

システムの”見える化”支援 サンプル設計書

～ PL/SQL ～

DCR

株式会社第一コンピュータリソース

システムの“見える化”支援では、PL/SQLから以下を生成いたします。

<PL/SQL>		<VSSDドキュメント>
宣言	⇒	画面定義「処理タブ」
処理詳細	⇒	画面定義「処理タブ」
問合せ定義	⇒	画面定義「問合せタブ」、「処理タブ」
更新項目定義	⇒	画面定義「更新項目タブ」
カーソル	⇒	画面定義「問合せタブ」、「処理タブ」

宣言

PL/SQL

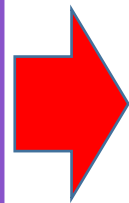
```
-- ユーザー関連処理パッケージ
CREATE OR REPLACE PACKAGE BODY PK_USER
IS
-- ユーザー登録
FUNCTION FN_REGIST_USER(
  P_USER_ID IN M_USER.USER_ID%TYPE,
  P_USER_NAME IN M_USER.USER_NAME%TYPE,
  P_GROUP_ID IN M_USER.GROUP_ID%TYPE,
  P_TEL IN M_USER.TEL%TYPE,
  P_FAX IN M_USER.FAX%TYPE,
  P_SAL IN M_USER.SAL%TYPE,
  P_CMT IN M_USER.CMT%TYPE
) RETURN NUMBER
IS
  WK_LOGIN_USER M_USER.USER_ID%TYPE;
  WK_RET NUMBER := -1;
  WK_CNT PLS_INTEGER;
  WK_GROUP_ID M_USER.GROUP_ID%TYPE;
BEGIN
  WK_LOGIN_USER := PK_COMMON.LOGIN_USER_ID;

  -- IDチェック
  SELECT COUNT(*)
  INTO WK_CNT
  FROM M_USER
  WHERE USER_ID = P_USER_ID;

  IF WK_CNT > 0 THEN
    DBMS_OUTPUT.PUT_LINE('指定したユーザーIDは既に存在します。');
    RETURN WK_RET;
  END IF;

  -- グループ情報取得
  BEGIN
    SELECT GROUP_ID
    INTO WK_GROUP_ID
    FROM M_GROUP
    WHERE GROUP_ID = P_GROUP_ID;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('該当するグループが見つかりません。');
      WK_RET := SQLCODE;
      RETURN WK_RET;
  END;

  -- 登録処理
  INSERT INTO M_USER (
    USER_ID,
    USER_NAME,
    GROUP_ID
```



VSSD

ファンクション

処理名	処理ID	コメント
1 FN_REGIST_USER	FN_REGIST_USER	
2 FN_UPDATE_USER	FN_UPDATE_USER	
3 FN_DELETE_USER	FN_DELETE_USER	

ファンクションパラメータ

項目名	項目ID	IN / OUT	データ型	クラス
1 P_USER_ID	P_USER_ID	IN	固定長文字	
2 P_USER_NAME	P_USER_NAME	IN	可変長文字	
3 P_GROUP_ID	P_GROUP_ID	IN	固定長文字	
4 P_TEL	P_TEL	IN	可変長文字	
5 P_FAX	P_FAX	IN	可変長文字	
6 P_SAL	P_SAL	IN	数値	
7 P_CMT	P_CMT	IN	可変長文字	
8 戻り値			数値	

ファンクション内の変数・定数

項目名	項目ID	種別	データ型	クラス	配列
1 WK_LOGIN_USER	WK_LOGIN_USER	変数	固定長文字		
2 WK_RET	WK_RET	変数	数値		
3 WK_CNT	WK_CNT	変数	数値		
4 WK_GROUP_ID	WK_GROUP_ID	変数	固定長文字		

処理詳細

PL/SQL

```
BEGIN
  WK_LOGIN_USER := PK_COMMON.LOGIN_USER_ID;

  -- IDチェック
  SELECT COUNT(*)
  INTO WK_CNT
  FROM M_USER
  WHERE USER_ID = P_USER_ID;

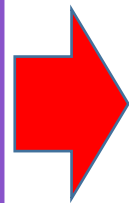
  IF WK_CNT > 0 THEN
    DBMS_OUTPUT.PUT_LINE('指定したユーザーIDは既に存在します。');
    RETURN WK_RET;
  END IF;

  -- グループ情報取得
  BEGIN
    SELECT GROUP_ID
    INTO WK_GROUP_ID
    FROM M_GROUP
    WHERE GROUP_ID = P_GROUP_ID;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('該当するグループが見つかりません。');
      WK_RET := SQLCODE;
      RETURN WK_RET;
  END;

  -- 登録処理
  INSERT INTO M_USER (
    USER_ID,
    USER_NAME,
    GROUP_ID,
    TEL,
    FAX,
    SAL,
    CMT,
    CREATE_USER_ID,
    CREATE_DATE,
    UPDATE_USER_ID,
    UPDATE_DATE
  ) VALUES (
    P_USER_ID,
    P_USER_NAME,
    P_GROUP_ID,
    P_TEL,
    P_FAX,
    P_SAL,
    P_CMT,
    WK_LOGIN_USER,
    SYSDATE,
    WK_LOGIN_USER,
    SYSDATE
  );

  COMMIT;

  WK_RET := SQLCODE;
  RETURN WK_RET;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('例外が発生しました。');
```



プログラムを可能な限り日本語に変換。
関連するオブジェクトにはリンクを張ることで
トレーサビリティが向上

VSSD

The screenshot shows the VSSD interface with the following components:

- Object Explorer:** Shows a tree view of the database objects, including tables (M_BASE, M_GROUP, M_USER, V_GROUP_USER), views (V_GROUP_USER), and a script (PK_USER: PK_USER).
- Table View:** A table with columns '処理名' (Process Name), '処理ID' (Process ID), and 'コメント' (Comment). It lists processes like FN_REGIST_USER, FN_UPDATE_USER, and FN_DELETE_USER.
- Process Detail View:** A detailed view of the 'PK_USER' process, showing a list of steps in Japanese, such as '1. <処理:変数>WK_LOGIN_USERに<PK_COMMON>LOGIN_USER_IDをセットする' and '2. select.FN_REGIST_USER.1を実行する'.

問合せは処理詳細に記述
するのではなく、問合せ定義
として外出してリンクを張る
ことで可読性が向上

問合せ定義

PL/SQL

```
BEGIN
  WK_LOGIN_USER := PK_COMMON.LOGIN_USER_ID;

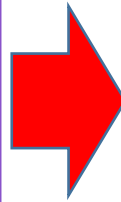
  -- IDチェック
  SELECT COUNT(*)
  INTO WK_CNT
  FROM M_USER
  WHERE USER_ID = P_USER_ID;

  IF WK_CNT > 0 THEN
    DBMS_OUTPUT.PUT_LINE('指定したユーザーIDは既に存在します。');
    RETURN WK_RET;
  END IF;

  -- グループ情報取得
  BEGIN
    SELECT GROUP_ID
    INTO WK_GROUP_ID
    FROM M_GROUP
    WHERE GROUP_ID = P_GROUP_ID;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('該当するグループが見つかりません。');
      WK_RET := SQLCODE;
      RETURN WK_RET;
  END;
```

```
EXECUTE IMMEDIATE 'TRUNCATE TABLE M_USER';
```



VSSD

問合せ名	問合せID	関連処理名	パラメータタイプ
select.FN_REGIST_USER.1	select.FN_REGIST_USER.1	FN_REGIST_USER	
select.FN_REGIST_USER.2	select.FN_REGIST_USER.2	FN_REGIST_USER	
select.FN_UPDATE_USER.1	select.FN_UPDATE_USER.1	FN_UPDATE_USER	
select.FN_DELETE_USER.1	select.FN_DELETE_USER.1	FN_DELETE_USER	

問合せ定義

詳細 項目移送 論理名表示 物理名表示

関連処理 FN_REGIST_USER パラメータタイプ 結果タイプ

取得項目

グループID

テーブル

グループマスタ

条件

グループID = 処理パラメータ.P_GROUP_ID

問合せは
問合せ定義として
外出しされる

解析ツールやVSSDが対応して
いない問い合わせは、
処理詳細の本文に取込まれる。

処理定義

処理名 PR_CREATE_TEST_USER

パラメータ / 戻り値

変数 / 定数

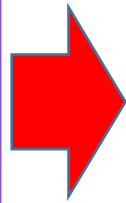
処理詳細

- 以下で設定されたSQLを実行する
TRUNCATE TABLE M_USER
- ループ変数WK_USER_CNTが1から処理変数MAX_USER_CNTまでの間、以下の処理を繰り返す
 - select.PR_CREATE_TEST_USER.1を実行する
 - select.PR_CREATE_TEST_USER.1の結果をセットする
処理:変数WK_GROUP_ID ← select.PR_CREATE_TEST_USER.1の結果.GROUP_ID
 - 例外【データ取得エラー】が発生した場合
 - RAISE_APPLICATION_ERRORを実行する

更新項目定義

PL/SQL

```
-- 登録処理
INSERT INTO M_USER (
  USER_ID
  , USER_NAME
  , GROUP_ID
  , TEL
  , FAX
  , SAL
  , CMT
  , CREATE_USER_ID
  , CREATE_DATE
  , UPDATE_USER_ID
  , UPDATE_DATE
) VALUES (
  P_USER_ID
  , P_USER_NAME
  , P_GROUP_ID
  , P_TEL
  , P_FAX
  , P_SAL
  , P_CMT
  , PK_COMMON.LOGIN_USER_ID
  , SYSDATE
  , PK_COMMON.LOGIN_USER_ID
  , SYSDATE
);
COMMIT;
```



VSSD

The screenshot shows the VSSD interface with the PK_USER table selected. The '更新項目定義' (Update Item Definition) window is open, displaying a table of update items and a detailed view of the 'ユーザーID' (User ID) item.

更新項目名	更新項目ID	区分	関連処理名	パラメータタイプ	コメント
insert.FN_REGIST_USER.1	insert.FN_REGIST_USER.1	追加	FN_REGIST_USER		
update.FN_UPDATE_USER.1	update.FN_UPDATE_USER.1	更新	FN_UPDATE_USER		
delete.FN_DELETE_USER.1	delete.FN_DELETE_USER.1	削除	FN_DELETE_USER		

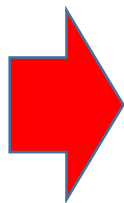
属性名	属性ID	更新内容	コメント
ユーザーID	USER_ID	処理パラメータ, P_USER_ID	
ユーザー名	USER_NAME	処理パラメータ, P_USER_NAME	
グループID	GROUP_ID	処理パラメータ, P_GROUP_ID	
電話番号	TEL	処理パラメータ, P_TEL	
FAX番号	FAX	処理パラメータ, P_FAX	
給料	SAL	処理パラメータ, P_SAL	
コメント	CMT	処理パラメータ, P_CMT	
作成者ID	CREATE_USER_ID	処理変数, WK_LOGIN_USER	
作成日	CREATE_DATE	SYSDATE	
更新者ID	UPDATE_USER_ID	処理変数, WK_LOGIN_USER	
更新日	UPDATE_DATE	SYSDATE	

追加・更新・削除は更新項目定義として外出される

カーソル

PL/SQL

```
CREATE OR REPLACE PROCEDURE PR_SHOW_USER_SAL
IS
^
WK_SAL^ M_USER.SAL%TYPE := 0;
-- グループ取得カーソル
CURSOR CUR_GROUP
IS
^
SELECT *
FROM M_GROUP;
^
REC_GROUP^ M_GROUP%ROWTYPE;
-- 指定グループのユーザー取得カーソル
CURSOR CUR_USER(C_GROUP_ID IN M_USER.GROUP_ID%TYPE)
IS
^
SELECT *
FROM M_USER
WHERE GROUP_ID = C_GROUP_ID;
^
REC_USER^ M_USER%ROWTYPE;
BEGIN
-- グループループ (OPEN+ループの場合)
OPEN CUR_GROUP;
LOOP
^
FETCH CUR_GROUP INTO REC_GROUP;
EXIT WHEN CUR_GROUP%NOTFOUND;
^
-- ユーザーループ (FORループの場合)
FOR REC_USER IN CUR_USER(REC_GROUP.GROUP_ID) LOOP
^
WK_SAL := WK_SAL + REC_USER.SAL;
END LOOP;
^
-- 出力
DBMS_OUTPUT.PUT_LINE(REC_GROUP.GROUP_ID || ' = ' || WK_SAL);
WK_SAL := 0;
END LOOP;
CLOSE CUR_GROUP;
EXCEPTION
^
WHEN OTHERS THEN
^
DBMS_OUTPUT.PUT_LINE('例外が発生しました。' || SQLERRM);
END PR_SHOW_USER_SAL;
/
```

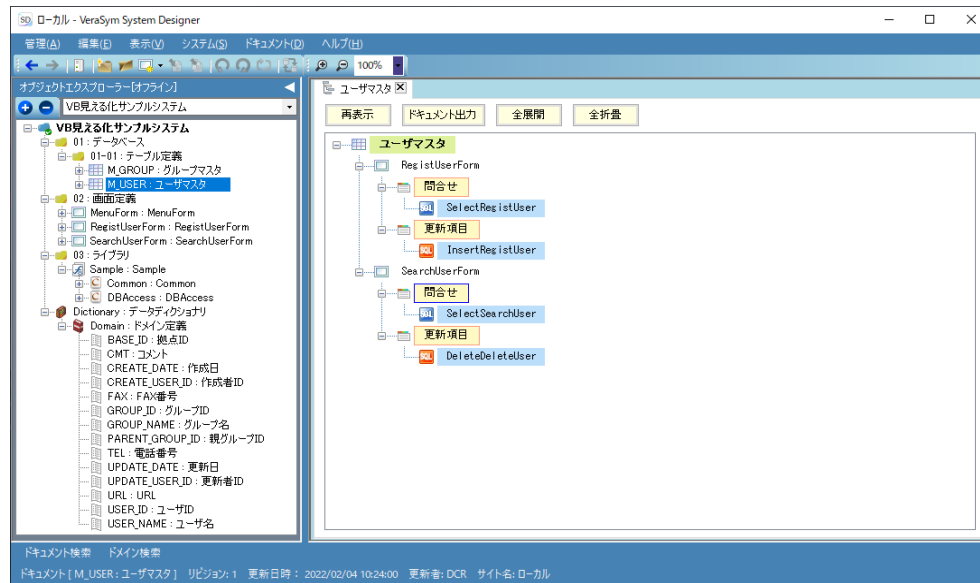


VSSD

カーソル宣言は
問合せ定義として
外出される

影響調査

オブジェクト関連図



テーブルや画面などのオブジェクトが他のオブジェクトから参照されているのか、また他のオブジェクトを参照しているのかをツリー構成で表現します。
仕様変更時の影響範囲確認の精度向上、工数削減が実現します。

CRUD表

ドキュメントID	ドキュメント名	区分	テーブルID/ビューID	テーブル名/ビュー名	作成 (C)	参照 (R)	更新 (U)	削除 (D)
7	SearchUserForm	SearchUserForm	M_USER	ユーザーマスタ	○	○	○	○
8	SearchUserForm	SearchUserForm	M_GROUP	グループマスタ	○	○	○	○
9	RegistUserForm	RegistUserForm	M_GROUP	グループマスタ	○	○	○	○
10	RegistUserForm	RegistUserForm	M_USER	ユーザーマスタ	○	○	○	○

画面や帳票、クラス定義の問合せ定義・更新項目定義で使用しているテーブル・ビュー定義の情報を基に自動生成可能です。

製品紹介、デモ、サンプル設計書ご希望はこちらまで



052-204-1377



vssd-info@dcr.co.jp



<https://vssd.dcr.co.jp/>

「コウシタイ」という心の想いを
“システムという形にする”



DCR

