”)を付けましょう。

第3部
SQLクエリ中級編

中級編で学ぶ内容

- JOIN句
  - 内部結合(中級編)
  - 外部結合
- GROUP BY句
- ORDER BY句
- HAVING句
- CASE式

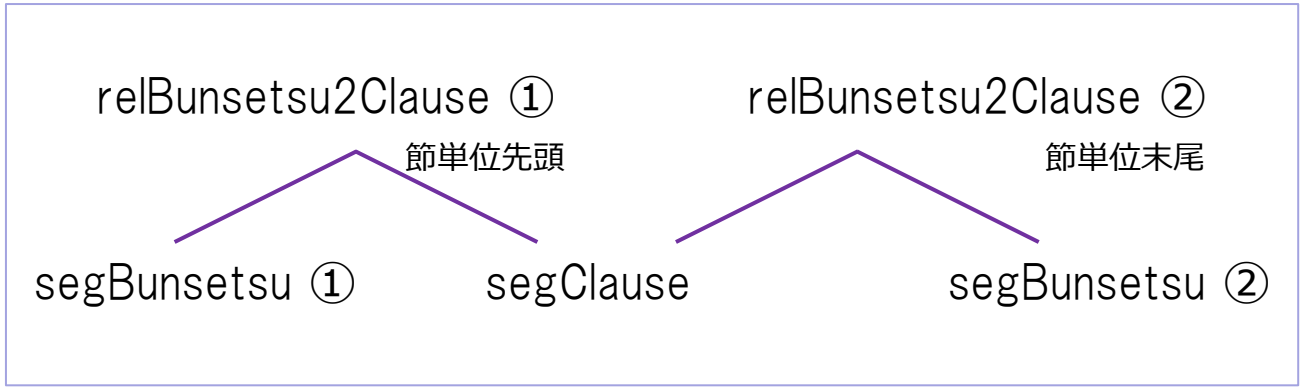
第3部　SQLクエリ中級編

① JOIN句 内部結合（中級編）

例題7

- **例題7**　同じセグメント・テーブルを結合

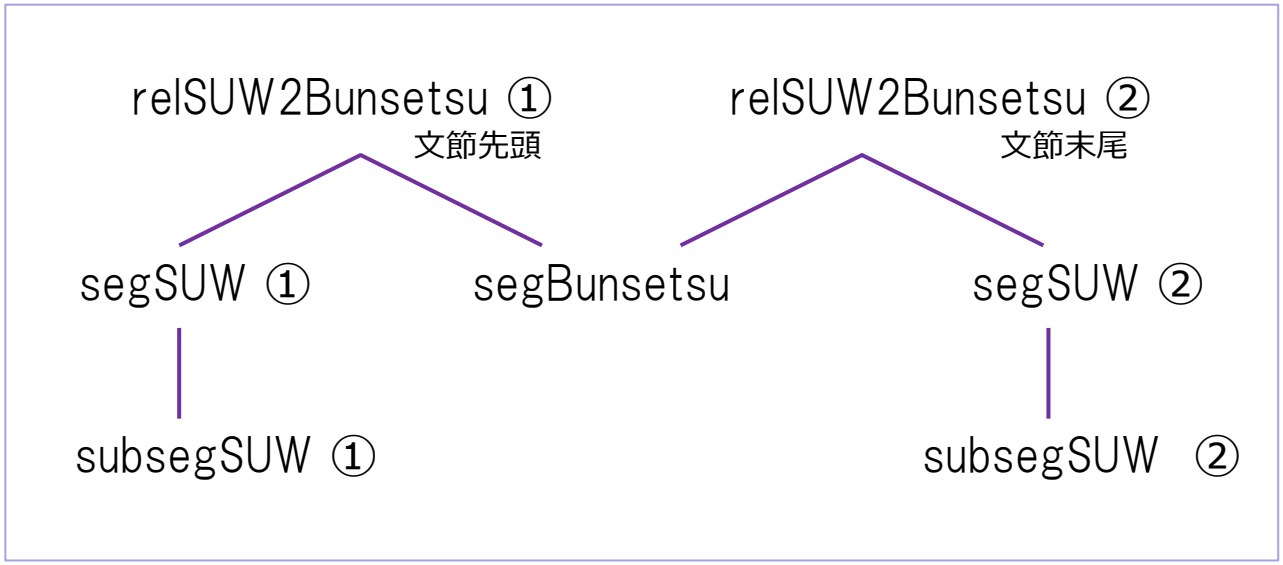
節単位の先頭の文節の基本形が「は」で終わり、かつ、
節単位の末尾の文節の基本形が「思」で始まる行を抽出し、
TalkID、節単位の基本形、節単位の先頭と末尾の文節の基本形を表示



例題8

- **例題8**　同じセグメント･テーブルを結合

文節の先頭の短単位の品詞が「名詞」であり、かつ、
文節の末尾の短単位の代表表記（SUWLemma）が「だ」である行を抽出し、
文節の基本形を表示



例題9

- **例題9**　同じセグメント・テーブルを結合

節単位内の隣接する二つの短単位の基本形の組合せを抽出し、TalkIDと合わせて表示

TalkID	SUWID	ClauseID	nth	len	Orthographic Transcription
D01F0023	00001006R	00001006R	1	5	よろしく
D01F0023	00001467R	00001006R	2	5	お
D01F0023	00001545R	00001006R	3	5	願い
D01F0023	00002012R	00001006R	4	5	し
D01F0023	00002173R	00001006R	5	5	ます

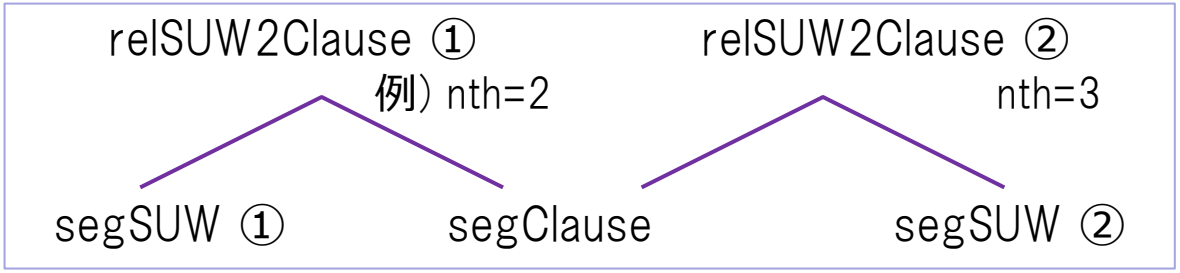
relSUW2Clause ①　　segSUW ①

Orthographic Transcription
よろしく
お
願い
し
ます

segSUW ②

TalkID	SUWID	ClauseID	nth	len
D01F0023	00001006R	00001006R	1	5
D01F0023	00001467R	00001006R	2	5
D01F0023	00001545R	00001006R	3	5
D01F0023	00002012R	00001006R	4	5
D01F0023	00002173R	00001006R	5	5

relSUW2Clause ②



※隣接する短単位はnthが1違う。
この条件をWHERE句で指定。

注意：segSUW ①とsegSUW ②のテーブルの名前が同じなので問題が生じます。
各テーブルに別名（例えば “segS1”, “segS2”）を付けましょう。
relSUW2Clause ①, ② も同様です（例えば “rel1”, “rel2” ）。

第3部　SQLクエリ中級編
② JOIN句 外部結合

内部結合　INNER JOIN　復習

- 対応する値がある行のみ結合

■ infoTalk

対応させるキー

TalkID	TalkType	SpeakerID
A01F0055	monolog	459
A02M0098	monolog	130
D01F0023	dialog	514
R00M0187	monolog	423

出力されず

対応

■ infoSpeaker

SpeakerID	SpeakerSex	SpeakerBirthPlace
459	男	東京都
130	女	東京都
514	男	神奈川県
131	男	群馬県

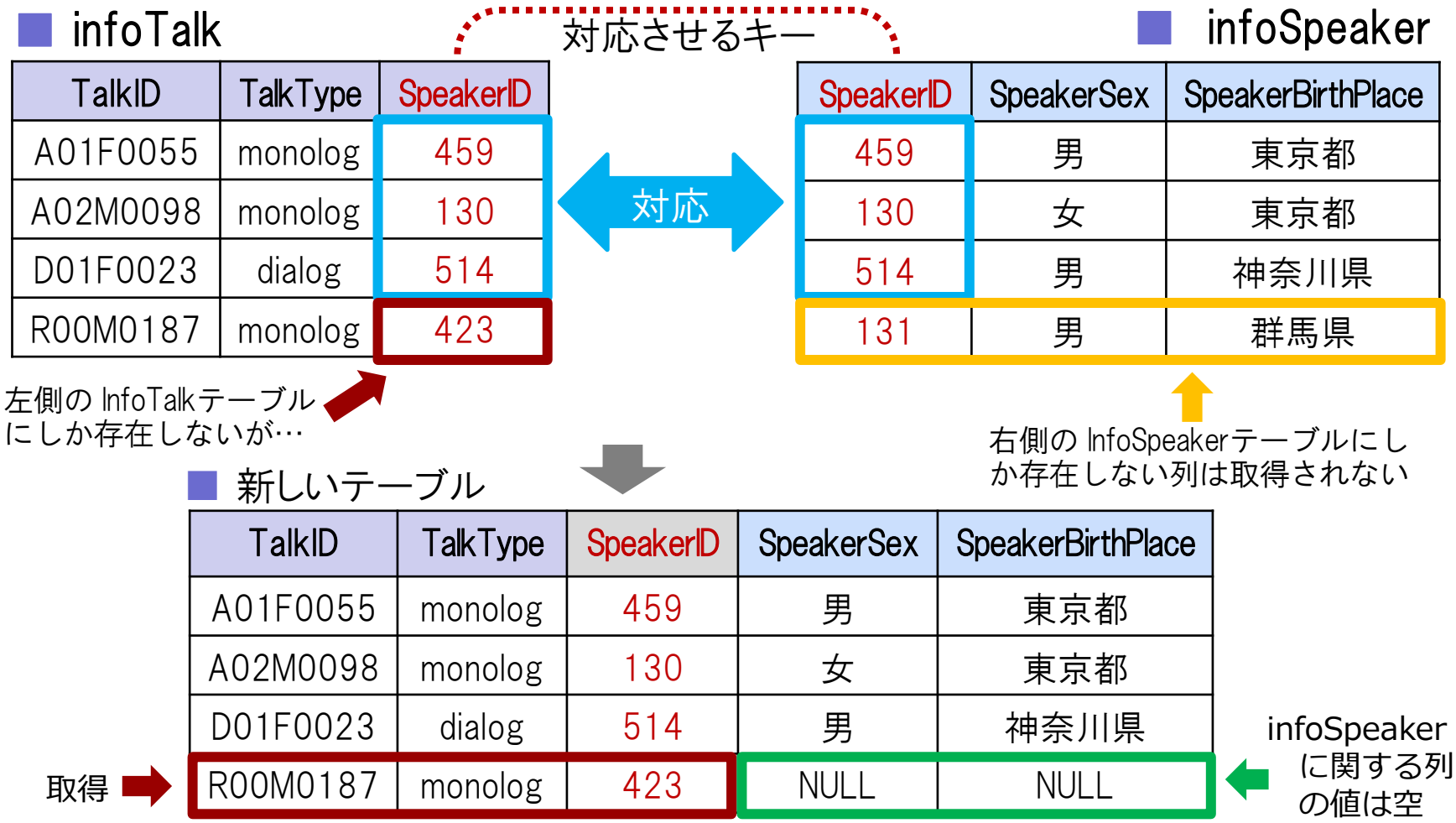
出力されず

■ 新しいテーブル

TalkID	TalkType	SpeakerID	SpeakerSex	SpeakerBirthPlace
A01F0055	monolog	459	男	東京都
A02M0098	monolog	130	女	東京都
D01F0023	dialog	514	男	神奈川県

外部結合 LEFT OUTER JOIN

対応する列＋左側のテーブルにのみ存在する列も取得



外部結合　LEFT OUTER JOIN

```
SELECT  infoTalk.*,  infoSpeaker.SpeakerSex,  infoSpeaker.SpeakerBirthPlace

FROM  infoTalk
```

結合元(左)のテーブル名

```
LEFT OUTER JOIN  infoSpeaker  ON  InfoTalk.SpeakerID  =  infoSpeaker.SpeakerID
```

結合先(右)のテーブル名　　対応させるキー

結合元

■ infoTalk

TalkID	TalkType	SpeakerID
A01F0055	monolog	459
A02M0098	monolog	130
D01F0023	dialog	514
R00M0187	monolog	423

対応させるキー

結合先

■ infoSpeaker

SpeakerID	SpeakerSex	SpeakerBirthPlace
459	男	東京都
130	女	東京都
514	男	神奈川県
131	男	群馬県

例題10

- **例題10** 外部結合

係り先を持たない文節を抽出し、TalkID、文節ID、文節の基本形を表示

segBunsetsu

TalkID	BunsetsuID	Orthographic Transcription
A01F0055	00007034L	(F えー)
A01F0055	00007208L	私共は
A01F0055	00008891L	乳児が
A01F0055	00009580L	音楽を
A01F0055	00009980L	どのように
A01F0055	00010420L	聞いているか
A01F0055	00011418L	また
A01F0055	00011679L	聴取に
A01F0055	00012363L	発達齢差が
A01F0055	00013335L	見られるか(0.134)を
A01F0055	00014414L	検討しております

linkDepBunsetsu

TalkID	BunsetsuID	Modifiee BunsetsuID
A01F0055	00007208L	00014414L
A01F0055	00008891L	00010420L
A01F0055	00009580L	00010420L
A01F0055	00009980L	00010420L
A01F0055	00010420L	00013335L
A01F0055	00011679L	00013335L
A01F0055	00012363L	00013335L
A01F0055	00013335L	00014414L

ヒント：文節テーブルに対して linkDepBunsetsu テーブルを外部結合し、
linkDepBunsetsu の列（例えばModifieeBunsetsuID ）が NULL である行を選択します。
値が NULL であるデータの検索には“IS NULL”演算子を用います
（例： ModifieeBunsetsuID IS NULL ）。

第3部　SQLクエリ中級編
③ GROUP BY句

GROUP BY句

グループ毎の集計に用いる

1. 指定された列(たとえば品詞)に含まれるグループ毎(たとえば、名詞、動詞、形容詞、…)にデータをまとめ上げる
2. 集計関数を用いてグループ毎に集計

subsegSUW

TalkID	PlainOrthographicTranscription	SUWPOS	SUWConjugateForm2
A01F0067	で	接続詞	
A01F0067	こちら	代名詞	
A01F0067	が	助詞	
A01F0067	時間	名詞	
A01F0067	情報	名詞	
A01F0067	のみ	助詞	
A01F0067	利用	名詞	
A01F0067	可能	形状詞	
A01F0067	な	助動詞	連体形
A01F0067	刺激	名詞	
A01F0067	で	助動詞	連用形
A01F0067	ある	動詞	終止形
A01F0067	と	助詞	
A01F0067	言え	動詞	連用形
A01F0067	ます	助動詞	終止形

SUWPOSで集計すると

SUWPOS	頻度
形状詞	1
助詞	3
助動詞	3
接続詞	1
代名詞	1
動詞	2
名詞	4

SUWPOSとSUWConjugateForm2で集計すると

SUWPOS	SUWConjugateForm2	頻度
形状詞		1
助詞		3
助動詞	終止形	1
助動詞	連体形	1
助動詞	連用形	1
接続詞		1
代名詞		1
動詞	終止形	1
動詞	連用形	1
名詞		4

GROUP BY句

行数(頻度)の取得

```
SELECT SUWPOS, COUNT(*) FROM subsegSUW GROUP BY SUWPOS
```

- SUWPOS, : GROUP BY で指定した列
- COUNT(*) : 集計関数
- SUWPOS : まとめ上げの列

TalkID	PlainOrthographicTranscription	SUWPOS	SUWConjugateForm2
A01F0067	で	接続詞	
A01F0067	こちら	代名詞	
A01F0067	が	助詞	
A01F0067	時間	名詞	
A01F0067	情報	名詞	
A01F0067	のみ	助詞	
A01F0067	利用	名詞	
A01F0067	可能	形状詞	
A01F0067	な	助動詞	連体形
A01F0067	刺激	名詞	
A01F0067	で	助動詞	連用形
A01F0067	ある	動詞	終止形
A01F0067	と	助詞	
A01F0067	言え	動詞	連用形
A01F0067	ます	助動詞	終止形

→

SUWPOS	COUNT(*)
形状詞	1
助詞	3
助動詞	3
接続詞	1
代名詞	1
動詞	2
名詞	4

GROUP BY句

行数(頻度)の取得

```
SELECT SUWPOS, SUWConjugateForm2, COUNT(*) AS 頻度 FROM subsegSUW
GROUP BY SUWPOS, SUWConjugateForm2
```

列名を「頻度」に

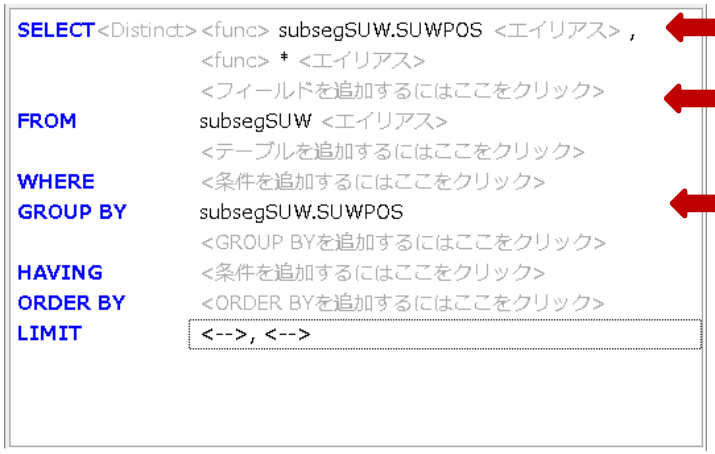
複数の列でまとめ上げ

TalkID	PlainOrthographicTranscription	SUWPOS	SUWConjugateForm2
A01F0067	で	接続詞	
A01F0067	こちら	代名詞	
A01F0067	が	助詞	
A01F0067	時間	名詞	
A01F0067	情報	名詞	
A01F0067	のみ	助詞	
A01F0067	利用	名詞	
A01F0067	可能	形状詞	
A01F0067	な	助動詞	連体形
A01F0067	刺激	名詞	
A01F0067	で	助動詞	連用形
A01F0067	ある	動詞	終止形
A01F0067	と	助詞	
A01F0067	言え	動詞	連用形
A01F0067	ます	助動詞	終止形

SUWPOS	SUWConjugateForm2	頻度
形状詞		1
助詞		3
助動詞	終止形	1
助動詞	連体形	1
助動詞	連用形	1
接続詞		1
代名詞		1
動詞	終止形	1
動詞	連用形	1
名詞		4

◇Navicat　GROUP BY

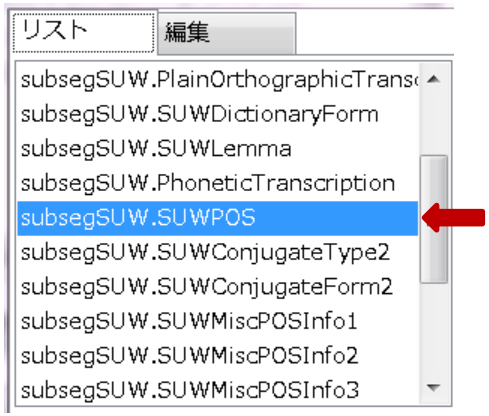
例）subsegSUW テーブルを対象に、品詞(SUWPOS)毎の頻度を求める



③ グループ化する列を追加

④「編集」で＊を記入した上で、<func>　を押して関数(Count) を選択

① ここをクリック



②列名リストが出現。
ここからグループ化する列名を選択。

※複数の列でグループ化する場合は
①②を繰り返す

GROUP BY句　集計関数

- GROUP BYとともに用いる主な集計関数

COUNT	行数を取得
SUM	合計値を取得
AVG	平均値を取得
MAX	最大値を取得
MIN	最小値を取得

例題11～13

- **例題11**　一つの列でまとめ上げ　＋　平均値

  モーラテーブル(segMora)を対象に、モーラ記号(MoraEntity)毎に、
  モーラ継続長の平均値を算出し、別名を“Avg”とした上で、
  モーラ記号と合わせて表示

- **例題12**　二つの列でまとめ上げ　＋　行数

  サブ短単位テーブルを対象に、品詞が「助詞」のものを抽出し、
  短単位の代表表記と品詞細分類(SUWMiscPOSInfo1)毎に頻度を算出した上で、
  別名を“Freq”とし、短単位の代表表記、品詞細分類と合わせて表示

- **例題13**　二つのセグメント・テーブルの結合　＋　一つの列でまとめ上げ

  短単位テーブル、サブ短単位テーブル、節単位テーブルを結合し、
  節単位の先頭の短単位を抽出した上で、短単位の品詞毎に頻度を算出し、
  別名を“Freq”として品詞と合わせて表示

第3部　SQLクエリ中級編
④ ORDER BY句

ORDER BY句

- 昇順に並び替え

例）

```
ORDER BY SUWPOS
ORDER BY COUNT(*)          … GROUP BY した時
ORDER BY TALKID, SUWPOS… 複数の列で
```

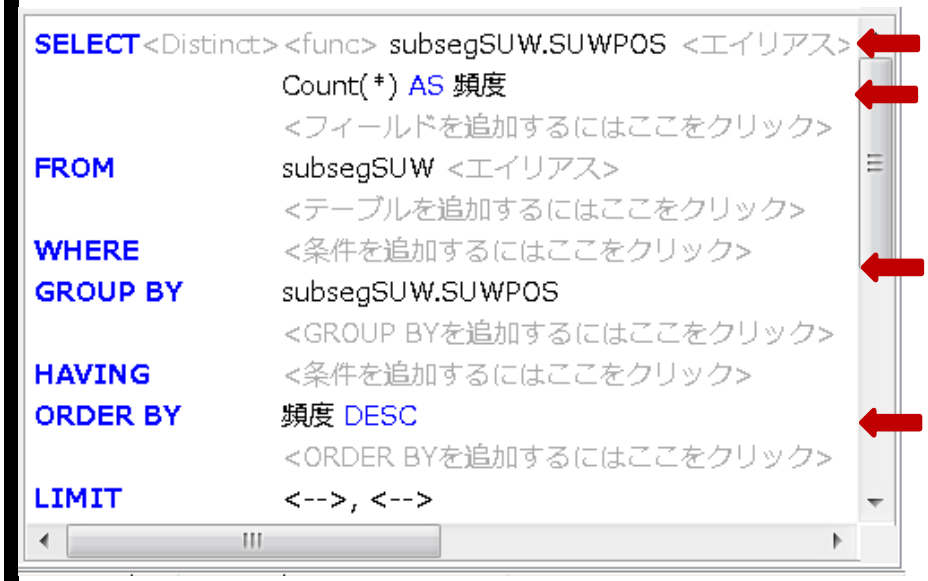
- 降順に並び替え　下記のように列名の最後に DESC を追加

例）

```
ORDER BY SUWPOS DESC
ORDER BY TALKID, SUWPOS DESC
```

◇Navicat　ORDER BY句

例）subsegSUW テーブルを対象に、品詞(SUWPOS)毎の頻度を求めたあと、
頻度の降順で並び替えて品詞と頻度を表示



② グループ化する列を追加

③「編集」で＊を記入した上で、<func>　を押して
関数(Count) を選択

① ここをクリックしてグループ化する列名を選択

④ ここをクリックして並び替えの対象とする列を追加
右横の「ASC」をクリックして「DESC」を選択

例題14

- **例題14**　二つの列でまとめ上げ　＋　行数　＋　並び替え

例題13の結果を、頻度(“Freq”)の降順で並び替えて表示

第3部　SQLクエリ中級編
⑤ HAVING句

HAVING句

- グループ毎のまとめ上げの後に、グループに対して条件で絞り込み

グループで
まとめ上げ
(GROUP BY)

グループに対し
条件で絞り込み
(HAVING)

※例えば件数が**1000以上**

TalkID	PlainOrthographic Transcription	SUWPOS
…	…	…
A01F0067	こちら	代名詞
A01F0067	が	助詞
A01F0067	時間	名詞
A01F0067	情報	名詞
A01F0067	のみ	助詞
A01F0067	利用	名詞
A01F0067	可能	形状詞
A01F0067	な	助動詞
A01F0067	刺激	名詞
A01F0067	で	助動詞
A01F0067	ある	動詞
A01F0067	と	助詞
A01F0067	言え	動詞
A01F0067	ます	助動詞
…	…	…

SUWPOS	頻度
代名詞	809
副詞	1516
助動詞	4381
助詞	10686
動詞	4298
名詞	6812
形容詞	493
形状詞	571
感動詞	3435
接尾辞	819
接続詞	405
接頭辞	173
言いよどみ	608
記号	28
連体詞	362

SUWPOS	頻度
副詞	1516
助動詞	4381
助詞	10686
動詞	4298
名詞	6812
感動詞	3435

※SQLで書くと

```
SELECT  SUWPOS,
        COUNT(*)  AS Freq
FROM  subsegSUW
GROUP BY  SUWPOS
HAVING  Freq  >=  1000
```

SQLの構文

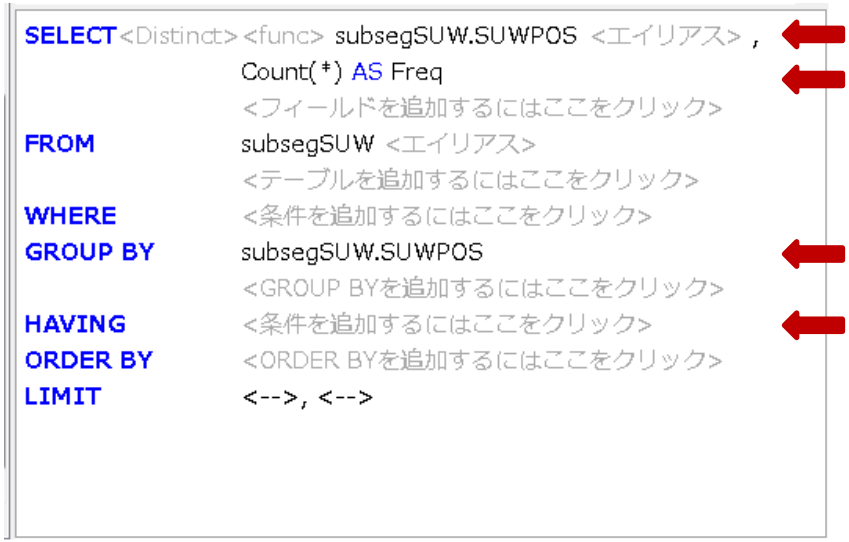
- SQLの構文

必須	SELECT	列名	
必須	FROM	テーブル名	
オプション	INNER JOIN	テーブル名2	… or LEFT JOIN ※
オプション	WHERE	抽出条件	
オプション	GROUP BY	グループ列名	
オプション	HAVING	グループに対する抽出条件	
オプション	ORDER BY	整列対象とする列名	

※　JOIN は複数回繰り返すことも可

◇Navicat　HAVING

例）subsegSUW テーブルを対象に、品詞（SUWPOS）毎の頻度を求めたあと、
頻度1000以上のものに限定して表示



② グループ化する列を追加
③「編集」で ＊ を記入した上で、<func>　を押して
関数(count) を選択し、別名“Freq”を付ける

① ここをクリックしてグループ化する列名を選択

④ ここをクリックして HAVING の条件を追加
（基本的にWHERE句の場合と同様）

例題15～16

- **例題15**　二つの列でまとめ上げ　＋　行数　＋ HAVING句

サブ短単位テーブルを対象に、短単位の代表表記と品詞毎に頻度を算出し、
別名を “Freq” とした上で、頻度が500以上のものに絞り込み、
短単位の代表表記、品詞、頻度を、頻度の降順で表示

- **例題16**　一つの列でまとめ上げ ＋　行数　＋　HAVING句

サブ短単位テーブルを対象に、品詞細分類が「格助詞」のものを抽出し、
短単位の代表表記毎に頻度を算出した上で、別名を ”Freq” とし、
頻度が500以上のものに絞り込んだ上で、
短単位の代表表記と頻度を、頻度の降順で表示

ヒント：WHERE句とHAVING句をともに用います。両者の違いに注意しましょう。

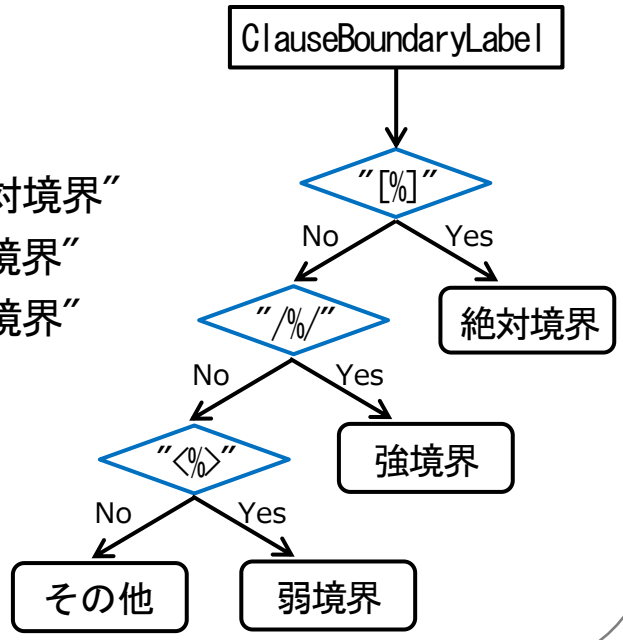
第3部　SQLクエリ中級編
⑥ CASE式

CASE式

- 条件式によって異なる値を返す

例） SELECT 文でのCASE式の利用

```
SELECT TalkID, ClauseID, ClauseBoundaryLabel,
  CASE
    WHEN ClauseBoundaryLabel LIKE "[%]" THEN "絶対境界"
    WHEN ClauseBoundaryLabel LIKE "/%/" THEN "強境界"
    WHEN ClauseBoundaryLabel LIKE "<%>" THEN "弱境界"
    ELSE "その他"
  END AS ClauseType    ← 列に別名を付与
FROM segClause
```



※ ClauseBoundaryLabel の例：“[文末]” “/並列節ガ/” “<理由節ノデ>”

CASE式を利用した集計

- 条件式によって得られる値毎にまとめ上げ

例） SELECT 文と GROUP BY句でのCASE式の利用

```
SELECT
    CASE
        WHEN  ClauseBoundaryLabel  LIKE  "[%]" THEN  "絶対境界"
        WHEN  ClauseBoundaryLabel  LIKE  "/%/" THEN  "強境界"
        WHEN  ClauseBoundaryLabel  LIKE  "<%>" THEN  "弱境界"
        ELSE  "その他"
    END AS  ClauseType,   ← 列に別名を付与
    COUNT(*) AS  "Freq"
 FROM  segClause
 GROUP BY ClauseType     ← 条件式（の別名）でグループ化することもできる
```

例題17

- **例題17**　SELECT文 と GROUP BY句 で CASE式

サブ短単位テーブルを対象に、短単位を「活用語」と「非活用語」に分け、
SUWType とした上で、SUWType毎の頻度を算出し、
別名を “Freq” としてSUWType と合わせて表示

ヒント：非活用語は活用形が空（SUWConjugateForm2 = “”）になります

例題18

- **例題18**　SELECT文 と GROUP BY句 でCASE式

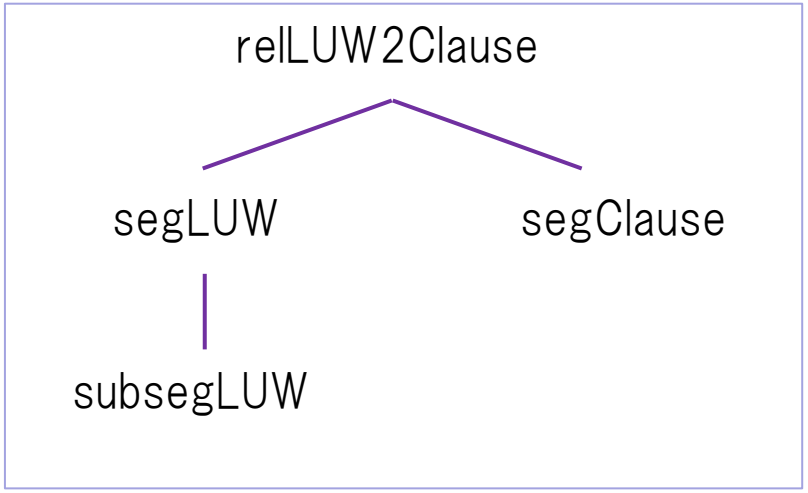
モーラテーブルを対象に、モーラ記号を「特殊拍」（「ン」「ッ」「ー」）と
それ以外の「一般拍」に分け、MoraType とした上で、
MoraType 毎の継続長の平均値を算出し、別名を“Duration”としてMoraTypeと合わせて表示

ヒント：「x = “a” OR x = “b”」 は「x IN （“a”, ”b”）」 と簡潔に書くことができます

例題19

- **例題19**　SELECT文 と GROUP BY句 でCASE式

節単位の先頭の長単位(subsegLUW)の品詞(LUWPOS)が
「接続詞」(conj)か否か(non-conj)に分け、 CUType とした上で、
CUType 毎に節単位に含まれる長単位数の平均値、最大値、最小値を算出し、
別名を “Avg”, “Max”, “Min” として CUType と合わせて表示



第3部　SQLクエリ中級編
練習問題

課題1

- **課題1**

アクセント句（segAP）の末尾の短単位が終助詞「ね」である行を抽出し、
アクセント句の基本形（OrthographicTranscription）と句末音調（fbt）を表示

課題2

- **課題2**

課題1の終助詞「ね」を、節単位の末尾(final)とそれ以外(non-final)に分類し、Position とした上で、Position 毎、句末音調毎の頻度を表示

ヒント:アクセント句と節単位は階層関係にないため、relSUW2Clause を用います。
分類には CASE式を、集計には GROUP BY句を用います。

課題3

- **課題3**

品詞細分類(LUWMiscPOSInfo1)が「格助詞」である長単位を含む文節を抽出し、
文節の基本形と格助詞の代表表記(LUWLemma)を表示

課題4

- **課題4**

二格の文節（品詞細分類が「格助詞」であり代表表記が「に」である長単位を含む文節）と、
その文節が係る先の文節を抽出し、係り元と係り先の文節の基本形を表示

課題5

- **課題5**

二格の文節とヲ格の文節が、同じ文節に係るものを抽出し、
係り元の文節の基本形を二格･ヲ格の順に、係り先の文節の基本形と合わせて表示

ヒント：**課題4**のような結合を二つ（一方は二格、他方はヲ格）作ります。
「同じ文節に係る」という条件なので、両者の係り先の文節を共有する形にしましょう。

課題6

- **課題6**

課題5の結果を、二格・ヲ格の順ではなく、実際の文節の出現順に表示

ヒント: **課題5**のSELECTの記述を少し変更します。
二つの係り元文節の出現順(どちらがより早いタイミングで出現するか)に応じて、
二格・ヲ格の順に表示するか、ヲ格・二格の順に表示するかを決めます。
CASE式を2回使います。

課題7

- **課題7**

課題6を、二格、ヲ格の語順で出現する場合と、
ヲ格、二格の語順で出現する場合に分け、それぞれの頻度を表示

ヒント：**課題6**のCASE式を少し変更します。
集計にはGROUP BY句を用います。