MIS 533 – Database Management for Decision Making

**Lab Check – Simple Queries – Erwin Chege**

Instructions

* Write down the SQL codes according to the requirements given and take a screen shot of the results.
* All submissions should be made through ***Canvas***. Submissions through emails will be accepted ***ONLY*** under the circumstances when unresolvable technical difficulties occur.
* Keep a digital backup of your work and do ***NOT*** modify the backup file after the assignment is due. For any technical reason preventing me from opening/grading your assignment, you need to provide me the backup file with a last modified date earlier than the due date.
* Submit your assignment before the due date. Late submissions will be accepted with a penalty. A 10% grade will be deducted from your assignment score for each 24-hour increment overdue.
* All submissions should be in ***electronic format***. No paper-based submissions will be accepted.
* This is an ***individual*** assignment.

Problems

Please write queries based on the following requirements using DML In-Class dataset (SQLDataSet.sql). ***For each question, you are required to submit 1) SQL query code; 2) a screenshot of your query result. You should copy and paste your SQL query code to the Word document instead of taking a screenshot of your code. Missing either part for each question will result in a 0 for this question.***

1. List unique item classes stored in my database.

|  |  |
| --- | --- |
| 1) | SELECT DISTINCT item\_class  FROM part; |
|  | A screenshot of a computer  Description automatically generated |

1. List the warehouse number and the number of different parts stored in each warehouse, only include those warehouse with more than 2 different parts.

|  |  |
| --- | --- |
| 2 | SELECT DISTINCT warehouse\_number, COUNT(DISTINCT part\_number)  FROM part  GROUP BY warehouse\_number  HAVING COUNT(DISTINCT part\_number) > 2; |
|  | A screenshot of a computer  Description automatically generated |

1. List the warehouse number and the maximum number of units on hand for parts stored in each of the warehouse. Rename the new column Max\_UOH.

|  |  |
| --- | --- |
| 3 | SELECT warehouse\_number, MAX(units\_on\_hand) AS Max\_UOH  FROM part  GROUP BY warehouse\_number; |
|  | A screenshot of a number  Description automatically generated |

1. List the part number, total dollar amount for each part stored, and the warehouse number for the part. Rename the calculated column TOTAL\_AMOUNT. TOTAL\_AMOUNT=Units\_ON\_HAND\*UNIT\_PRICE

|  |  |
| --- | --- |
| 4 | SELECT part\_number,  (units\_on\_hand \* unit\_price) AS TOTAL\_AMOUNT,  warehouse\_number  FROM part; |
|  | A screenshot of a graph  Description automatically generated |

1. List the class and total number of units on hand for each class. Rank your results in descending order on the total number of units on hand.

|  |  |
| --- | --- |
| 5 | SELECT item\_class,  SUM(units\_on\_hand) AS total\_units\_on\_hand  FROM part  GROUP BY item\_class  ORDER BY total\_units\_on\_hand DESC; |
|  | A screenshot of a computer  Description automatically generated |

1. List stored information for all the orders placed between August 3rd and August 6th of 2013. (not including August 3rd 2013 and August 6th 2013).

|  |  |
| --- | --- |
| 6 | SELECT \*  FROM orders  WHERE order\_date > TO\_DATE('2013-08-03', 'yyyy-mm-dd')  AND order\_date < TO\_DATE('2013-08-06', 'yyyy-mm-dd'); |
|  | A screenshot of a computer  Description automatically generated |

1. List all the order date and the number of orders placed on each date. Rank your results in descending order on the number of orders.

|  |  |
| --- | --- |
| 7 | SELECT order\_date, COUNT(\*) AS num\_orders  FROM orders  GROUP BY order\_date  ORDER BY num\_orders DESC; |
|  | A screenshot of a computer  Description automatically generated |

1. List the warehouse number and the average unit price for warehouses other than number 2.

|  |  |
| --- | --- |
| 8 | SELECT warehouse\_number, AVG(unit\_price) AS average\_unit\_price  FROM part  WHERE warehouse\_number <> 2  GROUP BY warehouse\_number; |
|  | A screenshot of a computer  Description automatically generated |

1. List the stored part information for SG parts stored in warehouse 2 or 3.

|  |  |
| --- | --- |
| 9 | SELECT \*  FROM part  WHERE item\_class = 'SG' AND warehouse\_number IN (2, 3); |
|  | A screen shot of a computer  Description automatically generated |

1. List the part number and part description for each part with letter D (or d) in the part\_description. Rank your results in ascending order on item class and then descending order on units\_on\_hand.

|  |  |
| --- | --- |
| 10 | SELECT part\_number, part\_description  FROM part  WHERE LOWER(part\_description) LIKE '%d%'  ORDER BY item\_class ASC, units\_on\_hand DESC; |
|  | A screenshot of a computer  Description automatically generated |

1. List the average unit price among all the parts. Rename the column as AVG\_UNIT\_PRICE

|  |  |
| --- | --- |
| 11 | SELECT AVG(unit\_price) AS AVG\_UNIT\_PRICE  FROM part; |
|  |  |

1. List the part number and part description for each part with nine characters or ten characters in the part\_description. Rank your results in ascending order on part number.

|  |  |
| --- | --- |
| 12 | SELECT part\_number, part\_description  FROM part  WHERE LENGTH(part\_description) IN (9, 10)  ORDER BY part\_number ASC; |
|  | A screenshot of a black and white screen  Description automatically generated |

1. Find the item classes and the number of parts under each class. Only show those classes with at least 3 types of parts.

|  |  |
| --- | --- |
| 13 | SELECT item\_class, COUNT(DISTINCT part\_number) AS num\_parts  FROM part  GROUP BY item\_class  HAVING COUNT(DISTINCT part\_number) >= 3; |
|  | A screenshot of a computer  Description automatically generated |

1. List the stored part information for SG parts or parts with more than 60 units stored in warehouse 3.

|  |  |
| --- | --- |
| 14 | SELECT \*  FROM part  WHERE (item\_class = 'SG' OR units\_on\_hand > 60) AND warehouse\_number = 3; |
|  | A screenshot of a computer  Description automatically generated |

1. List the number of different part stored in each warehouse, only include those warehouse with at most 3 parts.

|  |  |
| --- | --- |
| 15 | SELECT warehouse\_number, COUNT(DISTINCT part\_number) AS num\_parts  FROM part  GROUP BY warehouse\_number  HAVING COUNT(DISTINCT part\_number) <= 3; |
|  | A screenshot of a computer  Description automatically generated |