

Vazby mezi tabulkami (získávání dat z více tabulek, druhy spojení)

Vazby mezi tabulkami

Vazby mezi tabulkami lze také nazvat jako *relace* či o nich lze mluvit jako o *kardinalitách vztahu*. Kardinalita vztahu vyjadřuje skutečnost, kolik výskytů jedné entity může vstoupit do vztahu s kolika výskyty druhé entity. Existují tři typy vtaů (vazeb mezi tabulkami).

1:1

Relace typu 1:1 znamená, že právě jednomu záznamu v jedné tabulce odpovídá právě jeden v jiné tabulce. Jednotlivé záznamy v obou tabulkách jsou tak vlastně spojeny přímo.

Údaje z obou tabulek lze však prakticky umístit pouze do jedné tabulky (do jediného záznamu). Relace 1:1 má význam zejména u velmi rozsáhlých tabulek s mnoha sloupci. V takovém případě slouží druhá propojená tabulka jako odlehčená první, zejména pokud se s hodnotami ve druhé tabulce nepracuje příliš často.

U relace 1:1 by měl být nastaven primární klíč na sloupec, kterým budou obě tabulky propojeny.

Příklad: manžel má 1 manželku a ta má 1 manžela.

1:N

Jedná se o jednoznačně nejpoužívanější druh relace. Umožňuje, aby jednomu záznamu v první tabulce odpovídalo více záznamů v tabulce druhé. V praxi to tedy znamená, že v jedné tabulce se nachází nějaký záznam a k němu se v druhé tabulce nachází jeden, více nebo také žádný záznam.

U relace 1:N by měl být primární klíč zvolen v tabulce, která bude tvořit relaci 1, a to na sloupec, který bude propojen s druhou tabulkou.

Příklad: jeden zaměstnanec může mít N průchodu turniketem (příchodů a odchodů do práce).

N:M

N:M je méně častým typem relace, ovšem v určitých případech jediným, který může danou situaci řešit. Tato relace umožňuje, aby několika záznamům v první tabulce odpovídalo několik záznamů v tabulce druhé. Na rozdíl od předchozích dvou typů relací je u této nutné vytvořit tzv. *spojovací tabulku*. Jedná se o tabulku, díky níž je relaci vůbec možné uskutečnit. Této relace se dosahuje právě pomocí spojovací tabulky, která spojuje dvě relace typu 1:N.

Příklad: pacienti přišli v určité dny k lékaři, který jim každý den předepsal nějaký lék. Dvěma různým pacientům však mohl předepsat lék se stejným ID. Informace o takovém léku jsou pak uloženy v další tabulce.

Získávání dat z více tabulek

Chceme-li získat data z více tabulek a obdržet výsledek jako jeden řetězec díky jedinému dorazu, je nutné využít funkci spojování tabulek. Existuje několik druhů spojení, ty jsou uvedeny dále.

Druhy spojení

Klauzule WHERE

Jde o nejjednodušší způsob propojení (avšak nejvýše dvou) tabulek. Tabulky jsou propojeny na základě jedné společné hodnoty. Viz následující příklad.

```
SELECT z.id_zakaznika, z.jmeno, o.id_objednavky, o.cena FROM zakaznici z,
objednavky o WHERE z.id_zakaznika = o.id_zakaznika;
```

JOIN obecně

Umožňuje více způsobů spojení. Obecně vypadá jeho klauzule následovně.

```
SELECT [*] [seznam_polozek_vystupni_sestavy] FROM prvni_tabulka DRUH_SPOJENI
druha_tabulka [ON podminka_spojeni];
```

V podmínce spojení musí být slučovaná pole (primární klíč & cizí klíč) stejného datového typu, musí obsahovat stejný druh dat, nemusí však mít stejný název.

INNER JOIN

Viz příklad u klauzule WHERE (výlučné spojení) – spojení je provedeno na základě shody jednoho nebo více společných atributů; změnou pořadí tabulek dostaneme stejný výběr, jen se změněným pořadím.

Je možné porovnávat více společných polí tabulek náhradou klíčového slova ON za USING.

```
ON z.id_zakaznika = o.id_zakaznika → USING(id_zakaznika)
```

LEFT [OUTER] JOIN

Z první (levé) tabulky se ve výpisu objeví všechny řádky, z druhé (pravé) pouze řádky splňující podmínku (tj. jen objednávky mající zákazníka).

```
SELECT z.id_zakaznika, z.jmeno, o.id_objednavky, o.cena FROM zakaznici z LEFT
[OUTER] JOIN objednavky o ON z.id_zakaznika = o.id_zakaznika;
```

RIGHT [OUTER] JOIN

V tomto případě bude zahrnuta celá tabulka druhá/pravá. Stejného výsledku lze dosáhnout také levým spojením (LEFT JOIN) a prohozením pořadí tabulek.

```
SELECT z.id_zakaznika, z.jmeno, o.id_objednavky, o.cena FROM zakaznici z RIGHT
[OUTER] JOIN objednavky o ON z.id_zakaznika = o.id_zakaznika;
```

NATURAL LEFT JOIN

Jako součást spojení budou použita všechna shodná pole obou zdrojových tabulek.

```
SELECT z.id_zakaznika, z.jmeno, o.id_objednavky, o.cena FROM zakaznici z NATURAL
LEFT JOIN objednavky o;
```

FULL JOIN

Vypíše z obou tabulek i řádky, které spojovací kritéria nespĺňují (v MySQL zatím nelze použít).

```
SELECT z.id_zakaznika, z.jmeno, o.id_objednavky, o.cena FROM zakaznici z FULL JOIN
objednavky o ON z.id_zakaznika = o.id_zakaznika;
```