

プロジェクト名:〇〇会社製品開発システム

機能名: 部署情報

企業名: B-FORME

所属: B-FORME第1Java開発部隊

責任者:

Copyright 2021 B-FORME inc.

プロジェクト名	B-FORME ○○工場開発システム	機能名	部署情報	作成日	2023年5月29日
				更新日	2023年5月29日

No	年月日	内容	更新者
1	2023年5月29日	新規作成	梶浦 宰
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

プロジェクト名	B-FORME ○○工場開発システム	機能名	部署情報	作成日	
				更新日	

No	クラス名（論理名）	クラス名（物理名）	内容
1	部署情報リポジトリ	Bfmk07Repository	データソースを隔離し、保守や拡張性を高めてデータベースを扱う

No	メソッド名（論理名）	メソッド名（物理名）	内容
1	部署リスト処理	pageSelect	SQLから部署リストを取得する
2	ページ情報処理	pageDetail	ページに関する情報を取得する
3	ユーザー情報処理	autConf	操作権限の確認に必要な情報を取得する
4	登録処理	entry_exe	登録処理を行う
5	重複確認	entryCheck	重複するデータがあるか判断する
6	登録時確認	entry_subCheck	削除済の部署IDに重複があるか確認する
7	更新処理	entry_sub	更新処理を行う
8	削除確認	deleteCheck	対象部署があるか確認する
9	削除処理	delete_exe	削除処理を行う
10	社員所属確認	employeesCheck	社員人数を確認する
11	社員削除処理	deleteEmployees	対象部署の職員を削除する

備考

[illegible]

詳細設計_仕様

プロジェクト名	B-FORME ○○工場開発システム	機能名	部署情報	作成日	1899年12月30日
				更新日	1899年12月30日

引数（論理名）		メソッド名（論理名）	引数（物理名）	メソッド名（物理名）
IN	指定行数	部署リスト処理	IN	Integer
OUT	表示リスト		OUT	List<Bfmk07B elongingEntity >

〔1〕 SQL文

A) SQL文の設定を行う

```

SELECT
  b.affiliate_id,
  b.affiliate_name,
  b.affiliate_name_r,
  COUNT(u.affiliate_id) AS employees,
  b.apply_strt_date,
  b.apply_fin_date
FROM
  belonging b
LEFT OUTER JOIN user_info u
ON b.affiliate_id = u.affiliate_id
WHERE b.del_flg = 0
GROUP BY
  b.affiliate_id,
  b.affiliate_name,
  b.affiliate_name_r,
  b.apply_strt_date,
  b.apply_fin_date
ORDER BY employees desc,b.apply_strt_date desc
LIMIT 8 OFFSET ?
;

```

B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
LIMIT 8 OFFSET ?	INパラメータ：開始行数

〔2〕 出力

A) 設定したSQL文とINパラメータ：指定行数 から部署リストとして List<Bfmk07BelongingEntity>に設定する。

エンティティ 変数(論理名)	エンティティ変数 (物理名)	設定するキー項目
部署ID	affiliate_id	b.affiliate_id
部署名	affiliate_name	b.affiliate_name
部署名略称	affiliate_name_r	b.affiliate_name_r
所属人数	employees	employees
適用開始日	apply_strt_date	b.apply_strt_date
適用終了日	apply_fin_date	b.apply_fin_date

詳細設計_仕様

B) 設定したList<Bfmk07BelongEntity>を戻り値に設定する。

[3] 処理を完了する。

引数 (論理名)		メソッド名 (論理名)	引数 (物理名)		メソッド名 (物理名)
IN	--	ページ情報処理	IN	--	pageDetail
OUT	ページ情報[]		OUT	int[]	

[1] SQL文

A) SQL文の設定を行う

```
SELECT
COUNT(del_flg) AS sum,
COUNT(del_flg) DIV 8 AS pageMax,
COUNT(del_flg) MOD 8 AS sam
FROM
belonging b
WHERE
del_flg = 0
;
```

[2] 出力

A) 設定したSQL文からページ情報を戻り値であるページ情報[]に設定する。

受け取る変数 (論理名)	受け取る変数 (物理名)	設定するキー項目
ページ情報[]	PageDate[0]	sum
ページ情報[]	PageDate[1]	pageMax
ページ情報[]	PageDate[2]	sam

[3] 処理を完了する。

引数 (論理名)		メソッド名 (論理名)	引数 (物理名)		メソッド名 (物理名)
IN	部署情報DTO	ユーザー情報処理	IN	Bfmk07Dto	autConf
OUT	ユーザー情報[]		OUT	String[]	

[1] SQL文

A) SQL文の設定を行う

```
SELECT
b.affiliate_id AS id,
u.auth_div AS division
FROM
belonging b
LEFT OUTER JOIN user_info u
ON b.affiliate_id = u.affiliate_id
WHERE u.del_flg = 0
AND u.user_id = ?
LIMIT 1
```

詳細設計_仕様

;

- B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
u.user_id = ?	INパラメータ: 部署情報DTO: ユーザーID

[2] 出力

- A) 設定したSQL文からページ情報を出力する。

受け取る変数 (論理名)	受け取る変数 (物理名)	設定するキー項目
ユーザー情報[]	auth[0]	id
ユーザー情報[]	auth[1]	division

[3] 処理を完了する。

引数 (論理名)		メソッド名 (論理名)	引数 (物理名)		メソッド名 (物理名)
IN	入力フォーム	登録処理	IN	Bfmk07EntryForm	entry_exe
IN	部署情報DTO		IN	Bfmk07Dto	
OUT	エラー判定		OUT	boolean	

[1] SQL文

- A) SQL文の設定を行う

```
INSERT
  INTO belonging (
    fac_cd ,
    affilicate_id ,
    affilicate_name ,
    affilicate_name_r ,
    apply_strt_date ,
    apply_fin_date ,
    del_flg ,
    create_div ,
    create_date ,
    create_id ,
    update_date ,
    update_id ,
    note )
  VALUES ( ?, ?, ?, ?, ?, ?, '0', 'C', ?, ?, ?, ?, null )
;
```

- B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
fac_cd = ?	INパラメータ: 部署情報DTO: 工場CD
affilicate_id = ?	INパラメータ: 入力フォーム: 部署ID
affilicate_name = ?	INパラメータ: 入力フォーム: 部署名
affilicate_name_r = ?	INパラメータ: 入力フォーム: 部署名略称

詳細設計_仕様

apply_strt_date = ?	INパラメータ：入力フォーム：適用開始日
apply_fin_date = ?	INパラメータ：入力フォーム：適用終了日
create_date = ?	現在時刻
create_id = ?	INパラメータ：部署情報DTO：ユーザーID
update_date = ?	現在時刻
update_id = ?	INパラメータ：部署情報DTO：ユーザーID

< 条件：1 > エラーが発生した場合

- ・ 戻り値にfalseを設定する。

エラーが発生しない場合

- ・ 戻り値にtrueを設定する。

[2] 処理を完了する。

引数（論理名）	メソッド名（論理名）	引数（物理名）	メソッド名（物理名）
IN	部署ID	IN	String
OUT	重複判定	OUT	boolean

entryCheck

[1] SQL文

A) SQL文の設定を行う

```
SELECT
  affilicate_id
FROM
  belonging
WHERE del_flg = 0
AND affilicate_id = ?
LIMIT 1
;
```

B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
affilicate_id = ?	INパラメータ：部署ID

< 条件：1 > 要素を取得した場合

- ・ 戻り値にtrueを設定する。

< 条件：2 > 要素を取得しなかった場合

- ・ 戻り値にfalseを設定する。

[2] 処理を完了する。

引数（論理名）	メソッド名（論理名）	引数（物理名）	メソッド名（物理名）
IN	部署ID	IN	String
OUT	重複IDの判定	OUT	boolean

entry_subCheck

[1] SQL文

A) SQL文の設定を行う

詳細設計_仕様

```
SELECT
  affiliate_id
FROM
  belonging
WHERE del_flg = 1
AND affiliate_id = ?
LIMIT 1
;
```

B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
affiliate_id = ?	INパラメータ: 部署ID

< 条件: 1 > 要素を取得した場合

- ・ 戻り値にtrueを設定する。

< 条件: 2 > 要素を取得しなかった場合

- ・ 戻り値にfalseを設定する。

[2] 処理を完了する。

引数 (論理名)		メソッド名 (論理名)	引数 (物理名)		メソッド名 (物理名)
IN	入力フォーム	更新処理	IN	Bfmk07EntryForm	entry_sub
IN	部署情報DTO		IN	Bfmk07Dto	
OUT	エラー判定		OUT	boolean	

[1] SQL文

A) SQL文の設定を行う

```
UPDATE
  belonging
SET
  fac_cd=?,
  affiliate_name = ?,
  affiliate_name_r = ?,
  apply_strt_date = ?,
  apply_fin_date = ?,
  del_flg = 0,
  create_div = 'C',
  create_date = ?,
  create_id = ?,
  update_date = ?,
  update_id = ?,
  note = null
WHERE affiliate_id = ?
;
```

B) 指定情報を設定してSQLの処理をする

詳細設計_仕様

バインド	指定変数(論理名)
fac_cd = ?	INパラメータ：部署情報DTO：工場CD
affiliate_name = ?	INパラメータ：入力フォーム：部署名
affiliate_name_r = ?	INパラメータ：入力フォーム：部署名略称
apply_strt_date = ?	INパラメータ：入力フォーム：適用開始日
apply_fin_date = ?	INパラメータ：入力フォーム：適用終了日
create_date = ?	現在時刻
create_id = ?	INパラメータ：部署情報DTO：ユーザーID
update_date = ?	現在時刻
update_id = ?	INパラメータ：部署情報DTO：ユーザーID
affiliate_id = ?	INパラメータ：入力フォーム：部署ID

< 条件 : 1 > エラーが発生した場合

- ・ 戻り値にfalseを設定する。

エラーが発生しない場合

- ・ 戻り値にtrueを設定する。

[2] 処理を完了する。

引数 (論理名)	メソッド名 (論理名)	引数 (物理名)	メソッド名 (物理名)
IN	選択リスト[]	IN	List<String>
OUT	エラー判定	OUT	boolean

[1] SQL文

A) SQL文の設定を行う

```
SELECT
  affiliate_id
FROM
  belonging
WHERE del_flg = 0
AND affiliate_id = ?
LIMIT 1
;
```

B) 指定情報を設定してSQLを選択リストの要素数と同じだけループ処理する

バインド	指定変数(論理名)
affiliate_id = ?	INパラメータ：入力リスト[]

< 条件 : 1 > 要素を取得出来ない場合

- ・ 戻り値にtrueを設定する。

C) 戻り値にfalseを設定する

[2] 処理を完了する。

引数 (論理名)	メソッド名 (論理名)	引数 (物理名)	メソッド名 (物理名)
IN	選択リスト[]	IN	List<String>
IN	部署情報DTO	IN	Bfmk07Dto
OUT	エラー判定	OUT	boolean

詳細設計_仕様

[1] SQL文

A) SQL文の設定を行う

```
UPDATE
  belonging
SET
  del_flg = 1,
  update_date = ?,
  update_id = ?,
  create_div = 'U'
WHERE
  affilicate_id = ?
;
```

B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
update_date = ?	現在時刻
update_id = ?	INパラメータ : 部署情報DTO : ユーザーID
affilicate_id = ?	INパラメータ : 選択リスト[] : 部署ID

< 条件 : 1 > エラーが発生した場合

- ・ 戻り値にfalseを設定する。

< 条件 : 2 > エラーが発生しない場合

- ・ 戻り値にtrueを設定する。

[2] 処理を完了する。

引数 (論理名)	メソッド名 (論理名)	引数 (物理名)	メソッド名 (物理名)
IN	選択リスト []	IN	List<String>
OUT	社員確認 []	OUT	boolean[]
			employeesCheck

[1] SQL文

A) SQL文の設定を行う

```
SELECT
  COUNT(affilicate_id) AS employees
FROM
  user_info
WHERE del_flg = 0
AND affilicate_id = ?
LIMIT 1
;
```

B) 指定情報を設定してSQLを選択リストの要素数と同じだけループ処理する

バインド	指定変数(論理名)
affilicate_id = ?	INパラメータ : 選択リスト[] : 部署ID

< 条件 : 1 > 要素を取得した場合

詳細設計_仕様

- ・ 社員確認[]にtrueを設定する。

< 条件 : 2 > 要素を取得しない場合

- ・ 社員確認[]にfalseを設定する。

[2] 処理を完了する。

引数 (論理名)		メソッド名 (論理名)	引数 (物理名)		メソッド名 (物理名)
IN	部署ID	社員削除処理	IN	String	deleteEmployees
IN	部署情報DTO		IN	Bfmk07Dto	
OUT	エラー判定		OUT	boolean	

[1] SQL文

A) SQL文の設定を行う

```
UPDATE
user_info
SET
del_flg = 1,
create_div= 'U',
update_date = ?,
update_id = ?,
WHERE
affilicate_id = ?
;
```

B) 指定情報を設定してSQLの処理をする

バインド	指定変数(論理名)
update_date = ?	現在時刻
update_id = ?	INパラメータ : 部署情報DTO : ユーザーID
affilicate_id = ?	INパラメータ : 部署ID

< 条件 : 1 > エラーが発生した場合

- ・ 戻り値にtrueを設定する。

C) 戻り値にfalseを設定する。

[2] 処理を完了する。