

201新しい行の追加(1)(2)(3)(4)

新しい行の追加(1)(2)(3)(4)

values 句を使用

values句を使って行を追加するには、次のようにすればよい。

例 1: values を使った新しい行の追加

```
insert into 従業員
values('999999','99','三屋恵一郎','横浜市','39')
```

ここでvalues句で与えられている5つの定数は、テーブル「従業員」を構成している5つの項目に、しかも、順序通りに対応している。

5つの項目すべてを与えなくても、次のように項目名リストをテーブル名につけ加えれば、指定した項目だけに値を与えることが出来る。

例 2: 特定の項目に値を与える insert

```
insert into 従業員(従業員番号, 氏名, 住所) values('999998', '雪田修一', '稚内市');

insert into 従業員(従業員番号, 氏名, 年齢) values('999997', '堺龍男', '97');

insert into 従業員(従業員番号, 氏名, 住所) values('999996', '中村あずさ', '稚内市若葉台');
```

こうした時、selectで、このテーブルの内容を表示させれば次のような出力が得られるであろう。

```
select *
from 従業員
where 従業員番号 > 1000
```

従業員番号	部門番号	氏名	住所	年齢
999996		中村あずさ	稚内市若葉台	
999997		堺龍男		97
999998		雪田修一	稚内市	
999999	99	三屋恵一郎	横浜市	39

values句で指定されなかった項目では、デフォルト値（この例ではnull）が置かれていることがわかる。ここで、任意の項目を指定してうまく行くわけではないことに注意されたい。

次の例 では、insertに失敗する。

例 3: 項目の指定によっては insert に失敗する

```
insert into 従業員(氏名,住所)
values('姫宮貴子','鎌倉市')
```

実際には、指定した項目がまずい訳ではなく、「従業員番号」を指定しなかったことが失敗の原因である。なぜなら、そのために、「従業員番号」という項目に、nullを 設定しようとしたが、この項目では、nullの指定が許されていなかったからである。

例 4: 今度は insert に成功する。

```
insert into 従業員(従業員番号,氏名,住所)
values('999995','姫宮貴子','鎌倉市')
```

どのような項目にnullが許されないかの議論は、ここでは省略したい。しかし、従業員番号はキー項目である。キー項目はnullを許容しないししないことは明らかであろう。