

SQL II – spojování tabulek

Práce s daty, 16. března 2023

Kartézský součin

- Množinová operace
- Výsledkem je množina, která obsahuje všechny uspořádané dvojice prvků z první a ze druhé množiny (všechny možné kombinace)

A
1
2

B
3
4

A × B	
1	3
1	4
2	3
2	4

```
SELECT * FROM A, B;
```

```
SELECT * FROM A  
CROSS JOIN B;
```

Spojování tabulek

- Běžný způsob spojování tabulek (JOIN) pracuje následovně: každý řádek z první tabulky je porovnán s každým řádkem druhé tabulky. Pokud hodnoty z obou řádků vyhoví spojovací podmínce, vytvoří se nový řádek kombinující hodnoty z obou tabulek dle zadání.
 - Spojovací podmínka obvykle testuje shodu, tj. používá se operátor „=“
 - Běžné spojení se nazývá vnitřní (INNER JOIN)
- Následující zápisy spojení tabulek jsou ekvivalentní:
 - `SELECT item1, item2 FROM t1 JOIN t2 ON id1=id2;`
 - `SELECT item1, item2 FROM t1 INNER JOIN t2 ON id1=id2;`
 - `SELECT item1, item2 FROM t1, t2 WHERE id1=id2;`

Spojování více tabulek

Prostě kombinujeme tabulky a podmínky spojení dle potřeby:

- `SELECT id1, item1, item2, item3 FROM t1, t2, t3 WHERE id1=id2 AND id2=id3;`
- `SELECT id1, item1, item2, item3 FROM t1 INNER JOIN t2 INNER JOIN t3 ON id2=id3 ON id1=id2;`

V případě nejednoznačného názvu atributu:

- `SELECT t3.item_id, item3, item4 FROM t3, t4 WHERE t3.item_id=t4.item_id;`
- `SELECT t3.item_id, item3, item4 FROM t3 JOIN t4 USING (item_id);`

Klíče

Spojovací podmínka obecně může kombinovat libovolný počet atributů ze spojovaných tabulek. V praxi je však užitečné kombinovat zejména takové atributy, které plní funkce klíče:

- Vlastní klíč
 - Atribut nebo atributy, jejichž kombinace hodnot je v rámci tabulky unikátní
 - Např. UČO, rodné číslo, e-mail, jméno + věk + adresa
- Primární klíč
 - Vlastní klíč, který je chápán jako hlavní dané tabulce
 - Obvykle indexované číslo, které se automaticky zvyšuje
- Cizí klíč
 - Atribut v jiné tabulce sloužící ke spojení s primární tabulkou

Cvičení I

Ve cvičné databázi na [W3schools](https://www.w3schools.com/) vypište:

1. Ke každému názvu produktu zemi dodavatele.
2. Seznam názvů dodavatelů a kategorií, které dodávají. Bez duplicit.
3. Jména zaměstnanců společně s názvy dopravních společností, kteří vyřizovali objednávky zákazníků z Mexika. Bez duplicit, seřazený dle příjmení.
4. Názvy dodavatelů a jména zákazníků, kteří si objednali zboží od dodavatelů sídlících ve stejné zemi, jako oni sami bydlí. Vypsát včetně názvů dodavatelů a zemí.

Cvičení I – řešení, 1. část

1. Ke každému názvu produktu zemi dodavatele.

```
SELECT ProductName, Country FROM Products INNER JOIN  
Suppliers ON Products.SupplierID=Suppliers.SupplierID;
```

nebo

```
SELECT ProductName, Country FROM Products, Suppliers WHERE  
Products.SupplierID=Suppliers.SupplierID;
```

2. Seznam názvů dodavatelů a kategorií, které dodávají. Bez duplicit.

```
SELECT DISTINCT SupplierName, CategoryName FROM Products,  
Suppliers, Categories WHERE  
Products.SupplierID=Suppliers.SupplierID AND  
Products.CategoryID=Categories.CategoryID;
```

Cvičení I – řešení, 2. část

1. Jména zaměstnanců společně s názvy dopravních společností, kteří vyřizovali objednávky zákazníků z Mexika. Bez duplicit, dle příjmení.

```
SELECT DISTINCT FirstName, LastName, ShipperName FROM Employees, Orders, Customers, Shippers WHERE Orders.EmployeeID=Employees.EmployeeID AND Orders.CustomerID=Customers.CustomerID AND Orders.ShipperID=Shippers.ShipperID AND Country LIKE 'Mexico' ORDER BY LastName;
```
2. Názvy dodavatelů a jména zákazníků, kteří si objednali zboží od dodavatelů sídlících ve stejné zemi, jako oni sami bydlí.

```
SELECT DISTINCT SupplierName, CustomerName, Customers.Country FROM Suppliers, Products, Orders, OrderDetails, Customers WHERE Suppliers.SupplierID= Products.SupplierID AND Products.ProductID= OrderDetails.ProductID AND OrderDetails.OrderID=Orders.OrderID AND Orders.CustomerID=Customers.CustomerID AND Suppliers.Country=Customers.Country;
```


Jednostranné spojení

- Jednostranné spojování tabulek (LEFT JOIN nebo RIGHT JOIN): každý řádek z první tabulky je porovnán s každým řádkem druhé tabulky. Pokud hodnoty z obou řádků vyhoví spojovací podmínce, vytvoří se nový řádek kombinující hodnoty z obou tabulek dle zadání. Pokud LEFT JOIN nenajde odpovídající řádek ve druhé (pravé) tabulce, vytvoří se nový řádek, kde atributy z druhé tabulky budou NULL.
 - RIGHT JOIN funguje úplně stejně, pouze v opačném pořadí
- V tomto případě se z tabulky t1 (resp. t2) vypíše všechny řádky:
 - `SELECT item1, item2 FROM t1 LEFT JOIN t2 ON id1=id2;`
 - `SELECT item1, item2 FROM t1 RIGHT JOIN t2 ON id1=id2;`

Vychytávky

1. Jedna tabulka se v dotazu může **vyskytovat vícekrát**

- Např. při vzájemném hodnocení lze spojit tabulku se jmény uživatelů jak pro informaci o hodnoceném, tak pro informaci o hodnotiteli
- Na dublovanou tabulku se odvoláváme pomocí jejich aliasů
- Tabulka může být spojena i sama se sebou:
- `SELECT P.Jmeno, N.Jmeno FROM Zamestnanci P JOIN Zamestnanci N ON P.Nadrizeny=N.Id;`

2. Spojení mezi tabulkami je možné si **uložit jako pohledy (VIEW)**

- A s pohledy pak lze dále pracovat stejně jako s tabulkami:
- `CREATE VIEW pohled AS SELECT ...`
- `DROP VIEW pohled`

Cvičení II

Ve cvičné databázi na W3schools vypište:

1. Všechny(!) zákazníky jako dvojice *jméno a země*, plus *název dodavatele*, pokud je dodavatel ze stejné země jako zákazník.
2. Jména zákazníků (dvojice), kteří jsou ze stejného města.
3. *Datum objednávky* a dvojici *ID objednávek*, pro které platí, že byly objednány stejný den a zároveň hodnota ID první vypsané objednávky je větší než hodnota ID druhé vypsané objednávky.
4. Cokoliv dalšího, co vás napadne, že by bylo možné vypsát za sestavu.

Cvičení II – řešení, 1. část

1. Všechny(!) zákazníky jako dvojice *jméno a země*, plus *název dodavatele*, pokud je dodavatel ze stejné země jako zákazník.

```
SELECT CustomerName, Customers.Country, SupplierName FROM  
Customers LEFT JOIN Suppliers ON  
Customers.Country=Suppliers.Country;
```
2. Jména zákazníků (dvojice), kteří jsou ze stejného města.

```
SELECT A.CustomerName AS CustomerName1, B.CustomerName AS  
CustomerName2, A.City FROM Customers A, Customers B WHERE  
A.CustomerID > B.CustomerID AND A.City = B.City;
```

Cvičení II – řešení, 2. část

1. *Datum objednávky a dvojici ID objednávek, pro které platí, že byly objednány stejný den a zároveň hodnota ID první vypsané objednávky je větší než hodnota ID druhé vypsané objednávky.*
`SELECT A.OrderDate, A.OrderID, B.OrderID FROM Orders A,
Orders B WHERE A.OrderDate=B.OrderDate AND
A.OrderID>B.OrderID;`
2. Cokoliv dalšího, co vás napadne, že by bylo možné vypsát za sestavu.
:-)

Odkazy

Z MariaDB / MySQL Knowledge Base:

- [Spojování tabulek](#)
- [Klíče tabulek v relační databázi](#)
- [SELF JOIN](#)
- [Vytváření a používání pohledů](#)