

PRÁCA S ČASOM, KURZORY, PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

Základy databázových systémov

(PREDNÁŠKA 10)



PRÁCA S ČASOM



DÁTOVÉ TYPY

DATE

TIMESTAMP(n)

TIMESTAMP(n) WITH TIME ZONE

TIMESTAMP(n) WITH LOCAL TIME ZONE

INTERVAL YEAR(n) TO MONTH

INTERVAL DAY(n1) TO SECOND(n2)



AKTUÁLNY DÁTUM / ČAS

```
select sysdate  
from dual;
```

```
select current_date  
from dual;
```

```
select systimestamp  
from dual;
```

```
select current_timestamp  
from dual;
```



AKTUÁLNY DÁTUM / ČAS

select systimestamp from dual;

14-APR-15 05.06.53.951935 PM +02:00

Select localtimestamp from dual;

14-APR-15 05.06.53.951939 PM



ROZDIEL DÁTUMOV

Create table datum_tab(T1 date, T2 date);

Insert into datum_tab values(sysdate, sysdate-1);

Select T1-T2

from datum_tab;

→ 1

ROZDIEL DÁTUMOV VRÁTI POČET DNÍ!



AUTOMATICKÁ KONVERZIA (1)

Create table TAB1 (datum **date);**

Insert into TAB1 values (sysdate**);**

**Select to_char(datum, 'DD-MM-YYYY HH:MI:SS')
from TAB1;**

→ 14-04-2015 11:06:31



Insert into TAB1 values (systimestamp**);**

**Select to_char(datum, 'DD-MM-YYYY HH:MI:SS')
from TAB1;**

→ 14-04-2015 11:09:14



AUTOMATICKÁ KONVERZIA (2)

Create table TAB2 (datum **timestamp);**

Insert into TAB2 values (sysdate**);**

**Select to_char(datum, 'DD-MM-YYYY HH:MI:SS')
from TAB2;**

→ 14-04-2015 11:06:31



Insert into TAB2 values (systimestamp**);**

**Select to_char(datum, 'DD-MM-YYYY HH:MI:SS')
from TAB2;**

→ 14-04-2015 11:09:14



AUTOMATICKÁ KONVERZIA (3)

Create table TAB2 (datum **timestamp);**

Insert into TAB2 values (sysdate**);**

Insert into TAB2 values (systimestamp**);**

**Select datum
from TAB2;**

→14-APR-15 11.16.58.000000** AM --**sysdate****

→14-APR-15 11.17.06.208291** AM --**timestamp****

DÁTUMOVÉ KONVERZIE

TO_CHAR(value, [date_format], [nls_language])

Select **to_char**(sysdate, DATE_FORMAT)
from dual;

Select **to_date**('14.4.2015', DATE_FORMAT)
from dual;

VÝPIS DÁTUMU

**Select to_char(sysdate, DATE_FORMAT)
from dual;**

Skratka	Význam
Dátum	
DD	Deň v mesiaci (01 až 31)
Day	Deň v týždni, prvé písmeno je veľké (napr. Saturday)
MM	Mesiac (01 až 12)
Month	Názov mesiaca, prvé písmeno je veľké
MON	Prvé tri písmena z názvu mesiaca veľkými písmenami
YY	Rok (00 až 99)
YYYY	Rok (vrátane storočia, napr. 1999, alebo 1901)
RR	Rok (00 až 99)
RRRR	Rok (vrátane storočí, napr. 1999, alebo 2001)
Čas	
HH	Hodina (01 až 12)
HH24	Hodina (00 až 23)
MI	Minúta (00 až 60)
SS	Sekunda (00 až 59)

http://docs.oracle.com/cd/B28359_01/olap.111/b28126/dml_commands_1029.htm#CACFIJDC

NASTAVENIE FORMÁTU SESSION

Select sysdate from dual;

→ 14-APR-15

alter session

set nls_date_format='DD.MM.YYYY HH:MI:SS';

Select sysdate from dual;

→ 14.04.2015 11:25:36

ZOZNAM OSÔB, KTORÉ MAJÚ DNES NARODENINY

```
create or replace function RCtoDatum(rc varchar2) -- VARCHAR BEZ DÍŽKY!!!
```

```
return date
```

```
is
```

```
pom varchar2(10);
```

```
begin
```

```
pom:=substr(rc,5,2)||'-';
```

-- DEŇ

```
if (substr(rc,3,1)=5 OR substr(rc,3,1)=0) THEN
```

```
    pom:=pom||'0'||substr(rc,4,1);
```

```
else
```

```
    pom:=pom||'1'||substr(rc,4,1);
```

```
end if;
```

-- MESIAC, pozor na ženy!

```
    pom:=pom||'-19'||substr(rc,1,2);
```

-- ROK

```
return to_date(pom, 'dd-mm-yyyy');
```

-- AK RODNÉ ČÍSLO NEMÁ KOREKTNÝ FORMÁT (RODNÉ ČÍSLO NIE JE MOŽNÉ

-- TRASFORMATOVАŤ NA DÁTUM NARODENIA), NASTANE VÝNIMKA

```
EXCEPTION WHEN OTHERS THEN return to_date('01-01-0001', 'dd-mm-yyyy');
```

```
end;
```

```
/
```

ZOZNAM OSÔB, KTORÉ MAJÚ DNES NARODENINY

Select meno, priezvisko

from os_udaje

where RCtoDatum(rod_cislo)=sysdate;

Je to ok?

PRÁCA S DÁTUMOM

Create table TAB1 (datum date);

Insert into TAB1 values(sysdate);

**Select count(*) from TAB1
where datum=sysdate;**

Aký bude výsledok?



PRÁCA S DÁTUMOM

Create table TAB1 (datum date);

Insert into TAB1 values(sysdate);

**Select count(*) from TAB1
where datum=sysdate;**

Aký bude výsledok?

COUNT(*)

0

PREČO?



PRÁCA S DÁTUMOM

Create table TAB1 (datum date);

Insert into TAB1 values(sysdate);

**Select count(*) from TAB1
where datum=sysdate;**

Aký bude výsledok?

COUNT(*)

0

PREČO?

POROVNÁVA TO AJ ČAS!!!

PRÁCA S DÁTUMOM

Create table TAB1(datum date);

Insert into TAB1 values(sysdate);

--RIEŠENIE POMOCOU TO_CHAR

Select count(*)

from TAB1

where to_char(datum, 'DD.MM.YYYY') = to_char(sysdate, 'DD.MM.YYYY');

Aký bude výsledok?

COUNT(*)

1

ZOZNAM OSÔB, KTORÉ MAJÚ DNES NARODENINY

Select meno, priezvisko

from os_udaje

where

to_char(RCToDatum(rod_cislo), 'DD.MM.YYYY')

=to_char(sysdate, 'DD.MM.YYYY');

Je to ok?



ZOZNAM OSÔB, KTORÉ MAJÚ DNES NARODENINY

Select meno, priezvisko

from os_udaje

where

to_char(RCToDatum(rod_cislo), 'DD.MM.YYYY')

=to_char(sysdate, 'DD.MM.YYYY');

**Je to ok? Vypíšte to zoznam ľudí,
KTORÍ SA DNES NARODILI!**



ZOZNAM OSÔB, KTORÉ MAJÚ DNES NARODENINY

Select meno, priezvisko

from os_udaje

where

```
to_char(RCToDatum(rod_cislo), 'DD.MM')  
    =to_char(sysdate, 'DD.MM');
```

MUSÍ SA ROVNAŤ DEŇ A MESIAC (rok nie)!



ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU TENTO MESIAC (V POSLEDNÝCH 30-TICH DŇOCH)

Select meno, priezvisko

```
from os_udaje o, student s, zap_predmety zp
where o.rod_cislo=s.rod_cislo
and s.os_cislo=zp.os_cislo
and to_char(datum_sk, 'MM')
=to_char(sysdate, 'MM');
```

Je to ok?



ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU TENTO MESIAC

MENO **PRIEZVISKO**

Branislav **Balaz**

Zdeno **Svetkovsky**

Lubos **Lehotsky**

Alojz **Gazo**

Zoltan **Sim**

Zdenko **Olzbut**

Peter **Stelbasky**

Peter **Stelbasky**

Peter **Stelbasky**

Peter **Stelbasky**

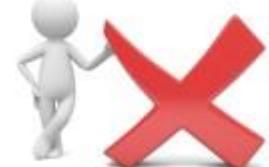
Peter **Minarik**

11 rows selected.

ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU TENTO MESIAC

```
Select meno, priezvisko, datum_sk
from os_udaje o, student s, zap_predmety zp
where o.rod_cislo=s.rod_cislo
      and s.os_cislo=zp.os_cislo
      and to_char(datum_sk, 'MM')
            =to_char(sysdate, 'MM');
```

Je to ok?



ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU TENTO MESIAC

MENO	PRIEZVISO	DATUM_SK
------	-----------	----------

Branislav	Balaz	12-APR-06
-----------	-------	-----------

Zdeno	Svetkovsky	30-APR-08
-------	------------	-----------

Lubos	Lehotsky	11-APR-04
-------	----------	-----------

Alojz	Gazo	25-APR-06
-------	------	-----------

Zoltan	Sim	30-APR-07
--------	-----	-----------

Zdenko	Olzbut	02-APR-07
--------	--------	-----------

Peter	Stelbasky	12-APR-06
-------	-----------	-----------

Peter	Stelbasky	12-APR-06
-------	-----------	-----------

Peter	Stelbasky	12-APR-06
-------	-----------	-----------

Peter	Stelbasky	12-APR-06
-------	-----------	-----------

Peter	Minarik	17-APR-06
-------	---------	-----------

11 rows selected.



ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU TENTO MESIAC

Select meno, priezvisko

```
from os_udaje o, student s, zap_predmety zp
where o.rod_cislo=s.rod_cislo
and s.os_cislo=zp.os_cislo
and to_char(datum_sk, 'MMYYYY')
=to_char(sysdate, 'MMYYYY');
```

Je to ok?



ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU MINULÝ MESIAC

Select meno, priezvisko

```
from os_udaje o, student s, zap_predmety zp
where o.rod_cislo=s.rod_cislo
and s.os_cislo=zp.os_cislo
and to_char(datum_sk, 'MM')      --minulý
=to_char(sysdate, 'MM')-1        --mesiac
and to_char(datum_sk, 'YYYY')
=to_char(sysdate, 'YYYY');       --rok
```

Je to ok?



ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU MINULÝ MESIAC

Nech sysdate = 10.1.2015

MENO	PRIEZVISKO	DATUM_SK
------	------------	----------

Branislav	Balaz	08-JAN-15	
-----------	-------	-----------	---

Zdeno	Svetkovsky	08-JAN-14	
-------	------------	-----------	---

Lubos	Lehotsky	20-DEC-14	
-------	----------	-----------	--

...predsa sa to nevypíše!

ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU MINULÝ MESIAC

Select meno, priezvisko

from os_udaje o, student s, zap_predmety zp

where o.rod_cislo=s.rod_cislo

and s.os_cislo=zp.os_cislo

and sysdate-datum_sk between 30 and 60;



FUNKCIE PRE PRÁCU S DÁTUMOM

Select sysdate from dual;

→14-APR-15

ADD_MONTHS(sysdate, pocet);

Select add_months(sysdate, 1) from dual;

→14-MAY-15

Select add_months(sysdate, -1) from dual;

→14-MAR-15

FUNKCIE PRE PRÁCU S DÁTUMOM

Select sysdate from dual;

→14-APR-15

SS:MI:HH DD:MM:YYYY



TRUNC(sysdate, DATE_FORMAT);

Select trunc(sysdate) from dual;

Select trunc(sysdate, 'DD') from dual;

→14-APR-15 (*oreže s presnosťou na deň*)

Select trunc(sysdate, 'MM') from dual;

→1-APR-15 (*oreže s presnosťou na mesiac*)

Select trunc(sysdate, 'YY') from dual;

→1-JAN-15 (*oreže s presnosťou na rok*)

FUNKCIE PRE PRÁCU S DÁTUMOM

Select sysdate from dual;

→14-APR-15

LAST_DAY(sysdate);

select LAST_DAY(sysdate) from dual;

→30-APR-15 11:14:57

(zmení sa len deň, zvyšok (MM, YY, HH, MI, SS) zostáva zachovaný)

TRUNC(ADD_MONTHS(sysdate,1), 'MM')-1

select TRUNC(ADD_MONTHS(sysdate,1), 'MM')-1 from dual;

→30-APR-15 12:00:00

ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU MINULÝ MESIAC

Select meno, priezvisko

from os_udaje o, student s, zap_predmety zp

where o.rod_cislo=s.rod_cislo

and s.os_cislo=zp.os_cislo

and **datum_sk**

BETWEEN TRUNC(ADD_MONTHS(sysdate, -1), 'MM')

AND LAST_DAY(ADD_MONTHS(sysdate, -1));

... ideálne riešenie (1)



FUNKCIE PRE PRÁCU S DÁTUMOM

Select sysdate from dual;

→14-APR-15

MONTHS_BETWEEN(sysdate, sysdate-100);

select MONTHS_BETWEEN(sysdate, sysdate-100) from dual;

→ 3.32258065

ZOZNAM OSÔB, KTORÉ SPRAVILI SKÚŠKU MINULÝ MESIAC

Select meno, priezvisko

from os_udaje o, student s, zap_predmety zp

where o.rod_cislo=s.rod_cislo

and s.os_cislo=zp.os_cislo

and MONTHS_BETWEEN(sysdate, datum_sk)

BETWEEN 0 AND 1;

... ideálne riešenie



DEŇ V TÝŽDNI

alter session set nls_date_language='English';

Select to_char(sysdate, 'DAY')
from dual;

→ SATURDAY

alter session set nls_date_language='Slovak';

Select to_char(sysdate, 'DAY')
from dual;

→ SOBOTA

Defaultné nastavenie pre asterix: American



VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

create or replace function VratNajblizsiuNedelu return date
is

 den varchar2(10);

begin

 for i IN 0..6

 LOOP

 select to_char(sysdate+i, 'DAY') into den
 from dual;

 if den like 'SUNDAY%' THEN

 return sysdate+i;

 end if;

 END LOOP;

 return null;

end;

/



VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

create or replace function VratNajblizsiuNedelu return date

is

 den varchar2(10);

begin

 select to_char(sysdate, 'DAY') into den

 from dual;

 case trim(den)

 when 'MONDAY' then return sysdate+6;

 when 'TUESDAY' then return sysdate+5;

 when 'WEDNESDAY' then return sysdate+4;

 when 'THURSDAY' then return sysdate+3;

 when 'FRIDAY' then return sysdate+2;

 when 'SATURDAY' then return sysdate+1;

 when 'SUNDAY' then return sysdate+7;

 else return null;

 end case;

end;

/



VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

create or replace function VratNajblizsiuNedelu return date

is

 den varchar2(10);

begin

 select to_char(sysdate, 'DAY') into den

 from dual;

 case trim(den)

 when 'MONDAY' then return sysdate+6;

 when 'TUESDAY' then return sysdate+5;

 when 'WEDNESDAY' then return sysdate+4;

 when 'THURSDAY' then return sysdate+3;

 when 'FRIDAY' then return sysdate+2;

 when 'SATURDAY' then return sysdate+1;

 when 'SUNDAY' then return sysdate+7;

 else return null;

 end case;

end;

/

--MUSÍME VEDIŤ NLS_DATE_LANGUAGE



VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

```
alter session set nls_date_language='Slovak';
select VratNajblizsiuNedelu from dual;
→ NULL
```

VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

create or replace function VratNajblizsiuNedelu return date

is

 den varchar2(10);

begin

 select to_char(sysdate, 'DAY', 'nls_date_language=American') into den

 from dual;

 case trim(den)

 when 'MONDAY' then return sysdate+6;

 when 'TUESDAY' then return sysdate+5;

 when 'WEDNESDAY' then return sysdate+4;

 when 'THURSDAY' then return sysdate+3;

 when 'FRIDAY' then return sysdate+2;

 when 'SATURDAY' then return sysdate+1;

 when 'SUNDAY' then return sysdate+7;

 else return null;

 end case;

end;

/



DEŇ V TÝŽDNI, MESIACI A ROKU

Select **to_char(sysdate, 'D')** --VRÁTI PORADIE

from dual;



→ 7 - sobota

1 - nedea

2 - pondelok

3 - utorok

4 - streda

5 - štvrtok

6 - piatok

7 - sobota

to_char(sysdate, 'D')

-- deň v týždni

to_char(sysdate, 'DD')

-- deň v mesiaci

to_char(sysdate, 'DDD')

--deň v roku

DEŇ V TÝŽDNI, MESIACI A ROKU

Select `to_char(sysdate, 'D')` --VRÁTI PORADIE

from dual;



→ 7 – nedel'a

7 - nedel'a

1 - pondelok

2 - utorok

3 - streda

4 - štvrtok

5 - piatok

6 - sobota

`to_char(sysdate, 'D')`

-- deň v týždni

`to_char(sysdate, 'DD')`

-- deň v mesiaci

`to_char(sysdate, 'DDD')`

--deň v roku

VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

create or replace function VratNajblizsiuNedelu return date

is

 den integer;

begin

 for i IN 0..6

 LOOP

 select to_char(sysdate+i, 'D') into den
 from dual;

 if den = 1 THEN

 return sysdate+i;

 end if;

 END LOOP;

 return null;

end

;

/

nedeľa alebo pondelok?



NLS_TERRITORY

Alter session set nls_territory='Slovakia';

→ 1 – pondelok

Alter session set nls_territory= 'America';

→ 1 – nedel'a

TO_CHAR(value, [format_mask], [nls_language])

nie je možné dať nls_territory ako parameter!



RIEŠENIE

select

```
decode(trim(to_char(sysdate,
                     'Day', 'nls_date_language=American')),
      'Monday', 1,
      'Tuesday', 2,
      'Wednesday', 3,
      'Thursday', 4,
      'Friday', 5,
      'Saturday', 6,
      'Sunday', 7)
```

from dual;



NLS_SESSION_PARAMETERS

Konfigurácia session:

Desc **nls_session_parameters**

- parameter
- value



NLS_SESSION_PARAMETERS

Select * from nls_session_parameters;

- NLS_LANGUAGE= 'AMERICAN';
- NLS_TERRITORY= 'AMERICA';
- NLS_CURRENCY= '\$';
- NLS_ISO_CURRENCY= 'AMERICA';
- NLS_NUMERIC_CHARACTERS= '.',';
- NLS_CALENDAR= 'GREGORIAN';
- NLS_DATE_FORMAT= 'DD-MON-RR';
- NLS_DATE_LANGUAGE= 'AMERICAN';
- NLS_SORT= 'BINARY';
- NLS_TIME_FORMAT= 'HH.MI.SSXFF AM';
- NLS_TIMESTAMP_FORMAT= 'DD-MON-RR HH.MI.SSXFF AM';
- NLS_TIME_TZ_FORMAT= 'HH.MI.SSXFF AM TZR';
- NLS_TIMESTAMP_TZ_FORMAT= 'DD-MON-RR HH.MI.SSXFF AM TZR';
- NLS_DUAL_CURRENCY= '\$';
- NLS_COMP= 'BINARY';
- NLS_LENGTH_SEMANTICS= 'BYTE';
- NLS_NCHAR_CONV_EXCP= 'FALSE';



VYPÍŠTE ZOZNAM ĽUDÍ, KTORÍ MAJÚ NARODENINY NAJBLIŽŠIU NEDEĽU

next_day(sysdate, param)

```
Alter session set nls_date_language='American';
Select next_day(sysdate, 'SUNDAY')
from dual;
```

```
Alter session set nls_territory='America';
Select next_day(sysdate, 1)
from dual;
```



DRUHÁ NEDEL'A V MESIACI

Ako na to?

Predpoklad: nls_territory='America'



april 2015						
po	ut	st	št	pi	so	ne
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10



DRUHÁ NEDEĽA V MESIACI

- **Zistiť 1.deň mesiaca**
 - Aký je to deň
 - Nedel'a → + 7 dní
 - Nie je to nedel'a
 - Nájdi najbližšiu nedelu → + 7 dní

DRUHÁ NEDEL'A V MESIACI

select case

```
when to_char(zaciatok, 'D') = 1 then --je to nedela  
    to_char(zaciatok+7, 'DD.MM.YYYY')  
else --prvý deň mesiaca nie je nedela  
    to_char(next_day(zaciatok, 1)+7, 'DD.MM.YYYY')  
end as druh_a_nedela
```

from

```
(select trunc(sysdate, 'MM') as zaciatok from dual);
```

DRUHÁ NEDEĽA V MESIACI

Premyslite si...

**Ako vyriešiť problém, ak nevieme hodnotu parametra
NLS_TERRITORY a ani ju nemôžeme zmeniť?**



MESIAC V ROKU, KVARTÁL

Select to_char(sysdate, 'W')

from dual; -- vráti porad. číslo týždňa v rámci mesiaca

Select to_char(sysdate, 'WW')

from dual; -- vráti porad. číslo týždňa v rámci roka

Select to_char(sysdate, ' Q')

from dual; -- vráti kvartál (jan – mar → 1, apr – jun →2, ...)

EXTRACT

Select sysdate from dual; → 14.4.2015

**Extract (DAY, MONTH, YEAR, HOUR, MINUTE, SECOND, TIMEZONE_HOUR,
TIMEZONE_MINUTE, TIMEZONE_REGION, TIMEZONE_ABBREVIATION)**

Select extract (**day** from sysdate) from dual;

→ 14

Select extract (**month** from sysdate) from dual;

→ 4

Select extract (**year** from sysdate) from dual;

→ 2015



EXTRACT

**Extract (DAY, MONTH, YEAR, HOUR, MINUTE, SECOND, TIMEZONE_HOUR,
TIMEZONE_MINUTE, TIMEZONE_REGION, TIMEZONE_ABBREVIATION)**

```
select extract (day from datum_sk)  
from zap_predmety;
```



RRRR VS. YYYY

Create table TAB1(datum date);

DD:MM:YYYY HH:MI:SS

DD:MM:RRRR HH:MI:SS

RR €

< 0 ; 49> → 21.storočie

<50 ; 99> → 20.storočie

insert into TAB1 values(to_date('1-1-99', 'DD-MM-RR'));

select to_char(datum, 'DD-MM-YYYY') from TAB1;

select to_char(datum, 'DD-MM-RRRR') from TAB1;

→01-01-1999

insert into TAB1 values (to_date('1-1-99', 'DD-MM-YY'));

select to_char(datum, 'DD-MM-YYYY') from TAB1;

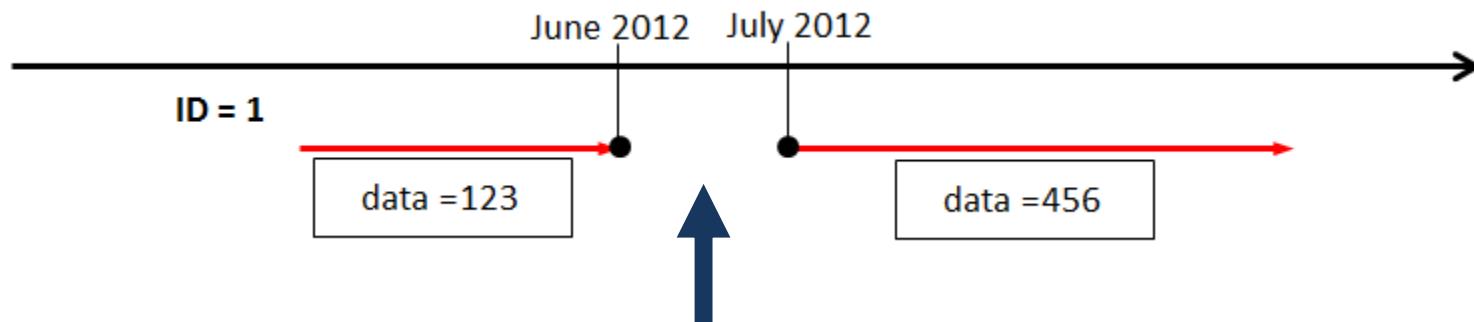
select to_char(datum, 'DD-MM-RRRR') from TAB1;

→01-01-2099 -- *vždy 21.storočie*

PRÁCA S ČASOVÝMI INTERVALMI (1)

- **Closed-closed representation**

ID	BD1	ED1	data
1	January 2012	June 2012	123
2	March 2012	November 2014	555
1	July 2012	December 2013	456

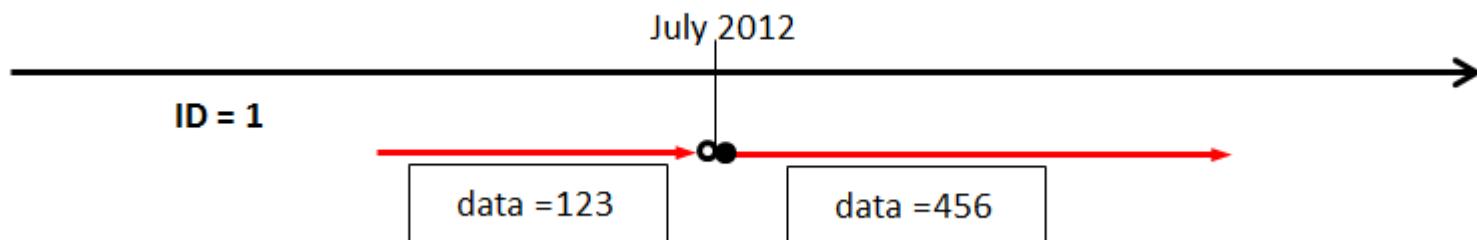


- **Pozor na zmenu granularity!**

PRÁCA S ČASOVÝMI INTERVALMI (2)

- **Closed-open representation**

<u>ID</u>	<u>BD1</u>	<u>ED1</u>	<u>data</u>
1	September 2011	July 2012	123
2	January 2012	December 2014	555
1	July 2012	January 2014	456

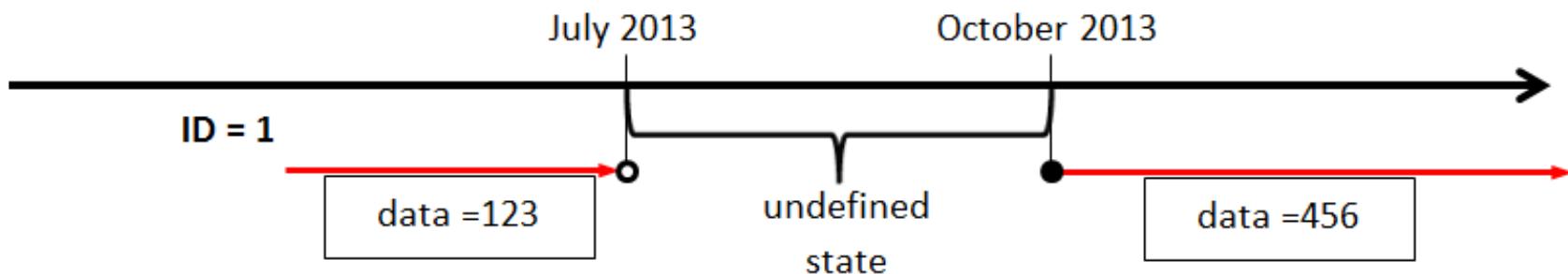


- Zmena granularity nespôsobí vznik nedefinovaných stavov

PRÁCA S ČASOVÝMI INTERVALMI (3)

- Nedefinované časové obdobia**

<u>ID</u>	<u>BD1</u>	<u>ED1</u>	data
1	September 2012	July 2013	123
2	January 2013	December 2014	555
1	October 2013	January 2014	234

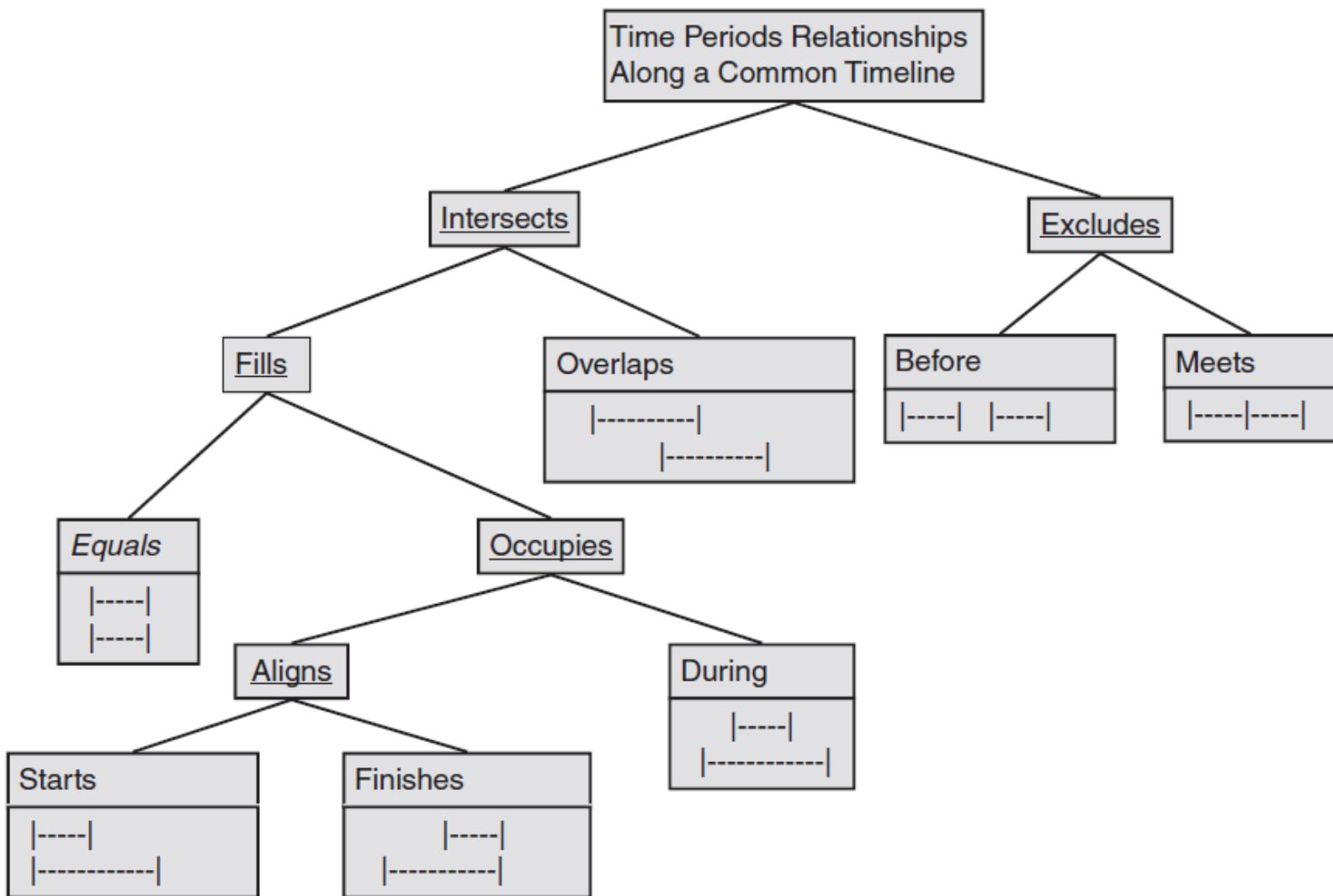


TRANSFORMÁCIA ČASOVÝCH INTERVALOV

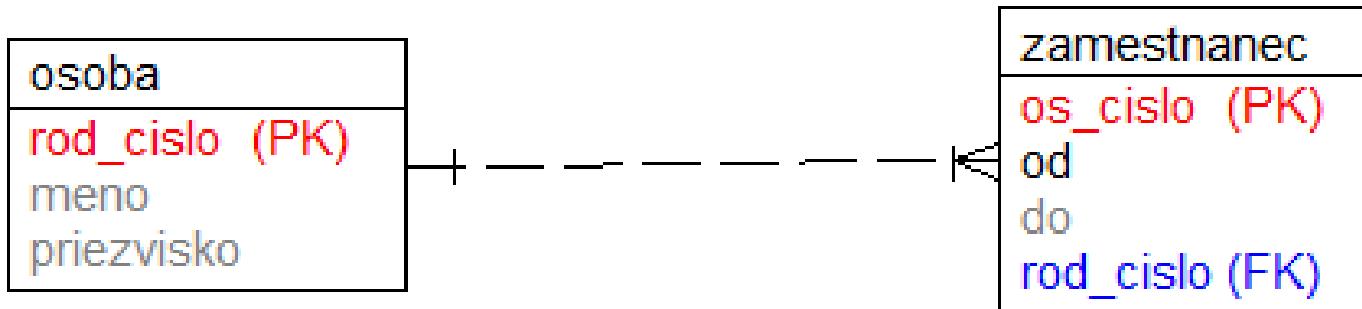
Representation	Predicate
$[a_1, a_2] \text{ equals } [b_1, b_2]$	$a_1 = b_1 \text{ AND } a_2 = b_2$
$[a_1, a_2] \text{ equals } (b_1, b_2)$	$a_1 = b_1 \text{ AND } a_2 + 1 = b_2$
$[a_1, a_2] \text{ equals } (b_1, b_i)$	$a_1 = b_1 + 1 \text{ AND } a_2 + 1 = b_1 + b_i$
$(a_1, a_2) \text{ equals } [b_1, b_2]$	$a_1 = b_1 \text{ AND } a_2 = b_2 + 1$
$(a_1, a_2) \text{ equals } (b_1, b_2)$	$a_1 = b_1 \text{ AND } a_2 = b_2$
$(a_1, a_2) \text{ equals } (b_1, b_i)$	$a_1 = b_1 + 1 \text{ AND } a_2 = b_1 + b_i$
$(a_1, a_i) \text{ equals } [a_2, b_1]$	$a_1 + 1 = a_2 \text{ AND } a_1 + a_i = b_1 + 1$
$(a_1, a_i) \text{ equals } [a_2, b_1)$	$a_1 + 1 = a_2 \text{ AND } a_1 + a_i = b_1$
$(a_1, a_{i_1}) \text{ equals } (a_2, a_{i_2})$	$a_1 = a_2 \text{ AND } a_{i_1} = a_{i_2}$



VZŤAHY MEDZI ČASOVÝMI INTERVALMI (ALLEN RELATIONSHIPS)



AKTUÁLNI ZAMESTNANCI



Select meno, priezvisko

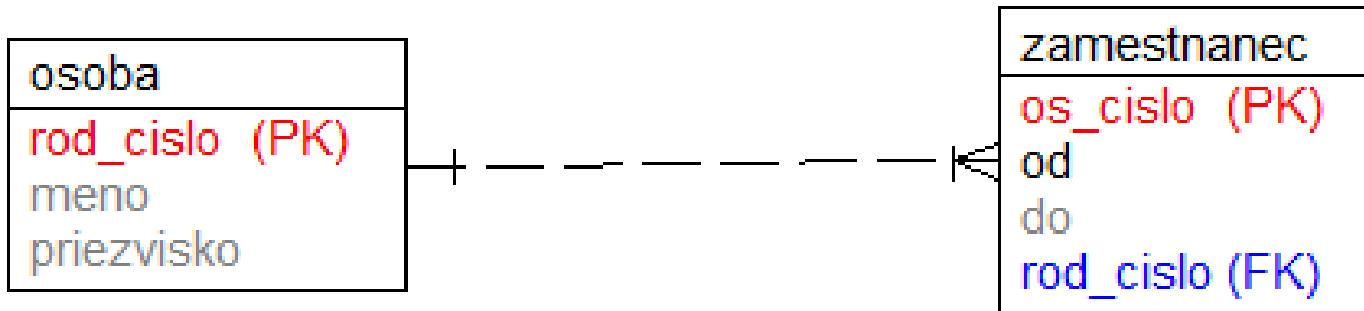
from osoba, zamestnanec

where osoba.rod_cislo=zamestnanec.rod_cislo

and od <= sysdate

and (do >= sysdate OR do IS NULL);

AKTUÁLNI ZAMESTNANCI



Select meno, priezvisko

from osoba, zamestnanec

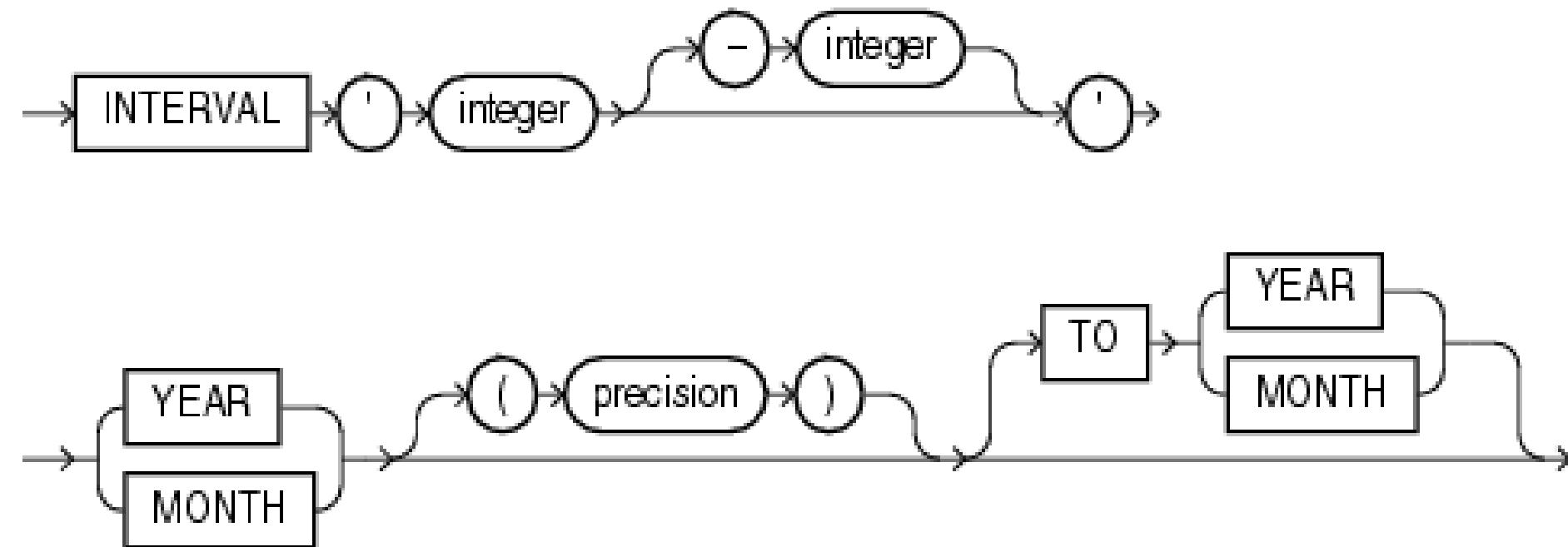
where osoba.rod_cislo=zamestnanec.rod_cislo

and od <= sysdate

and (NVL(do, sysdate)>=sysdate);

INTERVAL

INTERVAL YEAR TO MONTH



INTERVAL

Form of Interval Literal

INTERVAL '123-2' YEAR(3) TO MONTH

INTERVAL '123' YEAR(3)

INTERVAL '300' MONTH(3)

INTERVAL '4' YEAR

INTERVAL '50' MONTH

INTERVAL '123' YEAR

Interpretation

An interval of 123 years, 2 months. You must specify the leading field precision if it is greater than the default of 2 digits.

An interval of 123 years 0 months.

An interval of 300 months.

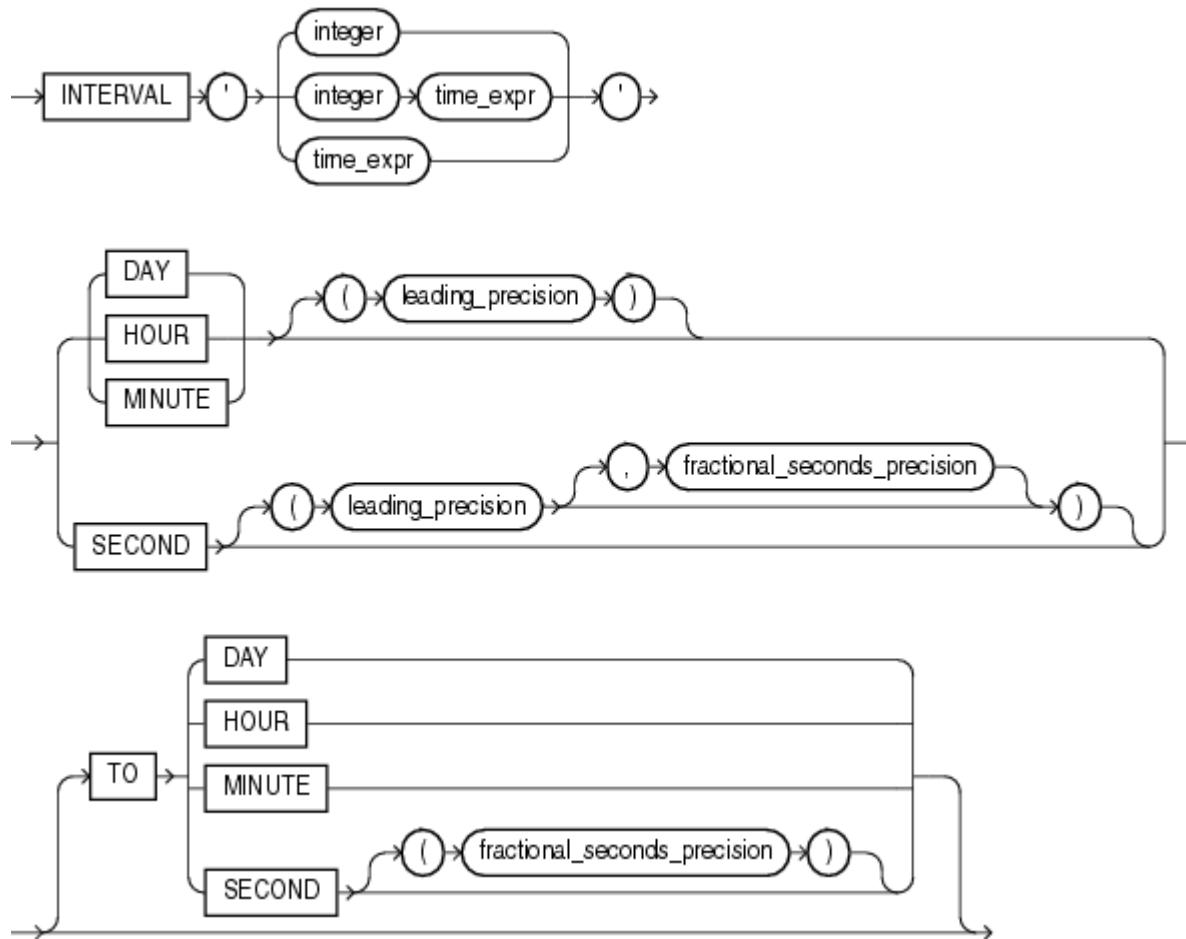
Maps to INTERVAL '4-0' YEAR TO MONTH and indicates 4 years.

Maps to INTERVAL '4-2' YEAR TO MONTH and indicates 50 months or 4 years 2 months.

Returns an error, because the default precision is 2, and '123' has 3 digits.

INTERVAL

INTERVAL DAY TO SECOND



INTERVAL

Form of Interval Literal

INTERVAL '4 5:12:10.222' DAY TO SECOND(3)

INTERVAL '4 5:12' DAY TO MINUTE

INTERVAL '400 5' DAY(3) TO HOUR

INTERVAL '400' DAY(3)

INTERVAL '11:12:10.2222222' HOUR TO
SECOND(7)

INTERVAL '11:20' HOUR TO MINUTE

INTERVAL '10' HOUR

INTERVAL '10:22' MINUTE TO SECOND

INTERVAL '10' MINUTE

INTERVAL '4' DAY

INTERVAL '25' HOUR

INTERVAL '40' MINUTE

INTERVAL '120' HOUR(3)

INTERVAL '30.12345' SECOND(2,4)

Interpretation

4 days, 5 hours, 12 minutes, 10 seconds, and 222 thousandths of a second.

4 days, 5 hours and 12 minutes.

400 days 5 hours.

400 days.

11 hours, 12 minutes, and 10.2222222 seconds.

11 hours and 20 minutes.

10 hours.

10 minutes 22 seconds.

10 minutes.

4 days.

25 hours.

40 minutes.

120 hours.

30.1235 seconds. The fractional second '12345' is rounded to '1235' because the precision is 4.

INTERVAL

**select interval '14' month
from dual;**

→ +01-02 (*jeden rok a 2 mesiacov*)

**select sysdate + interval '1-2' year(3) to month
from dual;**

→ 14.06.2016 (*k dnešnému dátumu sa
pripočíta 1 rok a 2 mesiace*)

INTERVAL

select systimestamp

+ INTERVAL '4 5:12:10.222' DAY TO SECOND(3)

from dual;



→ **INTERVAL '4 5:12:10.222' DAY TO SECOND(3)**

→ 4 dni, 5 hodín, 12 minút,
10 sekúnd a 222 tisícin.

→ **18.04.15 16:24:21,546000000 +02:00**

KURZORY



HOSTITEĽSKÁ PREMENNÁ

Definícia:

Hostiteľská premenná (HP) je premennou programu, ktorá sa deklaruje v hostiteľskom programe, na ktorú sa odkazuje integrovaný príkaz SQL.

Označenie HP v SQL príkaze

V rôznych dialektoch je označenie rôzne. Môže byť, že v príkaze SQL je meno premennej predchádza znak ":" alebo "\$", alebo sa používa meno premennej, tak ako je deklarované

Príklad :

\$OC, \$Meno

:OC, :Meno

OC, Meno

SELECT ... INTO

```
set serveroutput on
```

```
declare
```

```
    v_meno varchar2(20);
```

```
    v_priezvisko varchar2(20);
```

```
begin
```

```
select meno, priezvisko into v_meno, v_priezvisko
```

```
    from os_udaje
```

```
        where rod_cislo='841106/3456';
```

```
dbms_output.put_line(v_meno||' '||v_priezvisko);
```

```
end;
```

```
/
```

-- Select ... into musí vrátiť PRÁVE 1 RIADOK!

	MENO	PRIEZVISKO	ROD_CISLO
1	Peter	Novak	841106/3456
2	Stanislav	Steinmuller	840312/7845
3	Janos	Toth	860907/1259
4	Marek	Ratrocch	850130/3695
5	Bohuslav	Biely	841201/1248
6	Branislav	Balaz	830514/5341

SELECT ... INTO

```
set serveroutput on
```

```
declare
```

```
    v_meno varchar2(20);
```

```
    v_priezvisko varchar2(20);
```

```
v_rod_cislo char(11) := '841106/3456';
```

```
begin
```

```
select meno, priezvisko into v_meno, v_priezvisko
```

```
from os_udaje
```

```
where rod_cislo=v_rod_cislo;
```

```
dbms_output.put_line(v_meno || ' ' || v_priezvisko);
```

```
end;
```

```
/
```

	MENO	PRIEZVISKO	ROD_CISLO
1	Peter	Novak	841106/3456
2	Stanislav	Steinmuller	840312/7845
3	Janos	Toth	860907/1259
4	Marek	Ratrock	850130/3695

SELECT ... INTO

```
set serveroutput on
```

```
declare
```

```
v_meno varchar2(20);
```

```
v_priezvisko varchar2(20);
```

```
v_rod_cislo char(11);
```

```
begin
```

```
v_rod_cislo:='841106/3456';
```

```
select meno, priezvisko into v_meno, v_priezvisko
```

```
from os_udaje
```

```
where rod_cislo=v_rod_cislo;
```

```
dbms_output.put_line(v_meno||' '||v_priezvisko);
```

```
end;
```

```
/
```

	MENO	PRIEZVISKO	ROD_CISLO
1	Peter	Novak	841106/3456
2	Stanislav	Steinmuller	840312/7845
3	Janos	Toth	860907/1259
4	Marek	Ratrock	850130/3695

SELECT ... INTO

```
set serveroutput on
```

```
declare
```

```
    v_meno varchar2(20);
```

```
    v_priezvisko varchar2(20);
```

```
begin
```

```
select meno, priezvisko into v_meno, v_priezvisko
```

```
    from os_udaje
```

```
    --where rod_cislo='841106/3456';
```

```
    dbms_output.put_line(v_meno||' '||v_priezvisko);
```

```
end;
```

```
/
```

-- Select ... into musí vrátiť PRÁVE 1 RIADOK!

	MENO	PRIEZVISKO	ROD_CISLO
1	Peter	Novak	841106/3456
2	Stanislav	Steinmuller	840312/7845
3	Janos	Toth	860907/1259
4	Marek	Ratrocch	850130/3695
5	Bohuslav	Biely	841201/1248
6	Branislav	Balaz	830514/5341

SELECT ... INTO

```
set serveroutput on

declare
  v_meno varchar2(20);
  v_priezvisko varchar2(20);
begin
  select meno, priezvisko into v_meno, v_priezvisko
    from os_udaje
  where rod_cislo='841106/3456';
  dbms_output.put_line(v_meno||' '|v_priezvisko);
end;
/
```

	MENO	PRIEZVISKO	ROD_CISLO
1	Peter	Novak	841106/3456
2	Stanislav	Steinmuller	840312/7845
3	Janos	Toth	860907/1259
4	Marek	Ratrocch	850130/3695
5	Bohuslav	Biely	841201/1248
6	Branislav	Balaz	830514/5341

Error report -

ORA-01422: exact fetch returns more than requested number of rows

ORA-06512: at line 5

01422. 00000 - "exact fetch returns more than requested number of rows"

***Cause: The number specified in exact fetch is less than the rows returned.**

***Action: Rewrite the query or change number of rows requested**

HOSTITEĽSKÁ PREMENNÁ

```
declare
    v_meno varchar2(20);
    v_priezvisko varchar2(20);
    v_rod_cislo char(11);
begin
    v_meno:='Michal';
    v_priezvisko:='Kvet';
    v_rod_cislo:='123456/7890';
    insert into os_udaje(meno, priezvisko, rod_cislo)
        values(v_meno, v_priezvisko, v_rod_cislo);
end;
/
```

podobne pre Update, Delete, Select...

VÝSLEDOK

```
select *  
from os_udaje  
where priezvisko='Kvet';
```

	ROD_CISLO	MENO	PRIEZVISKO	ULICA	PSC	OBEC
1	123456/7890	Michal	Kvet	(null)	(null)	(null)

BULK COLLECT

V DBS Oracle je možné výsledky príkazu SELECT uložiť do kolekcie (nested table) a ďalej spracovať.

**Select vyraz1, vyraz2 ..., vyrazN
bulk collect into prem1, prem2, ... premN
from...**

BULK COLLECT - PRÍKLAD

Declare type t_zaznam is record(

t_meno os_udaje.meno%type,
t_priezv os_udaje.priezvisko%type,
t_rod_cislo os_udaje.rod_cislo%type);

type t_tab is table of t_zaznam;

tab t_tab;

begin

select meno, priezvisko, rod_cislo

bulk collect into tab

from os_udaje;

for i in 1 .. tab.count

loop

dbms_output.put_line(tab(i).t_rod_cislo);

end loop;

end;

/

BULK COLLECT - PRÍKLAD

```
Declare type t_zaznam is record( _____  
          t_meno os_udaje.meno%type,  
          t_priezv os_udaje.priezvisko%type,  
          t_rod_cislo os_udaje.rod_cislo%type);  
          type t_tab is table of t_zaznam;  
          tab t_tab; _____  
begin  
    select meno, priezvisko, rod_cislo  
      bulk collect into tab _____  
        from os_udaje;  
    for i in 1 .. tab.count  
      loop  
        dbms_output.put_line(tab(i).t_rod_cislo);  
      end loop;  
    end;  
  /
```

KURZOR - DEFINÍCIA

- Definícia (Cannan)
 - Kurzor je mechanizmus, ktorý umožní sprístupniť riadky tabuľky.
- Definícia (Pokorný)
 - Kurzor je objekt jazyka SQL, ktorý čísluje záznamy v množine získanej príkazom SELECT a dovoľuje aktualizovať alebo odstraňovať bežný záznam adresovaný kurzorom
- Definícia (Km)
 - **Kurzor je objekt jazyka SQL, ktorý sprístupňuje záznamy v množine získanej príkazom SELECT**
- Analógia –spracovanie kurzoru je podobné k práci so súborom

TYPY KURZOROV

- Sekvenčný kurzor – *v DBS Oracle*
- Posuvný kurzor
- Kurzor pre zmeny

SEKVENČNÝ KURZOR

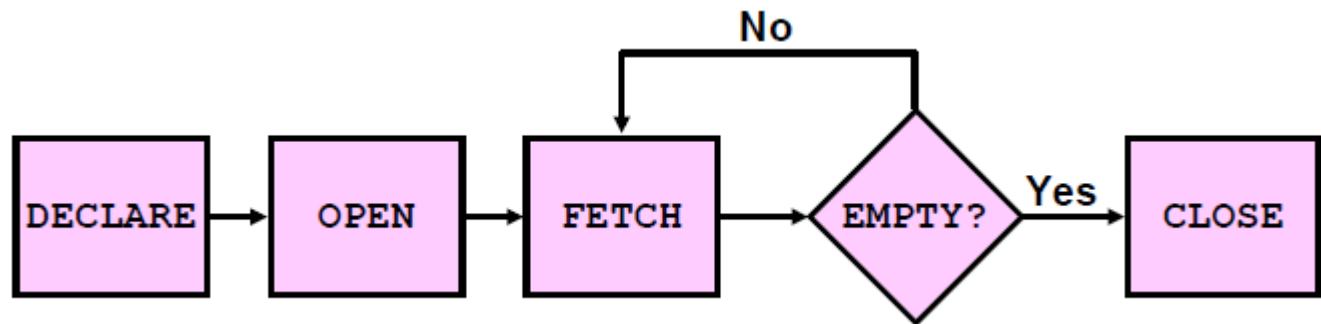
- Sekvenčný kurzor umožní spracovávať riadky sekvenčne podobne ako záznamy v sekvenčnom súbore
- Sekvenčný kurzor umožní spracovávanie záznamov len v jednom smere od začiatku do konca množiny sprístupnenej príkazom SELECT.

POSUVNÝ KURZOR

- Posuvný kurzor (SCROLL) umožňuje sa pohybovať v množine záznamov získanej príkazom SELECT.
- Posuvný kurzor umožňuje priamy prístup k vybranému záznamu z množiny záznamov, ktoré sú výsledkom príkazu SELECT. Od danej pozície je možné sprístupniť nasledovný záznam, predchádzajúci záznam alebo posun o niekoľko pozícií.
- *Poznámka* – v Oracle nie je tento druh kurzora

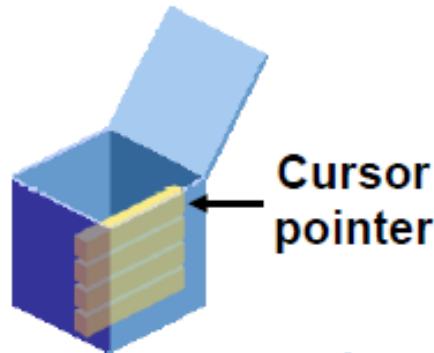
PRÍKAZY PRE PRÁCU S KURZORMI

- DECLARE
- OPEN
- FETCH
- UPDATE/DELETE
- CLOSE

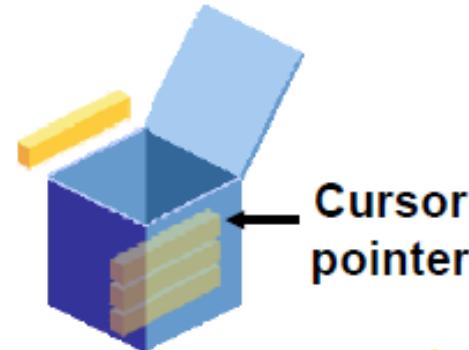


PRÍKLADY PRE PRÁCU S KURZORMI

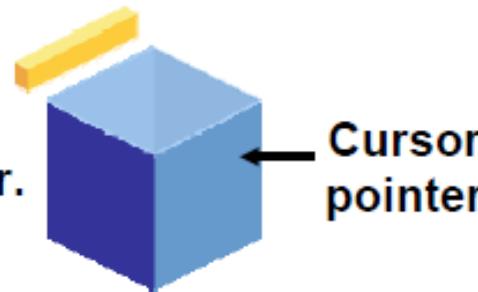
- 1 Open the cursor.



- 2 Fetch each row,
one at a time.



- 3 Close the cursor.



OTVORENIE SEKVENČNÉHO KURZORA

- Príkaz OPEN zabezpečí:
 - Syntaktickú kontrolu príkazu SELECT
 - Kontroluje prístupové práva
 - Kontrolu existencie tabuliek
 - Kontrolu existenciu mien tabuliek a stĺpcov
 - Určí aktívnu množinu
 - Alokuje pamäť pre príkaz
 - Nastaví smerník na začiatok pamäťového priestoru
- ! Príkaz OPEN nevykonáva výber riadku z tabuľky

UZATVORENIE KURZORA

- Príkaz CLOSE zabezpečí:
 - Zrušenie aktívnej množiny záznamov
 - Uvoľnenie alokovej pamäti
- Príkaz CLOSE môže byť použitý až po použití príkazov:
 - DECLARE
 - OPEN

SPRÍSTUPNENIE DÁT

- Príkaz FETCH Posunie smerník na aktuálny riadok aktívnej množiny a vykoná výber dát aktuálneho riadku z tabuľky databázy a ich umiestnenie do alokovanej pamäti

KURZORY A TRANSAKcie

- DBS zvyčajne **zamkne všetky riadky**, ktoré sprístupňuje v kurzore
- Príkaz COMMIT a ROLLBACK **uzatvárajú všetky otvorené kurzory**

ATRIBÚTY KURZORA

Atribút	Návratová hodnota
%ISOPEN	vráti TRUE ak je kurzor otvorený, FALSE ak je kurzora zavretý.
%FOUND	vráti INVALID_CURSOR ak je kurzor deklarovaný, ale nie je otvorený, alebo už je uzavretý. vráti NULL ak je otvorený, ale neboli vykonaný príkaz FETCH. vráti TRUE, ak FETCH načíta hodnoty do premenných vráti FALSE, ak neboli vrátený riadok.
%NOTFOUND	vráti INVALID_CURSOR ak je kurzor deklarovaný, ale nie je otvorený, alebo už je uzavretý. vráti NULL ak je otvorený, ale neboli vykonaný príkaz FETCH. vráti FALSE, ak FETCH načíta hodnoty do premenných. vráti TRUE, ak neboli vrátený riadok.
%ROWCOUNT	vráti INVALID_CURSOR, ak je kurzor deklarovaný, ale nie je otvorený, alebo už je uzavretý. inak vráti počet riadkov kurzora.

PRÍKLAD ORACLE

```
declare cursor cur_st is select rod_cislo, meno, priezvisko, os_cislo
    from os_udaje JOIN student using(rod_cislo);
rod_cislo varchar2(11);
meno varchar2(15);
priezvisko varchar2(15);
os_cislo integer;

begin
open cur_st;
loop
fetch cur_st into rod_cislo, meno, priezvisko, os_cislo;
exit when cur_st%notfound;
dbms_output.put_line(rpad(rod_cislo, 15) || rpad(meno, 20) || rpad(priezvisko, 20) || os_cislo);
end loop;
close cur_st;
end;
/
```

PRÍKLAD ORACLE

```
--v2
declare cursor cur_st is select rod_cislo, meno, priezvisko, os_cislo
                      from os_udaje JOIN student using(rod_cislo);
type typ is record(rod_cislo varchar2(11),
                   meno varchar2(15),
                   priezvisko varchar2(15),
                   os_cislo integer);
riadok typ;
begin
open cur_st;
loop
fetch cur_st into riadok;
exit when cur_st%notfound;
dbms_output.put_line(rpad(riadok.rod_cislo, 15) || rpad(riadok.meno, 20) || rpad(riadok.priezvisko, 20) ||
                     riadok.os_cislo);
end loop;
close cur_st;
end;
/
```

PRÍKLAD ORACLE

```
declare cursor cur_st is select rod_cislo, meno, priezvisko, os_cislo
                           from os_udaje JOIN student using(rod_cislo);
begin
for riadok in cur_st
loop
  dbms_output.put_line(rpad(riadok.rod_cislo, 15)
                       || rpad(riadok.meno, 20) || rpad(riadok.priezvisko, 20) || riadok.os_cislo);
end loop;
end;
/
```

Premenná riadok je RECORD!

Otvorenie a zatvorenie kurzoru zabezpečí systém automaticky

PRÍKLAD ORACLE

```
begin
for riadok in (select rod_cislo, meno, priezvisko, os_cislo
    from os_udaje JOIN student using(rod_cislo))
loop
    dbms_output.put_line(rpad(riadok.rod_cislo, 15)
        || rpad(riadok.meno, 20) || rpad(riadok.priezvisko, 20) || riadok.os_cislo);
end loop;
end;
/
```

PRÍKLAD 2 - ORACLE

Zoznam studentov

841106/3456: Peter Novak

 501512 100 - Informatika, 0 -

840312/7845: Stanislav Steinmuller

 501469 101 - Pocitacove inzinierstvo, 0 -

860907/1259: Janos Toth

 501414 100 - Informatika, 0 -

850130/3695: Marek Ratroch

 550545 100 - Informatika, 0 -

841201/1248: Bohuslav Biely

830514/5341: Branislav Balaz

 501567 100 - Informatika, 0 -

PRÍKLAD 2 - ORACLE

840312/7845: Stanislav Steinmuller

501469 101 - Pocitacove inzinierstvo, 0 -

860907/1259: Janos Toth

501414 100 - Informatika, 0 -

850130/3695: Marek Ratroch

550545 100 - Informatika, 0 -

841201/1248: Bohuslav Biely

830514/5341: Branislav Balaz

501567 100 - Informatika, 0 -

501319 100 - Informatika, 0 -

781015/4431: Peter Kapustny

550020 102 - Manaagment, 0 -

PRÍKLAD 2 ORACLE (1)

```
declare
cursor cur_ou IS select meno, priezvisko, rod_cislo
    from os_udaje;
v_ou cur_ou%rowtype;
begin
dbms_output.put_line('Zoznam studentov');
open cur_ou;
loop
fetch cur_ou into v_ou;
exit when cur_ou%NOTFOUND;
dbms_output.put_line(v_ou.rod_cislo ||':'|| v_ou.meno ||' '|| v_ou.priezvisko);
end loop;
close cur_ou;
end;
/
```

PRÍKLAD2 – ORACLE (2)

```
DECLARE
cursor cur_st(p_rc os_udaje.rod_cislo%type) IS
    select os_cislo, st_odbор, popis_odbору,
           st_zameranie, popis_zamerania
      from student JOIN
           st_odbory USING(st_odbор,st_zameranie)
     where rod_cislo=p_rc;
v_st cur_st%rowtype;
BEGIN
open cur_st(v_ou.rod_cislo);
loop
    fetch cur_st into v_st;
    exit when cur_st%NOTFOUND;
    dbms_output.put_line(lpad(v_st.os_cislo, 10, ' ')
    ||' '|| v_st.st_odbор ||'-'|| v_st.popis_odbору ||','|| v_st.st_zameranie ||'-'|| v_st.popis_zamerania);
end loop;
close cur_st;
END;
/
```

PRÍKLAD 2 – ORACLE (VÝSL.)

```
declare
cursor cur_ou IS select meno, priezvisko, rod_cislo
    from os_udaje;
v_ou cur_ou%rowtype;
cursor cur_st(p_rc os_udaje.rod_cislo%type) IS
    select os_cislo, st_odbор, popis_одбору,
           st_замерание, popis_замерания
    from student JOIN
         st_одбory USING(st_одбор, st_замерание)
    where rod_cislo=p_rc;
v_st cur_st%rowtype;

begin
dbms_output.put_line('Zoznam studentov');
open cur_ou;
loop
fetch cur_ou into v_ou;
exit when cur_ou%NOTFOUND;
dbms_output.put_line(v_ou.rod_cislo ||':'|| v_ou.meno ||' '|| v_ou.priezvisko);
--KURZOR 2
open cur_st(v_ou.rod_cislo);
loop
fetch cur_st into v_st;
exit when cur_st%NOTFOUND;
dbms_output.put_line(lpad(v_st.os_cislo, 10, ' ')
||' '|| v_st.st_одбор ||'-'|| v_st.popis_одбору ||', '|| v_st.st_замерание ||'-'|| v_st.popis_замерания);
end loop;
close cur_st;
end loop;
close cur_ou;
end;
/
```

KURZOR FOR UPDATE

```
declare
cursor cur1 is select rod_cislo from student
for update;
begin
for i in cur1 loop
update student
    set ukoncenie=sysdate
        where current of cur1;
end loop;
end;
/
```

EXECUTE IMMEDIATE

```
create or replace procedure zmen_vysledok
(oc  student.os_cislo%TYPE,
rok  zap_predmety.skrok%TYPE,
cp   zap_predmety.cis_predm%TYPE,
vysl zap_predmety.vysledok%TYPE )
AS
pom varchar2(200);
BEGIN
IF ( (oc IS NOT NULL ) AND (rok > 0) AND
(cp IS NOT NULL) AND (vysl IN ('1','2','3','4','A','B','C','D','E','F'))) THEN
pom:= 'UPDATE zap_predmety'|||
      ' SET vysledok = "'||| vysl||'"'|||
      ' WHERE os_cislo = '|||oc |||
      ' AND cis_predm = "'|||cp||'"'|||
      ' AND skrok = '||| rok;
EXECUTE IMMEDIATE pom;
END IF;
END;
/
```

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT



PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
    cislo1 number(3) := 10;
    cislo2 number(3);
    vysledok number(3);
begin
    vysledok := cislo1 + cislo2;
    if ( vysledok IS NOT NULL) then
        DBMS_OUTPUT.PUT_LINE(vysledok);
    else
        DBMS_OUTPUT.PUT_LINE('nedefinovana hodnota');
    end if;
end;
/
```

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
    cislo1 NUMBER(3);
    cislo2 NUMBER(3);
begin
    IF ( cislo1 = cislo2 ) THEN
        dbms_output.put_line (' su rovnač');
    ELSE
        dbms_output.put_line (' nie su rovnač');
    END IF;
end;
/
```

Nie sú rovnaké....

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

declare

```
cislo1 NUMBER(3);  
cislo2 NUMBER(3);
```

begin

```
IF ( cislo1 <> cislo2 ) THEN
```

```
    dbms_output.put_line ('nie su rovnake');
```

```
ELSE
```

```
    dbms_output.put_line ('su rovnake');
```

```
END IF;
```

end;

/

Sú rovnaké....

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

declare

```
cislo1 NUMBER(3);  
cislo2 NUMBER(3);
```

begin

```
IF ( cislo1 <> cislo2 ) THEN
```

```
    dbms_output.put_line ('nie su rovnake');
```

```
ELSE
```

```
    dbms_output.put_line ('su rovnake');
```

```
END IF;
```

end;

/

Sú rovnaké....

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

AND	TRUE	FALSE	NULL	OR	TRUE	FALSE	NULL	NOT	
TRUE	TRUE	FALSE	NULL	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	NULL	FALSE	TRUE
NULL	NULL	FALSE	NULL	NULL	TRUE	NULL	NULL	NULL	NULL

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select vysledok, count(*)      V      COUNT(*)  
from zap_predmety  
group by vysledok;  
-----  
A=1          47  
B=1,5        48  
C=2          65  
D=2,5        55  
null         269
```

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select avg(vysledok)  
from zap_predmety;
```

V	COUNT(*)	AVG(VYSLEDOK)
A=1	47	
B=1,5	48	1,798
C=2	65	
D=2,5	55	
null	269	

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select count(*)  
from zap_predmety;
```

V	COUNT(*)	COUNT(*)
A=1	47	
B=1,5	48	484
C=2	65	
D=2,5	55	
null	269	

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select count(vysledok)  
from zap_predmety;
```

V	COUNT(*)	COUNT(VYSLEDOK)
A=1	47	
B=1,5	48	
C=2	65	
D=2,5	55	
null	269	215

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select sum(vysledok) /count(*)  
from zap_predmety;
```

V	COUNT(*)	SUM(VYSLEDOK) /COUNT(*)
A=1	47	
B=1,5	48	
C=2	65	
D=2,5	55	
null	269	0,799

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select sum(vysledok) /count(vysledok)  
from zap_predmety;
```

V	COUNT(*)

A=1	47
B=1,5	48
C=2	65
D=2,5	55
null	269

SUM(VYSLEDOK) /
COUNT(VYSLEDOK)

1,798

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
select avg(nvl(vysledok,4))  
from zap_predmety;
```



V	COUNT(*)	AVG(NVL(VYSLEDOK,4))
A=1	47	-----
B=1,5	48	3,021
C=2	65	
D=2,5	55	
null	269	<i>Korektný výsledok</i>

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
pom integer;
vysl integer;
begin
pom:=1;
case pom
when 1 then vysl:=1;
when 2 then vysl:=2;
when 3 then vysl:=3;
end case;
dbms_output.put_line(vysl);
end;
/
```

1

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
pom integer;
vysl integer;
begin


pom:=1;


case pom
when 1 then vysl:=1;
when 2 then vysl:=2;
when 3 then vysl:=3;
end case;
dbms_output.put_line(vysl);
end;
/
```

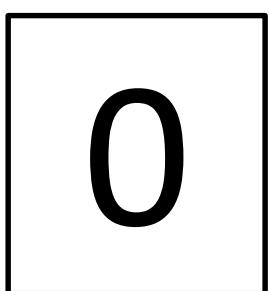
ERROR at line 1:
ORA-06592: CASE not found while executing
CASE statement
ORA-06512: at line 6

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
pom integer;
vysl integer;
begin


pom:=1;


case pom
when 1 then vysl:=1;
when 2 then vysl:=2;
else vysl:=0;
end case;
dbms_output.put_line(vysl);
end;
/
```



PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
pom integer;
vysl integer;
Begin
pom:=1;
case
when pom=1 then vysl:=1;
when pom=2 then vysl:=2;
when pom=3 then vysl:=3;
end case;
dbms_output.put_line(vysl);
end;
/
```

1

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
    pom integer;
    vysl integer;
Begin
    pom:=1;
    case
        when pom=1 then vysl:=1;
        when pom=2 then vysl:=2;
        when pom=3 then vysl:=3;
    end case;
    dbms_output.put_line(vysl);
end;
/
```

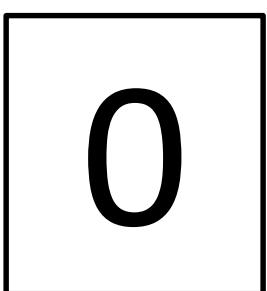
ERROR at line 1:
ORA-06592: CASE not found while executing
CASE statement
ORA-06512: at line 6

PROBLEMATIKA NEDEFINOVANÝCH HODNÔT

```
declare
pom integer;
vysl integer;
Begin


pom:=1;


case
when pom=1 then vysl:=1;
when pom=2 then vysl:=2;
when pom is null then vysl:=0;
end case;
dbms_output.put_line(vysl);
end;
/
```



ĎAKUJEM ZA POZORNOSŤ