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
```

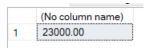
2. Przygotuj skrypt nr 2:

- 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.

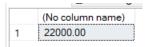
```
begin transaction;
declare @salary numeric(8, 2);
set @salary = (select salary from employees where employee_id =
100);
set @salary = @salary - 2000;
update employees set salary = @salary where employee_id = 100;
select @salary;
commit transaction;
go
```

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

sesja dla skryptu 1.



Sesja dla skryptu 2.



4. Wyświetl pensję pracownika o identyfikatorze równym 100.

select salary from employees where employee id = 100;



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
```

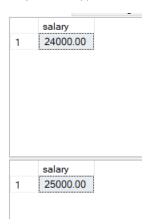
2. Przygotuj skrypt nr 2:

- 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.



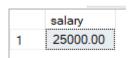
Sesja dla skryptu 2.

```
(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;



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
```

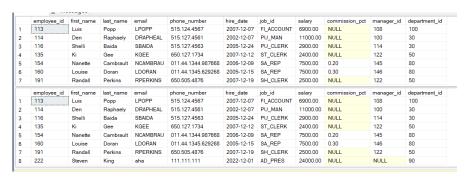
2. Przygotuj skrypt nr 2:

- 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.



sesja dla skryptu 2.

```
(1 row affected)

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

4. Wyświetl wszystkich pracowników, którzy zostali zatrudnieni w grudniu.

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

5. Jakie zjawisko zaprezentowano w tym zadaniu?

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
```

```
resource_type resource_subtype resource_database_id resource_description resource_associated_entity_id resource_lock_partition request_mode request_tetus request_reference_count request_lefetime request_exe 1 DATABASE LOCK GRANT 1 0 72
```

- 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
```

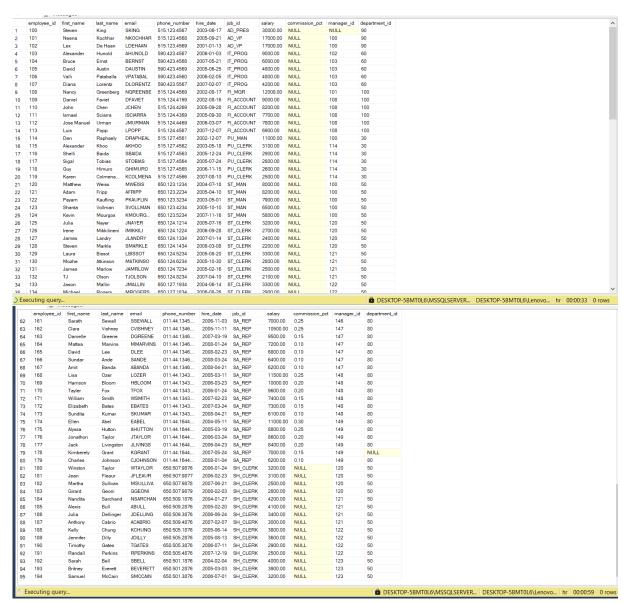
```
employee_id first_name last_name email phone_number hire_date job_id salary commission_pct manager_id department_id

resource_type resource_subtype resource_database_id resource_description resource_essociated_entity_id resource_lock_partition request_mode request_type request_status request_reference_count request_leftime request_ession_id request_ess
```

- 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;



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.

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_enterpression_id request_exession_id request_exess

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.

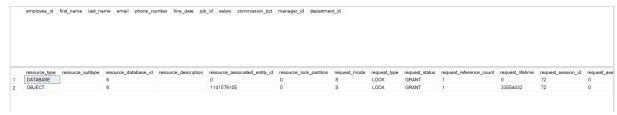
- 1. W ramach sesji nr 1 rozpocznij transakcję.
- 2. W ramach sesji nr 1 wyświetl listę blokad, za które odpowiedzialna jest ta sesja.

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

```
resource_type resource_subbye resource_database_id resource_database_id resource_associated_entity_id resource_lock_partition request_mode request_type request_tstatus request_reference_count request_fetime request_ession_id req
```

- 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;
```



- 6. W ramach sesji nr 2 rozpocznij transakcję.
- 7. W ramach sesji nr 2 wyświetl wszystkich pracowników.

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



8. W ramach sesji nr 1 wycofaj transakcję.

rollback transaction;

sesja 1.

Commands completed successfully.

Completion time: 2024-12-17T22:18:50.5263377+01:00

sesja 2.

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id				
98	197	Kevin	Feeney	KFEENEY	650.507.9822	2006-05-23	SH_CLERK	3000.00	NULL	124	50				
99	198	Donald	OConnell	DOCONNEL	650.507.9833	2007-06-21	SH_CLERK	2600.00	NULL	124	50				
100	199	Douglas	Grant	DGRANT	650.507.9844	2008-01-13	SH_CLERK	2600.00	NULL	124	50				
101	200	Jennifer	Whalen	JWHALEN	515.123.4444	2003-09-17	AD_ASST	4400.00	NULL	101	10				
102	201	Michael	Hartstein	MHARTSTE	515.123.5555	2004-02-17	MK_MAN	13000.00	NULL	100	20				
103	202	Pat	Fay	PFAY	603.123.6666	2005-08-17	MK_REP	6000.00	NULL	201	20				
104	203	Susan	Mavris	SMAVRIS	515.123.7777	2002-06-07	HR_REP	6500.00	NULL	101	40				
105	204	Hermann	Baer	HBAER	515.123.8888	2002-06-07	PR_REP	10000.00	NULL	101	70				
106	205	Shelley	Higgins	SHIGGINS	515.123.8080	2002-06-07	AC_MGR	12008.00	NULL	101	110				
107	206	William	Gietz	WGIETZ	515.123.8181	2002-06-07	AC_ACCOU	8300.00	NULL	205	110				
108	222	Steven	King	aha	111,111,111	2022-12-01	AD_PRES	24000.00	NULL	NULL	90				
9 Q	uery executed	l successful	ly.								■ DESKTO	5BMT0L6\MSSQLSERVER.	DESKTOP-5	. DESKTOP-5BMT0L6\Lenovo	DESKTOP-5BMT0L6\Lenovo hr 00:

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
```



10. W ramach sesji nr 2 zatwierdź transakcję.

commit transaction;

```
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.





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';
```



- 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.

\sim	\sim m	mп	+ •
<u>ر</u>	om	1117	」しょ

COMMIT

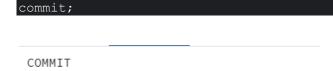
Query returned successfully in 39 msec.

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.



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

5. W ramach sesji nr 2 zatwierdź transakcję.

Query returned successfully in 56 msec.

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	Г						
	Query	returned	success	fully	in	53	msec	

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

	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

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
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;



8. W ramach sesji nr 1 zatwierdź transakcję.

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 Query returned successfully in 184 msec.

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



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

Po zatwierdzeniu transakcji w kroku 8.

SELECT * FROM pg locks;

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	17401	12073	[null]	[null]	[null]	[null]	[null]	[null]	[null]	8/34	4552	AccessShareLock	true
2	virtualxid	[null]	[null]	[null]	[null]	8/34	[null]	[null]	[null]	[null]	8/34	4552	ExclusiveLock	true

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

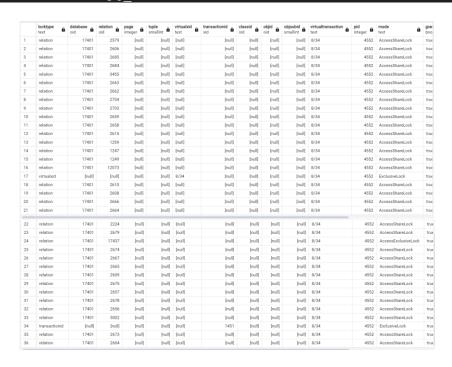
LOCK TABLE jobs IN ACCESS EXCLUSIVE MODE;

LOCK TABLE

Query returned successfully in 70 msec.

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

SELECT * FROM pg locks;



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

update jobs set max_salary = max_salary + 100;

Query returned successfully in 62 msec.

- 6. W ramach sesji nr 2 rozpocznij transakcję.
- 7. W ramach sesji nr 2 wyświetl wszystkie stanowiska.

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

8. W ramach sesji nr 1 wycofaj transakcję.

rollback;

sesja 1.

ROLLBACK

Query returned successfully in 72 msec.

Total rows: 20 of 20 Waiting for the query to complete... 00:00:02.735 Ln 5, Col 20

sesja 2.

	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	MK_MAN	Marketing Manager	9000	15000
16	MK_REP	Marketing Representative	4000	9000
17	HR_REP	Human Resources Representa	4000	9000
18	PR_REP	Public Relations Representative	4500	10500
19	IT_PROG	Programmer	4000	11000
20	NOWE	NNN	2500	4000

9. W ramach sesji nr 2 zatwierdź transakcję.

commit;

Query returned successfully in 59 msec.

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



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

Po wycofaniu transakcji w kroku 8. (po zwolnieniu blokady wyłącznej)

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