

# Cleverlance

## SQL 1

Jednoduché dotazy  
nad jednou tabulkou

# SQL

- Structured Query Language
- vychází z angličtiny
- Vlastnosti SQL:
  - Výkonnost
  - Snadná použitelnost
  - Komplexní funkčnost

# 1.Dotaz

```
SELECT * FROM zamestnanci;
```

- V nejjednodušším tvaru musí příkaz SELECT obsahovat:
  - Klauzuli SELECT, která specifikuje, které prvky (sloupce) se mají zobrazit (vybrat)
  - Klauzuli FROM, která určuje odkud, z které tabulky (tabulek) se data vybírají

# 1.Dotaz

The screenshot shows a database application window titled "DB\_01". The main area is the "Query Builder" tab, which contains the following SQL query:

```
SELECT *  
FROM zamestnanci;
```

Below the query builder is the "Query Result" tab, which displays the results of the query. The results are shown in a table with the following columns: ZAMESTNANEC\_ID, JMENO, PRIJMENI, POZICE, NADRIZENY\_ID, DATUM\_NASTUPU, and ODDeleni. The table contains 16 rows of data.

ZAMESTNANEC_ID	JMENO	PRIJMENI	POZICE	NADRIZENY_ID	DATUM_NASTUPU	ODDELENI
1	3 Ivan	Vagner	Reditel	(null)	01.01.1998 00:00:00	3
2	8 Adela	Stadnikova	Ucetni	3	01.12.2001 00:00:00	1
3	12 Milan	Pavelka	Vedouci skladu	3	01.04.2002 00:00:00	7
4	18 Lucie	Slavikova	Vedouci vyroby	3	01.07.2000 00:00:00	5
5	26 Eliska	Sazavska	Cukrarka	18	01.08.2001 00:00:00	5
6	14 Tamara	Kounovska	Asistentka	3	01.11.2003 00:00:00	3
7	21 Milos	Ponury	PR manager	3	01.01.2001 00:00:00	3
8	1 Petr	Maly	Ucetni	8	23.06.2003 00:00:00	1
9	2 Alzbeta	Vesela	Uklizecka	14	15.08.1999 00:00:00	3
10	4 Anna	Bubova	Uklizecka	14	15.08.1999 00:00:00	3
11	6 Ales	Charvat	Ucetni	8	01.01.2003 00:00:00	1
12	7 Bohumil	Vetvicka	Ucetni	6	01.01.2005 00:00:00	1
13	9 Marcela	Kolinska	Uklizecka	18	17.03.2004 00:00:00	5
14	11 Jan	Novotny	Skladnik	12	01.05.2003 00:00:00	7
15	13 Martina	Stranska	Asistentka	12	01.05.2003 00:00:00	7
16	15 Tomas	Milic	Referent	3	15.02.2002 00:00:00	3

# Selekce

The screenshot shows a database query builder window titled "DB\_01". The "Query Builder" tab is active, displaying the SQL query: `SELECT * FROM zamestnanci;`. Below the query, the "Query Result" tab shows the results of the query. The results are displayed in a table with 16 rows and 7 columns. The columns are: ZAMESTNANEC\_ID, JMENO, PRIJMENI, POZICE, NADRIZENY\_ID, DATUM\_NASTUPU, and ODDELENI. The table contains the following data:


ZAMESTNANEC_ID	JMENO	PRIJMENI	POZICE	NADRIZENY_ID	DATUM_NASTUPU	ODDELENI	
1	3	Ivan	Vagner	Reditel	(null)	01.01.1998 00:00:00	3
2	8	Adela	Stadnikova	Ucetni	3	01.12.2001 00:00:00	1
3	12	Milan	Pavelka	Vedouci skladu	3	01.04.2002 00:00:00	7
4	18	Lucie	Slavikova	Vedouci vyroby	3	01.07.2000 00:00:00	5
5	26	Eliska	Sazavska	Cukrarka	18	01.08.2001 00:00:00	5
6	14	Tamara	Kounovska	Asistentka	3	01.11.2003 00:00:00	3
7	21	Milos	Ponury	PR manager	3	01.01.2001 00:00:00	3
8	1	Petr	Maly	Ucetni	8	23.06.2003 00:00:00	1
9	2	Alzbeta	Vesela	Uklizecka	14	15.08.1999 00:00:00	3
10	4	Anna	Bubova	Uklizecka	14	15.08.1999 00:00:00	3
11	6	Ales	Charvat	Ucetni	8	01.01.2003 00:00:00	1
12	7	Bohumil	Vetvicka	Ucetni	6	01.01.2005 00:00:00	1
13	9	Marcela	Kolinska	Uklizecka	18	17.03.2004 00:00:00	5
14	11	Jan	Novotny	Skladnik	12	01.05.2003 00:00:00	7
15	13	Martina	Stranska	Asistentka	12	01.05.2003 00:00:00	7
16	15	Tomas	Milic	Referent	3	15.02.2002 00:00:00	3

# Formulace dotazu


Vyber účetní



Vyber zaměstnance, kteří pracují na pozici účetní



Vyber ty záznamy z tabulky zamestnanci, které mají ve sloupci pozice hodnotu 'Ucetni'



```
SELECT * FROM zamestnanci WHERE  
pozice='Ucetni'
```

# Selekce

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

```
SELECT *  
FROM zamestnanci  
WHERE pozice = 'Ucetni';
```

The query is highlighted in yellow. Below the query editor, the "Query Result" tab is active, showing the results of the query. The results are displayed in a table with the following columns: ZAMESTNANEC\_ID, JMENO, PRIJMENI, POZICE, NADRIZENY\_ID, DATUM\_NASTUPU, ODDELENI, and ZAKLADNI\_PLAT. The table contains 4 rows of data.

	ZAMESTNANEC_ID	JMENO	PRIJMENI	POZICE	NADRIZENY_ID	DATUM_NASTUPU	ODDELENI	ZAKLADNI_PLAT
1	8	Adela	Stadnikova	Ucetni	3	01.12.2001 00:00:00	1	21100
2	1	Petr	Maly	Ucetni	8	23.06.2003 00:00:00	1	15000
3	6	Ales	Charvat	Ucetni	8	01.01.2003 00:00:00	1	15000
4	7	Bohumil	Vetvicka	Ucetni	6	01.01.2005 00:00:00	1	13000

# Selekce – porovnání s řetězcem

```
SELECT *  
FROM zamestnanci  
WHERE pozice='Ucetni';
```

- jmeno='Jan'
- jmeno <> 'JAN'
- jmeno <> 'Jan'



# Selekce – porovnání s číslem

```
SELECT *  
FROM zamestnanci  
WHERE oddeleni=3;
```

# Selekce – porovnání s datem

```
SELECT *  
FROM zamestnanci  
WHERE  
datum_nastupu='01.01.2003';
```

Oracle:

```
TO_DATE('2004-03-17','yyyy-mm-dd')
```

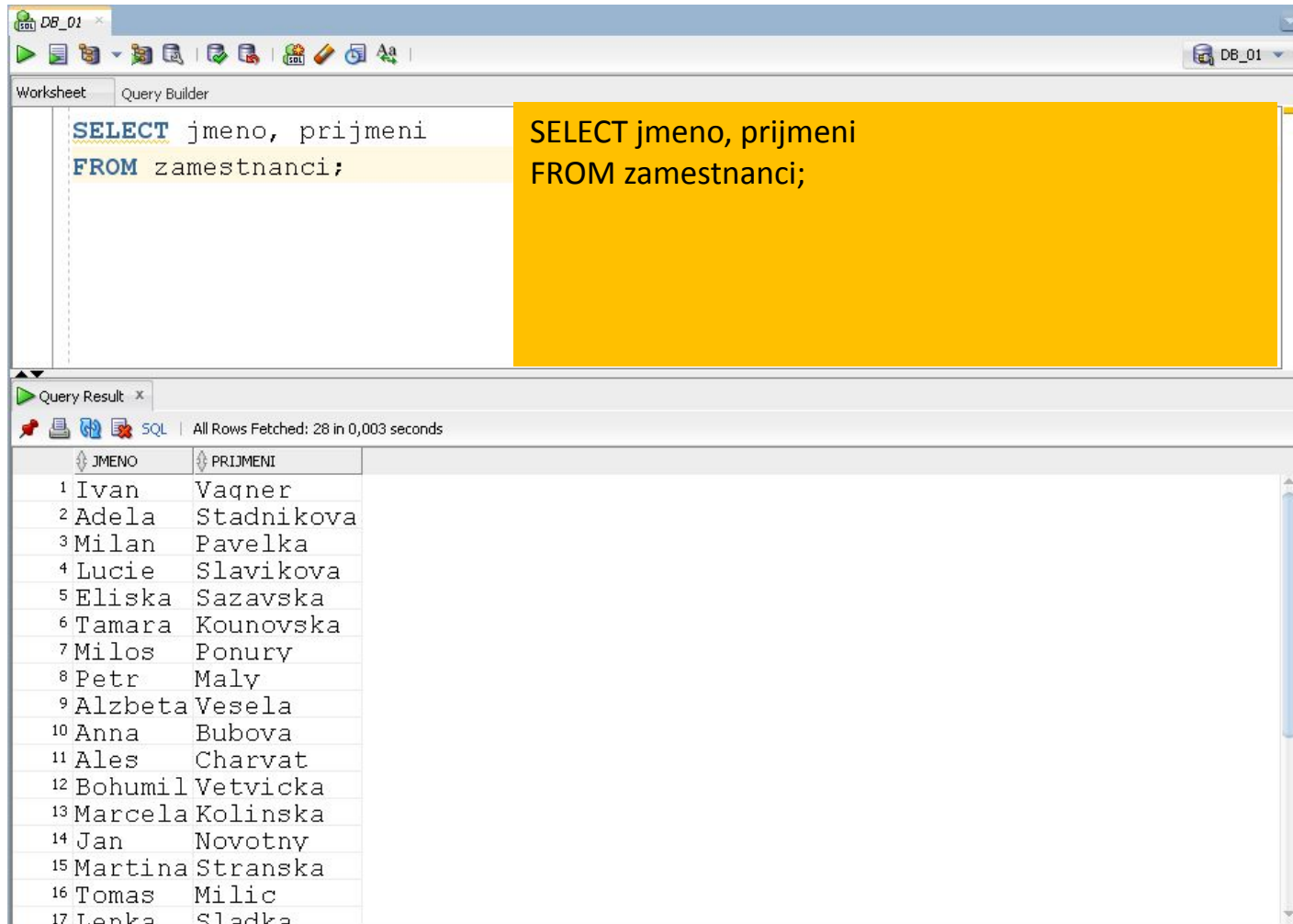
MSSQL:

```
convert(datetime,'2004-03-17',120)
```

# Operátory porovnání

Operátor	Význam
=	Rovná se
>	Větší než
>=	Větší nebo roven
<	Menší než
<=	Menší nebo roven
<> !=	Nerovná se

# Projekce



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

```
SELECT jmeno, prijmeni
FROM zamestnanci;
```

The bottom window is titled "Query Result" and shows the results of the query. The status bar indicates "All Rows Fetched: 28 in 0,003 seconds". The results are displayed in a table with two columns: "JMENO" and "PRIJMENI".

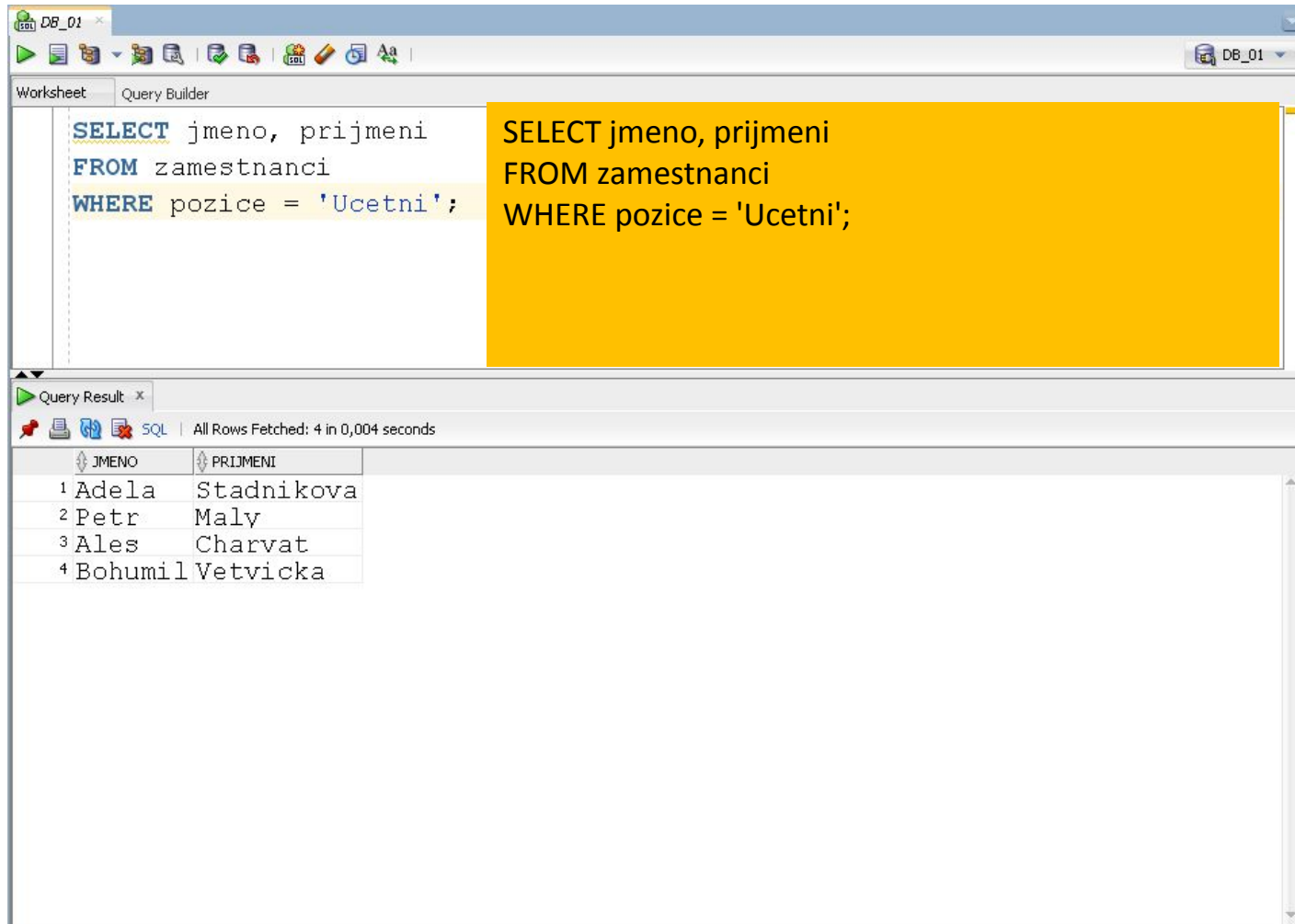
	JMENO	PRIJMENI
1	Ivan	Vagner
2	Adela	Stadnikova
3	Milan	Pavelka
4	Lucie	Slavikova
5	Eliska	Sazavska
6	Tamara	Kounovska
7	Milos	Ponury
8	Petr	Maly
9	Alzbeta	Vesela
10	Anna	Bubova
11	Ales	Charvat
12	Bohumil	Vetvicka
13	Marcela	Kolinska
14	Jan	Novotny
15	Martina	Stranska
16	Tomas	Milic
17	Lenka	Sladka

# Projekce

- Projekce omezuje výsledek dotazu pouze na sloupce, které jsou vyjmenovány za klíčovým slovem SELECT

```
SELECT jmeno, prijmeni  
FROM zamestnanci;
```

# Příklad: Selekce a projekce



The screenshot shows a database query tool interface. The top window is titled "DB\_01" and contains a "Query Builder" tab. The SQL query is displayed in a text area:

```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE pozice = 'Ucetni';
```

A yellow highlight is present over the query text. Below the query, a yellow box contains the same query text:

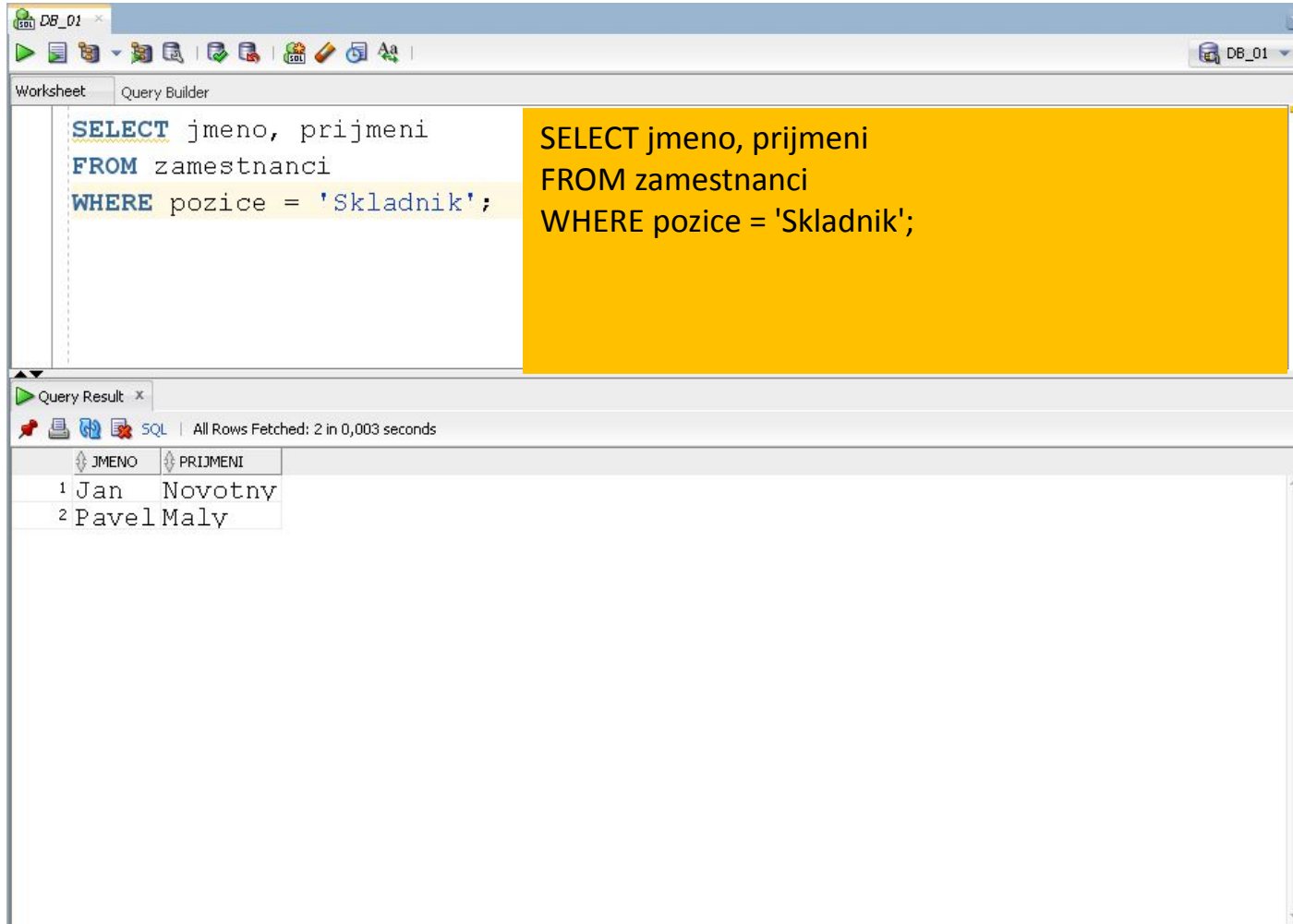
```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE pozice = 'Ucetni';
```

The bottom window is titled "Query Result" and shows the results of the query. It indicates "All Rows Fetched: 4 in 0,004 seconds". The results are displayed in a table with two columns: "JMENO" and "PRIJMENI".

	JMENO	PRIJMENI
1	Adela	Stadnikova
2	Petr	Maly
3	Ales	Charvat
4	Bohumil	Vetvicka

# Cvičení

1. Vypište jméno a příjmení všech skladníků.



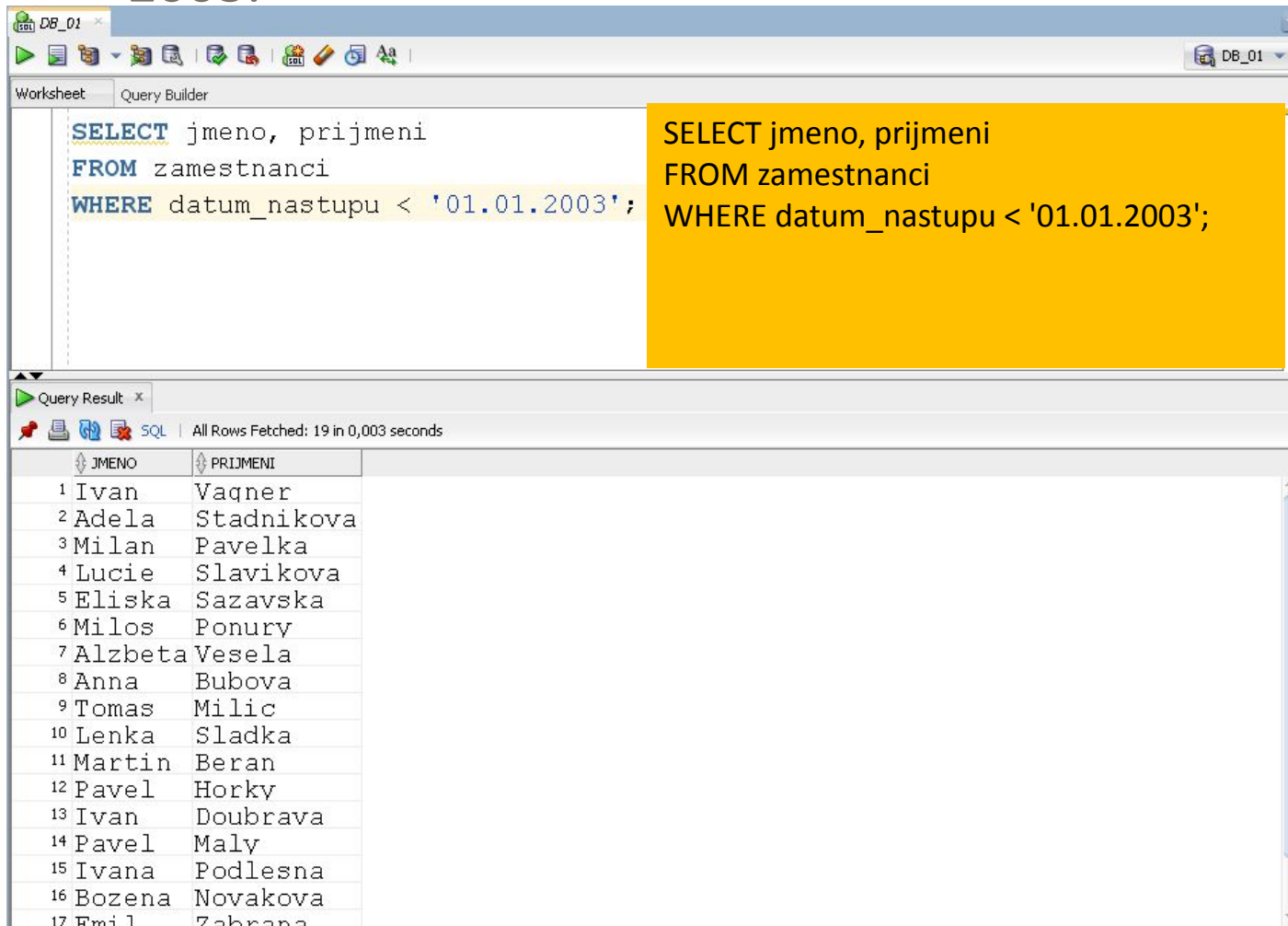
The screenshot shows a database query tool interface. The top window is titled "DB\_01" and contains a "Query Builder" tab. The SQL query entered is:

```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE pozice = 'Skladnik';
```

A yellow highlight is placed over the query text. Below the query builder is a "Query Result" window. It shows the execution status: "All Rows Fetched: 2 in 0,003 seconds". The results are displayed in a table with two columns: "JMENO" and "PRIJMENI".

	JMENO	PRIJMENI
1	Jan	Novotny
2	Pavel	Maly

2. Vypište jména zaměstnanců, kteří nastoupili před rokem 2003.



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

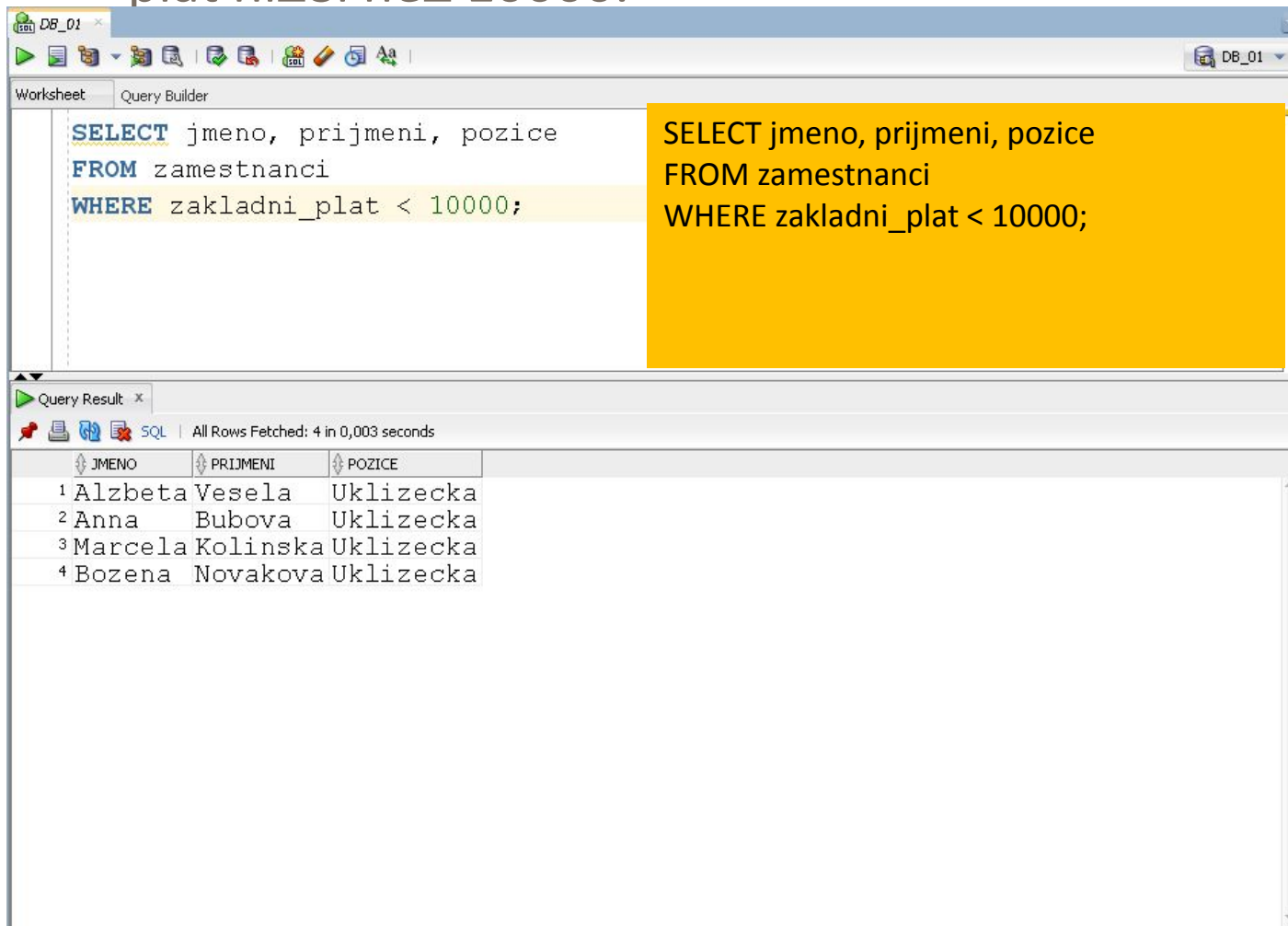
```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE datum_nastupu < '01.01.2003';
```

The query results are displayed in a table with 17 rows. The columns are labeled "JMENO" and "PRIJMENI".

	JMENO	PRIJMENI
1	Ivan	Vagner
2	Adela	Stadnikova
3	Milan	Pavelka
4	Lucie	Slavikova
5	Eliska	Sazavska
6	Milos	Ponury
7	Alzbeta	Vesela
8	Anna	Bubova
9	Tomas	Milic
10	Lenka	Sladka
11	Martin	Beran
12	Pavel	Horky
13	Ivan	Doubrava
14	Pavel	Maly
15	Ivana	Podlesna
16	Bozena	Novakova
17	Emil	Zabrana



3. Vypište jména a pozice zaměstnanců, kteří mají základní plat nižší než 10000.



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 jmeno, prijmeni, pozice
FROM zamestnanci
WHERE zakladni_plat < 10000;
```

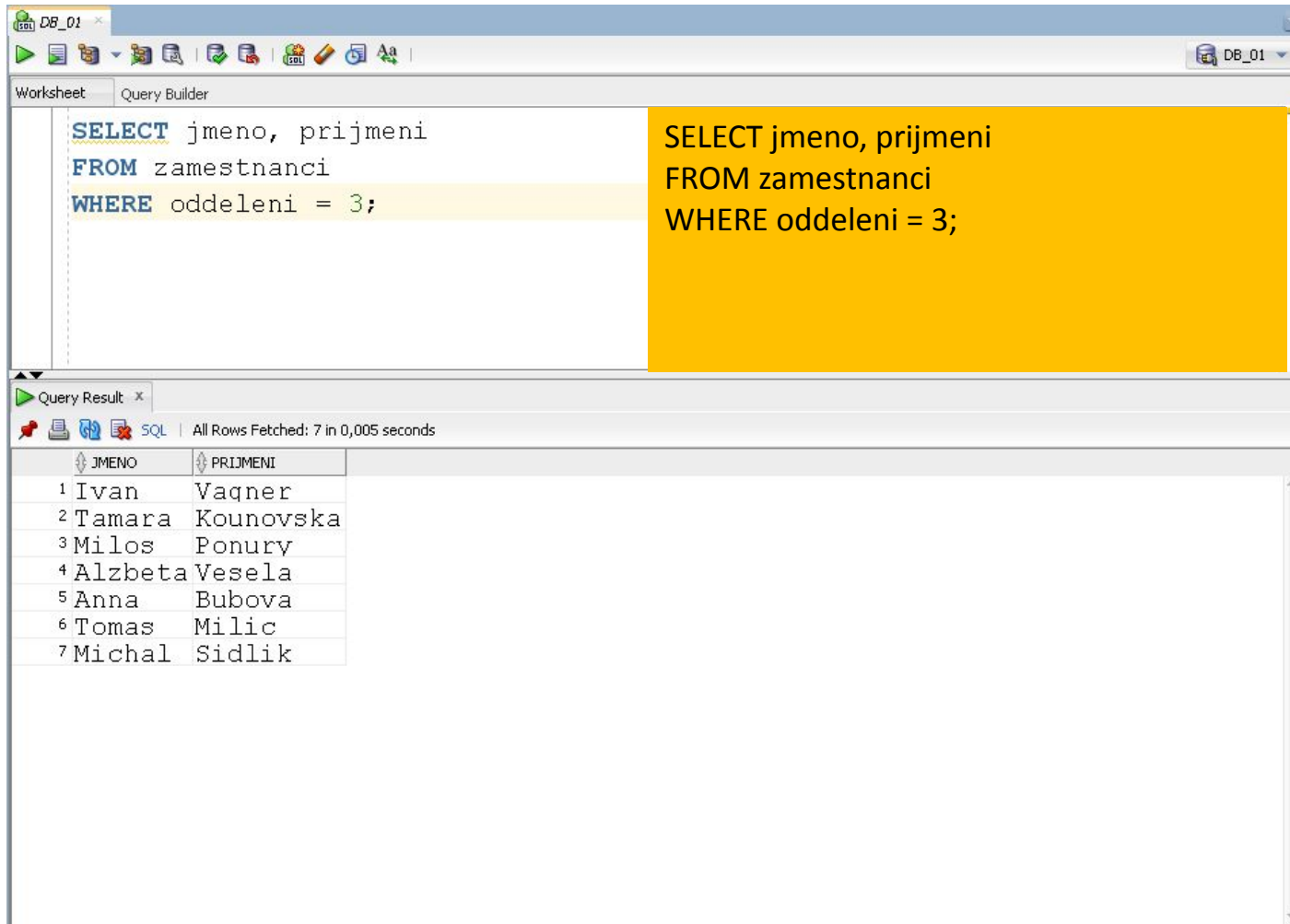
A yellow highlight is present over the query text. To the right of the query, a yellow box contains the same query text:

```
SELECT jmeno, prijmeni, pozice
FROM zamestnanci
WHERE zakladni_plat < 10000;
```

The bottom window, titled "Query Result", shows the results of the query. It indicates "All Rows Fetched: 4 in 0,003 seconds". The results are displayed in a table with the following columns: JMENO, PRIJMENI, and POZICE.

	JMENO	PRIJMENI	POZICE
1	Alzbeta	Vesela	Uklizecka
2	Anna	Bubova	Uklizecka
3	Marcela	Kolinska	Uklizecka
4	Bozena	Novakova	Uklizecka

## 4. Vypište zaměstnance oddělení číslo 3



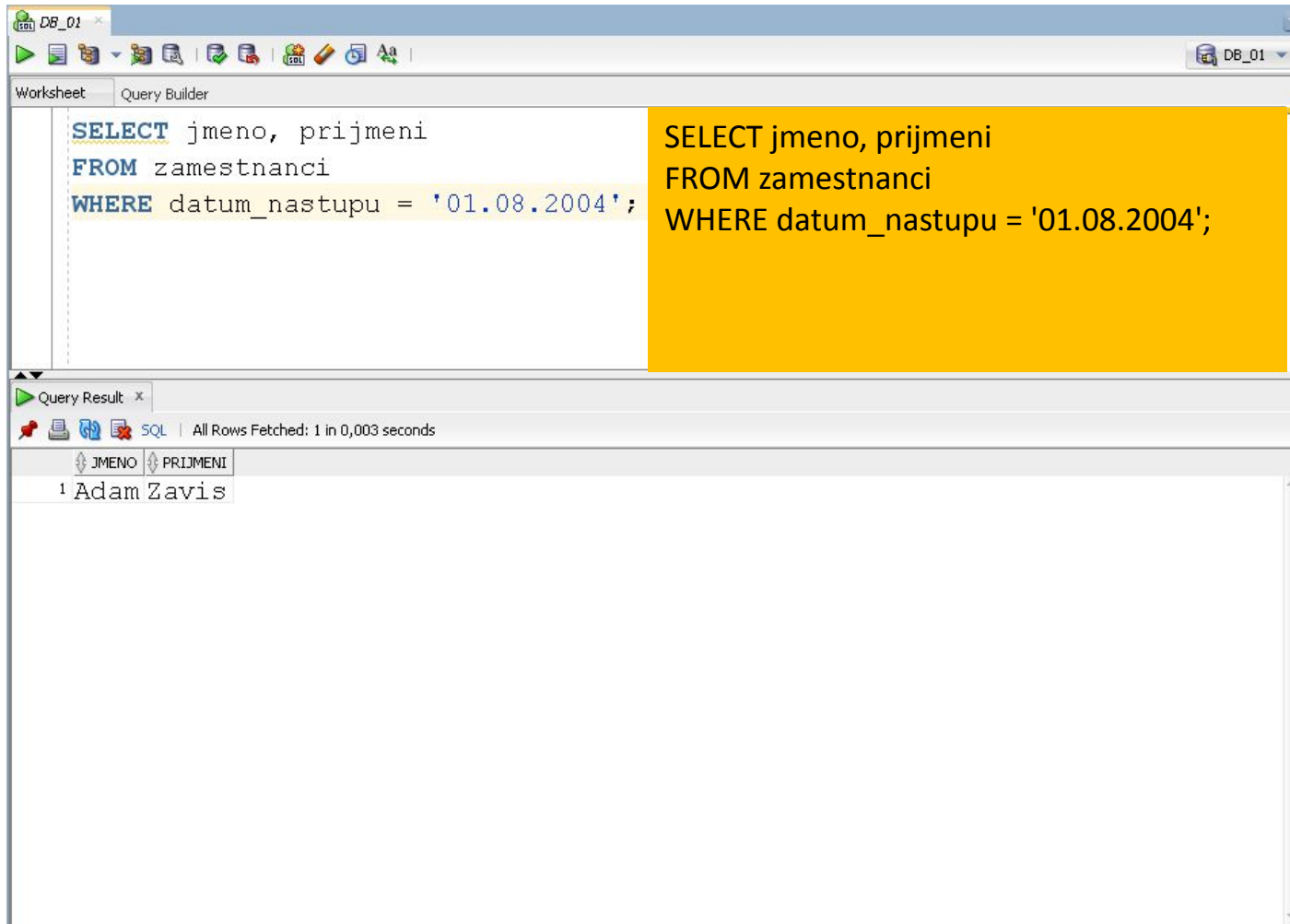
The screenshot shows a SQL query builder window with the following SQL query:

```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE oddeleni = 3;
```

The query results are displayed in a table with 7 rows and 2 columns: JMENO and PRIJMENI.

	JMENO	PRIJMENI
1	Ivan	Vagner
2	Tamara	Kounovska
3	Milos	Ponury
4	Alzbeta	Vesela
5	Anna	Bubova
6	Tomas	Milic
7	Michal	Sidlik

## 5. Kdo nastoupil do firmy 1.8.2004?



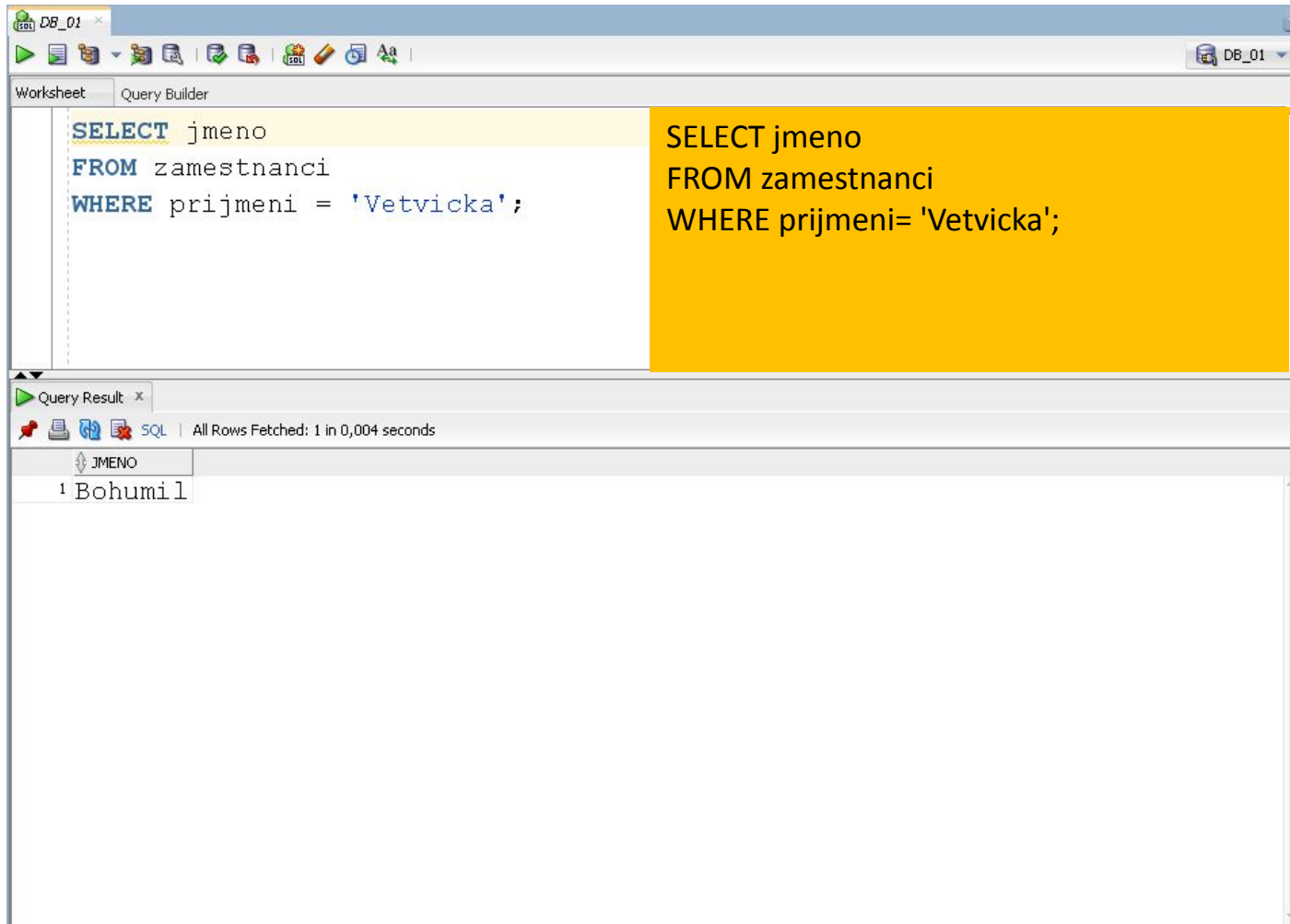
The screenshot shows a database query tool interface. The top window, titled "DB\_01", contains a "Query Builder" tab with the following SQL query:

```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE datum_nastupu = '01.08.2004';
```

A yellow highlight is present over the query text. Below the query editor, the "Query Result" window shows the execution status: "All Rows Fetched: 1 in 0,003 seconds". The result is displayed in a table with two columns: "JMENO" and "PRIJMENI".

JMENO	PRIJMENI
1 Adam	Zavis

## 6. Jaké je křestní jméno zaměstnance Větvíčky?



The screenshot displays a SQL query editor window titled "DB\_01" with a "Query Builder" tab. The query text is:

```
SELECT jmeno  
FROM zamestnanci  
WHERE prijmeni = 'Vetvicka';
```

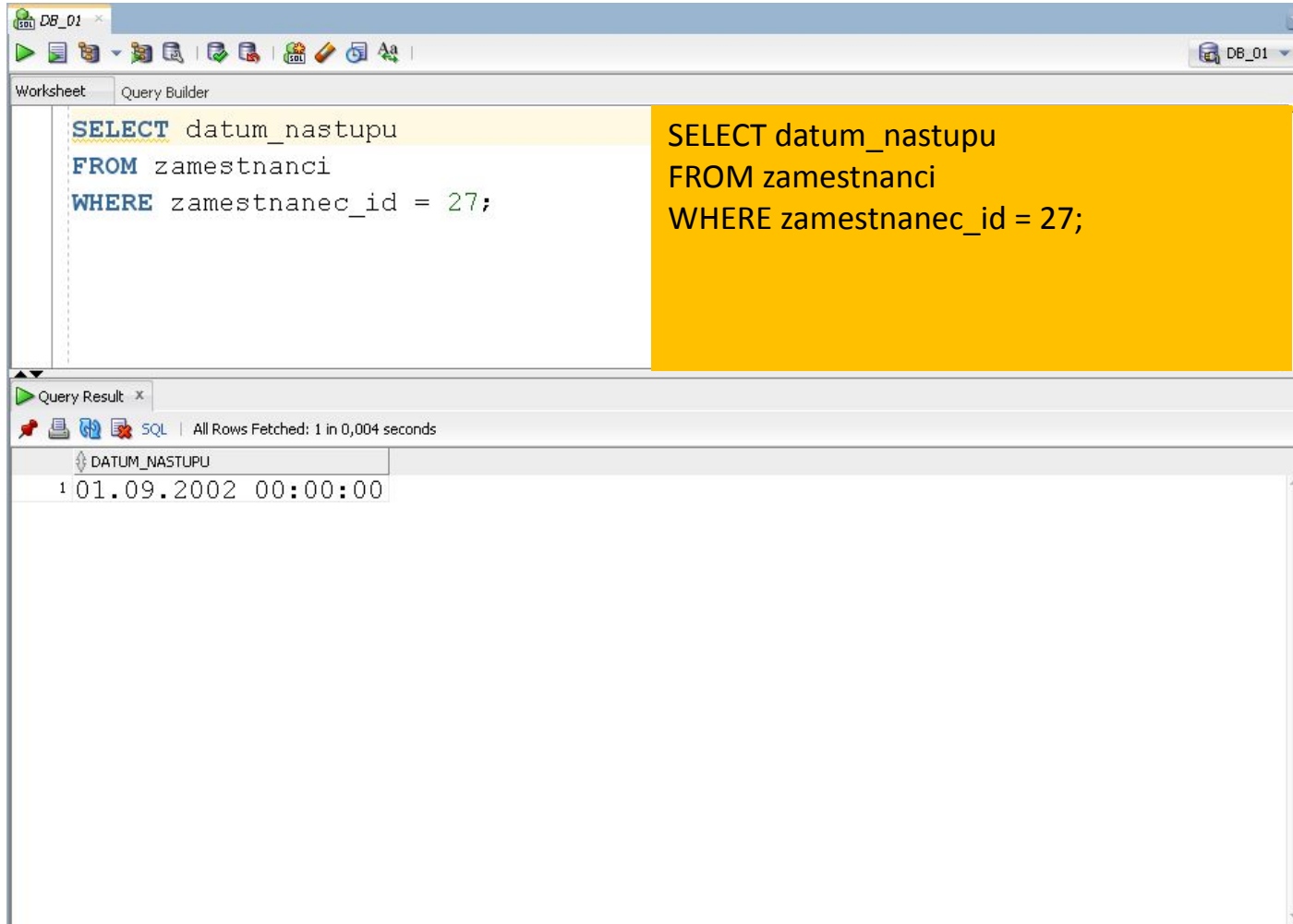
A yellow highlight is present over the query text. Below the query editor, the "Query Result" window shows the following result:

JMENO
1 Bohumil

The "Query Result" window also displays "All Rows Fetched: 1 in 0,004 seconds".

# Cvičení

## 7. Kdy nastoupil zaměstnanec s číslem 27?



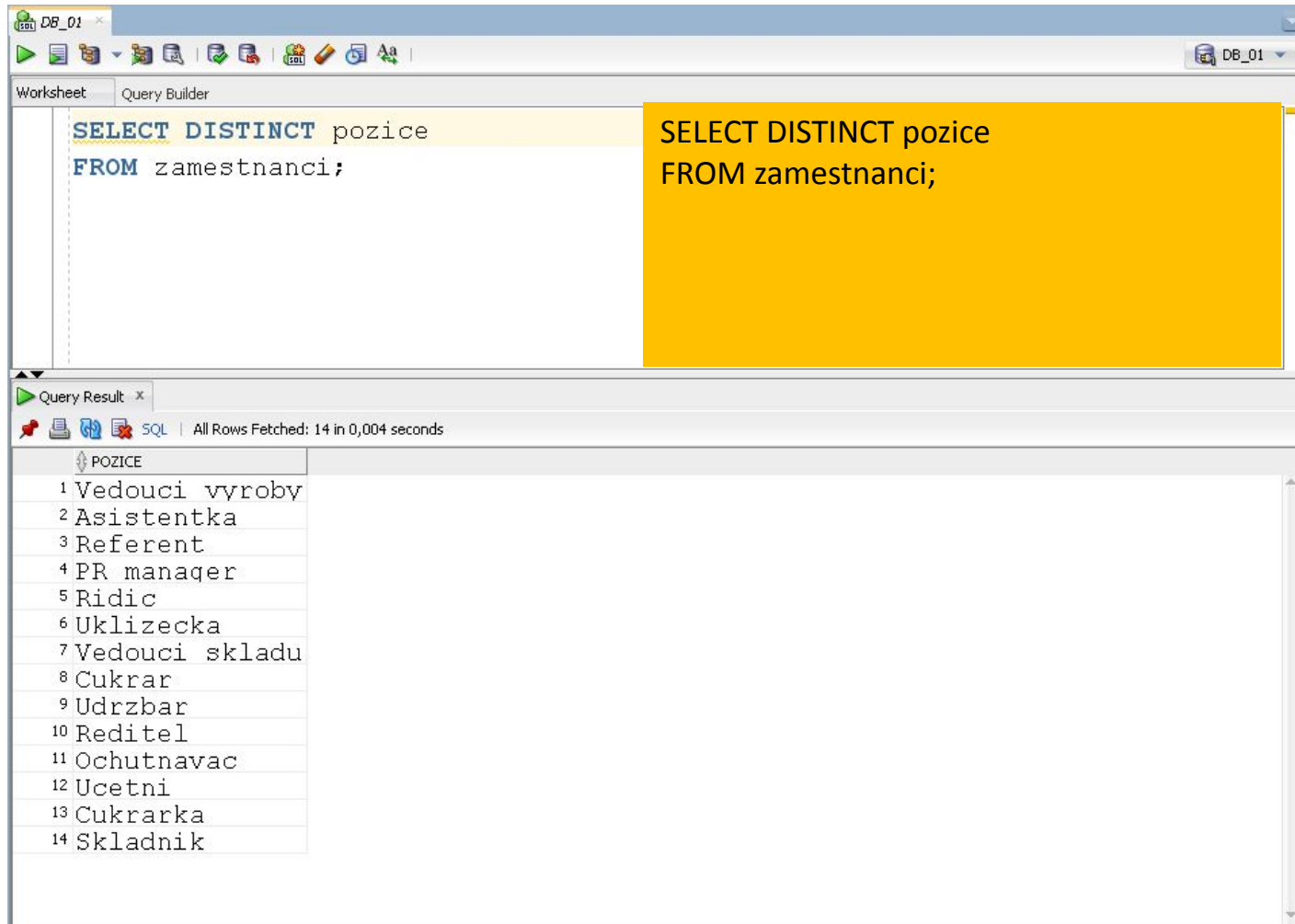
The screenshot shows a SQL query builder window with the following content:

```
SELECT datum_nastupu
FROM zamestnanci
WHERE zamestnanec_id = 27;
```

The query result is displayed in a table with the following data:

DATUM_NASTUPU
1 01.09.2002 00:00:00

# DISTINCT



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

```
SELECT DISTINCT pozice  
FROM zamestnanci;
```

The query results are displayed in a table below. The table has a column header "POZICE" and 14 rows of data. The status bar indicates "All Rows Fetched: 14 in 0,004 seconds".

POZICE
1 Vedouci vyroby
2 Asistentka
3 Referent
4 PR manager
5 Ridic
6 Uklizecka
7 Vedouci skladu
8 Cukrar
9 Udrzbar
10 Reditel
11 Ochutnavac
12 Ucetni
13 Cukrarka
14 Skladnik

# DISTINCT

- Duplicity se z výstupu odstraňují pomocí klíčového slova DISTINCT
- klíčové slovo DISTINCT se píše za klíčové slovo SELECT a před seznam sloupců v klauzuli SELECT

```
SELECT DISTINCT pozice  
FROM zamestnanci;
```

# Výraz

```
SELECT jmeno, prijmeni,  
12*zakladni_plat  
FROM zamestnanci;
```

Operátor	Popis
+	Sčítání
-	Odčítání
*	Násobení
/	Dělení

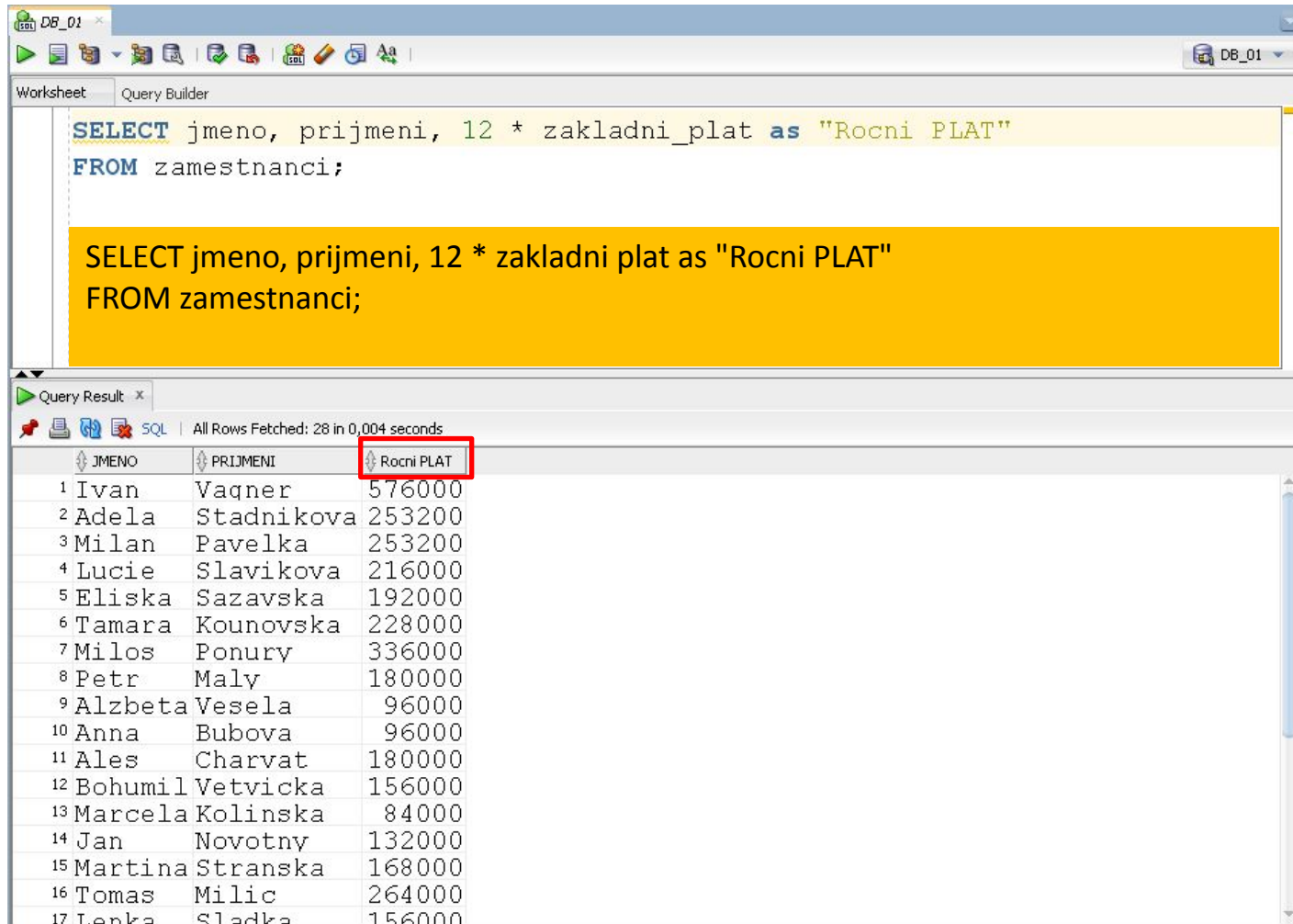


# Priorita operátorů



- Násobení a dělení má přednost před sčítáním a odečítáním
- Operátory se stejnou prioritou se provádějí zleva doprava
- Použití závorek mění pořadí provádění a zjednodušují výrazy

# alias, AS



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

```
SELECT jmeno, prijmeni, 12 * zakladni_plat as "Rocni PLAT"  
FROM zamestnanci;
```

A yellow highlight is placed over the query text. Below the query editor, the "Query Result" window shows the output of the query. The status bar indicates "All Rows Fetched: 28 in 0,004 seconds". The result is displayed as a table with three columns: "JMENO", "PRIJMENI", and "Rocni PLAT". The "Rocni PLAT" column header is highlighted with a red box. The table contains 17 rows of data.

	JMENO	PRIJMENI	Rocni PLAT
1	Ivan	Vagner	576000
2	Adela	Stadnikova	253200
3	Milan	Pavelka	253200
4	Lucie	Slavikova	216000
5	Eliska	Sazavska	192000
6	Tamara	Kounovska	228000
7	Milos	Ponury	336000
8	Petr	Maly	180000
9	Alzbeta	Vesela	96000
10	Anna	Bubova	96000
11	Ales	Charvat	180000
12	Bohumil	Vetvicka	156000
13	Marcela	Kolinska	84000
14	Jan	Novotny	132000
15	Martina	Stranska	168000
16	Tomas	Milic	264000
17	Lenka	Sladka	156000

# alias, AS

- Alias umožňuje přejmenovat sloupec výsledku dotazu

```
SELECT jmeno, prijmeni,  
12*zakladni_plat AS  
"Rocni plat"  
FROM zamestnanci;
```

# SELECT - shrnutí

```
SELECT * | { [DISTINCT]  
sloupec | výraz [alias], ... }  
FROM tabulka  
[WHERE podmínka (y) ]
```

# AND

- klauzule WHERE může více podmínek, které musí být splněny současně
- Vyberte zaměstnance, kteří nastoupili roce 2003 a dříve a pracují jako účetní

```
SELECT * FROM zamestnanci
```

```
WHERE datum_nastupu <= '31-12-2003'
```

```
AND pozice = 'Ucetni';
```

# OR

- Spojkou OR jsou spojeny podmínky, z nichž alespoň jedna musí být splněna
- Příklad  
Vyberte všechny cukráře a cukrářky

```
SELECT * FROM zamestnanci  
WHERE pozice='Cukrar' OR  
pozice='Cukrarka';
```

# BETWEEN AND

- umožňuje vybrat takové záznamy, pro které je hodnota v uvedeném sloupci v určitém rozmezí (včetně hraničních hodnot)

... **WHERE** sloupec **BETWEEN** dolni\_mez  
**AND** horni\_mez

- Možno použít pro:
  - datum
  - číslo
  - textový řetězec

# BETWEEN AND

Příklad:

Vypište zaměstnance kteří nastoupili v roce 2003

```
SELECT *  
FROM zamestnanci  
WHERE datum nastupu BETWEEN  
'01-01-2003' AND '31-12-2003';
```



# IN

- Porovnává hodnotu se seznamem
- Příklad:  
Vyberte všechny cukráře a cukrářky

```
SELECT *  
FROM zamestnanci  
WHERE pozice IN  
( 'Cukrar', 'Cukrarka' );
```

# LIKE

- Porovnává řetězec s uvedenou maskou
- Zástupné znaky:
  - `_` (podtržítko) nahrazuje právě jeden znak
  - `%` nahrazuje libovolný počet znaků (nula a více znaků)
- Maska `'_a%'` odpovídá slovům:  
Pavel        `_` nahrazuje `P`, `%` nahrazuje `'vel'`  
maminka    `_` nahrazuje `m`, `%` nahrazuje `'minka'`

# LIKE

- Příklad:

Kteří zaměstnanci mají příjmení, které začíná na písmeno N:

```
SELECT jmeno, prijmeni  
FROM zamestnanci  
WHERE prijmeni LIKE 'N%';
```

# Masky – cvičení

- Které z vyjmenovaných slov odpovídají masce?

- LIKE 'N%'

- ~~'novak'~~, 'Novak', 'N', ~~'Hanák'~~,



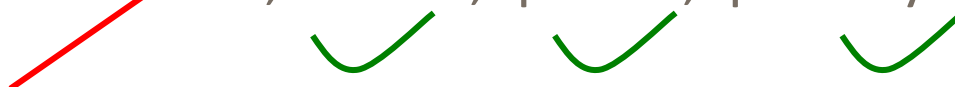
- LIKE '\_a%'

- 'pa', 'papa', ~~'ahoj'~~, 'pavel'



- LIKE '%les%'

- ~~'Lesník'~~, 'lesnik', 'prales', 'polesny'



# IS NULL

- NULL je speciální hodnota pro hodnoty, které nejsou vyplněny
- NULL se nemůže porovnávat pomocí = nebo <>
- Pro porovnání je třeba použít IS NULL nebo IS NOT NULL
- Příklad:  
Který zaměstnanec nemá nadřízeného?

```
SELECT jmeno, prijmeni, pozice  
FROM zamestnanci  
WHERE nadrizeny_id IS NULL;
```

# NOT

- Negace následující podmínky
- Příklad:

Vypiš jména všech zaměstnanců kromě účetních

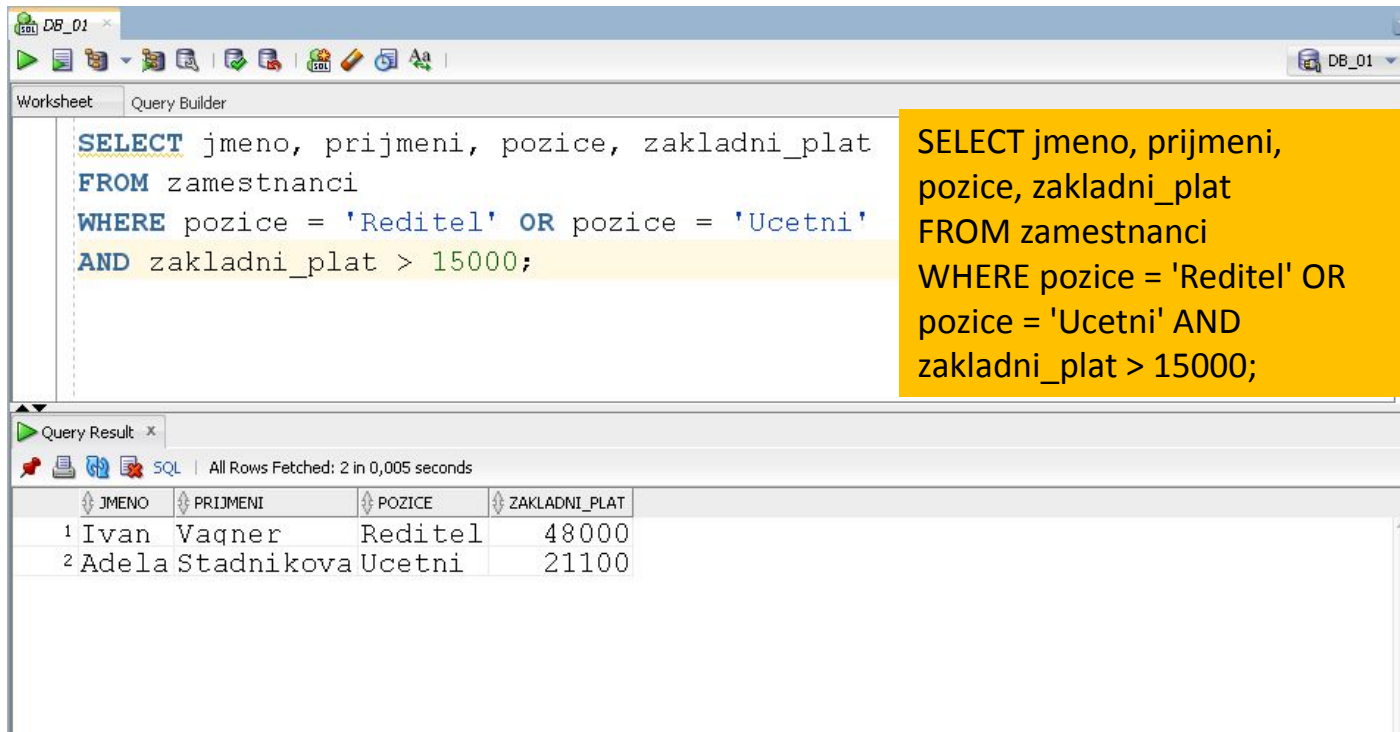
```
SELECT *  
FROM zamestnanci  
WHERE NOT pozice= 'Ucetni';
```

# Priority operatoru

Pořadí vyhodnocení	Operátor
1	Aritmetické operace ( * / + - )
2	Operátor zřetězení (    )
3	Porovnání
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	NOT logická podmínka
7	AND logická podmínka
8	OR logická podmínka

# Cvičení

8. Vypište zaměstnance, kteří pracují na pozici 'Reditel' nebo jsou na pozici 'Ucetni' s platem větším než 15000



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

```
SELECT jmeno, prijmeni, pozice, zakladni_plat
FROM zamestnanci
WHERE pozice = 'Reditel' OR pozice = 'Ucetni'
AND zakladni_plat > 15000;
```

The query results are displayed in a table with the following data:

	JMENO	PRIJMENI	POZICE	ZAKLADNI_PLAT
1	Ivan	Vagner	Reditel	48000
2	Adela	Stadnikova	Ucetni	21100



## 9. Vypište zaměstnance :

- a) kteří mají na druhém místě v křestním jméně písmeno 'a'
- b) kteří nemají na druhém místě v křestním jméně písmeno 'a'
- c) jejich příjmení končí písmenem 'a'
- d) kteří mají kdekoliv ve jméně 'i'
- e) kteří pracují na pozici 'Ucetni' nebo 'Referent'
- f) kteří nemají nadřízené

```
SELECT jmeno, prijmeni
FROM zamestnanci
WHERE nadrizeny_id IS NULL
OR (jmeno LIKE '%a%' OR prijmeni LIKE '%a%'
OR jmeno LIKE '%i%' OR prijmeni LIKE '%i%'
OR pozice IN ('Ucetni', 'Referent'))
```

# ORDER BY

- Třídění výsledku dotazu podle jednoho nebo více sloupců
- Vzestupné třídění ASC (defaultní)
- Sestupné třídění DESC

```
SELECT prijmeni, jmeno  
FROM zamestnanci  
ORDER BY prijmeni, jmeno ASC;
```

# Agregační funkce

- AVG      *průměr*
- COUNT    *počet*
- MAX        *maximum*
- MIN        *minimum*
- SUM        *součet*

# COUNT

- Kolik má tabulka zamestnanci záznamů?

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

- Kolik zaměstnanců nastoupilo v průběhu roku 2003?

```
SELECT COUNT (*)  
FROM zamestnanci  
WHERE datum_nastupu BETWEEN  
'01.01.2003' AND '31.12.2003';
```

# AVG

- Jaký je průměrný základní plat v tabulce zamestnanci?

```
SELECT AVG (zakladni_plat)  
FROM zamestnanci;
```

# MAX a MIN

- Kolik je maximální plat?

```
SELECT MAX (zakladni_plat)  
FROM zamestnanci;
```

- Od kdy pracuje ve firmě zaměstnanec, který je pracuje ve firmě pracuje nejdéle?

```
SELECT MIN (datum_nastupu)  
FROM zamestnanci;
```

# Operátor zřetězení

- Umožňuje spojit libovolné řetězce
- || (dvě svislé čáry)

```
SELECT 'Tabulka zamestnanci ma '  
|| count(*) || ' zaznamu' AS  
"Vysledek dotazu"  
FROM zamestnanci;
```

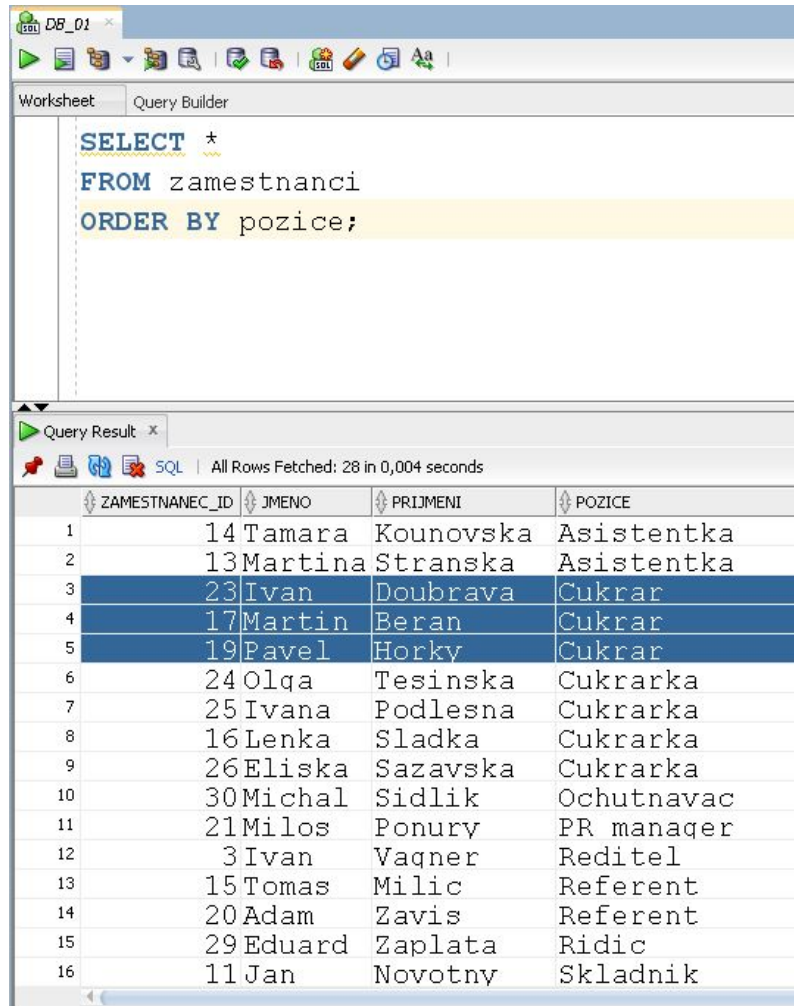
	Vysledek dotazu
▶ 1	Tabulka zamestnanci ma 29 zaznamu ...

# GROUP BY

- GROUP BY umožňuje seskupit řádky a získávat souhrnné informace za jednotlivé skupiny
- Užití společně s agregačními funkcemi
- Všechny sloupce, které jsou v seznamu sloupců SELECT, které nejsou v agregačních funkcích, musí být uvedeny v klauzuli GROUP BY



# GROUP BY



```
SELECT *
FROM zamestnanci
ORDER BY pozice;
```

ZAMESTNANEC_ID	JMENO	PRIJMENI	POZICE
1	14 Tamara	Kounovska	Asistentka
2	13 Martina	Stranska	Asistentka
3	23 Ivan	Doubrava	Cukrar
4	17 Martin	Beran	Cukrar
5	19 Pavel	Horky	Cukrar
6	24 Olga	Tesinska	Cukrarka
7	25 Ivana	Podlesna	Cukrarka
8	16 Lenka	Sladka	Cukrarka
9	26 Eliska	Sazavska	Cukrarka
10	30 Michal	Sidlik	Ochutnavac
11	21 Milos	Ponury	PR manaqer
12	3 Ivan	Vaqner	Reditel
13	15 Tomas	Milic	Referent
14	20 Adam	Zavis	Referent
15	29 Eduard	Zaplata	Ridic
16	11 Jan	Novotny	Skladnik

Asistentk  
Cukra  
r  
Cukrark  
a

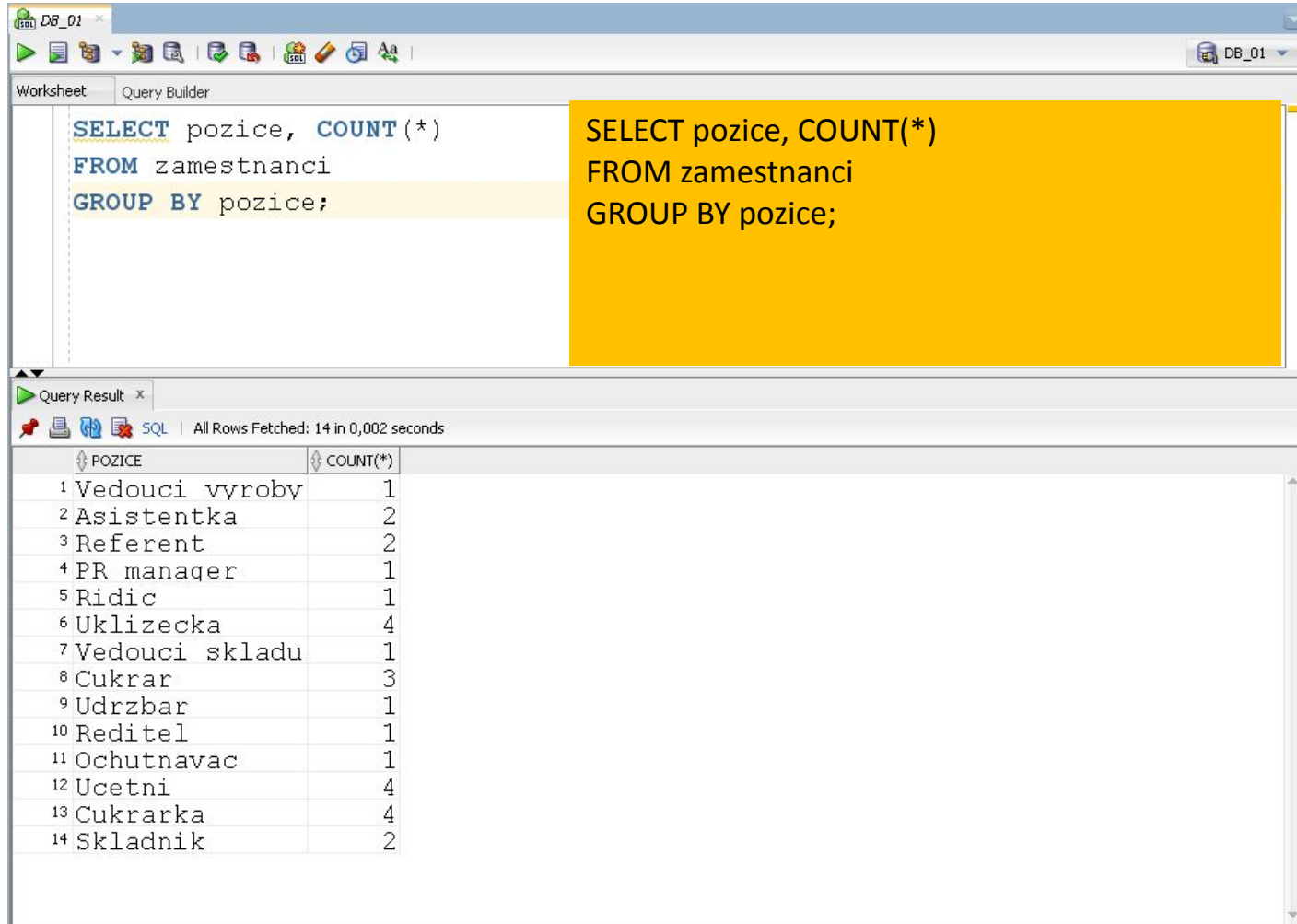
# GROUP BY

- Příklad:

Kolik lidí pracuje na jednotlivých pozicích?

```
SELECT pozice, COUNT (*)  
FROM zamestnanci  
GROUP BY pozice;
```

# GROUP BY



The screenshot shows a database application window titled "DB\_01". The main area is the "Query Builder" tab, which contains a SQL query:

```
SELECT pozice, COUNT(*)  
FROM zamestnanci  
GROUP BY pozice;
```

A yellow highlight is present under the "GROUP BY" clause in the original image. To the right of the query, a yellow box contains the same query text.

Below the query builder is the "Query Result" tab, which displays the results of the query. The status bar indicates "All Rows Fetched: 14 in 0,002 seconds". The results are shown in a table with two columns: "POZICE" and "COUNT(\*)".

POZICE	COUNT(*)
1 Vedouci vyroby	1
2 Asistentka	2
3 Referent	2
4 PR manager	1
5 Ridic	1
6 Uklizecka	4
7 Vedouci skladu	1
8 Cukrar	3
9 Udrzbar	1
10 Reditel	1
11 Ochutnavac	1
12 Ucetni	4
13 Cukrarka	4
14 Skladnik	2

# HAVING

- Podmínky, které se vztahují ke skupině nemůžeme psát do klauzule WHERE
- Vlastnosti skupiny píšeme do klauzule HAVING
- Můžeme používat agregační funkce, které nemusí být uvedeny v klauzuli SELECT

# HAVING

- Příklad:

Vypište pozice a počet zaměstnanců u pozic, které mají průměrný plat alespoň 15000

```
SELECT pozice, COUNT (*)  
FROM zamestnanci  
GROUP BY pozice  
HAVING AVG(zakladni_plat) >= 15000;
```

# Pořadí klíčových slov v dotazu

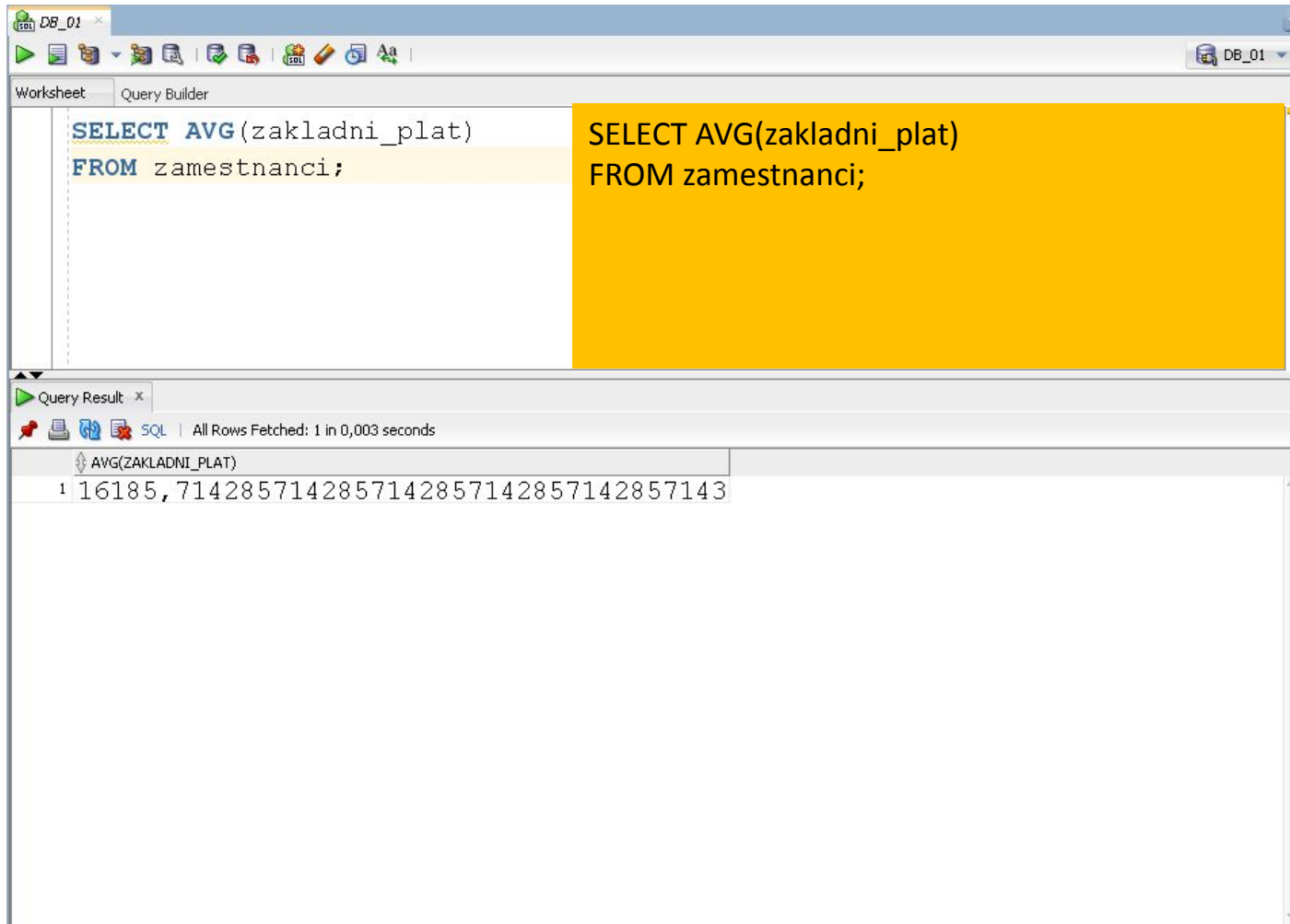
SELECT	<i>sloupce, výrazy</i>
FROM	<i>tabulka</i>
[WHERE	<i>podmínky]</i>
[GROUP BY	<i>výraz pro seskupení]</i>
[HAVING	<i>podmínky pro skupiny]</i>
[ORDER BY	<i>sloupce]</i>

# Cvičení (pokračování)

10. Kolik je průměrný základní plat?
11. Kolik ve firmě pracuje účetních?
12. Kolik je průměrný plat uklízeček?
13. Vypište průměrný plat podle oddělení
14. Vypište průměrný plat na odděleních, které mají více než jednoho zaměstnance.
15. Napište seznam zaměstnanců seříděný podle toho, jak dlouho ve firmě pracují.
16. Jaký je průměrný plat zaměstnanců, kteří nastoupili před rokem 2000?

# Cvičení (pokračování)

10. Kolik je průměrný základní plat?



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

```
SELECT AVG(zakladni_plat)
FROM zamestnanci;
```

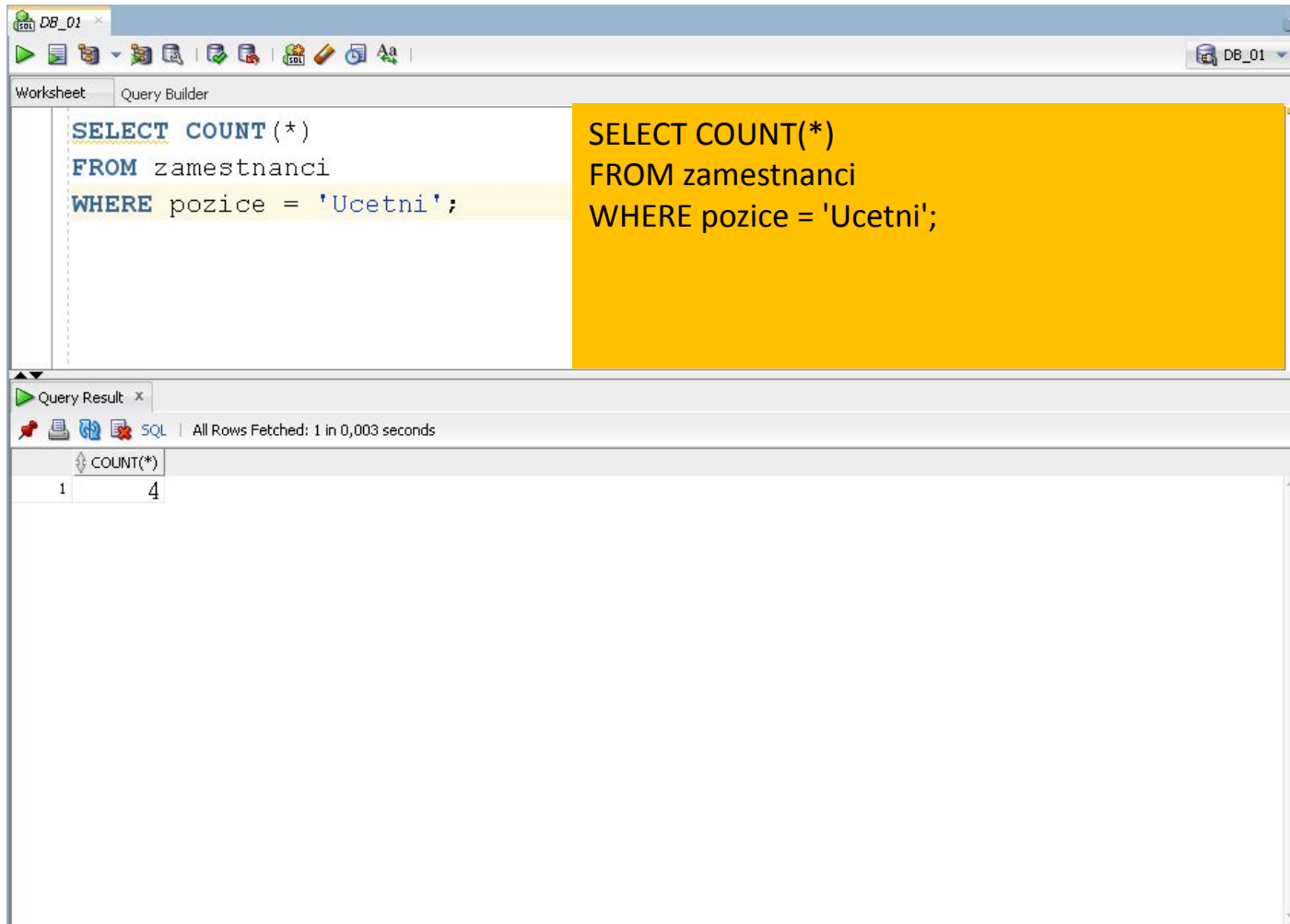
The query is highlighted in yellow. Below the query, the "Query Result" window is visible, showing the result of the query. The result is a single row with the value 16185,7142857142857142857142857143.

AVG(ZAKLADNI_PLAT)
1 16185,7142857142857142857142857143



# Cvičení (pokračování)

11. Kolik ve firmě pracuje účetních?



The screenshot shows a database query tool interface. The top window is titled "DB\_01" and contains a "Query Builder" tab. The SQL query is displayed in a text area:

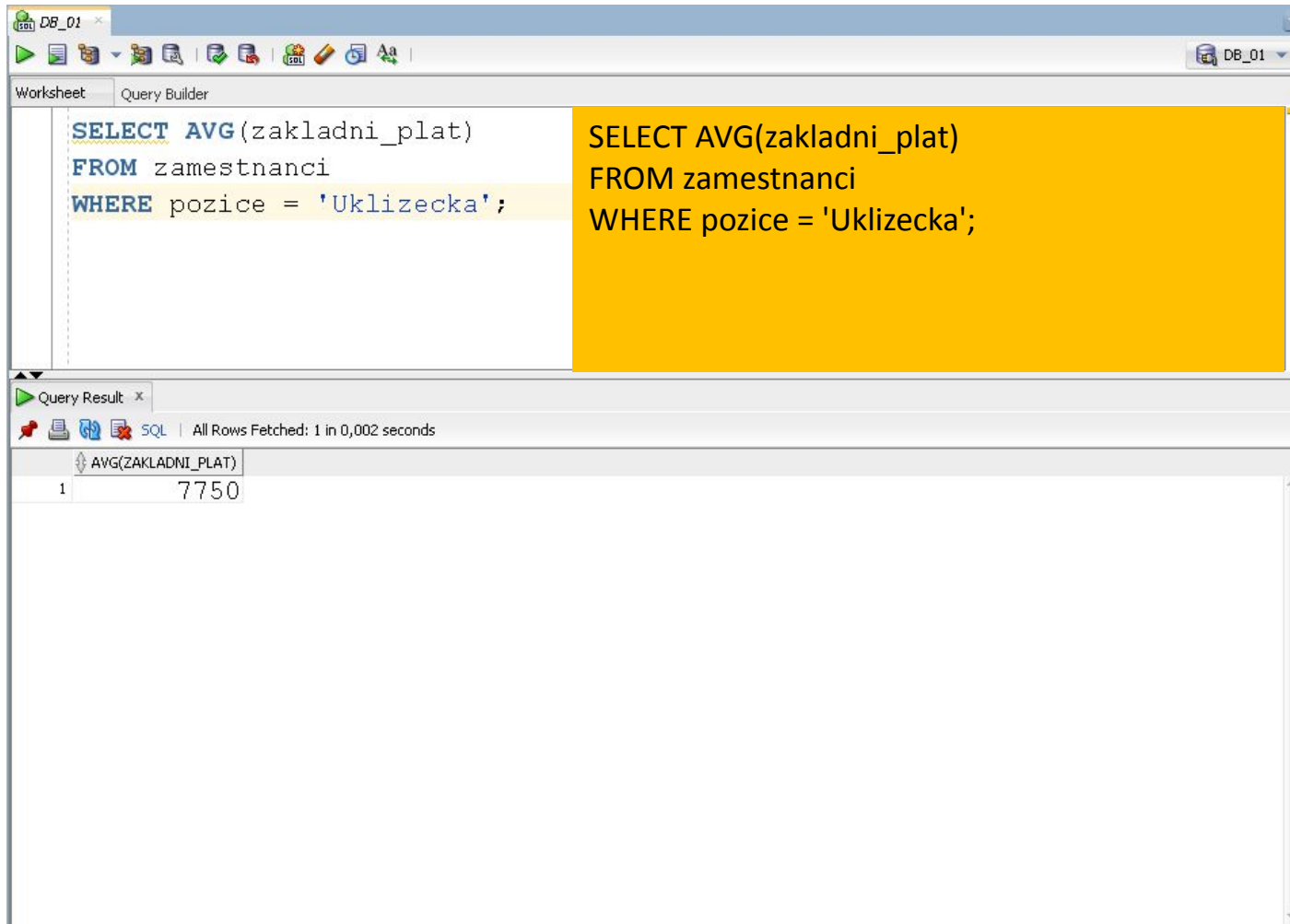
```
SELECT COUNT(*)  
FROM zamestnanci  
WHERE pozice = 'Ucetni';
```

A yellow highlight is visible over the query text. Below the query builder is a "Query Result" window. It shows the execution status: "All Rows Fetched: 1 in 0,003 seconds". The result is displayed in a table with one row and one column:

COUNT(*)
4

# Cvičení (pokračování)

12. Kolik je průměrný plat uklízeček?



The screenshot shows a SQL query editor window with two panes. The left pane shows the query in a code editor, and the right pane shows the same query on a yellow background. Below the editor is a 'Query Result' window displaying the result of the query.

```
SELECT AVG(zakladni_plat)
FROM zamestnanci
WHERE pozice = 'Uklizecka';
```

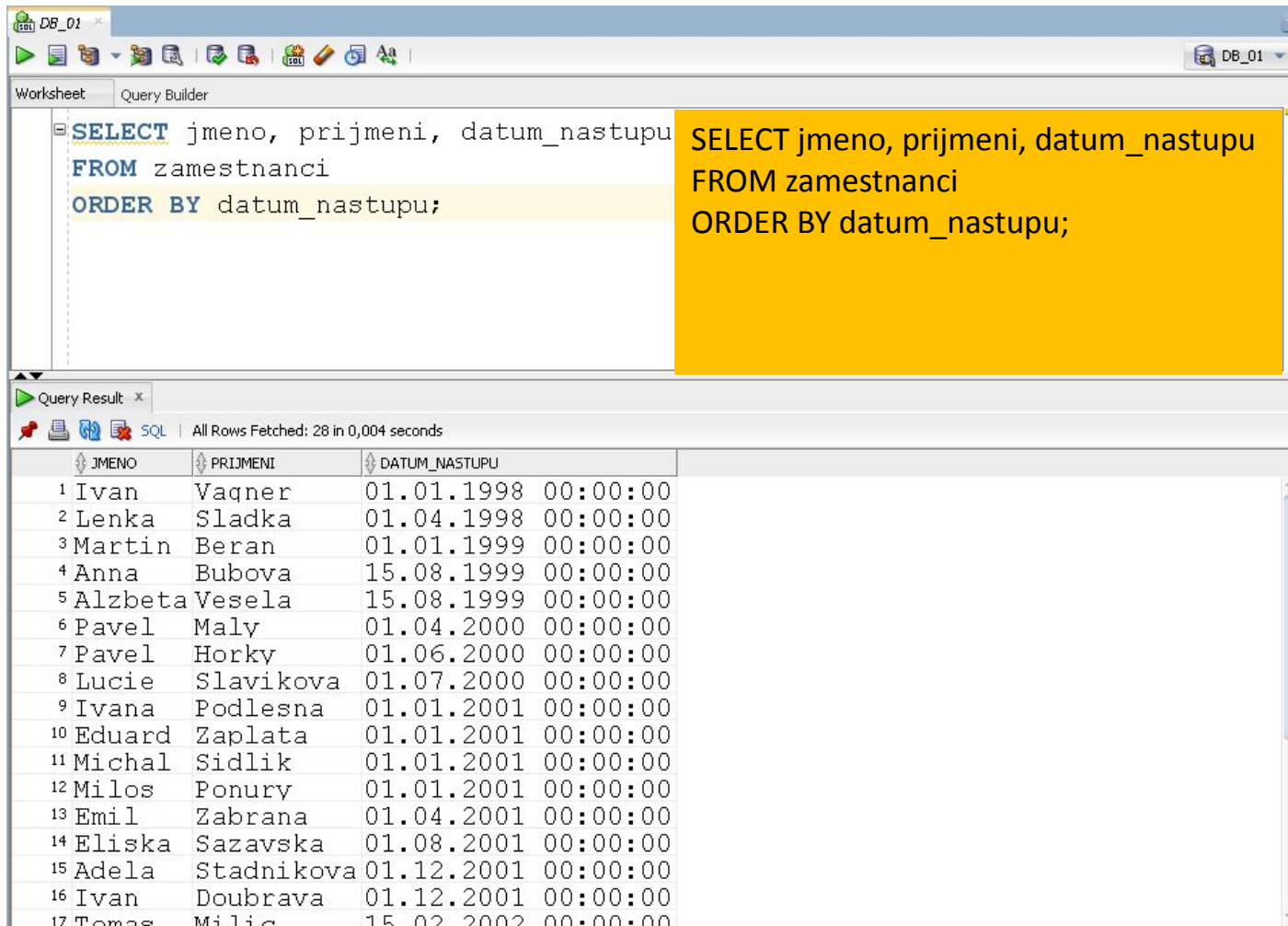
AVG(ZAKLADNI_PLAT)
1 7750





# Cvičení (pokračování)

15. Napište seznam zaměstnanců seřazený podle toho, jak dlouho ve firmě pracují.

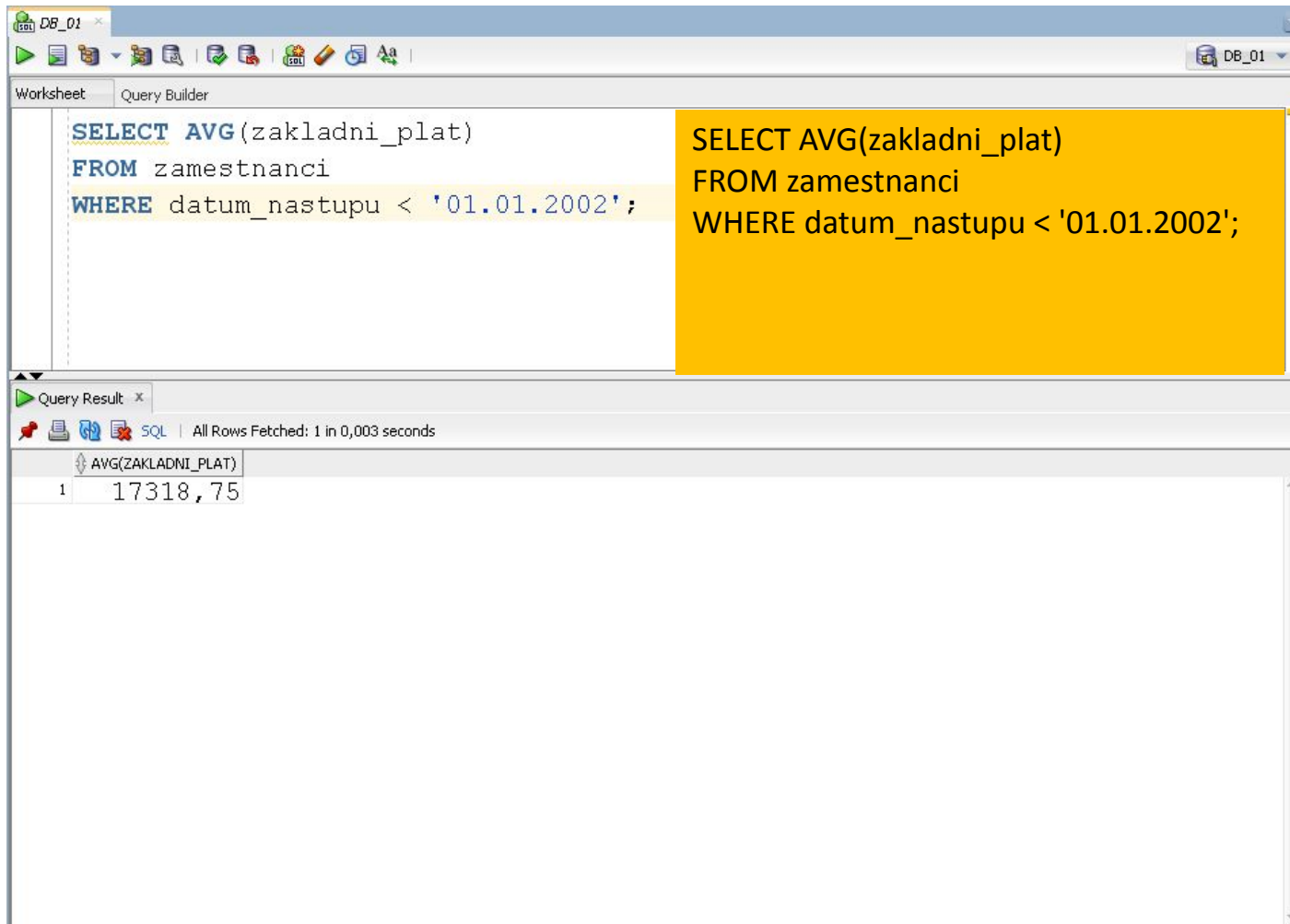


The screenshot shows a database query builder interface. The top part displays the SQL query: `SELECT jmeno, prijmeni, datum_nastupu FROM zamestnanci ORDER BY datum_nastupu;`. Below the query, the results are shown in a table with columns for name, last name, and start date. The results are sorted by start date in ascending order.

	JMENO	PRIJMENI	DATUM_NASTUPU
1	Ivan	Vagner	01.01.1998 00:00:00
2	Lenka	Sladka	01.04.1998 00:00:00
3	Martin	Beran	01.01.1999 00:00:00
4	Anna	Bubova	15.08.1999 00:00:00
5	Alzbeta	Vesela	15.08.1999 00:00:00
6	Pavel	Maly	01.04.2000 00:00:00
7	Pavel	Horky	01.06.2000 00:00:00
8	Lucie	Slavikova	01.07.2000 00:00:00
9	Ivana	Podlesna	01.01.2001 00:00:00
10	Eduard	Zaplata	01.01.2001 00:00:00
11	Michal	Sidlik	01.01.2001 00:00:00
12	Milos	Ponury	01.01.2001 00:00:00
13	Emil	Zabrana	01.04.2001 00:00:00
14	Eliska	Sazavska	01.08.2001 00:00:00
15	Adela	Stadnikova	01.12.2001 00:00:00
16	Ivan	Doubrava	01.12.2001 00:00:00
17	Tomas	Milic	15.02.2002 00:00:00

# Cvičení (pokračování)

16. Jaký je průměrný plat zaměstnanců, kteří nastoupili před rokem 2002?



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 AVG(zakladni_plat)
FROM zamestnanci
WHERE datum_nastupu < '01.01.2002';
```

A yellow highlight box on the right side of the query editor contains the same query text.

The bottom window, titled 'Query Result - x', shows the execution status: 'All Rows Fetched: 1 in 0,003 seconds'. Below this, a table displays the result of the query:

	AVG(ZAKLADNI_PLAT)
1	17318,75