

# Cleverlance

## SQL

Dotazy nad více tabulkami  
- Joiny

# Spojení tabulek

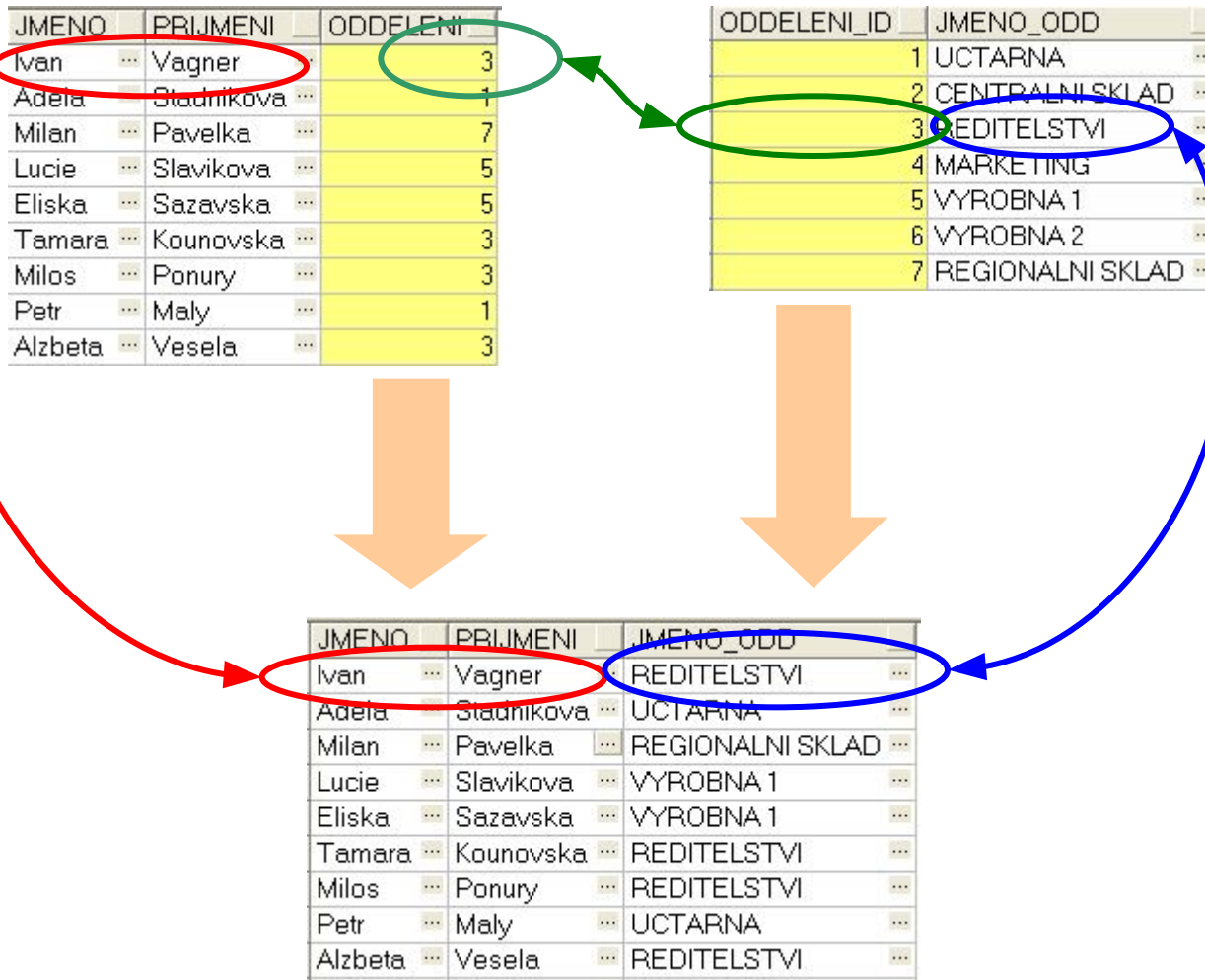
JMENO	PRIJMENI	ODDELENI
Ivan	Vagner	3
Adela	Stadnikova	1
Milan	Pavelka	7
Lucie	Slavikova	5
Eliska	Sazavska	5
Tamara	Kounovska	3
Milos	Ponury	3
Petr	Maly	1
Alzbeta	Vesela	3

ODDELENI_ID	JMENO_ODD
1	UCTARNA
2	CENTRALNI SKLAD
3	REDITELSTVI
4	MARKETING
5	VYROBNA 1
6	VYROBNA 2
7	REGIONALNI SKLAD



JMENO	PRIJMENI	JMENO_ODD
Ivan	Vagner	REDITELSTVI
Adela	Stadnikova	UCTARNA
Milan	Pavelka	REGIONALNI SKLAD
Lucie	Slavikova	VYROBNA 1
Eliska	Sazavska	VYROBNA 1
Tamara	Kounovska	REDITELSTVI
Milos	Ponury	REDITELSTVI
Petr	Maly	UCTARNA
Alzbeta	Vesela	REDITELSTVI

# Spojení tabulek



# Spojování tabulek



JMENO	PRIJMENI	ODDELENI	ODDELENI_ID	JMENO_ODD
Ivan	Vagner	3	3	REDITELSTVI
Adela	Stadnikova	1	1	UCTARNA
Milan	Pavelka	7	7	REGIONALNI SKLAD
Lucie	Slavikova	5	5	VYROBNA 1
Eliska	Sazavska	5	5	VYROBNA 1
Tamara	Kounovska	3	3	REDITELSTVI
Milos	Ponury	3	3	REDITELSTVI
Petr	Maly	1	1	UCTARNA
Alzbeta	Vesela	3	3	REDITELSTVI

# Spojení / Join

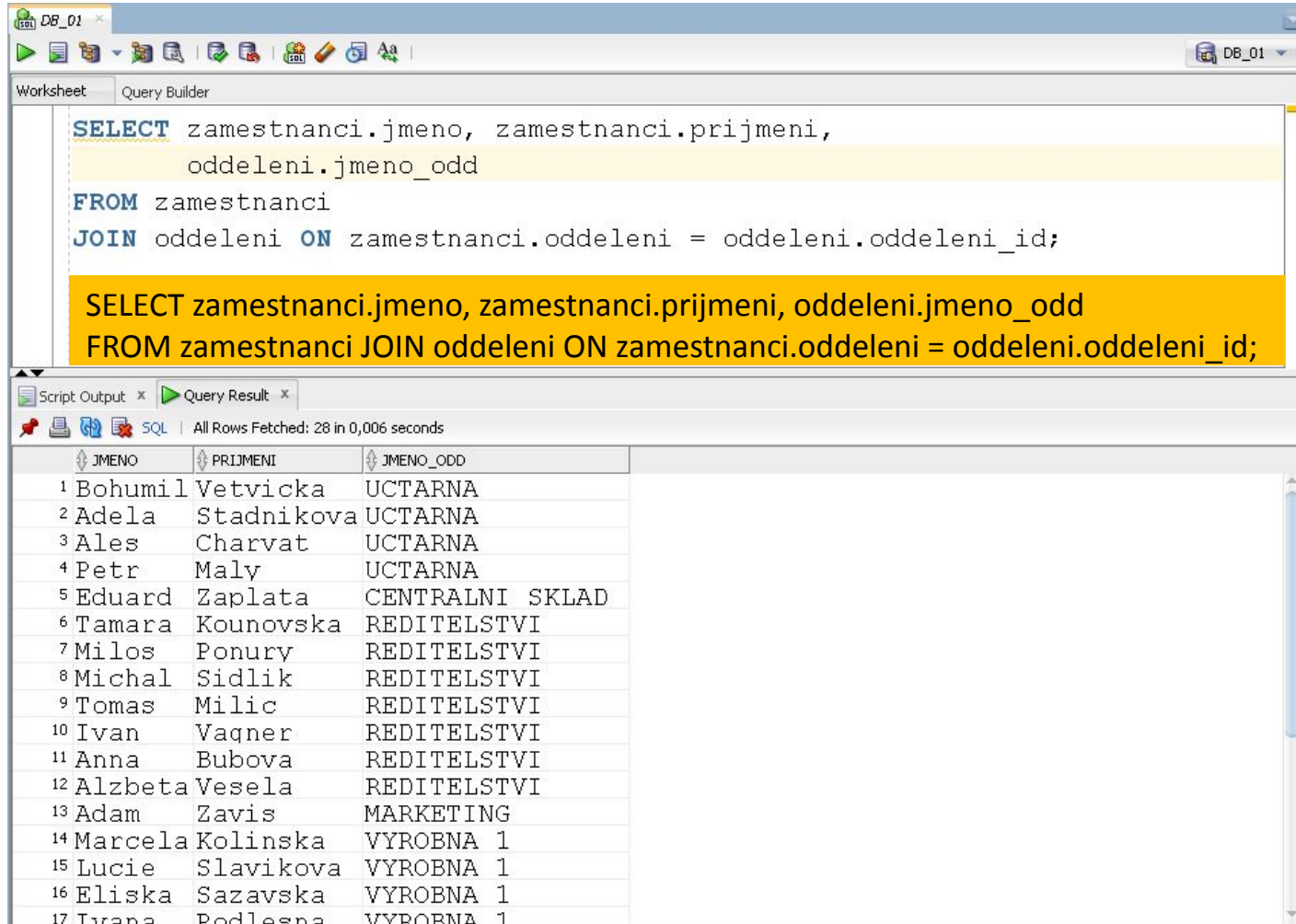
- Spojení tabulek umožňuje v jednom dotazu získat data z více tabulek
- Syntaxe

```
SELECT table1.column, table2.column  
FROM table1  
JOIN table2 ON table1.column1 = table2.column2;
```

- Alternativní Oracle syntaxe
- Podmínka spojení je v klauzuli WHERE

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

# Spojení tabulek



The screenshot shows a database query editor window titled "DB\_01". The main area displays a SQL query in the "Query Builder" tab. The query is as follows:

```
SELECT zamestnanci.jmeno, zamestnanci.prijmeni,  
       oddeleni.jmeno_odd  
FROM zamestnanci  
JOIN oddeleni ON zamestnanci.oddeleni = oddeleni.oddeleni_id;
```

Below the query, a yellow highlight shows the same query with the JOIN clause explicitly written:

```
SELECT zamestnanci.jmeno, zamestnanci.prijmeni, oddeleni.jmeno_odd  
FROM zamestnanci JOIN oddeleni ON zamestnanci.oddeleni = oddeleni.oddeleni_id;
```

The "Query Result" tab shows the output of the query, displaying 17 rows of data. The columns are labeled "JMENO", "PRIJMENI", and "JMENO\_ODD".

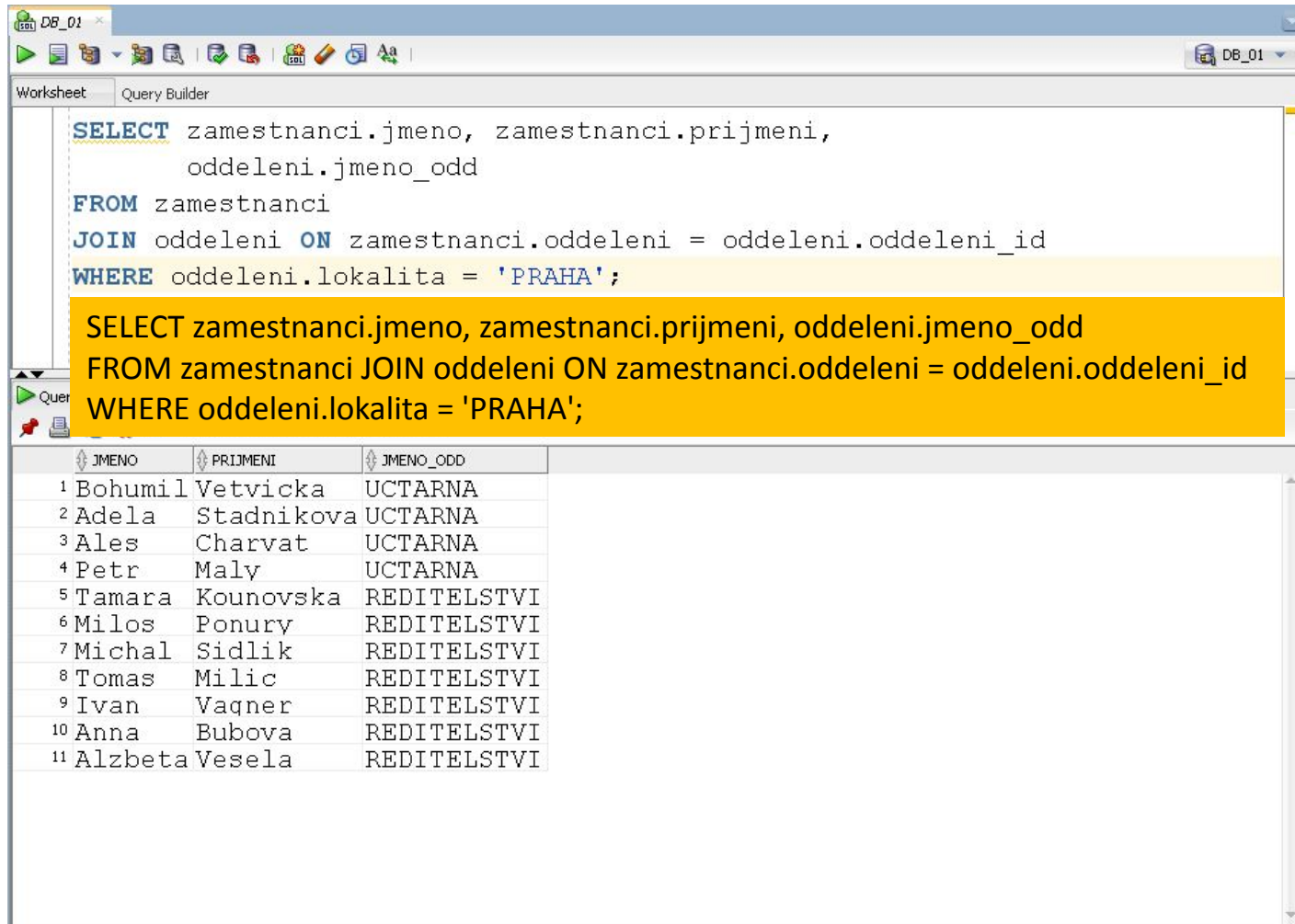
	JMENO	PRIJMENI	JMENO_ODD
1	Bohumil	Vetvicka	UCTARNA
2	Adela	Stadnikova	UCTARNA
3	Ales	Charvat	UCTARNA
4	Petr	Maly	UCTARNA
5	Eduard	Zaplata	CENTRALNI SKLAD
6	Tamara	Kounovska	REDITELSTVI
7	Milos	Ponury	REDITELSTVI
8	Michal	Sidlik	REDITELSTVI
9	Tomas	Milic	REDITELSTVI
10	Ivan	Vagner	REDITELSTVI
11	Anna	Bubova	REDITELSTVI
12	Alzbeta	Vesela	REDITELSTVI
13	Adam	Zavis	MARKETING
14	Marcela	Kolinska	VYROBNA 1
15	Lucie	Slavikova	VYROBNA 1
16	Eliska	Sazavska	VYROBNA 1
17	Ivana	Podlesna	VYROBNA 1

# Příklad

- Vypište seznam zaměstnanců a jmen oddělení, které sídlí v Praze

	JMENO	PRIJMENI	JMENO_ODD
1	Ivan	Vagner	REDITELSTVI
2	Adela	Stadnikova	UCTARNA
3	Tamara	Kounovska	REDITELSTVI
4	Milos	Ponury	REDITELSTVI
5	Petr	Maly	UCTARNA
6	Alzbeta	Vesela	REDITELSTVI
7	Anna	Bubova	REDITELSTVI
8	Ales	Charvat	UCTARNA
9	Bohumil	Vetvicka	UCTARNA
10	Tomas	Milic	REDITELSTVI

# Příklad - řešení



The screenshot shows a database query builder window titled "DB\_01". The "Query Builder" tab is active, displaying a SQL query. The query is as follows:

```
SELECT zamestnanci.jmeno, zamestnanci.prijmeni,  
       oddeleni.jmeno_odd  
FROM zamestnanci  
JOIN oddeleni ON zamestnanci.oddeleni = oddeleni.oddeleni_id  
WHERE oddeleni.lokalita = 'PRAHA';
```

The query results are displayed in a table below the query editor. The table has three columns: JMENO, PRIJMENI, and JMENO\_ODD. The results are as follows:

	JMENO	PRIJMENI	JMENO_ODD
1	Bohumil	Vetvicka	UCTARNA
2	Adela	Stadnikova	UCTARNA
3	Ales	Charvat	UCTARNA
4	Petr	Maly	UCTARNA
5	Tamara	Kounovska	REDITELSTVI
6	Milos	Ponury	REDITELSTVI
7	Michal	Sidlik	REDITELSTVI
8	Tomas	Milic	REDITELSTVI
9	Ivan	Vagner	REDITELSTVI
10	Anna	Bubova	REDITELSTVI
11	Alzbeta	Vesela	REDITELSTVI



# Aliasy pro jména tabulek

```
SELECT zamestnanci.jmeno, zamestnanci.prijmeni,  
oddeleni.jmeno_odd  
FROM zamestnanci  
JOIN oddeleni ON zamestnanci.oddeleni =  
oddeleni.oddeleni_id;
```

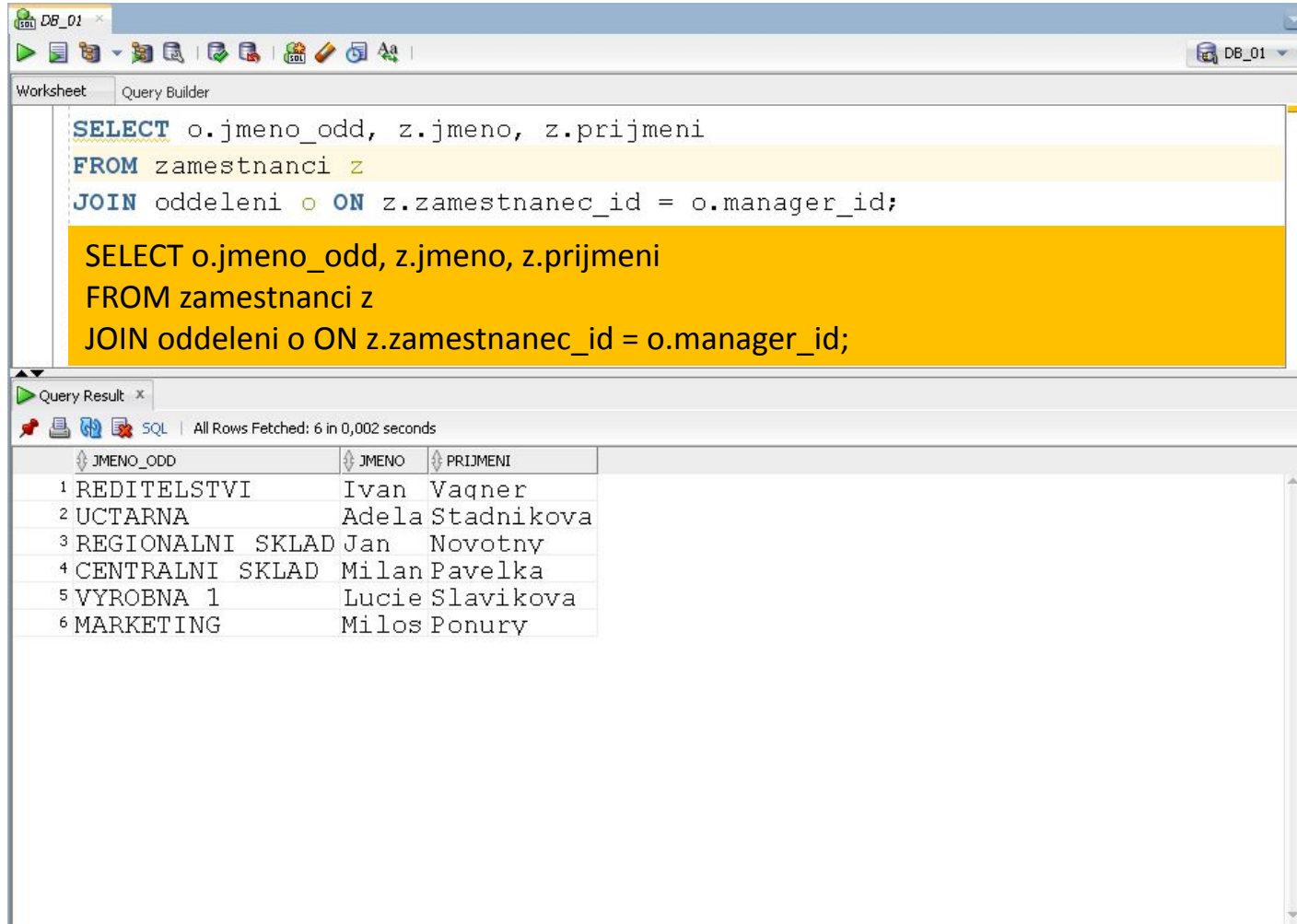
```
SELECT zam.jmeno, zam.prijmeni,  
odd.jmeno_odd  
FROM zamestnanci zam  
JOIN oddeleni odd ON zam.oddeleni =  
odd.oddeleni_id;
```

# Příklad

- Vypište seznam oddělení a jména jejich managerů.(Podle manager\_id v tabulce oddeleni)

	JMENO_ODD	JMENO	PRIJMENI
1	REDITELSTVI	Ivan	Vagner
2	UCTARNA	Adela	Stadnikova
3	REGIONALNI SKLAD	Milan	Pavelka
4	VYROBNA 1	Lucie	Slavikova
5	VYROBNA 2	Eliska	Sazavska
6	MARKETING	Milos	Ponury
7	CENTRALNI SKLAD	Anna	Bubova

# Příklad řešení



The screenshot shows a database query builder interface. The top window, titled "DB\_01", contains a "Query Builder" tab with the following SQL query:

```
SELECT o.jmeno_odd, z.jmeno, z.prijmeni  
FROM zamestnanci z  
JOIN oddeleni o ON z.zamestnanec_id = o.manager_id;
```

The query is highlighted in yellow. Below the query, the results are displayed in a table with the following columns: JMENO\_ODD, JMENO, and PRIJMENI. The results are as follows:

	JMENO_ODD	JMENO	PRIJMENI
1	REDITELSTVI	Ivan	Vagner
2	UCTARNA	Adela	Stadnikova
3	REGIONALNI SKLAD	Jan	Novotny
4	CENTRALNI SKLAD	Milan	Pavelka
5	VYROBNA 1	Lucie	Slavikova
6	MARKETING	Milos	Ponury

The bottom window, titled "Query Result", shows the results of the query. It indicates that all rows were fetched in 0,002 seconds. The results are displayed in a table with the following columns: JMENO\_ODD, JMENO, and PRIJMENI. The results are as follows:

# Kartézský součin

- Pokud podmínka spojení chybí nebo není platná, pak je výsledkem kartézský součin, který obsahuje všechny kombinace řádků
- Kartézský součin tabulky s 20 záznamy s tabulkou s 30 záznamy obsahuje 600 záznamů

# Kartézský součin

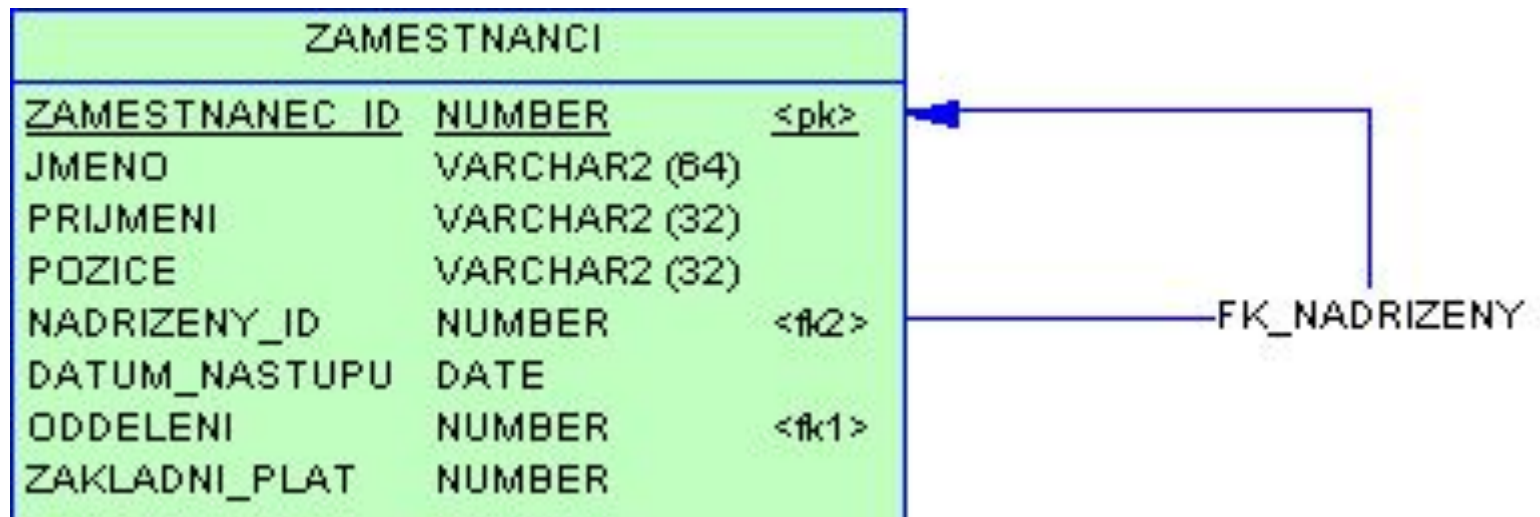
The screenshot shows a database query tool interface. The top window, titled 'DB\_01', contains a 'Query Builder' tab with the following SQL query:

```
SELECT COUNT (*)  
FROM zamestnanci, oddeleni;
```

Below the query editor, the 'Query Result' tab displays the execution status: 'All Rows Fetched: 1 in 0,007 seconds'. The result is shown in a table with one row and one column:

COUNT(*)
196

# Spojení tabulky se stejnou tabulkou

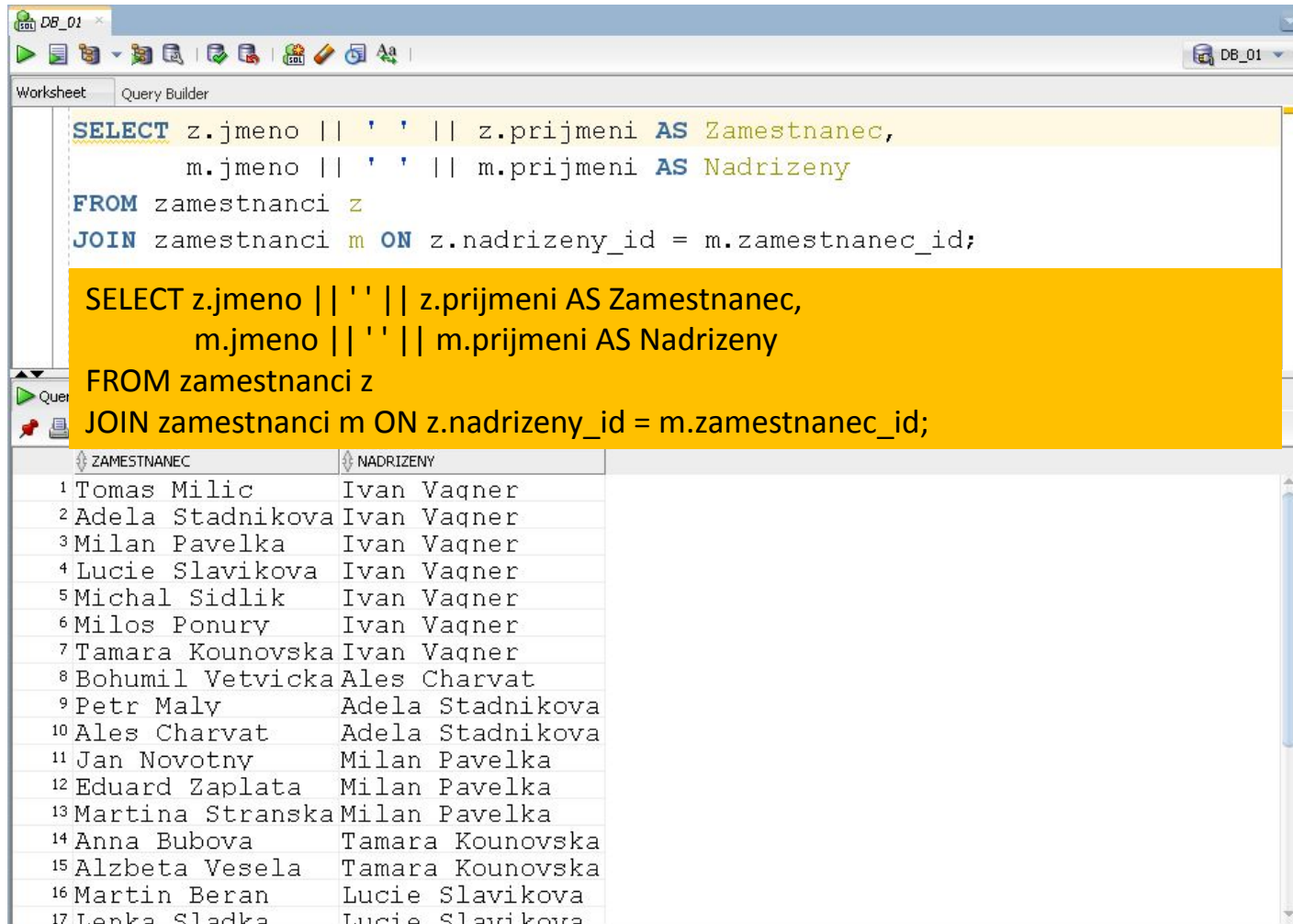


# Příklad

- Vypište jména zaměstnanců a jména jejich přímých nadřízených

	ZAMESTNANEC		NADRIZENY	
1	Michal Sidik	...	Ivan Vagner	...
2	Tomas Milic	...	Ivan Vagner	...
3	Milos Ponury	...	Ivan Vagner	...
4	Tamara Kounovska	...	Ivan Vagner	...
5	Lucie Slavikova	...	Ivan Vagner	...
6	Milan Pavelka	...	Ivan Vagner	...
7	Adela Stadnikova	...	Ivan Vagner	...
8	Ales Charvat	...	Adela Stadnikova	...
9	Petr Maly	...	Adela Stadnikova	...
10	Eduard Zaplata	...	Milan Pavelka	...

# Příklad - řešení



The screenshot shows a database query editor window titled "DB\_01". The "Query Builder" tab is active, displaying a SQL query. The query is highlighted in yellow and reads:

```
SELECT z.jmeno || ' ' || z.prijmeni AS Zamestnanec,  
       m.jmeno || ' ' || m.prijmeni AS Nadrizeny  
FROM zamestnanci z  
JOIN zamestnanci m ON z.nadrizeny_id = m.zamestnanec_id;
```

Below the query, the results are displayed in a table with two columns: "ZAMESTNANEC" and "NADRIZENY". The results are as follows:

	ZAMESTNANEC	NADRIZENY
1	Tomas Milic	Ivan Vagner
2	Adela Stadnikova	Ivan Vagner
3	Milan Pavelka	Ivan Vagner
4	Lucie Slavikova	Ivan Vagner
5	Michal Sidlik	Ivan Vagner
6	Milos Ponury	Ivan Vagner
7	Tamara Kounovska	Ivan Vagner
8	Bohumil Vetvicka	Ales Charvat
9	Petr Maly	Adela Stadnikova
10	Ales Charvat	Adela Stadnikova
11	Jan Novotny	Milan Pavelka
12	Eduard Zaplata	Milan Pavelka
13	Martina Stranska	Milan Pavelka
14	Anna Bubova	Tamara Kounovska
15	Alzbeta Vesela	Tamara Kounovska
16	Martin Beran	Lucie Slavikova
17	Lenka Sladka	Lucie Slavikova

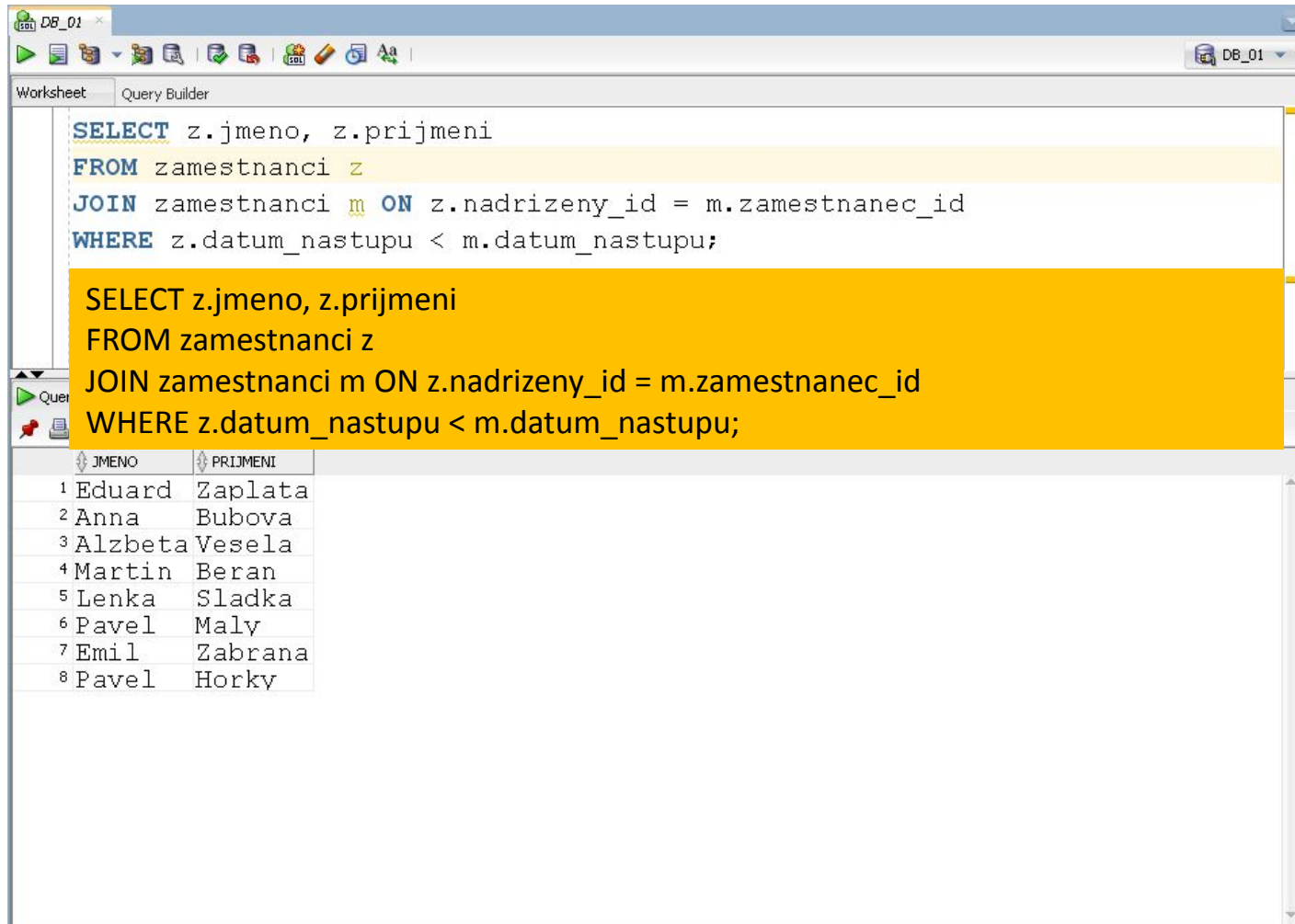


# Příklad

- Kteří zaměstnanci nastoupili dříve než jejich nadřízení?

	JMENO	PRIJMENI
1	Eduard	Zaplata
2	Martin	Beran
3	Lenka	Sladka
4	Pavel	Horky
5	Anna	Bubova
6	Alzbeta	Vesela
7	Emil	Zabrana
8	Pavel	Maly

# Příklad - řešení



The screenshot shows a database query builder window titled "DB\_01". The interface includes a toolbar with various icons and a main area divided into "Worksheet" and "Query Builder" tabs. The "Query Builder" tab is active, displaying a SQL query. Below the query, there is a yellow highlighted area containing a simplified version of the query. At the bottom, a table displays the results of the query, with columns for "JMENO" and "PRIJMENI".

```
SELECT z.jmeno, z.prijmeni
FROM zamestnanci z
JOIN zamestnanci m ON z.nadrizeny_id = m.zamestnanec_id
WHERE z.datum_nastupu < m.datum_nastupu;
```

```
SELECT z.jmeno, z.prijmeni
FROM zamestnanci z
JOIN zamestnanci m ON z.nadrizeny_id = m.zamestnanec_id
WHERE z.datum_nastupu < m.datum_nastupu;
```

	JMENO	PRIJMENI
1	Eduard	Zaplata
2	Anna	Bubova
3	Alzbeta	Vesela
4	Martin	Beran
5	Lenka	Sladka
6	Pavel	Maly
7	Emil	Zabrana
8	Pavel	Horky

# Spojení více tabulek

- Spojení tří tabulek

```
SELECT table1.column, table2.column, table3.column  
FROM table1  
JOIN table2 ON table1.column1 = table2.column2  
JOIN table3 ON table2.column3 = table3.column4;
```

- Alternativní Oracle syntaxe

```
SELECT table1.column, table2.column, table3.column  
FROM table1, table2, table3  
WHERE table1.column1 = table2.column2 AND  
table2.column3 = table3.column4;
```

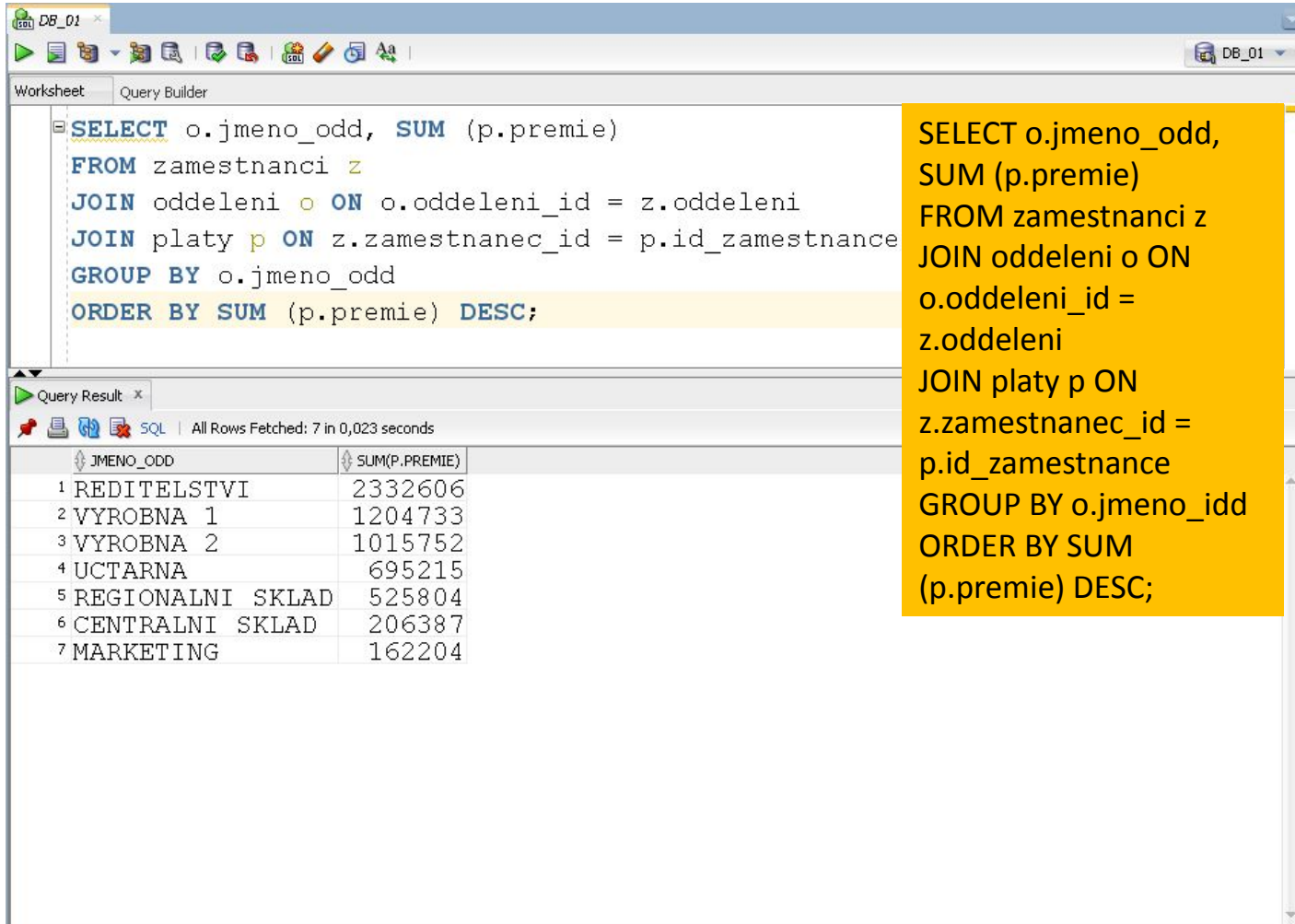
- Pro spojení n tabulek potřebujeme n-1 spojovacích podmínek

# Příklad

- Napište seznam oddělení a celkových částek vyplacených na odměnách. Výsledek seřadíte podle odměn od nejvyšších po nejnižší.

	JMENO_ODD		SUM(P.PREMIE)
1	REDITELSTVI	...	328888
2	REGIONALNI SKLAD	...	38000
3	VYROBNA 1	...	32498
4	UCTARNA	...	22998
5	CENTRALNI SKLAD	...	14499
6	VYROBNA 2	...	11499

# Příklad - řešení



The screenshot shows a database query builder interface. The top window is titled "DB\_01" and contains a "Query Builder" tab. The SQL query is as follows:

```
SELECT o.jmeno_odd, SUM (p.premie)
FROM zamestnanci z
JOIN oddeleni o ON o.oddeleni_id = z.oddeleni
JOIN platy p ON z.zamestnanec_id = p.id_zamestnance
GROUP BY o.jmeno_odd
ORDER BY SUM (p.premie) DESC;
```

The bottom window is titled "Query Result" and shows the results of the query. It indicates that all rows were fetched in 0,023 seconds. The results are displayed in a table with two columns: "JMENO\_ODD" and "SUM(P.PREMIE)".

JMENO_ODD	SUM(P.PREMIE)
1 REDITELSTVI	2332606
2 VYROBNA 1	1204733
3 VYROBNA 2	1015752
4 UCTARNA	695215
5 REGIONALNI SKLAD	525804
6 CENTRALNI SKLAD	206387
7 MARKETING	162204

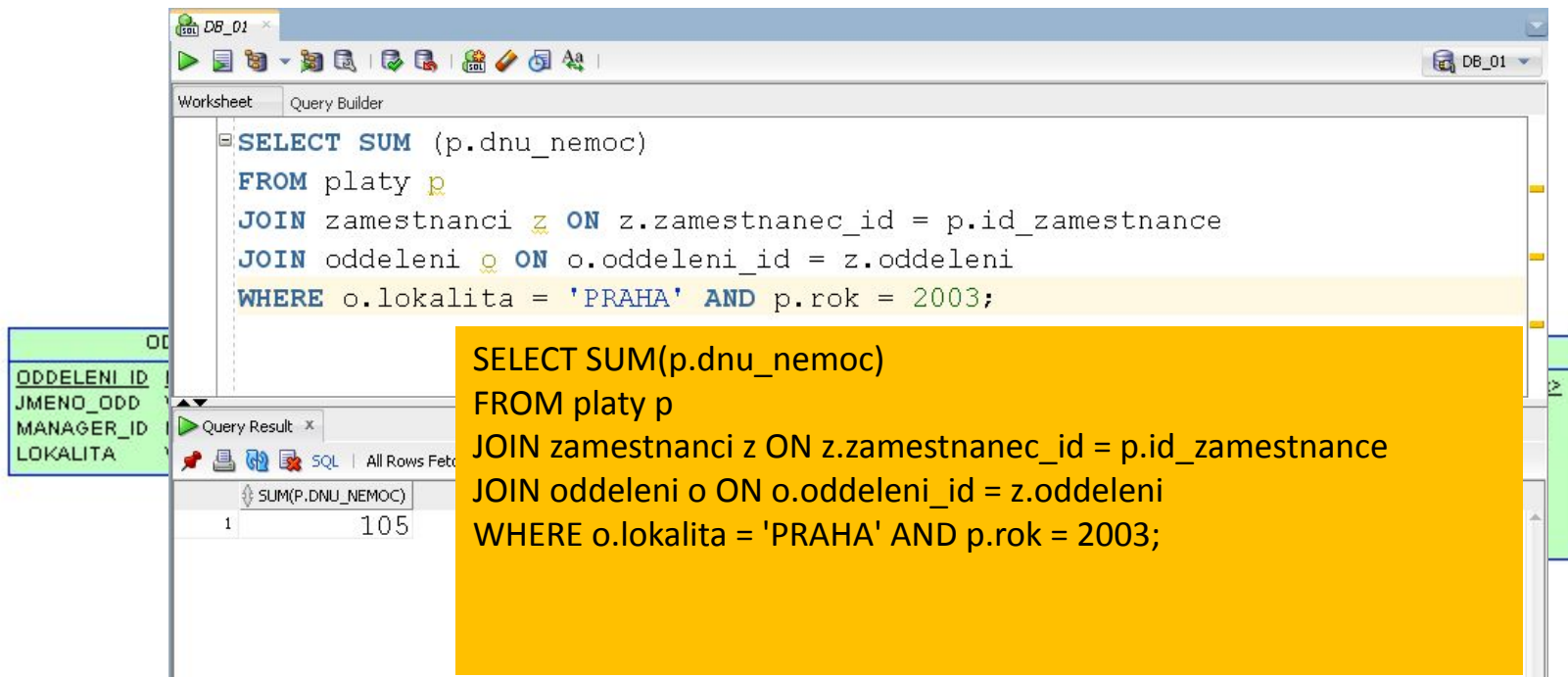
A yellow callout box on the right side of the screenshot contains the following SQL query:

```
SELECT o.jmeno_odd,
SUM (p.premie)
FROM zamestnanci z
JOIN oddeleni o ON
o.oddeleni_id =
z.oddeleni
JOIN platy p ON
z.zamestnanec_id =
p.id_zamestnance
GROUP BY o.jmeno_idd
ORDER BY SUM
(p.premie) DESC;
```

# Spojení více tabulek

- Příklad:

Jak dlouho byli celkem v roce 2003 nemocní zaměstnanci, kteří pracují v Praze?



The screenshot shows a database query builder interface. The main window displays the following SQL query:

```
SELECT SUM (p.dnu_nemoc)
FROM platy p
JOIN zamestnanci z ON z.zamestnanec_id = p.id_zamestnance
JOIN oddeleni o ON o.oddeleni_id = z.oddeleni
WHERE o.lokalita = 'PRAHA' AND p.rok = 2003;
```

Below the query, a 'Query Result' window shows the following result:

SUM(P.DNU_NEMOC)
105

A yellow box highlights the SQL query and the result table. The result table has one row with the value 105.

# Neekvivalentní vazba

- Neekvivalentní spojení je například vazba mezi tabulkami ZAMESTNANCI a PLAT\_TRIDY. Znamená to, že žádný sloupec tabulky ZAMESTNANCI neodpovídá přímo žádnému sloupci tabulky PLAT\_TRIDY.

PLAT_TRIDY		
<u>TRIDA</u>	NUMBER	<pk>
MIN_PLAT	NUMBER	
MAX_PLAT	NUMBER	

TRIDA	MIN_PLAT	MAX_PLAT
1	0	5000
2	5000	10000
4	15000	20000
5	20000	25000
6	25000	30000
7	30000	35000
8	35000	40000
9	40000	100000
3	10000	15000

ZAMESTNANCI		
<u>ZAMESTNANEC_ID</u>	NUMBER	<pk>
JMENO	VARCHAR2 (64)	
PRIJMENI	VARCHAR2 (32)	
POZICE	VARCHAR2 (32)	
NADRIZENY_ID	NUMBER	<fk2>
DATUM_NASTUPU	DATE	
ODDELENI	NUMBER	<fk1>
ZAKLADNI_PLAT	NUMBER	

# Neekvivalentní vazba

- Vztah mezi tabulkami je takový, že hodnota sloupce ZAKLADNI\_PLAT v tabulce ZAMESTNANCI musí být mezi hodnotami MIN\_PLAT a MAX\_PLAT v tabulce PLAT\_TRIDY. Takový vztah vzniká použitím jiného operátoru než rovnítka (=).

```
SELECT z.prijmeni, z.zakladni_plat, p.trida
FROM zamestnanci z, plat_tridy p
WHERE z.zakladni_plat
      BETWEEN p.min_plat AND p.max_plat;
```



# OUTER JOIN

The screenshot displays a database query builder interface. The query editor shows the following SQL query:

```
SELECT z.jmeno, z.prijmeni, sum(p.premie)
FROM zamestnanci z
LEFT JOIN platy p ON z.zamestnanec_id = p.id_zamestnance
GROUP BY z.jmeno, z.prijmeni
ORDER BY z.jmeno, z.prijmeni;
```

The query results are displayed in a table with the following columns: JMENO, PRIJMENI, and SUM(P.PREMIE). The results are as follows:

	JMENO	PRIJMENI	SUM(P.PREMIE)
1	Adam	Zavis	(null)
2	Adele	Stadnikova	5219
3	Ales	Charvat	(null)
4	Alzbeta	Vesela	(null)
5	Anna	Bubova	2266
6	Bohumil	Vetvicka	1927
7	Bozena	Novakova	1832
8	Eduard	Zaplata	3776
9	Eliska	Sazavska	(null)
10	Emil	Zabrana	(null)
11	Ivan	Doubrava	(null)
12	Ivan	Vagner	183020
13	Ivana	Podlesna	12191
14	Jan	Novotny	2518
15	Lenka	Sladka	6955
16	Lucie	Slavikova	7117

# OUTER JOIN – Vnější vazba

- Spojení dvou tabulek obsahuje pouze takové záznamy, pro které je splněna propojovací tabulka
- Vnější vazba umožňuje do výsledku dotazu zařadit i takové záznamy, proto které neexistuje ekvivalentní záznam v jedné z tabulek

# OUTER JOIN – Vnější vazba

- Syntax (SQL 99):
  - **LEFT** nebo **RIGHT** označuje stranu (tabulku), která má veškeré informace a tyto budou zobrazeny – nemusí být zobrazeny všechny informace z připojované tabulky (informace mohou chybět)

```
SELECT table1.column, table2.column  
FROM table1  
LEFT JOIN table2 ON table1.column = table2.column;
```

# OUTER JOIN – Vnější vazba

- Syntax (Oracle):
  - (+) je na té straně podmínky, kde předpokládáme nedostatek informací

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column [ (+) ] = table2.column[(+)];
```

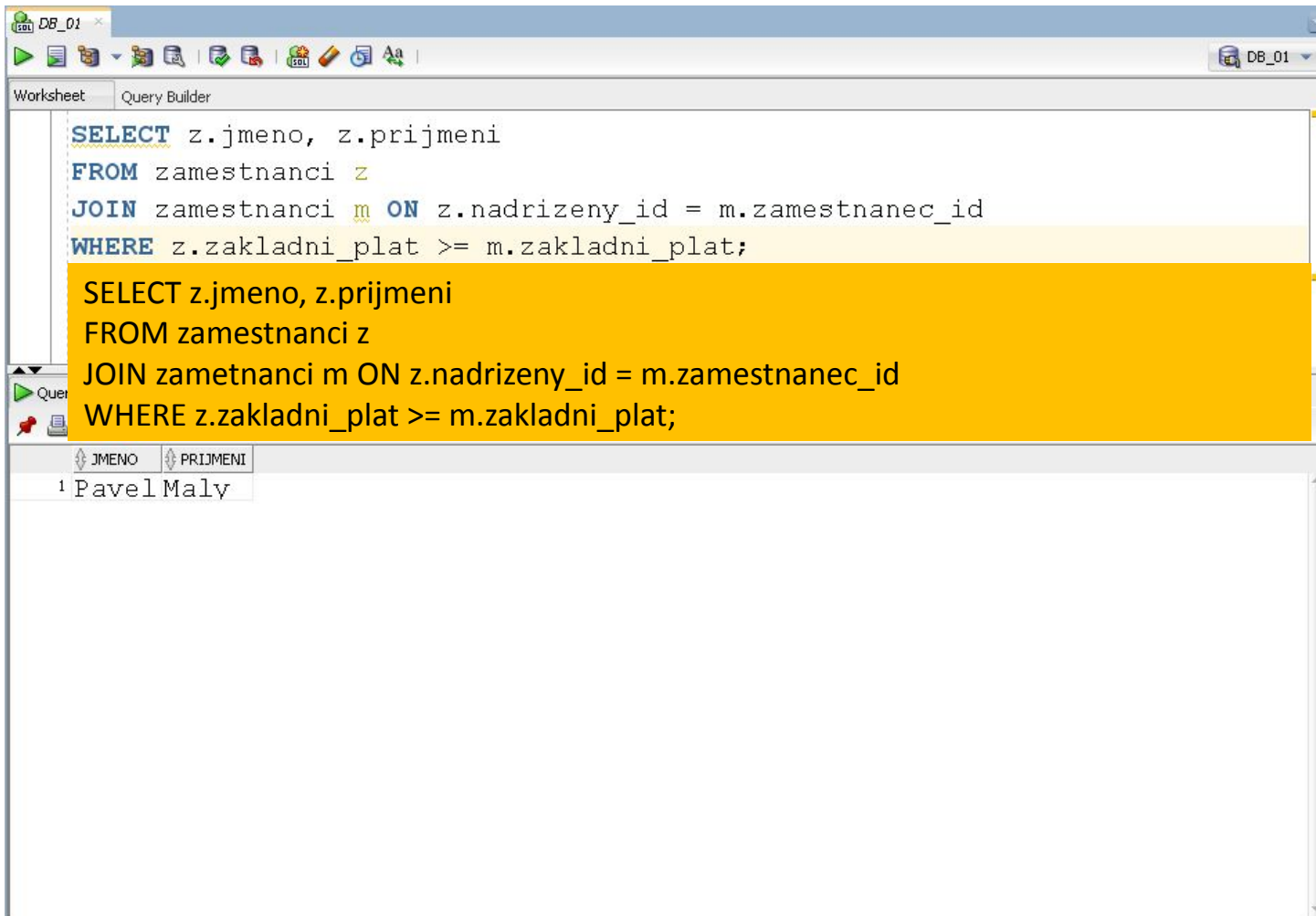
# Spojování tabulek pomocí SQL:

## Syntax 99

```
SELECT table1.column, table2.column
FROM table1
[CROSS JOIN table2] |
[NATURAL JOIN table2] |
[JOIN table2 USING (column name)] |
[JOIN table2
 ON(table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
 ON (table1.column_name = table2.column_name)];
```

# Cvičení

1. Kteří zaměstnanci mají vyšší nebo stejný plat jako jejich nadřízení?



The screenshot shows a database query builder window titled "DB\_01". The main area contains a SQL query:

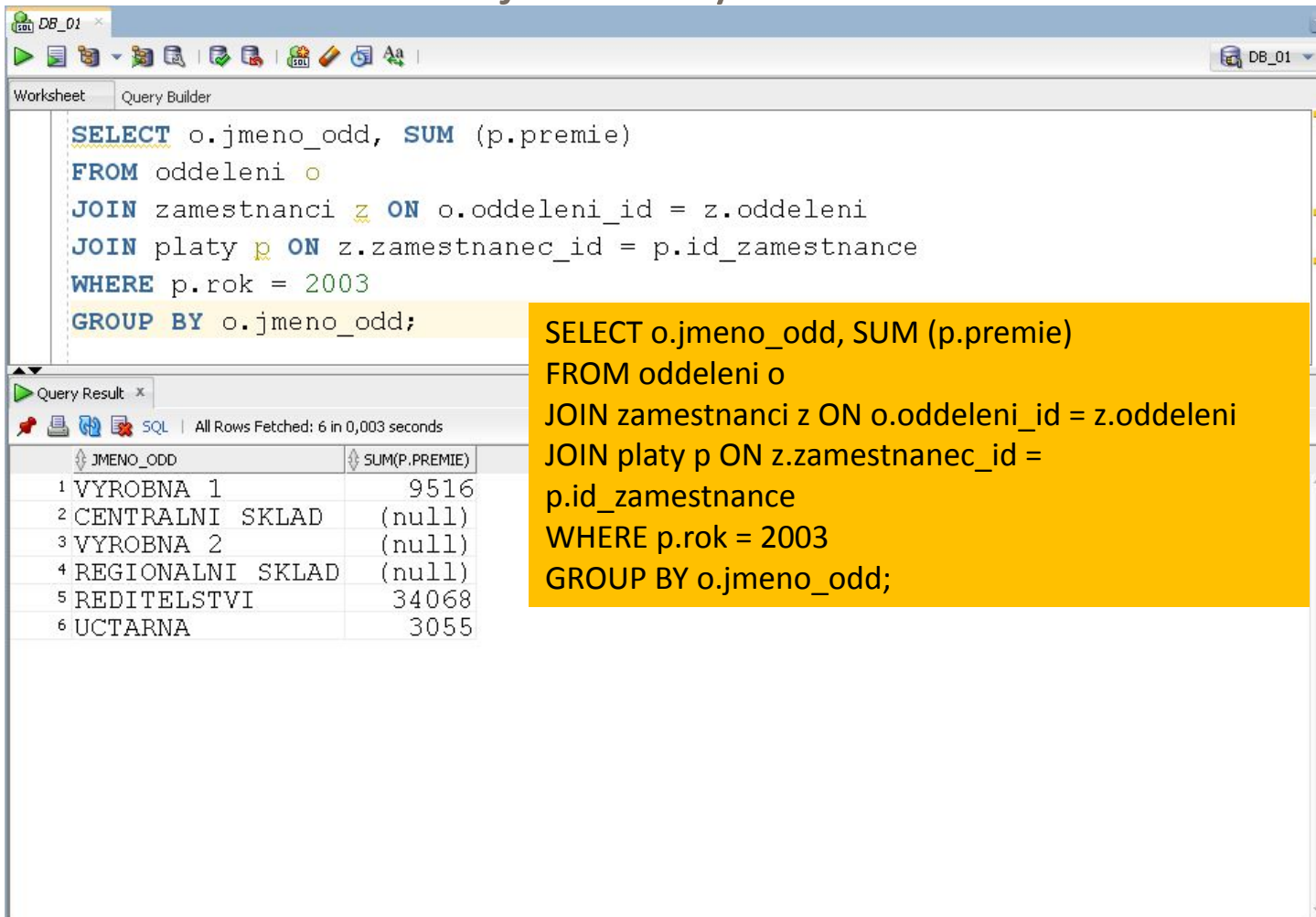
```
SELECT z.jmeno, z.prijmeni  
FROM zamestnanci z  
JOIN zamestnanci m ON z.nadrizeny_id = m.zamestnanec_id  
WHERE z.zakladni_plat >= m.zakladni_plat;
```

The query is highlighted in yellow. Below the query, the results are displayed in a table with columns "JMENO" and "PRIJMENI". The first result is "1 Pavel Maly".

	JMENO	PRIJMENI
1	Pavel	Maly

# Cvičení

2. Kolik bylo v roce 2003 vyplaceno na odměnách zaměstnancům jednotlivých oddělení



The screenshot shows a database query builder interface. The top part displays a SQL query in the Query Builder window. The query is as follows:

```
SELECT o.jmeno_odd, SUM (p.premie)
FROM oddeleni o
JOIN zamestnanci z ON o.oddeleni_id = z.oddeleni
JOIN platy p ON z.zamestnanec_id = p.id_zamestnance
WHERE p.rok = 2003
GROUP BY o.jmeno_odd;
```

The bottom part of the screenshot shows the Query Result window, which displays the results of the query in a table format. The table has two columns: JMENO\_ODD and SUM(P.PREMIE). The results are as follows:

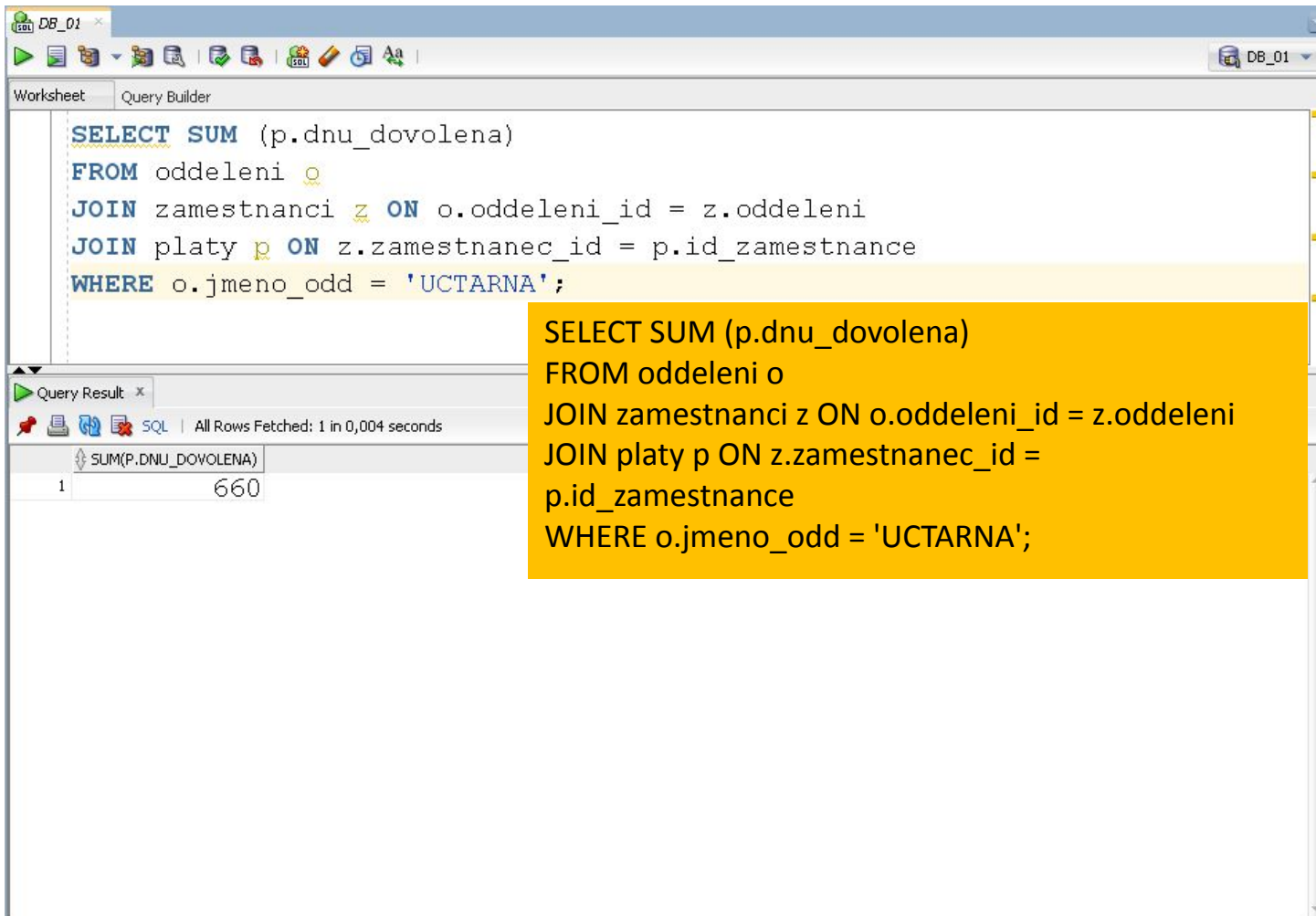
JMENO_ODD	SUM(P.PREMIE)
1 VYROBNA 1	9516
2 CENTRALNI SKLAD	(null)
3 VYROBNA 2	(null)
4 REGIONALNI SKLAD	(null)
5 REDITELSTVI	34068
6 UCTARNA	3055

A yellow highlight box is present over the SQL query in the Query Result window, containing the following text:

```
SELECT o.jmeno_odd, SUM (p.premie)
FROM oddeleni o
JOIN zamestnanci z ON o.oddeleni_id = z.oddeleni
JOIN platy p ON z.zamestnanec_id =
p.id_zamestnance
WHERE p.rok = 2003
GROUP BY o.jmeno_odd;
```

# Cvičení

3. Kolik dní dovolené si vybrali zaměstnanci Účtárny?



The screenshot shows a database query builder interface. The top window is titled "DB\_01" and contains a "Query Builder" tab. The SQL query is as follows:

```
SELECT SUM (p.dnu_dovolena)
FROM oddeleni o
JOIN zamestnanci z ON o.oddeleni_id = z.oddeleni
JOIN platy p ON z.zamestnanec_id = p.id_zamestnance
WHERE o.jmeno_odd = 'UCTARNA';
```

The bottom window is titled "Query Result" and shows the result of the query. It indicates "All Rows Fetched: 1 in 0,004 seconds". The result is displayed in a table with one row and one column:

SUM(P.DNU_DOVOLENA)
1 660

A yellow callout box highlights the SQL query text, showing a slightly different version of the query:

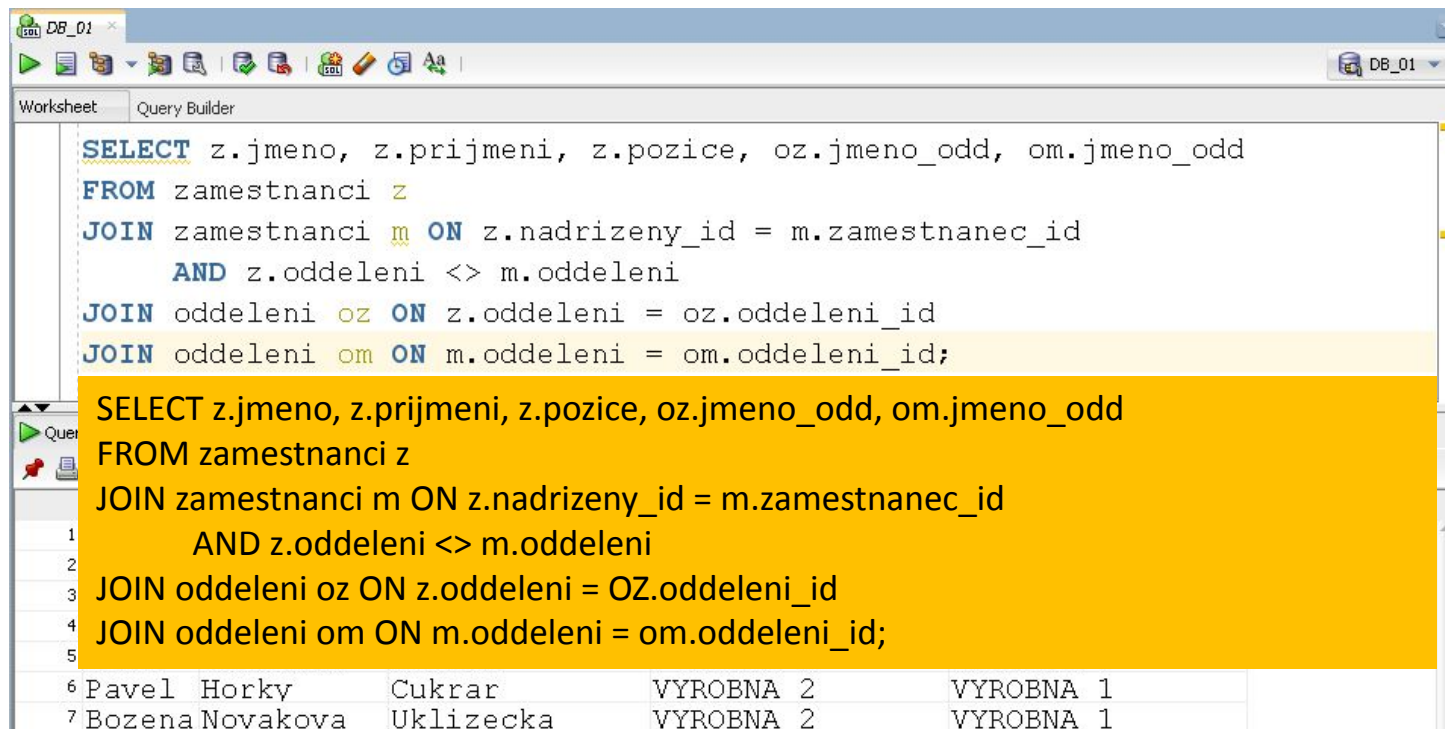
```
SELECT SUM (p.dnu_dovolena)
FROM oddeleni o
JOIN zamestnanci z ON o.oddeleni_id = z.oddeleni
JOIN platy p ON z.zamestnanec_id =
p.id_zamestnance
WHERE o.jmeno_odd = 'UCTARNA';
```





# Cvičení

5. Kdo má nadřízeného z jiného oddělení než sám pracuje? Vypište celé jméno, pozici a jméno oddělení, ze kterého pochází on i jeho nadřízený.



The screenshot shows a database query builder window titled "DB\_01". The main area displays a SQL query in the "Query Builder" tab. The query is as follows:

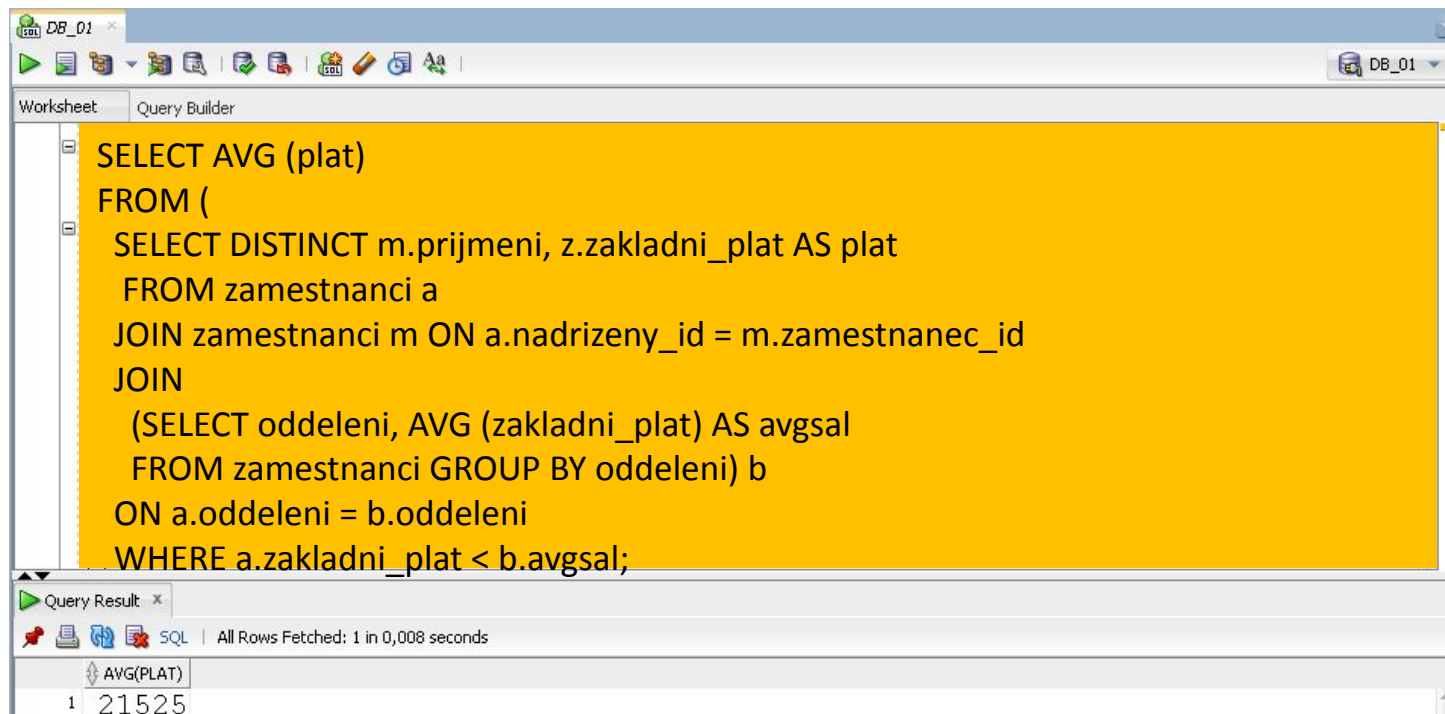
```
SELECT z.jmeno, z.prijmeni, z.pozice, oz.jmeno_odd, om.jmeno_odd
FROM zamestnanci z
JOIN zamestnanci m ON z.nadrizeny_id = m.zamestnanec_id
AND z.oddeleni <> m.oddeleni
JOIN oddeleni oz ON z.oddeleni = oz.oddeleni_id
JOIN oddeleni om ON m.oddeleni = om.oddeleni_id;
```

Below the query, a yellow highlight covers the same query text. At the bottom, a table shows the results of the query:

6	Pavel Horky	Cukrar	VYROBNA 2	VYROBNA 1
7	Bozena Novakova	Uklizecka	VYROBNA 2	VYROBNA 1

# Cvičení

6. Kolik je průměrný plat nadřízených zaměstnanců, kteří mají plat nižší než je průměrný plat na jejich oddělení. Každého nadřízeného započítejte do průměru jen jednou.



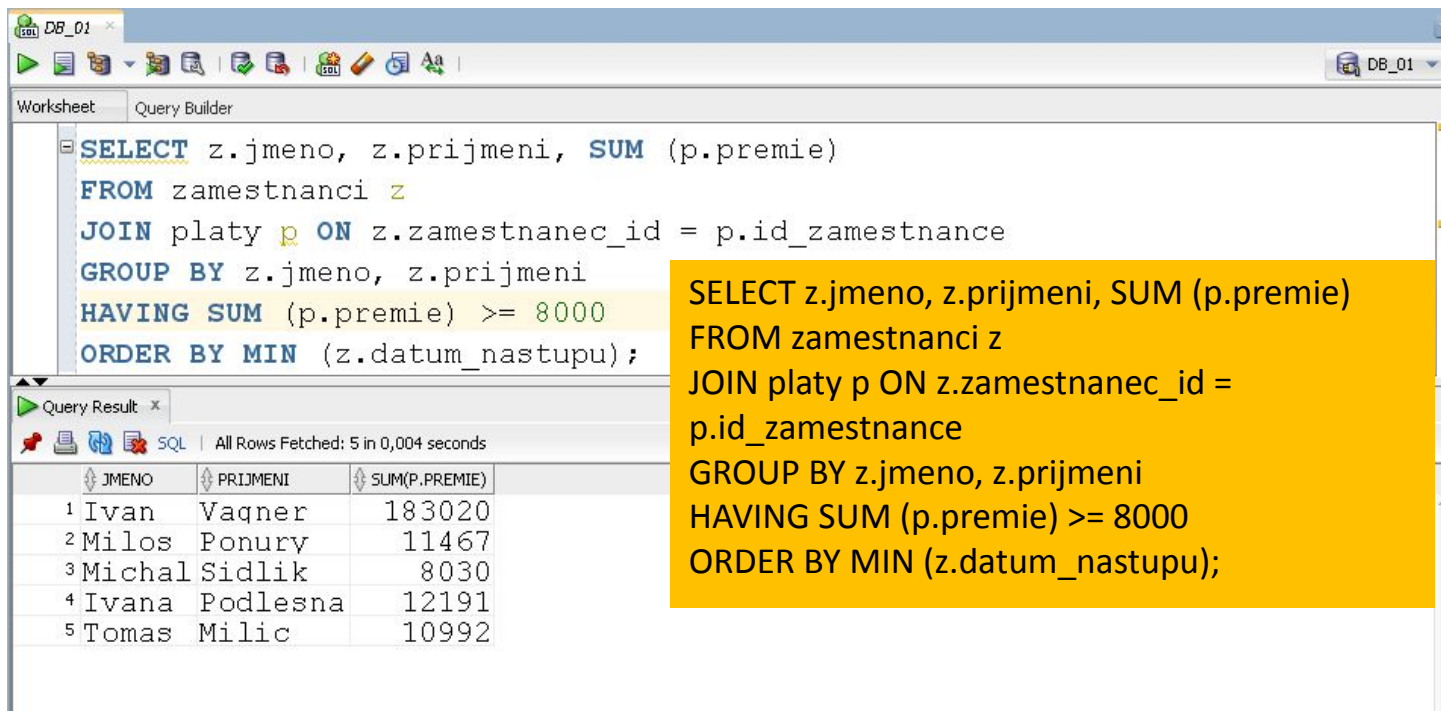
The screenshot shows a SQL Query Builder window with a yellow background for the query text. The query is as follows:

```
SELECT AVG (plat)
FROM (
  SELECT DISTINCT m.prijmeni, z.zakladni_plat AS plat
  FROM zamestnanci a
  JOIN zamestnanci m ON a.nadrizeny_id = m.zamestnanec_id
  JOIN
  (SELECT oddeleni, AVG (zakladni_plat) AS avgsal
  FROM zamestnanci GROUP BY oddeleni) b
  ON a.oddeleni = b.oddeleni
  WHERE a.zakladni_plat < b.avgsal;
```

Below the query, the 'Query Result' window shows the following output:

AVG(PLAT)
1 21525

7. Vyberte zaměstnance, kteří po dobu svého pracovního poměru získali za prémie alespoň 8000 Kč, a seřadte je podle data nástupu.



The screenshot shows a SQL query builder window with the following SQL query:

```
SELECT z.jmeno, z.prijmeni, SUM (p.premie)
FROM zamestnanci z
JOIN platy p ON z.zamestnanec_id = p.id_zamestnance
GROUP BY z.jmeno, z.prijmeni
HAVING SUM (p.premie) >= 8000
ORDER BY MIN (z.datum_nastupu);
```

The query results are displayed in a table below:

	JMENO	PRIJMENI	SUM(P.PREMIE)
1	Ivan	Vagner	183020
2	Milos	Ponury	11467
3	Michal	Sidlik	8030
4	Ivana	Podlesna	12191
5	Tomas	Milic	10992

A yellow box highlights the following SQL query:

```
SELECT z.jmeno, z.prijmeni, SUM (p.premie)
FROM zamestnanci z
JOIN platy p ON z.zamestnanec_id =
p.id_zamestnance
GROUP BY z.jmeno, z.prijmeni
HAVING SUM (p.premie) >= 8000
ORDER BY MIN (z.datum_nastupu);
```