BADANIE POZIOMÓW IZOLACJI W SQLDEVELOPER

1. Izolacja READ_COMMITED

a. Dirty read

- Cel: zbadanie, czy poziom izolacji READ_COMMITED chroni przed anomalią „dirty read”
- Przebieg:

CZAS	Sesja 1	Sesja 2
1.	SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
2.	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000	SET TRANSACTION ISOLATION LEVEL READ COMMITTED
3.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000
4.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002	
5.	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 12000	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000
6.	COMMIT	

7.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 12000

- Wnioski: izolacja READ_COMMITED zabezpiecza przed anomalią „dirty read”.

b. Non-repeatable read

- Cel: zbadanie, czy poziom izolacji READ_COMMITED chroni przed anomalią „non-repeatable read”
- Przebieg:

CZAS	Sesja 1	Sesja 2
1.	SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
2.	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000	SET TRANSACTION ISOLATION LEVEL READ COMMITTED
3.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000
4.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002	
5.	COMMIT	
6.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 12000

- Wnioski: izolacja READ_COMMITTED nie chroni przed anomalią „non-repeatable read”.

c. Phantom read

- Cel: sprawdzenie, czy izolacja READ_COMMITTED chroni przed anomalią „phantom read”.
- Przebieg:

CZAS	Sesja 1	Sesja 2
1.	SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
2.		SET TRANSACTION ISOLATION LEVEL READ COMMITTED
3.	INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES (SEQ_ACCOUNT_NUMBER.NEXTVAL, 500)	SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS 20000
4.	COMMIT	
5.		SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS 20500

- Wnioski: izolacja READ_COMMITTED nie chroni przed anomalią „phantom read”.

d. Lost updates

- Cel: sprawdzenie, czy izolacja READ_COMMITTED chroni przed anomalią „lost updates”.
- Przebieg:

CZAS	Sesja 1	Sesja 2
1.	SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
2.		SET TRANSACTION ISOLATION LEVEL READ COMMITTED

3.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002	
4.		UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 11000 WHERE ACCOUNT_NUMBER = 11110002 CZEKA
5.	COMMIT 12000	CZEKA
6.		UPDATE
7.		COMMIT 11000
8.	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 11000	

- Wnioski: izolacja READ_COMMITTED nie chroni przed anomalią “lost updates”. W tym eksperymencie widać też działanie blokady na modyfikowanym rekordzie – transakcja 2 musi zaczekać, aż transakcja 1 skończy działania na rekordzie (COMMIT), żeby móc przeprowadzić swoje operacje.

e. Działanie klauzuli FOR UPDATE

- Cel: zbadanie działania klauzuli FOR UPDATE.
- Przebieg:

CZAS	Sesja 1	Sesja 2
1.	SET TRANSACTION ISOLATION LEVEL READ COMMITTED	
2.	SELECT ACCOUNT_NUMBER, ACCOUNT_BALANCE	SET TRANSACTION ISOLATION LEVEL READ COMMITTED

	FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 FOR UPDATE			
3.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002		SELECT ACCOUNT_NUMBER, ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 FOR UPDATE	CZEKA
4.	COMMIT	12000		CZEKA
5.			SELECT	12000

- Wnioski: klauzula FOR UPDATE zapobiega anomalii „lost updates”, ponieważ wstrzymuje operację SELECT ... FOR UPDATE we wszystkich innych transakcjach odwołujących się do tego samego rekordu. Powoduje to, że pozostałe transakcje „dostaną” dane dopiero po commicie, czyli „będą świadome” zmian poczynionych w tym czasie przez transakcję 1.

2. Izolacja SERIALIZABLE

Ta izolacja jest bardziej restrykcyjna niż READ_COMMITTED, więc wszystkie anomalie, które nie występują w przypadku poprzednio rozpatrywanej izolacji, nie będą występować również tutaj (w tym przypadku „dirty read”). W związku z tym nie będą rozpatrywane poniżej.

a. Non-repeatable read

- Cel: sprawdzenie, czy izolacja SERIALIZABLE chroni przed anomalią „non-repeatable read”.

- Przebieg:

CZAS	Sesja 1	Sesja 2
1.	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
2.	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000
3.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002	
4.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000
5.	COMMIT 12000	
6.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 10000
7.		COMMIT
8.		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 11110002 12000

- Wnioski: izolacja SERIALIZABLE zapobiega anomalii „non-repeatable read”, poddtrzymując stan bazy z momentu rozpczęcia transakcji aż do momentu zakończenia transakcji.

b. Phantom read

- Cel: sprawdzenie, czy izolacja SERIALIZABLE chroni przed anomalią „phantom read”
- Przebieg:

CZAS	Sesja 1		Sesja 2	
1.	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
2.	SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS	10000	SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS	10000
3.	INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES (SEQ_ACCOUNT_NUMBER.NEXTVAL, 500);			
4.	SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS	10500	SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS	10000
5.	COMMIT			
6.			SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS	10000
7.			COMMIT	
8.			SELECT SUM(ACCOUNT_BALANCE) FROM ACCOUNTS	10500

- Wnioski: izolacja SERIALIZABLE zapobiega anomalii „phantom read” w ten sam sposób, jak przy anomalii „non-repeatable read”.

c. Lost update

- Cel: sprawdzenie, czy izolacja SERIALIZABLE chroni przed anomalią „lost updates”.

- Przebieg:

CZAS	Sesja 1	Sesja 2
1.		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
2.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002	
3.		UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 11000 WHERE ACCOUNT_NUMBER = 11110002 CZEKA
4.	COMMIT 12000	ERROR: CAN'T SERIALIZE ACCESS FOR THIS TRANSACTION
5.	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
6.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 13000 WHERE ACCOUNT_NUMBER = 11110002	
7.		UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 11000 WHERE ACCOUNT_NUMBER = 11110002 CZEKA
8.	ROLLBACK	CZEKA
9.		COMMIT 11000

- Wnioski: izolacja SERIALIZABLE chroni przed anomalią „lost updates”, uwzględniając także możliwość rollbacku po stronie transakcji blokującej.

3. Izolacja READ ONLY

Ta izolacja ma wszystkie restrykcje, co izolacja SERIALIZABLE, więc również nie będą występować w niej te wszystkie anomalie, którym zapobiega ta ostatnia.

a. Zapis

- Cel: sprawdzenie, czy izolacja READ ONLY zabezpiecza przed zapisem z transakcji o tym poziomie izolacji.
- Przebieg:

CZAS	Sesja 1		Sesja 2
1.	SET TRANSACTION READ ONLY		
2.	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 12000 WHERE ACCOUNT_NUMBER = 11110002	ERROR: MAY NOT PERFORM INSERT/ DELETE/ UPDATE OPERATION INSIDE A READ ONLY TRANSACTION	

- Wnioski: izolacja READ ONLY w istocie zabezpiecza przed zapisem.