ORACLE

Isolation

Mag. Thomas Griesmayer

concurrency consistency

- Data concurrency: Multiple users can access data at the same time.
- Data consistency: Every user sees always a consistent view of the data. This includes changes made by the user's own transactions and committed transactions of other users.

Preventable phenomena

- Dirty reads - A transaction reads data that has been written by another transaction that has not been committed yet.
- Nonrepeatable (fuzzy) reads - A transaction rereads data it has previously read and finds that another committed transaction has modified or deleted the data.
- Phantom reads - A transaction reruns a query returning a set of rows that satisfies a search condition and finds that another committed transaction has inserted additional rows that satisfy the condition.

http://docs.oracle.com/cd/E11882_01/server.112/e40540/consist.htm (30.9.2019)

dirty read

```
UPDATE CUSTOMER
SET    BALANCE = -100
WHERE  CUSTOMER_ID = 4;
```

C_ID	NAME	BALANCE
1	Fritz	€ 800
2	Susi	€ 1000
3	Werner	€ -200
4	Hans	€ 0
5	Alex	€ 400
6	Thomas	€ 100

```
SELECT *
FROM   CUSTOMER
WHERE  CUSTOMER_ID = 4;
```

C_ID	NAME	BALANCE
4	Hans	€ -100

```
SELECT SUM(BALANCE) as BALANCE
FROM   CUSTOMER;
```

BALANCE
€ 2000

fuzzy read

```
SELECT *
FROM   CUSTOMER
WHERE  CUSTOMER_ID IN (4,5);
```

C_ID	NAME	BALANCE
4	Hans	€ 0
5	Alex	€ 400

```
SELECT *
FROM   CUSTOMER
WHERE  CUSTOMER_ID IN (4,5);
```

C_ID	NAME	BALANCE
4	Hans	€ -100

```
UPDATE CUSTOMER
SET    BALANCE = -100
WHERE  CUSTOMER_ID = 4;

DELETE FROM CUSTOMER
WHERE  CUSTOMER_ID = 5;
COMMIT;
```

C_ID	NAME	BALANCE
1	Fritz	€ 800
2	Susi	€ 1000
3	Werner	€ -200
4	Hans	€ 0
5	Alex	€ 400
6	Thomas	€ 100

phantom read

```
SELECT count(*) as NUMBERLINES,
       sum(BALANCE) as BALANCE
FROM   CUSTOMER
```

NUMBERLINES	BALANCE
6	€ 2100

```
INSERT INTO customer
VALUES (7,'Max',300);
COMMIT;
```

```
SELECT count(*),
       sum(balance)
FROM   customer
```

NUMBERLINES	BALANCE
7	€ 2400

C_ID	NAME	BALANCE
1	Fritz	€ 800
2	Susi	€ 1000
3	Werner	€ -200
4	Hans	€ 0
5	Alex	€ 400
6	Thomas	€ 100

Isolation levels

- Oracle Database provides the transaction isolation levels:
  - Read Committed Isolation Level
  - Serializable Isolation Level
  - Read-Only Isolation Level

http://docs.oracle.com/cd/E11882_01/server.112/e40540/consist.htm (30.9.2019)

Read committed

- The read committed isolation level is the Oracle default isolation level.
- Every query executed by a transaction sees only data committed before the query - and NOT the TRANSACTION - began.
- This level of isolation is appropriate for database environments in which few transactions are likely to conflict.
- Possible:
  - ~~dirty reads~~
  - fuzzy reads
  - phantom reads

http://docs.oracle.com/cd/E11882_01/server.112/e40540/consist.htm (30.9.2019)

Serializable

- Every query executed by a transaction sees only data committed before the transaction - and NOT the QUERY - began.
- A serializable transaction operates in an environment that makes it appear as if no other users were modifying data in the database.
- The database generates an error when a serializable transaction tries to update or delete data changed by a different transaction that committed after the serializable transaction began:

```
ORA-08177: Cannot serialize access for this transaction
```

- Possible:
  - ~~dirty reads~~
  - ~~fuzzy reads~~
  - ~~phantom reads~~

https://en.wikipedia.org/wiki/Bitmap_index (15.9.2019)

Read-only

- The read-only isolation level is similar to the serializable isolation level, but read-only transactions do not permit data to be modified in the transaction unless the user is `SYS`.
- Read-only transactions are useful for generating reports in which the contents must be consistent with respect to the time when the transaction began.

https://en.wikipedia.org/wiki/Bitmap_index (15.9.2019)