**Name(s)\_Ryder Dale Walton && Chance Turner\_**

**Homework 10**

**Due Tue, Nov 22**

**(25 pts)**

**You can do this in teams of 2 people**

Read the final paragraph at the bottom of page 420, and complete the following exercises at the end of chapter 8: Problems 19 – 27 (you can omit problem 22; it is done for you below). Copy/paste the working SQL below; clearly label each problem.

* All of your answers should consist of *one* SQL statement. That is, no fair creating new permanent tables with intermediate results, and then writing SQL to query those tables.
* Not every question is directly related to this chapter. Some of it is more basic and may require you to review chapter 7.
* A script to create the database is provided on bboard.
* NOTE: Many of your results will show more than what the book’s answer shows. The book is only showing the first part of the results.

**Turnin:** Submit this Word file to the bboard link.

**Answers**

**Label each problem clearly, copy/paste your sql below after the Hint for each problem.**

Problem 19:

SELECT \* FROM LGPRODUCT WHERE PROD\_PRICE > 50;

Problem 20:

* Hint: When you need to check to see if a field contains NULL, you have to say …**where blah IS NULL**. Do not use **where blah = NULL**. This is because technically NULL is not considered a value.

SELECT E.EMP\_NUM, EMP\_LNAME, EMP\_FNAME, SAL\_AMOUNT

FROM LGEMPLOYEE E JOIN lgsalary\_history SH

ON E.emp\_num = SH.emp\_num

WHERE E.dept\_num = 300 AND sal\_end IS NULL

ORDER BY sal\_amount DESC;

Problem 21:

* Hint: We will discuss in class

SELECT E.EMP\_NUM, EMP\_LNAME, EMP\_FNAME, SAL\_AMOUNT

FROM LGEMPLOYEE E JOIN lgsalary\_history SH

ON E.emp\_num = SH.emp\_num

WHERE SH.sal\_from = (SELECT MIN(sal\_from) –correlated subquery b/c of aliases

FROM lgsalary\_history SH2

group by sh2.emp\_num

having sh.emp\_num = sh2.emp\_num) –because I am using group by, need having!

group by e.emp\_num, emp\_lname, emp\_fname, sal\_amount

ORDER BY E.EMP\_NUM ASC;

Problem 22:

Hint: See a solution below (that’s right, this one is free). We will discuss this in class.

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 l join lgproduct p ON l.prod\_sku = p.prod\_sku) join

(lgline l2 join lgproduct 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;

Problem 23:

* Use the following dates instead of what is given in the book: November 1, 2013 and December 5, 2013.
* We will discuss in class.

SELECT E.emp\_num, E.emp\_fname, E.emp\_lname, E.emp\_email, Total

FROM lgemployee E JOIN

(SELECT I.employee\_id, SUM(L.line\_qty) AS Total

FROM lginvoice I join lgline L ON I.inv\_num = L.inv\_num

JOIN lgproduct P ON L.prod\_sku = P.prod\_sku

JOIN lgbrand B ON B.brand\_id = P.brand\_id

WHERE B.brand\_name = 'BINDER PRIME'

AND I.inv\_date BETWEEN '2013-11-1' AND '2013-12-5'

GROUP BY I.employee\_id) as table1

ON E.emp\_num = table1.employee\_id

WHERE Total = (SELECT MAX(Total)

FROM (SELECT I.employee\_id, SUM(L.line\_qty) AS Total

FROM lginvoice I join lgline L ON I.inv\_num = L.inv\_num

JOIN lgproduct P ON L.prod\_sku = P.prod\_sku

JOIN lgbrand B ON B.brand\_id = P.brand\_id

WHERE B.brand\_name = 'BINDER PRIME'

AND I.inv\_date BETWEEN '2013-11-1' AND '2013-12-5'

GROUP BY I.employee\_id) as table2)

ORDER BY E.emp\_lname

;

--This one had me utterly stumped. I worked on it for several hours before researching online for help. The website below showed me how I could create a virtual column by a name that I wanted to give it without having to do a SELECT subquery. It also showed me how to use a FROM subquery and how to name that virtual table. I struggled with that concept during this homework assignment quite a bit.

Note: The solution to this problem from the website is not actually correct. I just used it to get me unstuck as I was struggling with how to correctly join the tables here.

<https://www.coursehero.com/file/p1jp3fj4/Figure-P823-Employees-with-most-Binder-Prime-units-sold-46-Chapter-8-Advanced/>

Problem 24:

Hint: Consider Intersect

SELECT C.cust\_code, C.cust\_fname, C.cust\_lname

FROM lgcustomer C, lginvoice I, lgemployee E

WHERE C.cust\_code = I.cust\_code

AND I.employee\_id = E.emp\_num

AND I.employee\_id = 83649

INTERSECT

SELECT C.cust\_code, C.cust\_fname, C.cust\_lname

FROM lgcustomer C, lginvoice I, lgemployee E

WHERE C.cust\_code = I.cust\_code

AND I.employee\_id = E.emp\_num

AND I.employee\_id = 83677

ORDER BY C.cust\_lname, C.cust\_fname;

Problem 25:

Hint: Consider Union

SELECT C2.cust\_code, C2.cust\_fname, C2.cust\_lname, C2.cust\_street, C2.cust\_city,

C2.cust\_state, C2.cust\_zip, I2.inv\_date, I2.inv\_total AS Largest\_Invoice

FROM lgcustomer C2, lginvoice I2

WHERE C2.cust\_state = 'AL'

AND C2.cust\_code = I2.cust\_code

AND i2.inv\_total = (SELECT MAX(I3.INV\_TOTAL)

FROM lginvoice I3, lgcustomer C3

WHERE I3.cust\_code = C3.cust\_code

GROUP BY C3.cust\_code

HAVING C3.cust\_code = C2.cust\_code)

UNION

SELECT C4.cust\_code, C4.cust\_fname, C4.cust\_lname, C4.cust\_street, C4.cust\_city,

C4.cust\_state, C4.cust\_zip, I4.inv\_date, C4.cust\_balance AS Largest\_Invoice

FROM lgcustomer C4 left join lginvoice I4 on c4.cust\_code = i4.cust\_code

WHERE C4.cust\_state = 'AL'

AND c4.cust\_balance = 0.00

AND I4.inv\_date IS NULL

ORDER BY C2.cust\_lname;

;

Problem 26:

Hint: Get the average price in one FROM subquery; join that with another FROM subquery to get the units sold information.

SELECT DISTINCT B.brand\_name, B.brand\_type, --I thought of a different way to do this one

(SELECT AVG(prod\_price)

FROM lgbrand B2 JOIN lgproduct P2 ON B2.brand\_id = P2.brand\_id

GROUP BY B2.brand\_name

HAVING B2.brand\_name = B.brand\_name) AS Average\_Price,

(SELECT SUM(L.LINE\_QTY)

FROM lgline L JOIN lgproduct P3 ON L.prod\_sku = P3.prod\_sku

GROUP BY P3.BRAND\_ID

HAVING P3.BRAND\_ID = B.brand\_id) AS Units\_Sold

FROM lgbrand B JOIN lgproduct P ON B.brand\_id = P.brand\_id

ORDER BY B.brand\_name;

--Not sure if it is better or not, but it works and it was how I naturally thought of it.

Problem 27:

Hint: The question should be worded “…that cost more than the most expensive premium brand *product*” (product should be singular).

SELECT B.brand\_name, B.brand\_type, P.prod\_sku, P.prod\_descript, P.prod\_price

FROM lgbrand B JOIN lgproduct P ON B.brand\_id = P.brand\_id

WHERE P.prod\_price = (SELECT MAX(P2.prod\_price)

FROM lgbrand B2 JOIN lgproduct P2 ON B.brand\_id = P.brand\_id

WHERE B2.brand\_type != 'PREMIUM');

--Yields the desired result in the book