DB – Project 1

Prateek Agarwal (prat0318@gmail.com)

```
-- 1. What is the price of the part named "Dirty Harry"?
SELECT price
FROM   parts
WHERE  pname = 'Dirty Harry';

--   PRICE
----------
--    14

--  2.   What orders have been shipped after date '03-feb-95'?
SELECT orders.ono
FROM   orders
WHERE  To_date(shipped, 'DD-MON-YY') > To_date('03-FEB-95', 'DD-MON-YY');

--     ONO
----------
--    1022
--    1023

-- 3.   What are the ono and cname values of customers whose orders have not been
shipped (i.e., the shipped column has a null value)?
SELECT orders.ono,
       customers.cname
FROM   customers,
       orders
WHERE  customers.cno = orders.cno
       AND orders.shipped IS NULL;

-- no results found

--  4.   Retrieve the names of parts whose quantity on hand (QOH) is between 20
and 70.
SELECT pname
FROM   parts
WHERE  qoh BETWEEN 20 AND 70;

-- PNAME
--------------------------------------------------------------------------------
--Land Before Time IV


--  5.   Get all unique pairs of cno values for customers that have the same zip
code.
SELECT c1.cno,
       c2.cno
FROM   customers c1,
       customers c2
WHERE  c1.zip = c2.zip
       AND c1.cno < c2.cno;

--     CNO    CNO
---------- ----------
--    1111   2222
```

```
--  6.   Create a nested SQL select statement that returns the cname values of
customers who have placed orders with employees living in Fort Dodge.
SELECT c.cname
FROM   customers c
WHERE  'Fort Dodge' = ALL (SELECT z.city
                           FROM   employees e,
                                  zipcodes z,
                                  orders o
                           WHERE  c.cno = o.cno
                                  AND o.eno = e.eno
                                  AND e.zip = z.zip);

--CNAME
--------------------------------------------------------------------------------
--Bertram

--  7.   What orders have been shipped to Wichita?
SELECT o.ono
FROM   orders o,
       customers c,
       zipcodes z
WHERE  o.cno = c.cno
       AND c.zip = z.zip
       AND z.city = 'Wichita';

--     ONO
----------
--    1021
--    1020
--    1022

--  8.   Get the pname values of parts with the lowest price.
SELECT pname
FROM   (SELECT *
        FROM   parts
        ORDER  BY price ASC)
WHERE  ROWNUM <= 1;

--PNAME
--------------------------------------------------------------------------------
--Dirty Harry

--  9.   What is the name of the part with the lowest price? (use qualified
comparison in your predicate, i.e., <=all).
SELECT pname
FROM   parts
WHERE  price <= ALL (SELECT price
                     FROM   parts);

--PNAME
--------------------------------------------------------------------------------
--Dirty Harry



--  10.   What parts cost more than the most expensive Land Before Time part?
(Hint: you should use pattern-matching, e.g., pname like 'Land Before Time%').
```

```
SELECT pno
FROM   parts
WHERE  price > ALL (SELECT price
                    FROM   parts
                    WHERE  pname LIKE 'Land Before Time%');
--     PNO
----------
--   10601
--   10900

--  11.   Write a correlated query to return the cities of zipcodes from which an
order has been placed.
SELECT city
FROM   zipcodes z
WHERE  (SELECT Count(*)
        FROM   customers c,
               orders o
        WHERE  c.cno = o.cno
               AND c.zip = z.zip) > 0;
--CITY
--------------------------------------------------------------------------------
--Wichita
--Fort Dodge

--  12.   Get cname values of customers who have placed at least one part order
through employee with eno = 1000.
SELECT DISTINCT c.cname
FROM   customers c,
       orders o,
       employees e
WHERE  c.cno = o.cno
       AND o.eno = e.eno
       AND e.eno = 1000;

--CNAME
--------------------------------------------------------------------------------
--Charles
--Barbara

--  13.   Get the total number of customers.
SELECT Count(*)
FROM   customers;

--COUNT(*)
----------
--     3

--  14.   Get the pname values of parts that cost more than the average cost of
all parts.
SELECT pname
FROM   parts
WHERE  price > (SELECT Avg(price)
                FROM   parts);

--PNAME
--------------------------------------------------------------------------------
--Sleeping Beauty
--Dr. Zhivago
```

```
--  15.   For each part, get pno and pname values along with the total sales in
dollars.
SELECT pno,
       pname,
       ( (SELECT SUM(qty)
          FROM   odetails
          WHERE  pno = parts.pno) * price ) SALES
FROM   parts;
```

PNO	PNAME	SALES
10506	Land Before Time I	19
10507	Land Before Time II	19
10508	Land Before Time III	38
10509	Land Before Time IV	57
10601	Sleeping Beauty	120
10701	When Harry Met Sally	19
10800	Dirty Harry	14
10900	Dr. Zhivago	24

```
--  16.   For each part, get pno and pname values along with the total sales in
dollars, but only for total sales exceeding $200.
SELECT pno,
       pname,
       ( (SELECT SUM(qty)
          FROM   odetails
          WHERE  pno = parts.pno) * price ) SALES
FROM   parts
WHERE  ( (SELECT SUM(qty)
          FROM   odetails
          WHERE  pno = parts.pno) * price ) > 200;

--no rows selected

--  17.   Repeat the last 2 queries, except this time create a view to simplify
your work. Define the view and each query on that view.
CREATE VIEW sales
AS
  SELECT pno,
         SUM(qty) TOTAL
  FROM   odetails
  GROUP  BY pno;

--  17.1
SELECT p.pno,
       pname,
       ( total * price ) SALES
FROM   parts p,
       sales s
WHERE  p.pno = s.pno;

--  17.2
SELECT p.pno,
       pname,
```

```
       ( total * price ) SALES
FROM   parts p,
       sales s
WHERE  p.pno = s.pno
       AND total * price > 200;

--  18.   Delete order 1021 and its order details.
DELETE FROM odetails
WHERE  ono = 1021;

DELETE FROM orders
WHERE  ono = 1021;

--  19.   Increase the cost of all parts by 5%.
UPDATE parts
SET    price = ( price * 1.05 );

--  20.   Retrieve employees by name in reverse alphabetical order.
SELECT ename
FROM   employees
ORDER  BY ename DESC;

--ENAME
--------------------------------------------------------------------------------
--Smith
--Jones
--Brown1
--Brown

--  21.   What tuples of Employees and Zipcodes do not participate in a join of
these relations? Use the outerjoin and minus operations.
SELECT z.zip,
       e.eno
FROM   employees e
       full join zipcodes z
              ON e.zip = z.zip
MINUS
SELECT z.zip,
       e.eno
FROM   employees e
       join zipcodes z
         ON e.zip = z.zip;


--     ZIP    ENO
---------- ----------
--   54444
--   61111
--   66002
```