## FÖ 11: Databaskursen

1. ***PLSQL ‐ introduktion***
2. ***Anonyma block***
3. ***Clean code och läsbarhet***
4. ***Variabeltyper och variabeldeklaration***
5. ***Olika loopar: Basic, while och for loop***

Pär Douhan, [pdo@du.se](mailto:pdo@du.se)

# PLSQL



PLSQL: en introduktion

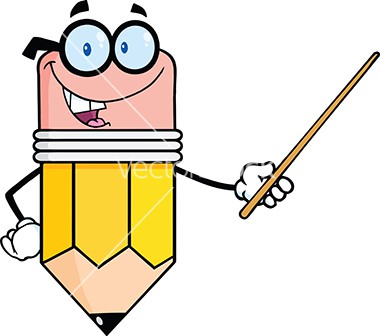
## Introduktion till Oracle PLSQL

### Vad är PL/SQL?

**PL/SQL = *"Oracle's Procedural Language extension to SQL".***

**PL/SQL‐språkets syntax, struktur och datatyper liknar ADA.**

**PL/SQL körs i en VM (Virtual Machine) precis som Java. Bägge finns i Oracle.**





**Oracle Server**

**Oracle PLSQL VM**

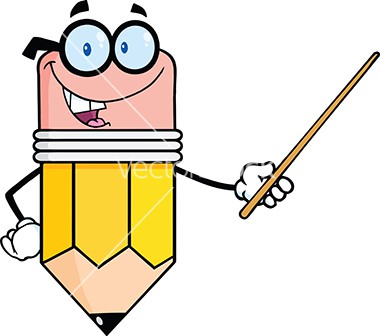
**Oracle JVM**

## Java eller PL/SQL?

### Java eller PL/SQL i Oracle?

**Använd Java när det inte går att använda PL/SQL.**

**Skicka email med bilagor?**

**‐ PL/SQL = svårt!**

**‐ Java = lätt!**

**Loopa igenom rader i en databastabell och göra något:**

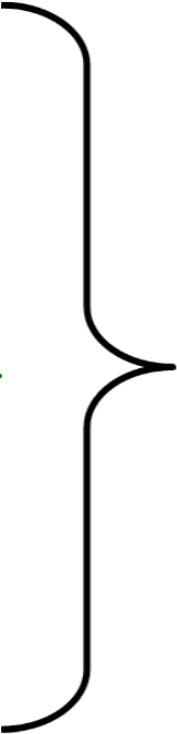
**‐ PL/SQL = lätt!**

**‐ Java = svårt!**

***"If it touch data *** ***PL/SQL" T. Kyte***

## Case sensitive

**PL/SQL är *inte case sensitive*.**

DecLAre variabeldeklareationer; BEGIN



vad programmet skall göra;

-- kommentar på en rad. Ignoreras av kompilatorn

**.**

/\* kommentarer till koden på flera rader. \*/

end;

**/**

anonymt PL/SQL-block

## Olika PL/SQL block

### Två olika typer av PL/SQL block

1. **Anonyma block**

***‐ Icke namngivna block.***

begin

dbms\_output.put\_line('Hello World!'); end;

**/**

SQL‐developer, F5 eller Run Script.

1. **Namngivna block**

***‐ Lagrade procedurer, funktioner, triggrar och paktet.***



## Clean code

***Tänk på att skriva bra PLSQL kod!***

**Clean code**

***Easy and Simple!***

* 1. ***Easy to read***
  2. ***Easy to follow***
  3. ***Easy to understand***
  4. ***Easy to modify***
  5. ***Easy to test***
  6. ***Works correctly***

#### **Att** koden ska fungera korrekt inbegriper att den är ***smart*** konstruerad och optimerad.

**Ett** exempel på smart kod är att utforma sin logik så att vi *förflyttar* så lite data som möjligt över nätverket!



***Data lives forever!***

## Läsbarhet

***Mikaela Hedbergs "rensa‐script"***

declare

cursor c\_objects is select object\_name, object\_type from user\_objects;

cursor c\_constraints is select owner, constraint\_name, constraint\_type, table\_name from user\_constraints

where constraint\_name not like upper('bin%'); v\_object\_name varchar2(100);

v\_object\_type varchar2(100); begin

for c in c\_constraints loop execute immediate

'alter table ' || c.table\_name ||

' drop constraint ' || c.constraint\_name; end loop;

for x in c\_objects loop

if upper(x.object\_type) not in ('PACKAGE BODY') then execute immediate

'drop ' || x.object\_type || ' ' || x.object\_name; end if;

end loop; end;

/

## Läsbarhet

***Med syntax highlighting***

**declare**

**cursor** c\_objects **is select object\_name,** object\_type **from** user\_objects**;**

**cursor** c\_constraints **is select** owner**,** constraint\_name**,** constraint\_type**,** table\_name

**from** user\_constraints

**where** constraint\_name **not like upper(**'bin%'**);**

v\_object\_name **varchar2(**100**);** v\_object\_type **varchar2(**100**); begin**

**for** c **in** c\_constraints **loop execute immediate**

'alter table ' **||** c**.**table\_name **||**

' drop constraint ' **||** c**.**constraint\_name**; end loop;**

**for** x **in** c\_objects **loop**

**if upper(**x**.**object\_type**) not in (**'PACKAGE BODY'**) then execute immediate**

'drop ' **||** x**.**object\_type **||** ' ' **||** x**.object\_name; end if;**

**end loop; end;**

**/**

## Läsbarhet

***Med syntax highlighting, indentering och monospaced font***

**declare**

**cursor c\_objects is select object\_name, object\_type from user\_objects;**

**cursor c\_constraints is select owner, constraint\_name, constraint\_type, table\_name from user\_constraints**

**where constraint\_name not like upper('bin%'); v\_object\_name varchar2(100);**

**v\_object\_type varchar2(100); begin**

**for c in c\_constraints loop execute immediate**

**'alter table ' || c.table\_name ||**

**' drop constraint ' || c.constraint\_name; end loop;**

**for x in c\_objects loop**

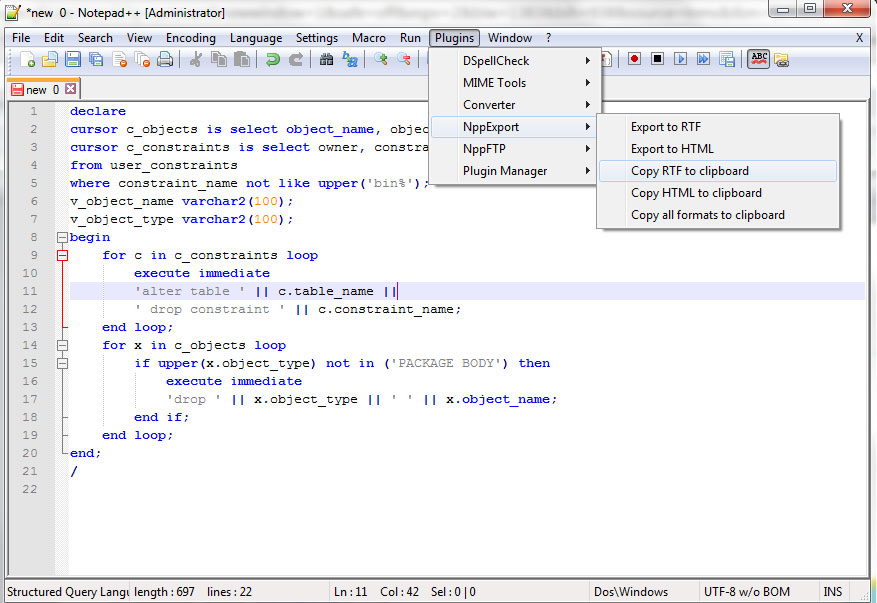
**if upper(x.object\_type) not in ('PACKAGE BODY') then execute immediate**

**'drop ' || x.object\_type || ' ' || x.object\_name; end if;**

**end loop; end;**

**/**

## Notepad ++



Klistra in RTF i labbrapporten

Välj SQL

Variabeltyper

### Variabeltyper i PL/SQL

1. **Skalär**

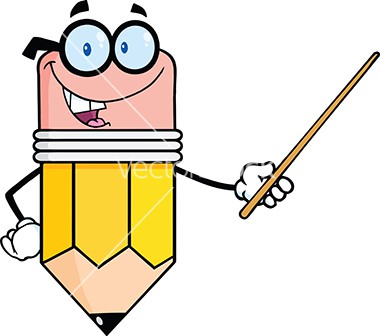
***‐ nummer, datum, teckensträngar, boolean***

1. **Komposit**

***‐ record, PL/SQL‐table, varray...***

1. **Referens**

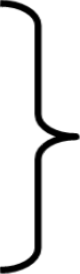
***‐ ref cursor***

1. **LOB**

***‐ BLOB***

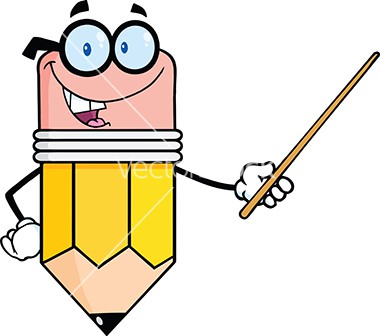
## Variabeltyper

### Variabeltyper som kommer att användas på kursen:

**number varchar2 date boolean**

**record**

**skalära datatyper**

true, false eller null

**komposit datatyp**

## Variabeldeklaration

### Variabeldeklaration och tilldelning av värden

declare

v\_namn varchar2(50);

v\_år varchar2(4) := to\_char(sysdate,'YYYY'); begin

-- tilldelar v\_namn en textsträng.

v\_namn := 'Hans Rosendahl';

dbms\_output.put\_line('Årets pedagog '|| v\_år || 'är: '|| v\_namn); end;

**/**

**:= ** **tilldelningsoperator.**

**= ** **operator för att utvärdera likhet. if a = b then -- true eller false.**

## Utskrift

**Utskrift slås på med *set serveroutput on***

**set serveroutput on**

**/\* slå på utskrift**

**med set serveroutput on \*/ declare**

**v\_namn varchar2(50);**

**v\_år varchar2(4) := to\_char(sysdate,'YYYY');**

**/\* hämtar dagens datum och gör om det till en teckensträng med fyra tecken.**

**YYYY medför att det kommer att bara visa årtal med fyra tecken \*/**

**begin**

**v\_namn := 'Hans Rosendahl';**

**dbms\_output.put\_line('Årets pedagog '|| v\_år || ' är: '|| v\_namn); end;**

*anonymous block completed*

**Årets pedagog 2020 är: Hans Rosendahl**

# PLSQL

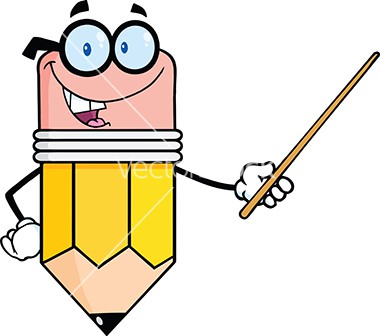


Loopar och styrstrukturer

## Aritmetiska operatorer

|  |  |
| --- | --- |
| ***Artitmetisk operator*** | ***Användning*** |
| + | Addition |
| ‐ | Subtraktion |
| \* | Multiplikation |
| / | Division |
| \*\* | Exponentiering |

**- 2 + 3 \* (10 - 2 \* 3) / 6 = x *1. Parenteser***

1. ***Potenser***
2. ***Multiplikation och division***
3. ***Subtraktion och addition***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **- 2 + 3 \* (4)** | **/** | **6** | **=** | **x** |
| **- 2 + 12** | **/** | **6** | **=** | **x** |
| **- 2** | **+** | **2** | **=** | **x** |
|  |  | **x** | **=** | **0** |

**2 \*\* 3 = 2\*2\*2 = 8**

## Styrstrukturer

***Styrstrukturer = Control Structures***

1. **if ... then ... end if;**
2. **if ... then ... else ... end if;**
3. **if ... then ... elsif ... then ... end if;**
4. **case ... when ... then ... end case;**

**OBS! ELSIF**



# CASE

**Exempel på *CASE‐sats:***

declare

v\_num number(7) := 12; v\_test number(1); begin

v\_test := mod(v\_num,2); case v\_test

when 0 then

dbms\_output.put\_line('talet är jämnt'); else

dbms\_output.put\_line('talet är udda'); end case;



end;

**/**

# IF

**Samma exempel med *IF‐sats:***

set serveroutput on declare

v\_num number := 13; v\_test number; begin

v\_test := mod(v\_num,2);

if v\_test = 0 then dbms\_output.put\_line('talet är jämnt');

else

dbms\_output.put\_line('talet är udda'); end if;



end;

**/**

# NULL?

**Testa *null‐värden* i en IF‐sats:**

set serveroutput on declare

1. **number := null;**
2. **number := null; begin**

if (a = b) then dbms\_output.put\_line('a');

elsif (a <> b) then

dbms\_output.put\_line('b'); else

dbms\_output.put\_line('c');

end if; end;

**/**

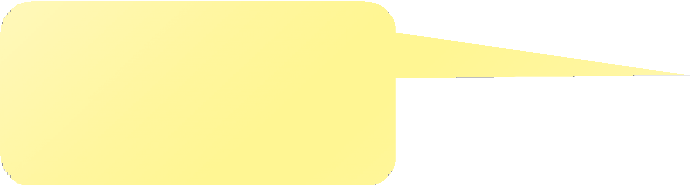
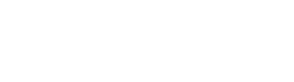
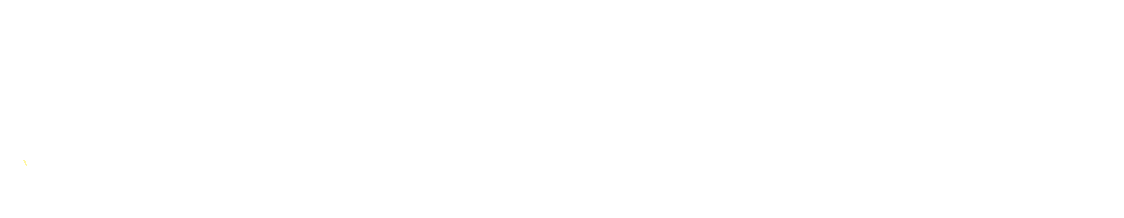
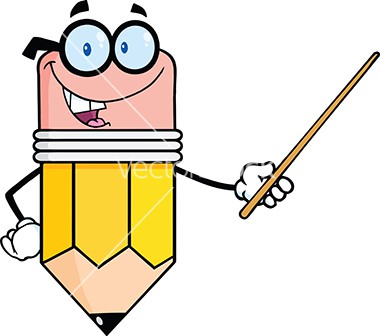
**Utskrift? c**

if ((a is null) and (b is null)) then



## Loopar

**Det finns tre olika typer av loopar i PL/SQL:**



1. **Basic loop**
2. **WHILE loop**
3. **FOR loop**

## Basic loop

**Exempel på *basic loop:***

set serveroutput on declare

i = 1

|  |  |  |  |
| --- | --- | --- | --- |
| **i number(3);** | **i** | **=** | **2** |
| **begin** | **i** | **=** | **3** |
| **i := 1;**  **loop** | **i**  **i** | **=**  **=** | **4**  **5** |

exit when i = 6; dbms\_output.put\_line('i = '||i);

i := i + 1;

end loop; end;

**/**

## While loop

**Exempel på *WHILE loop:***

set serveroutput on declare

i number(3); begin

i := 1;

while (i < 6) loop dbms\_output.put\_line('i = '||i);

i := i + 1;

end loop; end;

**/**

i = 1

i = 2

i = 3

i = 4

i = 5

## For loop

**Exempel på *FOR loop:***

set serveroutput on begin

for i in 1 .. 5 loop dbms\_output.put\_line('i = '||i);

end loop;

end;

**/**

i = 1

i = 2

i = 3

i = 4

i = 5

-- har inbyggd räknare, som vi kan kalla vad vi vill for counter in 1 .. 5 loop

dbms\_output.put\_line('i = '||counter);



## For select

**Exempel på *FOR SELECT loop:***

set serveroutput on begin

for rec in (select fnamn,enamn from student) loop dbms\_output.put\_line(rec.fnamn||' '||rec.enamn);

end loop;

end;

**/**

Bo Ek

Tommy Kvist Stina Klanth Hans Dalros Marie Ekholm



Aressa Kristossih Mgabhe Mboto Sonja Stenlund Stina Dalkvist Arne Möller

The End

