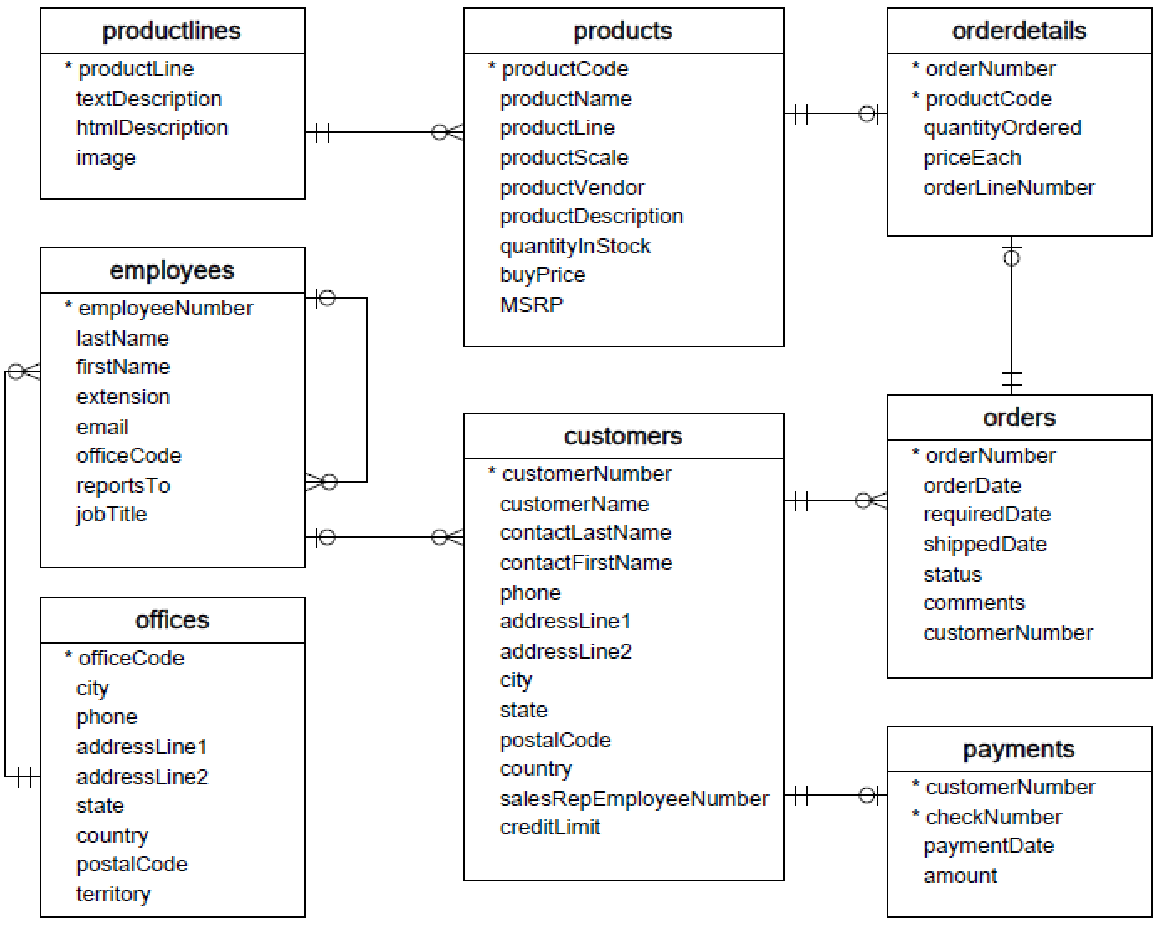
Database week 4 Answers

1.1 Assignment 1 (easy) Model Cars

Download WC_Week4_1_ModelCars.sql
Import it into PhpMyAdmin.

Database Schema:



It concerns the database of a factory that makes scale models of, among other things, cars, motorcycles and trains. Tables for customers, products, orders and linking tables are available.

1. Show the EmployeeNumber, full name and email of all the ‘Sales Representatives’ (Sales Rep).

```
SELECT EmployeeNumber, firstname, lastname
FROM Employees
WHERE jobtitle = 'Sales Rep';
```

2. Show all orders that have been cancelled (status cancelled).

```
SELECT ordernumber
FROM orders
```

```
WHERE `status`='cancelled';
```

3. Show all the data from all the Canadian customers.

```
SELECT *
FROM customers
WHERE country = 'Canada';
```

4. Show the productname and productline of all products that are either classic cars or vintage cars.

```
SELECT ProductName, ProductLine
FROM products
WHERE productline = 'Classic Cars'
      OR productline = 'Vintage Cars';


SELECT ProductName, ProductLine
FROM products
WHERE productline IN ('Classic Cars','Vintage Cars');
```

5. Write a query that joins the customer and employees table. You may Select all data for this query.

```
SELECT *
FROM customers c
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber;

SELECT *
FROM customers c, employees e
WHERE c.salesRepEmployeeNumber = e.employeeNumber;
```

6. Show the contacts of all the customers that have no credit limit.

```
SELECT contactFirstname, contactLastname
FROM customers
WHERE creditLimit = 0;
```

7. Show the officecode, city and country of all offices that have a ‘02’ in their postalcode.

```
SELECT officecode, city, country
FROM offices
WHERE postalcode like '%02%';
```

8. Show a list of all Ford model classic cars. Show the productname and the productline.

```
SELECT ProductName, ProductLine
FROM products
WHERE productname LIKE '%Ford%' AND productline = 'Classic Cars';
```

1.2 Assignment 2 (Medium) Model Cars

1. Show all payments made during the year 2004.

```
SELECT *
FROM payments
WHERE paymentdate BETWEEN '2004-01-01' AND '2004-12-31';


SELECT *
FROM payments
WHERE paymentdate >= '2004-01-01' AND paymentdate <='2004-12-31';
```

2. Show all customer names and customernumbers that have a 2nd adressline.

```
SELECT *
FROM customers
WHERE addressLine2 IS NOT NULL
```

1.3 Assignment 3 (Hard) Model Cars

1. Write an update query that assigns the EmployeeNumber of Julie Firrelli to all customers from the Netherlands.

```
UPDATE customers SET SalesRepEmployeenumber =
(
    SELECT Employeenumber
    FROM employees
    WHERE firstname = 'Julie'
            AND lastname ='Firrelli'
)
WHERE country = 'Netherlands';
```

1.4 Assignment 4 Marcia’s Dry Cleaning

Download WC_Week4_2_Marcia'sDrycleaning_CreateTables.sql en
WC_Week4_2_Marcia'sDrycleaning_Insert.sql
Import it into PhpMyAdmin

A. Show all data in each of the tables.

```
SELECT *
FROM customers;
```

Note there are two customers both named Betsy Miller.

```
SELECT *
FROM invoice;
```

```
SELECT *
FROM INVOICE_ITEM;
```

B. List the LastName, FirstName, andPhone of allcustomers.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER;
```

C. List the LastName, FirstName, and Phonefor all customerswith a FirstName of“Nikki”.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER
WHERE FirstName = 'Nikki';
```

D. List the LastName, FirstName, Phone,DateIn, and DateOut of all orders in excess of$100.00.

```
SELECT LastName, FirstName, Phone, DateIn, DateOut
FROM CUSTOMER, INVOICE
WHERE TotalAmount >100
    AND CUSTOMER.CustomerID = INVOICE.CustomerNumber;
```

E. List the LastName, FirstName, and Phoneof allcustomerswhose first name startswith 'B'.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER
WHERE FirstName LIKE 'B%';
```

F. List the LastName, FirstName, and Phone of all customers whose last name includes the characters 'cat'

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER
WHERE LastName LIKE '%cat%';
```

G. List the LastName, FirstName, and Phone for all customers whose second and third digits (from the left) of their phone number are 23. For example, any phone number with an area code of '723' would meet the criteria.

Note that since the phone numbers in this database include the area code, we are really finding phone numbers with '23' as the second and third numbers in the area code. We could, of course, write statements to find '23' in the prefix or in the 4-digit sequence portion of the phone number.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER
WHERE Phone LIKE '_23%';
```

H. Determine the maximum and minimum TotalAmount.

```
SELECT MAX (TotalAmt) AS MaxTotalAmount,
MIN (TotalAmt) AS MinTotalAmount
FROM INVOICE;
```

I. Determine the average TotalAmount.

```
SELECT AVG (TotalAmt) AS AvgTotalAmount
FROM INVOICE;
```

J. Count the number of customers.

```
SELECT Count (*)AS NumberOfCustomers
FROM CUSTOMER;
```

K. Group customers by LastName and then by FirstName.

```
SELECT LastName, FirstName
FROM CUSTOMER
GROUP BY LastName, FirstName;
```

L. Count the number of customers having each combination of LastName and FirstName.

```
SELECT LastName, FirstName,
COUNT (*) AS Last_First_Combination_Count
FROM CUSTOMER
GROUP BY LastName, FirstName;
```

M. Show the LastName, FirstName, and Phone of all customers who have had an order with TotalAmount greater than $100.00. Use a subquery. Present the results sorted by LastName in ascending order and then FirstName in descending order.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER
WHERE CustomerID IN
(
    SELECT CustomerNumber
    FROM INVOICE
    WHERE TotalAmount > 100
)
ORDER BY LastName, FirstName DESC;
```

N. Show the LastName, FirstName and Phone of all customers who have had an order with TotalAmount greater than $100.00. Use a join, but do not use JOIN ON syntax. Present the results sorted by LastName in ascending order and then FirstName in descending order.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER, INVOICE
WHERE CUSTOMER.CustomerID = INVOICE.CustomerNumber
AND TotalAmount > 100
ORDER BY LastName, FirstName DESC;
```

O. Show the LastName, FirstName and Phone of all customers who have had an order with TotalAmount greater than $100.00. Use a join using JOIN ON syntax. Present the results sorted by LastName in ascending order and then FirstName in descending order.

```
SELECT CUSTOMER.LastName, CUSTOMER.FirstName, CUSTOMER.Phone
FROM CUSTOMER
JOIN INVOICE ON CUSTOMER.CustomerID = INVOICE.CustomerNumber
WHERE INVOICE.TotalAmount>100;
```

P. Show the LastName, FirstName and Phone of all customers who have had an order with an Item named “Dress Shirt”. Use a subquery. Present the results sorted by LastName in ascending order and then FirstName in descending order.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER
WHERE CustomerID IN
(
    SELECT CustomerNumber
    FROM INVOICE
    WHERE InvoiceNumber IN
            (
                SELECT InvoiceNumber
                FROM INVOICE_ITEM
                WHERE Item = 'Dress Shirt'
            )
)
ORDER BY LastName, FirstName DESC;
```

Q. Show the LastName, FirstName and Phone of all customers who have had an order with an Item named “Dress Shirt”. Use a join, but do not use JOIN ON syntax. Present the results sorted by LastName in ascending order and then FirstName in descending order.

```
SELECT LastName, FirstName, Phone
FROM CUSTOMER, INVOICE, INVOICE_ITEM
WHERE CUSTOMER.CustomerID = INVOICE.CustomerNumber
AND INVOICE.InvoiceNumber = INVOICE_ITEM.InvoiceNumber
AND INVOICE_ITEM.Item = 'Dress
ORDER BY LastName, FirstName DESC
```

1.5 Assignment 5 Morgan Import

Download WC_Week4_3_MorganImporting_CreateTables.sql en WC_Week4_3_MorganImporting_Insert.sql
Import it into PhpMyAdmin

A. Show all data in each of the tables.

```
SELECT *
FROM ITEM;

SELECT *
FROM SHIPMENT;

SELECT *
FROM SHIPMENT_ITEM;
```

B. List the ShipmentID, ShipperName, and ShipperInvoiceNumber of all shipments.

```
SELECT ShipmentID, ShipperName, ShipperInvoiceNumber
FROM SHIPMENT;
```

C. List the ShipmentID, ShipperName, and ShipperInvoiceNumber for all shipments that have an insured value greater than $10,000.00.

```
SELECT ShipmentID, ShipperName, ShipperInvoiceNumber
FROM SHIPMENT
WHERE InsuredValue > 10000;
```

D. List the ShipmentID, ShipperName, and ShipperInvoiceNumber of all shippers whose name starts with “AB”.

```
SELECT ShipmentID, ShipperName, ShipperInvoiceNumber
FROM SHIPMENT
WHERE ShipperName LIKE 'AB%';
```

E. Assume DepartureDate and ArrivalDate are in the format MM/DD/YY. List the ShipmentID, ShipperName, ShipperInvoiceNumber, and ArrivalDate of all shipments that departed in December.

```
SELECT ShipmentID, ShipperName, ShipperInvoiceNumber, ArrivalDate
FROM SHIPMENT
WHERE DepartureDate LIKE '12%';
```

F. Assume DepartureDate and ArrivalDate are in the format MM/DD/YY. List the ShipmentID, ShipperName, ShipperInvoiceNumber, and ArrivalDate of all shipments that departed on the tenth day of any month.

```
SELECT ShipmentID, ShipperName, ShipperInvoiceNumber, ArrivalDate
FROM SHIPMENT
WHERE DepartureDate LIKE '___10%';
```

G. Determine the maximum and minimum InsuredValue.

```
SELECT MAX (InsuredValue) AS MaxInsuredValue,
MIN (InsuredValue) AS MinInsuredValue,
FROM SHIPMENT;
```

H. Determine the average InsuredValue.

```
SELECT AVG (InsuredValue) AS AvgInsuredValue
FROM SHIPMENT;
```

I. Count the number of shipments.

```
SELECT COUNT (*) AS NumberOfShipments
FROM SHIPMENT;
```

J. Show ItemID, Description, Store, and a calculated column named USCurrencyAmount that is equal to LocalCurrencyAmount times the ExchangeRate for all rows of ITEM.

```
SELECT ItemID, Description, Store,
LocalCurrencyAmount * ExchangeRate AS USCurrencyAmount
FROM ITEM;
```

K. Group item purchases by City and Store.

```
SELECT City, Store
FROM ITEM
GROUP BY City, Store;
```

L. Count the number of purchases having each combination of City and Store.

```
SELECT City, Store,
COUNT (*) AS City_Store_Combination_Count
FROM ITEM
GROUP BY City, Store;
```

M. Show the ShipperName, ShipmentID and DepartureDate of all shipments that have an item with a value of $1,000.00 or more. Use a subquery. Present results sorted by ShipperName in ascending order and then DepartureDate in descending order.

```
SELECT ShipperName, ShipmentID, DepartureDate
FROM SHIPMENT
WHERE ShipmentID IN
```

```
(SELECT ShipmentID
FROM SHIPMENT_ITEM
WHERE Value >= 1000)
ORDER BY ShipperName, DepartureDate DESC;
```

N. Show the ShipperName, ShipmentID, and DepartureDate of all shipments that have an item with a value of $1000.00 or more. Use a join. Present results sorted by ShipperName in ascending order and then DepartureDate in descending order.

```
SELECT ShipperName, SHIPMENT.ShipmentID, DepartureDate
FROM SHIPMENT, SHIPMENT_ITEM
WHERE SHIPMENT.ShipmentID = SHIPMENT_ITEM.ShipmentID
    AND (Value = 1000 OR Value > 1000)
ORDER BY ShipperName, DepartureDate DESC;
```

O. Show the ShipperName, ShipmentID, and DepartureDate of the shipments for items that were purchased in Singapore. Use a subquery. Present results sorted by ShipperName in ascending order and then DepartureDate in descending order.

```
SELECT ShipperName, ShipmentID, DepartureDate
FROM SHIPMENT
WHERE ShipmentID IN
(
    SELECT ShipmentID
    FROM SHIPMENT_ITEM
    WHERE ItemID IN
        (SELECT ItemID
        FROM ITEM
        WHERE City = 'Singapore')
)
ORDER BY ShipperName, DepartureDate DESC;
```

P. Show the ShipperName, ShipmentID, and DepartureDate of all shipments that have an item that was purchased in Singapore. Use a join, but do not use JOIN ON syntax. Present results sorted by ShipperName in ascending order and then DepartureDate in descending order.

```
SELECT DISTINCT ShipperName, SHIPMENT.ShipmentID, DepartureDate
FROM SHIPMENT, SHIPMENT_ITEM, ITEM
WHERE SHIPMENT.ShipmentID = SHIPMENT_ITEM.ShipmentID
AND SHIPMENT_ITEM.ItemID = ITEM.ItemID
AND City = 'Singapore'
ORDER BY ShipperName, DepartureDate DESC;
```

Q. Show the ShipperName, ShipmentID, and DepartureDate of all shipments that have an item that was purchased in Singapore. Use a join using JOIN ON syntax. Present results sorted by ShipperName in ascending order and then DepartureDate in descending order.

```
SELECT DISTINCT SHIPMENT.ShipperName, SHIPMENT_ITEM.ShipmentID,
SHIPMENT.DepartureDate
FROM ITEM JOIN (SHIPMENT JOIN SHIPMENT_ITEM ON SHIPMENT.ShipmentID =
SHIPMENT_ITEM.ShipmentID) ON ITEM.ItemID = SHIPMENT_ITEM.ItemID
WHERE ITEM.City='Singapore'
ORDER BY ShipperName, DepartureDate DESC;
```