**Sample Solutions for Homework 10**

These are the book solutions; they are certainly not the only way to do these queries.

19

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

20

SELECT e.emp\_num, emp\_lname, emp\_fname, sal\_amount

FROM lgemployee e JOIN lgsalary\_history s ON e.emp\_num = s.emp\_num

WHERE sal\_end IS NULL

AND dept\_num = 300

ORDER BY sal\_amount DESC;

21

Could you do this for one employee? Join the tables, show the sal\_amount for the min sal\_from for that employee:

select e.emp\_num, emp\_lname, sal\_amount from lgemployee e join lgsalary\_history h on

e.emp\_num = h.emp\_num

where e.emp\_num = 83653 AND sal\_from = (select min(sal\_from) from lgsalary\_history where emp\_num = 83653)

We want to do it for all employees, so just do a correlated subquery.

SELECT e.emp\_num, emp\_lname, emp\_fname, sal\_amount

FROM lgemployee e join lgsalary\_history s ON e.emp\_num = s.emp\_num

WHERE sal\_from = (SELECT Min(sal\_from)

FROM lgsalary\_history s2 WHERE e.emp\_num = s2.emp\_num)

ORDER BY e.emp\_num;

22

They get this by creating two copies of the line items data, and then joining those two copies where the brand id is the same.

GIVE THIS ONE TO THEM.

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;

-- Notice the similarity b/w this and a recursive join. You will do this whenever you have results that depend on two rows in the same table or joined tables.

23

SELECT emp.emp\_num, emp\_fname, emp\_lname, emp\_email, total

FROM lgemployee emp JOIN

(SELECT employee\_id, Sum(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 brand\_name = 'BINDER PRIME'

AND INV\_DATE BETWEEN '01-NOV-13' AND '06-DEC-13'

GROUP BY employee\_id) sub

ON emp.emp\_num = sub.employee\_id

where total = (Select max(numSold) from (SELECT Sum(line\_qty) as NumSold

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 brand\_name = 'BINDER PRIME'

AND INV\_DATE BETWEEN '01-NOV-13' AND '05-DEC-13'

GROUP BY employee\_id) table1)

--What do you notice about using a FROM subquery? You will use it when you want the result of an aggregate function. So, you have to make a subquery and join with it so that you can get the Total column in the select statement.

--The nested subqueries in the WHERE requires you to get all of the totals again, and then select the max from that.

NOTE: when you are making a from subquery, the subquery “table” must always be given an alias name (like table1 in the above)

24

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

FROM lgcustomer c JOIN lginvoice i ON c.cust\_code = i.cust\_code

WHERE employee\_id = 83649

INTERSECT

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

FROM lgcustomer c JOIN lginvoice i ON c.cust\_code = i.cust\_code

WHERE employee\_id = 83677

ORDER BY cust\_lname, cust\_fname;

25

SELECT c.cust\_code, cust\_fname, cust\_lname, cust\_street, cust\_city, cust\_state, cust\_zip,

inv\_date, inv\_total AS "Largest Invoice"

FROM lgcustomer c JOIN lginvoice i ON c.cust\_code = i.cust\_code

WHERE cust\_state = 'AL'

AND inv\_total = (SELECT Max(inv\_total)

FROM lginvoice i2

WHERE i2.cust\_code = c.cust\_code)

UNION

SELECT cust\_code, cust\_fname, cust\_lname, cust\_street, cust\_city, cust\_state, cust\_zip, NULL, 0 –you can hard code these values that are wanted to show up

FROM lgcustomer

WHERE cust\_state = 'AL'

AND cust\_code NOT IN (SELECT cust\_code FROM lginvoice)

ORDER BY cust\_lname, cust\_fname;

26

Note: They generate a table with average price, they generate a table with units sold, then join that together with original table data.

SELECT brand\_name, brand\_type, Round(avgprice,2) AS "Average Price", "Units Sold"

FROM lgbrand b JOIN (SELECT brand\_id, Avg(prod\_price) AS avgprice

FROM lgproduct

GROUP BY brand\_id) sub1

ON b.brand\_id = sub1.brand\_id

JOIN (SELECT brand\_id, Sum(line\_qty) AS "Units Sold"

FROM lgproduct p JOIN lgline l ON p.prod\_sku = l.prod\_sku

GROUP BY brand\_id) sub2

ON b.brand\_id = sub2.brand\_id

ORDER BY brand\_name;

27

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

SELECT brand\_name, brand\_type, prod\_sku, prod\_descript, prod\_price

FROM lgproduct p JOIN lgbrand b ON p.brand\_id = b.brand\_id

WHERE brand\_type <> 'PREMIUM'

AND prod\_price > (SELECT Max(prod\_price)

FROM lgproduct p JOIN lgbrand b ON p.brand\_id = b.brand\_id

WHERE brand\_type = 'PREMIUM');