**Database Management Quick Start:**

**Run the Database Manager:**

powershell

.\Manage-PurchaseDatabase.ps1 -DatabasePath "C:\Users\JohnO'NeillSr\OneDrive - AwesomeWildStuff.com\JT Custom Trailers\Auctions\purchasesdata.db"

This gives you an interactive menu with all these options:

1. **View/Search items**
2. **Edit individual prices** (including "Our Price")
3. **Bulk price adjustments**
4. **Export to Excel** → Edit → Import back
5. **Mark items as sold**
6. **Run custom SQL queries**

**Common Tasks:**

**1. Override a Single Price:**

powershell

*# In the menu, select option 3, then:*

*# - Choose the ItemID (e.g., 1 for first microwave)*

*# - Select which price to change (1 for "Our Asking Price")*

*# - Enter new price (e.g., 175)*

**2. Quick SQL Commands (Alternative Method):**

If you prefer direct SQL, run with -DirectSQL flag:

powershell

.\Manage-PurchaseDatabase.ps1 -DatabasePath ".\purchasesdata.db" -DirectSQL

Then use queries like:

sql

*-- View all items with prices*

SELECT ItemID, Description, PerItemCost, SuggestedPrice FROM Items

*-- Update "Our Price" for specific item*

UPDATE Items SET SuggestedPrice = 175 WHERE ItemID = 1

*-- Increase all prices by 10%*

UPDATE Items SET SuggestedPrice = SuggestedPrice \* 1.10

*-- Set minimum $100 profit on all items*

UPDATE Items SET SuggestedPrice = PerItemCost + 100 WHERE (SuggestedPrice - PerItemCost) < 100

**3. Excel Method (Best for Bulk Edits):**

1. Export your data: Menu option 6
2. Open in Excel
3. Edit the "Our Asking Price" column
4. Save the CSV
5. Import back: Menu option 7

**Examples:**

Based on your data, here are some specific commands:

sql

*-- Change Greystone microwave price to $175*

UPDATE Items SET SuggestedPrice = 175 WHERE Description LIKE '%Greystone microwave%'

*-- Set all refrigerators to $450*

UPDATE Items SET SuggestedPrice = 450 WHERE Description LIKE '%Refrigerator%'

*-- Add 20% to all current prices*

UPDATE Items SET SuggestedPrice = ROUND(SuggestedPrice \* 1.20, 2)

*-- Set minimum acceptable prices at 50% above cost*

UPDATE Items SET MinAcceptablePrice = ROUND(PerItemCost \* 1.5, 2)

**Pro Tips:**

**Price Strategy Overrides:**

sql

*-- Premium items (high market value)*

UPDATE Items

SET SuggestedPrice = ROUND(CurrentMarketAvg \* 0.90, 2)

WHERE CurrentMarketAvg > 500

*-- Quick movers (lower margin for volume)*

UPDATE Items

SET SuggestedPrice = ROUND(PerItemCost \* 2.5, 2)

WHERE Description LIKE '%mattress%' OR Description LIKE '%microwave%'

*-- Round all prices to nearest $5 for cleaner pricing*

UPDATE Items

SET SuggestedPrice = ROUND(SuggestedPrice / 5.0) \* 5

WHERE SuggestedPrice > 0

**View Current Pricing:**

sql

SELECT

ItemID,

SUBSTR(Description, 1, 30) as Item,

PerItemCost as Cost,

SuggestedPrice as OurPrice,

ROUND((SuggestedPrice/PerItemCost - 1) \* 100, 1) as Margin

FROM Items

WHERE IsActive = 1

ORDER BY ItemID

**Quick Reference Commands:**

| **Task** | **SQL Command** |
| --- | --- |
| **Change one price** | UPDATE Items SET SuggestedPrice = 200 WHERE ItemID = 1 |
| **Increase all by 10%** | UPDATE Items SET SuggestedPrice = SuggestedPrice \* 1.10 |
| **Set minimum margin** | UPDATE Items SET SuggestedPrice = PerItemCost \* 3 WHERE SuggestedPrice < PerItemCost \* 3 |
| **Round to $5** | UPDATE Items SET SuggestedPrice = ROUND(SuggestedPrice/5)\*5 |
| **Mark as sold** | UPDATE Items SET SoldPrice = 180, IsActive = 0 WHERE ItemID = 1 |

**Most Common Use Case:**

Overriding a single asking price is easy:

1. Run the manager script
2. Choose option 3 (Edit single item prices)
3. Pick the item by ID
4. Choose 1 (Our Asking Price)
5. Enter the new price

The database manager makes it super easy to maintain full control over pricing while still benefiting from automated market research