How to save files

MySQL(.SQL): After executing all the commands/answers, click on the file menu on the top left and save as Yourname.sql.

As shown below:

Order Management Schema Details

This document captures the scenario of simple order management functionality of an online retail store.

Typical purchase scenario: A customer places an order for N products specifying quantity for each line item of the order. Every product belongs to a product class (or category). All products ordered in one order, are shipped to customer's address (in India or outside) by a shipper in one shipment. Order can be paid using either Cash, Credit Card or Net Banking.

There can be customers who may not have placed any order. Few customers would have cancelled their orders (As a whole order, no cancellation of individual item allowed). Few orders may be 'In process' status. There can also be products that were never purchased.

Shippers use optimum sized cartons (boxes) to ship an order, based on the total volume of all products and their quantities. Dimensions of each product (L, W, H) is also stored in the database. To keep it simple, all products of an order are put in one single appropriately sized carton for shipping.

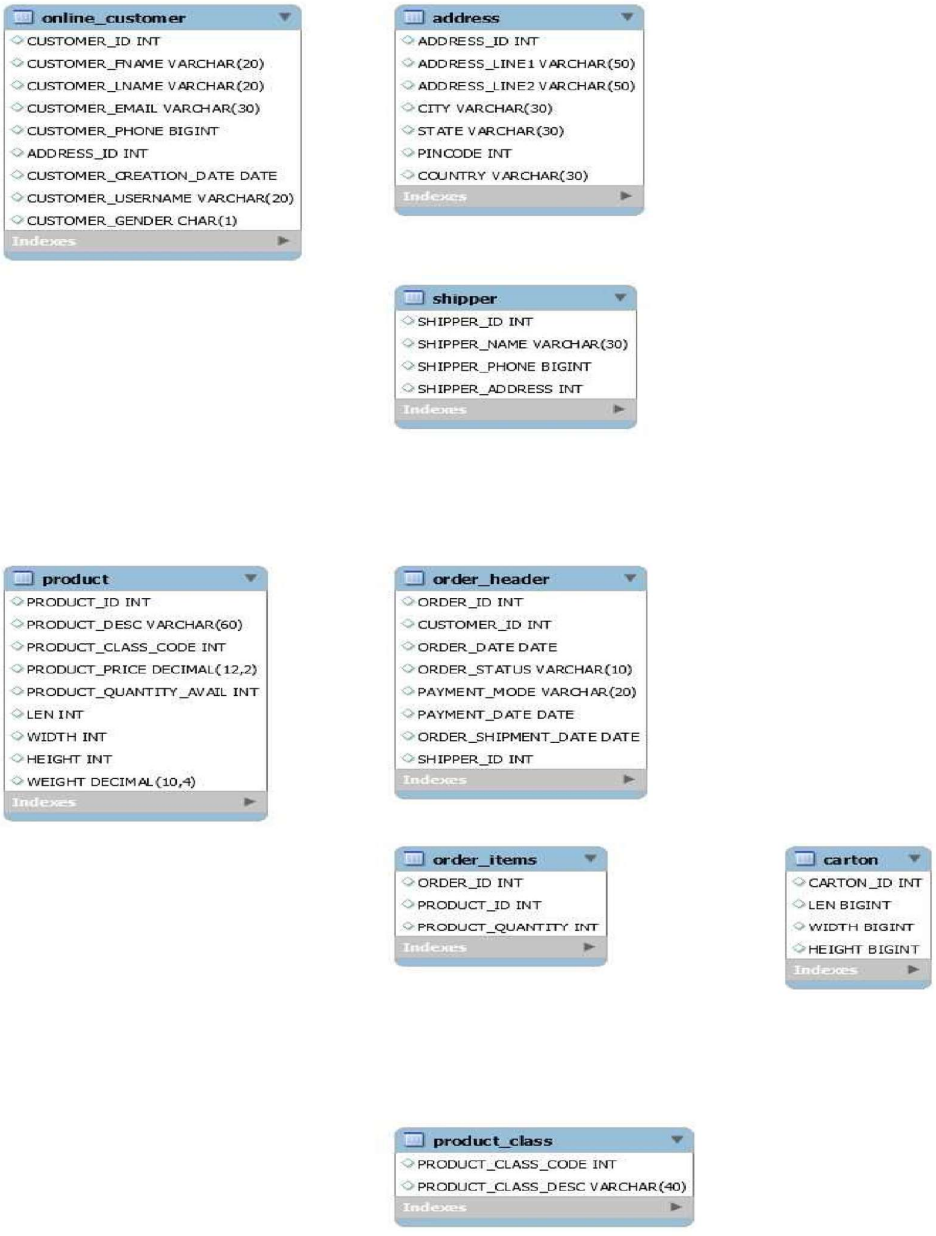
Project- (MYSQL)

You are hired by a chain of online retail stores "Reliant retail limited". They provided you with "orders" database and seek answers to the following queries as the results from these queries will help the company in making data driven decisions that will impact the overall growth of the online retail store.

All the questions comes under MYSQL and the queries should be executed in MYSQL. (SQL Script -orders.sql)

All Questions carry 8 marks. Total Marks (8 x 10) = 80

ER Diagram


online_customer
CUSTOMER_ID INT
CUSTOMER_FNAME VARCHAR(20)
CUSTOMER_LNAME VARCHAR(20)
CUSTOMER_EMAIL VARCHAR(30)
CUSTOMER_PHONE BIGINT
ADDRESS_ID INT
CUSTOMER_CREATION_DATE DATE
CUSTOMER_USERNAME VARCHAR(20)
CUSTOMER_GENDER CHAR(1)
Indexes
address
ADDRESS_ID INT
ADDRESS_LINE1 VARCHAR(50)
ADDRESS_LINE2 VARCHAR(50)
CITY VARCHAR(30)
STATE VARCHAR(30)
PINCODE INT
COUNTRY VARCHAR(30)
Indexes
shipper
SHIPPER_ID INT
SHIPPER_NAME VARCHAR(30)
SHIPPER_PHONE BIGINT
SHIPPER_ADDRESS INT
Indexes
product
PRODUCT_ID INT
PRODUCT_DESC VARCHAR(60)
PRODUCT_CLASS_CODE INT
PRODUCT_PRICE DECIMAL(12,2)
PRODUCT_QUANTITY_AVAIL INT
LEN INT
WIDTH INT
HEIGHT INT
WEIGHT DECIMAL(10,4)
Indexes
order_header
ORDER_ID INT
CUSTOMER_ID INT
ORDER_DATE DATE
ORDER_STATUS VARCHAR(10)
PAYMENT_MODE VARCHAR(20)
PAYMENT_DATE DATE
ORDER_SHIPMENT_DATE DATE
SHIPPER_ID INT
Indexes
order_items
ORDER_ID INT
PRODUCT_ID INT
PRODUCT_QUANTITY INT
Indexes
carton
CARTON_ID INT
LEN BIGINT
WIDTH BIGINT
HEIGHT BIGINT
Indexes
product_class
PRODUCT_CLASS_CODE INT
PRODUCT_CLASS_DESC VARCHAR(40)
Indexes

SQL Graded Project Questions

1. Write a query to display customer full name with their title (Mr/Ms),

-- both first name and last name are in upper case, customer email id,

-- customer creation date and display customer's category after applying below

-- categorization rules:

-- 1) IF customer creation date Year <2005 Then Category A

-- 2) IF customer creation date Year >=2005 and <2011 Then Category B

-- 3) IF customer creation date Year>= 2011 Then Category C

Hint: Use CASE statement, no permanent change in table required.

NOTE: TABLES to be used - ONLINE_CUSTOMER TABLE]

#2. Write a query to display the following information for the products, which have not been sold: product_id, product_desc, product_quantity_avail, product_price, inventory values

-- (product_quantity_avail*product_price), New_Price after applying discount as per below criteria.

-- Sort the output with respect to decreasing value of Inventory_Value.

-- 1) IF Product Price > 200,000 then apply 20% discount

-- 2) IF Product Price > 100,000 then apply 15% discount

-- 3) IF Product Price =< 100,000 then apply 10% discount

Hint: Use CASE statement, no permanent change in table required.

[NOTE: TABLES to be used - PRODUCT, ORDER_ITEMS TABLE]

#3. Write a query to display Product_class_code, Product_class_description, Count of Product type in each product

-- class, Inventory Value (p.product_quantity_avail*p.product_price). Information should be displayed for only those product_class_code which have more than 1,00,000

-- Inventory Value. Sort the output with respect to decreasing value of Inventory_Value.

NOTE: TABLES to be used - PRODUCT_CLASS, PRODUCT_CLASS_CODE]

#4. Write a query to display customer_id, full name, customer_email, customer_phone and country of customers who

-- have cancelled all the orders placed by them

-- (USE SUB-QUERY)

-- [NOTE: TABLES to be used - ONLINE_CUSTOMER, ADDRESSS, OREDER_HEARDER]

#5. Write a query to display Shipper name, City to which it is catering, num of customer catered by the

-- shipper in the city and number of consignments delivered to that city for Shipper DHL

-- [NOTE: TABLES to be used - SHIPPER,ONLINE_CUSTOMER, ADDRESSS, OREDER_HEARDER]

#6. Write a query to display product_id, product_desc, product_quantity_avail, quantity sold, quantity available and

-- show inventory Status of products as below as per below condition:

-- a. For Electronics and Computer categories, if sales till date is Zero then show

-- 'No Sales in past, give discount to reduce inventory', if inventory quantity is less than 10% of quantity sold,

-- show 'Low inventory, need to add inventory', if inventory quantity is less than 50% of quantity sold,

-- show 'Medium inventory, need to add some inventory', if inventory quantity is more or equal to 50% of quantity sold,

-- show 'Sufficient inventory'

-- b. For Mobiles and Watches categories, if sales till date is Zero then show

-- 'No Sales in past, give discount to reduce inventory', if inventory quantity is less than 20% of quantity sold,

-- show 'Low inventory, need to add inventory', if inventory quantity is less than 60% of quantity sold,

-- show 'Medium inventory, need to add some inventory', if inventory quantity is more or equal to 60% of quantity sold,

-- show 'Sufficient inventory'

-- c. Rest of the categories, if sales till date is Zero then show

-- 'No Sales in past, give discount to reduce inventory', if inventory quantity is less than 30% of quantity sold,

-- show 'Low inventory, need to add inventory', if inventory quantity is less than 70% of quantity sold,

-- show 'Medium inventory, need to add some inventory', if inventory quantity is more or equal to 70% of quantity sold,

-- show 'Sufficient inventory'

-- (USE SUB-QUERY)

-- [NOTE: TABLES to be used - PRODUCT, PRODUCT_CLASS, ORDER_HEADER]

#7. Write a query to display order_id and volume of the biggest order (in terms of volume) that can fit in carton id 10

-- [NOTE: TABLES to be used - CARTON, ORDER_ITEMS, PRODUCT]

#8. Write a query to display customer id, customer full name, total quantity and total value (quantity*price) shipped

-- where mode of payment is Cash and customer last name starts with 'G'

-- [NOTE: TABLES to be used - ONLINE_CUSTOMER, ORDER_ITEMS, PRODUCT, ORDER_HEADER]

#9. Write a query to display product_id, product_desc and total quantity of products

-- which are sold together with product id 201 and are not shipped to city Bangalore and New Delhi.

-- Display the output in descending order with respect to tot_qty.

-- (USE SUB-QUERY)

-- [NOTE: TABLES to be used - order_items, product,order_head, online_customer, address]

#10 Write a query to display the order_id,customer_id and customer fullname

-- as total quantity of products shipped for order ids which are even

-- and shipped to address where pincode is not starting with "5"

-- [NOTE: TABLES to be used - online_customer,Order_header, order_items,address]