

### Übung 2

```
PL/SQL procedure successfully completed.
```

```
No errors.
```

RADIUS	AREA
3	28.27

### Übung 3

```
create table radius_values (
    radius number
);
insert into radius_values values ( 5 );
```

```
PL/SQL procedure successfully completed.
```

```
No errors.
```

RADIUS	AREA
3	28.27
5	78.54

"%rowtype" gibt einer Variable denselben Datentypen wie die Spalte

```
insert into areas (
    radius,
    area
)
select radius,
    3.1415927 * power(radius,2)
    from radius_values;
```

```
create table areas_
as
  select radius,
         3.1415927 * power(
           radius,
           2
         ) as area
    from radius_values;
```

Name	Null?	Type
RADIUS		NUMBER
AREA		NUMBER

#### Übung 4

```
insert into radius_values values ( 1 );
no rows selected
```

  

```
insert into radius_values values ( 5 );
```

RADIUS	AREA
5	78.54

#### Übung 5

```
980 3017185.63
990 3079075.01

RADIUS      AREA
-----      -----
1000      3141592.7

100 rows selected.
```

**Übung 6**

```
PL/SQL procedure successfully completed.
```

```
No errors.
```

RADIUS	AREA
3	28.27
6	113.1
10	314.16
15	706.86
21	1385.44
28	2463.01
36	4071.5

```
7 rows selected.
```

**Übung 7**

```
Table AREAS truncated.
```

```
7 153.94
6 113.1
5 78.54
4 50.27
3 28.27
2 12.57
1 3.14
```

```
PL/SQL procedure successfully completed.
```

### Übung 8

```
truncate table areas;
declare
  pi constant number(9, 7) default 3.1415927;
  area number(14, 2);
begin
  for radius_row in (
    select *
    from radius_values
  )
  loop
    area := pi * power(radius_row.radius, 2);
    insert into areas values(radius_row.radius, area);
  end loop;
  commit;
end;
/
select *
from areas;

truncate table areas;
insert into areas (radius, area)
select radius, power(radius, 2) * 3.1415927
  from radius_values;
select *
from areas;

drop table areas_;
create table areas_ as
select radius, power(radius, 2) * 3.1415927 as area
  from radius_values;
desc areas_;
```

### Übung 9a

```
begin
  l_radius := 3;
  l_area := 0;
  while l_area < 100
  loop
    l_area := l_pi * l_radius * l_radius;
    insert into areas values(l_radius, l_area);
    l_radius := l_radius + 1;
  end loop;
```

## Übung 12

```
-- Übung 12
-- Gegeben ist folgender fehlerhafter PL/SQL Programmcode:

declare
    grade integer;
begin
    grade := 3;
    case grade
        when 1 then dbms_output.put_line('Sehr gut');
        when 2 then dbms_output.put_line('Gut');
        when 3 then dbms_output.put_line('Befriedigend');
        when 4 then dbms_output.put_line('Genügend');
        when 5 then dbms_output.put_line('Nicht genügend');
    end case;
end;
||
```

Semikolon und End Case haben gefehlt.

## Übung 13

Wenn i = 8 ist gibt es einen case\_not\_found error, da plsql beim Fehlen eines else-zweig, automatisch else raise case\_not\_found hinzufügt.

```
declare
    i integer;
begin
for i in 1..8 loop
    case
        when i = 1 then dbms_output.put_line('Montag: 8-12 und 13-18 Uhr.');
        when i = 2 then dbms_output.put_line('Dienstag: 8-13 Uhr.');
        when i = 3 then dbms_output.put_line('Mittwoch: 9-12 Uhr und 14-20 Uhr.');
        when i = 4 then dbms_output.put_line('Donnerstag: 9-13 Uhr.');
        when i = 5 then dbms_output.put_line('Freitag: 10-15 Uhr.');
        when i = 6 then dbms_output.put_line('Samstag: 7-11 Uhr.');
        when i = 7 then dbms_output.put_line('Sonntag: geschlossen.');
        else dbms_output.put_line('Ungültiger Wochentag.');
    end case;
end loop;
end;
```