

## Bazy Danych

### Ćwiczenie 8: Podstawy języka SQL, część 3, złączenia tabel

opracował: dr hab. inż. Artur Gramacki (a.gramacki@issi.uz.zgora.pl)

1. Poruszane zagadnienia: polecenie `SELECT`, złączenia tabel, iloczyn kartezjański, złączenia równościowe, złączenia nierównościowe, złączenia zewnętrzne, operatory `UNION` oraz `UNION ALL`.
2. Warunkiem uzyskania w ćwiczeniu takich jak poniżej wyników jest praca na demonstracyjnym schemacie relacyjnym, o którym mowa jest w jednym z poprzednich ćwiczeń. Dlatego też należy upewnić się, że odpowiedni skrypt wykonał się bezbłędnie (poprawnie utworzyły się wszystkie tabele oraz załadowały się wszystkie dane). Dla pewności warto przed rozpoczęciem ćwiczenia po prostu utworzyć schemat od nowa, nawet jeżeli wydaje ci się, że ten, który masz aktualnie zainstalowany jest poprawny.

#### Zadanie 1

Wyznaczyć iloczyn kartezjański tabel `dept` oraz `emp`. Ile rekordów wynikowych otrzymamy? Czy wynik ma jakąś wartość praktyczną? Jeżeli tak, to jaką?

#### Zadanie 2

Powtórzyć powyższe ćwiczenie dla tabel `ord`, `item` oraz `product`. Uwaga na możliwość zawieszenia się aplikacji klienckiej a być może i całego serwera MySQL (dlaczego?).

#### Zadanie 3

Wyświetlić dane o pracownikach pracujących w dziale o nazwie `Sales`. Dla tabel zdefiniować aliasy i posługiwać się nimi przy odwoływaniu do kolumn. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+
| first_name | last_name | name |
+-----+-----+-----+
| Midori     | Nagayama  | Sales |
| Colin     | Magee     | Sales |
| Henry     | Giljum    | Sales |
| Yasmin    | Sedeghi   | Sales |
| Mai       | Nguyen    | Sales |
| Radha     | Patel     | Sales |
| Andre     | Dumas     | Sales |
+-----+-----+-----+
7 rows in set (0.05 sec)
```

#### Zadanie 4

Zapytanie z poprzedniego punktu przerobić w taki sposób, aby było błędne i generowało błąd świadczący o niepoprawnym odwołaniu się do nazwy kolumny (*ERROR 1052 (23000): Column '%s' In %s is ambiguous*). Czy potrafisz powiedzieć jak taki błąd świadomie wywołać, zanim jeszcze zaczniesz modyfikować zapytanie?

#### Zadanie 5

Wyświetlić dane o pracownikach pracujących w regionie o nazwie *Europe*. Dla tabel zdefiniować aliasy i posługiwać się nimi przy odwoływaniu się do kolumn. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+-----+
| first_name | last_name | Dzial      | Region |
+-----+-----+-----+-----+
| Andre      | Dumas     | Sales      | Europe |
| Marta      | Havel     | Operations | Europe |
| Bela       | Dancs     | Operations | Europe |
| Sylvie     | Schwartz  | Operations | Europe |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

#### Zadanie 6

Wyświetlić stany magazynowe wszystkich produktów z rozbiem na poszczególne hurtownie (wymagane dane są w tabelach *warehouse*, *product*, *inventory*, *emp*). Ograniczamy się tylko do tych produktów, których sprzedaż, czyli różnica wartości pól *inventory.max\_in\_stock* oraz *inventory.amount\_in\_stock* jest mniejsza niż 10. Dane posortować wg. nazwy hurtowni a następnie wg. nazwy produktu. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+-----+-----+-----+
| Adres hurt. | Szef      | Nazwa produktu      | Stan biez. | Stan max | Roznica |
+-----+-----+-----+-----+-----+-----+
| 283 King Street | Molly Urguhart | Junior Soccer Ball | 993 | 1000 | 7 |
| 283 King Street | Molly Urguhart | Winfield Bat       | 173 | 175 | 2 |
| 68 Via Centrale | Roberta Menchu | Alomar Glove       | 98 | 100 | 2 |
| 68 Via Centrale | Roberta Menchu | Black Hawk Knee Pads | 175 | 175 | 0 |
| 68 Via Centrale | Roberta Menchu | Safe-T Helmet      | 132 | 140 | 8 |
| 68 Via Centrale | Roberta Menchu | Winfield Bat       | 97 | 100 | 3 |
| 6921 King Way   | Ben Biri   | Alexeyer Pro Lifting Bar | 70 | 70 | 0 |
| 6921 King Way   | Ben Biri   | New Air Pump       | 35 | 35 | 0 |
| 6921 King Way   | Ben Biri   | Pro Curling Bar    | 65 | 70 | 5 |
| 6921 King Way   | Ben Biri   | Prostar 20 Pound Weight | 61 | 70 | 9 |
| 86 Chu Street   | Antoinette Catchpole | Himalaya Tires     | 135 | 140 | 5 |
| 86 Chu Street   | Antoinette Catchpole | Safe-T Helmet      | 250 | 250 | 0 |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.03 sec)
```

#### Zadanie 7

Utworzyć tabelę o nazwie *price\_category*. Pokazano ją na poniższym rysunku. W tabeli tej powinny pojawić się następujące rekordy (musisz je sam wprowadzić):

```
+-----+-----+-----+-----+
| id | name          | price_min | price_max |
+-----+-----+-----+-----+
| 1  | Low Summer   | 0         | 100       |
| 2  | Medium Summer | 101       | 500       |
| 3  | High Summer  | 501       | 2000      |
```

4	Low Winter	0	50
5	Medium Winter	51	300
6	High Winter	301	2000

Ich sens jest następujący: ustalamy odpowiednie przedziały cenowe i przypisujemy im nazwy. Ustaliliśmy więc 6 takich przedziałów - 3 dla sezonu letniego i 3 dla zimowego. Dane z tej tabeli wykorzystamy do oceny w jakich przedziałach cenowych mieszczą się poszczególne produkty (tabela *product*).

Następnie należy wykonać zapytanie, które nazywa się *połączenie nierównościowe* (ang. *theta-join*). Złączamy ze sobą relacje, które nie są powiązane ze sobą więzami integralnościowymi. W poniższym przykładzie wyświetlono listę produktów oraz na bazie tabeli *price\_category* sprawdzono, w jakich przedziałach cenowych mieszczą się ceny poszczególnych produktów. Czy potrafisz sensownie zinterpretować otrzymane wyniki? Kiedy w wyniku wykonania zapytania otrzymamy dokładnie tyle wierszy, ile rekordów zawiera tabela *product*? Kiedy zapytanie nie zwróci ani jednego rekordu?

Nazwa produktu	Nazwa przedziału	Przedział	Cena
Ace Ski Boot	Medium Summer	101.00 -- 500.00	200.00
Ace Ski Boot	Medium Winter	51.00 -- 300.00	200.00
Ace Ski Pole	Low Summer	0.00 -- 100.00	21.95
Ace Ski Pole	Low Winter	0.00 -- 50.00	21.95
Alexeyer Pro Lifting Bar	Low Summer	0.00 -- 100.00	65.00
Alexeyer Pro Lifting Bar	Medium Winter	51.00 -- 300.00	65.00
Alomar Glove	Low Summer	0.00 -- 100.00	75.00
Alomar Glove	Medium Winter	51.00 -- 300.00	75.00
Black Hawk Elbow Pads	Low Summer	0.00 -- 100.00	8.00
Black Hawk Elbow Pads	Low Winter	0.00 -- 50.00	8.00
Black Hawk Knee Pads	Low Summer	0.00 -- 100.00	9.00
Black Hawk Knee Pads	Low Winter	0.00 -- 50.00	9.00
Bunny Boot	Medium Summer	101.00 -- 500.00	150.00
Bunny Boot	Medium Winter	51.00 -- 300.00	150.00
Bunny Ski Pole	Low Summer	0.00 -- 100.00	16.25
Bunny Ski Pole	Low Winter	0.00 -- 50.00	16.25
Cabrera Bat	Low Summer	0.00 -- 100.00	45.00
Cabrera Bat	Low Winter	0.00 -- 50.00	45.00
Chapman Helmet	Low Summer	0.00 -- 100.00	22.89
Chapman Helmet	Low Winter	0.00 -- 50.00	22.89
Grand Prix Bicycle	High Summer	501.00 -- 2000.00	1669.00
Grand Prix Bicycle	High Winter	301.00 -- 2000.00	1669.00
Grand Prix Bicycle Tires	Low Summer	0.00 -- 100.00	16.00
Grand Prix Bicycle Tires	Low Winter	0.00 -- 50.00	16.00
Griffey Glove	Low Summer	0.00 -- 100.00	80.00
Griffey Glove	Medium Winter	51.00 -- 300.00	80.00
Himalaya Bicycle	High Summer	501.00 -- 2000.00	582.00
Himalaya Bicycle	High Winter	301.00 -- 2000.00	582.00
Himalaya Tires	Low Summer	0.00 -- 100.00	18.25
Himalaya Tires	Low Winter	0.00 -- 50.00	18.25
Junior Soccer Ball	Low Summer	0.00 -- 100.00	11.00
Junior Soccer Ball	Low Winter	0.00 -- 50.00	11.00
Major League Baseball	Low Summer	0.00 -- 100.00	4.29
Major League Baseball	Low Winter	0.00 -- 50.00	4.29
New Air Pump	Low Summer	0.00 -- 100.00	20.00

New Air Pump	Low Winter	0.00 -- 50.00	20.00
Pro Curling Bar	Low Summer	0.00 -- 100.00	50.00
Pro Curling Bar	Low Winter	0.00 -- 50.00	50.00
Pro Ski Boot	Medium Summer	101.00 -- 500.00	410.00
Pro Ski Boot	High Winter	301.00 -- 2000.00	410.00
Pro Ski Pole	Low Summer	0.00 -- 100.00	40.95
Pro Ski Pole	Low Winter	0.00 -- 50.00	40.95
Prostar 10 Pound Weight	Low Summer	0.00 -- 100.00	8.00
Prostar 10 Pound Weight	Low Winter	0.00 -- 50.00	8.00
Prostar 100 Pound Weight	Low Summer	0.00 -- 100.00	45.00
Prostar 100 Pound Weight	Low Winter	0.00 -- 50.00	45.00
Prostar 20 Pound Weight	Low Summer	0.00 -- 100.00	12.00
Prostar 20 Pound Weight	Low Winter	0.00 -- 50.00	12.00
Prostar 50 Pound Weight	Low Summer	0.00 -- 100.00	25.00
Prostar 50 Pound Weight	Low Winter	0.00 -- 50.00	25.00
Prostar 80 Pound Weight	Low Summer	0.00 -- 100.00	35.00
Prostar 80 Pound Weight	Low Winter	0.00 -- 50.00	35.00
Puckett Bat	Low Summer	0.00 -- 100.00	47.00
Puckett Bat	Low Winter	0.00 -- 50.00	47.00
Safe-T Helmet	Low Summer	0.00 -- 100.00	60.00
Safe-T Helmet	Medium Winter	51.00 -- 300.00	60.00
Slaker Water Bottle	Low Summer	0.00 -- 100.00	7.00
Slaker Water Bottle	Low Winter	0.00 -- 50.00	7.00
Steinbach Glove	Low Summer	0.00 -- 100.00	80.00
Steinbach Glove	Medium Winter	51.00 -- 300.00	80.00
Winfield Bat	Low Summer	0.00 -- 100.00	50.00
Winfield Bat	Low Winter	0.00 -- 50.00	50.00
World Cup Net	Medium Summer	101.00 -- 500.00	123.00
World Cup Net	Medium Winter	51.00 -- 300.00	123.00
World Cup Soccer Ball	Low Summer	0.00 -- 100.00	28.00
World Cup Soccer Ball	Low Winter	0.00 -- 50.00	28.00

+-----+-----+-----+-----+  
66 rows in set (0.02 sec)

### Zadanie 8

Z tabeli *emp* wyświetlić dane o pracownikach (nazwisko pracownika, nazwisko bezpośredniego przełożonego). Należy zauważyć, że w tabeli *emp* na kolumnie *manager\_id* założony jest klucz obcy z referencją do klucza głównego *id* w tej samej tabeli.

W kolumnie *manager\_id* wpisany jest identyfikator *szefa* danego pracownika. Zwróćmy uwagę, że pole *manager\_id* dla pracownika na stanowisku *President* ma wartość *NULL*, co oznacza, że nie ma on swojego zwierzchnika. Widać to wyraźnie, gdy wyświetlimy odpowiednie kolumny z tabeli *emp*:

last_name	id	manager_id
Velasquez	1	NULL
Ngao	2	1
Nagayama	3	1
Quick-To-See	4	1
Ropeburn	5	1
Urguhart	6	2
Menchu	7	2
Biri	8	2
Catchpole	9	2

```

| Havel          | 10 |          2 |
| Magee         | 11 |          3 |
| Giljum        | 12 |          3 |
| Sedeghi       | 13 |          3 |
| Nguyen        | 14 |          3 |
| Dumas         | 15 |          3 |
| Maduro        | 16 |          6 |
| Smith         | 17 |          6 |
| Nozaki        | 18 |          7 |
| Patel         | 19 |          7 |
| Newman        | 20 |          8 |
| Markarian     | 21 |          8 |
| Chang         | 22 |          9 |
| Patel         | 23 |          9 |
| Dancs         | 24 |         10 |
| Schwartz      | 25 |         10 |
+-----+-----+
25 rows in set (0.00 sec)

```

Oczekiwany wynik pokazano poniżej:

```

+-----+-----+-----+-----+
| last_name | name | last_name | name |
+-----+-----+-----+-----+
| Magee     | Sales Representative | Nagayama | VP, Sales |
| Giljum    | Sales Representative | Nagayama | VP, Sales |
| Sedeghi   | Sales Representative | Nagayama | VP, Sales |
| Nguyen    | Sales Representative | Nagayama | VP, Sales |
| Dumas     | Sales Representative | Nagayama | VP, Sales |
| Maduro    | Stock Clerk         | Urguhart | Warehouse Manager |
| Smith     | Stock Clerk         | Urguhart | Warehouse Manager |
| Nozaki    | Stock Clerk         | Menchu   | Warehouse Manager |
| Patel     | Stock Clerk         | Menchu   | Warehouse Manager |
| Newman    | Stock Clerk         | Biri     | Warehouse Manager |
| Markarian | Stock Clerk         | Biri     | Warehouse Manager |
| Chang     | Stock Clerk         | Catchpole | Warehouse Manager |
| Patel     | Stock Clerk         | Catchpole | Warehouse Manager |
| Dancs     | Stock Clerk         | Havel    | Warehouse Manager |
| Schwartz  | Stock Clerk         | Havel    | Warehouse Manager |
| Ropeburn  | VP, Administration  | Velasquez | President |
| Quick-To-See | VP, Finance        | Velasquez | President |
| Ngao      | VP, Operations      | Velasquez | President |
| Nagayama  | VP, Sales           | Velasquez | President |
| Urguhart  | Warehouse Manager   | Ngao     | VP, Operations |
| Menchu    | Warehouse Manager   | Ngao     | VP, Operations |
| Biri      | Warehouse Manager   | Ngao     | VP, Operations |
| Catchpole | Warehouse Manager   | Ngao     | VP, Operations |
| Havel     | Warehouse Manager   | Ngao     | VP, Operations |
+-----+-----+-----+-----+
24 rows in set (0.00 sec)

```

### Zadanie 9

Z tabeli *ord* w rekordach, gdzie *id* < 100 usunąć dane o „opiekunach” tych zamówień (użyć polecenia UPDATE. O jaką kolumnę chodzi?). Następnie należy wyświetlić numery zamówień, daty ich złożenia oraz imię i nazwisko opiekuna danego zamówienia. Użyć operator LEFT OUTER

JOIN. Do uzyskania napisów *--brak danych--* użyć funkcji IFNULL. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+-----+
| id | Data      | Imie      | Nazwisko  |
+-----+-----+-----+-----+
| 97 | 28-08-1992 | --brak danych-- | --brak danych-- |
| 98 | 31-08-1992 | --brak danych-- | --brak danych-- |
| 99 | 31-08-1992 | --brak danych-- | --brak danych-- |
| 100 | 31-08-1992 | Colin     | Magee     |
| 101 | 31-08-1992 | Mai      | Nguyen    |
| 102 | 01-09-1992 | Andre    | Dumas     |
| 103 | 02-09-1992 | Andre    | Dumas     |
| 104 | 03-09-1992 | Andre    | Dumas     |
| 105 | 04-09-1992 | Colin     | Magee     |
| 106 | 07-09-1992 | Henry    | Giljum    |
| 107 | 07-09-1992 | Andre    | Dumas     |
| 108 | 07-09-1992 | Yasmin   | Sedeghi   |
| 109 | 08-09-1992 | Colin     | Magee     |
| 110 | 09-09-1992 | Colin     | Magee     |
| 111 | 09-09-1992 | Colin     | Magee     |
| 112 | 31-08-1992 | Henry    | Giljum    |
+-----+-----+-----+-----+
16 rows in set (0.00 sec)
```

Dane z tabeli *ord* po modyfikacji wyglądają następująco:

```
+-----+-----+
| id | sales_rep_id |
+-----+-----+
| 97 | NULL        |
| 98 | NULL        |
| 99 | NULL        |
| 100 | 11         |
| 101 | 14         |
| 102 | 15         |
| 103 | 15         |
| 104 | 15         |
| 105 | 11         |
| 106 | 12         |
| 107 | 15         |
| 108 | 13         |
| 109 | 11         |
| 110 | 11         |
| 111 | 11         |
| 112 | 12         |
+-----+-----+
16 rows in set (0.00 sec)
```

Przed rozpoczęciem kolejnego ćwiczenia nie zapomnij przywrócić tabeli *ord* do pierwotnego stanu.

### Zadanie 10

Wyświetlić dane o pracownikach (tabela *emp*) oraz zamówieniach (numer *id* oraz data złożenia zamówienia), którymi się opiekują. Pamiętajmy o tym, że jeden pracownik może opiekować się więcej niż jednym zamówieniem. Jak należy interpretować brak danych w polach *id* oraz

*date\_ordered*? Jak należy interpretować rekordy z tymi samymi nazwiskami? Jako wynik powinniśmy otrzymać:

```

+-----+-----+-----+-----+
| first_name | last_name  | id  | date_ordered |
+-----+-----+-----+-----+
| Ben        | Biri       | NULL | NULL         |
| Antoinette | Catchpole  | NULL | NULL         |
| Eddie      | Chang      | NULL | NULL         |
| Bela       | Dancs      | NULL | NULL         |
| Andre      | Dumas      | 102 | 01-09-1992  |
| Andre      | Dumas      | 103 | 02-09-1992  |
| Andre      | Dumas      | 104 | 03-09-1992  |
| Andre      | Dumas      | 107 | 07-09-1992  |
| Henry      | Giljum     | 97  | 28-08-1992  |
| Henry      | Giljum     | 106 | 07-09-1992  |
| Henry      | Giljum     | 112 | 31-08-1992  |
| Marta      | Havel      | NULL | NULL         |
| Elena      | Maduro     | NULL | NULL         |
| Colin      | Magee     | 100 | 31-08-1992  |
| Colin      | Magee     | 105 | 04-09-1992  |
| Colin      | Magee     | 109 | 08-09-1992  |
| Colin      | Magee     | 110 | 09-09-1992  |
| Colin      | Magee     | 111 | 09-09-1992  |
| Alexander  | Markarian  | NULL | NULL         |
| Roberta    | Menchu     | NULL | NULL         |
| Midori     | Nagayama   | NULL | NULL         |
| Chad       | Newman     | NULL | NULL         |
| LaDoris    | Ngao       | NULL | NULL         |
| Mai        | Nguyen     | 98  | 31-08-1992  |
| Mai        | Nguyen     | 99  | 31-08-1992  |
| Mai        | Nguyen     | 101 | 31-08-1992  |
| Akira      | Nozaki     | NULL | NULL         |
| Radha      | Patel      | NULL | NULL         |
| Vikram     | Patel      | NULL | NULL         |
| Mark       | Quick-To-See | NULL | NULL         |
| Audry      | Ropeburn   | NULL | NULL         |
| Sylvie     | Schwartz   | NULL | NULL         |
| Yasmin     | Sedeghi   | 108 | 07-09-1992  |
| George     | Smith      | NULL | NULL         |
| Molly      | Urganhart  | NULL | NULL         |
| Carmen     | Velasquez  | NULL | NULL         |
+-----+-----+-----+-----+

```

36 rows in set (0.00 sec)

### Zadanie 11

Połączyć w jeden zbiorczy wynik dane z tabel *dept* oraz *region*. Użyć operatora UNION. Następnie zamienić UNION na UNION ALL i opisać jakie zaszły zmiany. Jako wynik powinniśmy otrzymać:

```

+-----+
| name          |
+-----+
| Africa / Middle East |
| Asia          |
| Europe        |
| North America |

```

```

| South America      |
| Administration    |
| Finance           |
| Operations        |
| Sales             |
+-----+
9 rows in set (0.01 sec)

```

oraz

```

+-----+
| name              |
+-----+
| Africa / Middle East |
| Asia              |
| Europe            |
| North America     |
| South America     |
| Administration    |
| Finance           |
| Operations        |
| Operations        |
| Operations        |
| Operations        |
| Operations        |
| Sales             |
| Sales             |
| Sales             |
| Sales             |
| Sales             |
+-----+
17 rows in set (0.00 sec)

```

### Zadanie 12

Połączyć w jeden zbiorczy wynik nazwiska pracowników z tabeli *emp* oraz nazwy klientów z tabeli *customer*. Użyć operatora UNION oraz funkcji CONCAT do odpowiedniego sformatowania wyjścia. Posortować wynik od A do Z. Jako wynik powinniśmy otrzymać:

```

+-----+
| Pracownicy i klienci |
+-----+
| Beisbol Si! <-- customer |
| Big John's Sports Emporium <-- customer |
| Biri <-- emp |
| Catchpole <-- emp |
| Chang <-- emp |
| Dancs <-- emp |
| Delhi Sports <-- customer |
| Dumas <-- emp |
| Futbol Sonora <-- customer |
| Giljum <-- emp |
| Hamada Sport <-- customer |
| Havel <-- emp |
| Kam's Sporting Goods <-- customer |
| Kuhn's Sports <-- customer |
| Maduro <-- emp |

```



```

| Magee <-- emp |
| Markarian <-- emp |
| Menchu <-- emp |
| Muench Sports <-- customer |
| Nagayama <-- emp |
| Newman <-- emp |
| Ngao <-- emp |
| Nguyen <-- emp |
| Nozaki <-- emp |
| OJ Atheletics <-- customer |
| Ojibway Retail <-- customer |
| Patel <-- emp |
| Patel <-- emp |
| Quick-To-See <-- emp |
| Ropeburn <-- emp |
| Schwartz <-- emp |
| Sedeghi <-- emp |
| Smith <-- emp |
| Sporta Russia <-- customer |
| Sportique <-- customer |
| Sweet Rock Sports <-- customer |
| Unisports <-- customer |
| Urguhart <-- emp |
| Velasquez <-- emp |
| Womansport <-- customer |
+-----+
40 rows in set (0.01 sec)

```

### Zadanie 13

Zmodyfikować zapytanie z poprzedniego ćwiczenia, tak aby najpierw pojawiły się nazwiska pracowników a potem nazwy klientów. Dane powinny być posortowane od A do Z (oddzielnie pracownicy i oddzielnie klienci). Należy użyć pewnego triku, polegającego na umiejętnym użyciu tzw. pseudokolumny i następnie posortowaniu wyniku podług niej. Jako wynik powinniśmy otrzymać:

```

+---+-----+
| 1 | Pracownicy i klienci |
+---+-----+
| 1 | Biri <-- emp |
| 1 | Catchpole <-- emp |
| 1 | Chang <-- emp |
| 1 | Dancs <-- emp |
| 1 | Dumas <-- emp |
| 1 | Giljum <-- emp |
| 1 | Havel <-- emp |
| 1 | Maduro <-- emp |
| 1 | Magee <-- emp |
| 1 | Markarian <-- emp |
| 1 | Menchu <-- emp |
| 1 | Nagayama <-- emp |
| 1 | Newman <-- emp |
| 1 | Ngao <-- emp |
| 1 | Nguyen <-- emp |
| 1 | Nozaki <-- emp |
| 1 | Patel <-- emp |
| 1 | Patel <-- emp |

```

```
| 1 | Quick-To-See <-- emp |
| 1 | Ropeburn <-- emp |
| 1 | Schwartz <-- emp |
| 1 | Sedeghi <-- emp |
| 1 | Smith <-- emp |
| 1 | Urguhart <-- emp |
| 1 | Velasquez <-- emp |
| 2 | Beisbol Si! <-- customer |
| 2 | Big John's Sports Emporium <-- customer |
| 2 | Delhi Sports <-- customer |
| 2 | Futbol Sonora <-- customer |
| 2 | Hamada Sport <-- customer |
| 2 | Kam's Sporting Goods <-- customer |
| 2 | Kuhn's Sports <-- customer |
| 2 | Muench Sports <-- customer |
| 2 | OJ Atheletics <-- customer |
| 2 | Ojibway Retail <-- customer |
| 2 | Sporta Russia <-- customer |
| 2 | Sportique <-- customer |
| 2 | Sweet Rock Sports <-- customer |
| 2 | Unisports <-- customer |
| 2 | Womansport <-- customer |
+---+-----+
40 rows in set (0.00 sec)
```