Brandon Russell <rbrandon87@gmail.com>

Week 2 ERD

Brandon Russell <rbrandon87@gmail.com> Sun, Feb 21, 2016 at 6:36 PM
To: Ronald Savarese <rlsavarese@gmail.com>
Cc: Jacob Peterson <a3sir86@gmail.com>, Felix Sedegbe <FSEDEGBE@student.umuc.edu>, Kesha Robinson <ladydevine00@yahoo.com>

Group 3 Project SQL Feb 21 2016 Russell Origina...

Group 3 Project SQL Feb 21 2016_Ron v2.sql

Group 3 Project SQL Feb 21 2016 Russell v2.sql

All,

If you could please review the different SQL files we currently have. So I have the original SQL file I made. Then, I made another version, Russell v2, with recommendations from Ron-except I left the ShipmentDetails_ID constraint in the OrderDetails. I made another version, Ron v2, with this constraint removed....as he suggested earlier. If I don't get input from everyone tonight I will go ahead and upload both versions to the discussion and let the TA know we are still hashing out some of the details. She did make a point to say before we don't have to have our deliverables 100% ready at the end of each week...so it shouldn't be an issue.

Brandon

On Sun, Feb 21, 2016 at 3:57 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

> - While I am OK with the current iteration of the ERD, it was never intended that the shipment and order tables be connected; now that we are linking those two tables we're saying that customer orders could possibly be shipments. It was only ever intended to be shipments to the store, so that we can track inventory and compute ToS. Not that it won't still work that way, but now we'll have more NULL values that exist in our database because not every order will necessarily be a shipment and vice a versa, if all that makes sense.
>
> **The ShipmentDetails and Order tables must be linked. I am not saying anything about making shipments for customers. This link is mandated so that we know exactly which item from which "shipment to the store" is going to which customer. If we don't provide this link then we will not know which item from which shipment is being sold and we will have literally zero idea on "Time On Shelf". Once I added test data and created my two views this became very obvious. There won't be any nulls. Basically a OrderDetails instance must belong to one ShipmentDetails instance and an item from the ShipmentDetails can belong to one or more OrderDetails. All we are saying here is the item from this shipment is the item going to the customer in this order.**
>
> - Looking at your table creation statements, where you have your constraints at the end: don't you need to list the field you're referencing for the foreign key constraint? You may not have to in Oracle, to be honest though, it would be good practice to do so for the sake of having a clean design.
>
> **I did?**
>
> - I still think you should leave out the views, both of them actually. Leave those for the query deliverables, we can turn them into views afterwards. I would definitely re-write the ToS one you have as it is, so let's leave those out. *If you don't want to for some reason, please change the comment describing it at the very least. It won't show ToS for items still in stock, it's going to show ToS for every item ever received, irregardless of whether it was sold or not.
>
> **No problem, as stated before I only added them to verify the design would work...and it does. We can remove those for now and reuse them later. I think my view is still quite useful. We should definitely have a query that shows items "In Stock" and their ToS, this way we know to sell the oldest item to the customer first. That's not to say we should not also have a query like you are saying which shows ToS for all items...**
>
> - The UpdateItemQty_Trigger you've created needs to be revised to the following:
>
> ```
> /* Create trigger */
> /*This trigger will update an items quantity after an order is inserted into the OrderDetails table*/
> CREATE OR REPLACE TRIGGER UpdateItemQty_Trigger AFTER INSERT ON OrderDetails
> FOR EACH ROW
>    BEGIN
>         UPDATE Item
>         SET Total_Qty = Total_Qty - :New.Order_Details_Qty
>         WHERE Item_ID = :New.Item_ID;
>         dbms_output.put_line ('Customer Order Quantity deducted!!!');
>     END;
> ```
>
> Reason being, the data in shipment details shouldn't be altered. That is essentially our record of how many of what items were shipped to the store. Whenever inventory reconciliation time comes around, a store would need to know when, how many, etc... The only quantity data that should be altered is in the Item table itself, once orders are placed, and received. Pretty much like you have it, except for the extra bit on the above trigger.
>
> **So again, going back to the link between the ShipmentDetails and OrderDetails. First, we need this link I'm positive on that. Second, we need to update the ShipmentDetails quantity so we don't try and resale an item already sold. If we need to know the original quantity from a shipment we can easily run a query against the OrderDetails table to count the number of times that ShipmentDetails ID shows up. Otherwise we could make a separate column in ShipmentDetails for the "Original Quantity, but this is unnecessary since it can be derived on the fly.**
>
> - Don't worry about data dictionary query, that is part of my deliverable next week. I have a different one that I'm going to use than what you have, so I took that part out.
>
> **Ok, let's remove this.**

> - I also can't say that I'm crazy about the DESCRIBE statement after each object creation entry, I get that you're intent is likely to verify object creation and the details afterwards, however, the resulting script output is going to be so long (and happen faster than you'll be able to verify), that it won't be as useful as just doing a data dictionary query after all is said and done. This would be useful if you were using scripts to say update a database perhaps. Having said that, I didn't take them out because it's not going to make or break anything, just not necessary or as useful as one might think.
>
> **Just there to verify everything is working, we can remove these.**
>
> Alright, that's it from me. I was going to make a comment about the way you've handled sequences, however, I'm a fan of not using too many triggers, so I like it. I'm used to SQL Server having an IDENTITY for number data types that does the auto incrementing, much easier!
>
> **In a nut shell, I agree let's remove the views, data dictionary query, and describe statements. However, I am positive we need to have the ShipmentDetails_ID foreign key in the OrderDetails table. Likewise, we can't modify the trigger as you mentioned otherwise…again the whole thing just wouldn't work.**
>
> **If we don't have said the OrderDetails/ShipmentDetails constraint and update in the trigger…how are we going to know which item was sold to which customer? This is important so that we know how long a single specific item has been sitting on a shelf, right?**
>
> **If I could get everybody's input on this it would be great. I don't want to have a messed up design now, otherwise it is going to heavily affect the remainder of this project.**
>
> **Thanks,**
>
> **Brandon**
>
> On Sun, Feb 21, 2016 at 3:15 PM, Ronald Savarese <rlsavarese@gmail.com> wrote:
>
>> Brandon, overall it looks great, nice work.
>>
>> So here are just my notes/observations, mixed in with my .02 cents :) -
>>
>> - While I am OK with the current iteration of the ERD, it was never intended that the shipment and order tables be connected; now that we are linking those two tables we're saying that customer orders could possibly be shipments. It was only ever intended to be shipments to the store, so that we can track inventory and compute ToS. Not that it won't still work that way, but now we'll have more NULL values that exist in our database because not every order will necessarily be a shipment and vice a versa, if all that makes sense.
>>
>> - Looking at your table creation statements, where you have your constraints at the end: don't you need to list the field you're referencing for the foreign key constraint? You may not have to in Oracle, to be honest though, it would be good practice to do so for the sake of having a clean design.
>>
>> - I still think you should leave out the views, both of them actually. Leave those for the query deliverables, we can turn them into views afterwards. I would definitely re-write the ToS one you have as it is, so let's leave those out. *If you don't want to for some reason, please change the comment describing it at the very least. It won't show ToS for items still in stock, it's going to show ToS for every item ever received, irregardless of whether it was sold or not.
>>
>> - The UpdateItemQty_Trigger you've created needs to be revised to the following:
>>
>> ```
>> /* Create trigger */
>> /*This trigger will update an items quantity after an order is inserted into the OrderDetails table*/
>> CREATE OR REPLACE TRIGGER UpdateItemQty_Trigger AFTER INSERT ON OrderDetails
>> FOR EACH ROW
>>    BEGIN
>>         UPDATE Item
>>         SET Total_Qty = Total_Qty - :New.Order_Details_Qty
>>         WHERE Item_ID = :New.Item_ID;
>>         dbms_output.put_line ('Customer Order Quantity deducted!!!');
>>     END;
>> ```
>>
>> Reason being, the data in shipment details shouldn't be altered. That is essentially our record of how many of what items were shipped to the store. Whenever inventory reconciliation time comes around, a store would need to know when, how many, etc... The only quantity data that should be altered is in the Item table itself, once orders are place, and received. Pretty much like you have it, except for the extra bit on the above trigger.
>>
>> - Don't worry about data dictionary query, that is part of my deliverable next week. I have a different one that I'm going to use than what you have, so I took that part out.
>>
>> - I also can't say that I'm crazy about the DESCRIBE statement after each object creation entry, I get that you're intent is likely to verify object creation and the details afterwards, however, the resulting script output is going to be so long (and happen faster than you'll be able to verify), that it won't be as useful as just doing a data dictionary query after all is said and done. This would be useful if you were using scripts to say update a database perhaps. Having said that, I didn't take them out because it's not going to make or break anything, just not necessary or as useful as one might think.
>>
>> Alright, that's it from me. I was going to make a comment about the way you've handled sequences, however, I'm a fan of not using too many triggers, so I like it. I'm used to SQL Server having an IDENTITY for number data types that does the auto incrementing, much easier!
>>
>> I've added the changes I mentioned above, except for where it was just opinion (like taking out DESCRIBE). New SQL Statement is in deliverables folder with my name at end.
>>
>> Let me know what you all think!
>>
>> Ron

On Sun, Feb 21, 2016 at 10:29 AM, Brandon Russell <rbrandon87@gmail.com> wrote:

I just uploaded a new one, labeled as Feb 21, so please look at that one. I just made some minor code formatting modifications and a minor change to one of the views...nothing big.

On Sun, Feb 21, 2016 at 11:20 AM, Ronald Savarese <rlsavarese@gmail.com> wrote:

Sorry I've been MIA since earlier this week, thought there was a chance I couldn't attend meeting so didn't volunteer. I'm going to take another look at week 3 deliverable now.

On Sun, Feb 21, 2016 at 8:16 AM, Brandon Russell <rbrandon87@gmail.com> wrote:

Thanks Jacob. Is everybody else okay with the Week 3 deliverable? Let me know sometime today so I can submit in the discussion/group locker today.

Brandon

On Sat, Feb 20, 2016 at 6:32 PM, Jacob Peterson <a3sir86@gmail.com> wrote:

Alright well I looked over this weeks deliverable and I am good to go with everything to be submitted. After reading the TA's post about the shipment table I don't think we really need to change anything. In the real world we would definitely need to add more tables for vendor information and what not, but I think the way we have implemented it works fine.

On Sat, Feb 20, 2016 at 6:17 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

Unfortunately I am out of town at an event, as I mentioned two days ago. I asked if someone else could host the meeting but I never got a response from anyone...

On Feb 20, 2016 6:02 PM, "Jacob Peterson" <a3sir86@gmail.com> wrote:

Are we doing a google hangout meeting tonight?

On Thu, Feb 18, 2016 at 7:39 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

Hey guys,

I just remembered I will be out of town this weekend and I most likely won't be able to make the meeting. Could someone please host the meeting this Saturday and take meeting notes. If you guys could just let me know via the notes and/or email if the week 3 deliverable looks good and I can upload to the discussion and group locker on Sunday evening.

Also, I just noticed the TA made a post in the discussion about our ERD. She didn't seem impressed with the Shipment/ShipmentDetails tables, but I just replied to her asking for more clarification because since creating the DDL/DMLs I have run many successful queries...so I don't know why she has an issue. She also apparently has an issue with using "INTEGER" lol, so I went ahead and made everything a "NUMBER" in the SQL script and the ERD.

Brandon

On Thu, Feb 18, 2016 at 2:31 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

True, but I needed to do some queries just to make sure the design would work. Hopefully this will atleast make those weeks work a little lighter...

On Feb 18, 2016 2:17 PM, "Ronald Savarese" <rlsavarese@gmail.com> wrote:

You know Brandon, after looking at the deliverables, you're getting into other people's tasks by doing any queries! Looks like Week 5 or possibly even week 8 (create function). We can help Jacob with ToS query when he starts working on those :).

Ron

On Wed, Feb 17, 2016 at 7:58 PM, Ronald Savarese <rlsavarese@gmail.com> wrote:

Will take a look when I get home tonight

On Feb 17, 2016 7:38 PM, "Brandon Russell" <rbrandon87@gmail.com> wrote:

The only thing I can think of right now, is we don't have anything to stop an order from pulling from a shipment that already has 0 quantity. This would probably need to be a trigger as well...if I have time tomorrow I'll mess with it.

On Wed, Feb 17, 2016 at 8:35 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

I also updated the ERD, see both updated files in the week 3 folder.

On Wed, Feb 17, 2016 at 8:35 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

Hey guys,

Okay so after banging my head against the wall a few times I think I finally figured out a query for Time on Shelf...I think. So, basically I had to modify the schema slightly. I added the "Shipment_Details_ID" to the OrderDetails table. This way we know which item from which shipment went to a customer. This way we can know for sure exactly which items are still in stock. I also included in one of my triggers to reduce the "Shipment_Qty" when a particular item is sold...in addition to the "Total_Qty" being reduced. This made it way easier to determine which items were still in stock. Originally I was concerned about modifying the "ShipmentDetails", but then I realized we can just query "OrderDetails" and do a count on "Shipment_Details_Id" if we ever needed to know the original quantity from a particular shipment.

...I think this works, but let me know what you guys think.

Brandon

On Wed, Feb 17, 2016 at 7:10 PM, Ronald Savarese <rlsavarese@gmail.com> wrote:

Neither is "NUMERIC", wonderful logic there.

On Feb 17, 2016 6:05 PM, "Brandon Russell" <rbrandon87@gmail.com> wrote:

The reasoning on my week 2 exercise feedback was something along the lines of it not being an "official" oracle datatype, but rather a generic sql datatype which is not supported universally in all Oracle versions? I didn't know that, but whatever...

On Wed, Feb 17, 2016 at 7:03 PM, Ronald Savarese <rlsavarese@gmail.com> wrote:

Yeah, I'm not sure why that is the case. Best practices in database development are that you use a *number* as the primary key, but it doesn't have to be the NUMERIC data type. The irony is that Oracle uses NUMBER, which you can specify has a precision and scale (decimals) or not while the ER assistant just indicates NUMERIC and indicates it has a precision and scale. I'm sorry, but I would never choose to create a primary key that has precision and scale over a regular integer. Incidentally, an Integer is also a number.

I think I'm going to ask this professor to explain the preference of one over another :). It bothers me when things like that affect our grade, but have no real bearing on the real world.

On Wed, Feb 17, 2016 at 5:36 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

> Ron,
>
> Didn't you say in your last class the professor didn't like the use of INTEGER? Well it looks like this professor is the same, I was told for exercise 2 that I should've used NUMBER....I guess I will update my week 3 deliverable?
>
> On Tue, Feb 16, 2016 at 11:46 AM, Ronald Savarese <rlsavarese@gmail.com> wrote:
>
>> Great Brandon, will take a look tonight or tomorrow!
>>
>> Ron
>>
>> On Mon, Feb 15, 2016 at 12:20 PM, Brandon Russell <rbrandon87@gmail.com> wrote:
>>
>>> Okay guys, I got the code created. Please review when possible and let me know what you think.
>>>
>>> I made code to
>>> 1. drop all tables/sequences
>>> 2. create tables with primary/foreign keys
>>> 3. created indexes on foreign keys
>>> 4. created a view to display the TOS in days for all items (bumped against ShipmentDetails and ordered to see which item has longest time on shelf)
>>> 5. created two triggers to either add or substract from item quantity depending on if we are adding to the orderdetails or shipmentdetails tables
>>> 6. created sequences
>>> 7. finally inserted some test data.
>>>
>>> I think that's everything, let me know if anything is missing or wrong.
>>>
>>> Brandon
>>>
>>> **Group 3 Project SQL.sql**
>>>
>>> On Mon, Feb 15, 2016 at 10:40 AM, Brandon Russell <rbrandon87@gmail.com> wrote:
>>>
>>>> **Group 3 ERD Russell Version 2152016.erd**
>>>>
>>>> **Russell_ERD DataModeler.PNG**
>>>>
>>>> Hey guys I started working on our project DDL/DMLs. While doing so I noticed a few problems with the ERD. Most of it was small stuff like I split customer name into First & Last and also I abbreviated Quantity in a few of the tables to Qty. I think the only big thing I noticed was the "Order" table. Apparently Order is a reserved word so I had to change the name to "CustomerOrder" and then the Primary key to "Customer_Order_ID". I'll have the majority of the code done today, I'll upload into the week 3 folder when done so you guys can see.
>>>>
>>>> Brandon
>>>>
>>>> On Sun, Feb 14, 2016 at 8:39 PM, Ronald Savarese <rlsavarese@gmail.com> wrote:
>>>>
>>>>> So that is where the ItemInventories table would have made more sense from a *nomenclature* perspective, because it stores the item, and the date received as an item in the stores inventory. With the ShipmentDetails and Shipment we are just assuming those items will go into the store's inventory when they are received, which is OK for our purposes. This is also why leaving the quantity in ItemInventories would have made it easier from a programming perspective. When we discussed it, I couldn't see it so forgot some of that info... How we would handle it is something like this:
>>>>>
>>>>> This is the business logic - A customer purchases an item, and it will need to be removed from inventory using the LIFO / FIFO method (we still need to decide which we're using).
>>>>>
>>>>> How we would handle it from a programming perspective, is we would either write a trigger or stored procedure which would be an AFTER UPDATE and then it would search for the particular item purchased and remove however many necessary. We would probably want to write a stored procedure so that we can first check to make sure there is enough in stock from that particular date, and then go from there.
>>>>>
>>>>> Doing the programming for LIFO / FIFO valuation is rather complex, not going to lie. This is an advanced database class though!
>>>>>
>>>>> On Sun, Feb 14, 2016 at 11:21 AM, Brandon Russell <rbrandon87@gmail.com> wrote:
>>>>>
>>>>>> One more thing I was just thinking about... I'll probably figure it out once I start creating the database, once we have determined which item will be sold to a customer based on its TOS how do we update the shipment and shipment details tables so this wont be sold again? Would we just delete that record...then also update the quantity in the items table?

Deleting the record from shipment doesn't seem appealing to me, just be cause what if our store wanted to preserve that information for some kind of analytics later. Should we just include some flag attribute in shipment details to show an item is sold?

On Feb 14, 2016 12:02 PM, "Brandon Russell" <rbrandon87@gmail.com> wrote:

> Ron,
>
> Thanks for explaining that. I guess I was just confusing shipment as being a part of a customer shipment. I'll start working on the tables and inserting data tonight
>
>> **From:** Ronald Savarese <rlsavarese@gmail.com>
>> **To:** Brandon Russell <rbrandon87@gmail.com>
>> **Cc:** Jacob Peterson <a3sir86@gmail.com>; Kesha <ladydevine00@yahoo.com>; "FSEDEGBE@student.umuc.edu" <FSEDEGBE@student.umuc.edu>
>> **Sent:** Sunday, February 14, 2016 2:20 AM
>> **Subject:** Re: Week 2 ERD
>>
>> Hey Brandon,
>>
>> So without even intending on it, what we have here is a rather complex database scenario because of the SOW requirements! We basically have to have some inventory / accounting mechanism set up in order to account for the "time on shelf" requirement. In accounting you have what is known as LIFO / FIFO (Last In First Out / First In First Out), which is two different variations on valuation. While we're not trying to necessarily do a valuation on our stores inventory, we are trying to figure out how long a particular item was on the shelf, which means we are going to have to know which item was sold to the customer first. Was it the last item received, or the first? I hope I'm making sense here.
>>
>> Thinking about how to explain this through an email is difficult... I'm going to try and do my best here, if you need to have a call tomorrow or early in the week let's do that. I'll try going through your questions line by line.
>>
>> *"Should the Shipment_ID be placed in the OrderDetails table? Or maybe the Order_ID attribute placed in the Shipment table?"*
>> No, because those two sets of tables deal with different groups of people and types of transactions. The Order and OrderDetails table deal with products sold from the store and to the customer, while the shipment and ShipmentDetails tables deal with shipments to the store, from the vendors. We can adjust inventory through triggers or stored procedures.
>>
>> *"I guess what I'm getting at, I don't see how we are going to say "okay customer placed an order here is the order details...now somehow here is the shipment details even though there is no actual link between the two?"*
>> We aren't providing shipment details at the time of customer order, shipments are for orders placed to vendors... if this were a real robust RDBMS, you'd see all kinds of tables built out for the customers, and then for the vendors. We are limited to 7 tables max, so we need to keep things simpler. So when it comes to customers purchasing things, just think they are coming into the store and buying them. We don't need to add a foreign key reference for the shipment tables because everything is already related through the Item_ID attribute. We can query and use joins from here.
>>
>> *"I know that they share a relationship with the item table, but I don't think this is sufficient to show the relationship between an order and a shipment, right? I don't see how an item_ID is going to uniquely identify a relationship between Order and Shipment?"*
>> The Item_ID is all we need to tie everything together, because the Item_ID is present in both ShipmentDetails, and OrderDetails. The only thing we need is to do some slightly complicated programming in order to figure out time on shelf. I'll explain after I get through these.
>>
>> *"Thinking back to our conversation we said that we are creating the Shipment and Shipment Details table because the customer could only order a single type of item, however why are we saying that? Since Order has a one-to-many relationship with orderdetails, couldn't the order then have as many of any type of item the customer wanted?"*
>> We said that we are creating the Shipment and ShipmentDetails tables to track the items that the store orders, and come into the store as inventory. At the end of the day, we said we wanted to do a store inventory database. The Order and OrderDetails tables are for customer purchases. So it sounds like you are getting the two mixed up, they are for different purposes.
>>
>> Now, is it true that we could simply add a "VendorID" foreign key reference to the Order

table, and a corresponding "Vendor" table, and then get rid of both Shipment and ShipmentDetails? Yes, we could do that.... HOWEVER, that would be considered bad practice in the real world because remember now for every single transaction, either the Customer_ID or the Vendor_ID field would contain a NULL value. Going another logical step forward, we'd also have to add the "Date_Received" to the Order table as well, which would mean two guaranteed NULL values per record entry. NULL values are never good, you want to avoid them when possible. If we are creating an environment where we are saying there MUST be a NULL value for every record, that isn't good. I should have started this paragraph by stating - this isn't normalized to the 3NF either, because the Vendor_ID OR the Customer_ID aren't necessarily dependent on the Primary Key field, whereas if you have a table that exists for each, they are. Man, I hope I'm making sense here... it's late :).

So to explain how we'll handle the ToS (Time on Shelf) requirement once we build everything out, we would have to create a stored procedure or trigger that fired after orders were placed by the customer. We would have to make a determination first - Are we going to sell inventory that's been sitting on the shelf longer (LIFO method) or pull from inventory that has been there the least amount of time (FIFO method). So the procedure would then evaluate the date an item is received against the order date, and pull those entries first (whichever method we choose). We can run queries to find out how long items have sat on the shelf by doing something like this:

```
SELECT i.Item_ID, SYSDATE - s.Date_Received AS ToS, s.Shipment_ID
FROM Item i
JOIN ShipmentDetails sd ON sd.Item_ID = i.Item_ID
JOIN Shipment s ON s.Shipment_ID = sd.Shipment_ID
FULL OUTER JOIN OrderDetails od ON od.Item_ID = i.Item_ID
FULL OUTER JOIN Order o ON o.Order_ID = od.Order_ID
WHERE o.Order_Date IS NULL
ORDER BY  ToS DESC;
```

With this query we should get an itemized list of each item from different shipments - so we can look at the same item from different shipments *SHOULD* they have different ToS values. It's late, and we could probably clean this up and use aggregates, but I'm hoping the general idea is here. Also, just know that we'd have to use some Oracle function to get a number value out of the date difference, which we'd wrap the SYSDATE - s.Date_Received function in.

Remember, I'm not an Oracle guy... so if my syntax is off, I apologize, I tried to double check against Oracle docs, and it looks OK. Please also keep in mind without testing this in SQL Developer, I can't say this is the exact query we'd use, but it should be in the ballpark :).

FYI - I revised the ERD once more, the shipment and shipment details table were reversed, we need the relationship to be one to many with Item and ShipmentDetails, then Many to one with ShipmentDetails to Shipment. It is Group3_ERD_021416_1220.

Thanks team,
Ron

On Sat, Feb 13, 2016 at 6:54 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

> Should the Shipment_ID be placed in the OrderDetails table? Or maybe the Order_ID attribute placed in the Shipment table?
>
> I guess what I'm getting at, I don't see how we are going to say "okay customer placed an order here is the order details...now somehow here is the shipment details even though there is no actual link between the two?"
>
> I know that they share a relationship with the item table, but I don't think this is sufficient to show the relationship between an order and a shipment, right? I don't see how an item_ID is going to uniquely identify a relationship between Order and Shipment?
>
> Thinking back to our conversation we said that we are creating the Shipment and Shipment Details table because the customer could only order a single type of item, however why are we saying that? Since Order has a one-to-many relationship with orderdetails, couldn't the order then have as many of any type of item the customer

wanted?

I guess I'm still confused on the shipment/shipment_details tables and I really want to make sure I understand this before I start building the tables.

Thanks,
Brandon

On Sat, Feb 13, 2016 at 6:39 PM, Jacob Peterson <a3sir86@gmail.com> wrote:

> lastest ERD posted to Google Drive as Group3_ERD_021316_1832
>
> On Sat, Feb 13, 2016 at 5:54 PM, Brandon Russell <rbrandon87@gmail.com> wrote:
>
>> Here's the link for the Hangout:
>>
>> https://hangouts.google.com/call/uvwjaiiypp4hkpgl633qsutifma
>>
>> Brandon
>>
>> On Sat, Feb 13, 2016 at 5:03 PM, Jacob Peterson <a3sir86@gmail.com> wrote:
>>
>>> I definitely see your point too. I have a few ideas I guess we can talk more about it during the meeting.
>>>
>>> On Sat, Feb 13, 2016 at 4:52 PM, Brandon Russell <rbrandon87@gmail.com> wrote:
>>>
>>>> Jacob,
>>>>
>>>> That was my original thoughts as well, however if that's the case then why have a separate item and inventory table? Why not just put a quantity attribute in the items table? I figured we needed a separate inventory table that way we could have an individual record for every item. That way if they had different dates received/different locations in the store, this information would be portrayed there...maybe I'm not thinking about it right though?
>>>>
>>>> Brandon
>>>>
>>>> On Sat, Feb 13, 2016 at 4:27 PM, Jacob Peterson <a3sir86@gmail.com> wrote:
>>>>
>>>>> In my opinion you wouldn't have a 100 entries of the 2GB DDR3 item you would just have 100 stored in the quantity attribute and have a subtract 1 calculation performed on that number every time you sold one.
>>>>>
>>>>> On Sat, Feb 13, 2016 at 3:21 PM, Brandon Russell <rbrandon87@gmail.com> wrote:
>>>>>
>>>>>> Ron,
>>>>>>
>>>>>> I was looking at the ERD some more today. We can discuss this at the meeting here in a little while.
>>>>>>
>>>>>> But, I'm just thinking in regards to the "ItemInventories" table should we remove the "Quantity_in_Stock" attribute?
>>>>>>
>>>>>> Let's say we have an item in the item table for a "2GB DDR3 stick of ram". Now we go and make 100 records in the inventory table because we have that many on hand. Why would we have a quantity attribute listed 100 times saying the same thing? I think we could just pull that via a query, right?
>>>>>>
>>>>>> Another attribute we might add to the inventory table is an item location field...
>>>>>>
>>>>>> Brandon
>>>>>>
>>>>>> On Fri, Feb 12, 2016 at 4:50 PM, Ronald Savarese <rlsavarese@gmail.com> wrote:
>>>>>>
>>>>>>> Notes on ERD -
>>>>>>>
>>>>>>> **Consistency in Naming Conventions** - one thing that is of paramount importance when working on a development team is consistency in your naming convention. So with ours for example, we had some entities and attributes using the line spacing to denote separate words, and then others simply used camel case

(CamelCase). I am agnostic to which one we use, but It seemed like less work switching all the attributes over to the line spacing, while Entities will use camel case. It isn't unusual to see the two mixed, but they have to be used across the board for it's particular application. I hope that makes sense.

Ability to deliver on requirements for SOW - a few things were missing in order to fully deliver on what our SOW specified, one was a way to track inventory. You might think, well we had an item table, isn't that enough? In OLTP databases it isn't good design to have one item for every entry for an item. I think you'll get why, but here is an example. Let's say you are a parts manufacturer, and you need to keep an inventory of your parts. If you have 1k of one part, that is going to take up 1k records in your database alone, queries searching for information on that part would have to go through each record. It's better to have a separate table for keeping track of inventory, that way you can use a Quantity attribute. You'll also keep better track of when parts come in and out. Because of this I added the ItemInventories table. I set up another relationship between it and the item table to say an item has an entry in the ItemInventories table. I also added a few attributes to some tables to meet other SOW requirements:

- "For each product this database will include information about the hardware and software specification of the device." I added the Spec_Document attribute to Item entity, and gave it a BLOB data type so we could store actual spec documents and reference them from the database.
- "Store management will be able to see a breakdown of how well certain products are selling based on time on shelf." I simply added a Date_Received attribute to the ItemInventories entity to satisfy this requirement. This way we can write a query with a function to derive how long something has sat on the shelf based on when it was received, and when it was ordered.

At this point, all SOW requirements are capable of being met by writing queries, as the data will exist, and is connected properly.
Data Types - I went through and changed all the primary key data types to INTEGER, as NUMERIC to me denotes it has a precision and a scale (the ER assistant uses length & width). I believe that Oracle may use NUMERIC as a parent data type for INTEGERs, but we are using a platform agnostic Data Modeler. I will preface by stating this is one of those things that I got into a little bit of a disagreement with over my prof and TA in my last class. I am more used to MSSQL where NUMERIC denotes it is money, and so it would be a decimal. I would never use a decimal as a primary key when an INT will always work just fine.
You guys can change them back if you'd like, or we can discuss this with the professor and see if they have an opinion. In my last class they wanted us to use NUMERIC with no real explanation of why that over INT. Just because it was a number... hah.

I have uploaded an updated ERD as well, take a look and let me know what you all think.

Thanks!
Ron

On Fri, Feb 12, 2016 at 7:25 AM, Kesha <ladydevine00@yahoo.com> wrote:

> Thanks all,
>
> I will be traveling this weekend 12 hours drive so I may not be available for the call tomorrow. Thanks so much for reviewing the ERD and I look forward to reviewing the drafts. I will have time to review today.
>
> Kesha
>
> ---
>
> **From:** Ronald Savarese <rlsavarese@gmail.com>

To: Brandon Russell <rbrandon87@gmail.com>
Cc: Kesha <ladydevine00@yahoo.com>; "a3sir86@gmail.com" <a3sir86@gmail.com>; "FSEDEGBE@student.umuc.edu" <FSEDEGBE@student.umuc.edu>
Sent: Friday, February 12, 2016 1:41 AM
Subject: Re: Week 2 ERD

I reviewed both ERD's submitted so far, and both are missing some key requirements that the SOW describes. I am going to put together some notes and redo the ERD tomorrow, then submit to the group by COB tomorrow. Good starting point, we just need to make sure the ERD is inline with what is spelled out in the SOW, and we also need to be mindful of relationships. Brandon is correct about the many-to-many relationship being a no-no in database design, one-to-one is fine, you just don't see it that often because rarely is 'one and only one thing' ever associated with 'one and only one thing' in the real world.

As an example to illustrate with what we have with Kesha's model, you could essentially combine the Product and Item entities, there is no real reason to separate them. A "Product" likely is an "Item" for our purposes.

Ron

On Thu, Feb 11, 2016 at 8:00 PM, Brandon Russell <rbrandon87@gmail.com> wrote:

> Kesha and group,
>
> When you have a chance take a look at my version of the ERD based on Kesha's work, it's in the week 2 folder.
>
> Kesha, I noticed you had a many-to-many relationship which we cannot do and also a one-to-one relationship which I have been told is poor practice. After rectifying those two things, I realized there wasn't a table for what Jacob mentioned in the SOW about the Customer's setup being linked to the item listing. I added this and also removed a few things to keep us at 5 tables amongst other things.
>
> If everyone could take a look and let me know what you think. I'll take a closer look tomorrow, only had a couple hours today to look at it.
>
> Brandon
>
> On Thu, Feb 11, 2016 at 6:28 PM, Kesha <ladydevine00@yahoo.com> wrote:
>
>> Week 2 ERD is ready to review
>> Week 2
>>
>> image
>>
>> Week 2
>>
>> Group3ComputerStoreDoc (1).docx 3:27 pm
>>
>> View on **drive.google.com** Preview by Yahoo

Good Morning, Please find the latest draft using the last ERD posted.
Kesha