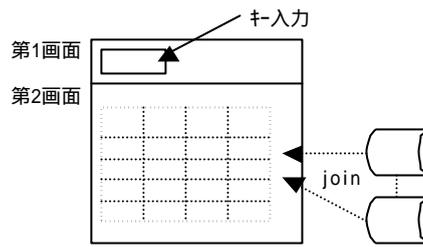


1. 画面パターンの種類

(1) 検索系

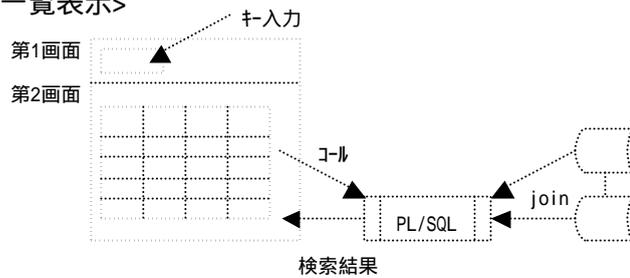
パターン1 <標準一覧表示>

- SQL文一つによる検索一覧表示。
- fetch等、ロジックは組めない。
- SQL文一つのための検索(join等可能)。



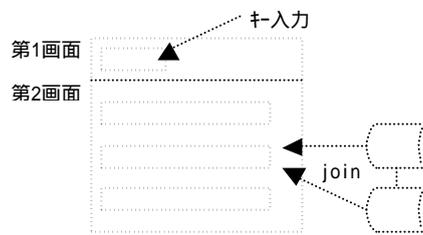
パターン1a <PL/SQLによる標準一覧表示>

- JSPよりPL/SQLエンジンコールし、検索結果を画面表示する。
- 画面表示はパターン1と同じ



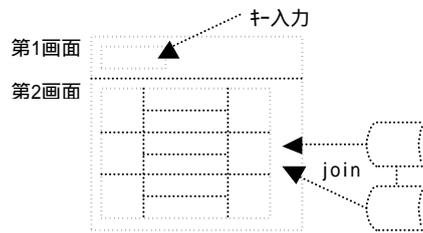
パターン2 <標準明細表示>

- 検索結果を1レコード毎に表示する。



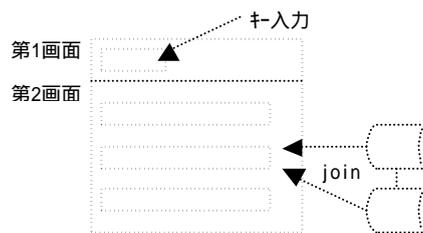
パターン3 <特殊一覧表示>

- 検索結果を人手で段組表示する。
- パターン1の特殊系



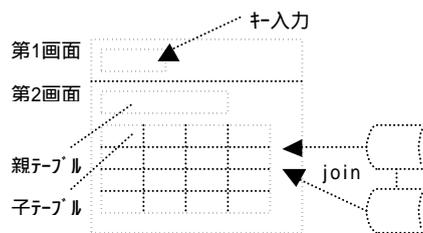
パターン4 <特殊明細表示>

- 表示を人手で指示する。
- パターン2の特殊系



パターン5 <明細一覧検索>

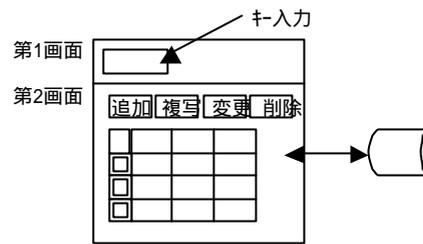
- 親テーブルはUserInfoTag使用。
- 子テーブルはQueryTag使用。
- パターン1の特殊系



(2) 更新系

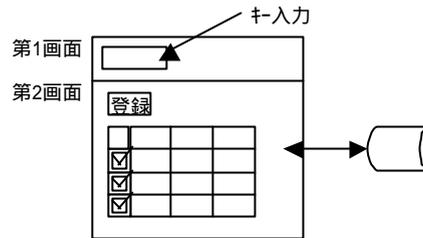
パターン 6 <標準一覧登録>

- ・ 検索パターン 1 の更新系
- ・ 更新するTableを検索し、レコードを指示し、追加、複写、変更、および削除する。



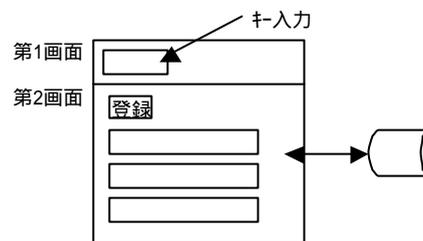
パターン 6 a <標準一覧登録>

- ・ パターン 6 の特殊系
- ・ 更新するTableを検索し、レコードを指示し、変更のみを行う。



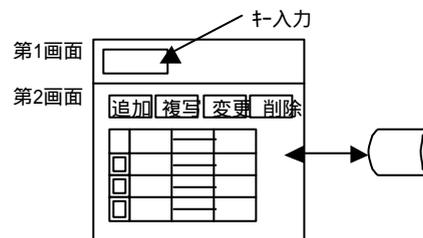
パターン 7 <標準明細登録>

- ・ 検索パターン 2 の更新系
- ・ 更新するTableを検索し、レコードのみを変更する。



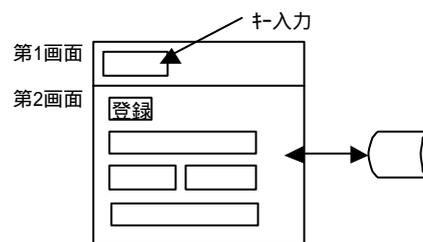
パターン 8 <特殊一覧登録>

- ・ 検索パターン 3 の更新系
- ・ 更新するTableを検索し、レコードを指示し、追加、複写、変更、および削除する。
- ・ 検索結果表示のとき段組表示する。



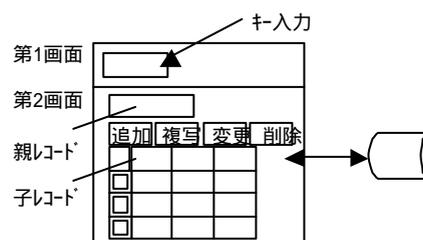
パターン 9 <特殊明細登録>

- ・ 検索パターン 4 の更新系
- ・ 更新するTableを検索し、レコードのみを変更する。
- ・ 検索結果の表示を人手で変える。



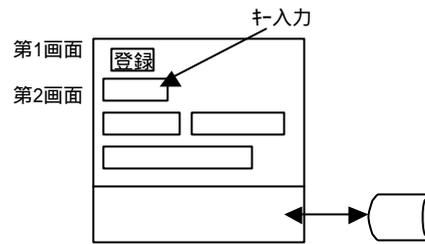
パターン 10 <明細一覧登録>

- ・ 検索パターン 5 の更新系
- ・ 更新するTableを検索し、子のレコードを指示することで、追加、複写、変更、および削除する。



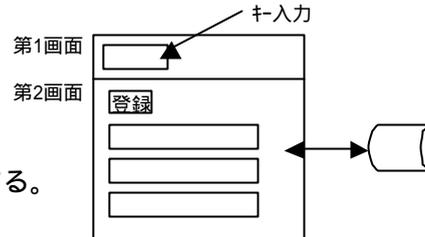
パターン11 <エントリー登録>

- 更新するTableを検索せずに、レコードを追加する。



パターン12 <一覧 明細登録>

- 更新するTableを検索し、子のレコードを指示することで、追加、複写、変更、および削除する。
- 検索画面は、パターン1 or 3で、更新画面は、パターン7 or 9に遷移する。



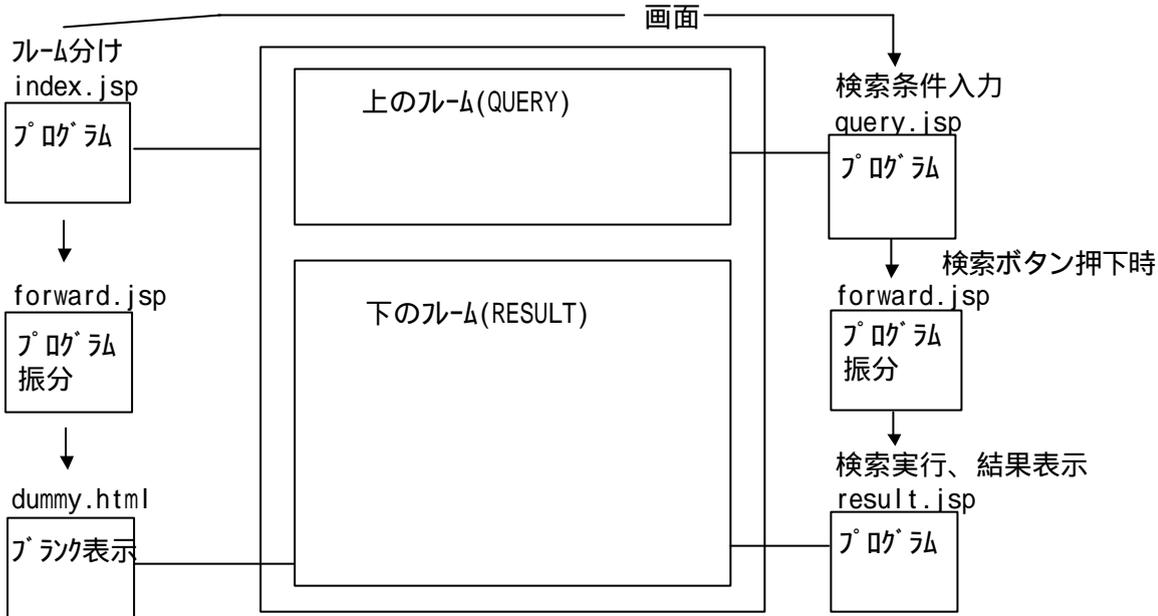
2. 各パターンプログラムの機能、及びフレームの関係

(1) 検索系(全パターン共通)

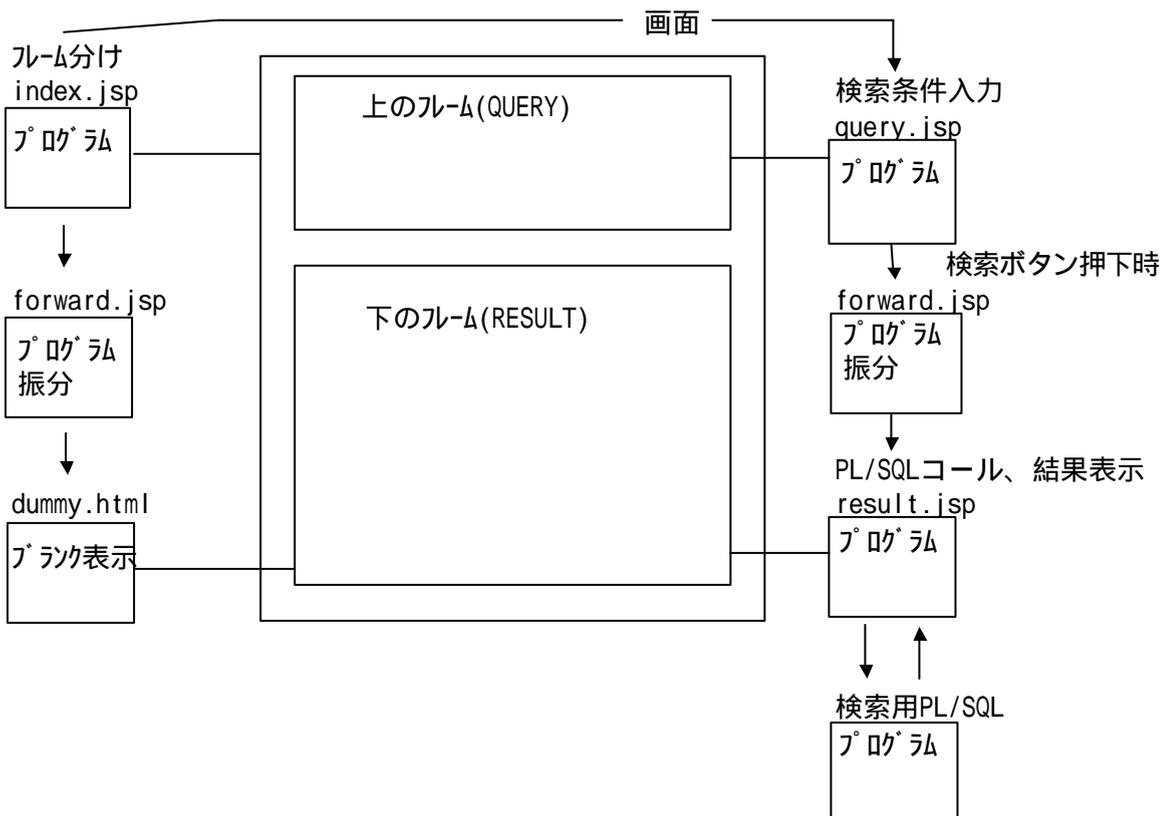
検索画面は、パターンに関係なく画面の遷移は共通で、検索にPL/SQLを使うか使わないかの違いがある。

1つの検索文で対応しきれない(WORKテーブルを使用して検索する)場合、検索用PL/SQLが必要。

a. PL/SQLなし

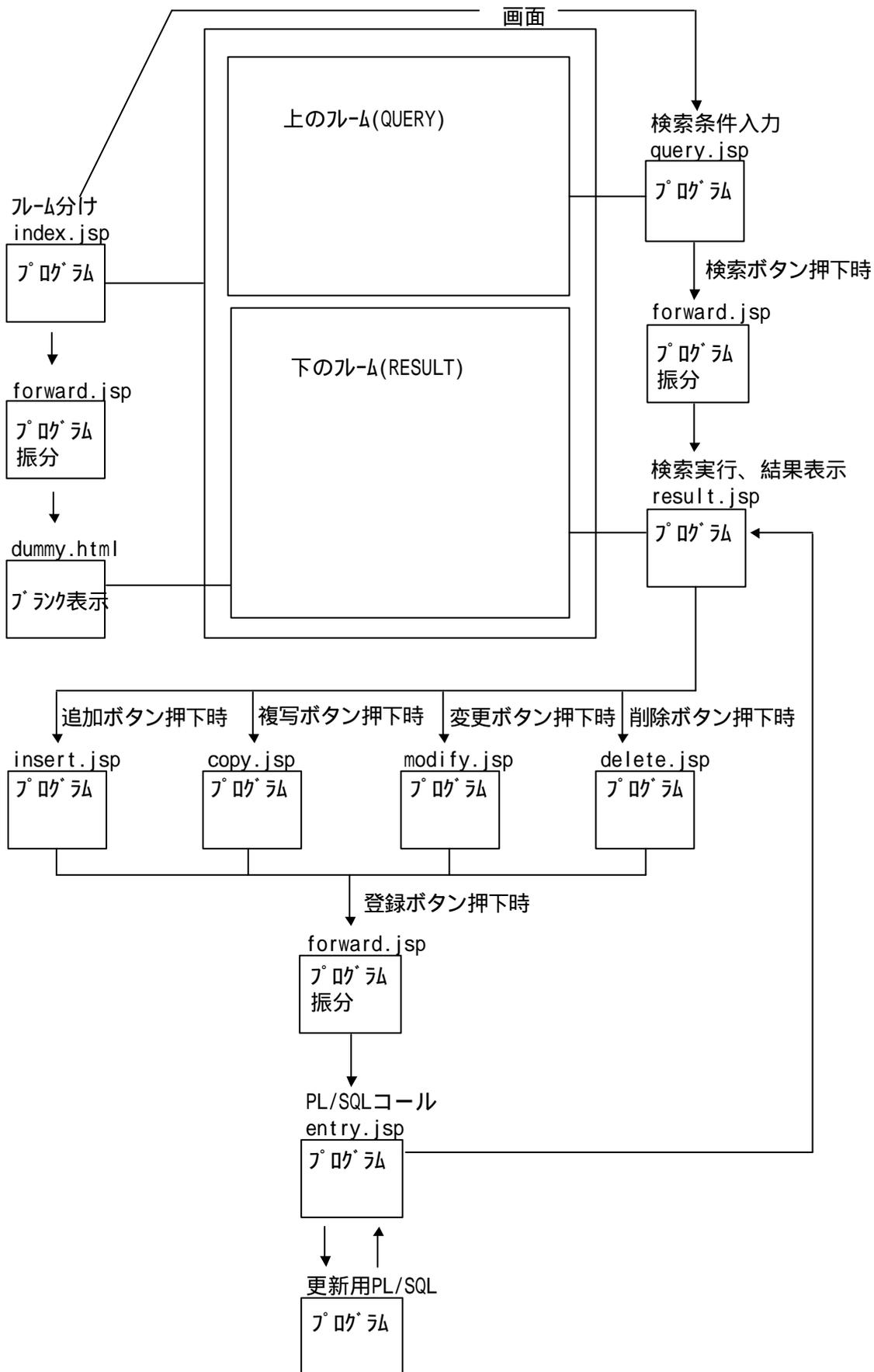


b. PL/SQLあり

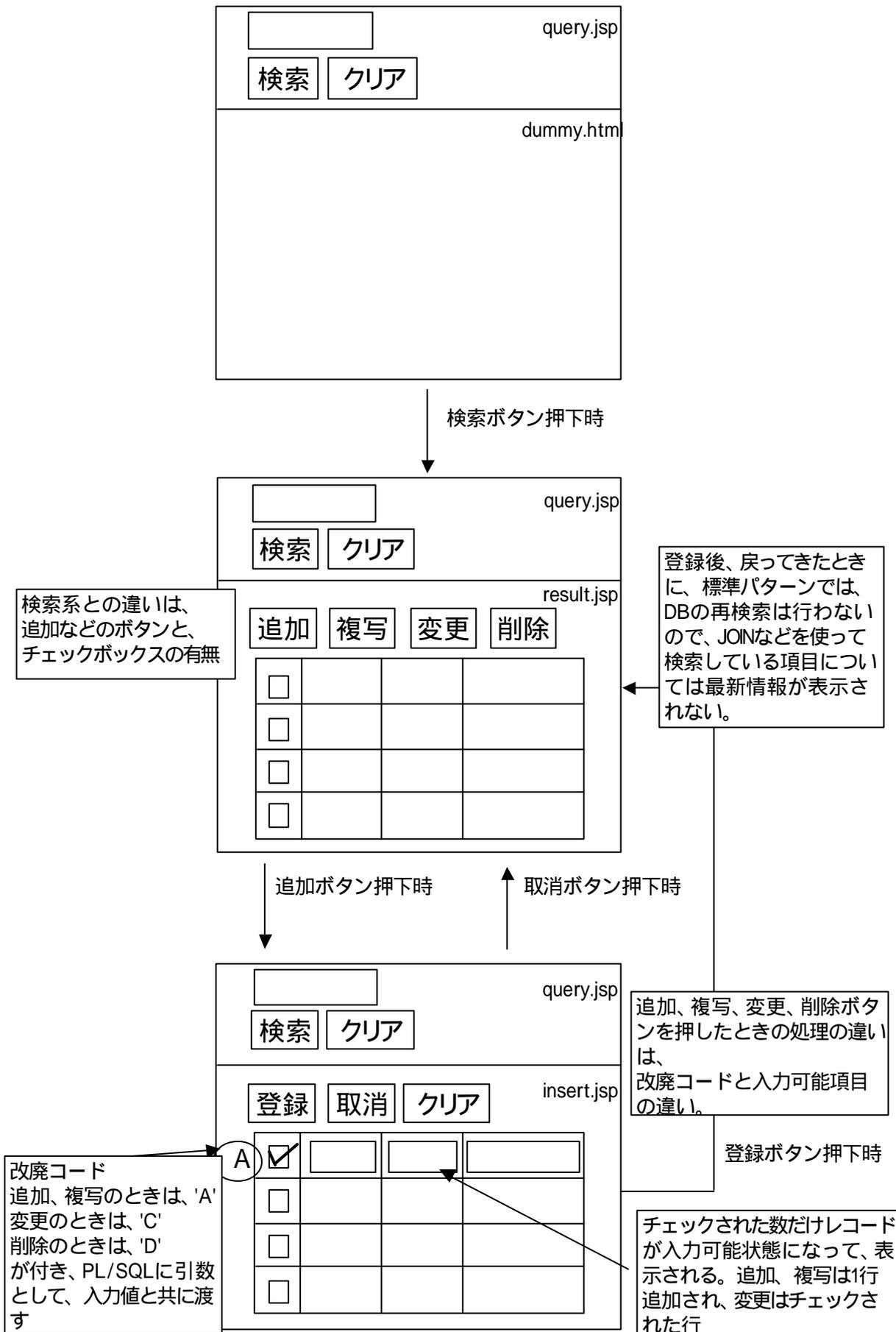


(2) 更新系(パターン6、8、10)

最も標準的な更新パターンで、検索結果のレコードをチェックして追加、複写などのボタンを押して登録する場合

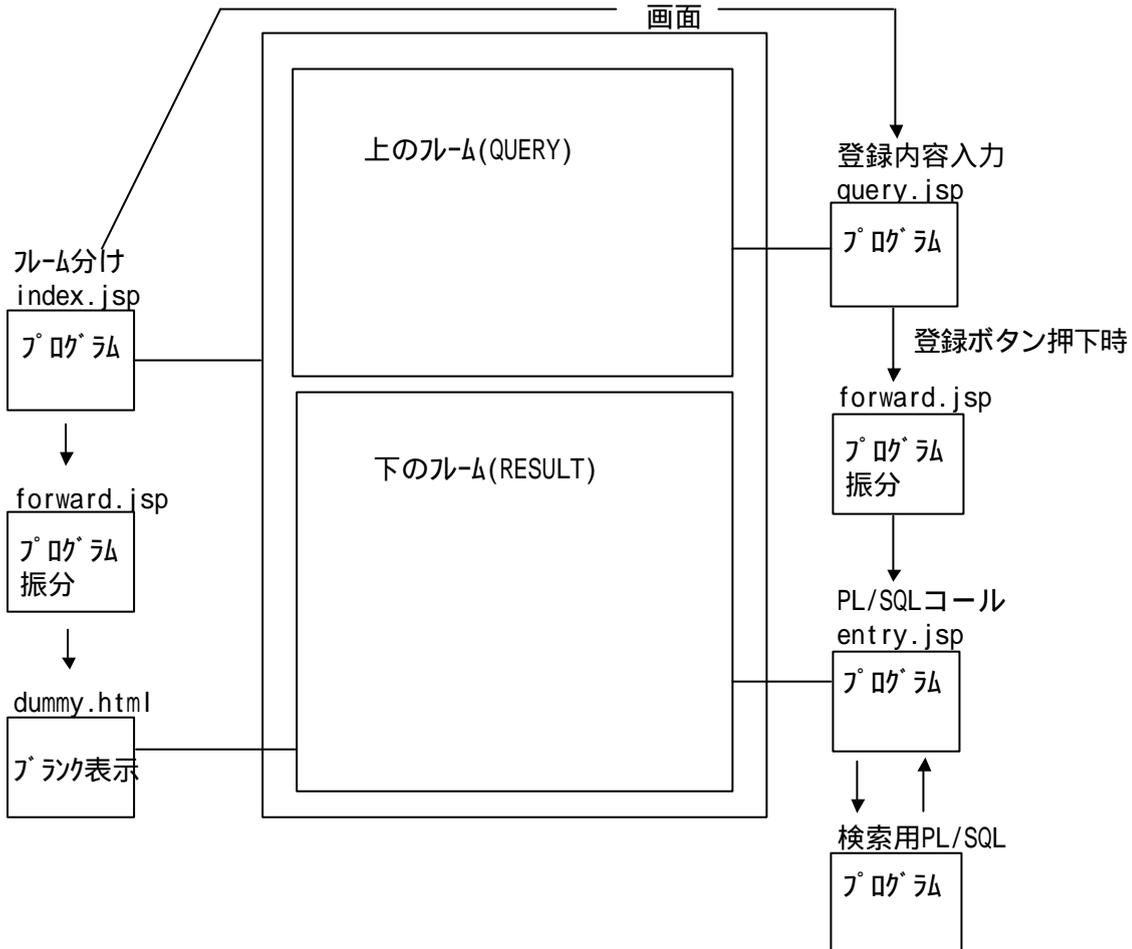


更新系画面(パターン 6、8)の画面遷移



(4) エントリー登録系

エントリー登録用の画面で、基本的にはPL/SQLを使った検索系画面と同じ。
PL/SQLの中でINSERT文を使ってDBに書き込む。検索系のPL/SQLを使ってるため
引数でカーソルを返す必要がある。



3 .JSPエンジンによる入力項目のチェック基準

(1) 基本事項

- 基本的に入力項目チェックはJSPエンジンにて行う。その場合、エラーメッセージは、エンジン側で用意されている為、新たに定義する必要はない。
- JSPエンジンでチェックできない項目については、PL/SQLにて行う。その場合、エラーメッセージを新たに定義する必要がある。

(2) 画面 (JSP)で行えるチェック内容

NULLチェック (仕様書に記入要、エラーメッセージは不要)

ColumnCheckTag に指定することで、個別の項目に対してのみ可能
但し、複数の項目に対して少なくとも1つ入力必須といったチェックは不可

リソースファイルで定義している項目 (仕様書のチェック基準書不要)

リソースファイルに指定されている桁数、DBType に基づくチェックが行える

DBType	意味	チェック内容
K	全角	桁数、全角文字チェック
R	小数 (XXXX.YY の場合は、62 と表す)	桁数、半角数字チェック
S9	半角数字 ('0' ~ '9','-')	桁数、('0' ~ '9','-')チェック
X	半角文字 (半角カナ含まず)	桁数、半角文字 (半角カナ含まず)
XH	半角文字 (半角カナ含む)	桁数、半角文字 (半角カナ含む)
X9	数字型文字列 ('0' ~ '9','-')	桁数、('0' ~ '9','-')チェック
XK	半角/全角混在	桁数
XL	半角小文字 (半角カナ含まず)	桁数、大文字は小文字に変換
XU	半角大文字 (半角カナ含まず)	桁数、小文字は大文字に変換
YM01	YYYYMM01 または、00000000, 99999999	8桁の数字、YYYYMM01に変換
TM31	YYYYMM31 または、00000000, 99999999	8桁の数字、YYYYMM31に変換
YMD	DBType が YYYYMMDD	8桁の数字
YMDH	YYYYMMDD24MISS	14桁の数字

(3) PL/SQLで行うチェック内容

- S9型のカラムに対するマイナス不可のチェック。
- YMD型のカラムに対する不正日付、過去日付のチェック。
- データベースのマスタチェック。
(エンジンにそれらの機能を追加することは可能)

4 .JSPからPL/SQLのCALLについて

(1) PL/SQLの種類

・JSP側からCALLするPL/SQLは、2種類 (検索・エントリー登録用、更新用) がある。

検索、エントリー登録用

・1度のSELECT文で検索できない場合やワークテーブルを使って検索する場合。

・検索を行わないで、直接登録する場合 (エントリー登録)。

更新用

・DBを一度検索し、その後、追加、変更や削除を行う場合。

(2) 引数

・引数にユーザー定義の型を使っているため、共通で使う型については、Oracle のユーザー毎に定義しておく必要がある。(標準のスクリプトを用意)

・PL/SQLからJSPに返すエラーメッセージのオブジェクト型配列にエラーコードをセットするためのスタアドプロシジャ (SP) をコンパイルする必要がある。(標準のPGを用意)

検索、エントリー登録用

引数	変数名	型	CALL元(JSP)	CALL先(PL/SQL)
結果	P_KEKKA	NUMBER型	-	0:正常 1:警告 2:異常
エラーメッセージ	P_ERRMSG	オブジェクト型配列	-	エラーメッセージ用
検索結果	P_RC1	CUST CURSOR	-	検索結果のカーソル
入力値	P_ARGS	VARCHAR2型配列	画面からの入力値 (1)	-

(1) 入力値は(カラム名はなし)指定した順番に配列で渡す。その為に、配列タイプ(VARRAY)を宣言する必要がある。(この配列タイプは、すべての検索用PL/SQL共通で使用する)

(注) 標準で用意しているスクリプトのVARCHAR2型配列は100バイト* 30個となっている。不足する場合はスクリプトの変更が必要となる。

また、エラーメッセージのオブジェクト型配列は、100個となっている。

更新用

引数	変数名	型	CALL元(JSP)	CALL先(PL/SQL)
結果	P_KEKKA	NUMBER型	-	0:正常 1:警告 2:異常
エラーメッセージ	P_ERRMSG	オブジェクト型配列	-	エラーメッセージ用
カラム	P_NAMES	VARCHAR2型	カラム名文字列 (カンマ区切り)	-
登録条件	P_SYSARGS	オブジェクト型配列	登録条件配列 (2) (A:追加 C:変更 D:削除)	-
入力値	P_XXXARGS	オブジェクト型配列	登録データ配列 (3)	-

(2) 対象レコードに対して改廃コード(A, C, D)も引数で渡すため、PL/SQL内でその改廃コードを使って処理を振り分ける。

(3) あらかじめ引数の型をオブジェクト型配列として宣言する。

(注) 標準で用意しているスクリプトのオブジェクト型配列は100個となっている。不足する場合はスクリプトの変更が必要となる。

(3) インターフェース

検索用、エントリー登録用

InputTagなどのValueを文字列の配列タイプで受け取り、検索結果をカーソルで返す。

```
CREATE OR REPLACE PACKAGE AAA IS      /* パッケージ名 */
TYPE CUST_CURSOR IS REF CURSOR ;
PROCEDURE BBB (                       /* プロシージャ名 */
  KEKKA OUT NUMBER,                   /* 結果0 :正常1 :警告2 :異常*/
  ERRMSG OUT ERR_MSG_ARRAY,          /* エラーメッセージ配列*/
  RC1 OUT CUST_CURSOR,               /* 結果はカーソルタイプで返す。*/
  P_ARGS IN ARG_ARRAY );             /* 引数は、文字列の配列タイプで受け取る。*/
END;
/
CREATE OR REPLACE PACKAGE BODY AAA IS
PROCEDURE BBB (
  KEKKA OUT NUMBER ,
  ERRMSG OUT ERR_MSG_ARRAY,
  RC1 OUT CUST_CURSOR,
  P_ARGS IN ARG_VARRAY,
) AS

  P_PN VARCHAR2(11) := P_ARGS(1);     /* JSPで指定した順序で配列に入っている */
  P_CDK VARCHAR2(2) := P_ARGS(2);

BEGIN

  OPEN RC1 FOR
  SELECT * FROM DB01

END BBB;
END AAA;
/
```

更新用

ア．登録データの配列の宣言

```
CREATE OR REPLACE TYPE ZZZARG AS OBJECT
(
  SSS VARCHAR2(20),
  TTT VARCHAR2(200),
  OOO VARCHAR2(10) );
/
CREATE OR REPLACE TYPE ZZZARG_ARRAY AS VARRAY(100) OF ZZZARG;
/
```

イ．アのオブジェクト型配列を受け取り、結果とエラーメッセージ配列を返す。

```
CREATE OR REPLACE PACKAGE XXX_PKG AS
PROCEDURE YYY(
  P_KEKKA OUT NUMBER,                /* 結果0 :正常1 :警告2 :異常*/
  P_ERRMSG OUT ERR_MSG_ARRAY,        /* エラーメッセージ配列*/
  P_NAMES IN VARCHAR2,              /* カラム名チェック用文字列*/
  P_SYSARGS IN SYSARG_ARRAY,         /* 登録条件配列*/
  P_ZZZARG IN ZZZARG_ARRAY );        /* 登録データ配列*/
```

```

END;
/
CREATE OR REPLACE PACKAGE BODY XXX_PKG AS
CREATE OR REPLACE PROCEDURE YYY(
  P_KEKKA OUT NUMBER ,
  P_ERRMSG OUT ERR_MSG_ARRAY,
  P_NAMES IN VARCHAR2,
  P_SYSARGS IN SYSARGARRAY,
  P_ZZZARGS IN ZZZARG_ARRAY,
) IS

END YYY;
END XXX;
/

```

(4) エラーメッセージ

JSP側にエラーを返すにはオブジェクト型配列にSPを使用してセットする。

```

HYBS.SET_ERRMSG(
  P_ERRMSG, /* エラーメッセージセット用SP */
  P_SYSARGS(I).NO, /* エラーメッセージ配列 */
  2, /* 行番号 */
  'ERR0001', /* エラー結果(0:正常 1:警告 2:異常) */
  P_ZZZARGS(I).SSS, /* エラーコード */
  ... /* エラーメッセージ1 */
); /* エラーメッセージ2 */

```

エラーメッセージは、エラーコードの定義によって1から5まで返すことが可能。