WPS による各種データベー ス入出力方法

(WPS バージョン 3.1.1)

第1版

2015 年 1 月

株式会社ブレインパッド

目次

1. はじめに	3
[WPS がサポートしているデータベース]	4
2. DB2 とのインタフェース	5
[SAMPLE サンプル DB]	5
2.1. ネイティブドライバー経由の DB テーブルの入出力	5
[SQL 言語を使用した DB 読み取り]	6
[libname ステートメントによる DB 入力]	9
[WPS データセットの DB への出力方法]	13
2.2. ODBC ドライバー経由の DB テーブルの入出力	15
[ODBC ドライバーのインストール]	15
[PC 側の ODBC ドライバーの設定]	15
[SQL 言語を使用した DB 読み取り]	17
[libname ステートメントによる DB 入力]	19
[WPS データセットの DB への出力方法]	22
3. MySQL とのインタフェース	24
[sakila サンプル DB]	24
3.1. ネイティブドライバー経由の DB テーブルの入出力	24
[ネイティブドライバのインストールと Path の定義]	25
[SQL 言語を使用した DB 読み取り]	26
[libname ステートメントによる DB 入力]	29
[WPS データセットの DB への出力方法]	33
3.2. ODBC ドライバー経由の DB テーブルの入出力	34
[ODBC ドライバのインストール]	35
[PC 側の ODBC ドライバーの設定]	35
[SQL 言語を使用した DB 読み取り]	36
[libname ステートメントによる DB 入力]	39
[WPS データセットの DB への出力方法]	42
4. ORACLE とのインタフェース	43
[scott サンプルスキーマ]	43
4.1. ネイティブドライバー経由の DB テーブルの入出力	44
[SQL 言語を使用した DB 読み取り]	44
[libname ステートメントによる DB 入力]	47
[WPS データセットの DB への出力方法]	50
4.2. ODBC ドライバー経由の DB テーブルの入出力	52

[ODBC ドライバのインストール]	52
[PC 側の ODBC ドライバーの設定]	52
[SQL 言語を使用した DB 読み取り]	54
[libname ステートメントによる DB 入力]	57
[WPS データセットの DB への出力方法]	59
5. PostregSQL とのインタフェース	61
[dvdrental サンプル DB]	61
5.1. ネイティブドライバー経由の DB テーブルの入出力	62
[SQL 言語を使用した DB 読み取り]	62
[libname ステートメントによる DB 入力]	65
[WPS データセットの DB への出力方法]	69
5.2. ODBC ドライバー経由の DB テーブルの入出力	71
[ODBC ドライバのインストール]	71
[PC 側の ODBC ドライバーの設定]	71
[SQL 言語を使用した DB 読み取り]	73
[libname ステートメントによる DB 入力]	75
[WPS データセットの DB への出力方法]	78
6. SQL Server とのインタフェース	80
[AdventureWorks2012 サンプル DB]	80
6.1. ネイティブドライバー経由の DB テーブルの入出力	81
[SQL 言語を使用した DB 読み取り]	81
[libname ステートメントによる DB 入力]	84
[WPS データセットの DB への出力方法]	88
6.2. ODBC ドライバー経由の DB テーブルの入出力	90
[ODBC ドライバのインストール]	90
[PC 側の ODBC ドライバーの設定]	91
[SQL 言語を使用した DB 読み取り]	93
[libname ステートメントによる DB 入力]	96
[WPS データセットの DB への出力方法]	99

1. はじめに

WPS は DB2, MySQL, Netezza, Oracle, PostgreSQL, SQL Server, Teradata などの商用、ま たはオープンソースのデータベース (以下 DB とも表記) に対するアクセスインターフェー ス機能をサポートしています。インターフェース機能に含まれるのは、データの読み書き、 テーブルの作成・編集・削除を含みます。サポートするインタフェース方式は、DB ごとに 固有のコマンドライン接続方式 (CLI ドライバー、ネイティブドライバー) に加えて、多く の DB に対してそれぞれの DB ベンダーから提供されているオープンデータベース接続方 式 (ODBC ドライバー) の両方です。

本資料は 64 ビット版 Windows8.1 上で動作する以下のデータベースプロダクトをローカル 環境にインストールし、各プロダクトに付随しているサンプルデータベースを例として CLI ドライバー経由および ODBC ドライバ経由で WPS3.1.1 からそれぞれの DB テーブルの入 出力を行う場合を例示し、WPS の DB インタフェース機能の使い方を説明したものです。

DB2 Express-C 10.5 (SAMPLE データベース) MySQL 5.6 (sakila データベース (スキーマ)) ORACLE Express Edition 11.2 (scott スキーマ) PostgreSQL 9.3.5 (dvdrental データベース) SQL Server 2012 Express Edition (AdventureWorks2012 データベース)

なお、本資料では、主に SHIFT-JIS エンコード(MS932 エンコード)の日本語データを取扱っています。そのため、WPS ワークベンチの設定が、以下のとおり設定されていることを前提としています。

- ・起動オプションに encoding="SHIFT-JIS" 1
- ・ワークスペースのテキストファイルエンコードに MS932²

^{1 「}WPS サーバーエクスプローラー」ビューの「ローカルサーバー」を右クリック→「プロパティ(R)」をクリック → 「プロパティ:ローカルサーバー」画面の左領域の「起動オプション」をクリック → 右領域の「追加」をクリッ ク → 「起動オプション」画面出現 →「選択」をクリック → 出現した「起動オプションを選択」画面のオプション リストの中から「ENCODING」を探してクリック → 「OK」をクリック → 「起動オプション」画面の「名前:」欄 に ENCODING が入っていることを確認 → 「値:」欄に SHIFT-JIS を入力 → 「OK」。なお、バージョン 3.1.1 か ら LOCALE="JAPANESE_JAPAN"を指定できるようになり、この指定を行うと、ENCODING="SHIFT-JIS" に自動設定 されるようになりました。

^{2 「}メニューバー」の「ウィンドウ(W)」→ 「設定(P)」→ 左領域の「一般」→ 「ワークスペース」をクリック → 右領域の左下部にある「テキスト・ファイル・エンコード(T)」の「その他(O)」ラジオボタンをクリック → 入力欄に MS932 とキー入力 → 「OK」。なお、バージョン 3.1.1 では既定の設定になりました。

プログラムをバッチ実行する場合は、-encoding SHIFT-JIS (起動オプション)を指定して 実行することが必要です。

[WPS がサポートしているデータベース]

WPS3.1.1 でアクセスインタフェースがサポートされている DB とそのインターフェース機能の種類とアクセスの種類は以下のとおりです。

			(1	ノツノコ	- 一 へ 你	式月日ノ		
データベース	読み取 り	書き込 み	更新	新規 テーブ ルの作 成	パスス ルー (暗黙)	パスス ルー (明示)	バルクロード	バルクアッ プロード
Action Matrix	0	0	0	0	0	0	×	×
DB2	0	•	0	•	0	0	● (z/OS以外)	×
Greenplum	0	•	0	0	0	0		×
Informix	•	•	0	•	0	0	×	×
Kognitio	0		×		0	0	×	×
MySQL	0	0	0	0	0	0	×	×
Netezza	•		0		0	0		0
ODBC	0	0	0	•	0	0	(SQL Serverのみ)	×
OLEDB	0	0	0	0	0	0	×	×
Oracle	•	0	0		0	0		×
PostgreSQL	0	0	0	•	0	0		×
SAND	•	0	0	0	0	0		×
SQL Server	•	0	0		0	0		×
Sybase	•		0		0	0	×	×
Teradata		0	0	0	0	0	×	×
Vertica		•	•		0	0	×	×

(インタフェース機能)

OS 別サポート状況は以下のとおりです。

			(0					
データベース	AIX pSeries(P ower)	Linux x86	Mac OS X	Solaris SPARC	Solaris x86	Windows x86	Linux System z	z/OS System z
Action Matrix	×	0	×	×	×	0	×	×
DB2	0	0	×	0	0	0	0	0
Greenplum	0	0	0	×	0	•	×	×
Informix	×	×	×	×	×	•	×	×
Kognitio	×	0	×	×	0	•	×	×
MySQL			0		•	0	0	×
Netezza			0	0	•	0	×	×
ODBC			0	0	•	0	×	×
OLEDB	×	×	×	×	×	0	×	×
Oracle	0	0	0	0	0	0	0	×
PostgreSQL	×	0	0	×	0	•	0	×
SAND	0	0	×	0	0	×	×	×
SQL Server	×	×	×	×	×	•	×	×
Sybase	0	0	×		0		×	×
Teradata		0	×				×	
Vertica		0	×	×			×	×

(OS環境)

※ ODBC 接続については OS や対象 DB によってサードパーティ製品の導入が必要になる 場合もあります。

なお、ここに示した例は、ローカル環境にインストールした DB に管理者ユーザーでログイ ンできる環境で実行しています。ネットワーク環境や、セキュリティ環境の違いにより、こ こで示したプログラムや設定では DB アクセスが実行できない場合もあります。その場合 は、DB アクセス環境等を確認の上、必要な修正を加えて実行してください。

2. DB2 とのインタフェース

[SAMPLE サンプル DB]

DB2 Express-C 10.5 に付随する sample データベースを例として用います。

9			データベース管理 -	DB2ADMIN.STA	FF - IBM Data Stud	lio		
ファイル(F) 編集(E) ナビゲート(N) 検索(A)	プロジェ	クト(P) Data 実行	テ(R) ウィンドウ(W) へルプ(H)					
📬 • 🖩 🗞 🛆 💽 🚳 👘 • 🔒 7	ウティビテ	ィー: データペースの智	理 - 🖪 📳 🗟 💁 - 🔺	7 - [出 - 第 - 休	$\phi \star \phi \star \phi$			ウ・アクセス 📑 🚺
◎ 管理エクスプローラー □ □	國 刻	スク・ランチャー	🗉 SAMPLE 🛛 🗖 DB2ADMINS	TAFF 22				
E 🔮 💕 🖬 🗢 🗐 🐃 🖄 🌣	DE	32ADMIN.STA	FF					∌ ♦
a 🗄 localhost		ID [SMALLINT]	NAME [VARCHAR(9 OCTETS)]	DEPT [SMALLINT]	JOB [CHAR(5 OCTETS)]	YEARS [SMALLINT]	SALARY [DECIMAL(7, 2)]	COMM [DECIMAL(7,2)] ^
4 😤 DB2	1	10	Sanders	20	Mor	7	98357.50	
a 🏭 SAMPLE (DB2 for Linux	2	20	Pernal	20	Sales	8	78171.25	612.45
ご 変更プラン		30	Marenghi	38	Mar	5	77506.75	
🗀 🕿	4	40	O'Brien	38	Sales	6	78006.00	846.55
🗀 Ez-	5	50	Hanes	15	Mar	10	80659.80	
二 第引	6	60	Quigley	38	Sales		66808.30	650.25
(二) 制約		70	Rothman	15	Sales	7	76502.83	1152.00
Ε-1/1/2	8	80	James	20	Clerk		43504.60	128.20
MQI	9	90	Koonitz	42	Sales	6	38001.75	1386.70
	10	100	Plotz	42	Mar	7	78352.80	
- 7t-7		110	Ngan	15	Clerk	5	42508.20	206.60
		120	Naughton	38	Clerk		42954.75	180.00
>>====================================		130	Yamaguchi	42	Clerk	6	40505.90	75.60
○ 表2パー2	14	140	Frave	51	Mar	6	91150.00	
(1) 1(y7p7-1)		150	Williams	51	Sales	6	79456.50	637.65
パーディション・グループ	16	160	Molinare	10	Mar	7	82959.20	
> 🇀 詳細なアクセス制御		170	Kermisch	15	Clerk	4	42258.50	110.10
D アプリケーション・オブジェ・	18	180	Abrahams	38	Clerk	3	37009.75	236.50
▶ 🗀 ユーザーとグループ	19	190	Sneider	20	Clerk	8	34252.75	126.50
> 🇀 フェデレーテッド・データベ		200	Scoutten	42	Clerk		41508.60	84.20
🗀 XML 2+-7		<						
🗁 作業セット		·	000 011015				Torrett of AL	
	•	ER: localhost -	DB2 - SAMPLE				9へ(表示 35 行 🧕	ロレの結果とユーのオブションの変更
		🗆 17-1- 💷 SOL	の結果					• • • • • • • •
	70/	7-	傳					
< >								

2.1. ネイティブドライバー経由の DB テーブルの入出力

まず、コマンドラインインタフェース(CLIドライバー、ネイティブドライバー)を用いて WPS から DB テーブルの入出力を行ってみましょう。

ネイティブドライバーは DB と一緒にインストールされており、一般に ODBC ドライバー より高速に DB アクセスができる点でメリットがあると言われています。

WPS の DB2 エンジンはネイティブドライバーを経由して DB2 データベースとアクセスを 行う仕組みを提供します。

[SQL 言語を使用した DB 読み取り]

DB へのアクセスは SQL 言語を用いて行うことが多く、WPS でも SQL プロシジャを用い て SQL 言語による DB アクセスが可能です。

SQL 言語を用いて WPS からネイティブドライバー経由で DB2 の DB アクセスを行う場合 は、以下のように、エンジン名として DB2 を指定し、CONNECT 文のカッコ内に datasrc=DB名 などの接続オプションや schema=スキーマ名、dbmax_text=n (文字テキス トの読み取り最大長さの設定) などの libname オプションを必要に応じて指定します。

以下は sample データベースの staff テーブルから、DEPT=10 または 20 に該当するレコー ドのみ抽出した全カラムを含む WPS データセット WORK.staff を作成しています。

```
proc sql;
  connect to DB2 (datasrc=sample);
  create table staff as
  select *
  from connection to DB2
  (
    select *
    from staff
    where DEPT in (10, 15);
  );
  disconnect from DB2;
quit;
```

※3行目の create table 文はWPS データセット WORK.staff を作成する指定です。 8行目の from 節で指定した staff は sample データベースの staff テーブルを参照しています。(データベースのテーブル名の指定は、本来、「スキーマ名.テーブル名」の形式で指定します。しかし、テーブルの所属するスキーマ名が接続する DB2 ユーザー名と同じ場合は「スキーマ名.」の部分は省略できるため、ここでは省略しています。)

(WPS ワークベンチの実行ログ)

```
8
         proc sql;
           connect to DB2 (datasrc=sample);
9
NOTE: Successfully connected to database DB2 as alias DB2.
10
           create table staff as
           select *
11
           from connection to DB2
12
13
             select *
14
             from staff
15
16
             where DEPT in (10, 15);
17
           );
NOTE: Data set "WORK.staff" has 8 observation(s) and 7 variable(s)
           disconnect from DB2;
18
NOTE: Successfully disconnected from database DB2.
19
         quit;
```

```
NOTE: Procedure sql step took :
real time : 1.594
cpu time : 0.031
```

作成された WPS データセット WORK.staff データセットの内容を確認します。

	Local Server の W	ORK.STAFF.DATA	🛛 🔀 🔁 DB2.s	as				
	ID	NAME	DEPT	o Job	YEARS	SALARY	OMM	^
1	50	Hanes	15	Mgr	10	80659.80		
2	70	Rothman	15	Sales	7	76502.83	1152.00	
3	110	Ngan	15	Clerk	5	42508.20	206.60	
4	160	Molinare	10	Mgr	7	82959.20		
5	170	Kermisch	15	Clerk	4	42258.50	110.10	
6	210	Lu	10	Mgr	10	90010.00		
7	240	Daniels	10	Mgr	5	79260.25		
8	260	Jones	10	Mgr	12	81234.00		
<								>

※ 注意: DB2 ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジ ャの CONNECT 文の() 内のオプションに、その他の指定が必要となる場合があります。 指定可能なオプションは、以下のように、ヘルプビューで確認できます。

まず、ヘルプビュー³の左上にある「内容」をクリックします。ヘルプ全体の目次が表示されますので、WPS Reference for Language Elements をクリックします。



WPS レファレンスを展開し、WPS Engine for DB2 を選択します。

³ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

E TIN O	🕒 🗢 💠 🗖 🗖
陷 内容 🎾 検索 📽 関連トビック	💷 ブックマーク 🛗 インデックス
スコープ: デフォルト	
🕨 🍥 Workbench User Guide	
WPS Reference for Language	e Elements
About this Guide	
How to Read EBNF Nota	tion and Syntax Diagrams
WPS Core	
WPS Graphing	
WPS Statistics	
WPS Timeseries	
WPS Communicate	
WPS Engine for Actian M	atrix
WPS Engine for DB Files	
MIDS Engine for DB2	
BE WPS Engine for Informity	111
N WPS Engine for Kognition WPS Engine for Kognition	
WPS Engine for MySOL	
WPS Engine for NETEZZ	A
WPS Engine for ODBC	
WPS Engine for OLEDB	
WPS Engine for Oracle	
WPS Engine for Postgres	OL .
WPS Engine for SAND	
WPS Engine for SQL Ser	ver
WPS Engine for Sybase	
WPS Engine for Teradata	i
WPS Engine for Vertica	
Appendix	
Legal Notices	
WPS Workbench User Guide	2

Library Engines \rightarrow DB2 を選択します。 (z/OS 版⁴の場合は DB2_zOS を選択)



データベース接続オプション(connection option)が表示されます。

⁴ z/OS 版では procedurs にあるデータ抽出用の DB2EXT プロシジャも利用可能です。



これらの接続オプション (user= password= datasrc= noprompt= prompt= required=) は DB2 ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプション指定 としても、また、次に節で説明する libname ステートメントによる DB 接続オプションとし ても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを入力し、実行します。

libname in DB2 datasrc=sample;

この指定は、WPS がデータベース sample に DB2 ネイティブドライバー経由で接続し、 sample データベースのテーブルとビューをライブラリ名 in で参照できるようにする指定 です。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設 定や、他の接続オプションが必要となる場合があります。

さらに、libname ステートメントには schema=スキーマ名、dbmax_text=n(文字テキスト の読み取り最大長さの設定)のような libname オプションを追加指定できます。以下のよう な libname オプションが指定可能です。(前出の DB2 エンジンのヘルプ画面から確認でき ます。)なお、schema=スキーマ名を指定しない場合の既定のスキーマ名は DB 接続ユーザ 一名です。

libname-option

libname-option
[access = [READONLY] | autocommit = [YES | N0] | bulkload = [YES | N0] | bl_allow_read_access = [YES | N0] | bl_cpu_parallelism = n |
data_buffer_size = n | bl_disk_parallelism = n | bl_indexing_mode = [autoselect | rebuild | incremental | deferred] | bl_cad_replace = [YES |
N0] | bl_log = " fully_qualified_log_file_name " | bl_method = cliload | bl_recoverable = [YES | N0] | connection = [sharedread | unique | shared |
globalread | global] | cursor type = [dynamic | forward_only | keyset_driven | static] | docommit = n | dbconint = " Database_command " | dbcreate_table_opt = " oreale_table_options " | doben_name = [DENS | SAS] | dblibnit = " Database_command " | dbcreate_table_opt = " oreale_table_opt = " oreale_table_options " | doben_name = [DENS | SAS] | dblibnit = " Database_command " | dbcreate_table_opt = " oreale_table_opt = " Ostabase_command " | dbcreate_table_opt = " oreale_table_opt = "

READONLY A	Yes by the second only columns of the YES by
autocommit of the YES	t in = tablesnace name
	in a northuff = n
bulkload t(=) YES by	preserve col names of = book YES
	preserve_tab_names = + (YES) -
bl cpu parallelism = n	
data buffer size = n	preserve_names = YES
bl disk parallelism = n	
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	query_timeout = n
rebuild	read_isolation_level (=) (=) (RR) (RR)
- incremental	
deferred	
bl_load_replace = YES	read_lock_type = ROW
bi_log = " ruiy quaimed log me name "	schema = schema name
bi_method = cilioad	
	sql=functions by = brows
connection (=) sharedread	ALL
- unique	stringdates
shared	
-> globalread	update_isolation_level * = * * R R
global	
cursor_type = - dynamic	Supdate lock type S = S S ROW
forward_only	
keyset_driven	utilconn_transient = + YES
dbcommit - n	
dbconinit = "Database command "	Unsupported Ontions
dbconterm = "Database command "	
dbgen name a = b a pave a	i ne tollowing options are not yet supported:
SAS	AUTHDOMAIN BL ALLOW WRITE ACCESS
dblibinit = " Database command "	BL_COPY_LOCATION
dblibterm = " Database command "	BL_OPTIONS
dbmax_text = n	BL_REMOTE_FILE CONNECTION GROUP
direct_exe = delete	DBINDEX
direct_sql + = + YES	DBROULLKEYS DBPROMPT
	DBSASLABEL DBSUCEPARM
NOR	• DEFER
NOGENSQL	REREAD_EXPOSURE SQL_FUNCTIONS_COPY
NOFUNCTIONS	
+ NOMULTOUTJOINS	
	1

さて、実行すると、ログに 「ライブラリ名 in が割り当てられた」というメッセージが表 示されます。このメッセージはDB2エンジン経由でDBと接続できたことを表します。

29 libname in DB2 datasrc=sample; NOTE: Library in assigned as follows: Engine: DB2 Physical Name: sample

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に sample データベースの テーブル名およびビュー名が含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。



※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる だけ避けてください。

さて、DB2 エンジンを libname ステートメントで指定した後は、以下のプログラムのよう に、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ入力 に用いることもできます。この例では、staff テーブルを入力として職種 (JOB) 別の平均給 料 (SALARY) の棒グラフを描いています。

goptions cback=white; title "Salary mean by Job"; proc gchart data=in.staff; vbar job/sumvar=salary type=mean patternid=midpoint width=10 space=5 mean; run;



※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、staff テーブルから職種(JOB)が"Mgr"か"Sales"のいずれかに該当するレコード を選択した変数 Name, Job, Salary を含む WPS データセット WORK.Mgr_or_Sales を DATA ステップで作成しています。

```
data Mgr_or_Sales;
  set in.staff(keep=Name Job Salary);
  where Job in ('Mgr', 'Sales');
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステー

トメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名やスキーマ名)を libname ステートメン トで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで sample テーブルから職種が "Mgr" か "Sales" に該当するレコ ードを選択した WPS データセット WORK.Mgr_or_Sales を DB2 の Mgr_or_Sales テーブ ルへ出力してみましょう。以下のようにプログラムを書いて実行します。

```
proc dbload data=Mgr_or_Sales dbms=DB2;
    datasrc=sample;
    table=Mgr_or_Sales;
    load;
run;quit;
```

※ 注意: DB にアクセスするため DB2 エンジンの接続オプションや libname オプション がさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で指定を追 加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は sample データベースの sales テーブルを読んで、SALES_PERSON="LEE"に 該当するレコードのみ抽出した全カラムを含むテーブル LEE を作成しています。

```
libname sales DB2 datasrc=sample;
proc sql;
  create table sales.LEE as
  select *
  from sales.sales
  where sales_person = "LEE";
quit;
```

※ この例では、Sales ライブラリは DB2 の sample データベースを参照するよう最初の libname ステートメントで定義しています。したがって、SQL プロシジャの中の Sales ラ イブラリは WPS データライブラリではなく、sample DB を表すものと解釈され、読み取 りも書き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname sales DB2 datasrc=sample;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=sales;
    delete LEE;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data sales.LEE;
    set sales.sales;
    where sales_person = "LEE";
run;
```

2.2. ODBC ドライバー経由の DB テーブルの入出力

ODBC ドライバーは、Windows 環境や Linux 環境から各種 DB とのインタフェースを共通 方式で行うためのソフトウェアです。ほとんどの DB の ODBC ドライバーが DB ベンダや サードパーティから提供されており、ODBC ドライバーの設定を行えば、DB ごとの細かい アクセス条件等の設定を気にせずに、共通の構文で DB アクセスが実現できるという利点が あります。

[ODBC ドライバーのインストール]

DB2のODBCドライバーはDB2インストール時に自動でインストールされます。もしも、 次の[PC側のODBCドライバーの設定]でDB2のODBCドライバーが見当たらない場合 は、インストール環境を確認してください。

[PC 側の ODBC ドライバーの設定]

WPS が導入されている PC 側では、以下のような手順で、DB2 の sample データベースを ODBC データソースとして登録しておきます。

コントロールパネル → システムとセキュリティ → 管理ツールで ODBC データソースを 選択します。

🖄 🗋 🔋			管理ツール				-		<
ファイル ホーム 共有	表示							^	0
レビー 貼り付け X 切り取り ロビー 貼り付け ズスのコビー スタートカッ	 トの貼り付け	】 []] 助先 コピー先 肖	× 中 除 名前の 変更	新しい フォルダー オ ア	्रे रवार्	₩ 開く ▼ 2 編集 イ 砂 履歴	 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	れ	
クリップボード		整理		新規		開く	遥択		
(€) → ↑ ((() + ⊐>	トロール パネル 🕨 :	システムとセキュリティ	ィ ▶ 管理ツー	ılı	~ ¢	管理ツール	の検索	, o	
▲☆ お気に入り	名前	*		更新日時	1	種類	サイズ		^
📕 ダウンロード	🔝 iSCSI イニシ	エーター		2013/08/22 1	5:57	ショートカット		2 KB	
📰 デスクトップ	📷 ODBC データ	7 ソース (32 ビット)		2013/08/22 8	:56	ショートカット		2 KB	
💹 最近表示した場所	📷 ODBC データ	7 ソース (64 ビット)	Ν	2013/08/22 1	5:59	ショートカット		2 KB	
	😿 Windows P	owerShell (x86)	7	2013/08/23 0	:34	ショートカット		3 KB	
▷ 🔣 ホームグループ	📷 Windows P	owerShell ISE (x86)	2013/08/22 1	5:55	ショートカット		2 KB	
	📷 Windows P	owerShell ISE		2013/08/22 1	5:55	ショートカット		2 KB	
Þ 🖳 PC	📷 Windows 🗴	モリ診断		2013/08/22 1	5:52	ショートカット		2 KB	
	🛃 イベント ビュー	7-		2013/08/22 1	5:55	ショートカット		2 KB	
> 📬 ネットワーク	🛃 コンピューター	の管理		2013/08/22 1	5:54	ショートカット		2 KB	
	🍺 コンポーネント	サービス		2013/08/22 1	5:57	ショートカット		2 KB	
	🗟 サービス			2013/08/22 1	5:54	ショートカット		2 KB	¥
21 個の項目								8== (

(左図)「追加」をクリックします。

(右図)リストにある DB2 用 ODBC ドラ	ライバーを選択し、「完了」を押します
ODBC データ ソース アドミニストレーター (64 ビット)	データ ソースの新規作成
ユーザー DSN システム DSN アライル アライル アメイル アメール アメール	セットアップするデータ ソースのドライバーを選択してください(S) 名類 IBM DB2 ODBC DRIVE IBM DB2 ODBC DRIVE IBM DB2 ODBC DRIVE Consort Access BASE Driver (*.td), *.ndx, *.ndx Microsoft Access Text Driver (*.td), *.ndx, *.ndx Microsoft Access Text Driver (*.td), *.ndx, *.xds) Microsoft Access Text Driver (*.td), *.ndx, *.xds) Microsoft Access Text Driver (*.td), *.ndx, *.xds) MySQL ODBC 5.3 Unicode Driver v
OK キャンセル 道用(A) ヘルプ	< 戻る(B) 売了 キャンセル

(左図)新規データソースの名前と説明にはユーザーが自由に入力できます。ここでは、名前に「DB2_SAMPLE」、説明には「DB2 Express-C sample database」と入力し「OK」を押します。

(右図) ODBC データソースリストに DB2 用 ODBC ドライバー「DB2_SAMPLE」が追加 されたことを確認し、これを選択してから「構成」を押します。

000		ODBC データ ソース アドミニストレーター (64 ビット)
ODBC	, IDM DD2 Pフ1ハー - 追加	ユーザー DSN システム DSN ファイル DSN ドライバー トレース 接続プール パージョン情報
ODBC に登録する DI 規の別名を作成するか および記述を変更した	82 データベースの別名を選択するか、あるいは新 こめに「追加」を選択してください。データ・ソース名 ・り、デフォルトを適用することが可能です。	ユーザーデータンー2(U): 名前 ブラドンオーム ドライバー Adverture/Wink 32(04 fc)・ DOBC Driver 11 for SQL Server FGM Adverture/Wink 32(04 fc)・ 11 MM DB2 DOBC DRIVER - 082(DDF) Excel Files 46 fc)・ Microsoft Excel Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 46 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS5 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS6 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS6 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS6 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS6 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdor MS6 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdox, *xdor MS7 Access Database 47 fc)・ Microsoft Access Driver (* xxf, *xdox, *xdox
データ・ソース名	DB2_SAMPLE	MYSQL_SAKILA 32/64 E91 MySQL ODBC 5.3 ANSI Driver
データベース別名	SAMPLE v 追加(A)	< >
記述	DB2 Express-C sample database	0082 ユーザーデータリースには癒まれたデータプルバイダーへの勝利力活動する撮影が破損されています。ユーザ データータックスは現在のユーザーになみ変更され、このコンとユーターでの本規用できます。
		OK キャンセル 道用(A) へルプ

「接続」をクリックし接続テストを行います。

	CLI/ODBC 設定 - DB2_SAM	IPLE
-タ・ソース 詳細設定		
データ・ソーフタ	DB2_SAMPLE	
記述	ress-C sample database	
データ・ソースに接続して	縄成情報を検索する	
ユーザー ID		
パスワード		
	□ パスワードの保管(S)	DB2 メッセージ
	接続(C)	接続は正常にテストされました。
	CLI/ODBC サポート・ユーティリティ	
		ок 🖓

正常に接続できたなら「OK」を押し ODBC ドライバーの設定を終了します。

[SQL 言語を使用した DB 読み取り]

SQL 言語を用いてWPS から ODBC ドライバー経由で DB2 の DB アクセスを行う場合は、 以下のように、エンジン名として ODBC を指定し、CONNECT 文のカッコ内に datasrc=ODBC データソース名 などの接続オプションや dbmax_text=n (文字テキストの 読み取り最大長さの設定) などの libname オプションを必要に応じて指定します。

下記は、ODBC ドライバーを用いて、sample データベースの staff テーブルから、DEPT=10 または20に該当するレコードのみ抽出した全カラムを含むWPS データセットWORK.staff を作成する例となっています。

```
proc sql;
  connect to ODBC (datasrc=DB2_SAMPLE);
  create table staff as
  select *
   from connection to ODBC
  (
    select *
    from staff
   where DEPT in (10, 15);
  );
  disconnect from ODBC;
quit;
```

(WPS ワークベンチの実行ログ)

```
314
         proc sql;
315
           connect to ODBC (datasrc=DB2_SAMPLE);
NOTE: Successfully connected to database ODBC as alias ODBC.
           create table staff as
316
317
           select *
318
           from connection to ODBC
319
           (
320
             select *
321
             from staff
             where DEPT in (10, 15);
322
```

```
323 );
NOTE: Data set "WORK.staff" has 8 observation(s) and 7 variable(s)
324 disconnect from ODBC;
NOTE: Successfully disconnected from database ODBC.
325 quit;
NOTE: Procedure sql step took :
    real time : 0.359
    cpu time : 0.015
```

※ ODBC ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の()内のオプションに、その他の指定が必要となる場合があります。以下の ようにヘルプビューで指定可能なオプションが確認できます。

ヘルプビュー5の左上にある「内容」をクリックします。

ヘルプ全体の目次が表示されます。WPS Reference for Language Elements をクリックします。



WPS レファレンスを展開し、WPS Engine for ODBC を選択します。



⁵ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

その後、展開を何度か行うと接続オプション(connection option) が表示されます。



これらのオプション(user= password= datasrc= noprompt= prompt= readbuff= required=) は ODBC ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプシ ョン指定としても、また、次に説明する libname ステートメントによるデータベースアクセ スの接続オプションとしても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを実行します。

libname in ODBC datasrc=DB2_SAMPLE;

この指定は、WPS がデータソース DB2_SAMPLE に ODBC ドライバー経由で接続し、ライ ブラリ名 in で参照できるようにする指定です。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設定や、他の接続オプション が必要となる場合があります。

さらに、libname ステートメントには dbmax_text=n(文字テキストの読み取り最大長さの 設定)などの libname オプションを追加指定できます。指定可能な libname オプションは **DB2** エンジンの libname オプションとほぼ同じですが、詳細は前出の **ODBC** エンジンのヘルプ画面から確認してください。(Connection-option の下に libname-option が記載されています。)

実行すると、ログに「ライブラリ名 in が割り当てられた」というメッセージが表示されま す。このメッセージは ODBC 経由で DB と接続できたことを表します。

<pre>345 libname in ODBC datasrc=DB2_sample;</pre>	
NOTE: Library in assigned as follows:	
Engine: ODBC	
Physical Name: DB2_sample	

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に sample データベースの テーブル名およびビュー名が含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。

Librariae			כר	パティ	STAFF			_ □
→ III In → ACT	フィルタ入力	カラム	-					⇔ → ⇒ →
Image: CATALOG Image: CATALOG	インデックス 万万山 詳細 全般	名前 ② COMM ③ DEPT ③ ID ④ JOB ④ NAME ④ SALARY ④ YEARS	タイプ number number number character character number number	長さ 8 8 5 9 8 8 8	フォーマット 9.2 6. 5. \$9. 9.2 6.	インフォオーマット 9.2 6. 6. \$5. \$9. 9.2 6.	5/JL COMM DEPT ID JOB NAME SALARY YEARS	
□ PURCHASCORDER □ SALES □ STAFF □	?	<					ОК	キャンセル
등 VXST 등 그ピー Cbri+C SVST 등 XBR 開始 등 VDEF 第 時後 開始 등 VDEF 第 デークセットをインボート 등 VDEF 등 デークセットをインボート 등 VDEF 등 デークセットを2025年 등 VDEF 등 VDEF 등 2025年								

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> <u>テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS</u> セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる <u>だけ避けてください。</u>

さて、ODBC エンジンを libname ステートメントで指定した後は、以下のプログラムのように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ入力に用いることもできます。この例では、staff テーブルを入力として職種 (JOB) 別の平均給料 (SALARY)の棒グラフを描いています。

goptions cback=white; title "Salary mean by Job"; proc gchart data=in.staff; vbar job/sumvar=salary type=mean patternid=midpoint width=10 space=5 mean; run;



※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、staff テーブルから職種(JOB)が"Mgr"か"Sales"のいずれかに該当するレコード を選択した変数 Name, Job, Salary を含む WPS データセット WORK.Mgr_or_Sales を DATA ステップで作成しています。

```
data Mgr_or_Sales;
  set in.staff(keep=Name Job Salary);
  where Job in ('Mgr', 'Sales');
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステー

トメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名やスキーマ名)を libname ステートメン トで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで sample テーブルから職種が "Mgr" か "Sales" に該当するレコ ードを選択した WPS データセット WORK.Mgr_or_Sales を DB2 の Mgr_or_Sales テーブ ルへ出力してみましょう。以下のようにプログラムを書いて実行します。

```
proc dbload data=Mgr_or_Sales dbms=ODBC;
    datasrc=DB2_SAMPLE;
    table=Mgr_or_Sales;
    load;
run;quit;
```

※ 注意: DB にアクセスするための権限や ODBC エンジンの接続オプションや libname オ プションがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で 指定を追加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は sample データベースの sales テーブルを読んで、SALES_PERSON="LEE"に 該当するレコードのみ抽出した全カラムを含むテーブル LEE を作成しています。

```
libname sales ODBC datasrc=DB2_SAMPLE;
proc sql;
  create table sales.LEE as
  select *
  from sales.sales
  where sales_person = "LEE";
quit;
```

※ この例では、Sales ライブラリは ODBC の DB2_SAMPLE データソースを参照するよう 最初の libname ステートメントで定義しています。したがって、SQL プロシジャの中の Sales ライブラリは WPS データライブラリではなく、sample DB を表すものと解釈され、 読み取りも書き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname sales ODBC datasrc=DB2_SAMPLE;
/* 同じ名前のテーブルを一旦削除 */
proc sql;
    drop table sales.LEE;
quit;
/* DATAステップでDBテーブルを読み書き */
data sales.LEE;
    set sales.sales;
    where sales_person = "LEE";
run;
```

3. MySQL とのインタフェース

[sakila サンプル DB]

MySQL Community Server 5.6 に付随する sakila データベースを例として用います。

M		MySQ	L Workbench			м2	- 🗆 🗙
A Local instance MySQL56 ×							
File Edit View Query Databas	se Server Tools Scrip	ting Help					
8 8 6 8 8 8 8							
Navigator	Query 1 actor	actor ×				SOI Additions	
SCHEMAS 43 J		0.0					6 Lines
9 Filter objecte				uorows • 🔀	$ \ge <$		1 Jamp to
	I · Seccer ·	FROM Sakiia.a	ccor;			Automatic	context help is disabled
Tables	<					Use the to	olbar to manually get
▶ ■ actor	In manual III at an					or to togal	e current caret position e automatic help.
address	Kesult Grid	ter Nows:	Edit: 🜠 🗄	Export/Im			
category	actor_id first_name	last_name	last_update		Result		
country	PENELOPE	GUINESS	2006-02-15 04:34:33		Gna		
customer	2 50	CHARE	2006-02-15 04:34:33				
film actor	A TENNITEED	DAVIS	2006-02-15 04:24:22		Form		
▶ ■ film_bird	5 IOHNNY	LOLLOBRIGIDA	2006-02-15 04-34-33		Editor		
film_category	6 BETTE	NICHOLSON	2006-02-15 04:34:33				
▶ film_text	7 GRACE	MOSTEL	2006-02-15 04:34:33				
inventory	8 MATTHEW	IOHANSSON	2006-02-15 04:34:33		Field Types		
Ianguage	9 X0E	SWANK	2006-02-15 04:34:33				
payment	10 CHRISTIAN	GABLE	2006-02-15 04:34:33				
rental	11 ZERO	CAGE	2006-02-15 04:34:33		Query		
▶ ■ store	12 KARL	BERRY	2006-02-15 04:34:33				
Views	13 UMA	WOOD	2006-02-15 04:34:33				
Stored Procedures	14 VIVIEN	BERGEN	2006-02-15 04:34:33		Execution		
► P tot	15 CUBA	OLIVIER	2006-02-15 04:34:33		Plan		
Management Schemas	16 FRED	COSTNER	2006-02-15 04:34:33				
Information	17 HELEN	VOIGHT	2006-02-15 04:34:33				
Table: actor	actor 1 ×			Apply	Revert	Context Help	Snippets
Columns:	Output						
actor id smallint(5) UN AI PK first name varchar(45)	Action Output	-					
last_name varchar(45)	Time Action			Message			Duration / Fetch
an optime an optimp	1 14:37:57 SELECT	* FROM sakila.actor	LIMIT 0, 1000	200 row(s) returned			0.000 sec / 0.000 sec
Object Info Session							

3.1. ネイティブドライバー経由の DB テーブルの入出力

まず、コマンドラインインタフェース(CLIドライバー、ネイティブドライバー)を用いて

WPS から DB テーブルの入出力を行ってみましょう。

[ネイティブドライバのインストールと Path の定義]

WPS は MySQL との接続を行うネイティブドライバーとして、Connector/C(libmysqlclient) ライブラリを用います。このファイル (libmysqlclient.dll) はインストールすると、MySQL インストールディレクトリの下の lib ディレクトリ内に保管されます。これをアクセスでき るように、以下のように、環境変数 Path に定義しておく必要があります。

(左図) コントロールパネル → システム → システムの詳細設定を開きます。

「環境変数」をクリック → 「Path」をクリックします。

(右図) ユーザー環境変数 Path を選択し、「編集」をクリックします。なお、他のユーザーでも有効な Path を設定したい場合はシステム環境変数の Path を選択し編集してください。

のユーザー環境変 変数 <mark>Path</mark> TEMP TMP	数(U) 値 C:¥Program Fles¥World Programming %USERPROFILE%¥AppData¥Local¥Temp %USERPROFILE%¥AppData¥Local¥Temp 新規(N) 編集(E) 削除(D)
変数 <mark>Path</mark> TEMP TMP	他 C:¥Program Files¥World Programming %USERPROFILE%¥AppData¥Local¥Temp %USERPROFILE%¥AppData¥Local¥Temp 新規(N) 編集(E) 削除(D)
Path TEMP TMP	C:VProgram Files¥World Programming %USERPROFILE%#AppData¥Local¥Temp %USERPROFILE%#AppData¥Local¥Temp 新規(N) 編集(E) 削除(D)
ТЕМР ТМР	%USERPROFILE%¥AppData¥Local¥Temp %USERPROFILE%¥AppData¥Local¥Temp 新規(N) 編集(E) 削除(D)
ТМР	%USERPROFILE%¥AppData¥Local¥Temp 新規(N) 編集(E) 削除(D)
	新規(N) 編集(E) 削除(D)
	·
システム環境変数(S)	J. J
変数	値
CLASSPATH	.;G:¥Program Files¥IBM¥SQLLIB¥java¥
ComSpec	C:¥Windows¥system32¥cmd.exe
DB2INSTANCE	DB2
FP_NO_HOST_C	NO
	新規(W) 編集(I) 削除(L)
	OK キャンセル
	システム環境変数(S) 変数 CLASSPATH ComSpec DB2INSTANCE FP_NO_HOST_C

変数値の最後に、以下のように、MySQL データベースの lib ディレクトリを追記し、「OK」 を押します。(実際に DB をインストールしたディレクトリ名を入力してください)

;C:¥Program	Files¥MySQL¥MySQL	Server	5.6¥lib¥	
-------------	-------------------	--------	----------	--

	ユーザー変数の編集 ×
変数名(N):	Path
変数値(V):	jram Files¥MySQL¥MySQL Server 5.6¥lib¥
	ОК 👌 + ФУСИ

WPS ワークベンチを起動している状態の場合は、Path の設定後、WPS ワークベンチを再 起動⁶してください。

ネイティブドライバーは、一般に ODBC ドライバーより高速に DB アクセスができる点で メリットがあると言われています。

WPS の MYSQL エンジンはネイティブドライバーを経由して MySQL データベースとアク セスを行う仕組みを提供します。

[SQL 言語を使用した DB 読み取り]

DB へのアクセスは SQL 言語を用いて行うことが多く、WPS でも SQL プロシジャを用い て SQL 言語による DB アクセスが可能です。

SQL 言語を用いて WPS からネイティブドライバー経由で MySQL の DB アクセスを行う 場合は、以下のように、エンジン名として MYSQL を指定し、CONNECT 文のカッコ内に server=サーバー名、database=DB 名、user=ユーザー名、password=パスワード などの接 続オプションや、後述する dbmax_text=n (文字テキストの読み取り最大長さの設定) など の libname オプションを指定します。

以下は sakila データベースの city テーブルから、country_id=50 に該当するレコードのみ抽 出した全カラムを含む WPS データセット WORK.japan_city を作成しています。

```
proc sql;
  connect to MYSQL (server=localhost database=sakila user=XXX
password=password);
  create table japan_city as
  select *
  from connection to MYSQL
  (
    select *
    from city
    where country_id=50
  );
  disconnect from MYSQL;
quit;
```

※ 2 行目の user=XXX password=password の XXX と password にはそれぞれユーザ名とパ スワードを指定します。

3 行目の create table 文は WPS データセット WORK.japan_city を作成する指定です。

⁶ メニューバーの「ファイル(**F**)」 → 「再起動」をクリックします。

8 行目の from 節で指定した city は sakila データベースの city テーブルを参照していま t_{o}

(WPS ワークベンチの実行ログ)

```
8
          proc sql;
9
           connect to MYSQL (server=localhost database=sakila user=XXX password=XXXX);
NOTE: Successfully connected to database \ensuremath{\mathsf{MYSQL}} as alias \ensuremath{\mathsf{MYSQL}} .
10
            create table japan_city as
            select *
11
12
            from connection to MYSQL
13
            (
14
              select *
15
              from city
16
              where country_id=50
17
            );
NOTE: Data set "WORK.japan_city" has 31 observation(s) and 4 variable(s)
18
            disconnect from MYSQL;
NOTE: Successfully disconnected from database MYSQL.
19
          quit;
NOTE: Procedure sql step took :
      real time : 0.015
      cpu time : 0.000
```

```
作成された WPS データセット WORK.staff データセットの内容を確認します。
```

🔁 M'	YSQL1.sas	ີ Local Server Ø	WORK.JAPAN_CI	TY.DATA 🕄	
	a city_id	city	a country_id	a last_update	^
1	10	Akishima	50	15FEB2006:04:45:25	
2	172	Fukuyama	50	15FEB2006:04:45:25	
3	203	Higashiosaka	50	15FEB2006:04:45:25	
4	204	Hino	50	15FEB2006:04:45:25	
5	205	Hiroshima	50	15FEB2006:04:45:25	
6	224	Isesaki	50	15FEB2006:04:45:25	
7	226	Iwaki	50	15FEB2006:04:45:25	
8	227	Iwakuni	50	15FEB2006:04:45:25	
9	228	Iwatsuki	50	15FEB2006:04:45:25	
10	229	Izumisano	50	15FEB2006:04:45:25	
11	253	Kakamigahara	50	15FEB2006:04:45:25	
12	256	Kamakura	50	15FEB2006:04:45:25	
13	260	Kanazawa	50	15FEB2006:04:45:25	
14	276	Koriyama	50	15FEB2006:04:45:25	
15	284	Kurashiki	50	15FEB2006:04:45:25	
16	287	Kuwana	50	15FEB2006:04:45:25	-
17	331	Matsue	50	15FEB2006:04:45:25	
18	338	Miyakonojo	50	15FEB2006:04:45:25	
19	355	Nagareyama	50	15FEB2006:04:45:25	
20	376	Okayama	50	15FEB2006:04:45:25	
21	377	Okinawa	50	15FEB2006:04:45:25	
22	380	Omiya	50	15FEB2006:04:45:25	
23	382	Onomichi	50	15FEB2006:04:45:25	
24	386	Otsu	50	15FEB2006:04:45:25	
25	440	Sagamihara	50	15FEB2006:04:45:25	
26	463	Sasebo	50	15FEB2006:04:45:25	
27	474	Shimonoseki	50	15FEB2006:04:45:25	
28	521	Tama	50	15FEB2006:04:45:25	
29	547	Tsuyama	50	15FEB2006:04:45:25	
30	552	Ueda	50	15FEB2006:04:45:25	
31	555	Urawa	50	15FEB2006:04:45:25	
<					~ `

※ 注意: MYSQL ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロ シジャの CONNECT 文の() 内のオプションに、その他の指定が必要となる場合がありま す。指定可能なオプションは、以下のように、ヘルプビューで確認できます。 まず、ヘルプビュー⁷の左上にある「内容」をクリックします。ヘルプ全体の目次が表示されますので、WPS Reference for Language Elements をクリックします。



WPS レファレンスを展開し、WPS Engine for MySQL を選択します。



Library Engines \rightarrow MySQL を選択すると、データベース接続オプション (connection option) が表示されます。



⁷ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

これらの接続オプション(user=password=database=server=port=)は MYSQL ドライバ ー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプション指定としても、 また、次に節で説明する libname ステートメントによる DB 接続オプションとしても用いる ことができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを入力し、実行します。

libname in MYSQL database=sakila user=XXX password=password;

※ user=XXX password=password の XXX と password にはそれぞれユーザ名とパスワー ドを指定します。

この指定は、WPS がデータベース sakila に MYSQL ネイティブドライバー経由で接続し、 sakila データベースのテーブルとビューをライブラリ名 in で参照できるようにする指定で す。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設定 や、他の接続オプションが必要となる場合があります。

さらに、libname ステートメントには dbmax_text=n(文字テキストの読み取り最大長さの 設定) などの libname オプションを追加指定できます。以下のような libname オプション が指定可能です。(前出の MYSQL エンジンのヘルプ画面から確認できます。)

<pre>libname-option [access = [READONLY] autocommit = [YES NO] bl_buffersize = r NO bl_lcal = [YES NO] bl_use_pipe = [YES NO] bulkod = dbconinit = "Database_command" dbconterm = "Database_comma dblibinit = "Database_command" dblibterm = "Database_comm NOGENSQL NOWHERE NOTUNCTIONS NOWLDUTJOINS insertbui preserve_names = [YES NO] qualifier = qualifer_name spool = [NO]]</pre>	<pre>></pre>
	-14
REALONLI	insertbuff = n
autocommit + = + YES +	preserve col names of = >>> YES ->>
NO	
\rightarrow hl huffereize = n	
	preserve_tab_names = or vYES or
<pre>bl_datafile = " path to file with / as separator "</pre>	
bl delete datafile (=) >>> YES	preserve_names w = w YES w
	a gualifian = gualifer name
	spool b(=) by YES
bl_use_pipe = YES	
NO NO	sql-functions =
bulkoad = > YES	
	timeout = n
connection w = w sharedread	utilconn_transient >=>>> YES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
globalread	
dbcommit = n	Unsupported Options
dbconinit = " Database command "	The following options are not yet supported:
dbconterm = " Database command "	AUTHDOMAIN
dbcreate_table_opts = " create table options "	CONNECTION_GROUP
	DBINDEX DBMSTEMP
	DBNULLKEYS
	DBPROMPT DBSASLABEL
dblibinit = " Database command "	DBSLICEPARM
dblibterm = " Database command "	DEFER ESCADE BACKSLASH
dbmax_text = n	MULTI_DATASRC_OPT
direct exe = delete	REREAD_EXPOSURE SOL FUNCTIONS CODY
	· SQL_FUNCTIONS_COPT
direct_sql * = > YES	
NO NO	
NONE NONE	
NOGENSQL	
NOWHERE	
NOFUNCTIONS	
NONTI BOUR TOTAL	

さて、実行すると、ログに 「ライブラリ名 in が割り当てられた」というメッセージが表示されます。このメッセージは MYSQL エンジン経由で DB と接続できたことを表します。

29 libname in MYSQL database=sakila user=XXX password=XXXX; NOTE: Library in assigned as follows: Engine: MYSQL Physical Name:

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に sakila データベースの テーブル名およびビュー名が含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。

😵 WPS サーバーエ 🙁 🏠 プロジェクトエクス 🏪 ファイルエクスプロ 📃 🗆							
© Local Server			プロバティ	: film	I		- • ×
	フィルタ入力	カラム					(→ → → → →
b m actor	インデックス						
actor_info	751	名前	タイプ	長さ	フォーマット	インフォオーマット	ラベル
address	言語	description	character	1024	\$1024.	\$1024.	description
Ell category	全般	⊛ film id	number	8	6.	6.	film id
El city		anguage_id	number	8	4.	4.	language_id
country		iast_update	number	8	DATETIME19.	DATETIME19.	last_update
El customer		8 length	number	8	6.	6.	length
eustomer_list		original_language_id	number	8	4.	4.	original_language_id
		rating	character	5	\$5.	\$5.	rating
		e release_year	number	8	DATE9.	DATE9.	release_year
更新 F5		rental_duration	number	8	4.	4.	rental_duration
		rental_rate rental_rate	number	0	0.2	0.2	renta_rate
		special features	character	54	\$54.	\$54.	special features
b 📰 lang		ittle	character	255	\$255.	\$255.	title
記 nice 家 データセットをインボート							
Pay E データセットをエクスポート							
▷ III ren クリップボードに内容をコピー		<					3
		<					>
attenter							
staff liet						0	OK キャンセル
≥ ■ store							
Sashelp							
> 📰 Sasuser							
🛛 📰 Work							
Filerefs							

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS <u>セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる</u> だけ避けてください。

さて、MYSQL エンジンを libname ステートメントで指定した後は、以下のプログラムのように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ入力に用いることもできます。この例では、film テーブルを入力として映画鑑賞年齢制限(rating)別の映画の本数の棒グラフを描いています。





※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、film テーブルから題名(TITLE)に"BIRD"が含まれるレコードを選択した変数 File_id, title, rental_rate, rating を含む WPS データセット WORK.film_BIRD を DATA ステ ップで作成しています。

data film_BIRD;

```
set in.film(keep=film_id title rental_rate rating);
where title contains "BIRD";
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名など)を libname ステートメントで指定 しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで film テーブルから題名に"BIRD"を含むレコードを選択した WPS データセット WORK.film_BIRD を MySQL の sakila データベースの film_BIRD テーブルへ 出力してみましょう。以下のようにプログラムを書いて実行します。

```
proc dbload data=film_BIRD dbms=MYSQL;
server=localhost;
user=XXX;
password=password;
database=sakila;
table=film_BIRD;
load;
run;quit;
```

※ 注意: DB にアクセスするため MYSQL エンジンの接続オプションや libname オプショ ンがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で指定を 追加してください。また、user=XXX password=*password*の XXX と *password* にはそれぞ れユーザ名とパスワードを指定します。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと

同じように扱えるようになります。

以下の例は sakila データベースの city テーブルを読んで、country_id=50 に該当するレコー ドのみ抽出した全カラムを含むテーブル japan_city を作成しています。

```
libname in MYSQL database=sakila user=XXX password=password;
proc sql;
  create table in.japan_city as
  select *
  from in.city
  where country_id=50;
quit;
```

※ この例では、in ライブラリは MySQL の sakila データベースを参照するよう最初の libname ステートメントで定義しています。したがって、SQL プロシジャの中の in ライブ ラリは WPS データライブラリではなく、sakila DB を表すものと解釈され、読み取りも書 き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname in MYSQL database=sakila user=XXX password=password;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=in;
   delete japan_city;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data in.japan_city;
   set in.city;
   where country_id=50;
run;
```

3.2. ODBCドライバー経由の DB テーブルの入出力

ODBC ドライバーは、Windows 環境や Linux 環境から各種 DB とのインタフェースを共通 方式で行うためのソフトウェアです。ほとんどの DB の ODBC ドライバーが DB ベンダや サードパーティから提供されており、ODBC ドライバーの設定を行えば、DB ごとの細かい アクセス条件等の設定を気にせずに、共通の構文で DB アクセスが実現できるという利点が あります。

[ODBCドライバのインストール]

MySQL の ODBC ドライバーは Connector/ODBC を MySQL のバージョンに合わせてイン ストールします。 もしも、次の [PC 側の ODBC ドライバーの設定] で MySQL の ODBC ドライバーが見当たらない場合は、インストール環境を確認してください。

[PC 側の ODBC ドライバーの設定]

WPS が導入されている PC 側では、以下のような手順で、MYSQL の sakila データベース を ODBC データソースとして登録しておきます。

コントロールパネル → システムとセキュリティ → 管理ツールで ODBC データソースを 選択します。



(左図)「追加」をクリックします。



(左図)新規データソースの名前と説明にはユーザーが自由に入力できます。ここでは、名
前に「MYSQL_SAKILA」、説明には「MySQL sakila database」と入力します。ユーザー名 とパスワードを入力すると、データベース欄のドロップリストが利用できますので、「sakila」 を選択します。次に「Test」を押して接続テストを行います。接続テストが成功することを 確認すると設定終了です。

(右図)「OK」を押して ODBC データソースリストに MySQL 用 ODBC ドライバー 「MYSQL_SAKILA」が追加されたことを確認します。

MySQL Connector	/ODBC Data Source	Configu 💌				
MySQL			3	ODBC	データ ソース アドミニストレーター (64 ピッ	ih) 💌
Connector/ODB	С		ユーザー DSN	システム DSN ファイル DS	SN ドライバー トレース 接続プール バージョン情報	ŧ
			ユーザー データ	ソース(U):		
Connection Parameter	s		名前	プラットフォーム	ドライバー	追加(D)
Data Source Name:	MYSQL_SAKILA		Adventure DB2_SAMPL Excel Files	/orks 32/64 ビット .E 32/64 ビット 64 ビット	ODBC Driver 11 for SQL Server IBM DB2 ODBC DRIVER - DB2COPY1 Microsoft Excel Driver (* vls. * vlsv. * vlsv.	削除(R)
Description:	MySQL sakila database		MS Access I MYSQL_SA	Database 64 분ット KILA 32/64 분ット	Microsoft Access Driver (*.mdb, *.accdb) MySQL ODBC 5.3 ANSI Driver	構成(C)
TCP/IP Server:	P	ort: 3306				
Named Pipe:						
User:	XXX		<		>	
Password:	••••			DBC ユーザー データソース データソースは現在のユー	には指定されたデータ プロバイダーへの接続方法に関す ザーにのみ認識され、このコンピューターでのみ使用できま	る情報が格納されています。ユーザ す。
Database:	V	Test				
Details >>	mysql performance_schema and	el Help			<u>Ок</u> +т>ти	適用(A) へルプ
	world V		J			

[SQL 言語を使用した DB 読み取り]

SQL 言語を用いて WPS から ODBC ドライバー経由で MySQL の DB アクセスを行う場合 は、以下のように、エンジン名として ODBC を指定し、CONNECT 文のカッコ内に datasrc=ODBC データソース名などの接続オプションや、dbmax_text=n(文字テキストの 読み取り最大長さの設定)などの libname オプションを指定します。

下記は、ODBC ドライバーを用いて、sakila データベースの city テーブルから、country_id=50 に該当するレコードのみ抽出した全カラムを含む WPS データセット WORK.japan_city を 作成する例となっています。

```
proc sql;
connect to ODBC (datasrc=MYSQL_SAKILA);
create table japan_city as
select *
from connection to ODBC
(
   select *
   from city
   where country_id=50
);
disconnect from ODBC;
```

quit;

(WPS ワークベンチの実行ログ)

```
8
         proc sql;
9
           connect to ODBC (datasrc=MYSQL_SAKILA);
NOTE: Successfully connected to database ODBC as alias ODBC.
10
           create table japan_city as
11
           select *
           from connection to ODBC
12
13
           (
14
             select *
15
             from city
16
             where country_id=50
17
           );
NOTE: Data set "WORK.japan_city" has 31 observation(s) and 4 variable(s)
           disconnect from ODBC;
18
NOTE: Successfully disconnected from database ODBC.
19
         quit;
NOTE: Procedure sql step took :
     real time : 0.031
     cpu time : 0.015
```

※ ODBC ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の()内のオプションに、その他の指定が必要となる場合があります。以下の ようにヘルプビューで指定可能なオプションが確認できます。

ヘルプビュー8の左上にある「内容」をクリックします。

ヘルプ全体の目次が表示されます。WPS Reference for Language Elements をクリックします。



WPS レファレンスを展開し、WPS Engine for ODBC を選択します。

⁸ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

陷 内容 🎾 検索 📽 関連トピック 💷 ブックマーク 蘭 ·
スコープ: デフォルト
Workbench User Guide
WPS Reference for Language Elements
How to Read Syntax Diagrams
WPS Core
WPS Graphing
WPS Statistics
WPS Timeseries
WPS Communicate
WPS Engine for Actian Matrix
WPS Engine for DB Files
WPS Engine for DB2
WPS Engine for Greenplum
WPS Engine for Informix
WPS Engine for Kognitio
WPS Engine for MySQL
WPS Engine for NETEZZA
WPS Engine for ODPC
WPS Engine for OL
WPS Engine for Oracle
WPS Engine for PostgreSQL
WPS Engine for SAND
US WPS Engine for SQL Server
WPS Engine for Sybase
WPS Engine for Teradata
WPS Engine for Vertica
▷ La Appendix
Legal Notices

その後、展開を何度か行うと接続オプション(connection option) が表示されます。

	15 🛊 🖹 📲 🕖 😓 ▽ 🗖	
陷 内容 🌮 検索 叱 関連トビック 💷 ブックマーク 📖 インデックス		
WPS Reference for Language Elements > WPS Engine for ODBC > Library Engines		~
ODBC		
LIENAME name ODBC { connection-option } [{ libname-option }] ;		
▷ LIBNAME → <i>name</i> → ODBC → connection-option →	ibname-option →	
<pre>connection-option [user = user_name password = password datasrc = DSN datasrc = ["] conn connection_string ["] prompt = ["] connection_string ["] readbuff = n requir</pre>	<pre>bection_string ["] noprompt = ["] ed = ["] connection_string ["]]</pre>	I
user = user name	\sim	
password = password	_	
datasrc = DSN	_	
datasrc + = + connection string	-	
hoprompt he connection string	-	
<pre>>> prompt >> => >> >></pre>	-	
readbuff = n	-	
<pre> required += + + + + + + + + + + + + + + + + + +</pre>		~

これらのオプション (user= password= datasrc= noprompt= prompt= readbuff= required=) は ODBC ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプシ

ョン指定としても、また、次に説明する libname ステートメントによるデータベースアクセスの接続オプションとしても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを実行します。

libname in ODBC datasrc=MYSQL_SAKILA;

この指定は、WPS がデータソース MYSQL_SAKILA に ODBC ドライバー経由で接続し、ラ イブラリ名 in で参照できるようにする指定です。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設定や、他の接続オプション が必要となる場合があります。

さらに、libname ステートメントには schema=スキーマ名、dbmax_text=n(文字テキスト の読み取り最大長さの設定)などの libname オプションを追加指定できます。指定可能な libname オプションは MYSQL エンジンの libname オプションとほぼ同じですが、詳細は前 出の ODBC エンジンのヘルプ画面から確認してください。(Connection-option の下に libname-option が記載されています。)

実行すると、ログに「ライブラリ名 in が割り当てられた」というメッセージが表示されま す。このメッセージは ODBC 経由で DB と接続できたことを表します。

345	libname in ODBC datasrc=MYSQL_SAKILA;	
NOTE:	Library in assigned as follows:	
E	ngine: ODBC	
P	Physical Name: MYSQL_SAKILA	

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に sakila データベースの テーブル名およびビュー名が含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。

🕵 WPS サーバーエ 🖾 ն プロジェクトエクス 🔓 ファイルエクスプロ 🦳 🗆							
🖌 💿 Local Server	0		プロバティ	: film			- 🗆 ×
Libraries	-						
4 🐺 In	フィルタ入力	カラム					⇔ ▼ ⇔ ▼ ▼
actor	インデックス	-					
actor_info	75L	名前	タイプ	長さ	フォーマット	インフォオーマット	5/VL
address	詳紙の	description	character	1024	\$1024.	\$1024.	description
Ell category	全般	(B) film_id	number	8	6.	6.	film_id
El city		language_id	number	8	4.	4.	language_id
E country		last_update	number	8	DATETIME19.	DATETIME19.	last_update
El customer		length	number	8	6.	6.	length
Customer_list		original_language_id	number	8	4.	4.	original_language_id
		😐 rating	character	5	\$5.	\$5.	rating
b iii film 🔍 🛤		elease_year	number	8	DATE9.	DATE9.	release_year
▷ III film 更新 F5		rental_duration	number	8	4.	4.	rental_duration
		ental_rate	number	8	8.2	8.2	rental_rate
De tilm 📑 De - Ctri+C		eplacement_cost	number	8	9.2	9.2	replacement_cost
▷ 🎫 Inve 🗙 削除 🛛 削除		special_features	character	54	\$54.	\$54.	special_features
		🖲 title	character	255	\$255.	\$255.	title
▶ ■ Pay ■ データセットをエクスボート		<					2
クリップボードに内容をコピー		(
Sale IP プロパティ(R) Alt+Enter		`					
staft	0						N/
a staff list							TK +7701
store	1						
> 🚮 Sashelp							
> 📰 Sasuser							
> 🚮 Work							
Filerefs							

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる さて、ODBC エンジンを libname ステートメントで指定した後は、以下のプログラムのように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ入力に用いることもできます。この例では、film テーブルを入力として映画鑑賞年齢制限(rating)別の映画の本数の棒グラフを描いています。





[※] 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、film テーブルから題名(TITLE)に"BIRD"が含まれるレコードを選択した変数 File_id, title, rental_rate, rating を含む WPS データセット WORK.film_BIRD を DATA ステ ップで作成しています。

<pre>data film_BIRD;</pre>	
<pre>set in.film(keep=film_id</pre>	<pre>title rental_rate rating);</pre>

where title contains "BIRD";
run;

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名やスキーマ名)を libname ステートメントで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで sakia テーブルから題名(TITLE)に"BIRD"が含まれるレコード を選択した変数 File_id, title, rental_rate, rating を含む WPS データセット WORK.film_BIRD を MySQL の sakila データベースの film_BIRD テーブルへ出力してみましょう。以下のよう にプログラムを書いて実行します。

```
proc dbload data=film_BIRD dbms=ODBC;
    datasrc=MYSQL_SAKIA;
    table=film_BIRD;
    load;
run;quit;
```

※ 注意: DB にアクセスするための権限や ODBC エンジンの接続オプションや libname オ プションがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で 指定を追加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は sakila データベースの city テーブルを読んで、country_id=50 に該当するレコー ドのみ抽出した全カラムを含むテーブル japan_city を作成しています。

```
libname in ODBC datasrc=MYSQL_SAKILA;
proc sql;
  create table in.japan_city as
  select *
  from in.city
  where country_id = 50;
quit;
```

※ この例では、in ライブラリは ODBC 経由で sakila データベースを参照するよう最初の libname ステートメントで定義しています。したがって、SQL プロシジャの中の in ライブ ラリは WPS データライブラリではなく、sakila DB を表すものと解釈され、読み取りも書 き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname in ODBC datasrc=MYSQL_SAKILA;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=in;
   delete japan_city;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data in.japan_city;
   set in.city;
   where country_id=50;
run;
```

4. ORACLE とのインタフェース

[scott サンプルスキーマ]

\$ORACLE_HOME/rdbms/admin/utlsampl.sql を sqlplus から実行して作成される scott ス キーマ上の DEPT テーブルと EMP テーブルを例として用います。

0	Oracle	SQL Developer		N	_ (- >
ファイル(E) 編集(E) 表示(Y) ナビゲート(E <mark>P 🖻 🗟 🗊 🦃 🍽 💿 🔹 📀</mark>	i) 実行(<u>R</u>) チーム(<u>M</u>) ツール(I) - 品 - 総	ウィンドウ(W) ヘルプ(日)		4		
接続 XE ×	DEPT ×					
<mark>≜</mark> scoπ ∋≡ #	 ・ 列 データ 制約 権限付- ・ ・ ・	与 統計 PJガー フラッシュ	バック「依存性」	詳細「パーティション	1984 ISQL	
BONUS	COLUMN_NAME	DATA_TYPE	0 NULLABLE	DATA_DEFAULT	COLUMN_ID COMMENT	5
DEPT	2 DNAME	VARCHAR2(14 BYTE)	No Yes	(null)	2 (null)	
SALGRADE	3 LOC	VARCHAR2 (13 BYTE)	Yes	(null)	3 (null)	
	EMP ×					
↓ (次を含む)	列 データ 制約 権限付・	与 統計 Nガー フラッシュ	バック 依存性	詳細 パーティション	索引 SQL	
ノボート	COLUMN_NAME	E 🚯 DATA_TYPE	I NULLABLE	DATA_DEFAULT	COLUMN_ID COMMENTS	6
オペアのレポート	1 EMPNO	NUMBER(4,0)	No	(null)	1 (null)	
	2 ENAME	VARCHAR2 (10 BYTE)	Yes	(null)	2 (null)	
├ (De TimesTenレポート	3 JOB	VARCHAR2 (9 BYTE)	Yes	(null)	3 (null)	
ー 〒 🕞 データ・ディクショナリ・レポート	4 MGR	NUMBER(4,0)	Yes	(null)	4 (null)	
□ 🕞 データ・モデラー・レポート	5 HIREDATE	DATE	Yes	(null)	5 (null)	
□	6 SAL	NUMBER(7,2)	Yes	(null)	6 (null)	
	7 COMM	NUMBER(7,2)	Yes	(null)	7 (null)	
	8 DEPTNO	NUMBER(2,0)	Yes	(null)	8 (null)	
	4					•

4.1. ネイティブドライバー経由の DB テーブルの入出力

まず、コマンドラインインタフェース(CLIドライバー、ネイティブドライバー)を用いて WPS から DB テーブルの入出力を行ってみましょう。

ネイティブドライバーは、DB と一緒にインストールされます。一般に ODBC ドライバー より高速に DB アクセスができる点でメリットがあると言われています。

WPSのORACLE エンジンはネイティブドライバーを経由してORACLE データベースとア クセスを行う仕組みを提供します。

[SQL 言語を使用した DB 読み取り]

DB へのアクセスは SQL 言語を用いて行うことが多く、WPS でも SQL プロシジャを用い て SQL 言語による DB アクセスが可能です。

SQL 言語を用いてWPS からネイティブドライバー経由でORACLE の DB アクセスを行う 場合は、以下のように、エンジン名として ORACLE を指定し、CONNECT 文のカッコ内に user=ユーザー名、password=パスワード、path=パス の接続オプションや schema=、 dbmax_text=n (文字テキストの読み取り最大長さの設定) などの libname オプションを指 定します。

以下は scott スキーマ上の EMP テーブルから、SAL>2000 に該当するレコードのみ抽出し た全カラムを含む WPS データセット WORK.EMP_SAL_OVER2000 を作成しています。

```
proc sql;
  connect to ORACLE (user=XXX password=password);
  create table EMP_SAL_OVER2000 as
  select *
  from connection to ORACLE
  (
    select *
    from scott.EMP
    where SAL>2000
  );
  disconnect from ORACLE;
  quit;
```

※ 2 行目の user=XXX password=password の XXX と password にはそれぞれユーザ名とパ スワードを指定します。

3 行目の create table 文は WPS データセット WORK.EMP_SAL_OVER2000 を作成する 指定です。

8 行目の from 節で指定した scott.EMP は scott スキーマ上の EMP テーブルを参照しています。

(WPS ワークベンチの実行ログ)

```
8
         proc sql;
           connect to ORACLE (user=XXX password=XXXXXXXX);
9
NOTE: Successfully connected to database ORACLE as alias ORACLE.
           create table EMP_SAL_OVER2000 as
10
11
           select *
12
           from connection to ORACLE
13
           (
             select *
14
15
             from scott.EMP
             where SAL>2000
16
17
            );
NOTE: Data set "WORK.EMP_SAL_OVER2000" has 5 observation(s) and 8 variable(s)
           disconnect from ORACLE;
18
NOTE: Successfully disconnected from database ORACLE.
19
         quit;
NOTE: Procedure sql step took :
     real time : 0.078
     cpu time : 0.031
```

作成された WPS データセット WORK. EMP_SAL_OVER2000 データセットの内容を確認 します。

2	😫 ORACLE1.sas 🖉 Local Server Ø WORK.EMP_SAL_OVER2000.DATA 🛛 🦳 🗖								- 0
	EMPNO	ENAME	JOB	MGR	HIREDATE	a SAL	COMM	DEPTNO	^
1	7566	JONES	MANAGER	7839	02APR1981:00:00:00	2975.00		20	
2	7698	BLAKE	MANAGER	7839	01MAY1981:00:00:00	2850.00		30	
3	7782	CLARK	MANAGER	7839	09JUN1981:00:00:00	2450.00		10	
4	7839	KING	PRESIDENT		17NOV1981:00:00:00	5000.00		10	
5	7902	FORD	ANALYST	7566	03DEC1981:00:00:00	3000.00		20	
		_							×
<									>

※ 注意: ORACLE ドライバーによる DB へのアクセス制限等の設定によっては、SQL プ

ロシジャの CONNECT 文の() 内のオプションに、その他の指定が必要となる場合があり ます。指定可能なオプションは、以下のように、ヘルプビューで確認できます。

まず、ヘルプビュー⁹の左上にある「内容」をクリックします。ヘルプ全体の目次が表示されますので、WPS Reference for Language Elements をクリックします。





Library Engines \rightarrow ORACLE を選択すると、データベース接続オプション (connection option) が表示されます。

⁹ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

(🕲 nư x 🗴 🦾 🍕 🤌 🗸 🖓	
陷 内容 🎾 検索 🗟 関連ドビック 💷 ブックマーク 📖 インデックス	
WPS Reference for Language Elements > WPS Engine for Oracle > Library Engines	~
ORACLE	
LIENAME name ORACLE { connection-option } [{ libname-option }] ;	
▷ LIBNAME ▷ name ▷ ORACLE)	
<pre>connection-option [user = ["] user_name ["] password = ["] password ["] path = ["] oracle_path_specification ["]]</pre>	
₩ + user + = * + user name * * * * *	
lapath be to cracle path specification be to compare the spec	~

これらの接続オプション(user= password= path=)は ORACLE ドライバー経由でのデー タベースアクセスの際の CONNECT 文の DBMS オプション指定としても、また、次に節で 説明する libname ステートメントによる DB 接続オプションとしても用いることができ、必 要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを入力し、実行します。

libname in ORACLE user=XXX password=*password* schema=scott; ※ user=XXX password=*password* の XXX と *password* にはそれぞれユーザ名とパスワー ドを指定します。

この指定は、WPS がスキーマ scott に ORACLE ネイティブドライバー経由で接続し、scott スキーマ上のテーブルとビューをライブラリ名 in で参照できるようにする指定です。(入出 力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設定や、他の接続オプションが必要となる場合があります。

さらに、libname ステートメントには shema=スキーマ名、 dbmax_text=n (文字テキスト の読み取り最大長さの設定) などの libname オプションを追加指定できます。以下のような libname オプションが指定可能です。(前出の ORACLE エンジンのヘルプ画面から確認できます。)



さて、実行すると、ログに 「ライブラリ名 in が割り当てられた」というメッセージが表示されます。このメッセージは ORACLE エンジン経由で DB と接続できたことを表します。

49 libname in ORACLE user=XXX password=XXXXXX schema=scott; NOTE: Library in assigned as follows: Engine: ORACLE Physical Name:

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に scott スキーマ上のテー ブル名が含まれていることがわかります。



※ 右の白抜き矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目 名と項目タイプを表すアイコンが表示されます。)、この scott スキーマにはビューが存在し ませんが、ビュー名の場合には矢印が付きません。

これらのテーブルやビューは、WPSデータセットに対する操作と同じファイル操作が可能です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパティ」を選択すると、カラム名などを確認することができます。

😤 WPS サーバーエ 🕺		۵ プロジェクトエクス 🛚 🧏 ファイ	ันรุงว่าว)									
 Local Server Libraries 					© 27ルタ入力	カラム	ブ	ׂםバティ	(: DEPT			×	•
▲ ■ In ■ BONI ■ DEPT ■ EMP ■ SALC	US - Q	開く	E5		インデックス カラム 詳細 金般	名前 IIII DNAME IIIII LOC IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	タイプ character character number	長さ 28 26 8	フォーマット \$28. \$26. 3.	インフォオーマット \$28. \$26. 3.	ラベル DNAME LOC DEPTNO		^
 Sashelp Sasuser Sasuser Work Filerefs 		ンボー コピー 削除 データヤットをインボート	Ctrl+C 削除			<						>	•
	R	データセットをエクスポート クリップボードに内容をコピー			?	1				[ОК	=+r>tzil	
	B	プロパティ(R)	Alt+Enter										

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> <u>テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS</u> セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる <u>だけ避けてください。</u>

さて、ORACLE エンジンを libname ステートメントで指定した後は、以下のプログラムの ように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ 入力に用いることもできます。この例では、EMP テーブルを入力として職種(JOB)別給 与(SAL) 平均を示す棒グラフを描いています。

goptions cback=white; title "Average Salary by Job"; proc gchart data=in.emp; vbar job/sumvar=SAL type=mean patternid=midpoint mean width=20 space=5; run;quit;



[※] 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、EMP テーブルの中で直属の上司(MGR)が社員名(ENAME)"BLAKE"の社員 番号(EMPNO)を持つ社員のみを抽出した WPS データセット WORK.BLAKE_grp を DATA ステップで作成しています。

data BLAKE_grp;
<pre>if _n_=1 then set in.emp(keep=ENAME EMPNO where=(ENAME="BLAKE")</pre>
rename=(EMPNO=SEARCHNO));
set in.emp;
if MGR=SEARCHNO;
run;

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

① DBLOAD プロシジャ

② SQL プロシジャ

③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名など)を libname ステートメントで指定 しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで EMP テーブルから"BLAKE"を直属の上司とする社員を選択した WPS データセット WORK.BLAKE_grp を ORACLE の scott スキーマ上の BLAKE_GRP テ ーブルへ出力してみましょう。以下のようにプログラムを書いて実行します。

```
proc dbload data=BLAKE_grp dbms=ORACLE;
user=XXX;
password=password;
schema=scott;
table=BLAKE_GRP;
load;
run;quit;
```

※ 注意: DB にアクセスするため ORACLE エンジンの接続オプションや libname オプシ ョンがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で指定 を追加してください。また、user=XXX password=*password*の XXX と *password* にはそれ ぞれユーザ名とパスワードを指定します。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は scott スキーマ上の dept テーブルを読んで、LOC="CHICAGO"に該当するレコ ードのみ抽出した全カラムを含む 1 件のレコードのみ含むテーブル CHICAGO を作成して います。

```
libname in ORACLE user=XXX password=password schema=scott;
proc sql;
  create table in.CHICAGO as
  select *
  from in.dept
  where LOC="CHICAGO";
quit;
```

※ この例では、in ライブラリは ORACLE データベースの scott スキーマを参照するよう最 初の libname ステートメントで定義しています。したがって、SQL プロシジャの中の in ラ イブラリは WPS データライブラリではなく、scott スキーマを表すものと解釈され、読み 取りも書き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname in ORACLE user=XXX password=password schema=scott;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=in;
   delete CHICAGO;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data in.CHICAGO;
   set in.DEPT;
   where LOC="CHICAGO";
run;
```

4.2. ODBCドライバー経由の DB テーブルの入出力

ODBC ドライバーは、Windows 環境や Linux 環境から各種 DB とのインタフェースを共通 方式で行うためのソフトウェアです。ほとんどの DB の ODBC ドライバーが DB ベンダや サードパーティから提供されており、ODBC ドライバーの設定を行えば、DB ごとの細かい アクセス条件等の設定を気にせずに、共通の構文で DB アクセスが実現できるという利点が あります。

[ODBC ドライバのインストール]

[PC 側の ODBC ドライバーの設定]

WPS が導入されている PC 側では、以下のような手順で、ORACLE データベースを ODBC データソースとして登録しておきます。

コントロールパネル → システムとセキュリティ → 管理ツールで ODBC データソースを 選択します。

🖄 l 🗋 👔 🖛 l		管理ツール				×
ファイル ホーム 共有	表示					^ 🕜
□ビー 貼り付け □ビー 貼り付け 記 ショートカ・		* 単 削除 名前の 変更	1000 1000 1000 1000 1000 1000 1000 100	● 届歴	まべて選択 課択解除 課択の切り替え	
クリップボード		整理	新規	開く	遥択	
ⓒ ⊙ - ↑ 🖄 ► ⊐>	/トロール パネル → システムとセキ	キュリティ ト 管理ツー	ı↓ v	♂管理ツ−ル	の検索	٩
▲ 🚖 お気に入り	名前		更新日時	種類	サイズ	^
🎉 ダウンロード	🙈 iSCSI イニシエーター		2013/08/22 15:5	7 ショートカット	2 KB	
■ デスクトップ	📷 ODBC データ ソース (32	Evr)	2013/08/22 8:56	ショートカット	2 KB	
📃 最近表示した場所	📷 ODBC データ ソース (64	Evr)	2013/08/22 15:59) ショートカット	2 KB	
	😿 Windows PowerShell	(x86) N	2013/08/23 0:34	ショートカット	3 KB	
▷ 🜏 ホームグループ	🔝 Windows PowerShell	ISE (x86)	2013/08/22 15:5	5 ショートカット	2 KB	
	🔝 Windows PowerShell	ISE	2013/08/22 15:5	5 ショートカット	2 KB	
Þ 🖳 PC	📷 Windows メモリ診断		2013/08/22 15:53	2 ショートカット	2 KB	
	🔝 イベント ビューアー		2013/08/22 15:5	5 ショートカット	2 KB	
▷ 📬 ネットワーク	🛃 コンピューターの管理		2013/08/22 15:54	+ ショートカット	2 KB	
	🔊 コンポーネント サービス		2013/08/22 15:5	7 ショートカット	2 KB	
	🗟 サービス		2013/08/22 15:54	キーショートカット	2 KB	~
21 個の項目					8	:

(左図)「追加」をクリックします。

(右図) リストにある ORACLE 用 ODBC ドライバーを選択し、「完了」を押します。

ODBC データ ソース アドミニストレーター (64 ビット)	×	データ ソースの新規作成 ×
ユーザー DSN システム DSN ファイル DSN ドライバー トレース 接続プール バージョン情報		
ユーザー データ ソース(U):		セットアップするデータ ソースのドライバーを選択してください(S)
名前 プラットフォーム ドライバー	追加(D) N	名前
Excel Files 64 Eyi+ Microsoft Excel Driver (*.xls, *.xlsr, *.xlsr MS Access Database 64 Eyi+ Microsoft Access Driver (*.mdb, *.accdb)	が 削除(R)	Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx Microsoft Access Driver (*.mdb, *.accdb)
	摄成(C)	Microsoft Access Text Driver (*.bxt, *.csv) Microsoft Excel Driver (*.xts, *.xtsx, *.xtsn, *.xtsb) MySQL ODBC 5.3 ANSI Driver MySQL ODBC 5.3 Unicode Driver ODBC Driver 11 for SQL Server ODBC Driver 11 for SQL Server
C >> OOBC ユーザーデータ ソースには施えされたデータ プロパイダーへの地域方法に従する指載 ー データ ソースは現在のユーザーにのみ認識され、このコンピューターでのみ使用できます。	が褚納されています。ユーザ	< k >
ОК \$ ₹7>セ/	適用(A) へルプ	< 戻る(8) 売了 キャンセル

(左図)新規データソースの名前と説明にはユーザーが自由に入力できます。ここでは、 Data Source Name に「XE」、Description には「ORACLE XE ODBC Driver」と入力します。

「Test Connection」ボタンを押して User Name と Password を入力し、「OK」ボタンを押 して「Connection Successful」のメッセージが出て接続テストが成功することを確認する と設定終了です。

(右図)「OK」を押して ODBC データソースリストに ORACLE 用 ODBC データソース 「XE」が追加されたことを確認します。

	ODBC データ ソース アドミニストレーター (64 ビット)					
		ΠΚ	ユーザー DSN システム D	5N ファイル DS	N ドライバー トレース 接続プール バージョン情報	
Data Source Name	×E		1-9-7-99-X(0):	75.474.4	F= //C	
Description	ORACLE XE ODBC Driver	Cancel	AdventureWorks	32/64 ピット	ODBC Driver 11 for SOL Server	追加(D)
TMC Causian Name		Help	DB2_SAMPLE	32/64 분ット	IBM DB2 ODBC DRIVER - DB2COPY1	削除(R)
User ID		Test Consection	MS Access Database MYSQL_SAKILA	64 ይላዮ 64 ይላዮ 32/64 ይላዮ	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsn Microsoft Access Driver (*.mdb, *.accdb) MySQL ODBC 5.3 ANSI Driver	構成(C)
Application Oracle Enable Result Sets	Workarounds SQLServer Migration : Image: Comparison of the service of the serv	onnection	< ODBC ユー ・ データ ソー ・ データ ソー ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	fー データソース スは現在のユー!	> には指定されたデータプロバイダーへの接続方法に関する信 チーにのみ認識され、アのコンピューターでのみ使用できます。	縦が格納されています。 :
Batch Autocommit	Mode Commit only if all statements succeed				0K +p>/z/L	適用(A)

[SQL 言語を使用した DB 読み取り]

SQL 言語を用いて WPS から ODBC ドライバー経由で ORACLE の DB アクセスを行う場 合は、以下のように、エンジン名として ODBC を指定し、CONNECT 文のカッコ内に user= ユーザー名、password=パスワード、datasrc=ODBC データソース名などの接続オプション や dbmax_text=n (文字テキストの読み取り最大長さの設定) などの libname オプションを 指定します。

下記は、ODBC ドライバーを用いて、scott スキーマ上の EMP テーブルから、SAL>2000 に 該当するレコードのみ抽出した全カラムを含む WPS データセット WORK.EMP_SAL_OVER2000 を作成しています。

```
proc sql;
  connect to ODBC (user=XXX password=password datasrc=XE);
  create table EMP_SAL_OVER2000 as
  select *
  from connection to ODBC
  (
    select *
    from scott.EMP
    where SAL>2000
  );
  disconnect from ODBC;
quit;
```

※ 2 行目の user=XXX password=*password* の XXX と *password* にはそれぞれユーザ名とパ スワードを指定します。

(WPS ワークベンチの実行ログ)

146	proc sql;
147	<pre>connect to ODBC (user=XXX password=XXXXXX datasrc=XE);</pre>
NOTE:	Successfully connected to database ODBC as alias ODBC.
148	create table EMP_SAL_OVER2000 as

149	select *
150	from connection to ODBC
151	(
152	select *
153	from scott.EMP
154	where SAL>2000
155);
NOTE: Dat	a set "WORK.EMP_SAL_OVER2000" has 5 observation(s) and 8 variable(s)
156	disconnect from ODBC;
NOTE: Suc	cessfully disconnected from database ODBC.
157	quit;
NOTE: Pro	cedure sql step took :
real	time : 0.015
CDU	time : 0 000

※ ODBC ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の()内のオプションに、その他の指定が必要となる場合があります。以下の ようにヘルプビューで指定可能なオプションが確認できます。

ヘルプビュー¹⁰の左上にある「内容」をクリックします。

ヘルプ全体の目次が表示されます。WPS Reference for Language Elements をクリックし

ます。



WPS レファレンスを展開し、WPS Engine for ODBC を選択します。

¹⁰ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

陷 内容 🎾 検索 📽 関連トピック 💷 ブックマーク 蘭 ·
スコープ: デフォルト
Workbench User Guide
WPS Reference for Language Elements
How to Read Syntax Diagrams
WPS Core
WPS Graphing
WPS Statistics
WPS Timeseries
WPS Communicate
WPS Engine for Actian Matrix
WPS Engine for DB Files
WPS Engine for DB2
WPS Engine for Greenplum
WPS Engine for Informix
WPS Engine for Kognitio
WPS Engine for MySQL
WPS Engine for NETEZZA
WPS Engine for ODPC
WPS Engine for OL
Use WPS Engine for Oracle
WPS Engine for PostgreSQL
WPS Engine for SAND
US WPS Engine for SQL Server
WPS Engine for Sybase
WPS Engine for Teradata
WPS Engine for Vertica
▷ La Appendix
Legal Notices

その後、展開を何度か行うと接続オプション(connection option) が表示されます。

	15 🛊 🖹 📲 🕖 😓 ▽ 🗖	
陷 内容 🌮 検索 叱 関連トビック 💷 ブックマーク 📖 インデックス		
WPS Reference for Language Elements > WPS Engine for ODBC > Library Engines		~
ODBC		
LIENAME name ODBC { connection-option } [{ libname-option }] ;		
▷ LIBNAME → <i>name</i> → ODBC → connection-option →	ibname-option →	
<pre>connection-option [user = user_name password = password datasrc = DSN datasrc = ["] conn connection_string ["] prompt = ["] connection_string ["] readbuff = n requir</pre>	<pre>tection_string ["] noprompt = ["] ed = ["] connection_string ["]</pre>	I
user = user name	\sim	
password = password	_	
datasrc = DSN	_	
datasrc + = + connection string	-	
hoprompt he connection string	-	
<pre>>> prompt >> => >> >></pre>	-	
readbuff = n	-	
<pre> required += + + + + + + + + + + + + + + + + + +</pre>		~

これらのオプション (user= password= datasrc= noprompt= prompt= readbuff= required=) は ODBC ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプシ

ョン指定としても、また、次に説明する libname ステートメントによるデータベースアクセ スの接続オプションとしても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを実行します。

libname in ODBC datasrc=XE user=XXX password=password schema=scott;

※ user=XXX password=password の XXX と password にはそれぞれユーザ名とパスワード を指定します。

この指定は、WPS がデータソース XE の scott スキーマ上のテーブルに ODBC ドライバー 経由で接続し、ライブラリ名 in で参照できるようにする指定です。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設定や、他の接続オプション (path=) が必要となる場合があります。

さらに、libname ステートメントには schema=スキーマ名、dbmax_text=n(文字テキスト の読み取り最大長さの設定)などの libname オプションを追加指定できます。指定可能な libname オプションは ORACLE エンジンの libname オプションとほぼ同じですが、詳細は 前出の ODBC エンジンのヘルプ画面から確認してください。(Connection-option の下に libname-option が記載されています。)

実行すると、ログに「ライブラリ名 in が割り当てられた」というメッセージが表示されま す。このメッセージは ODBC 経由で DB と接続できたことを表します。

167	libnam	e in ODBC	datasrc=XE	user=XXX	password=XXXXXX	<pre>schema=scott;</pre>
NOTE:	Library i	n assigned	l as follows	:		
	Engine:	ODBC				
	Physical N	lame: XE				

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に scott スキーマ上のテー ブル名が含まれていることがわかります。



※ 右の白抜き矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目 名と項目タイプを表すアイコンが表示されます。)、ビュー名の場合には矢印が付いていま せん。

これらのテーブルやビューは、WPSデータセットに対する操作と同じファイル操作が可能です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパティ」を選択すると、カラム名などを確認することができます。

😤 WPS サーバーエ 🙁	🕒 プロジェクトエクス 🛚 🦹	ファイルエクスプロ	- 0)								
Ocal Server					1 ∃4	7	ロバティ	(: DEPT			- • ×
	5			インデックス カラム 詳細い	名前 回 DNAME	タイプ character	長さ 28	フォーマット \$28.	インフォオーマット \$28.	5/SJL DNAME	^
▷	< ■<			全般	LOC DEPTNO	character number	26 8	\$26. 3.	\$26. 3.	LOC DEPTNO	_
⊳ 🚮 Sashelp ⊳ 🚮 Sasuser	更新 〕 コピー	F5 Ctrl+C									- 1
 Filerefs 	★ 削除 〒-タヤットをインポート	削除			<						×
ſ	 データセットをエクスポー クリップボードに内容をコ 	н ピー		?						ОК	+r>211
	🖀 ברולדי (R)	Alt+Enter									

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> <u>テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS</u> セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる <u>だけ避けてください。</u>

さて、ODBC エンジンを libname ステートメントで指定した後は、以下のプログラムのように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ入力に用いることもできます。この例では、EMP テーブルを入力として職種(JOB)別給与(SAL) 平均を示す棒グラフを描いています。

goptic	ons cback=	=white;		
title	"Average	Salary	by	Job";

proc gchart data=in.emp; vbar job/sumvar=SAL type=mean patternid=midpoint mean width=20 space=5; run;quit;



[※] 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、EMP テーブルの中で直属の上司(MGR)が社員名(ENAME)"BLAKE"の社員 番号(EMPNO)を持つ社員のみを抽出した WPS データセット WORK.BLAKE_grp を DATA ステップで作成しています。

data BLAKE_grp;
<pre>if _n_=1 then set in.emp(keep=ENAME EMPNO where=(ENAME="BLAKE")</pre>
rename=(EMPNO=SEARCHNO));
set in.emp;
if MGR=SEARCHNO;
run;

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

① DBLOAD プロシジャ

② SQL プロシジャ

③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名やスキーマ名)を libname ステートメントで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで EMP テーブルの中で直属の上司 (MGR) が社員名 (ENAME) "BLAKE" の社員番号 (EMPNO) を持つ社員のみを抽出した WPS データセット WORK.BLAKE_grp を ODBC ドライバー経由で XE データソースの scott スキーマ上の BLAKE_GRP テーブルへ出力してみましょう。以下のようにプログラムを書いて実行しま す。

<pre>proc dbload data=BLAKE_grp dbms=ODBC;</pre>
datasrc=XE;
user=XXX;
<pre>password=password;</pre>
<pre>schema=scott;</pre>
<pre>table=BLAKE_GRP;</pre>
load;
run;quit;

※ 注意: user=XXX password=password の XXX と password にはそれぞれユーザ名とパス ワードを指定します。また、DB にアクセスするための権限や ODBC エンジンの接続オプ ションや libname オプションがさらに必要な場合は、load ステートメントの前に、オプシ ョン名=値; の形式で指定を追加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は scott スキーマ上の dept テーブルを読んで、LOC="CHICAGO"に該当するレコ ードのみ抽出した全カラムを含む 1 件のレコードのみ含むテーブル CHICAGO を作成して います。

libname in ODBC datasrc=XE user=XXX password=password schema=scott;
proc sql;

```
create table in.CHICAGO as
select *
from in.dept
where LOC="CHICAGO";
quit;
```

※ この例では、in ライブラリは ODBC 経由で ORACLE データベースの scott スキーマを 参照するよう最初の libname ステートメントで定義しています。したがって、SQL プロシ ジャの中の in ライブラリは WPS データライブラリではなく、ORACLE DB を表すものと 解釈され、読み取りも書き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname in ODBC datasrc=XE user=XXX password=password schema=scott;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=in;
   delete CHICAGO;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data in.CHICAGO;
   set in.DEPT;
   where LOC="CHICAGO";
run;
```

5. PostregSQL とのインタフェース

[dvdrental サンプル DB]

dvdrental データベース¹¹を例として用います。

¹¹ <u>http://www.postgresqltutorial.com/load-postgresql-sample-database/</u>からダウンロードできます。

🕸 pgAdmin III 🛛 – 🗆									
ファイル(F) 編集(E) プラグイン(P) ビュー(V) ツール(T) ヘルプ(H)									
🐔 🛃 💼 🧏 🐼 🔎 🛄 🛃 🌽 🗰 - 🗣 💡									
オブジェクトブラウザ ×	プロパテ	ィー統計情報	属性情報	依存関係		Ŧ			
目 サーハクループ	to ita		(8	r					
PostgreSQL Server (localhost:5432)	20101		10	tor					
🖗 🤤 データベース (2)			1	6432					
e- i dvdrental		-	p	ostgres					
Event Triggers (0)		ル空間	p	g_default					
Extensions (1)	E ACL								
📄 🎨 スキーマ (1)	鳳型								
B- > public			a	ctor_id					
	1001万(相)	真.)	2	00					
全文検索構成 (0)	🗌 編創	ミデータ -Pos	stgreSQL S	erver (loca	alhost:543	2) 🗆 💌			
────────────────────────────────────	ファイル(F) 編集(E)	ビュー(V) ツー	ル(T) ∧ルプ(H)				
□ 金文検索ハーザ(0) □ 金文検索テンプレート(0)		🤊 🦚 🛅	6	' 💡 i 100 ŕ	T V				
●◆ 関数(8)		actor_id	first_name	last_name	last_update	^			
₽ 10 F− <u>Jµ (15)</u>		[PK] Senai		Cuinoga	2012 05 26				
🗎 🖷 🔤 actor	1	2	Nick	Wahlborg	2013-05-26				
	2	3	Ed	Chase	2013-05-26				
i⊕-⊡ city		4	Jennifer	Davis	2013-05-26				
🖻 🛅 country	5	5	Johnny	Lollobrigid	2013-05-26				
e customer	6	6	Bette	Nicholson	2013-05-26				
B fim actor	7	7	Grace	Mostel	2013-05-26				
⊕-🚾 film_category	8	8	Matthew	Johansson	2013-05-26				
inventory	9	9	Joe	Swank	2013-05-26				
in anguage	10	10	Christian	Gable	2013-05-26				
🖶 💼 rental	11	11	Zero	Cage	2013-05-26				
🕀 📠 staff	12	12	Karl	Berry	2013-05-26				
E store	13	13	Uma	Wood	2013-05-26				
E== (7)	14	14	Vivien	Bergen	2013-05-26				
- 🤏 Slonyレブリケーション (0)	15	15	Cuba	Olivier	2013-05-26				
postgres	16	16	Fred	Costner	2013-05-26				
■ 10 アーノル空間(2)	17	17	Helen	Voight	2013-05-26	U U			
⊕-ズ ログインロール (1)	100 行								
テーブルで詳細の回復 actor完了						0.00 秒			

5.1. ネイティブドライバー経由の DB テーブルの入出力

まず、コマンドラインインタフェース(CLIドライバー、ネイティブドライバー)を用いて WPS から DB テーブルの入出力を行ってみましょう。

ネイティブドライバーは、一般に ODBC ドライバーより高速に DB アクセスができる点で メリットがあると言われています。

WPS の POSTGSQL エンジンはネイティブドライバーを経由して PostgreSQL データベー スとアクセスを行う仕組みを提供します。

【重要な注意】PostgreSQLのネイティブドライバー経由のインタフェースの記述部分については、WPS3.1のビルド 678 (03.01.01.00.000678) 以降の対応状況を反映しています。

[SQL 言語を使用した DB 読み取り]

DB へのアクセスは SQL 言語を用いて行うことが多く、WPS でも SQL プロシジャを用い て SQL 言語による DB アクセスが可能です。

SQL 言語を用いて WPS からネイティブドライバー経由で PostgreSQL の DB アクセスを

行う場合は、以下のように、エンジン名として POSTGSQL を指定し、CONNECT 文のカ ッコ内に database=データベース名、user=ユーザー名、password=パスワード などの接続 オプションや shema=スキーマ名、dbmax_text=n (文字テキストの読み取り最大長さの設 定) などの後述する libname オプションを必要に応じて指定します。

以下は dvdrental データベースの public スキーマ上の city テーブルから、country_id=50 に 該当するレコードのみ抽出した全カラムを含む WPS データセット WORK.japan_city を作 成しています。

```
proc sql;
  connect to POSTGSQL (database=dvdrental user=XXX password=password);
  create table japan_city as
  select *
  from connection to POSTGSQL
  (
    select *
    from public.city
    where country_id=50
  );
  disconnect from POSTGSQL;
quit;
```

※ 2 行目の user=XXX password=*password* の XXX と *password* にはそれぞれユーザ名とパ スワードを指定します。

3行目の create table 文は WPS データセット WORK.japan_city を作成する指定です。
8 行目の from 節で指定した public.city は dvdrental データベースの public スキーマ上の city テーブルを参照しています。

(WPS ワークベンチの実行ログ)

```
8
         proc sql;
9
           connect to POSTGSQL (database=dvdrental user=XXX password=XXXXXXX);
NOTE: Successfully connected to database POSTGSQL as alias POSTGSQL.
10
           create table japan_city as
11
           select *
           from connection to POSTGSQL
12
13
             select *
14
             from public.city
15
16
             where country_id=50
17
           );
NOTE: Data set "WORK.japan_city" has 31 observation(s) and 4 variable(s)
           disconnect from POSTGSQL;
18
NOTE: Successfully disconnected from database POSTGSQL.
19
         quit;
NOTE: Procedure sql step took :
     real time : 0.0093
     cpu time : 0.000
```

作成された WPS データセット WORK.staff データセットの内容を確認します。

2 P	OSTGRESQI	L1.sas 🛛 🔳 Lo	ocal Server の W	ORK.JAPAN_CITY.	-	
	eity_id	🧧 city	ountry_id	a last_update		^
1	10	Akishima	50	15FEB2006:09:45:25.000000		
2	172	Fukuyama	50	15FEB2006:09:45:25.000000		
3	203	Higashiosaka	50	15FEB2006:09:45:25.000000		
4	204	Hino	50	15FEB2006:09:45:25.000000		
5	205	Hiroshima	50	15FEB2006:09:45:25.000000		
6	224	Isesaki	50	15FEB2006:09:45:25.000000		
7	226	Iwaki	50	15FEB2006:09:45:25.000000		
8	227	Iwakuni	50	15FEB2006:09:45:25.000000		
9	228	Iwatsuki	50	15FEB2006:09:45:25.000000		
10	229	Izumisano	50	15FEB2006:09:45:25.000000		
11	253	Kakamigahara	50	15FEB2006:09:45:25.000000		
12	256	Kamakura	50	15FEB2006:09:45:25.000000		
13	260	Kanazawa	50	15FEB2006:09:45:25.000000		
14	276	Koriyama	50	15FEB2006:09:45:25.000000		
15	284	Kurashiki	50	15FEB2006:09:45:25.000000		
16	287	Kuwana	50	15FEB2006:09:45:25.000000		
17	331	Matsue	50	15FEB2006:09:45:25.000000		
18	338	Miyakonojo	50	15FEB2006:09:45:25.000000		
19	355	Nagareyama	50	15FEB2006:09:45:25.000000		
20	376	Okayama	50	15FEB2006:09:45:25.000000		
21	377	Okinawa	50	15FEB2006:09:45:25.000000		
22	380	Omiya	50	15FEB2006:09:45:25.000000		
23	382	Onomichi	50	15FEB2006:09:45:25.000000		
24	386	Otsu	50	15FEB2006:09:45:25.000000		
25	440	Sagamihara	50	15FEB2006:09:45:25.000000		
26	463	Sasebo	50	15FEB2006:09:45:25.000000		
27	474	Shimonoseki	50	15FEB2006:09:45:25.000000		
28	521	Tama	50	15FEB2006:09:45:25.000000		
29	547	Tsuyama	50	15FEB2006:09:45:25.000000		
30	552	Ueda	50	15FEB2006:09:45:25.000000		
31	555	Urawa	50	15FEB2006:09:45:25.000000		
						¥
<					>	

※ <u>【重要な注意】WPS3.1.1 の POSTGSQL エンジンでは WPS データセットのデータ型</u> に変換できないデータベースカラムのデータ型 (ENUM 型,mpaa_rating 型, text[]型, tsvector 型) があり、これらのデータ型のカラムは除いて読み取る必要があります。対応としては後 述の ODBC ドライバーを用いる方法もあります。

※ 注意: POSTGSQL ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の()内のオプションに、その他の指定が必要となる場合があ ります。指定可能なオプションは、以下のように、ヘルプビューで確認できます。

まず、ヘルプビュー¹²の左上にある「内容」をクリックします。ヘルプ全体の目次が表示されますので、WPS Reference for Language Elements をクリックします。

10 ~17 2 ▶ 内容 🎾 検索 📽 関連トピック 💵 ブックマーク 贏 インデックス スコープ: デフォルト Workbench User Guide > WPS Reference for Language Elements WPS Workbench User Guide

WPS レファレンスを展開し、WPS Engine for PostgreSQL を選択します。

¹² ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォルダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示されるメニューでも同じです。

BNJ 8		~ - 8						
陷 内容 🎾 検索 📽 関連トビック 💷 ブックマーク 蘭 インデックス								
スコープ: デフォルト								
Workbench User Guide								
WPS Reference for Language Elements								
About this Guide								
How to Read EBNF Notation and Synt	tax Diagrams							
WPS Core								
WPS Graphing								
WPS Statistics								
WPS Timeseries								
WPS Communicate								
WPS Engine for Actian Matrix								
WPS Engine for DB Files								
WPS Engine for DB2								
WPS Engine for Greenplum								
WPS Engine for Informix								
WPS Engine for Kognitio								
WPS Engine for MySQL								
WPS Engine f ETEZZA								
WPS Engine for ODBC								
WPS Engine for OLEDB								
WPS Engine for Oracle								
WPS Engine for PostgreSQL								
WPS Engine for SAND								
WPS Engine for SQL Server								
WPS Engine for Sybase								
WPS Engine for Teradata								
WPS Engine for Vertica								
▷ Liui Appendix								
Legal Notices								
WPS Workbench User Guide								

Library Engines → POSTGRESQL を選択すると、データベース接続オプション

(connection option) が表示されます。



これらの接続オプション (user= password= database= server= port=) は POSTGSQL ドラ イバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプション指定とし ても、また、次に節で説明する libname ステートメントによる DB 接続オプションとしても 用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを入力し、実行します。

libname in POSTGSQL database=dvdrental user=XXX password=password
schema=public;

※ user=XXX password=password の XXX と password にはそれぞれユーザ名とパスワードを指定します。

この指定は、WPS がデータベース dvdrental の public スキーマに POSTGSQL ネイティブ ドライバー経由で接続し、dvdrental データベースの public スキーマ上のテーブルとビュー をライブラリ名 in で参照できるようにする指定です。(入出力両方可能) ただし、DB への 接続やデータテーブルの検索に必要な権限の設定や、他の接続オプションが必要となる場 合があります。

さらに、libname ステートメントには dbmax_text=n(文字テキストの読み取り最大長さの 設定)などの libname オプションを追加指定できます。以下のような libname オプション が指定可能です。(前出の POSTGSQL エンジンのヘルプ画面から確認できます。)



さて、実行すると、ログに 「ライブラリ名 in が割り当てられた」というメッセージが表

示されます。このメッセージは POSTGSQL エンジン経由で DB と接続できたことを表します。

```
29 libname in POSTGSQL database=dvdrental user=XXX password=XXXXXXX schema=public;
NOTE: Library in assigned as follows:
Engine: POSTGRESQL
Physical Name:
```

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に dvdrental データベース の public スキーマ上のテーブル名およびビュー名が含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。



 ※ 【重要な注意 1】WPS3.1.1 の POSTGSQL エンジンでは WPS データセットのデータ型 に変換できないデータ型(ENUM 型,mpaa_rating 型, text[]型, tsvector 型)があり、これら のデータ型のカラムを含むテーブルやビューは、アクセスエラーが発生し読み取りはでき ません。対応としては後述の ODBC ドライバーを用いてください。
 ※ 【重要な注意 2】:テーブル名をダブルクリックすると、データセットビューアが起動し、 テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる

さて、POSTGSQL エンジンを libname ステートメントで指定した後は、以下のプログラム のように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジ ャ入力に用いることもできます。この例では、customer_list ビューを入力として国(country) 別の登録顧客数の棒グラフを描いています。

goptions cback=white; title "Customer number by Country"; proc gchart data=in.customer_list; pie country; run;quit;

だけ避けてください。



※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、customer_list ビューから国名(country)を大文字化した値が"GREECE"のレコ ードを選択した変数 id, name, address, city, country を含む WPS データセット WORK.custromer_greece を DATA ステップで作成しています。

```
data customer_greece;
  set in.customer_list(keep=id name address city country);
  where upcase(country)="GREECE";
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

① DBLOAD プロシジャ

前節の最後のプログラムで customer_list ビューから国名 (country) を大文字化した値が "GREECE"のレコードを選択した変数 id, name, address, city, country を含む WPS データ セット WORK.custromer_greece を POSTGRESQL の dvdrental データベースの public ス キームの customer_greece テーブルへ出力してみましょう。以下のようにプログラムを書 いて実行します。

```
proc dbload data=customer_greece dbms=POSTGSQL;
  user=XXX;
  password=password;
  database=dvdrental;
  host=localhost;
  port=5432;
  table=public.customer_greece;
  load;
run;quit;
```

※ 注意: DB にアクセスするため POSTGSQL エンジンの接続オプションや libname オプ ションがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で指 定を追加してください。また、user=XXX password=password の XXX と password にはそ れぞれユーザ名とパスワードを指定します。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は dvdrental データベースの public スキーム上の city テーブルを読んで、 country_id=50 に該当するレコードのみ抽出した全カラムを含むテーブル japan_city を作成 しています。

```
libname in POSTGSQL database=dvdrental user=XXX password=password
schema=public;
proc sql;
  create table in.japan_city as
  select *
  from in.city
  where country_id=50;
quit;
```

※ この例では、in ライブラリは dvdrental データベースの public スキームを参照するよう 最初の libname ステートメントで定義しています。したがって、SQL プロシジャの中の in ライブラリは WPS データライブラリではなく、dvdrental データベースの public スキーム を表すものと解釈され、読み取りも書き出しも DB テーブルが対象になります。

【重要な注意】現時点では、実行後、再度 libname ステートメントを実行してください。そうでないと、データセットの内容やプロパティを確認できないという問題があります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname in POSTGSQL database=dvdrental user=XXX password=password
schema=public;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=in;
   delete japan_city;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data in.japan_city;
   set in.city;
   where country_id=50;
run;
```

【重要な注意】現時点では、実行後、再度 libname ステートメントを実行してください。そうでないと、データセットの内容やプロパティを確認できないという問題があります。

5.2. ODBC ドライバー経由の DB テーブルの入出力

ODBC ドライバーは、Windows 環境や Linux 環境から各種 DB とのインタフェースを共通 方式で行うためのソフトウェアです。ほとんどの DB の ODBC ドライバーが DB ベンダや サードパーティから提供されており、ODBC ドライバーの設定を行えば、DB ごとの細かい アクセス条件等の設定を気にせずに、共通の構文で DB アクセスが実現できるという利点が あります。

[ODBC ドライバのインストール]

PostgreSQL の ODBC ドライバーは PostgreSQL の動作環境とバージョンに合わせてイン ストールします。 もしも、次の [PC 側の ODBC ドライバーの設定] で PostgreSQL の ODBC ドライバーが見当たらない場合は、インストール環境を確認してください。

[PC 側の ODBC ドライバーの設定]

WPS が導入されている PC 側では、以下のような手順で、PostgreSQL の dvdrental データ ベースを ODBC データソースとして登録しておきます。

コントロールパネル → システムとセキュリティ → 管理ツールで ODBC データソースを 選択します。
 ② □ = □ ○ □ = □ ○ □ = □ 						
ファイル ホーム 共有	表示					^ 🕜
 ・ 「 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	- 移動先:	『 ··· ·· ·· ·· ·· ·· ·· ·· ·· ·	1000 日本 新しい フォルダー ブ	□パティ ● 履歴	 ■ すべて選択 ◎ 選択解除 ■ 選択の切り替え 	
クリップボード		整理	新規	開く	選択	
(⊙ ⊙ - ↑ (₫ + ⊐>	トロール パネル 🕨 システ	ムとセキュリティ → 管理ツー	-JL v	♂管理ツ−ノ	しの検索	P
▲☆ お気に入り	名前	*	更新日時	種類	サイズ	^
🔰 ダウンロード	🔒 iSCSI イニシエータ	-	2013/08/22 15:5	7 ショートカット	2 KB	
📰 デスクトップ	📷 ODBC データソー	ス (32 ビット)	2013/08/22 8:56	ショートカット	2 KB	
📃 最近表示した場所	📷 ODBC データソー	ス (64 ビット)	2013/08/22 15:5	9 ショートカット	2 KB	
	😿 Windows Power	Shell (x86) 🔨	2013/08/23 0:34	ショートカット	3 KB	
▷ 🜏 ホームグループ	📷 Windows Power	Shell ISE (x86)	2013/08/22 15:5	5 ショートカット	2 KB	
	Windows PowerShell ISE			5 ショートカット	2 KB	
D 🛤 PC	📷 Windows メモリ診	断	2013/08/22 15:5	2 ショートカット	2 KB	
	🛃 イベント ビューアー		2013/08/22 15:5	5 ショートカット	2 KB	
▷ 📬 ネットワーク	🛃 コンピューターの管理	Ē	2013/08/22 15:5	4 ショートカット	2 KB	
	🄊 コンポーネント サー	2X	2013/08/22 15:5	7 ショートカット	2 KB	
	🗟 サービス		2013/08/22 15:5	4 ショートカット	2 KB	~
21 個の項目					8	

(左図)「追加」をクリックします。

(右図) リストにある PostgreSQL 用 ODBC ドライバーを選択し、「完了」を押します。

ODBC データ ソース アドミニストレーター (64 ビット)	×	データ ソースの新規作成
ユーザー DSN システム DSN フィル DSN ドライバー トレース 接続ブール パーション構築 ユーザー テータ ソース(U): 名称 ブラットフォーム ドライバー Excel Files 64 とット Microsoft Excel Driver (*.xis, *.xisx, *.xist Microsoft Access Driver (*.mdb, *.accdb)	通加(D) 利除(R) 模成(C)	セットアップするテータ ソースのドライバーを選択してください(S) 名前 Microsoft Access Text Driver (*.btt, *.csv) Microsoft Excel Driver (*.btt, *.csv) MySQL ODBC 5:3 A MiSD Driver MySQL ODBC 5:3 Unicode Driver MySQL ODBC 5:3 Unicode Driver ODBC Driver I for SQL Server
COBC ユーザーデータ ソースには増走されたデータ プルバイダーへの抽想方法に取する情報 のOBC ユーザーデータ ソースには増走されたデータ プルバイダーへの抽想方法に取する情報 ー データ ソースは現在のユーザーにのみ回識だれ、20コンピューターでのみ使用できず。 OK キャンセル ; ;	Alekken Tいほす。ユーザ 恵用(A) へルプ	Code in XE PrestyreSQL ANSI/VSQL PostgreSQL Unicode(xSA) く く 反(6) 完了 キャンセル

(左図) データソース名と説明にはユーザーが自由に入力できます。ここでは、データソー ス名は「PostgreSQL30」(既定のまま)、説明には「PostgreSQL dvdrental database」と入 力します。データベース名には「dvdrental」、ユーザー名とパスワードを入力します。次に 「テスト」ボタンを押し接続テストを行います。接続テストが成功することを確認すると 「保存」ボタンを押し設定終了です。

(右図)ODBCデータソースリストにPostgreSQL用ODBCドライバーが「PostgreSQL30」 というデータソース名で追加されたことを確認します。

	PostgreSQL ANSI ODBC セットフ	Pップ ×	-		ODBC	データ ソース アドミニストレーター (64 ビッ	×۱)
データソース名: (N)	PostgreSQL30		E	ーザー DSN システム DS ユーザー データ ソース(U):	in ファイル DS	N ドライバー トレース 接続プール バージョン情報	E .
説明:(D)	PostgreSQL dvdrental database	テスト		名前	プラットフォーム	ドライバー	追加(D)
SSL Mode:(L)	無効 >	保存		AdventureWorks DB2_SAMPLE Excel Files	32/64 ビット 32/64 ビット 64 ビット	ODBC Driver 11 for SQL Server IBM DB2 ODBC DRIVER - DB2COPY1 Microsoft Excel Driver (*.xis, *.xisr, *.xisr	削除(R)
サーバー名: (S) データベース名: (b)	localhost dvdrental	キャンセル Port: 5432		MS Access Database MYSQL_SAKILA PostgreSQL30 XE	64 ビット 32/64 ビット 64 ビット 64 ビット	Microsoft Access Driver (*.mdb, *.accdb) MySQL ODBC 5.3 ANSI Driver PostgreSQL ANSI(x64) Oracle in XE	構成(C)
- 既定の認証 — ユーザー名: (U) バスワード:	000(オプション(高度な設定) データソース 全(特設定		く ODBC ユーサ ーデータソー	「- データソース 7は現在のコー	> には指定されたデータプロバイダーへの接続方法に開す 「ニアのよび時まれ、アのコンピューターアのみ使用できす	る情報が格納されています。ユーザ Fat
(w) PostgreS In the o	OL Ver7.8 Copyright (C) 1998-2006; Insight D riginal form, Japanese patch Hiroshi-saito	listribution Systems				ОК ‡ туди	通用(A) へルプ

[SQL 言語を使用した DB 読み取り]

SQL 言語を用いて WPS から ODBC ドライバー経由で PostgreSQL の DB アクセスを行う 場合は、以下のように、エンジン名として ODBC を指定し、CONNECT 文のカッコ内に datasrc=ODBC データソース名などの接続オプションや dbmax_text=n (文字テキストの読 み取り最大長さの設定) などの libname オプションを必要に応じて指定します。

下記は、ODBC ドライバーを用いて、dvdrental データベースの public スキーマ上の city テ ーブルから、country_id=50 に該当するレコードのみ抽出した全カラムを含む WPS データ セット WORK.japan_city を作成しています。

```
proc sql;
  connect to ODBC (datasrc=PostgreSQL30);
  create table japan_city as
  select *
  from connection to ODBC
  (
    select *
    from city
    where country_id=50
  );
  disconnect from ODBC;
quit;
```

(WPS ワークベンチの実行ログ)

```
8
         proc sql;
9
           connect to ODBC (datasrc=PostgreSQL30);
NOTE: Successfully connected to database ODBC as alias ODBC.
           create table japan_city as
10
           select *
11
12
           from connection to ODBC
13
           (
14
             select *
15
             from public.city
16
             where country_id=50
17
           );
NOTE: Data set "WORK.japan_city" has 31 observation(s) and 4 variable(s)
```

```
18 disconnect from ODBC;
NOTE: Successfully disconnected from database ODBC.
19 quit;
NOTE: Procedure sql step took :
    real time : 0.093
    cpu time : 0.015
```

※ ODBC ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の()内のオプションに、その他の指定が必要となる場合があります。以下の ようにヘルプビューで指定可能なオプションが確認できます。

ヘルプビュー13の左上にある「内容」をクリックします。

ヘルプ全体の目次が表示されます。WPS Reference for Language Elements をクリックし



WPS レファレンスを展開し、WPS Engine for ODBC を選択します。



その後、展開を何度か行うと接続オプション(connection option) が表示されます。

¹³ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

	To 🍰 🎂	₩ ⊿ 🗢 🔿	~
陷 内容 🂖 検索 📽 関連トビック 💷 ブックマーク 📖 インデックス			
WPS Reference for Language Elements > WPS Engine for ODBC > Library Engines			~
ODBC			
LIENAME name ODBC { connection-option } [{ libname-option }] ;			
▷ LIBNAME → name → ODBC → connection-option → →	⊲ ibname-opti	ion p	;}₩
<pre>connection-option [user = user_name password = password datasrc = DSN datasrc = ["] conn connection_string ["] prompt = ["] connection_string ["] readbuff = n requir</pre>	nection_string [ed = ["] conn	"] nopromp ection_string['	t = ["] "]]
► v user = user name			
	1		
prompt = + + connection string + + m + + m + + + + + + + + + + + + +	-		
required by = by by connection string by	1		
			~

これらのオプション(user= password= datasrc= noprompt= prompt= readbuff= required=) は ODBC ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプシ ョン指定としても、また、次に説明する libname ステートメントによるデータベースアクセ スの接続オプションとしても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを実行します。

libname in ODBC datasrc=PostgreSQL30 schema=public;

この指定は、WPS がデータソース PostgreSQL30 に ODBC ドライバー経由で接続し、ライ ブラリ名 in で参照できるようにする指定です。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限の設定や、他の接続オプション が必要となる場合があります。

さらに、libname ステートメントには schema=スキーマ名、dbmax_text=n(文字テキストの読み取り最大長さの設定)などの libname オプションを追加指定できます。指定可能な libname オプションは POSTGSQL エンジンの libname オプションとほぼ同じですが、詳細

は前出の ODBC エンジンのヘルプ画面から確認してください。(Connection-option の下に libname-option が記載されています。)

実行すると、ログに「ライブラリ名 in が割り当てられた」というメッセージが表示されま す。このメッセージは ODBC 経由で DB と接続できたことを表します。

```
345 libname in ODBC datasrc=PostgreSQL30 schema=public;
NOTE: Library in assigned as follows:
Engine: ODBC
Physical Name: PostgreSQL30
```

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、in ライブラリがあることを確認してください。この中に dvdrental データベース の public スキーマ上のテーブル名およびビュー名が含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。



※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> <u>テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS</u> セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる <u>だけ避けてください。</u>

さて、ODBC エンジンを libname ステートメントで指定した後は、以下のプログラムのように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジャ入力に用いることもできます。この例では、customer_list ビューを入力として国(country)別の登録顧客数の棒グラフを描いています。

```
goptions cback=white;
title "Customer number by Country";
proc gchart data=in.customer_list;
    pie country;
run;quit;
```



※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、customer_list ビューから国名(country)を大文字化した値が"GREECE"のレコ ードを選択した変数 id, name, address, city, country を含む WPS データセット WORK.custromer_greece を DATA ステップで作成しています。

```
data customer_greece;
  set in.customer_list(keep=id name address city country);
  where upcase(country)="GREECE";
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名やスキーマ名)を libname ステートメントで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで sakia テーブルから題名(TITLE)に"BIRD"が含まれるレコード を選択した変数 File_id, title, rental_rate, rating を含む WPS データセット WORK.film_BIRD を MySQL の film_BIRD テーブルへ出力してみましょう。以下のようにプログラムを書いて 実行します。

```
proc dbload data=customer_greece dbms=ODBC;
    datasrc=PostgreSQL30;
    table=public.customer_greece;
    load;
run;quit;
```

※ 注意: DB にアクセスするための権限や ODBC エンジンの接続オプションや libname オ プションがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で 指定を追加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は sakila データベースの city テーブルを読んで、country_id=50 に該当するレコードのみ抽出した全カラムを含むテーブル japan_city を作成しています。

```
libname in ODBC datasrc=PostgreSQL30 schema=public;
proc sql;
  create table in.japan_city as
  select *
  from in.city
  where country_id = 50;
quit;
```

※ この例では、in ライブラリは ODBC 経由で dvdrental データベースの public スキーマを 参照するよう最初の libname ステートメントで定義しています。したがって、SQL プロシ ジャの中の in ライブラリは WPS データライブラリではなく、DB を表すものと解釈され、 読み取りも書き出しも DB テーブルが対象になります。 ③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname in ODBC datasrc=PostgreSQL30 schema=public;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=in;
   delete japan_city;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data in.japan_city;
   set in.city;
   where country_id=50;
run;
```

6. SQL Server とのインタフェース

[AdventureWorks2012 サンプル DB]

SQL Server 2012 Express Edition に アタッチした Microsoft 提供の AdventureWorks2012 サンプル DB¹⁴を例として用います。

SQL Server Managenent Studio で AdventureWorks2012 DB をアタッチし、内容を確認してください。

(AdventureWorks2012 データベースのテーブル一覧と Person.Address テーブルの内容)

¹⁴ <u>http://msftdbprodsamples.codeplex.com/releases/view/55330</u> の DOWNLOADS タブから <u>AdventureWorks2012</u> <u>Data File</u> をダウンロードします。



6.1. ネイティブドライバー経由の DB テーブルの入出力

まず、コマンドラインインタフェース(CLIドライバー、ネイティブドライバー)を用いて WPS から DB テーブルの入出力を行ってみましょう。

ネイティブドライバーは DB と一緒にインストールされており、一般に ODBC ドライバー より高速に DB アクセスができる点でメリットがあると言われています。

WPS の SQLSERVR エンジンはネイティブドライバーを経由して SQL Server データベー スとアクセスを行う仕組みを提供します。

[SQL 言語を使用した DB 読み取り]

DB へのアクセスは SQL 言語を用いて行うことが多く、WPS でも SQL プロシジャを用い て SQL 言語による DB アクセスが可能です。

SQL 言語を用いて WPS からネイティブドライバー経由で SQL Server の DB アクセスを 行う場合は、以下のように、エンジン名として SQLSERVR を指定し、CONNECT 文のカ ッコ内に server=サーバー名、database=DB 名などの接続オプションや後述する dbmax_text=n (文字テキストの読み取り最大長さの設定) などの libname オプションを指 定します。

以下は AdventureWorks2012 データベースの Sales.Customer テーブルから、TerritoryID=1 に該当するレコードのみ抽出した全カラムを含む WPS データセット WORK.Customer を 作成しています。

```
proc sql;
  connect to SQLSERVR(server="PC3¥SQLEXPRESS" database=AdventureWorks2012);
  create table Customer as
  select *
  from connection to SQLSERVR
  (
    select *
    from Sales.Customer
    where TerritoryID = 1;
  );
  disconnect from SQLSERVR;
  quit;
```

※ 3 行目の create table 文は WPS データセット WORK.Customer を作成する指定です。
 8 行目の from 節で指定した Sales.Customer は AdventureWorks2012 データベースの sales スキーマ上の Customer テーブルを参照しています。

(WPS ワークベンチの実行ログ)

8	proc sql;
9	connect to SQLSERVR(server="DMT-PC3¥SQLEXPRESS" database=AdventureWorks2012);
NOTE:	Successfully connected to database SQLSERVR as alias SQLSERVR.
10	create table Customer as
11	select *
12	from connection to SQLSERVR
13	(
14	select *
15	from Sales.Customer
16	where TerritoryID = 1;
17);
NOTE:	Data set "WORK.Customer" has 3520 observation(s) and 7 variable(s)
18	disconnect from SQLSERVR;
NOTE:	Successfully disconnected from database SQLSERVR.
19	quit;
NOTE:	Procedure sql step took :
	real time : 0.062
	cpu time : 0.031

作成された WPS データセット WORK.Customer データセットの内容を確認します。

💷 Loc	🖩 Local Server Ø WORK.CUSTOMER.DATA 🛛 🗖 🗖							
	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	^	
1		934	1	AW0000001	3F5AE95E-B87D-4AED-95B4-C3797AFCB74F	130CT2008:11:15:07	_	
2		1028	1	AW0000002	E552F657-A9AF-4A7D-A645-C429D6E02491	130CT2008:11:15:07		
3		930	1	AW0000007	03E9273E-B193-448E-9823-FE0C44AEED78	130CT2008:11:15:07		
4		430	1	AW0000019	69AE5D43-31BE-4B76-BFBB-5A23C4788BBC	130CT2008:11:15:07		
5		1016	1	AW0000020	E010C10A-F1C3-4BBA-81CA-A7E083350400	130CT2008:11:15:07		
6		938	1	AW0000037	C6EBB29A-CC67-459C-90E3-339E0F912906	130CT2008:11:15:07		
7		1004	1	AW0000038	E5EDA3F3-4EF1-4806-BD36-E61F3CB2E044	130CT2008:11:15:07		
8		1290	1	AW0000043	9E449D3E-5D79-4F65-A4F1-10DD3B0E0766	130CT2008:11:15:07		
9		1282	1	AW0000055	F5D3997B-AD06-46A0-94AF-A4460F4BE471	130CT2008:11:15:07		
10		992	1	AW0000056	570B1682-9BB1-4359-984D-8F728E3D6973	130CT2008:11:15:07		
11		1270	1	AW0000073	9D5C135E-BD60-46E0-B371-843391637827	130CT2008:11:15:07		
12		528	1	AW00000074	73485E57-076D-4B10-BB48-6043DD873582	130CT2008:11:15:07		
13		1258	1	AW0000091	9F314FC3-9644-4412-A90A-5483F75F665R	130CT2008-11-15-07	×	
<							>	

※ 注意: SQLSERVR ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の() 内のオプションに、その他の指定が必要となる場合があ ります。指定可能なオプションは、以下のように、ヘルプビューで確認できます。

まず、ヘルプビュー¹⁵の左上にある「内容」をクリックします。ヘルプ全体の目次が表示されますので、WPS Reference for Language Elements をクリックします。



WPS レファレンスを展開し、WPS Engine for SQL Server を選択します。

() へいプ X	🗐 🗘 🌳 🔽 🗖					
▶ 内容 ≫ 検索 № 関連トピック	💵 ブックマーク 篇 インデックス					
スコープ: デフォルト						
> 🌸 Workbench User Guide						
WPS Reference for Language	e Elements					
About this Guide						
How to Read EBNF Notat	tion and Syntax Diagrams					
WPS Core						
WPS Graphing						
WPS Statistics						
WPS Timeseries						
WPS Communicate						
WPS Engine for Actian Ma	latrix					
WPS Engine for DB Files						
WPS Engine for DB2						
WPS Engine for Greenplu	um					
WPS Engine for Informix						
WPS Engine for Kognitio						
WPS Engine for MySQL						
▷ WPS Engine for NETEZZA	A					
▷ IIII WPS Engine for ODBC						
B WPS Engine for OLEDB						
WPS Engine for Oracle NPS Engine for Destars	201					
WPS Engine for Postgres	SQL .					
WPS Engine for SOL Sen	Ver					
WPS End	Ver					
WPS Engine for Teradata						
WPS Engine for Vertica						
Appendix						
Legal Notices						
WPS Workbench User Guide	2					
· •						

Library Engines \rightarrow SQLSERVR を選択すると、データベース接続オプション (connectionoption) が表示されます。

¹⁵ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

BNJ 3	To 🕆 🔶 🚺 🖋 🏷 🔍 To t
陷 内容 🌮 検索 📽 関連トビック 💷 ブックマーク 📾 インデックス	
WPS Reference for Language Elements > WPS Engine for SQL Server > Library Engines	
SQLSERVR	
<pre>LIBNAME name SQLSERVR { connection-option } [{ libname-option }] ;</pre>	
► LIBNAME → name → SQLSERVR → connection-option →	libname-option
<pre>connection-option [user = user_name password = password server = Server_name database = data driver = driver_name_to_place_in_connection_string]</pre>	base_name readbuff = n port = n
<pre>>> User = user name >> password = password >> server = server name</pre>	
database = database name	
$\begin{array}{c} & & \\$	
driver = driver name to place in connection string	

これらの接続オプション(user= password= server= database= readbuff= port= driver=) は SQLSERVR ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプ ション指定としても、また、次に節で説明する libname ステートメントによる DB 接続オプ ションとしても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを入力し、実行します。

libname person SQLSERVR server="PC3¥SQLEXPRESS" database=AdventureWorks2012
schema=Person;
libname sales SQLSERVR server="PC3¥SQLEXPRESS" database=AdventureWorks2012
schema=Sales;

この指定は、WPS がデータベース AdventureWorks2012 に SQLSERVR ネイティブドライ バー経由で接続し、Person スキーマ上のテーブルとビューをライブラリ名 person で、Sales スキーマ上のテーブルとビューをライブラリ名 sales で、それぞれ参照できるようにする 指定です。(入出力両方可能) ただし、DB への接続やデータテーブルの検索に必要な権限 の設定や、他の接続オプションが必要となる場合があります。

なお、server=サーバー名、database=DB 名指定は接続オプション、schema=指定は libname オプションです。libname オプションには他に、dbmax_text=n(文字テキストの読み取り最 大長さの設定)などがあり、以下のような libname オプションが指定可能です。



さて、実行すると、ログに person と sales の2個の「ライブラリ名が割り当てられた」 というメッセージが表示されます。このメッセージは SQLSERVR エンジン経由で DB と接 続できたことを表します。

29 libname person SQLSERVR server="DMT-PC3¥SQLEXPRESS" database=AdventureWorks2012 schema	1
29 ! =Person;	
NOTE: Library person assigned as follows:	
Engine: SQLSERVR	
Physical Name: AdventureWorks2012	
30 libname sales SQLSERVR server="DMT-PC3¥SQLEXPRESS" database=AdventureWorks2012 schema=	:
30 ! Sales;	
NOTE: Library sales assigned as follows:	
Engine: SQLSERVR	
Physical Name: AdventureWorks2012	

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、Person ライブラリと Sales ライブラリがあることを確認してください。これら の中に AdventureWorks2012 データベースの Person スキーマ、Sales スキーマ上のテーブ ル名およびビュー名がそれぞれ含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。

Ъ プロジェクトェクス 🕆 🎦 ファイルエ	クスプロ 😤 WPS サーバーエ	8 - 0							
Local Server Libraries A Person			G		プロパティ・	Address			- • ×
 Address Address 	開く		741.9入力	አጋፖ	201011				↔ • ⇒ • •
Busines Busines Busines Busines Controt Country Berson Berson StateProvin StateProvin viditionalC	更新 コピー 削除 データセットをインボート データセットをインボート クリップボートに内容をコピー プロ/57(R) ContactInfo	F5 Ctrl+C 削除 Alt+Enter	インデックス 1551 第三 金殿	之前 ④ AddressUne3 ⑤ AddressUne3 ⑥ AddressUne2 ⑥ Orty ④ ModifieDate ⑤ PostalCode ⑥ rorguid ⑥ spatialLocation ④ StateProvinceID	947 number character character character character character character number	 長さ フォーマット 8 11. 60 \$60. 300 \$30. 8 DATETIME20. 15 \$15. 36 \$36. 4000 \$HEX4000. 8 11. 	インフォオーマット 11. \$60. \$30. DATETIME20. \$15. \$36. \$HEX4000. 11.	5∿IJ AddressID AddressLine1 AddressLine2 City ModifiedDate PostalCode rowguid SpatialLocation StateProvinceID	^
 > Sales > Sales > Sashelp > Sasuser > Work > ■ A > ■ A 	,,		0	<				ОК	> \$7771

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> <u>テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS</u> セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる <u>だけ避けてください。</u>

さて、SQLSERVR エンジンを libname ステートメントで指定した後は、以下のプログラム のように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシジ ャ入力に用いることもできます。この例では、Sales.Salesperson テーブルを入力としてビ ジネスエリア別の売り上げ高の円グラフを描いています。

```
goptions cback=white;
title "Pie Chart of SalesYTD by SalesPersonID";
proc gchart data=sales.salesperson;
    pie businessentityid/sumvar=salesYTD;
run;
```



※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを読むことができるようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可能になるというメリットがあります。

下の例は、Person.Address テーブルから 3 つの都市名のいずれかに該当するレコードを選 択した WPS データセット WORK.address_subset を DATA ステップで作成しています。

```
data address_subset;
  set person.address(keep=AddressID AddressLine1 City);
  where City in ('San Francisco', 'Los Angeles', 'Seattle');
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法で WPS データセットを DB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に DB への接続に用いるエンジン名と接続先(DB 名やスキーマ名)を libname ステートメン

トで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで Person.Address テーブルから 3 つの都市名のいずれかに該当す るレコードを選択した WPS データセット WORK.address_subset を SQL Server の Person スキーマ上の Address_Subset テーブルへ出力してみましょう。以下のようにプログラムを 書いて実行します。

```
proc dbload data=address_subset dbms=SQLSERVR;
server="PC3¥SQLEXPRESS";
database=AdventureWorks2012;
table=Person.Address_Subset;
load;
run;quit;
```

※ 注意: DB にアクセスするため SQLSERVR エンジンの接続オプションや libname オプ ションがさらに必要な場合は、load ステートメントの前に、オプション名=値; の形式で指 定を追加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントでエンジン名を指定すると、DB テーブルを WPS データセットと 同じように扱えるようになります。

以下の例は AdventureWorks2012 データベースの Sales.Customer テーブルを読んで、 TerritoryID=1 に該当するレコードのみ抽出した全カラムを含むテーブル Sales.Customer_Territory_1 を作成しています。

```
libname sales SQLSERVR server="PC3¥SQLEXPRESS" database=AdventureWorks2012
schema=Sales;
proc sql;
  create table Sales.Customer_Territory_1 as
  select *
  from Sales.Customer
  where TerritoryID = 1;
quit;
```

※ この例では、Sales ライブラリは SQL Server DB の Sales スキーマを参照するよう最初 の libname ステートメントで定義しています。したがって、SQL プロシジャの中の Sales ライブラリは WPS データライブラリではなく、AdventureWorks2012 DB の Sales スキー マを表すものと解釈され、読み取りも書き出しも DB テーブルが対象になります。 ③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname sales SQLSERVR server="PC3¥SQLEXPRESS" database=AdventureWorks2012
schema=Sales;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=sales;
   delete Customer_Territory_1;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data Sales.Customer_Territory_1;
   set Sales.Customer;
   where TerritoryID = 1;
run;
```

6.2. ODBCドライバー経由の DB テーブルの入出力

ODBC ドライバーは、Windows 環境や Linux 環境から各種 DB とのインタフェースを共通 方式で行うためのソフトウェアです。ほとんどの DB の ODBC ドライバーが DB ベンダや サードパーティから提供されており、ODBC ドライバーの設定を行えば、DB ごとの細かい アクセス条件等の設定を気にせずに、共通の構文で DB アクセスが実現できるという利点が あります。

[ODBC ドライバのインストール]

DB ベンダーのサイトから ODBC ドライバーをダウンロードし、インストールしておきま す。ここでは、Microsoft のサイトから SQL Server の ODBC ドライバーをダウンロードし ています。



[PC 側の ODBC ドライバーの設定]

WPS が導入されている PC 側では、以下のような手順で、上記 SQL Server 2012 の AdventureWorks2012 データベースを ODBC データソースとして登録しておきます。

コントロールパネル → システムとセキュリティ → 管理ツールで ODBC データソース を選択します。

② □ □ □						
ファイル ホーム 共有	表示				^ 🔞	
▲ 切り取り □ ▲ 切り取り □ □ □ ▲ 切り取り □ □ □ □ □ □ □		★ 単数 2500 (100 km)	 新しい フォルダー 	プロパティ 日本 周く → 日本	 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
クリップボード		整理	新規	開く	遥択	
④ ④ ▼ ↑ 徳 > コントロール パネル > システムとセキュリティ > 管理ツール				✔ 0 管理ツール	の検索 ,0	
▲ 🗙 お気に入り	名前		更新日時	種類	サイズ ^	
🍺 ダウンロード	🔝 iSCSI イニシエーター		2013/08/22 15	:57 ショートカット	2 KB	
📰 デスクトップ	📆 ODBC データ ソース (32 ビット)	2013/08/22 8:	56 ショートカット	2 KB	
📃 最近表示した場所	📷 ODBC データ ソース (64 ビット)	2013/08/22 15	:59 ショートカット	2 KB	
	😿 Windows PowerSh	ell (x86) 🔨	2013/08/23 0:3	34 ショートカット	3 KB	
▷ 🔩 ホームグループ	🔝 Windows PowerSh	ell ISE (x86)	2013/08/22 15	:55 ショートカット	2 KB	
	🔝 Windows PowerSh	ell ISE	2013/08/22 15	:55 ショートカット	2 KB	
⊳ 🖳 PC	📷 Windows メモリ診断		2013/08/22 15	:52 ショートカット	2 KB	
	🛃 イベント ビューアー		2013/08/22 15	:55 ショートカット	2 KB	
▷ 📬 ネットワーク	🛃 コンピューターの管理		2013/08/22 15	:54 ショートカット	2 KB	
	🔊 コンポーネント サービス		2013/08/22 15	:57 ショートカット	2 KB	
	💼 サービス		2013/08/22 15	:54 ショートカット	2 KB 🗸 🗸	
21 個の項目					800	

(左図)「追加」をクリックします。

(右図) ダウンロードした SQL Server 用 ODBC ドライバーを選択し、「完了」を押します。

ODBC データ ソース アドミニストレーター (64 ビット)	×	データ ソースの新規作成
ユーザー DSN システム DSN ファイル DSN ドライバー トレース 接続プール バージョン情報		
ユーザー データ ソース(U):		セットアップするデータ ソースのドライバーを選択してください(S)
名前 ブラットフォーム ドライバー	追加(D)	名前
Excel Files 64 Eyr Microsoft Excel Driver (*.xls, *.xlsr, MS Access Database 64 Eyr Microsoft Access Driver (*.mdb, *.accdb)	心 削除(R)	Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx) Microsoft Access Driver (*.mdb, *.accdb)
	編成(C)	Microsoft Access Text Driver (*.txt, *.csv) Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
		ODBC Driver 11 for SQL Server
		SQL Server SQL Server Native Client 11.0
د		< >>
ODBC ユーザー データ ソースには指定されたデータ プロパイダーへの接続方法に留する # - データ ソースは現在のユーザーにのみ認識され、このコンピューターでのみ使用できます。	朝鮮が格納されています。ユーザ	
OK キャンセル	適用(A) へルプ	< 戻る(8) 完了 スキャンセル

(左図)新規データソースの名前と説明にはユーザーが自由に入力できます。ここではわか りやすく、名前に「AdventureWorks」、説明には「SQL Server Sample Database」と入力 を行いました。「接続する SQL Server を選択してください」には、SQL Server のサーバー 名を入力します。「次へ(N)」をクリックします。

(右図) ログイン方法の設定を確認して「次へ(N)」をクリックします。



(左図) この画面で「既定のデータベースを以下に変更する」にチェックを入れて、アクセスしたいデータベース名(ここでは AdventureWorks2012)をプルダウンリストから選択し、「次へ(N)」をクリックします。

(右図) そのまま「完了」をクリックします。

	Microsoft SQL Server 用の DSN の設定		Microsoft SQL Server 用の DSN の設定
SOL Server		\$OL Server	Microsoft SQL Server 用の DSN の設定 ▲ SQL Server 09.253 メラセーラを以下の含要に定定する(C) (Cetual) ● 「コームたちりたねちを使用する(Y) (日本)」 ● 「スキーシを取りたねちを使用する(Y) (日本)」 ● 「スキーシーを取りたるを使用する(Y) (日本)」 ● 「スキーシーを取りたる(Y) (日本)」 ● 「スキーシーを取りたる(Y) (日本)」 ● 「スキーシーを取りたる(Y) (日本)」 ● 「日本)」 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● <td< th=""></td<>
	□マルチオナスットフェールオード(-F) < 英老(B) 液へ(N) () ネヤンセル ヘルブ		- 総合用品(1)(日時代) 1

(左図)「データソースのテスト」をクリックし接続テストを行います。 (右図)テスト結果を確認し、「**OK**」をクリックし、戻ります。



AdventureWorks が ODBC データソースに追加されたことを確認し、「OK」を押して設定終 了です。

1-ザー データ ソース	(U):			
有則 AdvanturaWorks	22/64 Earb	ODBC Driver 11 for SQL Server	追加(D)	
Excel Files MS Access Datab	64 ビット 64 ビット base 64 ビット	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsn Microsoft Access Driver (*.mdb, *.accdb)	削除(R)	
			構成(C)	
< ODBC : ーデータ	ユーザー データ ソース י ソースは現在のユー	> いには指定されたデータプロバイダーへの接続方法に関する情報 ゲーにのみ認識され、このコンビューターでのみ使用できます。	股が格納されています。ユーザ	

[SQL 言語を使用した DB 読み取り]

SQL 言語を用いて WPS から ODBC ドライバー経由で SQL Server の DB アクセスを行う 場合は、以下のように、エンジン名として ODBC を指定し、CONNECT 文のカッコ内に datasrc=ODBC データソース名などの接続オプションや dbmax_text=n (文字テキストの読 み取り最大長さの設定) などの libname オプションを指定します。

以下は、ODBC ドライバーを用いて、AdventureWorks2012 データベースの Person.Address テーブルの3つのカラム (AddressID, AddressLine1, City) を City の値が 'San Francisco' か 'Los Angeles' か 'Seattle' のいずれかに該当するレコードを抽出し、AddressID の小さ い方から並べた WPS データセット WORK.A を作成する例となっています。

```
proc sql;
  connect to ODBC(datasrc=AdventureWorks);
  create table A as
  select *
  from connection to ODBC
  (
    select AddressID, AddressLine1, City
    from Person.Address
    where City in ('San Francisco', 'Los Angeles', 'Seattle')
    order by AddressID
  );
  disconnect from ODBC;
quit;
```

(WPS ワークベンチの実行ログ)

95	proc sql;								
96	connect	to ODBC(da	ata	src=Advent	tureWo	orks	5);		
NOTE:	Successfully	connected	to	database	ODBC	as	alias	ODBC.	

```
97
           create table A as
98
           select *
99
           from connection to ODBC
100
           (
             select AddressID, AddressLine1, City
101
102
             from Person.Address
103
             where City in ('San Francisco', 'Los Angeles', 'Seattle')
104
             order by AddressID
105
           );
NOTE: Data set "WORK.A" has 285 observation(s) and 3 variable(s)
           disconnect from ODBC;
106
NOTE: Successfully disconnected from database ODBC.
107
         quit;
NOTE: Procedure sql step took :
     real time : 0.046
     cpu time : 0.015
```

※ ODBC ドライバーによる DB へのアクセス制限等の設定によっては、SQL プロシジャの CONNECT 文の()内のオプションに、その他の指定が必要となる場合があります。以下の ようにヘルプビューで指定可能なオプションが確認できます。

ヘルプビュー16の左上にある「内容」をクリックします。

ヘルプ全体の目次が表示されます。WPS Reference for Language Elements をクリックし

ます。



WPS レファレンスを展開し、WPS Engine for ODBC を選択します。

¹⁶ ヘルプビューが存在しない場合は、「ウィンドウ(W)」→「ビューの表示(V)」→ 「その他(O)」→「ヘルプ」フォル ダー → 「ヘルプ」からビューを出現させます。また、「ヘルプ(H)」→「ヘルプ目次(H)」から別ウィンドウで表示さ れるメニューでも同じです。

陷 内容 🎾 検索 📽 関連トピック 💷 ブックマーク 蘭 ·
スコープ: デフォルト
Workbench User Guide
WPS Reference for Language Elements
How to Read Syntax Diagrams
WPS Core
WPS Graphing
WPS Statistics
WPS Timeseries
WPS Communicate
WPS Engine for Actian Matrix
WPS Engine for DB Files
WPS Engine for DB2
WPS Engine for Greenplum
US WPS Engine for Informix
WPS Engine for Kognitio
BIS WPS Engine for MySQL
WPS Engine for NETEZZA
WPS Engine for ODPC
WPS Engine for OLC
WPS Engine for Oracle
WPS Engine for PostgreSQL
Mas who Engine for SAND
WPS Engine for SQL Server
WPS Engine for Sybase
Digit WPS Engine for Vertical
 Management Management Management
Degra Appendix Degra Noticos

その後、展開を何度か行うと接続オプション(connection option) が表示されます。

	16 🛊 🖹 🖣 🕖 🗢 ⇒ 🔻	
陷 内容 🎾 検索 📽 関連トビック 💷 ブックマーク 🛗 インデックス		
WPS Reference for Language Elements > WPS Engine for ODBC > Library Engines		~
ODBC		
<pre>LIBNAME name ODBC { connection-option } [{ libname-option }] ;</pre>		
▷ LIBNAME → name → ODBC → Connection-option → C	ibname-option	⋈
<pre>connection-option [user = user_name password = password datasrc = DSN datasrc = ["] conn connection_string ["] prompt = ["] connection_string ["] readbuff = n require </pre>	ection_string["] noprompt = ad = ["] connection_string["]]	["]
w user = user name		
password = password		
datasrc = DSN	1	
→ datasrc → = → → → ← connection string → → → →	1	
hoprompt he connection string		
<pre>>> prompt >> </pre> <pre>connection string >> </pre>	1	
readbuff = n		
← required + = + + + + + + + + + + + + + + + + +	J	~

これらのオプション (user= password= datasrc= noprompt= prompt= readbuff= required=) は ODBC ドライバー経由でのデータベースアクセスの際の CONNECT 文の DBMS オプシ

ョン指定としても、また、次に説明する libname ステートメントによるデータベースアクセ スの接続オプションとしても用いることができ、必要に応じて指定します。

[libname ステートメントによる DB 入力]

WPS のエディタから以下のプログラムを実行します。

libname person ODBC datasrc=AdventureWorks schema=Person; libname sales ODBC datasrc=AdventureWorks schema=Sales;

この指定は、WPS がデータソース AdventureWorks に ODBC ドライバー経由で接続し、 Person スキーマをライブラリ名 person で、Sales スキーマのテーブルをライブラリ名 sales で、それぞれ参照できるようにする指定です。(入出力両方可能) なお、datasrc=指定は DB 接続オプション、shema=指定は libname オプションです。 libname オプションには他に、dbmax_text=n (文字テキストの読み取り最大長さの設定) な どがあります。指定可能な libname オプションは SQLSERVR エンジンの libname オプシ ョンとほぼ同じですが、詳細は前出の ODBC エンジンのヘルプ画面から確認してください。 (Connection-option の下に libname-option が記載されています。)

実行すると、ログに person と sales 2 個の「ライブラリ名が割り当てられた」というメッ セージが表示されます。このメッセージは ODBC 経由で DB と接続できたことを表します。

128 libname person ODBC datasrc=AdventureWorks schema=Person;
NOTE: Library person assigned as follows:
Engine: ODBC
Physical Name: AdventureWorks
100 library cales ODDC deterns Adventuralization cales
129 IIDname Sales ODBC datasrc=Adventureworks Schema=Sales;
NOTE: Library sales assigned as follows:
NOTE: Library sales assigned as follows: Engine: ODBC

実行後、WPS サーバーエクスプローラービュー内の ローカルサーバー の下の Libraries を展開し、Person ライブラリと Sales ライブラリがあることを確認してください。これら の中に AdventureWorks2012 データベースの Person スキーマ、Sales スキーマ上のテーブ ル名およびビュー名がそれぞれ含まれていることがわかります。



※ 右矢印がついたアイテムはテーブル名(展開すると、テーブルに含まれる項目名と項目 タイプを表すアイコンが表示されます。)、付いていないのはビュー名です。

これらのテーブルやビューは、WPS データセットに対する操作と同じファイル操作が可能 です。例えば、以下のように、テーブル名を右クリックして出現するメニューの「プロパテ ィ」を選択すると、カラム名などを確認することができます。

```
🔺 🙆 ローカルサーバー
  Libraries
    a 📰 Person
      Address
      > 🔝 Addres 🔍 開く
      Busines
                   更新
                                           F5
      Busines
      Busines 
□ ⊐ピー
                                        Ctrl+C
      ▷ 📰 Contac 🗙 削除
                                          削除
      Countr
      ▶ III EmailA I データセットをインポート...
      ▷ III Passwc III データセットをエクスポート...
      Person
                  クリップボードに内容をコピー
      Person
      Phone プロパティ(R)
                                      Alt+Enter
                                  2
      StateProvince
        vAdditionalContactInfo
        StateProvinceCountryRegion
    Sales
    🛛 薪 Sashelp
    🖻 🚮 Sasuser
     💭 Work
  Filerefs
```

🔁 プロジェクトェクスプローラー 😤 WPS サーバーェクスプローラー 🛛 🛛 隆 ファイルェクス:

•		プロパティ: /	Addre	SS		_ 🗆	×
フィルタ入力	カラム					⇐ ◄ ⇔	• •
1 万 万 5 万 5 石 、	名前 ④ AddressID ④ AddressLine1 ⑤ AddressLine2 ⑥ City ④ ModifiedDate ⑥ PostalCode ⑥ rowguid ⑥ SpatialLocation ⑥ StateProvinceID	917 number character character number character character character number	長さ 8 60 60 30 8 15 36 1 8	フォーマット 11. \$60. \$30. DATETIME20. \$15. \$36. \$. 11.	インフォオーマット 11. \$60. \$30. DATETIME20. \$15. \$36. \$. 11.	ラベル AddressID AddressLine1 AddressLine2 City ModifiedDate PostalCode rowguid SpatialLocation StateProvinceID	
?	1					OK キャンセ	ŀ

※ <u>【重要な注意】: テーブル名をダブルクリックすると、データセットビューアが起動し、</u> <u>テーブルの中身を表示しようとします。この操作はテーブルのサイズが大きいときは WPS</u> セッションが動かなくなったり不意に終了してしまうなどの危険を伴いますので、できる <u>だけ避けてください。</u>

さて、ODBC アクセスエンジンを libname ステートメントで指定すると、以下のプログラ ムのように、DB テーブルやビューを WPS データセットに変換することなく、直接プロシ ジャ入力に用いることもできます。この例は、sales.salesperson テーブルを入力としてビ ジネスエリア別の売り上げ高の円グラフを描いています。

```
goptions cback=white;
title "Pie Chart of SalesYTD by SalesPersonID";
proc gchart data=sales.salesperson;
    pie businessentityid/sumvar=salesYTD;
run;
```



※ 注意: グラフィック表示は WPS3.1.1 では日本語の表示はサポートされていません。

また、以下のように、SAS 言語の DATA ステップでも DB テーブルを処理することができ るようになります。また、SQL プロシジャを使用して処理することも可能ですので、libname ステートメントを使って DB アクセスを行うときは、SAS 言語と SQL 言語いずれも使用可 能になるというメリットがあります。

下の例は、前の節で SQL プロシジャで作成した WORK.A と同じ内容のデータセット WORK.B を DATA ステップで作成しています。

```
data B;
  set person.address(keep=AddressID AddressLine1 City);
  where City in ('San Francisco', 'Los Angeles', 'Seattle');
run;
```

[WPS データセットの DB への出力方法]

以下のいずれかの方法でWPS データセットをDB のテーブルへ出力できます。

- ① DBLOAD プロシジャ
- ② SQL プロシジャ
- ③ DATA ステップ

①の方法では DB への接続方法を DBLOAD プロシジャで指定しますので、libname ステートメントで DB エンジンを指定する必要はありません。しかし、②と③の方法では、事前に
 DB への接続方法を libname ステートメントで指定しておく必要があります。

① DBLOAD プロシジャ

前節の最後のプログラムで Person.Address テーブルから 3 つの都市名のいずれかに該当す るレコードを選択した WPS データセット WORK.A を SQL Server の Person スキーマ上 の Address_Subset_A テーブルへ出力してみましょう。以下のようにプログラムを書いて 実行します。

```
proc dbload data=A dbms=ODBC;
    datasrc=AdventureWorks;
    table=Person.Address_Subset_A;
    load;
run;
```

※ 注意: DB にアクセスするための権限や ODBC エンジンの接続オプションや libname オ プションがさらに必要な場合は、load ステートメントの前に、オプション名=値;の形式で 指定を追加してください。

② SQL プロシジャによるテーブルへの読み書き

Libname ステートメントで ODBC エンジンを指定すると、DB テーブルを WPS データセットと同じように扱えるようになります。

以下の例は AdventureWorks2012 データベースの Sales.Customer テーブルを読んで、 TerritoryID=1 に該当するレコードのみ抽出した全カラムを含むテーブル Sales.Customer_Territory_1を作成しています。

```
libname sales ODBC datasrc=AdventureWorks schema=Sales;
proc sql;
  create table Sales.Customer_Territory_1 as
  select *
  from Sales.Customer
  where TerritoryID = 1;
quit;
```

※ この例では、Sales ライブラリは SQL Server DB の Sales スキーマを参照するよう最初 の libname ステートメントで定義しています。したがって、SQL プロシジャの中の Sales ライブラリは WPS データライブラリではなく、AdventureWorks2012 DB の Sales スキー マを表すものと解釈され、読み取りも書き出しも DB テーブルが対象になります。

③ DATA ステップ

以下の例は②の SQL プロシジャを使った処理と同じことを SAS 言語の DATA ステッププ ログラムで行っています。

```
libname sales ODBC datasrc=AdventureWorks schema=Sales;
/* 同じ名前のテーブルを一旦削除 */
proc datasets lib=sales;
  delete Customer_Territory_1;
run;quit;
/* DATAステップでDBテーブルを読み書き */
data Sales.Customer_Territory_1;
  set Sales.Customer;
  where TerritoryID = 1;
run;
```

以上