Assignment Report

DATABASE FUNDAMENTALS

Jason Max Chalom 711985

GROUP MEMBERS:
JONAH GRAHAM HOOPER (728237)
ARIEL ROSENFELD (710567)
AVNISH DESAI (718369)

CONTENTS

NOTES

When talking about the ORDER BY statement and the sorting of the data it is assumed that if the direction of sorting is not specified then it is assumed to be in ascending order. If it is in descending order than it will be stated. This reflects SQL.

Where a query displays certain columns it is assumed that when data is displayed that the relevant data for those columns are also displayed.

Unless where stated, the like statements where string (varchar) comparisons are being done are not case-sensitive.

QUESTION 1 BUSSINESS RULES

- The Department Employs Employees.
- Employees Manages The Department.
- Employees Have Salaries.
- Employees Completes Invoices.
- Products Appear on Lines.
- Customers Places Invoices.
- Brands Includes Products.
- Products Provided By Supplies.
- Vendors Provides Supplies.

QUESTION 2 DATABASE SCREENSHOTS

Here are the screenshots of the database design. There is no real explanation since this is not really a query but the data we worked with.

LIST OF TABLE IN THE DATABASE

```
mysql> show tables;
+----------------------+
| Tables_in_710567_proj |
+----------------------+
| LGBRAND              |
| LGCUSTOMER           |
| LGDEPARTMENT         |
| LGEMPLOYEE           |
| LGINVOICE            |
| LGLINE               |
| LGPRODUCT            |
| LGSALARY_HISTORY     |
| LGSUPPLIES           |
| LGVENDOR             |
+----------------------+
10 rows in set (0.00 sec)
```

LGCUSTOMER

```
mysql> DESC LGCUSTOMER;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Cust_Code     | int(3)      | NO   | PRI | NULL    |       |
| Cust_Fname    | varchar(20) | NO   |     | NULL    |       |
| Cust_Lname    | varchar(20) | NO   |     | NULL    |       |
| Cust_Street   | varchar(50) | NO   |     | NULL    |       |
| Cust_City     | varchar(20) | NO   |     | NULL    |       |
| Cust_Province | varchar(20) | NO   |     | NULL    |       |
| Cust_ZIP      | varchar(5)  | NO   |     | NULL    |       |
| Cust_Balance  | double(8,2) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM LGCUSTOMER;
+-----------+------------+------------+---------------------------+-------------+---------------+----------+--------------+
| Cust_Code | Cust_Fname | Cust_Lname | Cust_Street               | Cust_City   | Cust_Province | Cust_ZIP | Cust_Balance |
+-----------+------------+------------+---------------------------+-------------+---------------+----------+--------------+
|         0 | John       | Doe        | 1 Yale Road               | Johannesburg | Gauteng       | 0000     |         1.00 |
|         1 | Pacific    | Rim        | 9 Jager Road              | Kaijuland   | Western Cape  | 14785    |     15000.00 |
|         2 | Iron Man   | 3          | 9 Tony Stark Road         | California  | Gauteng       | 14785    |     15700.00 |
|         3 | The Hunger | Games      | 11 Katniss Everdeen Street | Panem       | Limpopo       | 45678    |      5000.00 |
|         4 | Incep      | Tion       | 6 9th Street              | Panem       | Gauteng       | 45677    |        50.00 |
|         5 | Jane       | Doe        | 1 Yale Road               | Johannesburg | Gauteng       | 0000     |        50.00 |
+-----------+------------+------------+---------------------------+-------------+---------------+----------+--------------+
6 rows in set (0.00 sec)
```

LGINVOICE

```
mysql> DESC LGINVOICE;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Inv_Num     | int(3)      | NO   | PRI | NULL    |       |
| Cust_Code   | int(3)      | NO   | MUL | NULL    |       |
| Inv_Date    | date        | YES  |     | NULL    |       |
| Inv_Total   | double(8,2) | YES  |     | NULL    |       |
| Employee_ID | int(3)      | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGINVOICE;
+---------+-----------+------------+-----------+-------------+
| Inv_Num | Cust_Code | Inv_Date   | Inv_Total | Employee_ID |
+---------+-----------+------------+-----------+-------------+
|       1 |         0 | 2012-02-05 |   5000.00 |       83745 |
|       2 |         0 | 2012-08-06 |   7500.00 |       83745 |
|       3 |         0 | 2012-08-06 |  10000.00 |       83745 |
|       4 |         1 | 2011-08-06 |   9800.00 |       84039 |
|       5 |         1 | 2011-08-10 |  75000.00 |       84039 |
|       6 |         2 | 2011-09-10 |    500.00 |       84039 |
|       7 |         2 | 2011-01-10 |   5100.00 |       83649 |
|       8 |         2 | 2011-06-10 |  51020.00 |       83677 |
|       9 |         3 | 2010-06-10 | 151020.00 |       83677 |
|      10 |         4 | 2010-08-10 |   4020.00 |       83677 |
|      11 |         4 | 2010-08-19 |  14020.00 |       83745 |
|      12 |         5 | NULL       |      0.00 |       86987 |
|      13 |         5 | 2013-03-23 |      9.00 |       84789 |
+---------+-----------+------------+-----------+-------------+
13 rows in set (0.00 sec)
```

LGEMPLOYEE

```
mysql> DESC LGEMPLOYEE;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Emp_Num       | int(5)      | NO   | PRI | 0       |       |
| Emp_Fname     | varchar(20) | NO   |     | NULL    |       |
| Emp_Lname     | varchar(20) | NO   |     | NULL    |       |
| Emp_Email     | varchar(30) | NO   |     | NULL    |       |
| Emp_Phone     | varchar(10) | NO   |     | NULL    |       |
| Emp_Hiredate  | date        | NO   |     | NULL    |       |
| Emp_Title     | varchar(30) | NO   |     | NULL    |       |
| Emp_Comm      | double(2,2) | NO   |     | NULL    |       |
| Dept_Num      | int(3)      | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGEMPLOYEE;
+---------+-----------+-----------+--------------------------+------------+-------------+-----------------+----------+----------+
| Emp_Num | Emp_Fname | Emp_Lname | Emp_Email                | Emp_Phone  | Emp_Hiredate | Emp_Title       | Emp_Comm | Dept_Num |
+---------+-----------+-----------+--------------------------+------------+-------------+-----------------+----------+----------+
|   12345 | John      | Doe       | jd@gmail.com             | 0721234567 | 2014-03-17  | Senior Associate |     0.10 |      200 |
|   83649 | Luke      | Skywalker | lukesw@hotmail.com       | 0834567896 | 2012-12-05  | Manager         |     0.10 |      500 |
|   83677 | Mace      | Windu     | mastermace@hotmail.com   | 0831234568 | 2011-06-25  | Manager         |     0.60 |      200 |
|   83745 | Obi Wan   | Kenobi    | obk@hotmail.com          | 0837458965 | 2010-02-05  | Manager         |     0.23 |      300 |
|   84039 | Annakin   | Skywalker | annyskywalker@hotmail.com | 0837451215 | 2010-08-05  | Manager         |     0.23 |      400 |
|   84564 | Master    | Yoda      | yodamaster@hotmail.com   | 0827854123 | 2010-06-25  | Senior Associate |     0.60 |      300 |
|   84756 | Princess  | Leiya     | PLeiya@hotmail.com       | 0832547896 | 2012-05-14  | Junior Associate |     0.20 |      400 |
|   84789 | Han       | Solo      | flyingsolo@hotmail.com   | 0832544596 | 2012-06-14  | Junior Associate |     0.10 |      500 |
|   86987 | Chew      | Bacca     | chewy@hotmail.com        | 0114587456 | 2012-05-19  | CLERK1          |     0.20 |      200 |
+---------+-----------+-----------+--------------------------+------------+-------------+-----------------+----------+----------+
9 rows in set (0.00 sec)
```

LGDEPARTMENT

```
mysql> DESC LGDEPARTMENT;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Dept_num      | int(3)      | NO   | PRI | NULL    |       |
| Dept_Name     | varchar(20) | NO   |     | NULL    |       |
| Dept_Mail_Box | varchar(3)  | NO   |     | NULL    |       |
| Dept_Phone    | varchar(10) | NO   |     | NULL    |       |
| Emp_Num       | int(5)      | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGDEPARTMENT;
+----------+-----------+---------------+------------+---------+
| Dept_num | Dept_Name | Dept_Mail_Box | Dept_Phone | Emp_Num |
+----------+-----------+---------------+------------+---------+
|      200 | Admin     | 500           | 0111234567 |   83677 |
|      300 | Sales     | 250           | 0117852369 |   83745 |
|      400 | Marketing | 280           | 0117854589 |   84039 |
|      500 | Finance   | 210           | 0117858989 |   83649 |
+----------+-----------+---------------+------------+---------+
4 rows in set (0.00 sec)
```

LGLINE

```
mysql> DESC LGLINE;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Inv_Num    | int(3)      | NO   | PRI | NULL    |       |
| Line_Num   | int(3)      | NO   | PRI | NULL    |       |
| Prod_SKU   | int(3)      | NO   | MUL | NULL    |       |
| Line_Qty   | int(3)      | NO   |     | NULL    |       |
| Line_Price | double(8,2) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGLINE;
+---------+----------+----------+----------+------------+
| Inv_Num | Line_Num | Prod_SKU | Line_Qty | Line_Price |
+---------+----------+----------+----------+------------+
|       1 |        1 |        1 |        1 |      50.00 |
|       1 |        2 |        2 |        1 |      70.00 |
|       2 |        1 |        3 |        4 |     700.00 |
|       2 |        2 |        4 |        4 |     800.00 |
|       3 |        1 |        4 |        4 |     900.00 |
|       3 |        2 |        5 |        7 |    3500.00 |
|       4 |        1 |        1 |        2 |      20.00 |
|       4 |        2 |        2 |        2 |     200.00 |
|       4 |        3 |        3 |        2 |     400.00 |
|       5 |        1 |        4 |        2 |     400.00 |
|       5 |        2 |        5 |        3 |     500.00 |
|       5 |        5 |        4 |        3 |    1500.00 |
+---------+----------+----------+----------+------------+
12 rows in set (0.00 sec)
```

LGSALARY_HISTORY

```
mysql> DESC LGSALARY_HISTORY;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Emp_Num    | int(5)      | NO   | PRI | NULL    |       |
| Sal_From   | date        | NO   | PRI | NULL    |       |
| Sal_End    | date        | YES  |     | NULL    |       |
| Sal_Amount | double(8,2) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGSALARY_HISTORY;
+---------+------------+------------+------------+
| Emp_Num | Sal_From   | Sal_End    | Sal_Amount |
+---------+------------+------------+------------+
|   12345 | 2010-02-01 | 2011-03-02 |   12000.00 |
|   12345 | 2011-03-03 | NULL       |   97000.00 |
|   83649 | 2010-02-01 | 2011-03-02 |    1000.00 |
|   83649 | 2011-03-03 | NULL       |   87000.00 |
|   83677 | 2010-02-01 | 2011-03-02 |   12000.00 |
|   83677 | 2011-03-03 | NULL       |  123000.00 |
|   83745 | 2010-02-01 | 2011-03-02 |  121000.00 |
|   83745 | 2011-03-03 | 2013-12-31 |   12000.00 |
|   84039 | 2010-02-01 | 2011-03-02 |    4000.00 |
|   84039 | 2011-03-03 | 2013-12-31 |   40000.00 |
|   84564 | 2010-02-01 | 2011-03-02 |   41000.00 |
|   84564 | 2011-03-03 | 2013-12-31 |   11000.00 |
|   84564 | 2014-01-01 | NULL       |   12000.00 |
+---------+------------+------------+------------+
13 rows in set (0.00 sec)
```

LGBRAND

```
mysql> DESC LGBRAND;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Brand_ID    | int(3)      | NO   | PRI | NULL    |       |
| Brand_Name  | varchar(20) | NO   |     | NULL    |       |
| Brand_Type  | varchar(20) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGBRAND;
+----------+--------------------+------------+
| Brand_ID | Brand_Name         | Brand_Type |
+----------+--------------------+------------+
|        0 | Light Sabers       | Premium    |
|        2 | Tatooine Dust      | Regular    |
|        3 | Coruscant Painters | Premium    |
|        4 | Racer Pod Ballers  | Regular    |
+----------+--------------------+------------+
4 rows in set (0.00 sec)
```

LGPRODUCT

```
mysql> DESC LGPRODUCT;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Prod_SKU      | int(3)      | NO   | PRI | NULL    |       |
| Prod_Desc     | varchar(20) | NO   |     | NULL    |       |
| Prod_Type     | varchar(20) | NO   |     | NULL    |       |
| Prod_Base     | varchar(20) | NO   |     | NULL    |       |
| Prod_Category | varchar(20) | NO   |     | NULL    |       |
| Prod_Price    | double(5,2) | NO   |     | NULL    |       |
| Prod_QOH      | int(3)      | NO   |     | NULL    |       |
| Prod_Min      | int(3)      | NO   |     | NULL    |       |
| Brand_ID      | int(3)      | NO   | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

```
mysql> select * from LGPRODUCT;
+----------+--------------------+-------------+-------------+---------------+------------+----------+----------+----------+
| Prod_SKU | Prod_Desc          | Prod_Type   | Prod_Base   | Prod_Category | Prod_Price | Prod_QOH | Prod_Min | Brand_ID |
+----------+--------------------+-------------+-------------+---------------+------------+----------+----------+----------+
|        1 | Cleans like Baus   | Turpentine  | Water       | Cleaner       |      30.00 |      100 |        5 |        4 |
|        2 | No more holes      | Poly Filler | Polystyrene | Filler        |     400.00 |       70 |        7 |        2 |
|        3 | Best paint Tatooine | Paint       | Water       | Top Coat      |     300.00 |      100 |        5 |        3 |
|        4 | Opimus Primer      | Shiny       | Water       | Primer        |     450.00 |        5 |       10 |        4 |
|        5 | Navy Sealer        | Solid       | Cement      | Sealer        |     250.00 |       75 |        9 |        0 |
|        6 | Green Seal         | Solid       | Cement      | Sealer        |     100.00 |       50 |       12 |        3 |
|        7 | Tzanchan Rehydrate | Solid       | Water       | Sealer        |     999.98 |       90 |       34 |        0 |
|        8 | Almost finished    | Shiny       | Water       | Top coat      |     999.99 |        4 |        4 |        4 |
+----------+--------------------+-------------+-------------+---------------+------------+----------+----------+----------+
8 rows in set (0.00 sec)
```

LGSUPPLIES

```
mysql> DESC LGSUPPLIES;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| Vend_ID  | int(3)  | NO   | PRI | NULL    |       |
| Prod_SKU | int(3)  | NO   | PRI | NULL    |       |
+----------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGSUPPLIES;
+---------+----------+
| Vend_ID | Prod_SKU |
+---------+----------+
|       0 |        1 |
|       1 |        1 |
|       2 |        1 |
|       4 |        1 |
|       0 |        2 |
|       2 |        2 |
|       4 |        2 |
|       0 |        3 |
|       2 |        3 |
|       3 |        3 |
|       4 |        3 |
|       5 |        3 |
|       1 |        4 |
|       3 |        4 |
|       5 |        4 |
|       1 |        5 |
|       3 |        5 |
|       4 |        5 |
|       5 |        5 |
+---------+----------+
19 rows in set (0.00 sec)
```

LGVENDOR

```
mysql> DESC LGVENDOR;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Vend_ID       | int(3)      | NO   | PRI | NULL    |       |
| Vend_Name     | varchar(20) | NO   |     | NULL    |       |
| Vend_Street   | varchar(50) | NO   |     | NULL    |       |
| Vend_City     | varchar(20) | NO   |     | NULL    |       |
| Vend_Province | varchar(20) | NO   |     | NULL    |       |
| Vend_ZIP      | varchar(5)  | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM LGVENDOR;
+---------+--------------+----------------+--------------+---------------+----------+
| Vend_ID | Vend_Name    | Vend_Street    | Vend_City    | Vend_Province | Vend_ZIP |
+---------+--------------+----------------+--------------+---------------+----------+
|       0 | AB Devillies | 47 Bat Road    | Johannesburg | Gauteng       | 21458    |
|       1 | DW Steyn     | 47 Ball Road   | Pretoria     | Gauteng       | 14785    |
|       2 | F Du Plessis | 47 Field Road  | Cape Town    | Western Cape  | 47852    |
|       3 | S Gerrard    | 47 Penalty Road| Ceres        | Western Cape  | 41258    |
|       4 | C Adam       | 47 8th Road    | Roodeport    | Gauteng       | 36985    |
|       5 | C Crouch     | 47 10th Road   | Northcliff   | Gauteng       | 12345    |
+---------+--------------+----------------+--------------+---------------+----------+
6 rows in set (0.00 sec)
```

QUESTION 3

SQL

```
SELECT * FROM LGDEPARTMENT
```

DESCRIPTION

```
mysql> SELECT *
    -> FROM LGDEPARTMENT;
+----------+-----------+---------------+------------+---------+
| Dept_num | Dept_Name | Dept_Mail_Box | Dept_Phone | Emp_Num |
+----------+-----------+---------------+------------+---------+
|      200 | Admin     | 500           | 0111234567 |   83677 |
|      300 | Sales     | 250           | 0117852369 |   83745 |
|      400 | Marketing | 280           | 0117854589 |   84039 |
|      500 | Finance   | 210           | 0117858989 |   83649 |
+----------+-----------+---------------+------------+---------+
4 rows in set (0.00 sec)
```

This query will display all the columns and all the data from the table LGDEPARTMENT.

QUESTION 4

SQL

```
SELECT Prod_SKU, Prod_Desc, Prod_Type, Prod_Base, Prod_Category, Prod_Price
FROM LGPRODUCT
WHERE Prod_Base like 'water' AND Prod_Category like 'sealer';
```

DESCRIPTION

```
mysql> SELECT Prod_SKU, Prod_Desc, Prod_Type, Prod_Base, Prod_Category, Prod_Price
    -> FROM LGPRODUCT
    -> WHERE Prod_Base like 'water' AND Prod_Category like 'sealer';
+----------+--------------------+-----------+-----------+---------------+------------+
| Prod_SKU | Prod_Desc          | Prod_Type | Prod_Base | Prod_Category | Prod_Price |
+----------+--------------------+-----------+-----------+---------------+------------+
|        7 | Tzanchan Rehydrate | Solid     | Water     | Sealer        |     999.99 |
+----------+--------------------+-----------+-----------+---------------+------------+
1 row in set (0.00 sec)
```

This query displays the columns Prod_SKU, Prod_Desc, Prod_Type, Prod_Base, Prod_Category, and Prod_Price from the table LGPRODUCT, it then displays the data in the database which corresponds to having the Prod_Base value of 'water' and the Prod_Category value of 'sealer'. The case is not sensitive here.

QUESTION 5

SQL

```
SELECT Emp_Fname, Emp_Lname, Emp_Email
FROM LGEMPLOYEE
WHERE (Emp_Hiredate BETWEEN '2010-01-01' AND '2013-12-31')
ORDER BY Emp_Lname, Emp_Fname;
```

DESCRIPTION

```
mysql> SELECT Emp_Fname, Emp_Lname, Emp_Email
    -> FROM LGEMPLOYEE
    -> WHERE (Emp_Hiredate BETWEEN '2010-01-01' AND '2013-12-31')
    -> ORDER BY Emp_Lname, Emp_Fname;
+-----------+-----------+---------------------------+
| Emp_Fname | Emp_Lname | Emp_Email                 |
+-----------+-----------+---------------------------+
| Chew      | Bacca     | chewy@hotmail.com         |
| Obi Wan   | Kenobi    | obk@hotmail.com           |
| Princess  | Leiya     | PLeiya@hotmail.com        |
| Annakin   | Skywalker | annyskywalker@hotmail.com |
| Luke      | Skywalker | lukesw@hotmail.com        |
| Han       | Solo      | flyingsolo@hotmail.com    |
| Mace      | Windu     | mastermace@hotmail.com    |
| Master    | Yoda      | yodamaster@hotmail.com    |
+-----------+-----------+---------------------------+
8 rows in set (0.00 sec)
```

This query will display the columns Emp_Fname, Emp_Lname, Emp_Email from the table LGEMPLOYEE, and then it will display the data in the database where the employee hire date is between 2010-01-01 and 2013-12-31. The data returned will then be sorted first by the employee last name (Emp_Lname) and then by the employee first name (Emp_Fname).

QUESTION 6

SQL

```
SELECT Emp_Fname, Emp_Lname, Emp_Phone, Emp_Title, Dept_Num
FROM LGEMPLOYEE
WHERE Dept_Num = 300 OR BINARY Emp_Title like "CLERK1"
ORDER BY Emp_Lname, Emp_Fname;
```

DESCRIPTION

```
mysql> SELECT Emp_Fname, Emp_Lname, Emp_Phone, Emp_Title, Dept_Num
    -> FROM LGEMPLOYEE
    -> WHERE Dept_Num = 300 OR BINARY Emp_Title like "CLERK1"
    -> ORDER BY Emp_Lname, Emp_Fname;
+-----------+-----------+------------+-----------------+----------+
| Emp_Fname | Emp_Lname | Emp_Phone  | Emp_Title       | Dept_Num |
+-----------+-----------+------------+-----------------+----------+
| Chew      | Bacca     | 0114587456 | CLERK1          |      200 |
| Obi Wan   | Kenobi    | 0837458965 | Manager         |      300 |
| Master    | Yoda      | 0827854123 | Senior Associate |      300 |
+-----------+-----------+------------+-----------------+----------+
3 rows in set (0.00 sec)
```

This query will display the columns Emp_Fname, Emp_Lname, Emp_Phone, Emp_Title, Dept_Num from the table LGEMPLOYEE and then it will display the data where the department number (Dept_Num) is 300 or where the case-sensitive employee title (Emp_Title) is ‘CLERK1’. The use of BINARY is to type cast the statement (Emp_Title like "CLERK1") so that it will ensure a binary value of true or false is returned. The data is then sorted firstly by the employee last name (Emp_Lname) and then by the employee first name (Emp_Fname).

QUESTION 7

SQL

```
SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Fname,
   LGSALARY_HISTORY.Sal_From, LGSALARY_HISTORY.Sal_End,
   LGSALARY_HISTORY.Sal_Amount
FROM LGEMPLOYEE INNER JOIN LGSALARY_HISTORY ON
   LGEMPLOYEE.Emp_Num=LGSALARY_HISTORY.Emp_Num
WHERE LGEMPLOYEE.Emp_Num = 83731 OR LGEMPLOYEE.Emp_Num = 83745 OR
   LGEMPLOYEE.Emp_Num = 84039
ORDER BY LGEMPLOYEE.Emp_Num, LGSALARY_HISTORY.Sal_From;
```

DESCRIPTION

```
mysql> SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Fname, LGSALARY_HISTORY.Sal_From, LGSALARY_HISTORY.Sal_End, LGSALARY_HISTORY.Sal_Amount
    -> FROM LGEMPLOYEE INNER JOIN LGSALARY_HISTORY ON LGEMPLOYEE.Emp_Num=LGSALARY_HISTORY.Emp_Num
    -> WHERE LGEMPLOYEE.Emp_Num = 83731 OR LGEMPLOYEE.Emp_Num = 83745 OR LGEMPLOYEE.Emp_Num = 84039
    -> ORDER BY LGEMPLOYEE.Emp_Num, LGSALARY_HISTORY.Sal_From;
+---------+-----------+-----------+------------+------------+------------+
| Emp_Num | Emp_Lname | Emp_Fname | Sal_From   | Sal_End    | Sal_Amount |
+---------+-----------+-----------+------------+------------+------------+
|   83745 | Kenobi    | Obi Wan   | 2010-02-01 | 2011-03-02 |  121000.00 |
|   83745 | Kenobi    | Obi Wan   | 2011-03-03 | 2013-12-31 |   12000.00 |
|   84039 | Skywalker | Annakin   | 2010-02-01 | 2011-03-02 |    4000.00 |
|   84039 | Skywalker | Annakin   | 2011-03-03 | 2013-12-31 |   40000.00 |
+---------+-----------+-----------+------------+------------+------------+
4 rows in set (0.00 sec)
```

This query will display the columns Emp_Num, Emp_Lname and Emp_Fname from table LGEMPLOYEE and Sal_From, Sal_End, and Sal_Amount from table LGSALARY_HISTORY. The tables LGEMPLOYEE and LGSALARY_HISTORY are inner joined on the primary and foreign key Emp_Num. The inner join selects all the rows from both tables where there is a match between both of them. The data that is actually displayed from both tables is where the Emp_Num is equal to 83731, 83745 and 84039. The returned data is then sorted, firstly by the LGEMPLOYEE.Emp_Num and then the Sal_From attribute.

QUESTION 8

SQL

```
SELECT DISTINCT LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname,
   LGCUSTOMER.Cust_Street, LGCUSTOMER.Cust_City, LGCUSTOMER.Cust_Province,
   LGCUSTOMER.Cust_Zip
FROM LGCUSTOMER JOIN (LGINVOICE JOIN (LGLINE JOIN (LGPRODUCT JOIN LGBRAND ON
   LGPRODUCT.Brand_ID=LGBRAND.Brand_ID) ON LGLINE.Prod_SKU=LGPRODUCT.Prod_SKU)
   ON LGINVOICE.Inv_Num=LGLINE.Inv_Num) ON
   LGCUSTOMER.Cust_Code=LGINVOICE.Cust_Code
WHERE LGBRAND.Brand_Name like 'Coruscant Painters' AND LGPRODUCT.Prod_Category
   like 'Top Coat' AND (LGINVOICE.Inv_Date BETWEEN '2011-07-15' AND '2013-07-
   31')
ORDER BY LGCUSTOMER.Cust_Province, LGCUSTOMER.Cust_Lname,
   LGCUSTOMER.Cust_Fname;
```

DESCRIPTION

```
mysql> SELECT DISTINCT LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname, LGCUSTOMER.Cust_Street, LGCUSTOMER.Cust_City, LGCUSTOMER.Cust_Province, LGCUSTOMER.Cust_Zip
    -> FROM LGCUSTOMER JOIN (LGINVOICE JOIN (LGLINE JOIN (LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID) ON LGLINE.Prod_SKU=LGPRODUCT.Prod_SKU) ON LGINV
OICE.Inv_Num=LGLINE.Inv_Num) ON LGCUSTOMER.Cust_Code=LGINVOICE.Cust_Code
    -> WHERE LGBRAND.Brand_Name like 'Coruscant Painters' AND LGPRODUCT.Prod_Category like 'Top Coat' AND (LGINVOICE.Inv_Date BETWEEN '2011-07-15' AND '2013-07-31')
    -> ORDER BY LGCUSTOMER.Cust_Province, LGCUSTOMER.Cust_Lname, LGCUSTOMER.Cust_Fname;
+-----------+-----------+-------------+-------------+--------------+----------+
| Cust_Fname | Cust_Lname | Cust_Street  | Cust_City    | Cust_Province | Cust_Zip |
+-----------+-----------+-------------+-------------+--------------+----------+
| John       | Doe        | 1 Yale Road  | Johannesburg | Gauteng       | 0000     |
| Pacific    | Rim        | 9 Jager Road | Kaijuland    | Western Cape  | 14785    |
+-----------+-----------+-------------+-------------+--------------+----------+
2 rows in set (0.01 sec)
```

This query will display the columns Cust_Fname, Cust_Lname, Cust_Street, Cust_City, Cust_Province and Cust_Zip from the table LGCUSTOMER. The tables LGCUSTOMER and LGINVOICE have been joined on the attribute (primary and foreign key) Cust_Code, the tables LGINVOICE and LGLINE have been joined on the attribute (primary and foreign key) Inv_Num, the tables LGLINE and LGPRODUCT have been joined on the

attribute (primary and foreign key) Prod_SKU and the tables LGPRODUCT and LGBRAND have been joined on the attribute (primary and foreign key) Brand_ID. The join statements selects all the rows, from both tables which have been joined in each join, where there is a match between both of them. The data which is displayed in this query is where the Brand_name is 'Coruscant Painters', its Prod_Category is 'Top Coat' and its Inv_Date is betwwen 2011-07-15 and 2013-07-31. The returned data is then sorted, firstly by the Cust_Province, then Cust_Lname, and then Cust_Fname.

QUESTION 9

SQL

```
SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Email,
   LGEMPLOYEE.Emp_Title, LGDEPARTMENT.Dept_Name
FROM LGEMPLOYEE INNER JOIN LGDEPARTMENT ON
   LGEMPLOYEE.Dept_Num=LGDEPARTMENT.Dept_Num
WHERE Emp_Title like '%Associate'
ORDER BY LGDEPARTMENT.Dept_Name, LGEMPLOYEE.Emp_Title;
```

DESCRIPTION

```
mysql> SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Email, LGEMPLOYEE.Emp_Title, LGDEPARTMENT.Dept_Name
    -> FROM LGEMPLOYEE INNER JOIN LGDEPARTMENT ON LGEMPLOYEE.Dept_Num=LGDEPARTMENT.Dept_Num
    -> WHERE Emp_Title like '%Associate'
    -> ORDER BY LGDEPARTMENT.Dept_Name, LGEMPLOYEE.Emp_Title;
+---------+-----------+-----------------------+-----------------+-----------+
| Emp_Num | Emp_Lname | Emp_Email             | Emp_Title       | Dept_Name |
+---------+-----------+-----------------------+-----------------+-----------+
|   12345 | Doe       | jd@gmail.com          | Senior Associate | Admin     |
|   84789 | Solo      | flyingsolo@hotmail.com | Junior Associate | Finance   |
|   84756 | Leiya     | PLeiya@hotmail.com    | Junior Associate | Marketing |
|   84564 | Yoda      | yodamaster@hotmail.com | Senior Associate | Sales     |
+---------+-----------+-----------------------+-----------------+-----------+
4 rows in set (0.00 sec)
```

This query will display the columns LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Email, LGEMPLOYEE.Emp_Title, LGDEPARTMENT.Dept_Name from both the tables LGEMPLOYEE and LGDEPARTMENT respectively. The tables LGEMPLOYEE and LGDEPARTMENT have been joined on the attribute (primary and foreign key) Dept_Num. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The data which is displayed in this query is where the Employee Title (Emp_Title) ends in associate and nothing may come after it. This data is the sorted by the Dept_Name and then Emp_Title.

QUESTION 10

SQL

```
SELECT LGBRAND.Brand_Name, COUNT(*) AS 'Amount'
FROM LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID
GROUP BY LGBRAND.Brand_Name
ORDER BY LGBRAND.Brand_Name;
```

DESCRIPTION

```
mysql> SELECT LGBRAND.Brand_Name, COUNT(*) AS 'Amount'
    -> FROM LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID
    -> GROUP BY LGBRAND.Brand_Name
    -> ORDER BY LGBRAND.Brand_Name;
+--------------------+--------+
| Brand_Name         | Amount |
+--------------------+--------+
| Coruscant Painters |      2 |
| Light Sabers       |      2 |
| Racer Pod Ballers  |      3 |
| Tatooine Dust      |      1 |
+--------------------+--------+
4 rows in set (0.00 sec)
```

This query displays the columns Brand_Name from the table LGBRAND and a column called 'Amount'. This column 'Amount' is the count of all the records in LGPRODUCT i.e. the number of products of each brand name which are in the database. The data is grouped by each Brand Name and it is also sorted ASC by the Brand Name. The tables LGPRODUCT and LGBRAND have been joined on the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them.

QUESTION 11

SQL

```
SELECT Prod_Category, COUNT(*) AS 'Amount'
FROM LGPRODUCT
WHERE Prod_Base like 'water'
GROUP BY Prod_Category;
```

DESCRIPTION

```
mysql> SELECT Prod_Category, COUNT(*) AS 'Amount'
    -> FROM LGPRODUCT
    -> WHERE Prod_Base like 'water'
    -> GROUP BY Prod_Category;
+---------------+--------+
| Prod_Category | Amount |
+---------------+--------+
| Cleaner       |      1 |
| Primer        |      1 |
| Sealer        |      1 |
| Top Coat      |      2 |
+---------------+--------+
4 rows in set (0.01 sec)
```

This query displays the columns Prod_Category from the table LGPRODUCT and a column called 'Amount'. This column 'Amount' is the count of all the records in LGPRODUCT i.e. the number of products in each category that have a water base. The data which is selected is where the Prod_Base is 'water', and this is not case-sensitive. The data is grouped by each product category.

QUESTION 12

SQL

```
SELECT  Prod_Base, Prod_Type, COUNT(*) AS 'Amount'
FROM LGPRODUCT
GROUP BY Prod_Base, Prod_Type;
```

DESCRIPTION

```
mysql> SELECT Prod_Base, Prod_Type, COUNT(*) AS 'Amount'
    -> FROM LGPRODUCT
    -> GROUP BY Prod_Base, Prod_Type;
+-------------+-------------+--------+
| Prod_Base   | Prod_Type   | Amount |
+-------------+-------------+--------+
| Cement      | Solid       |      2 |
| Polystyrene | Poly Filler |      1 |
| Water       | Paint       |      1 |
| Water       | Shiny       |      2 |
| Water       | Solid       |      1 |
| Water       | Turpentine  |      1 |
+-------------+-------------+--------+
6 rows in set (0.00 sec)
```

This query displays the columns Prod_Base and Prod_Type from the table LGPRODUCT and also the column 'Amount'. The 'Amount' column is the count of all the records within each base and type combination from the LGPRODUCT table. The data is grouped by the unique combinations of the Prod_Base and Prod_Type.

QUESTION 13

SQL

```
SELECT LGBRAND.Brand_ID, COUNT(*) AS 'Amount'
FROM LGBRAND JOIN LGPRODUCT ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
GROUP BY Brand_ID
ORDER BY Brand_ID DESC;
```

DESCRIPTION

```
mysql> SELECT LGBRAND.Brand_ID, SUM(LGPRODUCT.Prod_QOH) AS 'Amount'
    -> FROM LGBRAND JOIN LGPRODUCT ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
    -> GROUP BY Brand_ID
    -> ORDER BY Brand_ID DESC;
+----------+--------+
| Brand_ID | Amount |
+----------+--------+
|        4 |    109 |
|        3 |    150 |
|        2 |     70 |
|        0 |    165 |
+----------+--------+
4 rows in set (0.00 sec)
```

This query displays the column Brand_ID from the table LGBRAND and a column called 'Amount'. The column 'Amount' is the sum of all products on hand for each brand ID. It is worked out by summing up all the data per grouping in the Prod_QOH column from the LGPRODUCT table. The data is grouped by the Brand_ID and sorted in descending order also by the Brand_ID. The tables LGBRAND and LGPRODUCT have been joined on the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them.

QUESTION 14

SQL

```
SELECT LGPRODUCT.Brand_ID, LGBRAND.Brand_Name, ROUND(AVG(LGPRODUCT.Prod_Price),
  2) AS 'Average'
FROM LGBRAND JOIN LGPRODUCT ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
GROUP BY LGPRODUCT.Brand_ID
ORDER BY LGBRAND.Brand_Name;
```

DESCRIPTION

```
mysql> SELECT LGPRODUCT.Brand_ID, LGBRAND.Brand_Name, ROUND(AVG(LGPRODUCT.Prod_Price), 2) AS 'Average'
    -> FROM LGBRAND JOIN LGPRODUCT ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
    -> GROUP BY LGPRODUCT.Brand_ID
    -> ORDER BY LGBRAND.Brand_Name;
+----------+-------------------+---------+
| Brand_ID | Brand_Name        | Average |
+----------+-------------------+---------+
|        3 | Coruscant Painters |  200.00 |
|        0 | Light Sabers      |  624.99 |
|        4 | Racer Pod Ballers |  493.33 |
|        2 | Tatooine Dust     |  400.00 |
+----------+-------------------+---------+
4 rows in set (0.01 sec)
```

This query is the same as Question 17. The query displays the columns Brand_ID and Brand_Name from table LGBRAND and a column called 'Amount'. The column 'Amount' is the rounded, to two decimal places, and is the average price of products of each brand in the LGBRAND and LGPRODUCT tables. The data is grouped by the Brand_ID and also sorted by the Brand Name. The tables LGBRAND and LGPRODUCT have been joined on

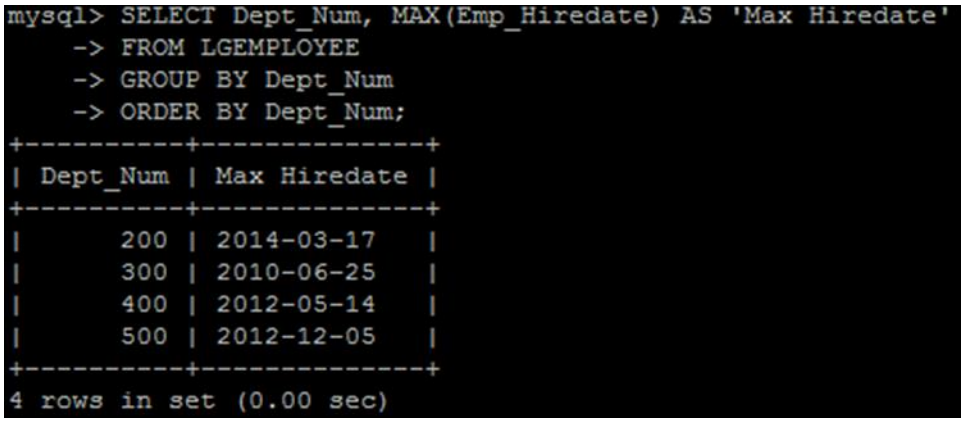
the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them.

QUESTION 15

SQL

```
SELECT Dept_Num, MAX(Emp_Hiredate) AS 'Max Hiredate'
FROM LGEMPLOYEE
GROUP BY Dept_Num
ORDER BY Dept_Num;
```

DESCRIPTION



This query is the same as Question 18. The query displays the columns Dept_Num from the table LGEMPLOYEE and a column called 'MAX Hiredate'. The data displayed in the query is the most recent employee hire date for each department in the database i.e. the data is grouped by the Dept_Num. The 'MAX Hiredate' column is the maximum or latest date within the Emp_Hiredate column from the LGEMPLOYEE table. The data is sorted by the Dept_Num in ascending order.

QUESTION 16

SQL

```
SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname,
   MAX(LGSALARY_HISTORY.Sal_Amount) AS 'Largest Salery Amount'
FROM LGEMPLOYEE JOIN LGSALARY_HISTORY ON
   LGEMPLOYEE.Emp_Num=LGSALARY_HISTORY.Emp_Num
WHERE LGEMPLOYEE.Dept_Num = 200
GROUP BY LGEMPLOYEE.Emp_Num
ORDER BY MAX(LGSALARY_HISTORY.Sal_Amount) DESC;
```

DESCRIPTION

```
mysql> SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname, MAX(LGSALARY_HISTORY.Sal_Amount) AS 'Largest Salery Amount'
    -> FROM LGEMPLOYEE JOIN LGSALARY_HISTORY ON LGEMPLOYEE.Emp_Num=LGSALARY_HISTORY.Emp_Num
    -> WHERE LGEMPLOYEE.Dept_Num = 200
    -> GROUP BY LGEMPLOYEE.Emp_Num
    -> ORDER BY MAX(LGSALARY_HISTORY.Sal_Amount) DESC;
+---------+-----------+-----------+----------------------+
| Emp_Num | Emp_Fname | Emp_Lname | Largest Salery Amount |
+---------+-----------+-----------+----------------------+
|   83677 | Mace      | Windu     |            123000.00 |
|   12345 | John      | Doe       |             97000.00 |
+---------+-----------+-----------+----------------------+
2 rows in set (0.01 sec)
```

This query is the same as Question 19. The query displays the columns Emp_Num, Emp_Fname, Emp_Lname from the table LGEMPLOYEE and also the column 'Largest Salary Amount'. The column 'Largest Salary Amount' is the maximum salary amount on record for each employee in department 200. The data is grouped by the LGEMPLOYEE.Emp_Num and sorted by the maximum salary amount on record for each employee in department 200. The tables LGEMPLOYEE and LGSALARY have been joined on the attribute (primary and foreign key) Emp_Num. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them.

QUESTION 17

SQL

```
SELECT LGPRODUCT.Brand_ID, LGBRAND.Brand_Name, ROUND(AVG(LGPRODUCT.Prod_Price),
  2) AS 'Average'
FROM LGBRAND JOIN LGPRODUCT ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
GROUP BY LGPRODUCT.Brand_ID
ORDER BY LGBRAND.Brand_Name;
```

DESCRIPTION

```
mysql> SELECT LGPRODUCT.Brand_ID, LGBRAND.Brand_Name, ROUND(AVG(LGPRODUCT.Prod_Price), 2) AS 'Average'
    -> FROM LGBRAND JOIN LGPRODUCT ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
    -> GROUP BY LGPRODUCT.Brand_ID
    -> ORDER BY LGBRAND.Brand_Name;
+----------+--------------------+---------+
| Brand_ID | Brand_Name         | Average |
+----------+--------------------+---------+
|        3 | Coruscant Painters |  200.00 |
|        0 | Light Sabers       |  624.99 |
|        4 | Racer Pod Ballers  |  493.33 |
|        2 | Tatooine Dust      |  400.00 |
+----------+--------------------+---------+
4 rows in set (0.01 sec)
```

This query is the same as Question 14. The query displays the columns Brand_ID and Brand_Name from table LGBRAND and a column called ‘Amount’. The column ‘Amount’ is the rounded, to two decimal places, and is the average price of products of each brand in the LGBRAND and LGPRODUCT tables. The data is grouped by the Brand_ID and also sorted by the Brand Name. The tables LGBRAND and LGPRODUCT have been joined on the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them.

QUESTION 18

SQL

```
SELECT Dept_Num, MAX(Emp_Hiredate) AS 'Max Hiredate'
FROM LGEMPLOYEE
GROUP BY Dept_Num
ORDER BY Dept_Num;
```

DESCRIPTION

```
mysql> SELECT Dept_Num, MAX(Emp_Hiredate) AS 'Max Hiredate'
    -> FROM LGEMPLOYEE
    -> GROUP BY Dept_Num
    -> ORDER BY Dept_Num;
+----------+--------------+
| Dept_Num | Max Hiredate |
+----------+--------------+
|      200 | 2014-03-17   |
|      300 | 2010-06-25   |
|      400 | 2012-05-14   |
|      500 | 2012-12-05   |
+----------+--------------+
4 rows in set (0.00 sec)
```

This query is the same as Question 15. The query displays the columns Dept_Num from the table LGEMPLOYEE and a column called 'MAX Hiredate'. The data displayed in the query is the most recent employee hire date for each department in the database i.e. the data is grouped by the Dept_Num. The 'MAX Hiredate' column is the maximum or latest date within the Emp_Hiredate column from the LGEMPLOYEE table. The data is sorted by the Dept_Num in ascending order.

QUESTION 19

SQL

```
SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname,
   MAX(LGSALARY_HISTORY.Sal_Amount) AS 'Largest Salery Amount'
FROM LGEMPLOYEE JOIN LGSALARY_HISTORY ON
   LGEMPLOYEE.Emp_Num=LGSALARY_HISTORY.Emp_Num
WHERE LGEMPLOYEE.Dept_Num = 200
GROUP BY LGEMPLOYEE.Emp_Num
ORDER BY MAX(LGSALARY_HISTORY.Sal_Amount) DESC;
```

DESCRIPTION

```
mysql> SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname, MAX(LGSALARY_HISTORY.Sal_Amount) AS 'Largest Salery Amount'
    -> FROM LGEMPLOYEE JOIN LGSALARY_HISTORY ON LGEMPLOYEE.Emp_Num=LGSALARY_HISTORY.Emp_Num
    -> WHERE LGEMPLOYEE.Dept_Num = 200
    -> GROUP BY LGEMPLOYEE.Emp_Num
    -> ORDER BY MAX(LGSALARY_HISTORY.Sal_Amount) DESC;
+---------+-----------+-----------+-----------------------+
| Emp_Num | Emp_Fname | Emp_Lname | Largest Salery Amount |
+---------+-----------+-----------+-----------------------+
|   83677 | Mace      | Windu     |             123000.00 |
|   12345 | John      | Doe       |              97000.00 |
+---------+-----------+-----------+-----------------------+
2 rows in set (0.01 sec)
```

This query is the same as Question 16. The query displays the columns Emp_Num, Emp_Fname, Emp_Lname from the table LGEMPLOYEE and also the column 'Largest Salary Amount'. The column 'Largest Salary Amount' is the maximum salary amount on record for each employee in department 200. The data is grouped by the LGEMPLOYEE.Emp_Num and sorted by the maximum salary amount on record for each employee in department 200. The tables LGEMPLOYEE and LGSALARY have been joined on the attribute (primary and foreign key) Emp_Num. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them.

QUESTION 20

SQL

```
SELECT LGCUSTOMER.Cust_Code, LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname,
   SUM(LGINVOICE.Inv_Total) AS 'Sum of Invoice Totals'
FROM LGCUSTOMER JOIN LGINVOICE ON LGCUSTOMER.Cust_Code = LGINVOICE.Cust_Code
WHERE LGINVOICE.Inv_Total > 1500
GROUP BY LGINVOICE.Cust_Code
ORDER BY SUM(LGINVOICE.Inv_Total) DESC;
```

DESCRIPTION

```
mysql> SELECT LGINVOICE.Cust_Code, LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname, SUM(LGINVOICE.Inv_Total) AS 'Sum of Invoice Totals'
    -> FROM LGCUSTOMER JOIN LGINVOICE ON LGCUSTOMER.Cust_Code = LGINVOICE.Cust_Code
    -> GROUP BY LGINVOICE.Cust_Code
    -> HAVING SUM(LGINVOICE.Inv_Total) > 1500
    -> ORDER BY SUM(LGINVOICE.Inv_Total) DESC;
+-----------+------------+------------+----------------------+
| Cust_Code | Cust_Fname | Cust_Lname | Sum of Invoice Totals |
+-----------+------------+------------+----------------------+
|         3 | The Hunger | Games      |            151020.00 |
|         1 | Pacific    | Rim        |             84800.00 |
|         2 | Iron Man   | 3          |             56620.00 |
|         0 | John       | Doe        |             22500.00 |
|         4 | Incep      | Tion       |             18040.00 |
+-----------+------------+------------+----------------------+
5 rows in set (0.00 sec)
```

The query displays the columns Cust_Code from LGINVOICE, Cust_Fname and Cust_Lname from table LGCUSTOMER and a column called ‘Sum of Invoice Totals’. The tables LGCUSTOMER and LGINVOICE have been joined on the attribute (primary and foreign key) Cust_Code. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The column ‘Sum of Invoice Totals’ is the sum of all invoice totals for customers with cumulative invoice totals greater than R1500. The data is sorted in descending order by the sum of invoice totals and is grouped by the Cust_Code.

QUESTION 21

SQL

```
SELECT LGDEPARTMENT.Dept_Num, LGDEPARTMENT.Dept_Name, LGDEPARTMENT.Dept_Phone,
   LGDEPARTMENT.Emp_Num, LGEMPLOYEE.Emp_Lname
FROM LGDEPARTMENT JOIN LGEMPLOYEE ON LGDEPARTMENT.Emp_Num = LGEMPLOYEE.Emp_Num
WHERE LGEMPLOYEE.Emp_Title like "manager"
ORDER BY LGDEPARTMENT.Dept_Name;
```

DESCRIPTION

```
mysql> SELECT LGDEPARTMENT.Dept_Num, LGDEPARTMENT.Dept_Name, LGDEPARTMENT.Dept_Phone, LGDEPARTMENT.Emp_Num, LGEMPLOYEE.Emp_Lname
    -> FROM LGDEPARTMENT JOIN LGEMPLOYEE ON LGDEPARTMENT.Emp_Num = LGEMPLOYEE.Emp_Num
    -> WHERE LGEMPLOYEE.Emp_Title like "manager"
    -> ORDER BY LGDEPARTMENT.Dept_Name;
+----------+-----------+------------+---------+-----------+
| Dept_Num | Dept_Name | Dept_Phone | Emp_Num | Emp_Lname |
+----------+-----------+------------+---------+-----------+
|      200 | Admin     | 0111234567 |   83677 | Windu     |
|      500 | Finance   | 0117858989 |   83649 | Skywalker |
|      400 | Marketing | 0117854589 |   84039 | Skywalker |
|      300 | Sales     | 0117852369 |   83745 | Kenobi    |
+----------+-----------+------------+---------+-----------+
4 rows in set (0.00 sec)

mysql>
```

The query displays the columns Dept_Num, Dept_Name, Dept_Phone and Emp_Num from the table LGDEPARTMENT and Emp_Lname from the table LGEMPLOYEE. The tables LGDEPARTMENT and LGEMPLOYEE have been joined on the attribute (primary and foreign key) Emp_Num. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays the data in each column relevant to each department manager. The data is sorted by the Dept_Name in ascending order. The joining on the attribute Emp_Num may determine the managers per department but the where statement reinforces that the data returned by the query is only relevant to the managers of each department.

QUESTION 22

SQL

```
SELECT LGSUPPLIES.Vend_ID, LGVENDOR.Vend_Name, LGBRAND.Brand_Name, COUNT(*) AS
   'Amount'
FROM LGVENDOR JOIN (LGSUPPLIES JOIN (LGPRODUCT JOIN LGBRAND ON
   LGPRODUCT.Brand_ID=LGBRAND.Brand_ID) ON
   LGSUPPLIES.Prod_SKU=LGPRODUCT.Prod_SKU) ON
   LGVENDOR.Vend_ID=LGSUPPLIES.Vend_ID
GROUP BY LGSUPPLIES.Vend_ID, LGPRODUCT.Brand_ID
ORDER BY LGVENDOR.Vend_Name, LGBRAND.Brand_Name;
```

DESCRIPTION

```
mysql> SELECT LGSUPPLIES.Vend_ID, LGVENDOR.Vend_Name, LGBRAND.Brand_Name, COUNT(*) AS 'Amount'
    -> FROM LGVENDOR JOIN (LGSUPPLIES JOIN (LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID) ON LGSUPPLIES.Prod_SKU=LGPRODUCT.Prod_SKU) ON LGVENDOR.Vend_I
D=LGSUPPLIES.Vend_ID
    -> GROUP BY LGSUPPLIES.Vend_ID, LGPRODUCT.Brand_ID
    -> ORDER BY LGVENDOR.Vend_Name, LGBRAND.Brand_Name;
+---------+-------------+-------------------+--------+
| Vend_ID | Vend_Name   | Brand_Name        | Amount |
+---------+-------------+-------------------+--------+
|       0 | AB Devillies | Coruscant Painters |      1 |
|       0 | AB Devillies | Racer Pod Ballers  |      1 |
|       0 | AB Devillies | Tatooine Dust      |      1 |
|       4 | C Adam       | Coruscant Painters |      1 |
|       4 | C Adam       | Light Sabers       |      1 |
|       4 | C Adam       | Racer Pod Ballers  |      1 |
|       4 | C Adam       | Tatooine Dust      |      1 |
|       5 | C Crouch     | Coruscant Painters |      1 |
|       5 | C Crouch     | Light Sabers       |      1 |
|       5 | C Crouch     | Racer Pod Ballers  |      1 |
|       1 | DW Steyn     | Light Sabers       |      1 |
|       1 | DW Steyn     | Racer Pod Ballers  |      2 |
|       2 | F Du Plessis | Coruscant Painters |      1 |
|       2 | F Du Plessis | Racer Pod Ballers  |      1 |
|       2 | F Du Plessis | Tatooine Dust      |      1 |
|       3 | S Gerrard    | Coruscant Painters |      1 |
|       3 | S Gerrard    | Light Sabers       |      1 |
|       3 | S Gerrard    | Racer Pod Ballers  |      1 |
+---------+-------------+-------------------+--------+
18 rows in set (0.00 sec)
```

The query displays the columns Vend_ID and Vend_Name from the table LGVENDOR and Brand_Name from the table LGBRAND and a column called ‘Amount’. The column ‘Amount’ is the count of products of each brand supplied by each vendor. There are three joins in this statement. The tables LGVENDOR and LGSUPPLIES have been joined on the attribute (primary and foreign key) Vend_ID. The tables LGSUPPLIES and LGPRODUCT have been joined on the attribute (primary and foreign key) Prod_SKU. The tables LGPRODUCT and LGBRAND have been joined on the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays the number of products of each brand supplied by each vendor. The data is grouped by the LGSUPPLIES.Vend_ID and LGPRODUCT.Brand_ID. The data is sorted by LGVENDOR.Vend_Name and then by the LGBRAND.Brand_Name.

QUESTION 23

SQL

```
SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Fname,
   SUM(LGINVOICE.Inv_Total) AS 'Sum of Invoice Totals'
FROM LGEMPLOYEE JOIN LGINVOICE ON LGEMPLOYEE.Emp_Num=LGINVOICE.Employee_ID
GROUP BY LGEMPLOYEE.Emp_Num
HAVING SUM(LGINVOICE.Inv_Total)>0
ORDER BY LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Fname;
```

DESCRIPTION

```
mysql> select LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Fname, sum(LGINVOICE.Inv_Total) as 'Sum of Invoice Totals'
    -> from LGEMPLOYEE join LGINVOICE on LGEMPLOYEE.Emp_Num=LGINVOICE.Employee_ID
    -> group by LGEMPLOYEE.Emp_Num
    -> having sum(LGINVOICE.Inv_Total)>0
    -> order by LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Fname;
+---------+-----------+-----------+----------------------+
| Emp_Num | Emp_Lname | Emp_Fname | Sum of Invoice Totals |
+---------+-----------+-----------+----------------------+
|   83745 | Kenobi    | Obi Wan   |             36520.00 |
|   84039 | Skywalker | Annakin   |             85300.00 |
|   83649 | Skywalker | Luke      |              5100.00 |
|   84789 | Solo      | Han       |                 9.00 |
|   83677 | Windu     | Mace      |            206060.00 |
+---------+-----------+-----------+----------------------+
5 rows in set (0.00 sec)
```

The query displays the columns Emp_Num, Emp_Lname and Emp_Fname from the table LGEMPLOYEE and the column 'Sum of Invoice Totals'. The column 'Sum of Invoice Totals' is the sum of invoice totals (Inv_Total) for all employees who completed an invoice. The tables LGEMPLOYEE and LGINVOICE have been joined on the attribute (primary and foreign key) Emp_Num (Employee_ID in LGINVOICE). The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The data is grouped by Emp_Num and sorted by Emp_Lname and then Emp_Fname. The data which is displayed is where the 'Sum of Invoice Totals' is greater than 0.

QUESTION 24

SQL

```
SELECT AVG(Prod_Price) AS 'Largest Average Product Price'
FROM LGPRODUCT
GROUP BY Brand_ID
ORDER BY AVG(Prod_Price) DESC
LIMIT 1;
```

DESCRIPTION

```
mysql> SELECT Brand_ID, AVG(Prod_Price) AS 'Largest Average Product Price'
    -> FROM LGPRODUCT
    -> GROUP BY Brand_ID
    -> ORDER BY AVG(Prod_Price) DESC
    -> LIMIT 1;
+----------+--------------------------------+
| Brand_ID | Largest Average Product Price |
+----------+--------------------------------+
|        0 |                     624.995000 |
+----------+--------------------------------+
1 row in set (0.00 sec)
```

The query displays the columns Brand_ID from the table LGPRODUCT and the column ‘Largest Average Product Price’. The ‘Largest Average Product Price’ column is the largest average product price of any brand. The data which is displayed is limited to the first entry which has been sorted in descending order which gives

the maximum average product price without using nested queries. The data is grouped by the Brand_ID and is sorted by the 'Largest Average Product Price' in descending order.

QUESTION 25

SQL

```
SELECT LGBRAND.Brand_ID, LGBRAND.Brand_Name, LGBRAND.Brand_Type,
   AVG(LGPRODUCT.Prod_Price) AS 'Average Product Price'
FROM LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID
GROUP BY LGBRAND.Brand_ID
ORDER BY AVG(LGPRODUCT.Prod_Price) DESC
LIMIT 1;
```

DESCRIPTION

```
mysql> SELECT LGBRAND.Brand_ID, LGBRAND.Brand_Name, LGBRAND.Brand_Type, AVG(LGPRODUCT.Prod_Price) AS 'Average Product Price'
    -> FROM LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID
    -> GROUP BY LGBRAND.Brand_ID
    -> ORDER BY AVG(LGPRODUCT.Prod_Price) DESC
    -> LIMIT 1;
+----------+-------------+------------+-----------------------+
| Brand_ID | Brand_Name  | Brand_Type | Average Product Price |
+----------+-------------+------------+-----------------------+
|        0 | Light Sabers | Premium    |            624.995000 |
+----------+-------------+------------+-----------------------+
1 row in set (0.00 sec)
```

The query displays the columns Brand_ID, Brand_Name and Brand_Type from the table LGBRAND and the column called 'Average Product Price'. The 'Average Product Price' is the average price of products for the brand that has the largest average product price. The data is grouped by Brand_ID and it is sorted by the 'Average Product Price' in descending order. The data which is displayed is limited to the first entry which has been sorted in descending order which gives the maximum average product price without using nested queries.

QUESTION 26

SQL

```
SELECT CONCAT(M.Emp_Fname, ' ', M.Emp_Lname) AS 'Manager_Name',
   LGDEPARTMENT.Dept_Name, LGDEPARTMENT.Dept_Phone, CONCAT(E.Emp_Fname, ' ',
   E.Emp_Lname) AS 'Emp_Name', CONCAT(LGCUSTOMER.Cust_Fname, ' ',
   LGCUSTOMER.Cust_Lname) AS 'Cust_Name', LGINVOICE.Inv_Date,
   LGINVOICE.Inv_Total
FROM LGCUSTOMER JOIN (LGINVOICE JOIN (LGEMPLOYEE E JOIN (LGDEPARTMENT JOIN
   LGEMPLOYEE M ON LGDEPARTMENT.Emp_Num=M.Emp_Num) ON
   E.Dept_Num=LGDEPARTMENT.Dept_Num) ON LGINVOICE.Employee_ID=E.Emp_Num) ON
   LGCUSTOMER.Cust_Code=LGINVOICE.Cust_Code
WHERE Cust_Lname like 'Doe' AND Inv_Date='2013-03-23';
```

DESCRIPTION

```
mysql> SELECT CONCAT(M.Emp_Fname, ' ', M.Emp_Lname) AS 'Manager_Name', LGDEPARTMENT.Dept_Name, LGDEPARTMENT.Dept_Phone, CONCAT(E.Emp_Fname, ' ', E.Emp_Lname) AS 'Emp_Na
me', CONCAT(LGCUSTOMER.Cust_Fname, ' ', LGCUSTOMER.Cust_Lname) AS 'Cust_Name', LGINVOICE.Inv_Date, LGINVOICE.Inv_Total
    -> FROM LGCUSTOMER JOIN (LGINVOICE JOIN (LGEMPLOYEE E JOIN (LGDEPARTMENT JOIN LGEMPLOYEE M ON LGDEPARTMENT.Emp_Num=M.Emp_Num) ON E.Dept_Num=LGDEPARTMENT.Dept_Num) O
N LGINVOICE.Employee_ID=E.Emp_Num) ON LGCUSTOMER.Cust_Code=LGINVOICE.Cust_Code
   -> WHERE Cust_Lname like 'Doe' AND Inv_Date='2013-03-23';
+---------------+-----------+------------+----------+-----------+------------+-----------+
| Manager_Name  | Dept_Name | Dept_Phone | Emp_Name | Cust_Name | Inv_Date   | Inv_Total |
+---------------+-----------+------------+----------+-----------+------------+-----------+
| Luke Skywalker | Finance   | 0117858989 | Han Solo | Jane Doe  | 2013-03-23 |      9.00 |
+---------------+-----------+------------+----------+-----------+------------+-----------+
1 row in set (0.00 sec)
```

The query displays the columns Dept_Name and Dept_Phone from the table LGDEPARTMENT and Inv_Date and Inv_Total from the table LGINVOICE. The column Manager_Name is also displayed from the table LGDEPARTMENT and it is a concatenation of Emp_Fname, a string containing an empty space, and Emp_Lname from table M. The column Emp_Name is displayed from the table LGINVOICE and it is a concatenation of Emp_Fname, a space and Emp_Lname from table E. The column Cust_Name is displayed from the table LGINVOICE and it is a concatenation of Cust_Fname, a space and Cust_Lname from the table LGCUSTOMER. Tables E and M are both copies of LGEMPLOYEE. The tables LGCUSTOMER and LGINVOICE have been joined on the attribute (primary and foreign key) Cust_Code. The tables LGINVOICE and E have been joined on the attribute (primary and foreign key) Emp_Num. The tables E and LGDEPARTMENT have been joined on the attribute (primary and foreign key) Dept_Num. The tables LGDEPARTMENT and M have been joined on the attribute (primary and foreign key) Emp_Num. The last join ignores all non-managers. The data is displayed where the last name of a customer is 'Doe' and the invoice date (Inv-Date) is 2013-03-23.

QUESTION 27

SQL

```
SELECT *
FROM LGPRODUCT
WHERE Prod_Price > 50;
```

DESCRIPTION

```
mysql> select *
    -> from LGPRODUCT
    -> where Prod_Price > 50;
+----------+--------------------+-------------+-------------+---------------+------------+----------+----------+----------+
| Prod_SKU | Prod_Desc          | Prod_Type   | Prod_Base   | Prod_Category | Prod_Price | Prod_QOH | Prod_Min | Brand_ID |
+----------+--------------------+-------------+-------------+---------------+------------+----------+----------+----------+
|        2 | No more holes      | Poly Filler | Polystyrene | Filler        |     400.00 |       70 |        7 |        2 |
|        3 | Best paint Tatooine | Paint       | Water       | Top Coat      |     300.00 |      100 |        5 |        3 |
|        4 | Opimus Primer      | Shiny       | Water       | Primer        |     450.00 |        5 |       10 |        4 |
|        5 | Navy Sealer        | Solid       | Cement      | Sealer        |     250.00 |       75 |        9 |        0 |
|        6 | Green Seal         | Solid       | Cement      | Sealer        |     100.00 |       50 |       12 |        3 |
|        7 | Tzanchan Rehydrate | Solid       | Water       | Sealer        |     999.98 |       90 |       34 |        0 |
|        8 | Almost finished    | Shiny       | Water       | Top coat      |     999.99 |        4 |        4 |        4 |
+----------+--------------------+-------------+-------------+---------------+------------+----------+----------+----------+
7 rows in set (0.00 sec)
```

The query displays all the columns from the table LGPRODUCT. The data which is displayed is where the Prod_Price is greater than 50.

QUESTION 28

SQL

```
SELECT LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname, LGSALARY_HISTORY.Sal_Amount
FROM LGSALARY_HISTORY JOIN LGEMPLOYEE ON
   LGSALARY_HISTORY.Emp_Num=LGEMPLOYEE.Emp_Num
WHERE LGEMPLOYEE.Dept_Num=300 AND LGSALARY_HISTORY.Sal_End IS NULL
ORDER BY LGSALARY_HISTORY.Sal_Amount DESC;
```

DESCRIPTION

```
mysql> SELECT LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname, LGSALARY_HISTORY.Sal_Amount
    -> FROM LGSALARY_HISTORY JOIN LGEMPLOYEE ON LGSALARY_HISTORY.Emp_Num=LGEMPLOYEE.Emp_Num
    -> WHERE LGEMPLOYEE.Dept_Num=300 AND LGSALARY_HISTORY.Sal_End IS NULL
    -> ORDER BY LGSALARY_HISTORY.Sal_Amount DESC;
+-----------+-----------+------------+
| Emp_Fname | Emp_Lname | Sal_Amount |
+-----------+-----------+------------+
| Master    | Yoda      |   12000.00 |
+-----------+-----------+------------+
1 row in set (0.00 sec)
```

The query displays the columns Emp_Fname and Emp_Lname from the table LGEMPLOYEE and Sal_Amount from the table LGSALARY_HISTORY. The tables LGSALARY_HISTORY and LGEMPLOYEE have been joined on the attribute (primary and foreign key) Emp_Num. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays the current salary for each employee in department 300 where their Sal_End (Salary end) is NULL since they are still current employees. The data is sorted by Sal_Amount in descending order.

QUESTION 29

SQL

```
SELECT Emp_Num, Sal_Amount
FROM LGSALARY_HISTORY
GROUP BY Emp_Num
HAVING MIN(Sal_From)
ORDER BY Emp_Num;
```

DESCRIPTION

```
mysql> SELECT Emp_Num, Sal_Amount
    -> FROM LGSALARY_HISTORY
    -> GROUP BY Emp_Num
    -> HAVING MIN(Sal_From)
    -> ORDER BY Emp_Num;
+---------+------------+
| Emp_Num | Sal_Amount |
+---------+------------+
|   12345 |   12000.00 |
|   83649 |    1000.00 |
|   83677 |   12000.00 |
|   83745 |  121000.00 |
|   84039 |    4000.00 |
|   84564 |   41000.00 |
+---------+------------+
6 rows in set (0.00 sec)
```

The query displays the columns Emp_Num and Sal_Amount from the table LGSALARY_HISTORY. The query returns the starting salary for each employee. This is calculated as the salary with the oldest start date i.e. the minimum Sal_From date. The data is sorted by the Emp_Num ascending and is grouped by the Emp_Num as well.

QUESTION 30

SQL

```
SELECT LGINVOICE.Inv_Num, LGLINE.Prod_SKU, LGPRODUCT.Prod_Desc,
   LGBRAND.Brand_ID
FROM LGINVOICE JOIN (LGLINE JOIN (LGPRODUCT JOIN LGBRAND ON
   LGPRODUCT.Brand_ID=LGBRAND.Brand_ID) ON LGLINE.Prod_SKU=LGPRODUCT.Prod_SKU)
   ON LGINVOICE.Inv_Num=LGLINE.Inv_Num
WHERE LGPRODUCT.Prod_Category like 'sealer' OR LGPRODUCT.Prod_Category like
   'top coat'
GROUP BY LGPRODUCT.Brand_ID, LGINVOICE.Inv_Num;
```

DESCRIPTION

```
mysql> SELECT LGINVOICE.Inv_Num, LGLINE.Prod_SKU, LGPRODUCT.Prod_Desc, LGBRAND.Brand_ID
    -> FROM LGINVOICE JOIN (LGLINE JOIN (LGPRODUCT JOIN LGBRAND ON LGPRODUCT.Brand_ID=LGBRAND.Brand_ID) ON LGLINE.Prod_SKU=LGPRODUCT.Prod_SKU) ON LGINVOICE.Inv_Num=LGLI
NE.Inv_Num
    -> WHERE LGPRODUCT.Prod_Category like 'sealer' OR LGPRODUCT.Prod_Category like 'top coat'
    -> GROUP BY LGPRODUCT.Brand_ID, LGINVOICE.Inv_Num;
+---------+----------+--------------------+----------+
| Inv_Num | Prod_SKU | Prod_Desc          | Brand_ID |
+---------+----------+--------------------+----------+
|       3 |        5 | Navy Sealer        |        0 |
|       5 |        5 | Navy Sealer        |        0 |
|       2 |        3 | Best paint Tatooine |        3 |
|       4 |        3 | Best paint Tatooine |        3 |
+---------+----------+--------------------+----------+
4 rows in set (0.00 sec)
```

The query displays the columns Inv_Num from the table LGINVOICE, and Prod_SKU and Prod_Desc from the table LGPRODUCT and Brand_ID from the table LGBRAND. The tables LGINVOICE and LGLINE have been joined on the attribute (primary and foreign key) Inv_Num. The tables LGLINE and LGPRODUCT have been joined on the attribute (primary and foreign key) Prod_SKU. The tables LGPRODUCT and LGBRAND have been joined on the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables

which have been joined where there is a match between both of them. The query displays data from the tables where the Prod_Category from LGPRODUCT is ‘sealer’ or it is ‘top coat’. The data is grouped by Brand_ID and then Inv_Num.

QUESTION 31

SQL

```
SELECT LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname,
   LGEMPLOYEE.Emp_Email, SUM(LGLINE.Line_Qty) AS 'Total sold'
FROM LGEMPLOYEE JOIN (LGINVOICE JOIN (LGLINE JOIN (LGPRODUCT JOIN LGBRAND ON
   LGBRAND.Brand_ID = LGPRODUCT.Brand_ID) ON LGPRODUCT.Prod_SKU =
   LGLINE.Prod_SKU) ON LGLINE.Inv_Num = LGINVOICE.Inv_Num) ON
   LGINVOICE.Employee_ID = LGEMPLOYEE.Emp_Num
WHERE LGBRAND.Brand_Name LIKE 'Coruscant Painters' AND LGINVOICE.Inv_Date
   BETWEEN '2011-11-01' AND '2013-12-01'
GROUP BY LGEMPLOYEE.Emp_Num
ORDER BY LGEMPLOYEE.Emp_Lname;
```

DESCRIPTION

```
mysql> select LGEMPLOYEE.Emp_Num, LGEMPLOYEE.Emp_Fname, LGEMPLOYEE.Emp_Lname, LGEMPLOYEE.Emp_Email, sum(LGLINE.Line_Qty) as 'Total sold'
    -> from LGEMPLOYEE join (LGINVOICE join (LGLINE join (LGPRODUCT join LGBRAND on LGBRAND.Brand_ID = LGPRODUCT.Brand_ID) on LGPRODUCT.Prod_SKU = LGLINE.Prod_SKU) on L
GLINE.Inv_Num = LGINVOICE.Inv_Num) on LGINVOICE.Employee_ID = LGEMPLOYEE.Emp_Num
    -> where LGBRAND.Brand_Name like 'Coruscant Painters' and LGINVOICE.Inv_Date between '2011-11-01' and '2013-12-01'
    -> group by LGEMPLOYEE.Emp_Num
    -> order by LGEMPLOYEE.Emp_Lname;
+---------+-----------+-----------+----------------+------------+
| Emp_Num | Emp_Fname | Emp_Lname | Emp_Email      | Total sold |
+---------+-----------+-----------+----------------+------------+
|   83745 | Obi Wan   | Kenobi    | obk@hotmail.com |          4 |
+---------+-----------+-----------+----------------+------------+
1 row in set (0.00 sec)
```

The query displays the columns Emp_Num, Emp_Fname, Emp_Lname and Emp_Email from the table LGEMPLOYEE and also the column 'Total sold'. The column 'Total sold' is the sum of Line_Qty from the table LGLINE. The tables LGEMPLOYEE and LGINVOICE have been joined on the attribute (primary and foreign key). The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays the employee who has sold the most 'Coruscant Painters' products between '2011-11-01' AND '2013-12-01'. The data is grouped by Emp_Num and is sorted by the employee last name (Emp_Lname).

QUESTION 32

SQL

```
SELECT L1.Cust_Code, LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname
FROM LGINVOICE L1 JOIN (LGCUSTOMER JOIN LGINVOICE L2 ON
   LGCUSTOMER.Cust_Code=L2.Cust_Code) ON L1.Cust_Code=LGCUSTOMER.Cust_Code
WHERE L1.Employee_ID=83649 AND L2.Employee_ID=83677
ORDER BY LGCUSTOMER.Cust_Lname, LGCUSTOMER.Cust_Fname;
```

DESCRIPTION

```
mysql> SELECT L1.Cust_Code, LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname
    -> FROM LGINVOICE L1 JOIN (LGCUSTOMER JOIN LGINVOICE L2 ON LGCUSTOMER.Cust_Code=L2.Cust_Code) ON L1.Cust_Code=LGCUSTOMER.Cust_Code
    -> WHERE L1.Employee_ID=83649 AND L2.Employee_ID=83677
    -> ORDER BY LGCUSTOMER.Cust_Lname, LGCUSTOMER.Cust_Fname;
+-----------+------------+------------+
| Cust_Code | Cust_Fname | Cust_Lname |
+-----------+------------+------------+
|         2 | Iron Man   | 3          |
+-----------+------------+------------+
1 row in set (0.00 sec)
```

The query displays the columns Cust_Code from the table L1, and Cust_Fname and Cust_Lname from the table LGCUSTOMER. The tables L1 and L2 are both copies of the table LGINVOICE. The tables L1 and LGCUSTOMER have been joined on the attribute (primary and foreign key) Cust_Code. The tables LGCUSTOMER and L2 have been joined on the attribute (primary and foreign key) Cust_Code. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays customer information of customers who have had at least one invoice completed by employee 83649 and employee 83677. The where statement has the conditions where Employee_ID (from L1) is 83649 and Employee_ID (from L2) is 83677. The data is sorted by Cust_Lname and then Cust_Fname.

QUESTION 33

SQL

```
SELECT LGCUSTOMER.Cust_Code, LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname,
  CONCAT(LGCUSTOMER.Cust_Street, ' ', LGCUSTOMER.Cust_City, ' ',
  LGCUSTOMER.Cust_Province, ' ', LGCUSTOMER.Cust_ZIP) AS 'Full_Address',
  LGINVOICE.Inv_Date, MAX(LGINVOICE.Inv_Total) AS 'Largest Invoice'
FROM LGCUSTOMER JOIN LGINVOICE ON LGCUSTOMER.Cust_Code=LGINVOICE.Cust_Code
WHERE LGCUSTOMER.Cust_Province like 'Gauteng'
GROUP BY LGCUSTOMER.Cust_Code;
```

DESCRIPTION

```
mysql> SELECT LGCUSTOMER.Cust_Code, LGCUSTOMER.Cust_Fname, LGCUSTOMER.Cust_Lname, CONCAT(LGCUSTOMER.Cust_Street, ' ', LGCUSTOMER.Cust_City, ' ', LGCUSTOMER.Cust_Provinc
e, ' ', LGCUSTOMER.Cust_ZIP) AS 'Full_Address', LGINVOICE.Inv_Date, MAX(LGINVOICE.Inv_Total) AS 'Largest Invoice'
    -> FROM LGCUSTOMER JOIN LGINVOICE ON LGCUSTOMER.Cust_Code=LGINVOICE.Cust_Code
    -> WHERE LGCUSTOMER.Cust_Province like 'Gauteng'
    -> GROUP BY LGCUSTOMER.Cust_Code;
+-----------+------------+------------+-----------------------------------------+------------+----------------+
| Cust_Code | Cust_Fname | Cust_Lname | Full_Address                            | Inv_Date   | Largest Invoice |
+-----------+------------+------------+-----------------------------------------+------------+----------------+
|         0 | John       | Doe        | 1 Yale Road Johannesburg Gauteng 0000   | 2012-02-05 |        10000.00 |
|         2 | Iron Man   | 3          | 9 Tony Stark Road California Gauteng 14785 | 2011-09-10 |        51020.00 |
|         4 | Incep      | Tion       | 6 9th Street Panem Gauteng 45677        | 2010-08-10 |        14020.00 |
|         5 | Jane       | Doe        | 1 Yale Road Johannesburg Gauteng 0000   | NULL       |            9.00 |
+-----------+------------+------------+-----------------------------------------+------------+----------------+
4 rows in set (0.00 sec)
```

The query displays the columns Cust_Code, Cust_Fname and Cust_Lname from the table LGCUSTOMER, and ‘Full Address’ , Inv_Date from the table LGINVOICE and the column called 'Largest Invoice'. The column ‘Full Address’ is a concatenation of Cust_Street, a string containing a space, Cust_City, a string containing a space, Cust_Province, a string containing a space and Cust_ZIP where all the non-whitespace components are columns from the table LGCUSTOMER. The column 'Largest Invoice' is the maximum value of Inv_Total grouped by the Cust_Code. The tables LGCUSTOMER and LGINVOICE have been joined on the attribute (primary and foreign key) Cust_Code. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays the customer information where the customer province (Cust_Province) is Gauteng. The data is grouped by the Cust_Code.

QUESTION 34

SQL

```
SELECT LGBRAND.Brand_Name, LGBRAND.Brand_Type, AVG(LGPRODUCT.Prod_Price) AS
   'Average', SUM(LGLINE.Line_Qty) AS 'Total Units Sold'
FROM LGBRAND JOIN (LGPRODUCT JOIN LGLINE ON LGPRODUCT.Prod_SKU=LGLINE.Prod_SKU)
   ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
GROUP BY LGBRAND.Brand_ID;
```

DESCRIPTION

```
mysql> SELECT LGBRAND.Brand_Name, LGBRAND.Brand_Type, AVG(LGPRODUCT.Prod_Price) AS 'Average', SUM(LGLINE.Line_Qty) AS 'Total Units Sold'
    -> FROM LGBRAND JOIN (LGPRODUCT JOIN LGLINE ON LGPRODUCT.Prod_SKU=LGLINE.Prod_SKU) ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
    -> GROUP BY LGBRAND.Brand_ID;
+--------------------+------------+------------+------------------+
| Brand_Name         | Brand_Type | Average    | Total Units Sold |
+--------------------+------------+------------+------------------+
| Light Sabers       | Premium    | 250.000000 |               10 |
| Tatooine Dust      | Regular    | 400.000000 |                3 |
| Coruscant Painters | Premium    | 300.000000 |                6 |
| Racer Pod Ballers  | Regular    | 310.000000 |               16 |
+--------------------+------------+------------+------------------+
4 rows in set (0.00 sec)
```

The query displays the columns Brand_Name and Brand_Type from the table LGBRAND and the columns called 'Average' and 'Total Units Sold'. The 'Average' column is the average of the Prod_Price from the table LGPRODUCT. The 'Total Units Sold' column is the sum of the Line_Qty from the table LGLINE. The tables LGBRAND and LGPRODUCT have been joined on the attribute (primary and foreign key) Brand_ID. The tables LGPRODUCT and LGLINE have been joined on the attribute (primary and foreign key) Prod_SKU. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays the sales data of the products in the database grouped by the Brand_ID.

QUESTION 35

SQL

```
SELECT LGBRAND.Brand_Name, LGBRAND.Brand_Type, LGPRODUCT.Prod_SKU,
   LGPRODUCT.Prod_Desc, LGPRODUCT.Prod_Price
FROM LGBRAND JOIN LGPRODUCT ON LGPRODUCT.Brand_ID = LGBRAND.Brand_ID
WHERE LGBRAND.Brand_Type NOT LIKE 'Premium' AND LGPRODUCT.Prod_Price > (
   SELECT MAX(LGPRODUCT.Prod_Price)
   FROM LGPRODUCT JOIN LGBRAND ON LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
   WHERE LGBRAND.Brand_Type LIKE 'Premium');
```

DESCRIPTION

```
mysql> select LGBRAND.Brand_Name, LGBRAND.Brand_Type, LGPRODUCT.Prod_SKU, LGPRODUCT.Prod_Desc, LGPRODUCT.Prod_Price
    -> from LGBRAND join LGPRODUCT on LGPRODUCT.Brand_ID = LGBRAND.Brand_ID
    -> where LGBRAND.Brand_Type not like 'Premium' and LGPRODUCT.Prod_Price > (
    -> select Max(LGPRODUCT.Prod_Price)
    -> from LGPRODUCT join LGBRAND on LGBRAND.Brand_ID=LGPRODUCT.Brand_ID
    -> where LGBRAND.Brand_Type like 'Premium')
    -> ;
+-------------------+------------+----------+-----------------+------------+
| Brand_Name        | Brand_Type | Prod_SKU | Prod_Desc       | Prod_Price |
+-------------------+------------+----------+-----------------+------------+
| Racer Pod Ballers | Regular    |        8 | Almost finished |     999.99 |
+-------------------+------------+----------+-----------------+------------+
1 row in set (0.00 sec)
```

The query displays the columns Brand_Name and Brand_Type from the table LGBRAND and Prod_SKU, Prod_Desc and Prod_Price from the table LGPRODUCT. The tables LGBRAND and LGPRODUCT have been joined on the attribute (primary and foreign key) Brand_ID. The join statement selects all the rows, from both tables which have been joined where there is a match between both of them. The query displays product data on products which are not a premium brand but cost more than the most expensive premium brand products. The where statement has a nested query which displays the product with the maximum product price. Here the tables LGBRAND and LGPRODUCT have been joined on the attribute (primary and foreign key) Brand_ID. The internal where statement checks that the Brand_Type is not premium.