

## 1. Perdita di Aggiornamento (Lost Update)

**Descrizione:** La perdita di aggiornamento si verifica quando due o più transazioni leggono lo stesso dato e poi lo aggiornano in modo concorrente, ma solo una delle modifiche viene mantenuta, causando la "perdita" dell'aggiornamento effettuato da una delle transazioni.

### Esempio:

-- Transazione 1

```
BEGIN;  
SELECT quantità FROM prodotti WHERE id = 'ABC' FOR UPDATE;  
UPDATE prodotti SET quantità = quantità - 10 WHERE id = 'ABC';  
COMMIT;
```

-- Transazione 2

```
BEGIN;  
SELECT quantità FROM prodotti WHERE id = 'ABC' FOR UPDATE;  
UPDATE prodotti SET quantità = quantità - 5 WHERE id = 'ABC';  
COMMIT;
```

Il tipo di isolamento minimo che si deve dichiarare per evitare l'anomalia della perdita di aggiornamento è il `REPEATABLE READ` e va dichiarato per entrambe le transazioni.

Quindi diventa:

-- Transazione 1

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
SELECT quantità FROM prodotti WHERE id = 'ABC' FOR UPDATE;  
UPDATE prodotti SET quantità = quantità - 10 WHERE id = 'ABC';  
COMMIT;
```

-- Transazione 2

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
SELECT quantità FROM prodotti WHERE id = 'ABC' FOR UPDATE;  
UPDATE prodotti SET quantità = quantità - 5 WHERE id = 'ABC';  
COMMIT;
```

## 2. Lettura Sporca (Dirty Read)

**Descrizione:** Una lettura sporca si verifica quando una transazione legge un dato modificato da un'altra transazione che non è stata ancora confermata.

### Esempio:

-- Transazione 1

BEGIN;

UPDATE conto SET saldo = saldo + 200 WHERE id = 456;

SELECT nome FROM conto WHERE id = 456;

UPDATE conto SET ultima\_operazione = CURRENT\_TIMESTAMP WHERE id = 456;

COMMIT;

-- Transazione 2

BEGIN;

SELECT \* FROM conto WHERE id = 456;

COMMIT;

Il tipo di isolamento minimo per evitare l'anomalia della lettura sporca è il `READ COMMITTED` che non va dichiarato in PostgreSQL perché è il livello di isolamento utilizzato di default.

### 3. Letture Inconsistenti (Non-repeatable Read)

**Descrizione:** Una lettura inconsistente si verifica quando una transazione legge più volte lo stesso dato, ma ottiene risultati diversi a causa di aggiornamenti concorrenti effettuati da altre transazioni tra le due letture.

#### Esempio:

-- Transazione 1

```
BEGIN;  
SELECT saldo FROM conto WHERE id = 789;  
SELECT saldo FROM conto WHERE id = 789;  
COMMIT;
```

-- Transazione 2

```
BEGIN;  
UPDATE conto SET saldo = 1100 WHERE id = 789;  
COMMIT;
```

Il tipo di isolamento minimo che si deve dichiarare per evitare l'anomalia della lettura inconsistente è il `REPEATABLE READ` che va dichiarato nella prima transazione.

Quindi diventa:

-- Transazione 1

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
SELECT saldo FROM conto WHERE id = 789;  
SELECT saldo FROM conto WHERE id = 789;  
COMMIT;
```

-- Transazione 2

```
BEGIN;  
UPDATE conto SET saldo = 1100 WHERE id = 789;  
COMMIT;
```

## 4. Aggiornamento Fantasma (Phantom Read)

**Descrizione:** L'aggiornamento fantasma si verifica quando una delle due transazioni lavora su alcune righe che l'altra transazione sta aggiornando senza un'adeguata sincronizzazione. Questo può portare a inconsistenze nei dati.

### Esempio:

```
-- Transazione 1
BEGIN;
UPDATE mostra SET prezzoridotto = prezzoridotto * 1.1 WHERE citta ILIKE 'Verona';
COMMIT;
```

```
-- Transazione 2
BEGIN;
SELECT AVG(prezzointero) FROM mostra WHERE citta ILIKE 'Verona';
SELECT AVG(prezzoridotto) FROM mostra WHERE citta ILIKE 'Verona';
COMMIT;
```

Il tipo di isolamento minimo che si deve dichiarare per evitare l'anomalia dell'aggiornamento fantasma è il `SERIALIZABLE` oppure il `REPEATABLE READ` essendo che in PostgreSQL è più restrittivo rispetto a quello standard, va dichiarato nella seconda transazione.

Quindi diventa:

```
-- Transazione 1
BEGIN;
UPDATE mostra SET prezzoridotto = prezzoridotto * 1.1 WHERE citta ILIKE 'Verona';
COMMIT;
```

```
-- Transazione 2
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT AVG(prezzointero) FROM mostra WHERE citta ILIKE 'Verona';
SELECT AVG(prezzoridotto) FROM mostra WHERE citta ILIKE 'Verona';
COMMIT;
```

## 5. Inserimento Fantasma (Phantom Insert)

**Descrizione:** L'anomalia dell'inserimento fantasma si verifica quando, durante l'esecuzione concorrente di due transazioni, una delle due inserisce nuovi record che influenzano il risultato dell'altra. Questo porta a risultati incoerenti o inattesi nelle operazioni sui dati.

### Esempio:

```
-- Transazione 1
BEGIN;
INSERT INTO mostra (titolo, inizio, fine, museo, citta, prezzointero, prezzoridotto)
VALUES ('Nuova Mostra', '2024-07-15', '2024-08-15', 'Museo di Verona', 'Verona', 15, 10);
COMMIT;
```

```
-- Transazione 2
BEGIN;
SELECT AVG(prezzointero) FROM mostra WHERE citta ILIKE 'Verona';
SELECT AVG(prezzoridotto) FROM mostra WHERE citta ILIKE 'Verona';
COMMIT;
```

Il tipo di isolamento minimo che si deve dichiarare per evitare l'anomalia dell'inserimento fantasma è il `SERIALIZABLE` oppure il `REPEATABLE READ` essendo che in PostgreSQL è più restrittivo rispetto a quello standard, va dichiarato nella seconda transazione.

Quindi diventa:

```
-- Transazione 1
BEGIN;
INSERT INTO mostra (titolo, inizio, fine, museo, citta, prezzointero, prezzoridotto)
VALUES ('Nuova Mostra', '2024-07-15', '2024-08-15', 'Museo di Verona', 'Verona', 15, 10);
COMMIT;
```

```
-- Transazione 2
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT AVG(prezzointero) FROM mostra WHERE citta ILIKE 'Verona';
SELECT AVG(prezzoridotto) FROM mostra WHERE citta ILIKE 'Verona';
COMMIT;
```

## 6. Mancata Serializzazione (Serialization Anomaly)

**Descrizione:** L'anomalia della mancata serializzazione è un'anomalia che si può verificare in PostgreSQL. Questa si presenta quando il risultato di un insieme di transazioni, tutte completate con successo, è inconsistente con tutti gli ordini di esecuzioni seriali delle stesse.

### Esempio:

```
-- Transazione 1
BEGIN;
SELECT saldo FROM conti WHERE conto = 'A';
UPDATE conti SET saldo = saldo + 50 WHERE conto = 'A';
COMMIT;

-- Transazione 2
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT saldo FROM conti WHERE conto = 'A';
SELECT saldo FROM conti WHERE conto = 'B';
UPDATE conti SET saldo = saldo - 50 WHERE conto = 'A';
UPDATE conti SET saldo = saldo + 50 WHERE conto = 'B';
COMMIT;
```

Il tipo di isolamento minimo che si deve dichiarare per evitare l'anomalia della mancata serializzazione è il `SERIALIZABLE` che va dichiarato nella seconda transazione.

Quindi diventa:

```
-- Transazione 1
BEGIN;
SELECT saldo FROM conti WHERE conto = 'A';
UPDATE conti SET saldo = saldo + 50 WHERE conto = 'A';
COMMIT;

-- Transazione 2
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT saldo FROM conti WHERE conto = 'A';
SELECT saldo FROM conti WHERE conto = 'B';
UPDATE conti SET saldo = saldo - 50 WHERE conto = 'A';
UPDATE conti SET saldo = saldo + 50 WHERE conto = 'B';
COMMIT;
```