SQL Programming for Database

Problem: Database Management for Large Service Firm

Solution: Basic SQL Functions and Extrapolation for Existing Database

#Ordering by Department Name

SELECT \* FROM lgdepartment

ORDER BY dept\_name;

#Listing attributes for Product with certain specifications

SELECT prod\_sku, prod\_descript, prod\_type, prod\_category, prod\_price

FROM lgproduct

WHERE prod\_base='Water'

AND prod\_category='sealer';

# Listing attributes of employee with certain specifications with order

SELECT emp\_fname, emp\_lname, emp\_email

FROM lgemployee

WHERE emp\_hiredate>='2005/01/01'

AND emp\_hiredate<='2014/12/31'

ORDER BY emp\_lname, emp\_fname;

# Listing attributes of employee with certain specifications or other with order

SELECT emp\_fname, emp\_lname, emp\_phone, emp\_title, dept\_num

FROM lgemployee

WHERE dept\_num=300

OR emp\_title='CLERK I'

ORDER BY emp\_lname, emp\_fname;

# Listing attributes of employee with certain specifications and others with order

SELECT lgsalary\_history.emp\_num, emp\_lname, emp\_fname, sal\_from, sal\_end, sal\_amount

FROM lgsalary\_history, lgemployee

WHERE lgemployee.emp\_num in (83731, 83745, 84039)

AND lgsalary\_history.emp\_num=lgemployee.emp\_num

ORDER BY lgsalary\_history.emp\_num, sal\_from;

#Joining fields with conditions

SELECT DISTINCT cust\_fname, cust\_lname, cust\_street, cust\_city, cust\_state, cust\_zip

FROM lgcustomer

inner join lginvoice INVOICE on lgcustomer.cust\_code=invoice.cust\_code

inner join lgline on invoice.inv\_num=lgline.inv\_num

inner join lgproduct on lgproduct.prod\_sku=lgline.prod\_sku

inner join lgbrand on lgbrand.brand\_id=lgproduct.brand\_id

AND inv\_date

BETWEEN '2015-07-05'

AND '2015-07-30'

AND brand\_name='FORESTERS BEST'

AND prod\_category='top coat'

ORDER BY cust\_state, cust\_lname, cust\_fname;

#Joining fields with conditions and order

SELECT lgemployee.emp\_num, emp\_lname, emp\_email, emp\_title, dept\_name

FROM lgemployee

inner join lgdepartment

ON lgdepartment.dept\_num=lgemployee.dept\_num

WHERE emp\_title LIKE'%ASSOCIATE'

ORDER BY dept\_name, emp\_title;

#Grouping post joining with order

SELECT brand\_name,

COUNT(lgbrand.brand\_id)

AS NUMPRODUCTS

FROM lgbrand

inner join lgproduct

ON lgbrand.brand\_id=lgproduct.brand\_id

GROUP BY lgbrand.brand\_name

ORDER BY brand\_name;

#Grouping post join with aggregate

SELECT prod\_category,

COUNT(prod\_base)

AS NUMPRODUCTS

FROM lgproduct

WHERE prod\_base='water'

GROUP BY prod\_category;

#Grouping post join with count and order

SELECT prod\_base, prod\_type,

COUNT(prod\_type)

AS NUMPRODUCTS

FROM lgproduct

GROUP BY prod\_base, prod\_type

ORDER BY prod\_base;

#Summation

SELECT brand\_id,

SUM(brand\_id)

AS TOTALINVENTORY

FROM lgproduct

GROUP BY brand\_id

ORDER BY brand\_id DESC;

#Rounding

SELECT lgbrand.brand\_id, brand\_name,

ROUND(AVG(prod\_price), 2)

AS AVGPRICE

FROM lgbrand

inner join lgproduct

ON lgbrand.brand\_id=lgproduct.brand\_id

GROUP BY lgbrand.brand\_id, lgbrand.brand\_name

ORDER BY brand\_name;

#Max

SELECT lgdepartment.dept\_num,

MAX(emp\_hiredate)

AS MOSTRECENT

FROM lgdepartment

inner join lgemployee ON lgdepartment.dept\_num = lgemployee.dept\_num

GROUP BY lgdepartment.dept\_num

ORDER BY lgdepartment.dept\_num;

#Max with grouping

SELECT lgemployee.emp\_num, emp\_fname, emp\_lname,

MAX(sal\_amount)

AS LARGESTSALARY

FROM lgemployee

inner join lgsalary\_history

ON lgemployee.emp\_num = lgsalary\_history.emp\_num

WHERE lgemployee.dept\_num = 200

GROUP BY lgemployee.emp\_num, emp\_fname, emp\_lname

ORDER BY LARGESTSALARY DESC

#Summing with grouping and condition

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

SUM(li.inv\_total)

AS INVOICETOTAL

FROM lgcustomer lc , lginvoice li

WHERE lc.cust\_code=li.cust\_code

GROUP BY lc.cust\_lname, lc.cust\_fname, lc.cust\_code

HAVING SUM(li.inv\_total) > 1500

ORDER BY INVOICETOTAL DESC;

#joining across fields

SELECT lgdepartment.dept\_num, dept\_name, dept\_phone, lgemployee.emp\_num, emp\_lname

FROM lgdepartment

inner join lgemployee

ON lgdepartment.emp\_num=lgemployee.emp\_num

ORDER BY dept\_name;

#Multiple joins

SELECT v.vend\_id, v.vend\_name, b.brand\_name,

COUNT(p.prod\_sku)

AS NUM\_PRODUCTS

FROM lgvendor v

inner join lgsupplies s

ON v.vend\_id = s.vend\_id

inner join lgproduct p

ON s.prod\_sku = p.prod\_sku

inner join lgbrand b

ON b.brand\_id = p.brand\_id

GROUP BY v.vend\_id, v.vend\_name, b.brand\_name

ORDER BY v.vend\_name, b.brand\_name;

#Aggregating with grouping post join with order

SELECT lgemployee.emp\_num, emp\_lname, emp\_fname,

SUM(inv\_total)

AS TOTALINVOICE

FROM lgemployee

inner join lginvoice

ON lgemployee.emp\_num = lginvoice.employee\_id

GROUP BY lgemployee.emp\_num, emp\_lname, emp\_fname

ORDER BY emp\_fname, emp\_lname;

#Top

SELECT ROUND(AVG(prod\_price), 2)

AS 'LARGEST AVERAGE'

FROM lgproduct

GROUP BY brand\_id

HAVING AVG(prod\_price) =

(SELECT

TOP 1 AVG(prod\_price)

FROM lgproduct

GROUP BY brand\_id

ORDER BY AVG(prod\_price) DESC);

#Averaging

SELECT b.brand\_id, brand\_name, brand\_type,

ROUND(AVG(prod\_price),2)

AS 'AVGPRICE'

FROM lgbrand b

inner join lgproduct p

ON b.brand\_id = p.brand\_id

GROUP BY b.brand\_id, b.brand\_name, b.brand\_type

HAVING

AVG(prod\_price) =

(SELECT

TOP 1 AVG(prod\_price)

FROM lgbrand b INNER JOIN lgproduct p

ON b.brand\_id = p.brand\_id

GROUP BY b.brand\_id, brand\_name, brand\_type

ORDER BY AVG(prod\_price) DESC);

#Listing multiple subsets

SELECT

m.emp\_fname

AS 'MANAGER LNAME', m.emp\_lname

AS 'MANAGER FNAME', d.dept\_name, d.dept\_phone, e.emp\_lname, e.emp\_fname, c.cust\_lname, c.cust\_fname, i.inv\_date, i.inv\_total

FROM lgdepartment d

inner join lgemployee e

ON d.dept\_num = e.dept\_num

INNER JOIN lgemployee m

ON d.emp\_num = m.emp\_num

INNER JOIN lginvoice i

ON i.employee\_id = e.emp\_num

INNER JOIN lgcustomer c

ON c.cust\_code = i.cust\_code

WHERE c.cust\_lname = 'Hagan' AND i.inv\_date = '2015-05-18';

#Multiple lists with various joins, grouping, ordering and aggregations

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

FROM lgemployee e

inner join lgsalary\_history s

ON s.emp\_num = e.emp\_num

WHERE

s.sal\_end IS NULL

AND e.dept\_num = '300'

ORDER BY s.sal\_amount DESC;

#Lists with various joins, grouping, conditions, order and aggregations

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

FROM lgemployee e

inner join lgsalary\_history s

ON e.emp\_num = s.emp\_num

WHERE sal\_from =

(SELECT MIN(sal\_from)

FROM lgemployee e

inner join lgsalary\_history s

ON s.emp\_num = e.emp\_num)

ORDER BY e.emp\_num;

#Multiple lists with joins, grouping, conditions, ordering and aggregations

SELECT i.inv\_num, i.line\_num, i.prod\_sku, i.prod\_descript, p.line\_num, p.prod\_sku, p.prod\_descript, i.brand\_id

FROM (

SELECT l.inv\_num, l.line\_num, lp.prod\_sku, lp.prod\_descript, lp.brand\_id, lp.prod\_category

FROM lgline l, lgproduct lp

WHERE l.prod\_sku = lp.prod\_sku

AND lp.prod\_category = 'Sealer'

) i, (

SELECT l2.line\_num, lp2.prod\_sku, lp2.prod\_descript, lp2.brand\_id, l2.inv\_num, lp2.prod\_category

FROM lgline l2, lgproduct lp2

WHERE l2.prod\_sku =lp2.prod\_sku

AND lp2.prod\_category = 'Top Coat'

) p

WHERE i.inv\_num = p.inv\_num

AND i.brand\_id = p.brand\_id

ORDER BY i.inv\_num, i.line\_num, p.line\_num DESC;

#Multiple lists with various joins, grouping, conditions, ordering and aggregations

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

SUM(l.line\_qty)

AS 'TOTAL\_UNITS'

FROM lgemployee e

inner join lginvoice i

ON e.emp\_num = i.employee\_id

inner join lgline l ON l.inv\_num = i.inv\_num

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

inner join lgbrand b ON p.brand\_id = b.brand\_id

WHERE b.brand\_name = 'BINDER PRIME'

AND i.inv\_date

BETWEEN '11-01-15'

AND '12-05-15'

GROUP BY e.emp\_num, emp\_fname, emp\_lname, emp\_email

HAVING SUM(l.line\_qty) =

(SELECT MAX(TOTAL\_UNITS)

FROM(SELECT e.emp\_num, emp\_fname, emp\_lname, emp\_email, SUM(l.line\_qty)

AS 'TOTAL\_UNITS'

FROM lgemployee e

inner join lginvoice i

ON e.emp\_num = i.employee\_id

inner join lgline l

ON l.inv\_num = i.inv\_num

inner join lgproduct p

ON p.prod\_sku = l.prod\_sku

inner join lgbrand b

ON p.brand\_id = b.brand\_id

WHERE b.brand\_name = 'BINDER PRIME'

AND i.inv\_date

BETWEEN '11-01-15' AND '12-05-15'

GROUP BY e.emp\_num, emp\_fname, emp\_lname, emp\_email) A)

ORDER BY emp\_lname

#Unqiue

SELECT DISTINCT lgcustomer.cust\_code, cust\_fname, cust\_lname

FROM lgcustomer

inner join lginvoice i

ON i.cust\_code = lgcustomer.cust\_code

inner join lginvoice v

ON v.cust\_code = lgcustomer.cust\_code

WHERE v.employee\_id = '83677'

AND i.employee\_id = '83649'

ORDER BY cust\_lname, cust\_fname;

#Multiple lists with various joins, grouping, conditions and aggregations

SELECT

C.cust\_code, C.cust\_fname, C.cust\_lname, C.cust\_street, C.cust\_city, C.cust\_state, C.cust\_zip, I.inv\_date, I.inv\_total

AS "LARGEST INVOICE"

FROM lgcustomer C INNER JOIN lginvoice I ON I.cust\_code = C.cust\_code

WHERE C.cust\_state = 'AL'

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

FROM lginvoice IV

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

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 c.cust\_state = 'AL'

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

FROM lginvoice IV WHERE IV.cust\_code = c.cust\_code)

UNION

SELECT cust\_code, cust\_fname, cust\_lname, cust\_street, cust\_city, cust\_state, cust\_zip, NULL, 0

FROM lgcustomer

WHERE cust\_state = 'AL'

AND cust\_code

NOT IN (SELECT cust\_code FROM lginvoice)