```
package database;

import transactions.Transaction;

public class Item implements DbItem{
    private String content;
    private final String lockedMessage = "the content of that item is locked for read or read and write.";
    private final String unlockedMenssage = "that item is not locked. please lock it before accessing.";
    private final String unlockedItemMessage = "unlocked item.";
    private final String lockedItemMessage = "locked item.";
    private final String alreadyLockedItemMessage = "that item is already locked by another transaction.";
    private final String columnName;
    private final long line;
    private boolean isReadLocked = false;
    private boolean isWriteLocked = false;
    private Transaction currentAcessingTransaction;

    public Item( String columnName, long line, String content){
        this.content = content;
        this.columnName = columnName;
        this.line = line;
    }

    @Override
    public boolean isReadLocked() {
        return this.isReadLocked;
    }

    @Override
    public boolean isWriteLocked() { return this.isWriteLocked; }

    @Override
    public String read(Transaction transaction) {
        if( this.isLocked() ){
            if( this.isWriteLocked && !isCurrentAcessingTransaction(transaction) ){
                return this.lockedMessage;
            }
            return this.content;
        }
        return this.unlockedMenssage;
    }

    @Override
    public void write(Transaction transaction, String content) {
        if( this.isLocked() ){
            if( !isCurrentAcessingTransaction(transaction) )
                System.out.println(this.lockedMessage);
            this.content = content;
            return;
        }
        System.out.println(this.unlockedMenssage);
    }

    @Override
    public boolean read_lock(Transaction transaction) {
        if( this.isLocked()) {
            System.out.println(alreadyLockedItemMessage);
            return false;
        }
        this.currentAcessingTransaction = transaction;
```

```
        this.isReadLocked = true;
        this.isWriteLocked = false;
        return true;
    }

    @Override
    public boolean write_lock(Transaction transaction) {
        if( this.isLocked() ) {
            return false;
        }
        this.currentAcessingTransaction = transaction;
        this.isReadLocked = false;
        this.isWriteLocked = true;
        return true;
    }

    @Override
    public boolean isLocked() {
        if( this.isWriteLocked || this.isReadLocked )
            return true;
        return false;
    }

    @Override
    public void unLock(Transaction transaction) {
        if( !this.isLocked() || !this.isCurrentAcessingTransaction(transaction)) {
            System.out.println(unlockedMenssage);
            return;
        }
        this.currentAcessingTransaction = null;
        this.isReadLocked = false;
        this.isWriteLocked = false;
    }

    @Override
    public String getPosition() {
        return "Column:" + this.columnName + " Line:" + this.line;
    }

    private boolean isCurrentAcessingTransaction(Transaction transaction ){
        if( transaction != this.currentAcessingTransaction)
            return false;
        return true;
    }
}
```