Course Name:Business Intelligence and Analytics
Professor Name:Prof. Saji.K.Mathew
Department Name:Department of Management Studies
Institute Name: Indian Institute of Technology Madras
Week:03
Lecture:11

SHOPSENSE CASE IN MySQL Workbench

So, let us go to the ShopSense retail case. ShopSense is a fashion retail chain and I will read out the case for you so that it will be easier for you to understand. Retail operations at ShopSense. ShopSense is a leading fashion retail chain headquartered in Bengaluru, India. The company operates about 150 retail stores in major cities of India. The company sells major global brands in footwear and clothing and also runs a successful loyalty program.

The retail industry is a lead growth segment among India's booming sectors. Footwear and clothing scores top on penetration in organized retail and brand consciousness is growing. So, it is a very successful business as we see from the case. With growing competition, ShopSense recognizes the value of data for business decisions.

Customer transaction data is a gold mine which when stored and analyzed leads to valuable insights for business. A well-developed database not only helps in generating financial reports, further database sub-segmentation brings marketing managers closer to the consumer segments who have similar behavior and help understand the way products fit into their lifestyles and expectations. This further supports personalized interactions through loyalty programs. ShopSense customer base consists of both walk-in customers who do not have a loyalty card and also customers who have registered for the loyalty program. Registered customers will have a non-zero customer ID and accrue loyalty points on every transaction they make.

Loyal customers are entitled for discounts on certain categories. For this, they have to produce their loyalty card or provide their cell number and ID card during checkout. Customer's checkout process consists of entry of selected items and preparation of a final bill after accounting for discounts and tax. Upon generation of bill, a customer makes payment either in cash or through card. On successful payment, a bill is printed and is issued to the customer along with the purchased items.

ShopSense being in customer retail does not sell items on credit. Customers could return items within two weeks of purchase along with the bill of items. ShopSense transaction processing system would enter the items in return category and credit the

amount back to the customer account either in cash or to the card depending on how the original payment was made. This is a brief history about ShopSense. We will see what all are the metadata.

I hope you will be knowing what is a metadata. Metadata is nothing but the data about data or the entire table can be summarized on what are the column names or field names. So, these are basically the table names and the field names. So, there are four tables as you see from this metadata which are first table is customer master table, second table is item master table, third table is order table and fourth table is order item table. And the fields or the column names in each of the tables are as follows.

The customer master table has the following fields, customer ID, the full name of the customer, city name, pin code, date of birth, entry date and sex. The item master table has the following fields which are brand, category, fit, style code, description, color, size and barcode. The order table has three fields which are order number, order date and customer ID. Then the fourth and last table that is the order item table has the following fields, order number, barcode, transaction type, transaction price, transaction quantity, tax and tax value and tax percentage. There are three more fields which is discount amount, discount percentage and MRP.

Metadata

Table and file names:

CustomerMaster

CustomerID FullName CityName Pincode DateofBirth EntryDate Sex

ItemMaster

Brand Category FIT STYLECODE Description COLOR Size Barcode

Order

OrderNo OrderDate CustomerID

OrderItem

OrderNo Barcode TrnType TrnPrice TrnQty TaxPer TaxValue
DiscAmount DiscPer MRP

So these are the various fields in all the tables. As you can see there are multiple foreign keys, so which are primary key in some table. So that is how we can analyze data from all these four tables. So before we move into how we work with all these data, I would want you to look into all these tables in excel format. I will just show that.

So this is the order table. So as I already said, this is the order number, this is the order date and this is the customer ID. So these are the three fields that are there in the order table. This is the next table which is order item table. As we see there are 7, 8, 9, 10 fields. There are 10 fields in this order item table which are order number, barcode, transaction type and so on. So if you scroll, this is a huge database. These are so many rows. As you can see, the rows run in thousands, almost 28,000 rows are there in this order item table. So it is a pretty big database.

The next table that I am going to show you is the item master which has about 8 columns and the rows again, it is running into thousands. So the field names in this table are brand, category, fit, style code, description, color, size and barcode. So these are the information in the item master table. The next table would be customer master table. In that, in this table you are having customer ID, full name, city name, pin code, date of birth, entry date and sex.

So this also will be running into thousands because there will be so many customers. So these are the 4 tables that we are going to work with. And before working with these tables, we need to have a basic understanding about how to work with the MySQL Workbench which is the database management system that we are using. It is a software which you have to install. So from the course outline itself, we will be getting to know how to install the MySQL Workbench. So please follow all the instruction and install the workbench so that we can proceed further. So this is how the MySQL workbench looks once it is installed.

So we will be working live on the workbench. This is just a snapshot of the workbench so that you can help, you know, I will help you understand what are the different parts of the workbench wherein you are going to work. So the right side of the screen which mentions as the panel view change, in this you can decrease, you know, hide the panels or whatever panels, like the output panel is in the below part.

So if you do not want, you can hide that. So it is basically to hide different panels and just view whatever panels you want in your screen. Then the query area that I have mentioned, that is the place where you are actually going to work with the relations or tables. So the query area is where you are going to type the queries and the output is, output area is the area below in the bottom part of the screen which is called as the action output. So here you will be getting all the errors if you are performing some, you know,

errors or if you are getting the correct query then you will get the correct output as well in that part.

Then there is a schema tab in which you can see all the schemas that you have created. So schema is nothing but a database name. So you can initially name your schema as whatever you want to name it. So I have named my schema as test, T-E-S-T. So you can just name it with a single syllable also. So it is up to you.

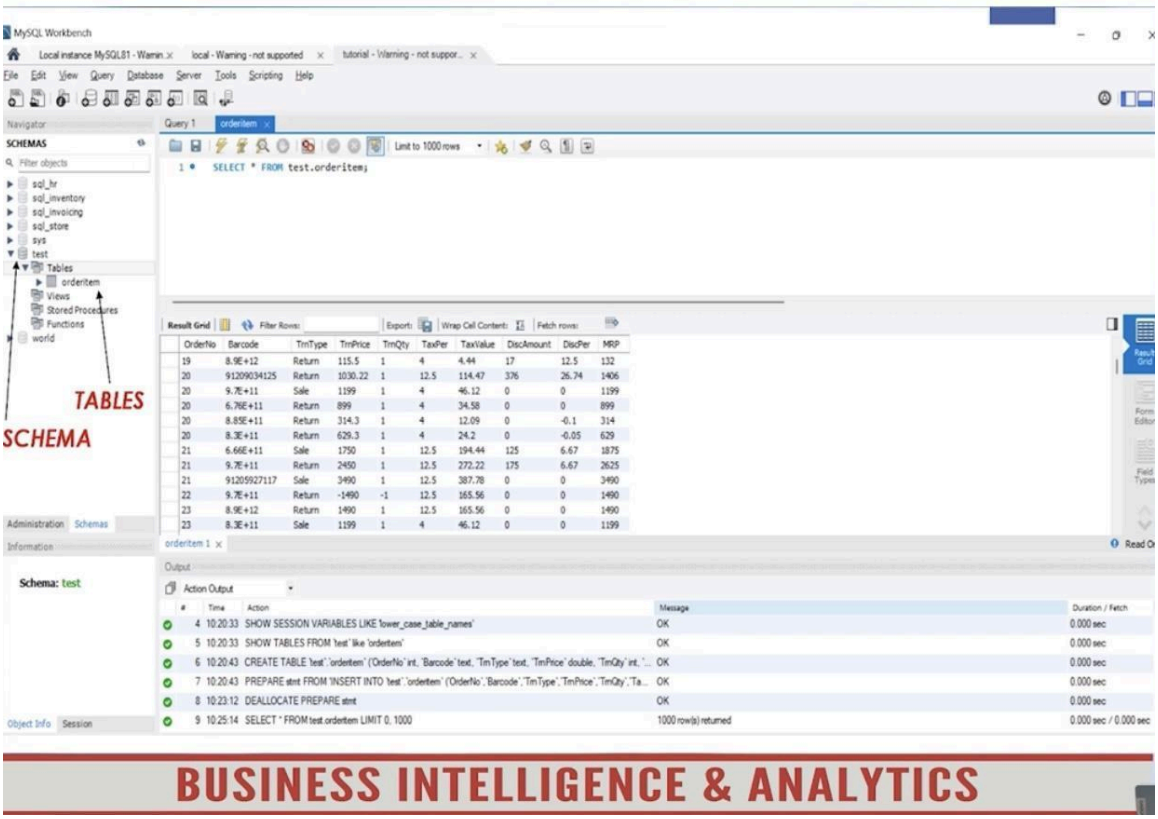
So there is a navigator panel in the left side. In that you will be able to do multiple functions like you can create tables, create schemas, insert tables, so all the, or even view all the columns that are there in the tables and so on. So it is basically a navigator panel. You can navigate between administration task as well as schema tasks. So this is a snapshot of something wherein we have typed some queries and got some output.

So you can see that the green tick in the output screen says that the query has been processed successfully without any errors and then all the functions that you have told to do, like the insert function etc has been performed successfully. So you can see a golden flash on top of the query area. That is the button for execute. So there are couple of keyboard shortcuts as well. If you do not want to use this, you can use that also or else after typing each query, you have to type that flash button which stands for execute option.

So that database kind of icon that is present on top, that is for creating a new schema. So initially after installing Workbench, what you have to do is you have to create a local instance. After that you have to create a schema wherein you can store all the tables that are there. As far as this particular case is concerned, we have only 4 tables. So all these 4 tables have to be imported into the schema that you have made.

So as I already told, see this example. In this, all the corresponding things that are mentioned along with the database icon like SQL_HR, SQL _inventory, test, sys, all those are schema names. So for this particular problem, I am using the test schema which I have given as the name. You can give whatever you want. So these are the, database is the one with the database icon and inside the database, you will have multiple tables.

So for example, here I am showing the order item table. So for querying from order item, how will I write if I want to see the whole table? The query for seeing every field is star. So select star from test.orderItem. That is test is the schema name, orderitem is the table name.



So from test schema, you are taking all the values of the orderitem table. That is the meaning of this query. So this is just a snapshot of the workbench. So yes, I will just show in live, how to work with the MySQL workbench.

So welcome to MySQL workbench. So this is how it looks once you have installed. So all before creating all these schemas and tables and everything, you have to initialize a connection first or create a local instance. I have already created that. So I am going into a local instance, which is SQL tutorial. And what I am doing is, initially, I have already formed the schema and also imported the tables, but I will just show how you are supposed to do it.

So if you go and right click on the navigator panel, you will see create schema. So that is where you have to create the schema. The schema name can be anything, say BIA. So BIA and I am applying it. So yes, it is going to create a schema called BIA.

It was executed and you click finish. That is all. So here we have got a BIA schema. Next thing what to do is, we have to import the tables to the BIA. If you open the tables in BIA, there are no tables because we have not yet imported it. Whereas in other

schemas that I have already formed, see the test schema and I will go into the tables.

So there are four tables already. I have already imported because it takes a lot of time to import, depending on the capacity of a laptop or the PC that you are using. It is going to take ample amount of time to import this table. So I have done it already, but I will just show a sample of how to do that. So you go into tables and you right click it and there is something called table data import wizard. So you have to click on that and you have to go and browse where the tables are there.

Just select the table, open it and click next. Here is the option for you to change the name of the table if you want because typing customer master every time for each query is going to take a lot of time. So you can just abbreviate it as CM and just click next. So you can reference the customer master table as CM henceforth. So the schema name that I am creating, that I have created now is called BIA and the table name is CM.

I am clicking next and it is going to show all the data types of each fields that are present in the table. For example, if I do not want a particular column, you can uncheck the column from here itself. So it is all your wish on how to import the table. I am importing it just like that. So it is preparing import and it will take a couple of minutes.

So since this table was a small table, it finished fast but multiple other tables are really big. So it will take time. So before you go into this, you can just import all the tables just like how I mentioned now and create a schema, import the 4 tables and you are ready to work for the problem now. So I will just click finish. For this example, I will use the test itself, test schema that I have already imported the tables here.

So we will just, we will begin the fun session wherein we will be discussing questions from the retail sense case. The question number 1 is, list quarter wise tax payable. So I want you to think for a few seconds how do we work with this. So for this, in order to get the answer for this or the query for this, we have to know the metadata.

This is the metadata. In this, we have to find the tax payable and that also quarter wise and to get the quarter wise information, we have already prefixed commands called quarter. So you can use that to get the answer. So I will show the answer and we will run it in workbench to see if it is executing properly. So I have typed the query.

So the query is select year by order date as year. So we are selecting the year from the order date as year. So if we give as, then the column name will become what we have mentioned in the codes. So the first column name is year and the second column name is what we are selecting, quarter of order date. So this is the prefixed command that is

already present.

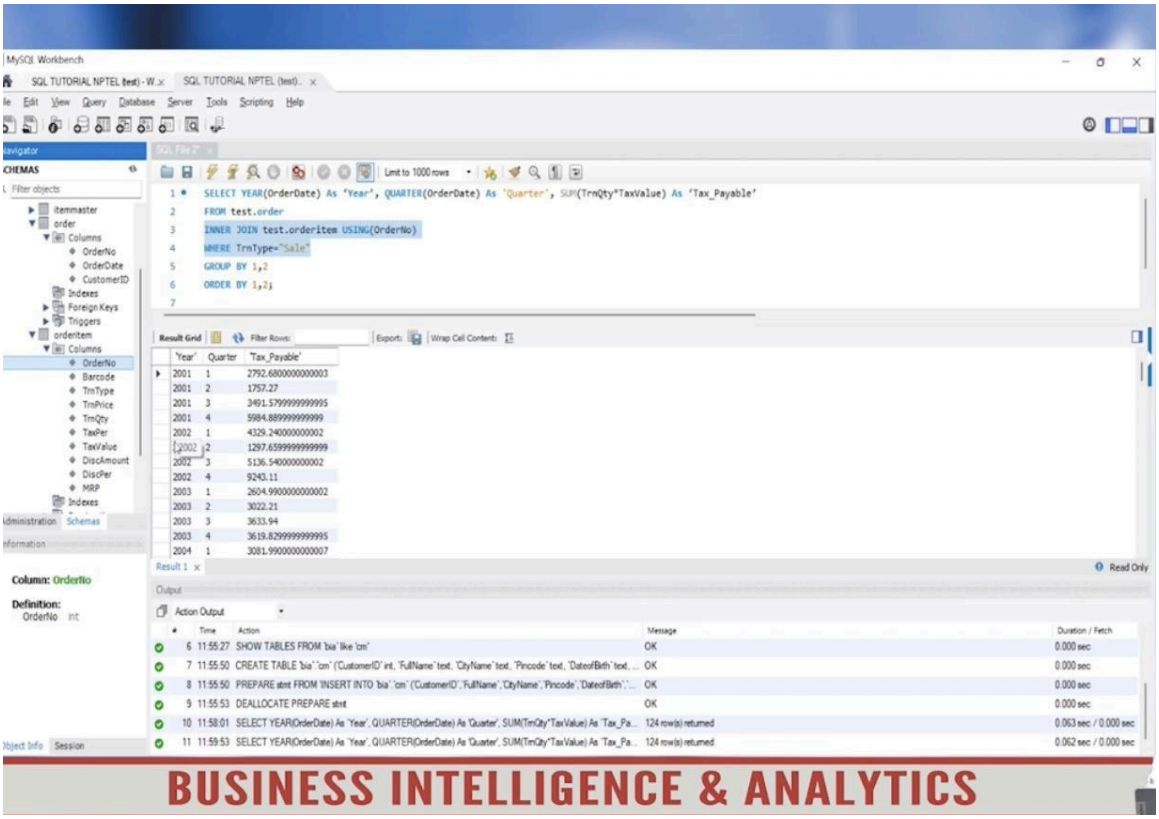
So you need not arrange yourself in quarters and all. It is a command that already exists. So you can just call it and quarter of order date as quarter. So the second column that we get in the output would be quarter. And how do we get the tax payable when we purchase something? The tax payable would be tax value into the transaction quantity.

So that is what we have done. The transaction quantity into tax value we are doing the sum of every such transactions and giving the output under the column which is under the name tax payable. So in order to do this, we need to select which schema, test schema, which table, order table. So the query comes as from test.order. So I will run the query again for you to see. So I have typed the query here and I am just executing it by clicking the flash. So here I am selecting all these three fields and from which table, order table. If you want to see which all columns are there in each table, you can just click this arrow. See for example, now we are able to see what are there in the order table.

If we click these columns, we will know. We have order number, order date and customer ID. So there are three fields in this order table. So if you are wanting to see the metadata before you write some query, then you can just go into this navigator panel and see what are the columns in each table that are present. So here we are doing inner join on order item. So basically what is order item containing? It is having order number and all these tax details, that is the tax value and transaction quantity.

Both these are there in order item table and not in order table. So what we have to do? We have to do a inner join. So what we are doing is we are inner joining both order and order item. Using what? There should be field on which you are performing this. So that is order number.

So if you see order number will be common for both the tables. Are you able to see that? In order table also order number is there, in order item table also order number is there. And finally what we are doing is, we are giving a filtering clause, that is where clause to filter out only those details where the transaction type is sale. Basically you have to sell something in order to get the tax payable details. So if it is a sale, then it will show it in the output.



And we are also doing the group by and order by functions. So you know what that stands for. And this is the output. As you can see year is there, quarter is there, then tax payable is there. So if you scroll and see you will have all the years and all the quarter details and the corresponding tax payable for each quarter.

So that was the first question. Next we will move on to the second question. So the second question for the day is, list customers and their average order value. So I will give you a few seconds to think about how do we calculate the average of the order value for each of the customer and we have to list the customers according to their customer name and the details. We will see how we work this query on workbench.

So I have typed the query here. So it is select customer ID full name. This is for extracting the customer details. So the first column of the output would be customer ID. The second column of the output would be customer name and the third column would be what? What are we supposed to find out now? It is the average order value for each of the customers. So how do we calculate the average order value? First of all, we will decrease the tax value or subtract the tax value from the transaction price so that we get the actual price of each item that they have purchased plus we will multiply that with the

quantity. Say you have purchased 2 soap for 10 rupees which had an additional 2 percent tax.

So we will first subtract the 2 percent tax and then multiply the price and the transaction quantity as, and we will perform the average operation which is a prefixed operation already. So you just have to pass the avg command, so that it will return the average value. So that is the third column. And in this, it is getting a little more complex as you see because we have to join, perform inner join on 2 tables along with the order table. Why is it, why do we have to take the customer master also now? Because we need the customer names and also the customer ID.

That is why we have to perform the inner join on customer master also. So if you see the tables, the customer master table has the customer ID and the full name of the customer. So we need the customer master table and if you see the order item table, you can see that the order number, the transaction price, the transaction quantity, the tax, everything is got from the order item table. So what we need is, we have to perform the inner join with both order item and also the customer master table and again do the filtering operation where the transaction type is sale. And also we are grouping it by, grouping it and ordering it using the group by and order by commands.

So this is the output that we have got. We have got the customer ID. As you can see, I am scrolling through the customer ID. These are the customer names and also we have got the average order value using the avg, avg command that we have already passed. So you can maybe note this query, so that you can work with the workbench when you are doing it. So this is the query. Select customer ID, full name, average of transaction quantity into within brackets transaction price minus tax value as averaged order value from test.order. We are inner joining two tables, inner join test.order item using order number and inner joining test.customer master using customer ID where transaction type is sale and then grouping by and ordering by. So that is it. This was the second question.

We will move on to the third question which is going to get a little more complex and we will come to know why it is getting complex and why there are, why we have to move to different kinds of operations rather than SQL in order to analyze multidimensional queries.

So we will get into that at the end. So we will go to the third question now. Before that I would like to show the answers for the first and second question. This is the answer for the first question. If you want to note down, you can note it down or if you want to work it by yourself, it is well and good. So this was the answer for the first question and this is the answer for the second question.

So we have finished up till here and we will move on to the next question. So herein what we are doing is, just ignore the things that are written in the brackets because we are moving into multidimensional query. So there will be multiple tables wherein there are no common fields. So we have to give more and more complex queries in order to get a small insight of the database. So what happens is that what we have got, what we could have got through a simple query is getting more and more complex and that is why we will discuss in the next class. Sir will be taking the next class on online analytical processing for multidimensional queries using data cube and all and you will come to know what is the use of those because as we solve these queries, you will understand how many inner joins and all those multidimensional commands that we have to use in order to get a small insights.

So that is why we are going into much more advanced process of data analytics like online OLAP, which is online analytical process in which data is stored in the form of, represented in the form of data cubes and so on. So you can just think about how we find the answer for this query which is to list the customers who were most frequent visitors during quarter 4. So this is the question. Herein we have to get the answer according to, for a particular quarter only, which is quarter 4 and we have to list the customers who were the most frequent also.

So we will see how to work that with the workbench. So this is the answer that we have, this is the query for processing the third question. So we have to group the customers who were the most frequent customers during quarter 4. So for that we are selecting the customer ID and full name in order to get the customer details and there is a function called count which will obviously return the count of the order number. So basically you will come to know how many times they have come.

So that is what is meant by count, as the frequency. So it will return the column with the name frequency. That is what is meant by ‘as frequency’ within quotes. So we are using only one table to inner join. So we are joining order table with customer master table using the key which is customer ID and where the quarter is already given.

The quarter is order, the order rate should be 4. So that is how we get the quarter which is the fourth quarter and we are grouping by and ordering by in descending order so that we get the most frequent initially. So that is how we get it. And you can see there is 1, 1, 1, 3, 1, 3 frequency of the people who are not coming regularly so they are mentioned in the first one and as you see ,as I am scrolling through the customers you can see that the frequency is decreasing. Initially it was 4 and then 3, 2 and 1.

So it is because we have mentioned as order by descending. So that is why it is going like that and this is only for quarter 4. You can do this similarly for other quarters or on bimonthly basis or whatever you want. You can do that. There are commands for that.

So this is the answer and if you want you can have a look at the answer. This is the query that we perform now on workbench. So you can note down this answer or even better, you can try on your own. So what we have done is we have listed all the customers, their name, their ID and also their count of times that they have purchased things and come to the shop as frequency by inner joining two tables which is order table and customer master table using the key customer ID wherein we have given the filter clause for getting only the data where the quarter is 4. So that is what we have done for this question. We will move on to the next question which is, list the customers who were most frequent visitors during each quarter of the year.

So basically in the previous question we did a slicing wherein we have sliced the quarters into 4 and only picked the quarter which was 4. Here there is no slicing. We have to get the output, wherein we have to list the customers during each quarter and not just the quarter 4. So you can just give a thing, you can just think about how to perform this query.

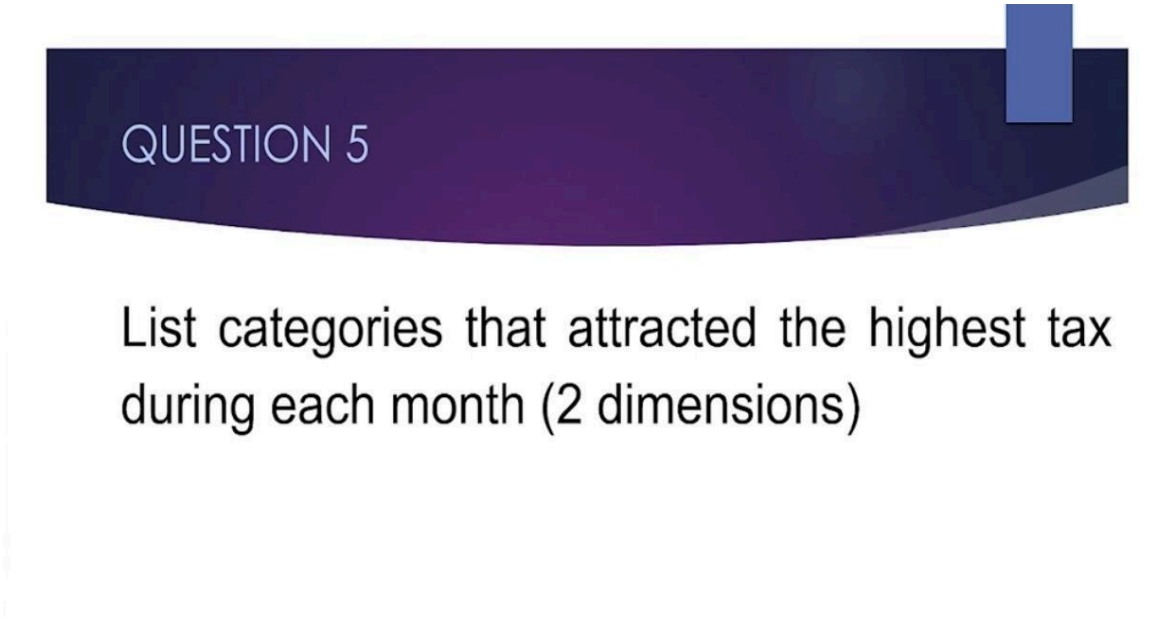
We will work this query on workbench now. So you can see we are going to type the query for getting the answer. So we have got the answer. How do we do this? In this we are selecting the order date as year and then we are getting the quarters, quarter wise information so that we can get it like quarter 1, this is the detail, quarter 2, quarter 3, quarter 4. So we are just extracting the quarter wise details and then printing the customer details which is customer ID, customer name and also we have to get the count of the order number which will basically tell the frequency of the visit.

So we are printing that as the frequency column. So herein we need only two tables. So we are joining order table and the customer master table. Similarly, just like the previous question but the only detail, only difference is that we are printing for all the quarters and not just where quarter equal to 4. So it is almost the same command and then if you run it, you are getting year wise, then quarter wise, then if you just scroll you can see that it is printing for all the first quarter information first and then second quarter and then third and then fourth for a particular year then going to next year.

So it is doing that. That is the importance of order by and group by clauses. So that we are getting it and then there is the customer ID, the customer name and the frequency of their visit. So basically it is not just for one quarter alone but it is printing for all the quarters. So I am scrolling it and it is a big output. So if you want to see the answer for

this question, it is this.

You can work with the workbench and see if you are getting the same output. So the fifth question is list categories that attracted the highest tax during each month. So now we are slowly moving into multidimensional queries. So it is going to become a little complicated because there will be no commonalities between two tables. So we have to do multiple operations and it is going to become a little complicated. So this question is about listing the categories which have attracted the highest tax and this also we have to do it month wise.



So we will see the answer now and we will perform the query on workbench. Yes, so we have printed the query which is, here there are multiple inner joins as you see because we have to get different things from different table and in order to get that, we have to perform multiple inner joins using certain keys. So here we have three tables which we are joining. One is order item table, another is item master table and another is order table. So you can see the columns of item master table which is brand, category, fit and all those and in this, what are we extracting? We are extracting the year or order date that is from the order item table. If you see the order item table, you can see that there is order number and in order table, there is order date also.

So basically you are joining both those tables to get the order date as year and then you

have to print it month wise. So you have to get the month of the order date. So you can just call the month and within brackets you can give order date, so that it will extract the month and give you. So you can print it as month, so that you will get the column with name as month.

Then you have to print the category, like what kind of category that particular item belongs. So it can be training, singles, outdoor and all those categories are there. So you have to print it category wise and also you have to get the tax value. So for tax value, we are multiplying it with transaction quantity. So if there is two quantities of soaps that you are buying, so it is double the tax value and we are printing it and since we need all those, all these records from three different tables, we are going to join three tables which are order item, item master and the order table. For item master, we are using the bar code key and for the order table, we are using order number because that is what is common between these and again where transaction type is sale and group by order by functions.

So this is how we get all the year wise information, then classified by month wise. So since we are grouping by year, initially you will get all those details for year 2001 then 2002 and 2003, that is how the group by works. So then you are getting it month wise, then the category wise, then the tax value for each category, that is what we have got is the query output. You can note the answer for the fifth question that we just performed in workbench and we will go on to the next question now. Question 6 which is list categories which enjoyed highest average discount during January among female customers. So this is becoming a little more, adding more dimensions into the query that we are going to write.

So basically it is a three dimensional query. You will come to know what three dimensional, two dimensional queries are when working with OLAP or online analytical processing, wherein data is formed on the, data is stored on the form of cubes. So basically it is a 3D structure or wherein you can extract data using multiple OLAP operations. So you will come to know more about that in the next class. But as of now you can write using SQL, on listing categories which enjoyed the highest average discount. So whatever got the highest average discount you have to list it category wise, during only one month which is January and only the sex should be female.

So that is what we are going to write now. Let us work it on workbench. So this is the query that I have written and the query is greater than, looking more complex than the output that we have got. So that is why we need other kind of OLAP operations which will make this much simpler than what we are doing now. So in this, we are selecting the category. So basically the first column should be category because we wanted

category wise and we are, what was the question? So we want the discount which attracted the highest discount.

So which category attracted the highest discount, that is what we want. So from the, for this how many tables? Just for getting a small information we are joining three tables which is order item, item master and order table. So wherein month is 1 because we want only during the January month. So basically we are using the order table in order to get the order details like order number and order date. Then we are also using the item master table which will be having the category as a key.

So we are extracting the category from the item master table and also we are printing it for the January month. And we are calculating the discount, as discount into transaction quantity because we might have purchased more than one item. So the singles category has had the highest amount of discount which is 63 that we have got as output for this query. So with that we come to the end. So before closing today's session, I would want you to go through the answer of the question 6 also and work out all these questions by yourself in workbench so that you get accustomed with MySQL Workbench which would be, multiple of you would be wanting to start your data science or data analyst career.

And this is the first stepping stone because you are starting to analyze 2D data which is stored in the form of tables. So this is the, I would call it the first stepping stone towards your data analysis journey. So we will have more complex operations which will be taught in the next class. So that is all for today. Thank you.