A1-20171-SQL-DBS301

DUE DATE: Nothing will be accepted after this date. It will get a zero

**Due BEFORE MIDNIGHT on the FRIDAY of WEEK 6**

**PLAGIARISM**   
When you submit an assignment, you are saying that the submission is your own work and as such you wish to be given credit for the work.

Occasionally, especially when working on a design problem or writing programs (but never on exams or tests!), it may be necessary to ask someone for a small amount of help. You are permitted to do so, provided you meet the following two conditions.

1 You acknowledge the help on the work you hand in, including explaining the work or portion of the work done or assisted by another person.

2 You understand the work that you hand in, so that you could explain the reasoning behind the parts of the work done for you or assisted by another.

Any other assistance by another person constitutes a violation called plagiarism and will be treated as such.

When 2 or more of you work together as a group, only one assignment should be submitted with all the members’ names on it at the time of submission. If two submissions exist, that are essentially the same, then both submissions cannot possibly make the claim that the work is "solely their own". This is a serious matter and is considered plagiarism. It is recorded in accordance with College policy and may result in a penalty such as expulsion from the College. Please do not put yourself in this position. Plagiarism is very easy to detect.

**Anyone not in this list cannot be added to the group later**

Name(s) Student ID(s) Section A, B or C

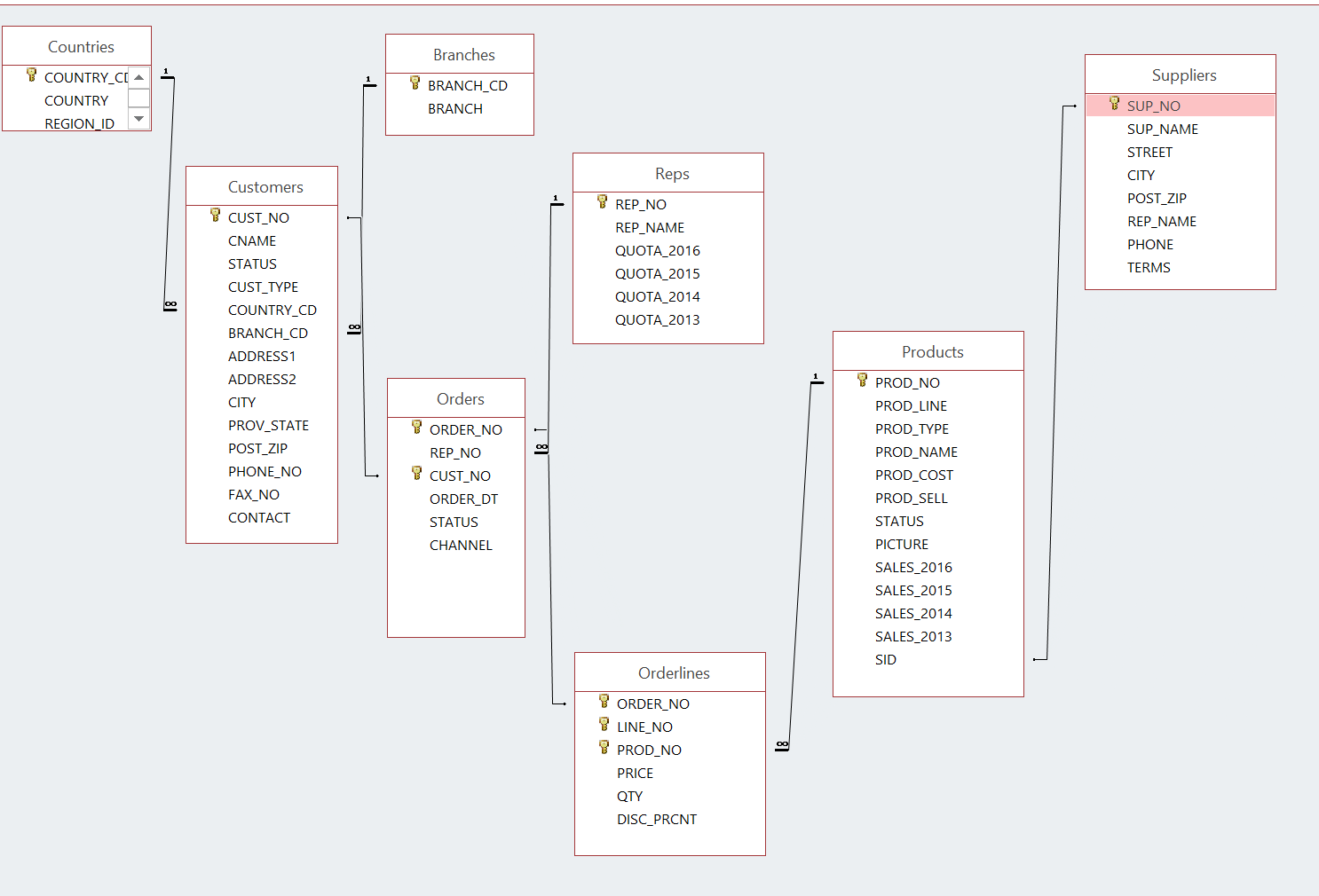
The above acknowledges you understand.

INSTRUCTION:

1 Rename this WORD file A1-xxxxx where xxxxx is replaced with your email account name, not the whole email. If you are in a group pick one of them.

2 When finished email this back to me and CC everyone in your group as proof it was sent.

3 The following tables are used for this assignment. You will be given a script to load.



NOTE:

If any output goes on for more than 100 lines, only cut and paste the first 100 or so. I don’t want to get this word document too big.

1. Display the customer number, customer name and country code for all the customers that are in SPAIN. The country code for Spain is SPA. Please note that you are given SPA, or spa or SpA to use and not Spain.

**SQL:**

**SELECT CUST\_NO, CNAME, COUNTRY\_CD**

**FROM CUSTOMERS**

**WHERE UPPER(COUNTRY\_CD) = 'SPA';**

**OUTPUT:**

**CUST\_NO CNAME COUNTRY\_CD**

**---------- ------------------------------------- ---------**

**1019 Supremax Montagna 5 SPA**

**1035 Ultra Sports 1 SPA**

**1095 Supremax Montagna 1 SPA**

**1096 Supremax Montagna 2 SPA**

**1097 Supremax Montagna 3 SPA**

**1098 GO Outlet Madrid SPA**

**6 rows selected**

2. How many orders have the product number 40302?

**SQL**

**SELECT count(ORDER\_NO)**

**FROM ORDERLINES**

**WHERE PROD\_NO = 40302;**

**OUTPUT:**

**COUNT(ORDER\_NO)**

**---------------------------------------**

**22**

3. List the customer number, customer name and order number for customers that ordered product 40302. Put result in customer number order.

**SQL:**

**SELECT c.CUST\_NO, c.CNAME , o.ORDER\_NO**

**FROM CUSTOMERS c**

**INNER JOIN ORDERS o**

**on c.CUST\_NO = o.CUST\_NO**

**INNER JOIN ORDERLINES l**

**on o.ORDER\_NO = l.ORDER\_NO**

**WHERE l.PROD\_NO = 40302**

**ORDER BY c.CUST\_NO;**

**OUTPUT:**

**CUST\_NO CNAME ORDER\_NO  
---------- ------------------------------ ----------  
 1001 GO Outlet Montreal 67  
 1005 GO Outlet Boston 153  
 1019 Supremax Montagna 5 78  
 1036 Ultra Sports 2 11  
 1039 Vacation Central 1 65  
 1041 Vacation Central 3 33  
 1045 Mountain Madness 3 16  
 1051 Sportwaren G.m.b.H. 1 139  
 1062 123 Fitness PTE Ltd 38  
 1066 Wilderness Wonderment Ltd 164  
 1069 Andes Camping Supplies 3 157  
  
 CUST\_NO CNAME ORDER\_NO  
---------- ------------------------------ ----------  
 1071 Lookout Below Ltd 37  
 1078 Act'N'Up Fitness 4 44  
 1083 Over the Top Cycles 1 77  
 1095 Supremax Montagna 1 205  
 1102 Pro Form Supplies 4 28  
 1121 GO Outlet Manchester 15  
 1127 Fresh Air Co 1 135  
 1129 Fresh Air Co 3 101  
 1130 Fresh Air Lte 4 120  
 1139 Fredies Sport Whse 1 194  
 1148 Juan's Sports 2 2  
  
 22 rows selected**

4. Display the customer number for Ultra Sports 5.

**SQL:**

**SELECT CUST\_NO**

**FROM CUSTOMERS**

**WHERE CNAME = 'Ultra Sports 5';**

**OUTPUT:**

**CUST\_NO**

**-----------------**

**1002**

5. Display all orders for United Kingdom. The word entered is United Kingdom and not UK. Show only cities that start with L.

Display the customer number, customer name, order number, product name, the total dollars for that line. Give that last column the name of TOTAL.

Put the output into customer number order from highest to lowest and display only order numbers less than 75

**SQL:**

**SELECT C.CUST\_NO, C.CNAME, O.ORDER\_NO, P.PROD\_NAME,**

**OL.PRICE\*(1 - OL.DISC\_PERC/100)\*OL.QTY AS TOTAL**

**FROM CUSTOMERS C**

**INNER JOIN ORDERS O**

**ON C.CUST\_NO = O.CUST\_NO**

**INNER JOIN ORDERLINES OL**

**ON O.ORDER\_NO = OL.ORDER\_NO**

**INNER JOIN PRODUCTS P**

**ON OL.PROD\_NO = P.PROD\_NO**

**WHERE C.CITY LIKE 'L%' AND O.ORDER\_NO < 75**

**ORDER BY CUST\_NO DESC;**

**OUTPUT:**

**CUST\_NO CNAME ORDER\_NO PROD\_NAME TOTAL**

**-------------------- ------------------------------ ---------------------- ---------------------- ----------------------**

**1120 GO Outlet London 61 Star Lite 5441.7**

**1120 GO Outlet London 49 Pocket Water Filter 3364.2**

**1120 GO Outlet London 29 MoonGlow 1960.8**

**1120 GO Outlet London 49 Day Tripper 11.2**

**1120 GO Outlet London 49 Pack n' Hike 214.84**

**1120 GO Outlet London 49 GO Cookset 1194.48**

**1120 GO Outlet London 29 GO Cookset 1300.32**

**1120 GO Outlet London 61 GO Camp Kettle 1395.03**

**1120 GO Outlet London 49 GO Headband 288**

**1120 GO Outlet London 61 GO Wristband 334.08**

**1120 GO Outlet London 29 GO Wristband 499.2**

**1120 GO Outlet London 29 Pocket U.V. Alerter 552.42**

**1120 GO Outlet London 49 RiverKind Shampoo 424.08**

**1120 GO Outlet London 29 RiverKind Detergent 1769.88**

**1120 GO Outlet London 61 Pro-Lite Water Filter 141.9**

**1120 GO Outlet London 61 Pocket Water Filter 8202.6**

**1120 GO Outlet London 61 MoonGlow 1425.45**

**17 rows selected**

6 Display a count of how many different country codes there are

**SQL:**

**SELECT COUNT(DISTINCT COUNTRY\_CD)**

**FROM CUSTOMERS;**

**OUTPUT:**

**COUNT(DISTINCTCOUNTRY\_CD)**

**---------------------------------------**

**14**

7. Find the total dollar value for all orders from London. Each row will show customer name, order number and total dollars for the order. Sort by order number

**SQL:**

**SELECT c.CNAME , o.ORDER\_NO ,sum(l.PRICE \* l.QTY) as TOTAL\_DOLLAR**

**FROM CUSTOMERS c INNER JOIN ORDERS o**

**on c.CUST\_NO = o.CUST\_NO**

**INNER JOIN ORDERLINES l**

**on o.ORDER\_NO = l.ORDER\_NO**

**WHERE c.CITY = 'London'**

**GROUP BY c.CNAME, o.ORDER\_NO**

**ORDER BY o.ORDER\_NO;**

**OUTPUT:**

**CNAME ORDER\_NO TOTAL\_DOLLAR**

**------------------------------ ---------- ---------------------------------------**

**GO Outlet London 29 6580**

**GO Outlet London 49 6446**

**GO Outlet London 61 17739**

**GO Outlet London 107 28466**

**Trees to Seas Ltd 122 5056**

**Trees to Seas Ltd 170 5728**

**6 rows selected**

**Going back to the same tables you have used for labs that came from demobld10g**

8 Display the (a) employee number, (b) full employee name, (c) job and (d) hire date.

- Limit the display to all employees hired in May, June, July, August or Dec

- The most recently hired employees are displayed first.

- Exclude people hired in 1992 to 1996

- Full name should be in the form à *Lastname, Firstname --* with an alias called *Full Name.*

- Hire date should point to the last day in May, June, July, August or December of that year (NOT to the exact hire date)

- The format is in the form of *May 31st of 1997* –better if there is no big gap between month and 31st

- The hire date column should be called *Start Date*.

**NOTE: Do NOT use a LIKE operator.**

You should display ONE row per output line by limiting the width of the *Full Name* to 25 characters.

**SQL:**

**SELECT EMPLOYEE\_ID,**

**SUBSTR((LAST\_NAME || ', ' || FIRST\_NAME),0,25) AS "Full Name", JOB\_ID, TO\_CHAR(ROUND(LAST\_DAY(HIRE\_DATE)), 'FMMonth ddth "of "yyyy') AS "Start Date"**

**FROM EMPLOYEES**

**WHERE TO\_CHAR(HIRE\_DATE, 'YY') NOT IN(92, 93 ,94, 95, 96)**

**AND TO\_CHAR(HIRE\_DATE, 'MM') IN (05,06,07,08,12)**

**ORDER BY HIRE\_DATE DESC;**

**OUTPUT:**

**EMPLOYEE\_ID Full Name JOB\_ID Start Date**

**----------- ------------------------- ---------- -------------------------------------------------**

**178 Grant, Kimberely SA\_REP May 31st of 1999**

**144 Vargas, Peter ST\_CLERK July 31st of 1998**

**202 Fay, Pat MK\_REP August 31st of 1997**

**104 Ernst, Bruce IT\_PROG May 31st of 1991**

**100 King, Steven AD\_PRES June 30th of 1987**

**5 rows selected**

9. List the employee number, full name, job and the modified salary for all employees

- whose monthly earning (without the increase) is outside the range $6,000 – $11,000

- and who are employed as a Vice Presidents or Managers (President is not counted here).

- You should use **Wild Card** characters for this.

- the modified salary for a VP will be 30% higher

- and managers a 20% salary increase.

- Sort the output by the top salaries (before this increase).

Heading will be: → *Employees with Increased Pay*

**The output lines should look like this sample line:**

Employee 101 named Neena Kochhar with Job ID of AD\_VP will have a new salary of $22100

**SQL:**

**SELECT 'Emp# ' || EMPLOYEE\_ID || ' named ' || FIRST\_NAME || ' ' || LAST\_NAME || ' who is ' || JOB\_ID || ' will have a new salary of $' ||**

**CASE**

**WHEN UPPER(JOB\_ID) LIKE '%VP' THEN SALARY\*1.3**

**WHEN UPPER(JOB\_ID) LIKE '%MAN' THEN SALARY\*1.2**

**WHEN UPPER(JOB\_ID) LIKE '%MGR' THEN SALARY\*1.2**

**ELSE SALARY**

**END AS "Employees with increased Pay"**

**FROM EMPLOYEES**

**WHERE (SALARY < 6000 OR SALARY > 11000)**

**AND (JOB\_ID LIKE '%VP' OR JOB\_ID LIKE '%MAN' OR JOB\_ID LIKE '%MGR')**

**ORDER BY SALARY DESC;**

**OUTPUT:**

**Employees with increased Pay**

**---------------------------------------------------------------------------------------------------------------------------------------Emp# 101 named Neena Kochhar who is AD\_VP will have a new salary of $22100**

**Emp# 102 named Lex De Haan who is AD\_VP will have a new salary of $22100**

**Emp# 201 named Michael Hartstein who is MK\_MAN will have a new salary of $15600**

**Emp# 205 named Shelley Higgins who is AC\_MGR will have a new salary of $14400**

**Emp# 124 named Kevin Mourgos who is ST\_MAN will have a new salary of $6960**

**5 rows selected**

10. Display last\_name, job id and salary for all employees who earn more than all lowest paid employees per department that are in locations outside the US.

Exclude President and Vice Presidents from this query.

Sort the output by job id ascending.

If a JOIN is needed you must use a “newer” method (USING/JOIN)

**SQL:**

**SELECT  
 LAST\_NAME,  
 SALARY,  
 JOB\_ID  
FROM EMPLOYEES   
WHERE SALARY > ALL (SELECT  
 MIN(SALARY)  
 FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT\_ID)  
 JOIN LOCATIONS USING (LOCATION\_ID)  
 WHERE COUNTRY\_ID != 'US'  
 GROUP BY DEPARTMENT\_ID)  
AND JOB\_ID NOT IN ('AD\_PRES','AD\_VP')  
ORDER BY JOB\_ID;**

**OUTPUT:**

**LAST\_NAME SALARY JOB\_ID**

**------------------------- ---------- ----------**

**Higgins 12000 AC\_MGR**

**Hunold 9000 IT\_PROG**

**Hartstein 13000 MK\_MAN**

**Zlotkey 10500 SA\_MAN**

**Abel 11000 SA\_REP**

11. Who are the employees (show last\_name, salary and job) who work either in IT , ACCOUNTING or MARKETING department and earn more than the worst paid person in the SHIPPING department.

Sort the output by the last name alphabetically.

**You need to use ONLY the Subquery method (NO joins allowed).**

**SQL:**

**select last\_name, salary , job\_id**

**from employees**

**where (department\_id = (select department\_id from departments where UPPER(department\_name) = 'IT')**

**or department\_id = (select department\_id from departments where UPPER(department\_name) = 'MARKETING')**

**or department\_id = (select department\_id from departments where UPPER(department\_name) = 'ACCOUNTING'))**

**and salary > (select min(salary) from employees where department\_id = (select department\_id from departments where UPPER(department\_name) = 'SHIPPING'))**

**order by last\_name;**

**OUTPUT:**

**LAST\_NAME SALARY JOB\_ID**

**------------------------- ---------------------- ----------**

**Ernst 6000 IT\_PROG**

**Fay 6000 MK\_REP**

**Gietz 8300 AC\_ACCOUNT**

**Hartstein 13000 MK\_MAN**

**Higgins 12000 AC\_MGR**

**Hunold 9000 IT\_PROG**

**Lorentz 4200 IT\_PROG**

**7 rows selected**

12. Display Department\_id, Job\_id and the Lowest salary for this combination but only if that Lowest Pay falls in the range $6000 - $18000.

Exclude people who

(a) work as some kind of *Representative* job from this query and

(b) departments IT and SALES

Sort the output according to the Department\_id and then by Job\_id.

You MUST NOT use the Subquery method.

**SQL:**

**select e.department\_id, e.job\_id, min(e.salary)**

**from employees e join departments d on e.department\_id = d.department\_id group by e.department\_id, e.job\_id**

**having (min(e.salary) between 6000 and 18000)**

**and d.DEPARTMENT\_NAME not in ('Sales', 'IT')**

**and e.job\_id not like '%REP'**

**and e.job\_id not like '%PRES'**

**order by e.department\_id , e.job\_id;**

**OUTPUT:**

**DEPARTMENT\_ID JOB\_ID MIN(E.SALARY)**

**---------------------- ---------- ----------------------**

**20 MK\_MAN 13000**

**90 AD\_VP 17000**

**110 AC\_ACCOUNT 8300**

**110 AC\_MGR 12000**