

```
INSERT INTO 家族 ( 従業員ID, 家族名, 続柄 )
VALUES('000000001', NULL, '子')
```

このSQLを実行すると、NOT NULL制約 (主キー制約) 違反になります。

家族表を定義するときに、PRIMARY KEYを指定しないで、NOT NULLとUNIQUEを用いて、次のように定義した場合も、同様の効果が得られます。

```
CREATE TABLE 家族
( 従業員ID CHAR(9) NOT NULL,
  家族名 VARCHAR(18) NOT NULL,
  続柄 VARCHAR(10),
  UNIQUE ( 従業員ID, 家族名 ) )
```

ただし、PRIMARY KEYを用いて明示的に指定した方が、その表の主キーが何であるかが明確になるという利点があります。

6.5 参照制約 (または外部キー制約) [REFERENCES]

参照制約は、外部キーとそれが参照する主キー (または候補キー) という参照関係が表間 (または表内) にある場合に、外部キーと主キーの間で満たさなければならない条件を満たすように制約することです。ここで、外部キーと主キーの間で満たさなければならない条件とは、外部キーの値は、参照する主キー (または候補キー) の (表中の) いずれかの値と一致するか、またはナル値を含む値でなければならないという条件です。

1つの列または複数の列の組から構成される外部キーを定義することによって、参照制約を定義します。外部キーを定義するときに、同じ構成の主キーをもつ表を指定するか、または同じ構成の候補キーをもつ表とその候補キーの列の並びを指定する必要があります。

1つの列に対する外部キーは、列を定義するときに列制約として指定できます。SQLでの参照制約のための参照指定の形式は、次のとおりです。

REFERENCES 表名 [(列名)]

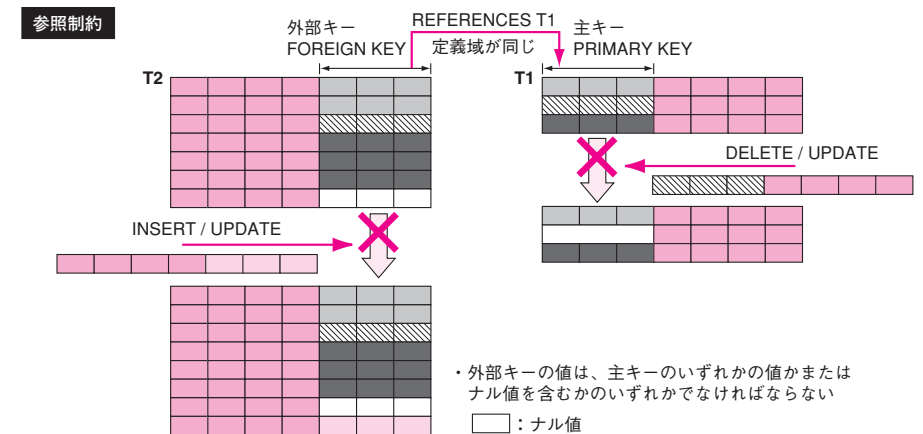
表名には、外部キーが参照するキーをもつ表の名前を指定します。外部キーの参照するキーが指定した表の主キーならば、列名を省略することができます。候補キーを参照するならば、列名を指定する必要があります。

複数の列に対する外部キーは、表を定義するときに表制約として指定できます。SQLでの参照制約のための参照制約定義の形式は、次のとおりです。

FOREIGN KEY(列名 [, 列名] ...)
REFERENCES 表名 [(列名 [, 列名] ...)]

FOREIGN KEYの後の列名の並びで指定した列の組を外部キーとし、REFERENCESの後に指定した表のキーを参照する参照制約が定義されます。参照先の表の列の並びの省略または指定に関する規則は、参照指定と同様です。1つの列でも表制約定義としてこの形式で指定できます。

外部キーをもつ参照表に行を挿入するか行を更新するとき、および外部キーによって参照される被参照表の行を更新するか行を削除をするときに、参照制約が検査されます (図6-2)。



▲ 図6-2 参照制約

【例2】

表6-1 (p.251) の従業員表と家族表を用いて、参照制約を定義する例を示します。

家族表の従業員IDを、従業員表の主キー (PRIMARY KEY) を参照する外部キーとして指定して、家族表を定義するSQLの例を示します。

```
CREATE TABLE 家族
( 従業員ID CHAR(9) REFERENCES 従業員表,
  家族名 VARCHAR(18), 続柄 VARCHAR(10),
  PRIMARY KEY ( 従業員ID, 家族名 ) )
```

上記のように参照制約が定義されているときに、従業員表に未だ格納されていない新入社員 ('000000032', '山本健一') の家族 ('山本恵理') の行を、次のSQLを用いて挿入しようとした場合を仮定します。

同様に、更新規則でSET NULLまたはSET DEFAULTを指定した場合には、それぞれ、自動的にナル値または列の既定値（後述のDEFAULT句で列に対して定義した値）に更新されます。

なお、更新規則で、NO ACTIONまたはRESTRICTを指定した場合は、更新規則を省略した場合と同様に被参照表の行を更新するときに参照制約の条件を満たすように制約されます。

【例3】

表6-1 (p.251) の従業員表と家族表を用いて更新規則を含む参照制約を定義する例を示します。

家族表の従業員IDを、従業員表の主キー（PRIMARY KEY）を参照する外部キーとして指定し、その外部キーによる参照制約での更新規則でCASCADEの参照動作を指定して、家族表を定義するSQLの例を示します。

```
CREATE TABLE 家族
( 従業員ID CHAR(9) REFERENCES 従業員表 ON UPDATE CASCADE,
  家族名 VARCHAR(18), 続柄 VARCHAR(10),
  PRIMARY KEY ( 従業員ID, 家族名 )
```

上記のように参照制約が定義されているときに、何らかの理由で、従業員('000000016', '鈴木一郎')の従業員IDを'000000017'に変更する必要が発生し、次のSQLを用いて従業員表の従業員ID'000000016'を'000000017'に更新しようとした場合を仮定します。

```
UPDATE 従業員表 SET 従業員ID = '000000017'
WHERE 従業員ID = '000000016'
```

このSQLを実行すると、更新規則でCASCADEの参照動作が指定されているので、家族表の従業員ID'000000016'も、自動的に'000000017'に更新され、従業員表と家族表の内容は、次のとおりになります。

▼表6-3 例3の実行結果

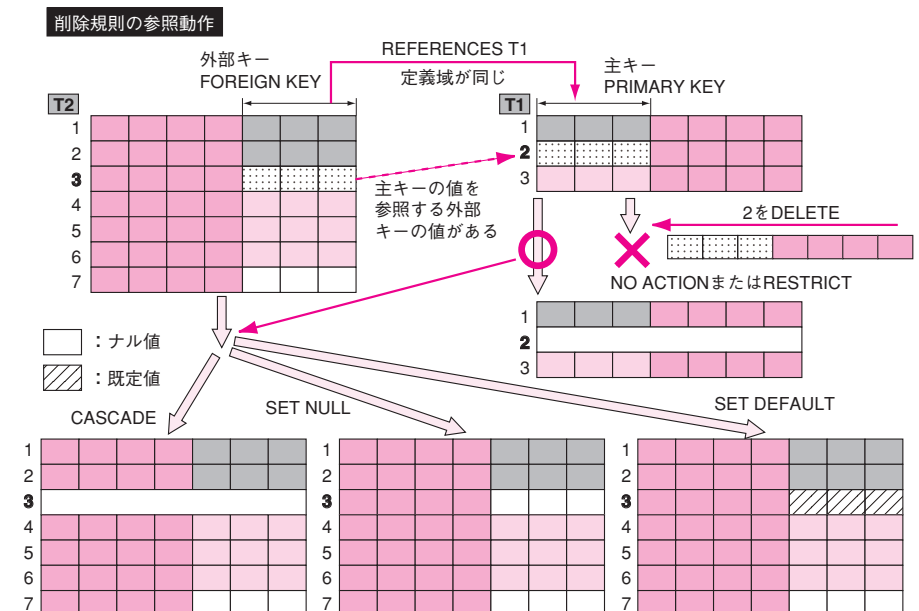
従業員		家族		
従業員ID	従業員名	従業員ID	家族名	続柄
000000001	山田太郎	000000001	山田花子	配偶者
000000009	田中夏美	000000001	山田瑞樹	子
000000018	川上次郎	000000001	山田香織	子
000000017	鈴木一郎	000000009	田中愼一	配偶者
000000002	佐藤春雄	000000017	鈴木太一	子
000000023	中村秋子	000000002	佐藤麗子	配偶者
000000030	本田冬樹	000000002	佐藤桜子	母
000000013	木村純一	000000013	木村恵理	配偶者
000000029	石田直子	000000013	木村幸司	子

なお、更新規則の参照動作としてCASCADEではなく、SET NULLを指定していたならば、家族表の従業員ID'000000016'は、'000000017'ではなく、ナル値に更新しようとしています。しかし、この例の場合には、主キー制約（NOT NULL制約）違反になるため、従業員表の従業員IDを更新できません。同様に、SET DEFAULTを指定していたならば、従業員ID列の既定値に更新しようと思いますが、この例の場合には、主キー制約（一意性制約）違反になる可能性があります。

また、外部キー定義時に、参照被トリガ動作の削除規則としてCASCADEを指定した場合には、外部キーによって参照される被参照表の行を削除すると、参照制約の検査は行われません。代わりに、参照されている主キー（または候補キー）の削除される値と同じ値を外部キーの値としてもつ参照表の行が、自動的に削除されます。

さらに、削除規則として、SET NULLまたはSET DEFAULTを指定した場合には、参照されている主キー（または候補キー）の削除される値と同じ値をもつ、参照表の行の外部キーの値が、更新規則の場合と同様に、それぞれ、自動的にナル値または列の既定値に更新されます（図6-4）。

削除規則で、NO ACTIONまたはRESTRICTを指定した場合は、削除規則を省略した場合と同様に被参照表の行を削除するときに参照制約の条件を満たすように制約されます。



▲図6-4 参照制約での削除規則の参照動作

【例4】

表6-1 (p.251) の従業員表と家族表を用いて、更新規則を含む参照制約を定義する例を示します。

家族表の従業員IDを、従業員表の主キー (PRIMARY KEY) を参照する外部キーとして指定し、その外部キーによる参照制約での**削除規則**でCASCADEの参照動作を指定して、家族表を定義するSQLの例を示します。

```
CREATE TABLE 家族
( 従業員ID CHAR(9) REFERENCES 従業員表 ON DELETE CASCADE,
  家族名 VARCHAR(18), 続柄 VARCHAR(10),
  PRIMARY KEY ( 従業員ID, 家族名 )
```

上記のように参照制約が定義されているときに、従業員 ('000000016', '鈴木一郎') が退職したので、次のSQLを用いて従業員表から従業員IDが'000000016'の行を、家族表から該当行を削除する前に削除しようとした場合を仮定します。

```
DELETE FROM 従業員表 WHERE 従業員ID = '000000016'
```

このSQLを実行すると、削除規則でCASCADEの参照動作が指定されているので、家族表の従業員ID'000000016'の行も、自動的に削除され、従業員表と家族表の内容は、次のとおりになります。

▼表6-4 例4の実行結果

従業員		家族		
従業員ID	従業員名	従業員ID	家族名	続柄
000000001	山田太郎	000000001	山田花子	配偶者
000000009	田中夏美	000000001	山田瑞樹	子
000000018	川上次郎	000000001	山田香織	子
		000000009	田中慎一	配偶者
000000002	佐藤春雄			
000000023	中村秋子	000000002	佐藤麗子	配偶者
000000030	本田冬樹	000000002	佐藤桜子	母
000000013	木村純一	000000013	木村恵理	配偶者
000000029	石田直子	000000013	木村幸司	子

なお、削除規則の参照動作としてCASCADEではなく、SET NULLを指定していたならば、家族表の従業員ID'000000016'の行は、削除されるのではなく、従業員IDをナル値に更新しようとします。しかし、この例の場合には、主キー制約 (NOT NULL制約) 違反になるため、従業員表の行を削除できません。同様に、SET DEFAULTを指定していたならば、従業員ID列の既定値に更新しようとしますが、この例の場合には、主キー制約 (一意性制約) 違反になる可能性があります。

6.7 表検査制約 [CHECK]

表検査制約は、表中の行が満たさなければならない条件を満たすように制約することです。1つの列または複数の列に関する条件を検査制約として定義することができます。

1つの列に対する表検査制約は、列を定義するときに列制約として指定できます。

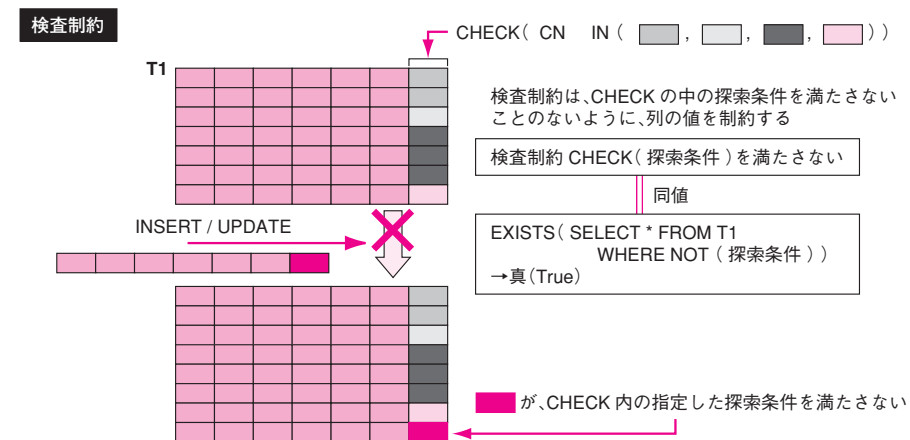
複数の列に対する表検査制約は、表を定義するときに表制約として指定できます。

SQLでの表検査制約のための検査制約定義の形式は、次のとおりです。

CHECK (探索条件)

列定義中で指定する場合、探索条件には、定義している列に対する条件を指定します。

表検査制約を定義した表に行を挿入するか、またはその表の行を更新するときに、表検査制約が検査されます (図6-5)。



▲図6-5 検査制約

【例5】

表6-1 (p.251) の家族表を用いて検査制約を定義する例を示します。

家族表の続柄に設定できる値を'配偶者'、'子'、'父'または'母'のいずれかに制約するために、検査制約を指定して、家族表を定義するSQLの例を示します。