

Julia Ruszer 247775

Dominik Gałkowski 247659

Jan Śladowski 247806

Wiktor Żelechowski 247833

Microsoft SQL Server

Zadanie 1.

1. Przygotuj skrypt nr 1:

- Rozpocznij transakcję.
- Zadeklaruj zmienną o nazwie *salary*.
- Przypisz jej wartość pensji pracownika o identyfikatorze równym 100.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Obniż o 1000 zł wartość zmiennej *salary*.
- Ustaw pensję pracownika o identyfikatorze równym 100 na wartość zmiennej *salary*.
- Wyświetl wartość zmiennej *salary*.
- Zatwierdź transakcję.

```
begin transaction;  
declare @salary numeric(8, 2);  
set @salary = (select salary from employees where employee_id =  
100);  
WAITFOR DELAY '00:00:15';  
set @salary = @salary - 1000;  
update employees set salary = @salary where employee_id = 100;  
select @salary;  
commit transaction;  
go
```

2. Przygotuj skrypt nr 2:

- Rozpocznij transakcję.
- Zadeklaruj zmienną o nazwie *salary*.
- Przypisz jej wartość pensji pracownika o identyfikatorze równym 100.
- Obniż o 2000 zł wartość zmiennej *salary*.
- Ustaw pensję pracownika o identyfikatorze równym 100 na wartość zmiennej *salary*.
- Wyświetl wartość zmiennej *salary*.
- Zatwierdź transakcję.

```
begin transaction;
declare @salary numeric(8, 2);
set @salary = (select salary from employees where employee_id = 100);
set @salary = @salary - 2000;
update employees set salary = @salary where employee_id = 100;
select @salary;
commit transaction;
go
```

3. Uruchom skrypt nr 1 i zaraz po nim skrypt nr 2 w ramach dwóch różnych sesji.

sesja dla skryptu 1.

	(No column name)
1	23000.00

Sesja dla skryptu 2.

	(No column name)
1	22000.00

4. Wyświetl pensję pracownika o identyfikatorze równym 100.

```
select salary from employees where employee_id = 100;
```

	salary
1	23000.00

5. Jakie zjawisko zaprezentowano w tym zadaniu?

Utracone modyfikacje (ang. lost updates)

6. Na jakim minimalnym poziomie izolacji nie mogłoby dojść do jego wystąpienia?

Repeatable Read

Zadanie 2.

1. Przygotuj skrypt nr 1:

- Rozpocznij transakcję.
- Wyświetl pensję pracownika o identyfikatorze równym 100.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Wyświetl pensję pracownika o identyfikatorze równym 100.
- Zatwierdź transakcję.

```
begin transaction;
select salary from employees where employee_id = 100;
WAITFOR DELAY '00:00:15';
select salary from employees where employee_id = 100;
```

```
commit transaction;  
go
```

2. Przygotuj skrypt nr 2:

- Rozpocznij transakcję.
- Podwyższ o 1000 zł pensję pracownika o identyfikatorze równym 100.
- Zatwierdź transakcję.

```
begin transaction;  
update employees set salary = salary + 1000 where employee_id = 100;  
commit transaction;  
go
```

3. Uruchom skrypt nr 1 i zaraz po nim skrypt nr 2 w ramach dwóch różnych sesji.

sesja dla skryptu 1.

	salary
1	24000.00

	salary
1	25000.00

Sesja dla skryptu 2.

```
(1 row affected)  
  
Completion time: 2024-12-17T21:59:24.2717795+01:00
```

4. Wyświetl pensję pracownika o identyfikatorze równym 100.

```
select salary from employees where employee_id = 100;
```

	salary
1	25000.00

5. Jakie zjawisko zaprezentowano w tym zadaniu?

Niepowtarzalne odczyty (ang. Nonrepeatable (fuzzy) reads/inconsistent analysis)

6. Na jakim minimalnym poziomie izolacji nie mogłoby dojść do jego wystąpienia?

Repeatable Read

Zadanie 3.

1. Przygotuj skrypt nr 1:

- Rozpocznij transakcję.
- Wyświetl wszystkich pracowników, którzy zostali zatrudnieni w grudniu.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Wyświetl wszystkich pracowników, którzy zostali zatrudnieni w grudniu.
- Zatwierdź transakcję.

```
begin transaction;
select * from employees where MONTH(hire_date) = 12;
WAITFOR DELAY '00:00:15';
select * from employees where MONTH(hire_date) = 12;
commit transaction;
go
```

2. Przygotuj skrypt nr 2:

- Rozpocznij transakcję.
- Dodaj nowego pracownika, który został zatrudniony 1 grudnia 2022 roku.
- Zatwierdź transakcję.

```
begin transaction;
INSERT INTO employees VALUES ( 222 , 'Steven' , 'King' , 'aha' ,
'111.111.111' , CONVERT (DATE, '01-12-2022', 105) , 'AD_PRES' , 24000
, NULL , NULL , 90 );
commit transaction;
go
```

3. Uruchom skrypt nr 1 i zaraz po nim skrypt nr 2.

sesja dla skryptu 1.

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	113	Luis	Popp	LPOPP	515.124.4567	2007-12-07	FI_ACCOUNT	6900.00	NULL	108	100
2	114	Den	Raphaely	DRAPHEAL	515.127.4561	2002-12-07	PU_MAN	11000.00	NULL	100	30
3	116	Shelli	Baida	SBAIDA	515.127.4563	2005-12-24	PU_CLERK	2900.00	NULL	114	30
4	135	Ki	Gee	KGEE	650.127.1734	2007-12-12	ST_CLERK	2400.00	NULL	122	50
5	154	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668	2006-12-09	SA_REP	7500.00	0.20	145	80
6	160	Louise	Doran	LDORAN	011.44.1345.629268	2005-12-15	SA_REP	7500.00	0.30	146	80
7	191	Randall	Perkins	RPERKINS	650.505.4876	2007-12-19	SH_CLERK	2500.00	NULL	122	50
8	222	Steven	King	aha	111.111.111	2022-12-01	AD_PRES	24000.00	NULL	NULL	90

```
(1 row affected)
```

```
Completion time: 2024-12-17T22:02:59.7887532+01:00
```

```
select * from employees where MONTH(hire_date) = 12;
```

Odczyt widmo (ang. Phantom reads/phantoms)

6. Na jakim minimalnym poziomie izolacji nie mogłoby dojść do jego wystąpienia?

Serializable

Zadanie 4.

1. W ramach sesji nr 1 rozpocznij transakcję.
2. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

```
begin transaction;
select *
from sys.dm_tran_locks
where request session id = @@spid
```

1	resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status	request_reference_count	request_lifetime	request_session_id	request_expire
	DATABASE		6		0	0	S	LOCK	GRANT	1	0	72	0

3. W ramach sesji nr 1 na cały czas trwania transakcji załóż blokadę dzieloną na tabeli pracowników.
4. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

```
select* from employees WITH (HOLDLOCK, tablock) where employee_id =
null;
select *
from sys.dm_tran_locks
where request session id = @@spid
```

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id	
1											
2											

	resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status	request_reference_count	request_lifetime	request_session_id	request_exe
1	DATABASE	6			0	0	S	LOCK	GRANT	1	0	72	0
2	OBJECT	6			1141579105	0	S	LOCK	GRANT	1	33554432	72	0

5. W ramach sesji nr 2 rozpocznij transakcję.
6. W ramach sesji nr 2 wyświetl wszystkich pracowników.
7. W ramach sesji nr 2 podwyższ o 100 zł pensję wszystkich pracowników.

```
begin transaction;
select * from employees;
```

```
update employees set salary = salary + 100;
```

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	30000.00	NULL	NULL	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000.00	NULL	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000.00	NULL	100	90
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000.00	NULL	102	60
5	104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000.00	NULL	103	60
6	105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4900.00	NULL	103	60
7	106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800.00	NULL	103	60
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07	IT_PROG	4200.00	NULL	103	60
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	2002-08-17	FL_MGR	12008.00	NULL	101	100
10	109	Daniel	Faviet	DFAVIET	515.124.4169	2002-08-16	FL_ACCOUNT	9000.00	NULL	108	100
11	110	John	Chen	JCHEN	515.124.4269	2005-09-28	FL_ACCOUNT	8200.00	NULL	108	100
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	2005-09-30	FL_ACCOUNT	7700.00	NULL	108	100
13	112	Jose Manuel	Uman	JMURMAN	515.124.4469	2006-03-07	FL_ACCOUNT	7800.00	NULL	108	100
14	113	Luis	Popp	LPOPP	515.124.4567	2007-12-07	FL_ACCOUNT	6900.00	NULL	108	100
15	114	Den	Raphaely	DRAPHEAL	515.127.4561	2002-12-07	PJ_MAN	11000.00	NULL	100	30
16	115	Alexander	Khoos	AKHOOS	515.127.4562	2003-05-18	PJ_CLERK	3100.00	NULL	114	30
17	116	Shelli	Baida	SBAIDA	515.127.4563	2005-12-24	PJ_CLERK	2900.00	NULL	114	30
18	117	Sigal	Tobias	STOBIAS	515.127.4564	2005-07-24	PJ_CLERK	2800.00	NULL	114	30
19	118	Guy	Himuro	GHIIMURO	515.127.4565	2006-11-15	PJ_CLERK	2600.00	NULL	114	30
20	119	Karen	Colmena	KCOLMENA	515.127.4566	2007-08-10	PJ_CLERK	2500.00	NULL	114	30
21	120	Matthew	Weiss	MWEISS	650.123.1234	2004-07-18	ST_MAN	8000.00	NULL	100	50
22	121	Adam	Fripp	AFRIPP	650.123.2234	2005-04-10	ST_MAN	8200.00	NULL	100	50
23	122	Payam	Kaufling	PKAUFLIN	650.123.3234	2003-05-01	ST_MAN	7900.00	NULL	100	50
24	123	Shanta	Vollman	SVOLLMAN	650.123.4234	2005-10-10	ST_MAN	6500.00	NULL	100	50
25	124	Kevin	Mouroug	KMOURUG	650.123.5234	2007-11-16	ST_MAN	5800.00	NULL	100	50
26	125	Julia	Nayer	JNAYER	650.124.1214	2005-07-16	ST_CLERK	3200.00	NULL	120	50
27	126	Irene	Mikkilineni	IMIKKILI	650.124.1224	2006-09-28	ST_CLERK	2700.00	NULL	120	50
28	127	James	Landry	JLANDRY	650.124.1334	2007-01-14	ST_CLERK	2400.00	NULL	120	50
29	128	Steven	Markle	SMARKLE	650.124.1434	2008-03-08	ST_CLERK	2200.00	NULL	120	50
30	129	Laura	Bissot	LBISSOT	650.124.1534	2008-08-20	ST_CLERK	3300.00	NULL	121	50
31	130	Mozhe	Atkinson	MATKINSO	650.124.6234	2005-10-30	ST_CLERK	2800.00	NULL	121	50
32	131	James	Marlow	JAMRLOW	650.124.7234	2005-02-16	ST_CLERK	2500.00	NULL	121	50
33	132	TJ	Olson	TJOLSON	650.124.8234	2007-04-10	ST_CLERK	2100.00	NULL	121	50
34	133	Jason	Mallin	JMALLIN	650.127.1934	2004-06-14	ST_CLERK	3300.00	NULL	122	50
35	134	Michael	Rones	MROVERS	650.127.1934	2006-08-26	ST_CLERK	2900.00	NULL	122	50

Executing query...DESKTOP-SBMTOL6\MSSQLSERVER...DESKTOP-SBMTOL6\Lenovo...hr00:00:330 rows

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
62	161	Sarath	Sewall	SSEWALL	011.44.1345...	2006-11-03	SA_REP	7000.00	0.25	146	80
63	162	Clara	Vishney	CVISHNEY	011.44.1346...	2005-11-11	SA_REP	10500.00	0.25	147	80
64	163	Danielle	Greene	DGREENE	011.44.1346...	2007-03-19	SA_REP	9500.00	0.15	147	80
65	164	Mattea	Marvins	MMARVINS	011.44.1346...	2008-01-24	SA_REP	7200.00	0.10	147	80
66	165	David	Lee	DLEE	011.44.1346...	2008-02-23	SA_REP	6800.00	0.10	147	80
67	166	Sundar	Ande	SANDE	011.44.1346...	2008-03-24	SA_REP	6400.00	0.10	147	80
68	167	Amit	Banda	ABANDA	011.44.1346...	2008-04-21	SA_REP	6200.00	0.10	147	80
69	168	Lisa	Ozer	LOZER	011.44.1343...	2005-03-11	SA_REP	11500.00	0.25	148	80
70	169	Harrison	Bloom	HBLOOM	011.44.1343...	2006-03-23	SA_REP	10000.00	0.20	148	80
71	170	Taylor	Fox	TFOX	011.44.1343...	2006-01-24	SA_REP	9600.00	0.20	148	80
72	171	William	Smith	WSMITH	011.44.1343...	2007-02-23	SA_REP	7400.00	0.15	148	80
73	172	Elizabeth	Bates	EBATES	011.44.1343...	2007-03-24	SA_REP	7300.00	0.15	148	80
74	173	Sundita	Kumar	SKUMAR	011.44.1343...	2008-04-21	SA_REP	6100.00	0.10	148	80
75	174	Ellen	Abel	EABEL	011.44.1644...	2004-05-11	SA_REP	11000.00	0.30	149	80
76	175	Alyssa	Hutton	AHUTTON	011.44.1644...	2005-03-19	SA_REP	8800.00	0.25	149	80
77	176	Jonathan	Taylor	JTAYLOR	011.44.1644...	2006-03-24	SA_REP	8600.00	0.20	149	80
78	177	Jack	Livingston	JLIVINGS	011.44.1644...	2006-04-23	SA_REP	8400.00	0.20	149	80
79	178	Kimberely	Grant	KGRANT	011.44.1644...	2007-05-24	SA_REP	7000.00	0.15	149	NULL
80	179	Charles	Johnson	CJOHNSON	011.44.1644...	2008-01-04	SA_REP	6200.00	0.10	149	80
81	180	Winston	Taylor	WTAYLOR	650.507.9876	2006-01-24	SH_CLERK	3200.00	NULL	120	50
82	181	Jean	Fleaur	JFLEAUR	650.507.9877	2006-02-23	SH_CLERK	3100.00	NULL	120	50
83	182	Martha	Sullivan	MSULLIVA	650.507.9878	2007-06-21	SH_CLERK	2500.00	NULL	120	50
84	183	Girard	Geoni	GGEONI	650.507.9879	2008-02-03	SH_CLERK	2800.00	NULL	120	50
85	184	Nandita	Sarchand	NSARCHAN	650.509.1876	2004-01-27	SH_CLERK	4200.00	NULL	121	50
86	185	Alexis	Bull	ABULL	650.509.2876	2005-02-20	SH_CLERK	4100.00	NULL	121	50
87	186	Julia	Dellinger	JDELLING	650.509.3876	2006-06-24	SH_CLERK	3400.00	NULL	121	50
88	187	Anthony	Cabrio	ACABRIO	650.509.4876	2007-02-07	SH_CLERK	3000.00	NULL	121	50
89	188	Kelly	Chung	KCHUNG	650.505.1876	2005-06-14	SH_CLERK	3800.00	NULL	122	50
90	189	Jennifer	Dilly	JDILLY	650.505.2876	2005-08-13	SH_CLERK	3600.00	NULL	122	50
91	190	Timothy	Gates	TGATES	650.505.3876	2006-07-11	SH_CLERK	2900.00	NULL	122	50
92	191	Randall	Perkins	RPERKINS	650.505.4876	2007-12-19	SH_CLERK	2500.00	NULL	122	50
93	192	Sarah	Bell	SBELL	650.501.1876	2004-02-04	SH_CLERK	4000.00	NULL	123	50
94	193	Britney	Everett	BEVERETT	650.501.2876	2005-03-03	SH_CLERK	3900.00	NULL	123	50
95	194	Samuel	McCain	SMCCAIN	650.501.3876	2006-07-01	SH_CLERK	3200.00	NULL	123	50

Executing query...DESKTOP-SBMTOL6\MSSQLSERVER...DESKTOP-SBMTOL6\Lenovo...hr00:00:590 rows

8. W ramach sesji nr 1 zatwierdź transakcję.

```
commit transaction;
```

sesja dla skryptu 1.

```
Commands completed successfully.

Completion time: 2024-12-17T22:11:51.8385090+01:00
```

sesja dla skryptu 2.

Query executed successfully.DESKTOP-SBMTOL6\MSSQLSERVER...DESKTOP-SBMTOL6\Lenovo...hr00:01:57

9. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

```
select *
from sys.dm_tran_locks
where request_session_id = @@spid
```

	resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status	request_reference_count	request_lifetime	request_session_id	request_expire_time
1	DATABASE		6		0	0	S	LOCK	GRANT	1	0	72	0

10. W ramach sesji nr 2 wycofaj transakcję.

```
rollback transaction;
```

Commands completed successfully.

Completion time: 2024-12-17T22:14:21.3883107+01:00

11. W którym momencie udało Ci się podwyższyć pensję wszystkich pracowników w ramach sesji nr 2?

Po zatwierdzeniu transakcji w kroku 8.

Zadanie 5.

1. W ramach sesji nr 1 rozpocznij transakcję.
2. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

```
begin transaction;
select *
from sys.dm_tran_locks
where request_session_id = @@spid
```

	resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status	request_reference_count	request_lifetime	request_session_id	request_expire_time
1	DATABASE		6		0	0	S	LOCK	GRANT	1	0	72	0

3. W ramach sesji nr 1 na cały czas trwania transakcji załóż blokadę wyłączną na tabeli pracowników.

4. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

5. W ramach sesji nr 1 podwyższ o 100 zł pensję wszystkich pracowników.

```
select* from employees WITH (HOLDLOCK, tablock, xlock) where
employee_id = null;
select *
from sys.dm_tran_locks
where request_session_id = @@spid
update employees set salary = salary + 100;
```

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id

```
begin transaction;  
select * from employees;
```

8. W ramach sesji nr 1 wycofaj transakcję.

sesja 1.

sesja 2.

9. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

	resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status	request_reference_count	request_lifetime	request_session_id	request_exe
1	DATABASE		6		0	0	S	LOCK	GRANT	1	0	72	0

10. W ramach sesji nr 2 zatwierdź transakcję.


```
Commands completed successfully.
```

```
Completion time: 2024-12-17T22:20:40.8809546+01:00
```

11. W którym momencie udało Ci się wyświetlić wszystkich pracowników w ramach sesji nr 2?

Po wycofaniu transakcji w kroku 8.

Zadanie 6.

1. Przygotuj skrypt nr 1:

- Rozpocznij transakcję.
- Ustaw maksymalną pensję na stanowisku President na 30000 zł.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Ustaw pensję pracownika o identyfikatorze równym 100 na 30000 zł.
- Zatwierdź transakcję.

```
begin transaction;
update jobs set max_salary = 30000 where job_title = 'President';
waitfor delay '00:00:15';
update employees set salary = 30000 where employee_id = 100;
commit transaction;
go
```

2. Przygotuj skrypt nr 2:

- Rozpocznij transakcję.
- Ustaw pensję pracownika o identyfikatorze równym 100 na 30000 zł.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Ustaw maksymalną pensję na stanowisku President na 30000 zł.
- Zatwierdź transakcję.

```
begin transaction;
update employees set salary = 30000 where employee_id = 100;
waitfor delay '00:00:15';
update jobs set max_salary = 30000 where job_title = 'President';
commit transaction;
go
```

3. Uruchom skrypt nr 1 i zaraz po nim skrypt nr 2 w ramach dwóch różnych sesji.

4. Poczekaj, aż zostanie wyświetlony odpowiedni komunikat.

```
(1 row affected)
Msg 1205, Level 13, State 51, Line 5
Transaction (Process ID 81) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Completion time: 2024-12-17T22:05:47.6080449+01:00
```

5. Jakie zjawisko zaprezentowano w tym zadaniu?

Zakleszczenie

PostgreSQL

Zadanie 1.

1. Przygotuj skrypt nr 1:

- Rozpocznij transakcję.
- Zadeklaruj zmienną o nazwie *salary*.
- Przypisz jej wartość maksymalnej pensji na stanowisku Programmer.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Obniż o 1000 zł wartość zmiennej *salary*.
- Ustaw maksymalną pensję na stanowisku Programmer na wartość zmiennej *salary*.
- Wyświetl wartość zmiennej *salary*.
- Zatwierdź transakcję.

```
begin;

do $$
declare
    salary numeric(6);
begin

    select max_salary into salary
    from jobs where job_title = 'Programmer';

    perform pg_sleep(15);

    salary := salary - 1000;

    update jobs set max_salary = salary where job_title =
'Programmer';
    raise notice 'salary = %', salary;

end;
```

```
$$ LANGUAGE plpgsql;  
  
commit;
```

2. Przygotuj skrypt nr 2:

- Rozpocznij transakcję.
- Zadeklaruj zmienną o nazwie *salary*.
- Przypisz jej wartość maksymalnej pensji na stanowisku Programmer.
- Obniż o 2000 zł wartość zmiennej *salary*.
- Ustaw maksymalną pensję na stanowisku Programmer na wartość zmiennej *salary*.
- Wyświetl wartość zmiennej *salary*.
- Zatwierdź transakcję.

```
begin;  
  
do $$  
declare  
    salary numeric(6);  
begin  
  
    select max_salary into salary  
    from jobs where job_title = 'Programmer';  
  
    salary := salary - 2000;  
  
    update jobs set max_salary = salary where job_title = 'Programmer';  
    raise notice 'salary = %', salary;  
  
end;  
$$ LANGUAGE plpgsql;  
  
commit;
```

3. Uruchom skrypt nr 1 i zaraz po nim skrypt nr 2 w ramach dwóch różnych sesji.

Sesja dla skryptu 1.

```
UWAGA: salary = 9000  
COMMIT
```

```
Query returned successfully in 15 secs 671 msec.
```


Sesja dla skryptu 2.

```
UWAGA: salary = 8000  
COMMIT
```

Query returned successfully in 62 msec.

4. Wyświetl maksymalną pensję na stanowisku Programmer.

```
select max_salary from jobs where job_title = 'Programmer';
```

	max_salary numeric (6) 
1	9000

5. Jakie zjawisko zaprezentowano w tym zadaniu?

Utracone modyfikacje (ang. lost updates)

6. Na jakim minimalnym poziomie izolacji nie mogłoby dojść do jego wystąpienia?


Repeatable Read

Zadanie 2.

1. W ramach sesji nr 1 rozpocznij transakcję.

2. W ramach sesji nr 1 wyświetl maksymalną pensję na stanowisku Programmer.

```
begin;  
select max_salary from jobs where job_title = 'Programmer';
```

	max_salary numeric (6) 
1	10000

3. W ramach sesji nr 2 rozpocznij transakcję.

4. W ramach sesji nr 2 podwyższ o 1000 zł maksymalną pensję na stanowisku Programmer.

```
begin;  
update jobs set max_salary = max_salary + 1000  
where job_title = 'Programmer';
```

UPDATE 1

Query returned successfully in 39 msec.

5. W ramach sesji nr 2 zatwierdź transakcję.


```
commit;
```

COMMIT

Query returned successfully in 39 msec.

6. W ramach sesji nr 1 wyświetl maksymalną pensję na stanowisku Programmer.

```
select max_salary from jobs where job_title = 'Programmer';
```

	max_salary numeric (6) 
1	11000

7. W ramach sesji nr 1 zatwierdź transakcję.


```
commit;
```

COMMIT

Query returned successfully in 56 msec.

8. W ramach sesji nr 1 wyświetl maksymalną pensję na stanowisku Programmer.

```
select max_salary from jobs where job_title = 'Programmer';
```

	max_salary numeric (6) 
1	11000

9. Jakie zjawisko zaprezentowano w tym zadaniu?

Niepowtarzalne odczyty (ang. Nonrepeatable (fuzzy) reads/inconsistent analysis)

10. Na jakim minimalnym poziomie izolacji nie mogłoby dojść do jego wystąpienia?

Repeatable Read

Zadanie 3.

1. W ramach sesji nr 1 rozpocznij transakcję.
2. W ramach sesji nr 1 wyświetl wszystkie stanowiska, na których minimalna pensja wynosi 2500 zł.

```
begin;  
select * from jobs where min_salary = 2500;
```

	job_id [PK] character varying (10)	job_title character varying (35)	min_salary numeric (6)	max_salary numeric (6)
1	PU_CLERK	Purchasing Clerk	2500	5500
2	SH_CLERK	Shipping Clerk	2500	5500

3. W ramach sesji nr 2 rozpocznij transakcję.
4. W ramach sesji nr 2 dodaj nowe stanowisko, na którym minimalna pensja wynosi 2500 zł.

```
begin;  
INSERT INTO jobs VALUES ( 'NOWE' , 'NNN' , 2500 , 4000 );
```

```
INSERT 0 1
```

Query returned successfully in 56 msec.

5. W ramach sesji nr 2 zatwierdź transakcję.

```
commit;
```

```
COMMIT
```

Query returned successfully in 41 msec.

6. W ramach sesji nr 1 wyświetl wszystkie stanowiska, na których minimalna pensja wynosi 2500 zł.

```
select * from jobs where min_salary = 2500;
```

	job_id [PK] character varying (10)	job_title character varying (35)	min_salary numeric (6)	max_salary numeric (6)
1	PU_CLERK	Purchasing Clerk	2500	5500
2	SH_CLERK	Shipping Clerk	2500	5500
3	NOWE	NNN	2500	4000

7. W ramach sesji nr 1 zatwierdź transakcję.

```
commit;
```

COMMIT

Query returned successfully in 53 msec.

8. W ramach sesji nr 1 wyświetl wszystkie stanowiska, na których minimalna pensja wynosi 2500 zł.

```
select * from jobs where min_salary = 2500;
```

	job_id [PK] character varying (10)	job_title character varying (35)	min_salary numeric (6)	max_salary numeric (6)
1	PU_CLERK	Purchasing Clerk	2500	5500
2	SH_CLERK	Shipping Clerk	2500	5500
3	NOWE	NNN	2500	4000

9. Jakie zjawisko zaprezentowano w tym zadaniu?

Odczyt widmo (ang. Phantom reads/phantoms)

10. Na jakim minimalnym poziomie izolacji nie mogłoby dojść do jego wystąpienia?

Repeatable Read

Zadanie 4.

1. W ramach sesji nr 1 rozpocznij transakcję.
2. W ramach sesji nr 1 wyświetl listę wszystkich blokad.

```
begin;  
SELECT * FROM pg_locks;
```

	locktype text	database oid	relation oid	page integer	tuple smallint	virtualxid text	transactionid xid	classid oid	objid oid	objsubid smallint	virtualtransaction text	pid integer	mode text	granted boolean
1	relation	16386	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
2	virtualxid	[null]	[null]	[null]	[null]	2/9	[null]	[null]	[null]	[null]	2/9	13104	ExclusiveLock	true

3. W ramach sesji nr 1 na cały czas trwania transakcji załóż blokadę dzieloną na tabeli stanowisk.

```
LOCK TABLE jobs IN SHARE MODE;
```

LOCK TABLE

Query returned successfully in 137 msec.

4. W ramach sesji nr 1 wyświetl listę wszystkich blokad.

```
SELECT * FROM pg_locks;
```

	locktype text	database oid	relation oid	page integer	tuple smallint	virtualxid text	transactionid xid	classid oid	objid oid	objsubid smallint	virtualtransaction text	pid integer	mode text	granted boolean
1	relation	16386	2579	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
2	relation	16386	2606	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
3	relation	16386	2685	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
4	relation	16386	2684	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
5	relation	16386	3455	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
6	relation	16386	2663	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
7	relation	16386	2662	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
8	relation	16386	2704	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
9	relation	16386	2703	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
10	relation	16386	2659	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
11	relation	16386	2658	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
12	relation	16386	2615	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
13	relation	16386	1259	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
14	relation	16386	1247	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
15	relation	16386	1249	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
16	relation	16386	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
17	virtualxid	[null]	[null]	[null]	[null]	2/9	[null]	[null]	[null]	[null]	2/9	13104	ExclusiveLock	true
18	relation	16386	2674	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
19	relation	16386	2610	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
20	relation	16386	2604	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
21	relation	16386	2667	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
22	relation	16386	5002	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
23	relation	16386	2666	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
24	relation	16386	2608	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
25	relation	16386	2664	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
26	relation	16386	2665	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
27	relation	16386	2679	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
28	relation	16386	2678	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
29	relation	16386	16422	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	ShareLock	true
30	relation	16386	2657	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
31	relation	16386	2656	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
32	relation	16386	2673	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
33	relation	16386	2609	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
34	relation	16386	2224	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true
35	relation	16386	2675	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/9	13104	AccessShareLock	true

5. W ramach sesji nr 2 rozpocznij transakcję.

6. W ramach sesji nr 2 wyświetl wszystkie stanowiska.

```
begin;  
select * from jobs;
```


	job_id [PK] character varying (10)	job_title character varying (35)	min_salary numeric (6)	max_salary numeric (6)
1	AD_PRES	President	20080	40000
2	AD_VP	Administration Vice President	15000	30000
3	AD_ASST	Administration Assistant	3000	6000
4	FI_MGR	Finance Manager	8200	16000
5	FI_ACCOUNT	Accountant	4200	9000
6	AC_MGR	Accounting Manager	8200	16000
7	AC_ACCOUNT	Public Accountant	4200	9000
8	SA_MAN	Sales Manager	10000	20080
9	SA_REP	Sales Representative	6000	12008
10	PU_MAN	Purchasing Manager	8000	15000
11	PU_CLERK	Purchasing Clerk	2500	5500
12	ST_MAN	Stock Manager	5500	8500
13	ST_CLERK	Stock Clerk	2008	5000
14	SH_CLERK	Shipping Clerk	2500	5500
15	IT_PROG	Programmer	4000	10000
16	MK_MAN	Marketing Manager	9000	15000
17	MK_REP	Marketing Representative	4000	9000
18	HR_REP	Human Resources Representa...	4000	9000
19	PR_REP	Public Relations Representative	4500	10500

7. W ramach sesji nr 2 podwyższ o 100 zł maksymalną pensję na wszystkich stanowiskach.

```
update jobs set max_salary = max_salary + 100;
```

job_id [PK] character varying (10)	job_title character varying (35)	min_salary numeric (6)	max_salary numeric (6)
1	AD_PRES	20080	40000
2	AD_VP	15000	30000
3	AD_ASST	3000	6000
4	FI_MGR	8200	16000
5	FI_ACCOUNT	4200	9000
6	AC_MGR	8200	16000
7	AC_ACCOUNT	4200	9000

8. W ramach sesji nr 1 zatwierdź transakcję.

```
commit;
```

```
COMMIT
```

```
Query returned successfully in 123 msec.
```

```
UPDATE 19
```

```
Query returned successfully in 36 secs 410 msec.
```

9. W ramach sesji nr 2 wycofaj transakcję.

```
rollback;
```

```
ROLLBACK
```

```
Query returned successfully in 184 msec.
```

10. W ramach sesji nr 1 wyświetl listę wszystkich blokad.

```
SELECT * FROM pg_locks;
```

	locktype text	database oid	relation oid	page integer	tuple smallint	virtualxid text	transactionid xid	classid oid	objid oid	objsubid smallint	virtualtransaction text	pid integer	mode text	granted boolean
1	relation	16386	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	2/10	13104	AccessShareLock	true
2	virtualxid	[null]	[null]	[null]	[null]	2/10	[null]	[null]	[null]	[null]	2/10	13104	ExclusiveLock	true

11. W którym momencie udało Ci się podwyższyć maksymalną pensję na wszystkich stanowiskach w ramach sesji nr 2?

Po zatwierdzeniu transakcji w kroku 8.

Zadanie 5.

1. W ramach sesji nr 1 rozpocznij transakcję.
2. W ramach sesji nr 1 wyświetl listę wszystkich blokad.

```
begin;  
SELECT * FROM pg_locks;
```

	locktype text	database oid	relation oid	page integer	tuple smallint	virtualxid text	transactionid xid	classid oid	objid oid	objsubid smallint	virtualtransaction text	pid integer	mode text	granted boolean
1	relation	16386	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
2	virtualxid	[null]	[null]	[null]	[null]	7/7	[null]	[null]	[null]	[null]	7/7	8268	ExclusiveLock	true

3. W ramach sesji nr 1 na cały czas trwania transakcji załóż blokadę wyłączną na tabeli stanowisk.

```
LOCK TABLE jobs IN EXCLUSIVE MODE;
```

LOCK TABLE

Query returned successfully in 206 msec.

4. W ramach sesji nr 1 wyświetl listę wszystkich blokad.

```
SELECT * FROM pg_locks;
```

	locktype text	database oid	relation oid	page integer	tuple smallint	virtualxid text	transactionid xid	classid oid	objid oid	objsubid smallint	virtualtransaction text	pid integer	mode text	granted boolean
1	relation	16386	2579	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
2	relation	16386	2606	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
3	relation	16386	2685	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
4	relation	16386	2684	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
5	relation	16386	3455	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
6	relation	16386	2663	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
7	relation	16386	2662	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
8	relation	16386	2704	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
9	relation	16386	2703	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
10	relation	16386	2659	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
11	relation	16386	2658	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
12	relation	16386	2615	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
13	relation	16386	1259	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
14	relation	16386	1247	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
15	relation	16386	1249	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
16	relation	16386	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
17	virtualxid	[null]	[null]	[null]	[null]	7/7	[null]	[null]	[null]	[null]	7/7	8268	ExclusiveLock	true
18	relation	16386	2667	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
19	relation	16386	5002	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
20	relation	16386	2666	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
21	relation	16386	2608	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
21	relation	16386	2608	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
22	relation	16386	2674	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
23	relation	16386	2610	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
24	relation	16386	2604	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
25	relation	16386	2679	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
26	relation	16386	2678	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
27	relation	16386	16422	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	ExclusiveLock	true
28	relation	16386	2657	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
29	relation	16386	2656	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
30	relation	16386	2673	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
31	relation	16386	2609	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
32	relation	16386	2224	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
33	relation	16386	2675	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
34	relation	16386	2664	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true
35	relation	16386	2665	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/7	8268	AccessShareLock	true

5. W ramach sesji nr 1 podwyższ o 100 zł maksymalną pensję na wszystkich stanowiskach.

```
update jobs set max_salary = max_salary + 100;
```

UPDATE 19

Query returned successfully in 162 msec.

6. W ramach sesji nr 2 rozpocznij transakcję.

7. W ramach sesji nr 2 wyświetl wszystkie stanowiska.

```
begin;  
select * from jobs;
```

	job_id [PK] character varying (10)	job_title character varying (35)	min_salary numeric (6)	max_salary numeric (6)
1	AD_PRES	President	20080	40000
2	AD_VP	Administration Vice President	15000	30000
3	AD_ASST	Administration Assistant	3000	6000
4	FI_MGR	Finance Manager	8200	16000
5	FI_ACCOUNT	Accountant	4200	9000
6	AC_MGR	Accounting Manager	8200	16000
7	AC_ACCOUNT	Public Accountant	4200	9000
8	SA_MAN	Sales Manager	10000	20080
9	SA_REP	Sales Representative	6000	12008
10	PU_MAN	Purchasing Manager	8000	15000
11	PU_CLERK	Purchasing Clerk	2500	5500
12	ST_MAN	Stock Manager	5500	8500
13	ST_CLERK	Stock Clerk	2008	5000
14	SH_CLERK	Shipping Clerk	2500	5500
15	IT_PROG	Programmer	4000	10000
16	MK_MAN	Marketing Manager	9000	15000
17	MK_REP	Marketing Representative	4000	9000
18	HR_REP	Human Resources Representa...	4000	9000
19	PR_REP	Public Relations Representative	4500	10500

8. W ramach sesji nr 1 wycofaj transakcję.

```
rollback;
```

```
ROLLBACK
```

```
Query returned successfully in 169 msec.
```

9. W ramach sesji nr 2 zatwierdź transakcję.

```
commit;
```

```
COMMIT
```

```
Query returned successfully in 216 msec.
```

10. W ramach sesji nr 1 wyświetl listę wszystkich blokad.

```
SELECT * FROM pg_locks;
```

	locktype text	database oid	relation oid	page integer	tuple smallint	virtualxid text	transactionid xid	classid oid	objid oid	objsubid smallint	virtualtransaction text	pid integer	mode text	granted boolean
1	relation	16386	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	7/9	8268	AccessShareLock	true
2	virtualxid	[null]	[null]	[null]	[null]	7/9	[null]	[null]	[null]	[null]	7/9	8268	ExclusiveLock	true

11. W którym momencie udało Ci się wyświetlić wszystkie stanowiska w ramach sesji nr 2?

Przy wykonywaniu zapytania select w kroku 7.

Zadanie 6.

1. Przygotuj skrypt nr 1:

- Rozpocznij transakcję.
- Ustaw minimalną pensję na stanowisku Programmer na 5000 zł.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Ustaw pensję pracownika o identyfikatorze równym 103 na 5000 zł.
- Zatwierdź transakcję.

```
begin;  
do $$  
begin  
    update jobs set min_salary = 5000 where job_title = 'Programmer';  
    perform pg_sleep(15);  
    update employees set salary = 5000 where employee_id = 103;  
end;  
end;
```

```
$$ LANGUAGE plpgsql;  
commit;
```

2. Przygotuj skrypt nr 2:

- Rozpocznij transakcję.
- Ustaw pensję pracownika o identyfikatorze równym 103 na 5000 zł.
- Wstrzymaj wykonanie skryptu na 15 sekund.
- Ustaw minimalną pensję na stanowisku Programmer na 5000 zł.
- Zatwierdź transakcję.

```
begin;  
do $$  
begin  
    update employees set salary = 5000 where employee_id = 103;  
    perform pg_sleep(15);  
    update jobs set min_salary = 5000 where job_title = 'Programmer';  
end;  
$$ LANGUAGE plpgsql;  
commit;
```

3. Uruchom skrypt nr 1 i zaraz po nim skrypt nr 2 w ramach dwóch różnych sesji.

4. Poczekaj, aż zostanie wyświetlony odpowiedni komunikat.

```
ERROR:  Proces 14628 oczekuje na ShareLock na transakcja 1455; zablokowany przez 4552.  
Proces 4552 oczekuje na ShareLock na transakcja 1456; zablokowany przez 14628.wykryto zakleszczenie  
  
BŁĄD: wykryto zakleszczenie  
SQL state: 40P01  
Detail: Proces 14628 oczekuje na ShareLock na transakcja 1455; zablokowany przez 4552.  
Proces 4552 oczekuje na ShareLock na transakcja 1456; zablokowany przez 14628.  
Hint: Przejrzyj dziennik serwera by znaleźć szczegóły zapytania.  
Context: podczas modyfikacji krotki (0,21) w relacji "jobs"  
wyrażenie SQL "update jobs set min_salary = 5000 where job_title = 'Programmer'"  
funkcja PL/pgSQL inline_code_block, wiersz 5 w wyrażenie SQL
```

5. Jakie zjawisko zaprezentowano w tym zadaniu?

Zakleszczenie