# **COMP-1701 - Transferring Data to Databases**

## **Data Science & Machine Learning (DSML) - RRC Polytech**

Final Project – Phase 1: Initial

**A computer screen shot of a diagram

Description automatically generatedTask 1.0:**

Run the people meta data loader, insert code directly before the all column query in the people table.

**Task 1.1:**

Find and replace within ERD and your boxstore script to replace occurrences of user\_ and date\_ to user and date.

*Outcome: system fields will have no underscores, ie: usermod, datemod, useract, dateact, active.*

**Task 1.2:**

-- add this code before the all column people  
-- table  
  
UPDATE people SET employee=1 WHERE p\_id BETWEEN 3 AND 10;

Transfer the people\_employee table script to your boxstore script, after the ENVELOPE query (ie: a new table section)

Add the people\_employee table to the boxstore ERD diagram.

After running the final JOIN, explain what might be occurring with our Boxstore Inc. company.

**Task 1.3: Create manufacturer (m) table**

Add a DROP/CREATE/TRUNCATE/INSERT/SELECT (DCTIS) block for the manufacturer (m) table block .

Do not change the people table. Use its CREATE and ALTER code, to develop 1 CREATE statement for the manufacturer table.

First put the manufacturer into the ERD, in the most appropriate spot within, that requires the shortest JOIN lines, mark the keys and cardinalities accordingly.  
  
Replace p\_id with man\_id

Add a column below called p\_id\_man, which will be the manufacturers’ main contact from the people table.

Replace first\_name and last\_name with man\_name VARCHAR(50)

Remove one other field that wouldn’t make much sense in having, as it relates to the people table.

Add a PRIMARY KEY, called man\_\_PK, using the appropriate column.

Add a UNIQUE key, called man\_\_UK, using the appropriate column  
  
Move the following insert data into the DCTIS block (noting that you will be updating this data and at run time):

INSERT INTO manufacturer (man\_name, p\_id\_man  
 , phone\_pri, phone\_sec, phone\_fax  
 , addr\_prefix, addr, addr\_code, addr\_info  
 , addr\_type\_id, tc\_id)

VALUES ( 'Boxstore Inc.', 1  
 , '222-222-2222', '333-333-3333', '444-4444-4444'  
 , 'Floor 1', '3 Road Runner Way', 'ROH HOH', NULL  
 , 3, 1);

INSERT INTO manufacturer   
 (man\_id, man\_name, addr, addr\_info, addr\_type\_id, tc\_id)

VALUES (2,101,'Apple Inc.','260-17 1st St','PO Box 2601',3,7)  
 , (3,202,'Samsung Electronics','221-6 2nd St','PO Box 24',3,5)  
 , (4,303,'Dell Technologies','90-62 3rd St','PO Box 2517',3,9)  
 , (5,404,'Hitachi','88-42 4th St','PO Box 2654',3,2)  
 , (6,505,'Sony','80-92 5th St','PO Box 4017',3,3)  
 , (7,606,'Panasonic','74-73 6th St' ,'PO Box 4958',3,4)  
 , (8,707,'Intel','71-9 7th St','PO Box 2934',3,8)  
 , (9,808,'LG Ele,ctronics','54-39 8th St','PO Box 9824',3,6)  
 , (10,909,'Microsoft','100-10 Ninth St','PO Box: 98245',3,10)  
;

**Task 1.4: Create a VIEW called manufacturer\_people\_reps**

Revamp the people thru geo table JOIN (not the ENVELOPE query), to now include the manufacturer first.

All other tables should show all of their columns, including the manufacturer table (skip the system fields user\* date\* active).

Only include from people table, only show the first\_name, last\_name, email\_addr, and phone\_pri.  
  
Create this into a VIEW and when running the VIEW, show only the “name” fields, street address, po box, city, province/state, country (no abbreviations), and all phone numbers. Order the view results by country, province/state, city, and manufacturer name.

Final Project – Phase 2: Receipt/Sales Logic and Methodology

**Task 2.1: Import orders\_\_item SQL script**

This should be added below the manufacturer table definition section…and the script is in the order of:

* item\_type (it) table
* item (i) table  
    
  After the staged ALL COLUMN query, create a JOIN to both:
  + include all item columns
  + exclude system columns
  + manufacturer for man\_id and man\_name fields
  + item\_type for it\_id and it\_desc fields
* item\_inventory (ii)  
    
  After the staged ALL COLUMN query, create a JOIN :
  + JOIN FROM the item table (include all item cols and all item\_inventory columns, exclude system columns)
* item\_price (ip)
  + no action required yet, other than loading it
* orders (o)  
    
  After the staged ALL COLUMN query, create a JOIN to both:
  + JOIN to people then to people\_employee to get the employee’s full name and their employee number
  + Same query, with a JOIN to get the customer name
* orders\_\_item (oi)  
    
  After the staged ALL COLUMN query, create a JOIN to all:
  + orders table, with all fields (exclude system fields)
  + item table JOIN (that you have written) to manufacturer/item\_type, with all fields (exclude system fields)
  + item\_price table, via JOIN and use order\_date to get the right price (between begin and end dates for the item’s price)
  + multiply the correct price by the quantity, while still showing the price and quantity fields
  + item\_inventory, GROUP’d to have serial numbers in a comma-delimited list (oi\_id JOIN)
  + GROUP BY needs all unaggregated columns showing up in your SELECT
* taxes (t) table
  + no JOIN required yet.

**Task 2.2: Boxstore ERD Completion**

With regards to the adding of people\_employee (pe) and manufacturer (m) to the ERD, you always want to keep the diagram lined up properly, and have the shortest lines you can manage.

NO NEED to connect any other occurrence of usermod/useract from subsequent tables to the p\_id in people (p), as the diagram will be far too line filled.

Add all tables from the orders\_\_item script to your ERD and JOIN them up

**Task 2.3: Receipts Views**

Need to create views to put this all together. Take the existing JOINs for the Task 2.1 and put them together to make a receipt.

**First** VIEW: **receipt\_00\_header\_company\_info** (use Boxstore Inc. only information)

**Second** VIEW: **receipt\_01\_header\_order\_employee\_customer** (use the orders JOIN to employees and customers that you made

**Third** VIEW: **receipt\_10\_body\_line\_items**, use the orders\_\_item JOIN to display the columns you would see on a receipt as line items, still showing serials numbers in the comma delimited list

**Fourth** VIEW: **receipt\_20\_footer\_subtotal** group the previous view by orders, removing all unnecessary columns

**Fifth** VIEW: **receipt\_21\_footer\_subtotal\_taxes**, cross join to the taxes table using the order\_date, to get the current taxes and multiply by the subtotal

**Last** VIEW: **receipt\_22\_footer\_grandtotal**, using the previous view, add the subtotals after taxes together.

When writing views, not that 2 columns of the same name will create an error, so careful on which columns you remove…maybe keep the PK columns at least, and any meta column you will need throughout.

## 