

# 1 Mapování vazeb

## 1.1 Úvod - upřesnění formulace

Zápisem tohoto typu:

```
SELF.b.isOpposite.isCollection.isUnique ^ SELF.b.a.isCollection.isUnique;
```

Mám na mysli následující:

**b** má protějšek **a** je unikátní kolekcí a zároveň protějšek **a** je unikátní kolekcí.

Ze zápisu tedy vyplývá, že vypisují pouze TRUE vlastnosti daný property.

Zkratka *OC* (Ordering Column) značí sloupec, určující pořadí prvků, tedy vlastnost property `isOrdered`.

*FK[B]* značí cizí klíč ze třídy *B*.

## 1.2 Vlastnosti mapované na prázdnou množinu

Jedná se o dvě základní vlastnosti tříd. Pokud je třída *Embedded*, nebo *Transient*, namapují se jejich property na prázdnou množinu.

- $\pi_0(b) \rightarrow \{0\}$

## 1.3 Jednostranně navigabilní vazby

Třída *A* vidí skrz svoji property na třídu *B*, ale *B* nemá žádnou takovou property a tak třídu *A* nevidí a ani neví, že je třídou *A* viděna. Při jednostranně navigabilním mapování sice třída *B* třídu *A* nevidí, ale v databázi na ni má uloženy FK.

Máme dva případy mapování:

- `SELF.b.isCollection.isOrdered;`  
 $\pi_1(b) \rightarrow \{sloupec, OC\}$
- `SELF.b.isCollection.isUnique;`  
`SELF.b.isCollection;`  
 $\pi_2(b) \rightarrow \{sloupec\}$
- Jako třetí případ můžeme považovat `SELF.b`. Tento případ je ale stejný, jako u případů oboustranně navigabilních, tak bude mezi nimi.

## 1.4 Oboustranně navigabilní vazby

Třída *A* vidí skrz svoji property na třídu *B* a třída vidí skrz svou property na třídu *A*.

Máme sedm případů mapování, které je asi nejlepší rozdělit podle toho, zda je `SELF.b.isOpposite` kolekce, či ne.

#### 1.4.1 SELF.b.isOpposite - isCollection = FALSE

V tomto případě máme obecně už jen dva možné výstupy:

- SELF.b.a;  
SELF.b;  
SELF.b.a.isCollection;  
SELF.b.a.isCollection.isUnique;  
 $\pi_3(b) \rightarrow \{sloupec, FK[A]\}$   
 $\pi_3(a) \rightarrow \{0\}$
- SELF.b.a.isCollection.isOrdered;  
 $\pi_4(b) \rightarrow \{sloupec, OC, FK[A]\}$   
 $\pi_4(a) \rightarrow \{0\}$

#### 1.4.2 SELF.b.isOpposite - isCollection = TRUE

Tato část je poměrně obsáhlá, ale ve finále vede pouze na pět různých mapování:

- SELF.b.isOpposite.isCollection.isOrdered  $\wedge$  SELF.b.a;  
 $\pi_5(b) \rightarrow \{sloupec, OC\}$   
 $\pi_5(a) \rightarrow \{sloupec, FK[B]\}$
- SELF.b.isOpposite.isCollection.isOrdered  $\wedge$  SELF.b.a.isCollection.isOrdered;  
 $\pi_6(b) \rightarrow \{sloupec, OC\}$   
 $\pi_6(a) \rightarrow \{sloupec, OC\}$   
(Vznik vazební tabulky s cizími klíči)
- SELF.b.isOpposite.isCollection.isUnique  $\wedge$  SELF.b.a;  
SELF.b.isOpposite.isCollection  $\wedge$  SELF.b.a;  
 $\pi_7(b) \rightarrow \{0\}$   
 $\pi_7(a) \rightarrow \{sloupec, FK[B]\}$
- SELF.b.isOpposite.isCollection.isUnique  $\wedge$  SELF.b.a.isCollection.isOrdered;  
SELF.b.isOpposite.isCollection  $\wedge$  SELF.b.a.isCollection.isOrdered;  
SELF.b.isOpposite.isCollection.isOrdered  $\wedge$  SELF.b.a.isCollection.isUnique;  
SELF.b.isOpposite.isCollection.isOrdered  $\wedge$  SELF.b.a.isCollection;  
 $\pi_8(b) \rightarrow \{0\}$   
 $\pi_8(a) \rightarrow \{sloupec, OC\}$   
(Vznik vazební tabulky s cizími klíči)
- SELF.b.isOpposite.isCollection.isUnique  $\wedge$  SELF.b.a.isCollection.isUnique;  
SELF.b.isOpposite.isCollection.isUnique  $\wedge$  SELF.b.a.isCollection;  
SELF.b.isOpposite.isCollection  $\wedge$  SELF.b.a.isCollection.isUnique;  
SELF.b.isOpposite.isCollection  $\wedge$  SELF.b.a.isCollection;

$$\pi_9(b) \rightarrow \{0\}$$

$$\pi_9(a) \rightarrow \{0\}$$

(Vznik vazební tabulky s cizími klíči)