**BUS 440 Project 1**

**Use Case:**

Tucker Electronics, LLC. needs to keep track of their office, electronic, and computer inventory to reduce costs, manage reorder points, and better track item locations within their facilities. They need to track which buildings contain what products and be able to contact the manager of each building to verify inventory and reorder products, when necessary. The New Technology Building (NTB) is a building managed by Langley T. Johnson, while the Michael Simms Building (MSB) is managed by Mariel J. Carter. Each manager has access to the main building phone number of the building he or she manages. Assistants keep track of office computers, servers, printers, scanners, printer toner, and wearables etc. using a simple spreadsheet. It looks something the data shown below, with a sample of a few of the items tracked. This includes all inventory stored in buildings, as well as purchase order information.

Add managers of buildings from use case

**Instructions:**

In this project, your goal is to replace the spreadsheet solution currently used by the administrative assistant with a relational database. Design a relational database solution. Be sure to note the attributes listed in the current spreadsheet as well as those discussed in the use case.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | Item Name | Building | Room Number(s) | Vendor ID | Vendor Name | Total Item Quantity on Hand | Item Quantity on Order | Purchase Order No. | Item Cost | Vendor AP Terms (Months) | Purchase Order Date | Arrival Date |
| 231134 | HP DeskJet 895Cse | New Technology Building | 325, 100 | 555 | Sam’s Supply Store | 20 (10 in each room) | 4 | 10001 | 119.55 | 30 | 08/05/21 | 02/25/22 |
| 342245 | HP Toner | New Technology Building | 325 | 555 | Sam’s Supply Store | 12 | 2 | 10001 | 35.00 | 30 | 08/05/21 | 02/25/22 |
| 254668 | DT Scanner | Michael Simms Building | 123 | 456 | John’s Warehouse Deals | 18 | 5 | 10003 | 199.99 | 30 | 08/05/21 | 02/25/22 |
| 254678 | DT Hard Drive | Michael Simms Building | 100 | 987 | Electronic Depot | 4 | 2 | 10004 | 400.88 | 30 | 08/05/21 | 02/25/22 |
| 254278 | DT 400 Cable | New Technology Building | 100 | 987 | Electronic Depot | 2 | 1 | 10005 | 19.66 | 30 | 08/05/21 | 02/25/22 |
| 112200 | Watch Wearable | New Technology Building | 333 | 444 | Hulkey Wearables | 100 | 195 | 11001 | 4.25 | 30 | 08/05/21 | 01/13/22 |
| 114300 | Watch Wearable | New Technology Building | 333 | 445 | Aboss Devices | 109 | 100 | 11002 | 4.25 | 30 | 08/05/21 | 01/13/22 |
| 231134 | HP DeskJet 895Cse | New Technology Building | 105 | 555 | Sam’s Supply Store | 5 | 2 | 11005 | 125.00 | 30 | 01/15/22 | 02/19/22 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

**Correct months to days in Vendor AP Terms**

**Part 1: Design the Database**

Step 1: Read the use case and carefully view the data. Write down the business rules from the data (e.g., An item may be stored in multiple rooms of a building). Write down the business rules from the use case narrative (e.g., Each building is managed by a manager).

* An item may be stored in multiple rooms of a building
* Each building is managed by a manager
* Multiple items can be purchased on the same day
* Multiple items can arrive on the same day
* Each building has a code, name, and phone number
* An item can be purchased on two different days
* An order can have multiple items
* A vendor can sell multiple items
* A vendor ID is associated with one and only one vendor
* An item has one and only one associated with one and only one price
* The same room number can appear in multiple buildings

Step 2: Define the entities, attributes, and relationships and model them. To do so, write the relational schema. Look at the sample records and any data found inside the use case description. Find the possible themes, and then setup the relational schema (or dependency diagram or functional dependency) to represent all of the attributes listed. If possible, identify a candidate primary key that uniquely defines each row.

INVENTORYPURCHASES (Item\_Number,Item\_Name,Building,Room\_Number(s),VendorID,Vendor\_Name,Total\_Quantity\_Item\_On\_Hand,Item\_Quantity\_On\_Order,Purchase\_Order\_Number,Item\_Cost,Vendor\_AP\_Terms(Months),Purchase\_Order\_Date,Arrival\_Date,Manager\_Name,Building\_Code,Building\_Phone\_Number)

Step 3: Normalize to third normal form (3NF). Show all work for each step (1NF, 2NF, and 3NF).

**1NF:**

Identify the primary key -> Since there are no attributes that aren’t not repeating in the rows we have to use a composite key

->A possible candidate is Item\_Number, Purchase\_Order\_Number

->Another candidate is Item\_Number,Room\_Number,Building\_Code

INVENTORYPURCHASES (**Item\_Number**,Item\_Name,Building,Room\_Number(s),VendorID,Vendor\_Name,Total\_Quantity\_Item\_On\_Hand,Item\_Quantity\_On\_Order,**Purchase\_Order\_Number**,Item\_Cost,Vendor\_AP\_Terms(Months),Purchase\_Order\_Date,Arrival\_Date,Manager\_Name,Building\_Code,Building\_Phone\_Number)

Remove repeating groups -> Room\_Number(s), Total\_Item\_Quantity\_On\_Hand

INVENTORYPURCHASES (**Item\_Number**,Item\_Name,Building,Room\_Number(s),VendorID,Vendor\_Name,Total\_ Item \_Quantity\_ On\_Hand,Item\_Quantity\_On\_Order,**Purchase\_Order\_Number**,Item\_Cost,Vendor\_AP\_Terms(Months),Purchase\_Order\_Date,Arrival\_Date,Manager\_Name,Building\_Code,Building\_Phone\_Number)

To:

**1NF:**

ROOM(Room\_Number**,Building\_Code**)

QUANTITYONHAND (Total\_ Item \_Quantity\_ On\_Hand **,Item\_Number,Building\_Code,Room\_Number**)

INVENTORYPURCHASES (**Item\_Number**,Item\_Name,Building,VendorID,Vendor\_Name,Item\_Quantity\_On\_Order,**Purchase\_Order\_Number**,Item\_Cost,Vendor\_AP\_Terms(Months),Purchase\_Order\_Date,Arrival\_Date,Manager\_Name,Building\_Code,Building\_Phone\_Number)

**2NF:**

**FROM:**

ROOM(**Room\_Number,Building\_Code**)

QUANTITYONHAND (Total\_ Item \_Quantity\_ On\_Hand **,Item\_Number,Building\_Code,Room\_Number**)

INVENTORYPURCHASES (**Item\_Number**,Item\_Name,Building,VendorID,Vendor\_Name,Item\_Quantity\_On\_Order,**Purchase\_Order\_Number**,Item\_Cost,Vendor\_AP\_Terms(Months),Purchase\_Order\_Date,Arrival\_Date,Manager\_Name,Building\_Code,Building\_Phone\_Number)

**To:**

ROOM(**Room\_Number,Building\_Code**)

QUANTITYONHAND (Total\_ Item \_Quantity\_ On\_Hand **,Item\_Number,Building\_Code,Room\_Number**)

ITEM(**Item\_Number**,Item\_Name , VendorID, Item\_Quantity\_On\_Order, Item\_Cost)

PURCHASEORDER(**Purchase\_Order\_Number**, Purchase\_Order\_Date, Arrival\_Date)

INVENTORYPURCHASES(***Purchase\_Order\_Number****,* ***Item\_Number***, , Vendor\_Name, Building\_Code, Vendor\_Name,Building,Building\_Manager,Building\_Phone\_Number, Vendor\_AP\_Terms(Months))

**3NF**:

**FROM**:

ROOM(**Room\_Number,Building\_Code**)

QUANTITYONHAND (Total\_ Item \_Quantity\_ On\_Hand **,Item\_Number,Building\_Code,Room\_Number**)

ITEM(**Item\_Number**,Item\_Name , VendorID, Total\_Item\_Quantity\_On\_Order, Item\_Cost)

PURCHASEORDER(**Purchase\_Order\_Number**, Purchase\_Order\_Date, Arrival\_Date)

INVENTORYPURCHASES(***Purchase\_Order\_Number****,* ***Item\_Number***, , Vendor\_Name, Building\_Code, Vendor\_Name,Building,Building\_Manager,Building\_Phone\_Number, Vendor\_AP\_Terms(Months))

**To**:

ROOM(**Room\_Number,Building\_Code**)

QUANTITYONHAND (Total\_ Item \_Quantity\_ On\_Hand **,*Item\_Number*,*Building\_Code*,Room\_Number**)

ITEM(**Item\_Number**,Item\_Name , Item\_Cost, *VendorID,* Total\_Item\_Quantity\_On\_Order)

PURCHASEORDER(**Purchase\_Order\_Number**, Purchase\_Order\_Date, Arrival\_Date)

INVENTORYPURCHASES(***Purchase\_Order\_Number****,* ***Item\_Number***, , *Building\_Code*)

VENDOR(**VendorID**, Vendor\_Name, Vendor\_AP\_Terms(Months))

BUILDING(**Building\_Code**, Building, Building\_Manager, Building\_Phone\_Number)

Step 4: Improve the Design Improving the design involves standardization of names, adding new attributes, adding new relationships and tables, etc. Update any relational schemas or dependency diagrams.

* Add managerIDS
* Correct the AP terms to days
* Rename total\_item\_quantity\_On\_Hand to Item\_Quantity\_On\_Hand
* Standardize Item\_Number and Building\_Code to ItemID and BuildingID
* Standardize Purchase\_Order\_Number to OrderID
* Specify Building to Building\_Name
* Standardize Entity name to one word descriptors
* Break up Building Manager into ManagerFName, ManagerMI, ManagerLName,

**FROM**:

ROOM(**Room\_Number,Building\_Code**)

QUANTITYONHAND (Total\_ Item \_Quantity\_ On\_Hand **,*Item\_Number*,*Building\_Code*,Room\_Number**)

ITEM(**Item\_Number**,Item\_Name , Item\_Cost, *VendorID,* Item\_Quantity\_On\_Order)

PURCHASEORDER(**Purchase\_Order\_Number**, , Purchase\_Order\_Date, Arrival\_Date)

INVENTORYPURCHASES(***Purchase\_Order\_Number****,* ***Item\_Number***, , *Building\_Code*)

VENDOR(**VendorID**, Vendor\_Name, Vendor\_AP\_Terms(Months))

BUILDING(**Building\_Code**, Building, Building\_Manager, Building\_Phone\_Number)

**TO**:

ROOM(**Room\_Number,*BuildingID***)

INVENTORY (Item \_Quantity\_ On\_Hand **,*ItemID*,*BuildingID*,Room\_Number**)

ITEM(**ItemID**,Item\_Name, Item\_Cost, *VendorID*, Item\_Quantity\_On\_Order)

ORDER(**OrderID,** Purchase\_Order\_Date, Arrival\_Date)

INVENTORYPURCHASE(***OrderID****,* ***ItemID***, *BuildingID*)

VENDOR(**VendorID**, Vendor\_Name, Vendor\_AP\_Terms(Days))

BUILDING(**BuildingID**, Building\_Name, *ManagerID*, Building\_Phone\_Number)

MANAGER (**ManagerID**, ManagerFName, ManagerMI, ManagerLName)

Step 5: Build the ERD in Visio or LucidChart (or hand-draw it).

(Consider getting your design approved before proceeding.)

A diagram of a computer program

Description automatically generated



**Part 2: Type data into spreadsheets**

In this part, populate the data into temporary spreadsheets, which we will later use to import into the tables.

1. Open Excel and create a file for each entity represented in your ERD. Note that you will have several spreadsheets of data. Name each file appropriately (e.g., ITEM).

2. Label the first row of each sheet with the attribute names from your ERD.

3. In rows 2 and following, type data into each spreadsheet representing the topic (theme) of that sheet, according to your ERD.

4. Save each Excel spreadsheet as a .csv file.

**You will need a separate spreadsheet for each of your tables**

**Part 3: Build the Database in MySQL from the Design and Import the Data**

In this part, import the records into the tables and build the PKs and FKs in My SQL.

1. From the Navigator Panel (left side), right-click the World schema and choose Create New Schema.
2. Name the schema TuckerElectronics and click Apply. Continue to click Apply and Finish.
3. From the Navigator panel, right-click and choose Refresh All to see the new database.
4. Expand the TuckerElectronics schema.
5. Import each table in the schema. To do so, complete these steps for each table:
   1. Right-click the TuckerElectronics database and choose Table Data Import Wizard.
   2. Navigate to the location where you stored the spreadsheets with the data.
   3. Open the sheets and import the data. Create a new table, item, by finding the item csv file and opening it. Click Next through the following steps to import the rows. Click Finish.
   4. Right-click on Tables and choose Refresh All, and then expand the Tables object to see the table. Hover over the table to see the three icons on the right. Click the third icon to run a SQL query listing the rows in the table.
   5. Import the data from each additional spreadsheet using similar steps. Right-click on Tables and choose Refresh All, and then expand the Tables object to see the table. Hover over the table to see the three icons on the right. Click the third icon to run a SQL query listing the rows in the table.
   6. Tell MySQL Workbench which schema to use. To do so, enter: **USE TuckerElectronics;**
   7. Create the primary key for ***each table*** (corresponding to the respective entity represented in your ERD). (See the examples below.) Each table should have only one (sole or composite) non-null primary key, indexed with no duplicates. If the PK does NOT have text-based data (e.g., it is all numbers), use this example. Here, we are changing (altering) the metadata associated with the already-existing vendor table. This table was built automatically during the process of importing the data from the spreadsheet.

**ALTER TABLE vendor ADD PRIMARY KEY(Vendor\_ID);**

If a PK contains text data, you must define the length of the primary key, as shown below in the text data of the building code (e.g., NTC) for the building table.

**ALTER TABLE building ADD PRIMARY KEY(Building\_Code(3));**

If the PK is composite, refer to the example below for the room table. Notice the ADD CONSTRAINT statement, where we give the constraint a name (pkc\_room) along with identifying the two fields that comprise the primary key.

**ALTER TABLE building\_room ADD CONSTRAINT pkc\_room PRIMARY KEY (BuildingCode(3), Room\_Number);**

If you have a table with a PK that has a data type of TEXT, you must modify (alter) the table again to change the data type from TEXT to VARCHAR and identify the appropriate length for that field. Below is an example. The VARCHAR (variable data type with a fixed length of three (instead of the variable length TEXT data type)) will assist as you build the FK in the next steps.

**ALTER TABLE building MODIFY Building\_Code VARCHAR(3);**

Create the foreign keys to establish the one-to-many (1:M) relationships between the tables in your ERD. See the examples below. Notice the use of a constraint to make the FK.

**ALTER TABLE item ADD CONSTRAINT FK\_Vendor\_code FOREIGN KEY (Vendor\_Code) REFERENCES vendor(Vendor\_Code);**

If you need to build a foreign keys to connect to a table with a composite PK, see the example below.

**ALTER TABLE inventory**

**ADD CONSTRAINT FK\_Inventory\_RoomNumber FOREIGN KEY (BuildingCode, RoomNumber) REFERENCES building\_room(BuildingCode, RoomNumber);**

**Part 4: Write SQL queries:**

In this part, write a SQL program to extract the data to answer each of these questions. Include a screen shot of your SQL code and results for each question in this Word (not pdf) document. Also include the .sql file with the script for all queries.

1. Which items are currently on order and from what vendor? Include the vendor name and location.

A screenshot of a computer

Description automatically generated

1. What is the list of inventory (items), along with the location (building and room) and the quantity on hand of each.

A screenshot of a computer

Description automatically generated

1. What is the aggregated total inventory of each item in each room of each building? Include each item number and name, along with its total.

A screenshot of a computer

Description automatically generated

1. What is the manager’s name and phone number of each building and the products stored in that building?

A screenshot of a computer

Description automatically generated

1. What is the aggregated total inventory of all items in all buildings by vendor name?
2. What items have purchase orders from what vendors with what amounts?

A screenshot of a computer

Description automatically generated

1. Write at least one more query of your choosing. It must include an average aggregate calculation of some type.

A screenshot of a computer program

Description automatically generated

1. Write at least one more query of your choosing. It must include a subquery.

A screenshot of a computer

Description automatically generated

**Part 5: Submit your work in a zipped file to Moodle by the due date. The zipped file should include:**

1. This document with all design steps and build steps (including SQL screenshots) completed.
2. ERD file via handwritten image, Visio, or LucidChart;
3. .sql file with all queries in a script via MySQL Workbench.