



Overview: In this part, the warehouse workers will use their mobile computer devices to pick parts for an order. They will use their device to navigate to the Picking Window and it will show them what areas need to be picked. From there, they will scan the shelf location, product, quantity, and the tote's location.

Note: This is an extension to Implementation 2: Shipping Customer Orders, so use that pdf file as a reference for this part. This demonstration is to show how I will use Pandas through Python to interact with my Oracle's database.

Note: Input\_Commands.py and Oracle\_Connection.py are the files used.

Step 1: Once the user has logged into their devices, they will have access to the database. From there they will navigate to the "Start Picking" window

```
Application Started
1. Start Picking
2. Stage Items
3. Process Orders
4. Move Parts
5. Process Receiving Items
0. Exit Program
Enter your Command: 1
```

Step 2: From there, the worker will choose what zones need picking. I have created multiple orders at this point, so there are multiple orders that need picking. The value for "Picks:" is to show how many parts in quantity need picking. From here, the worker will put in the Zone name ('A', 'C', 'D', 'E', or 'REC') and start picking from there.

```
Zone to Pick
A           Picks: 8
C           Picks: 12
D           Picks: 10
E           Picks: 2
REC        Picks: 4
Pick Bin:
```

Quick Note: As shown in the image below, for bin location 'C', there are currently 12 parts that need to be picked from different orders. You can use this image as a reference to follow the next steps

```

6 SELECT ORDER_ID, PRODUCT_ID, PICK_STATUS, BIN_LOCATION, ZONE_LOCATION,
7       SUM(QUANTITY) OVER(ORDER BY PRODUCT_ID ROWS UNBOUNDED PRECEDING) total_qty
8 FROM PICKS
9 WHERE BIN_LOCATION = 'C' AND PICK_STATUS = 'N';
0

```

Script Output x Query Result x

SQL | All Rows Fetched: 12 in 0.003 seconds

	ORDER_ID	PRODUCT_ID	PICK_STATUS	BIN_LOCATION	ZONE_LOCATION	TOTAL_QTY
1	15	9	N	C	C101	1
2	15	9	N	C	C101	2
3	13	10	N	C	C102	3
4	16	10	N	C	C102	4
5	16	10	N	C	C102	5
6	7	11	N	C	C103	6
7	13	11	N	C	C103	7
8	13	11	N	C	C103	8
9	11	12	N	C	C104	9
10	11	12	N	C	C104	10
11	11	12	N	C	C104	11
12	11	13	N	C	C105	12

Step 3: Let's assume the worker wants to do Zone 'C', then they will type in the zone name and be prompted what tote these items will be placed in. Let's assume the Worker picked a Box Tote labeled as 'PICK GREY\_TOTE\_101', then they will input that value in there as such

```

Pick Bin: C
Enter Tote Location: PICK GREY TOTE 101

```

From here, the worker will go to the location and scan the shelf's location to match the area they are being sent to.

```

Go to Area: C101
Scan Location: C101
Correct Location

```

Then, the user must pick the part they need and input the product's id once they've made sure they have the correct part. Each time they type in the product's id, then that indicates 1 part has been picked.

```

Product ID: 9
Scan Item: 9
Correct Part Scanned
Product ID: 9

```

From here, the user will keep on inputting the part's id until they are assigned to the next zone location. As shown below, they will go to Zone 'C102' next and repeat the process.

```

Correct Location
Product ID: 9
Scan Item: 9
Correct Part Scanned
Product ID: 9
Scan Item: 9
Correct Part Scanned
Go to Area: C102
Scan Location:

```

As a quick demonstration, up to this point, we have scan 2 parts of product id: 9 into grey\_tote\_101. As shown below you now see that these parts have been moved to a new location.

```

2 SELECT *
3 FROM ORDER_LIST
4 WHERE ZONE = 'GREY_TOTE_101';

```

ID	ORDER_ID	PRODUCT_ID	UNITS	BIN	ZONE
1	53	15	9	1 PICK	GREY_TOTE_101
2	54	15	9	1 PICK	GREY_TOTE_101

Step 4: Once the workers have gone through all their picks for Zone 'C', finally the PICKS table will be updated one last time. As shown below, there are no parts that need to be picked in Location 'C'.

```

SELECT ORDER_ID, PRODUCT_ID, PICK_STATUS, BIN_LOCATION, ZONE_LOCATION,
       SUM(QUANTITY) OVER(ORDER BY PRODUCT_ID ROWS UNBOUNDED PRECEDING) total_qty
FROM PICKS
WHERE BIN_LOCATION = 'C' AND PICK_STATUS = 'N';

```

ORDER_ID	PRODUCT...	PICK_STA...	BIN_LOC...	ZONE_LO...	TOTAL_QTY
----------	------------	-------------	------------	------------	-----------

For comparison, the image below shows the items that needs picking

```

6 SELECT ORDER_ID, PRODUCT_ID, PICK_STATUS, BIN_LOCATION, ZONE_LOCATION,
7       SUM(QUANTITY) OVER(ORDER BY PRODUCT_ID ROWS UNBOUNDED PRECEDING) total_qty
8 FROM PICKS
9 WHERE BIN_LOCATION = 'C' AND PICK_STATUS = 'N';
0

```

ORDER_ID	PRODUCT_ID	PICK_STATUS	BIN_LOCATION	ZONE_LOCATION	TOTAL_QTY
1	15	9 N	C	C101	1
2	15	9 N	C	C101	2
3	13	10 N	C	C102	3
4	16	10 N	C	C102	4
5	16	10 N	C	C102	5
6	7	11 N	C	C103	6
7	13	11 N	C	C103	7
8	13	11 N	C	C103	8
9	11	12 N	C	C104	9
10	11	12 N	C	C104	10
11	11	12 N	C	C104	11
12	11	13 N	C	C105	12

This image demonstrates the location for the parts' new location after being picked from the shelf

```

SELECT ORDER_ID, PRODUCT_ID, SUM(UNITS) QTY,
       SUM(SUM(UNITS)) OVER(ORDER BY
       ORDER_ID ROWS UNBOUNDED PRECEDING) TOTAL_QTY ,
       BIN, ZONE
FROM ORDER_LIST
WHERE ZONE = 'GREY_TOTE_101'
GROUP BY ORDER_ID, PRODUCT_ID, BIN, ZONE;

```

ORDER_ID	PRODUCT_ID	QTY	TOTAL_QTY	BIN	ZONE
7	11	1	1	PICK GREY_TOTE_101	
11	12	3	4	PICK GREY_TOTE_101	
11	13	1	5	PICK GREY_TOTE_101	
13	10	1	6	PICK GREY_TOTE_101	
13	11	2	8	PICK GREY_TOTE_101	
15	9	2	10	PICK GREY_TOTE_101	
16	10	2	12	PICK GREY_TOTE_101	

Final Note: As a demonstration, the worker will go back to their picking window. You will now see that all parts for Zone 'C' have been picked, and now the worker needs to focus on the other zones.

```

Enter your Command: 1
Picking List
Zone to Pick
A      Picks: 8
D      Picks: 10
E      Picks: 2
REC    Picks: 4
Pick Bin:

```