Due: Tuesday October 29, 2013 by 11:59 pm NO LATE ASSIGNMENTS ACCEPTED

Answer each of the following questions completely. Make sure that your answers are **neatly written and very readable**. Points will be deducted if your assignment is not presented in a neat format. Submit your assignment via WebCourses.
The database shown above is to be used for all of the problems in this homework assignment.

For each of the queries shown below, write an SQL expression that will return the correct results against the database shown above. Each problem is worth 20 points.

1. Display all product information for those products that have a price greater than $50.00.
   
   ```sql
   SELECT *
   FROM lgproduct
   WHERE prod_price > 50.00;
   ```

2. Display all of the invoice numbers that were completed by the employee with id = 84078.
   
   ```sql
   SELECT inv_num
   FROM lginvoice
   WHERE employee_ID = 84078;
   ```

3. Display the current salary for each employee in department 300. Assume that only current employees are kept in the system, and therefore the most current salary for each employee is the entry in the salary history with a NULL end date. Sort the output in descending order by salary amount.
   
   ```sql
   SELECT emp_num, emp_lname, emp_fname, sal_amount
   FROM lgemployee INNER JOIN lgsalary_history
   USING (emp_num)
   WHERE sal_end IS NULL and dept_num = 300
   ORDER BY sal_amount DESC;
   ```
   
   Or -
   
   ```sql
   SELECT lgemployee.emp_num, emp_lname, emp_fname, sal_amount
   FROM lgemployee INNER JOIN lgsalary_history
   ON (lgemployee.emp_num = lgsalary_history.emp_num)
   WHERE sal_end IS NULL and dept_num = 300
   ORDER BY sal_amount DESC;
   ```
   
   Or -
   
   ```sql
   SELECT lgemployee.emp_num, emp_lname, emp_fname, sal_amount
   FROM lgemployee NATURAL JOIN lgsalary_history
   WHERE sal_end IS NULL and dept_num = 300
   ORDER BY sal_amount DESC;
   ```

4. Display the starting salary for each employee. The starting salary would be the entry in the salary history with the oldest salary start date for each employee. Sort the output by employee number.
SELECT e.emp_num, emp_lname, emp_fname, sal_amount
FROM lgemployee AS e JOIN lgsalary_history AS s
ON e.emp_num = s.emp_num
WHERE sal_from = (SELECT min(sal_from)
FROM lgsalary_history AS s2
WHERE e.emp_num = s2.emp_num)
ORDER BY e.emp_num;

Or -

SELECT e.emp_num, emp_lname, emp_fname, sal_amount
FROM lgemployee as e NATURAL JOIN lgsalary_history
WHERE sal_from = (SELECT min(sal_from)
FROM lgsalary_history as s2
WHERE e.emp_num = s2.emp_num)
ORDER BY e.emp_num;

5. Display the invoice number, line numbers, product SKUs, product descriptions, and brand ID for sales of sealer and top coat products (sealer and top coat are product categories) of the same brand on the same invoice.

SELECT l.inv_num, l.line_num, p.prod_sku, p.prod_descript, l2.line_num, p2.prod_sku, p2.prod_descript, p.brand_id
FROM (lgline AS l INNER JOIN lgproduct AS p
ON l.prod_sku = p.prod_sku)
INNER JOIN (lgline AS l2 INNER JOIN lgproduct AS p2
ON l2.prod_sku = p2.prod_sku)
ON l.inv_num = l2.inv_num
WHERE p.brand_id = p2.brand_id
AND p.prod_category = 'Sealer'
AND p2.prod_category = 'Top Coat'
ORDER BY l.inv_num, l.line_num;

NOTE THAT THE FOLLOWING QUERY IS INCORRECT!!

SELECT l.inv_num, l.line_num, p.prod_sku, p.prod_descript, l2.line_num, p2.prod_sku, p2.prod_descript, p.brand_id
FROM (lgline AS l NATURAL JOIN lgproduct AS p)  NATURAL JOIN (lgline AS l2 NATURAL JOIN lgproduct AS p2)
WHERE p.brand_id = p2.brand_id
AND p.prod_category = 'Sealer'
AND p2.prod_category = 'Top Coat'
ORDER BY l.inv_num, l.line_num;
6. The Binder Prime Company wants to recognize the employee who sold the most of their products during a specified period. Write a query to display the employee number, employee first name, employee last name, e-mail address, and total units sold for the employee who sold the most Binder Prime brand products between November 1, 2011, and December 5, 2011. If there is a tie for most units sold, sort the output by employee last name. (This is a complex query.)

```sql
SELECT emp.emp_num, emp_lname, emp_fname, emp_email, total
FROM lgemployee AS emp
INNER JOIN
  (SELECT employee_id, sum(line_qty) AS total
   FROM lginvoice AS i
   INNER JOIN lgline AS l ON i.inv_num = l.inv_num
   INNER JOIN lgproduct AS p ON l.prod_sku = p.prod_sku
   INNER JOIN lgbrand AS b ON b.brand_id = p.brand_id
   WHERE brand_name = 'Binder Prime'
   AND inv_date BETWEEN '2011-11-01' AND '2011-12-06'
   GROUP BY employee_id) AS sub
ON emp.emp_num = sub.employee_id
WHERE total = (SELECT max(total)
FROM (SELECT employee_id, sum(line_qty) AS total
FROM lginvoice AS i
INNER JOIN lgline AS l ON i.inv_num = l.inv_num
INNER JOIN lgproduct AS p ON l.prod_sku = p.prod_sku
INNER JOIN lgbrand AS b ON b.brand_id = p.brand_id
WHERE brand_name = 'Binder Prime'
AND inv_date BETWEEN '2011-11-01' AND '2011-12-06'
GROUP BY employee_id) as sub1);
```

7. Display the customer code, first name, and last name of all customers who have had at least one invoice completed by employee 83649 and at least one invoice completed by employee 83677. Sort the output by customer last name and then first name.

**General SQL Format**

```sql
SELECT cust_code, cust_fname, cust_lname
FROM lgcustomer NATURAL JOIN lginvoice
WHERE employee_id = 83649
INTERSECT
SELECT cust_code, cust_fname, cust_lname
FROM lgcustomer NATURAL JOIN lginvoice
WHERE employee_id = 83677
ORDER BY cust_lname, cust_fname;
```

**MySQL Format**

```sql
SELECT cust_code, cust_fname, cust_lname
FROM lgcustomer
WHERE cust_code IN
  (SELECT cust_code
   FROM lgcustomer INNER JOIN lginvoice using (cust_code)
   WHERE employee_id = 83649)
AND
  cust_code IN
  (SELECT cust_code
   FROM lgcustomer INNER JOIN lginvoice using (cust_code)
   WHERE employee_id = 83677);
```
8. One of the purchasing managers is interested in the impact of product prices on the sale of products of each brand. Write a query to display the brand name, brand type, average price of products of each brand, and total units sold of products of each brand. Even if a product has been sold more than once, its price should only be included once in the calculation of the average price. However, you must be careful because multiple products of the same brand can have the same price, and each of those products must be included in the calculation of the brand’s average price.

SELECT brand_name, brand_type,
round(avgprice,2) AS average_price, units_sold
FROM (lgbrand as b
  INNER JOIN
  (SELECT brand_id, avg(prod_price) as avgprice
   FROM lgproduct
   GROUP BY brand_id) as sub1
  ON b.brand_id = sub1.brand_id)
  INNER JOIN
  (SELECT brand_id, sum(line_qty) AS units_sold
   FROM lgproduct as p NATURAL JOIN lgline
   GROUP BY brand_id) as sub2
  ON b.brand_id = sub2.brand_id
ORDER BY brand_name;

Or -

SELECT brand_name, brand_type,
round(avgprice,2) AS average_price, units_sold
FROM (lgbrand as b
  INNER JOIN
  (SELECT brand_id, avg(prod_price) as avgprice
   FROM lgproduct
   GROUP BY brand_id) as sub1
  USING (brand_id)
  INNER JOIN
  (SELECT brand_id, sum(line_qty) AS units_sold
   FROM lgproduct as p NATURAL JOIN lgline
   GROUP BY brand_id) as sub2
  USING (brand_id))
ORDER BY brand_name;

9. The purchasing manager is still concerned about the impact of price on sales. Write a query to display the brand name, brand type, product SKU, product description, and price of any products that are not a premium brand (brand type), but that cost more than the most expensive premium brand products.
SELECT brand_name, brand_type, prod_sku, prod_descript, prod_price
FROM lgproduct NATURAL JOIN lgbrand
WHERE brand_type <> 'premium'
    AND prod_price > (SELECT max(prod_price)
                       FROM lgproduct NATURAL JOIN lgbrand
                       WHERE brand_type = 'premium');

Or -

SELECT brand_name, brand_type, prod_sku, prod_descript, prod_price
FROM lgproduct INNER JOIN lgbrand USING (brand_id)
WHERE brand_type <> 'premium'
    AND prod_price > (SELECT max(prod_price)
                       FROM lgproduct INNER JOIN lgbrand USING (brand_id)
                       WHERE brand_type = 'premium');

Or -

SELECT brand_name, brand_type, prod_sku, prod_descript, prod_price
FROM lgproduct INNER JOIN lgbrand
    ON (lgproduct.brand_id = lgbrand.brand_id)
WHERE brand_type <> 'premium'
    AND prod_price > (SELECT max(prod_price)
                       FROM lgproduct INNER JOIN lgbrand
                       ON (lgproduct.brand_id = lgbrand.brand_id)
                       WHERE brand_type = 'premium');