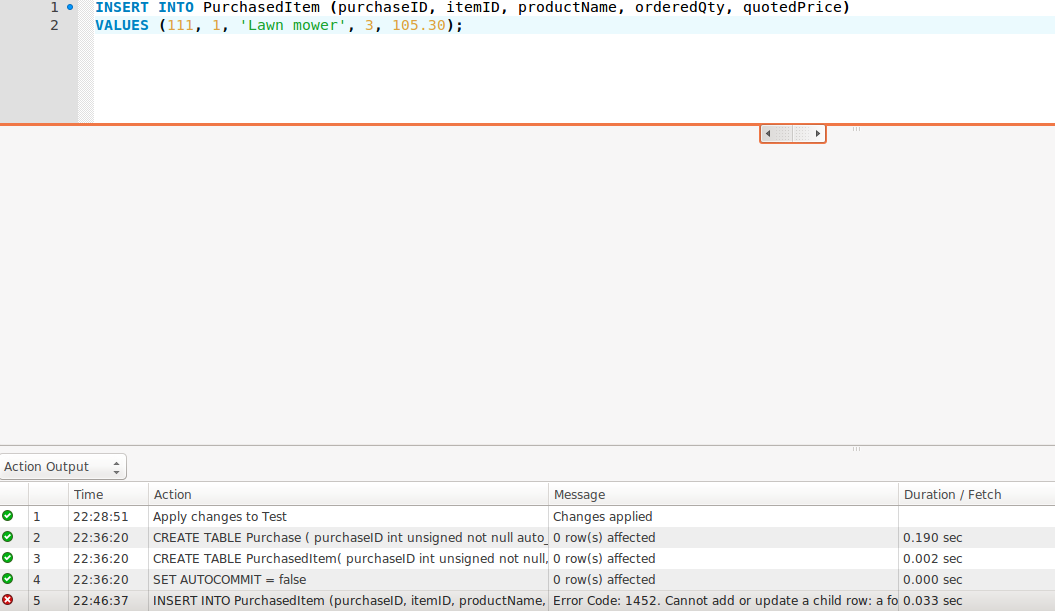
Pass Task 8.1 report

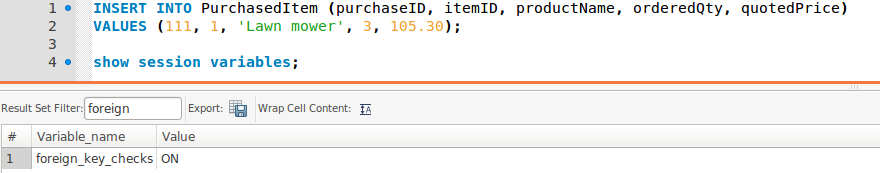
|  |  |
| --- | --- |
| Student Name | Lim Jia Lok |
| Student ID | 101212631 |

## SUB Task 8.1.1

#### Solution

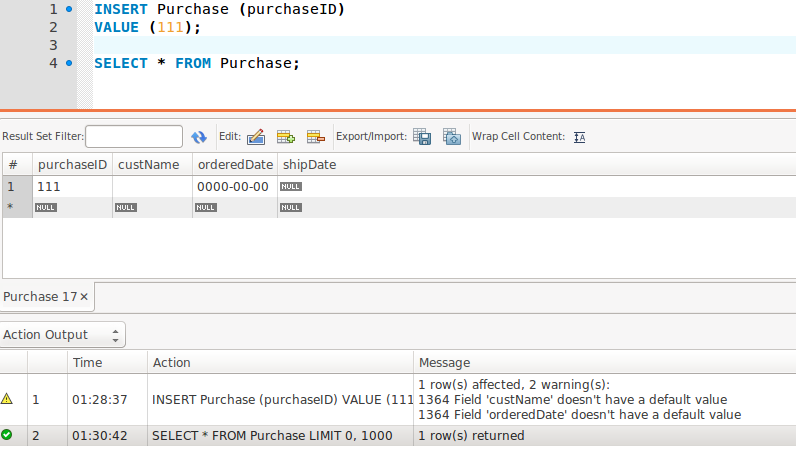
**A. FAILED**



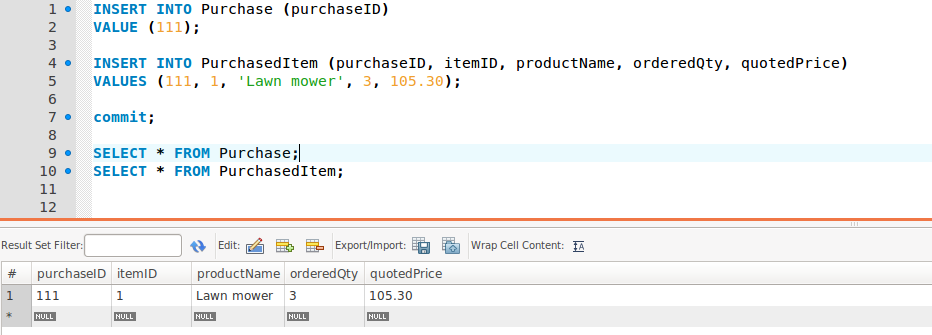


**B. SUCESSFUL**

**1. Insert purchaseID =111 into Purchase table**

****

**2. Inserting the values into PurchasedItem Table**

****

#### Comments/findings

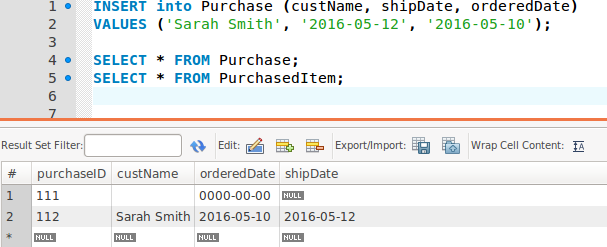
**A.** **The SQL command does not work because the table is a child table. This means that the purchaseID attribute for PurchasedItem is a foreign key that references the table Purchase that is the parent table. Since the purchaseID of 111 is not inserted into the Purchase table, the purchaseID in table PurchasedItem cannot reference it. Thus, the command does not work since the purchaseID in PurchasedItem did not match any purchaseID in Purchase table. This rule applies when the variable foreign\_key\_checks is ON.**

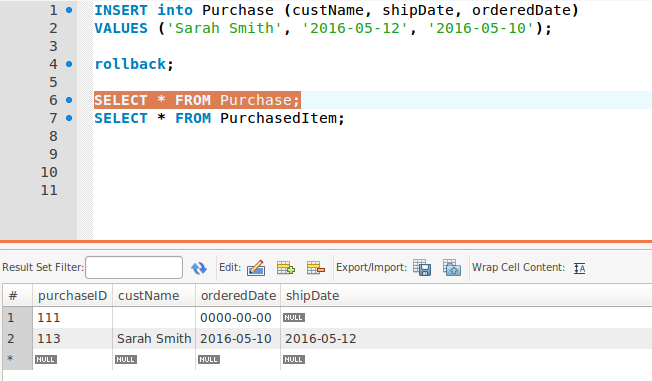
**B.** **This can be fix by inserting purchaseID = 111 into Purchase table to create a primary key that it can be reference back to (foreign key) in PurchasedItem table.**

## SUB Task 8.1.2

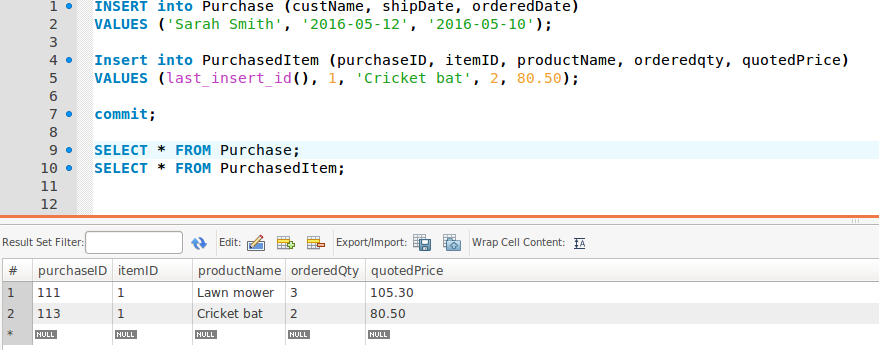
#### Solution

**A. Inserting the SQL and rolling back**





**B. Using last\_insert\_id() as a value for purchaseID in PurchasedItem table**



#### Comments/findings

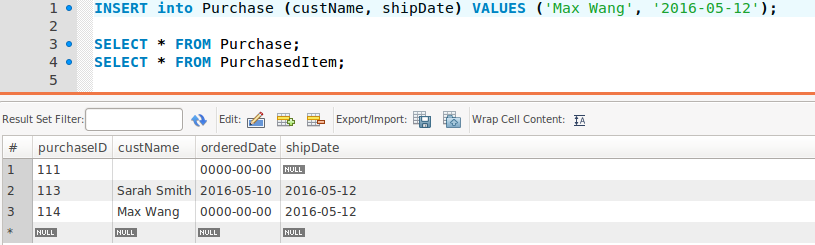
**A. After inserting the SQL statement, we can see that the purchasedID for Purchase table is 112. After rolling back, and inserting the same statement again, the purchaseID becomes 113. This is because of the auto-increment statement that we used for purchaseID in Purchase table during its creation. When we roll back, we could not get the 112 again because the auto-increment causes the 112 to already been used so it increment by 1 instead which becomes 113. The rollback does not affect the auto increment.**

**B. To insert SQL statement into table PurchasedItem table,? value should have been replaced by last\_insert\_id() which takes in the last generated ID for auto-increment. So, it takes in the 113 and becomes purchaseID for that statement. Essentially, it becomes a foreign key that references Sarah Smith.**

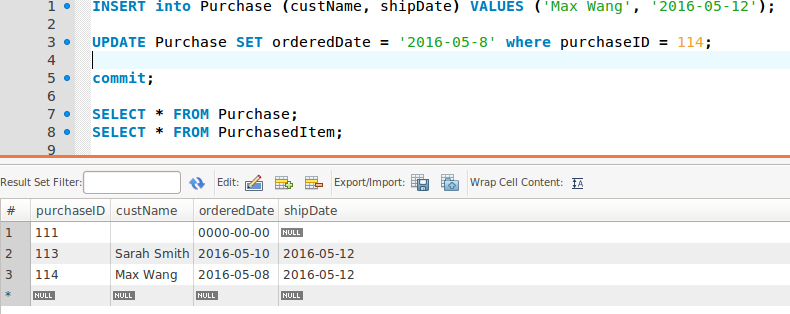
## SUB Task 8.1.3

#### Solution

**A. The orderedDate is defaulted to 0000-00-00.**



**B**. **Updating the Purchase table to include an orderedDate for purchasedID= 114**



#### Comments/findings

**A. Since we declared that the orderedDate attribute cannot be null, it defaults to 0000-00-00 as we do not declare an orderDate in the statement.**

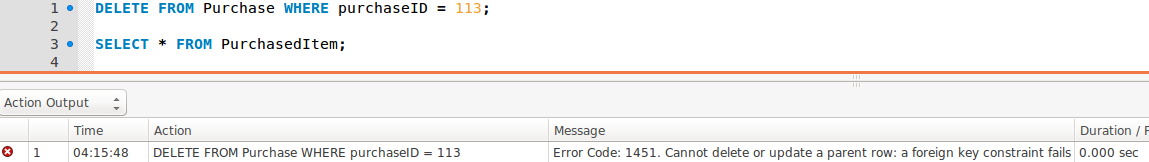
**B. To rectify this problem, we use the update function to insert orderedDate = ‘2016-05-08’ into the database using set function by specifying the purchasedID = 114 using where function.**

(Where applicable)

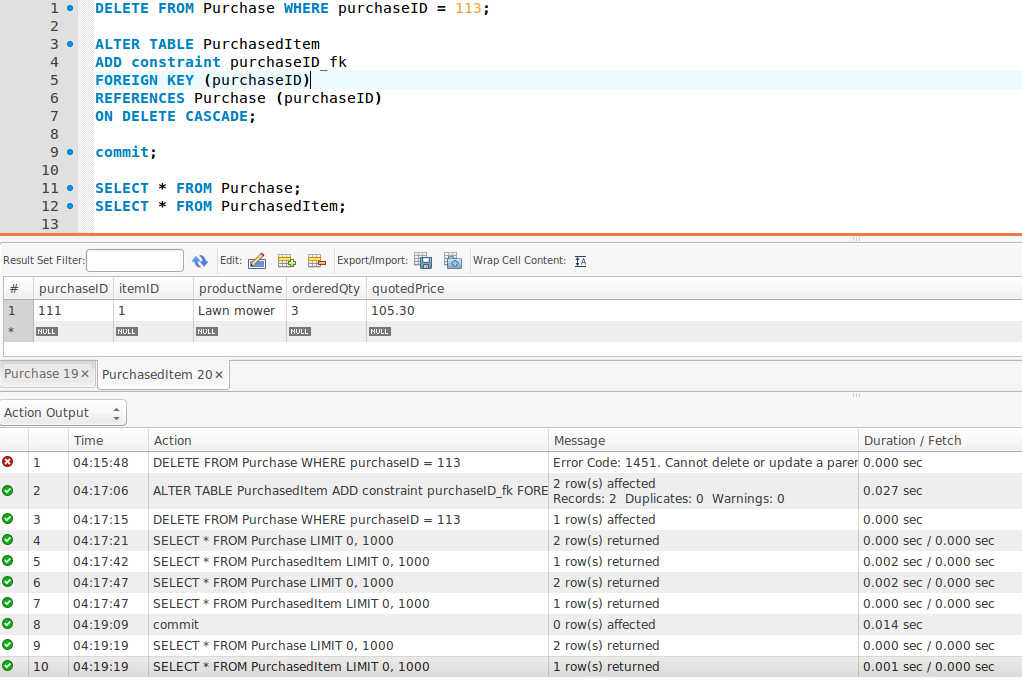
## SUB Task 8.1.4

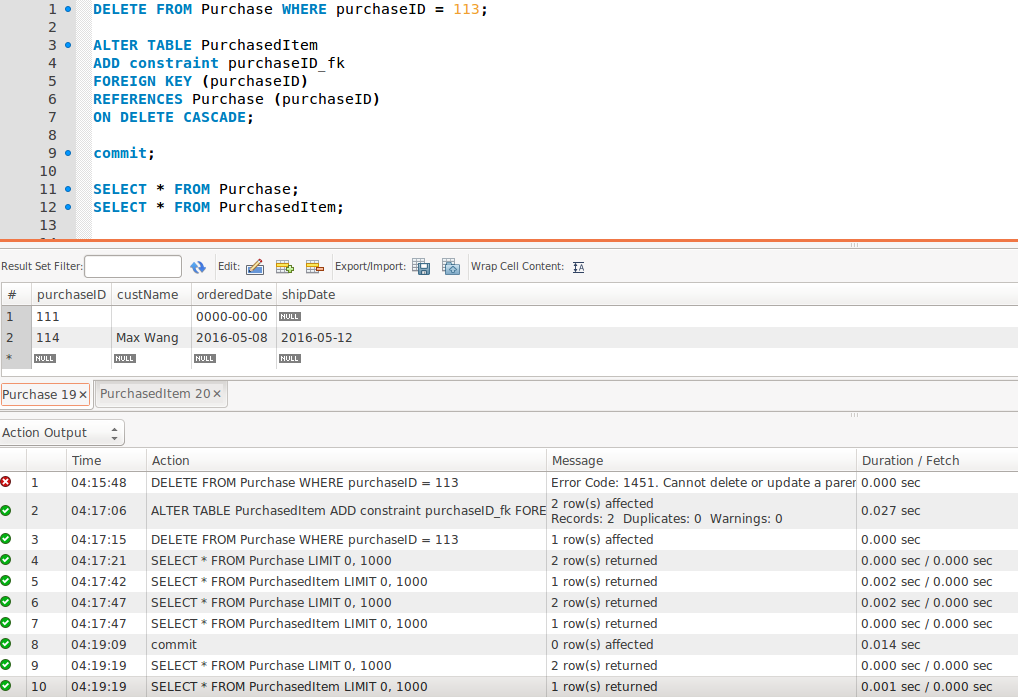
#### Solution

**A. Deletion failed**

****

**B. Deletion successful using on delete cascade**

****

****

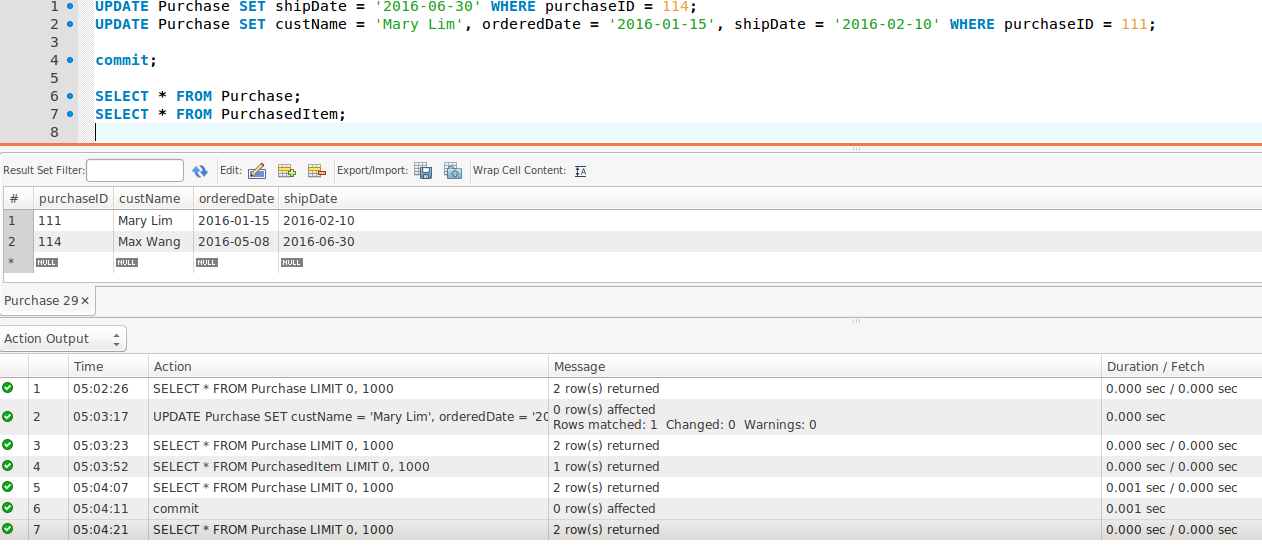
#### Comments/findings

**A. The deletion fails because the row we wanted to delete with the purchaseID =113(Primary Key) in Purchase table is referenced by a child row in PurchasedItem table with the same purchaseID = 113(foreign key).**

**B. To delete it, we need to use on delete cascade to delete the parent row together with its child row. It will specify that the child data is deleted when the parent data is deleted.**

## SUB Task 8.1.5

#### Solution



#### Comments/findings

**Using update function, we can set two attributes using one sql statement by including a comma in between.**