Deliverable IV – Query Design

Database driver: MariaDB
Database host: 74.220.207.161
Database name: sethtimm_4407
Database username: TBT
Database password: TBT
Message timmseth@isu.edu with IP address to whitelist your IP and to be given a username and password. Alternatively a database dump will be provided.

1. List the customer name, customer number, purchase agreement number, and purchase total for each purchase. Give the purchase total a meaningful name.

```
SET @sales_tax_rate = 7.5 / 100;
SELECT customer.name AS `customer name`,
  customer.id AS `customer number`,
  purchase_agreement.id AS `purchase agreement number`,
  round(sum(piece_of_eq.sale_price) * (1 + @sales_tax_rate), 2) AS `grand total`
FROM purchase_agreement,
  customer,
  line_item,
  piece_of_eq
WHERE purchase_agreement.fk_customer_id = customer.id
  AND line_item.fk_purchase_agreement_id = purchase_agreement.id
  AND line_item.fk_piece_of_eq_id = piece_of_eq.id_serial_number
GROUP BY purchase_agreement.id;
```

   Output:

```
customer name,customer number,purchase agreement number,grand total
Dylan,3,1,102.13
Dylan,3,2,102.13
Rosetta,2,3,102.13
Rosetta,2,4,102.13
Seth,1,5,102.13
Vincent,4,9,102.13
```

2. List the model description and manufacturer for all products whose manufacturer name starts with an “A” and has a third letter “r”. (Your answer should include Aeros, Airwave, and Airush.)

```
SELECT `description`,
       manufacturer
FROM product_category
WHERE manufacturer LIKE 'A_r%';
```

Output:

```
description,manufacturer
4fight LT (long Tail) Integral Size medium,Aeros
```

3. List all salesman information, including employee#, name, sales count, and sales total (SalesAmount) and commission earned in descending order by commission earned.

```
SELECT e.id AS `employee#`,
  e.name,
  s.sales_count AS `sales count`,
  s.sales_amount AS `sales total`,
  round(s.commission_percentage / 100 * sales_amount, 2) AS `commission earned`
FROM salesperson s
       INNER JOIN employee e ON s.fk_employee_id = e.id
ORDER BY `commission earned` DESC;
```

Output:

```
employee#,name,sales count,sales total,commission earned
5,E John Doe,50,500.00,2.50
4,D John Doe,40,400.00,1.60
3,C John Doe,30,300.00,0.90
2,B John Doe,20,200.00,0.40
1,A John Doe,10,100.00,0.10
```

4. List the customer name, serial number, model number, model description, and purchase agreement date for all items having a purchase agreement date of October 6, 2010.

```
SELECT c.name               AS `customer name`,
       poe.id_serial_number AS `serial number`,
       pc.id_model_number   AS `model number`,
       pc.description       AS `model description`,
       pa.date              AS `purchase agreement date`
FROM purchase_agreement pa
       INNER JOIN customer c ON pa.fk_customer_id = c.id
       INNER JOIN line_item li ON pa.id = li.fk_purchase_agreement_id
       INNER JOIN piece_of_eq poe ON li.fk_piece_of_eq_id = poe.id_serial_number
       INNER JOIN product_category pc ON poe.fk_product_category_id = pc.id_model_number
WHERE pa.date = date('2010-10-06');
```

Output:

```
customer name,serial number,model number,model description,purchase agreement date
Vincent,5,5,ProductCatFive,2010-10-06
```

5. List all orders that were placed over 1 month ago and have not been received. Include order number, order date placed, order date received, supplier number, supplier name, supplier contact, and supplier phone. List by order date placed from oldest to most recent. INTERVAL is useful here.

```
SELECT o.id            AS `order number`,
       o.date_placed   AS `order date placed`,
       o.date_received AS `order date recieved`,
       s.id            AS `supplier number`,
       s.name          AS `supplier name`,
       s.contact_email AS `supplier contact`,
       s.phone         AS `supplier phone`
FROM orders o
       INNER JOIN supplier s ON o.fk_supplier_id = s.id
WHERE o.date_placed < now() - INTERVAL 1 MONTH
  AND o.date_received IS NULL
ORDER BY o.date_placed ASC;
```

Output:

```
order number,order date placed,order date recieved,supplier number,supplier name,supplier contact,supplie
5,2019-02-05,,5,Supplies R Other,timmseth@isu.edu,5555555555
```

6. a. For each product category list the model number, description, and the total number of sales from that category.

```
SELECT pc.id_model_number AS `model number`, pc.description, count(*) AS `total sales`
FROM product_category pc
       INNER JOIN piece_of_eq poe ON pc.id_model_number = poe.fk_product_category_id
       INNER JOIN line_item li ON poe.id_serial_number = li.fk_piece_of_eq_id
GROUP BY pc.id_model_number;
```

Output:

```
model number,description,total sales
1,ProductCatOne,1
2,ProductCatTwo,1
3,ProductCatThree,1
4,ProductCatFour,1
5,ProductCatFive,2
```

b. As a continuation from the previous query (6a), [using a nested query] list the model number, description, and the total number of sales for the product category that has the highest number of

sales. SOMEWHAT HARD

```
SELECT *
FROM (
  SELECT pc.id_model_number AS `model number`, pc.description, count(*) AS `total sales`
       FROM product_category pc
              INNER JOIN piece_of_eq poe ON pc.id_model_number = poe.fk_product_category_id
  INNER JOIN line_item li ON poe.id_serial_number = li.fk_piece_of_eq_id
  GROUP BY pc.id_model_number
  ) AS product_frequencies
ORDER BY product_frequencies.`total sales` DESC
LIMIT 1;
```

Output:

```
model number,description,total sales
5,ProductCatFive,2
```

7. List the model number, description, and number of suppliers for those product categories that were supplied by multiple suppliers. (HARD)

```
SELECT product_category.id_model_number,
  product_category.description,
  COUNT(DISTINCT supplier.id) AS suppCount
FROM product_category
  INNER JOIN line_item_eq ON product_category.id_model_number = line_item_eq.fk_model_id
  INNER JOIN orders ON line_item_eq.fk_order_id = orders.id
  INNER JOIN supplier ON orders.fk_supplier_id = supplier.id
GROUP BY product_category.id_model_number
HAVING COUNT(*) > 1;
```

Output:

```
id_model_number,description,suppCount
5,ProductCatFive,2
```

8. Given a customer name, list all credit cards associated with that customer. Include card number, card type (Visa, Mastercard, etc.), and expiration year and month, in the format yyyy/mm. (Arbitrarily select a customer name.)

```
SELECT c.name,
  cc.type,
  cc.number,
  CONCAT(cc.exp_year, '/', cc.exp_month) AS expiration
FROM customer c
       INNER JOIN purchase_agreement pa ON c.id = pa.fk_customer_id
```

```
            INNER JOIN credit_card cc ON pa.fk_cc_number = cc.number
WHERE c.name = 'Seth';
```

Output:

```
name,type,number,expiration
Seth,Discover,3456345634563456,2020/03
```

9. List all products in inventory that were purchased more than 18 months ago. Display the serial number, purchase date, supplier ID, supplier name, supplier phone number.

```
SELECT piece_of_eq.id_serial_number, orders.date_placed, supplier.id, supplier.name, supplier.phone
FROM piece_of_eq
  INNER JOIN product_category ON piece_of_eq.fk_product_category_id = product_category.id_model_number
  INNER JOIN line_item_eq ON product_category.id_model_number = line_item_eq.fk_model_id
  INNER JOIN orders ON line_item_eq.fk_order_id = orders.id
  INNER JOIN supplier ON orders.fk_supplier_id = supplier.id
WHERE orders.date_placed < (NOW() - INTERVAL 18 MONTH);
```

Output:

```
id_serial_number,date_placed,id,name,phone
2,2014-02-02,2,Supplies R We,5555555555
5,2014-02-02,2,Supplies R We,5555555555
```

10. List supplier number and supplier name for all suppliers for which there are no current orders. Sort the list in ascending order by supplier name.

```
SELECT supplier.id, supplier.name
FROM supplier
  INNER JOIN orders ON supplier.id = orders.fk_supplier_id
WHERE orders.date_received IS NOT NULL
ORDER BY supplier.name ASC;
```

Output:

```
id,name
4,Supplies R Them
1,Supplies R Us
2,Supplies R We
3,Supplies R You
```