**HOMEWORK 1**

**What you need to submit in your assignment document (MS Word or PDF):**

**For every question submit:**

**1. SQL query**

**2. SCREENSHOT OF THE OUTPUT (wrong screenshots or missing screenshots will result in deduction of 50% of the points)**

**3. if the number of rows in result set exceeds 15, limit your results to the first 15 rows only**

1. (20 pts) Write a select statement that return 6 columns from invoices table named, vendor id, invoice number, due date, invoice total, %15, and Plus %15. These columns should contain the following details:
   1. Vendor ID: vendor\_id
   2. Invoice Number: invoice\_number
   3. Due Date: Invoice\_due\_date
   4. 15%: 15% percent of the invoice\_total
   5. Plus 15%: invoice\_total + the 15% of invoice\_total

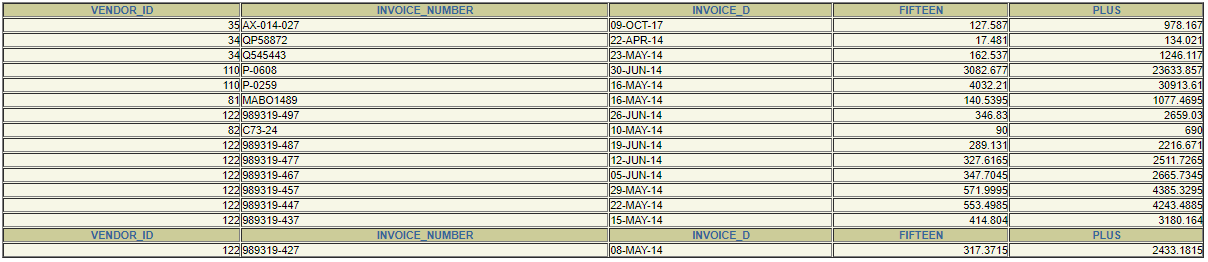
Ex. If invoice\_total is 1000 then the 15% should be (1000\*0.15)=150

“select vendor\_id, invoice\_number, invoice\_due\_date,

(invoices.invoice\_total \* .15) as fifteen,

(invoices.invoice\_total + (invoice\_total \* .15)) as plus

from invoices where rownum <= 15;”



1. (20pts) Filter the results of the previous question to only show those records with an invoice\_total that’s greater than or equal to 500 and less than or equal to 1000. Then, sort the result set in descending order by invoice\_due\_date.

“select vendor\_id, invoice\_number, invoice\_due\_date,

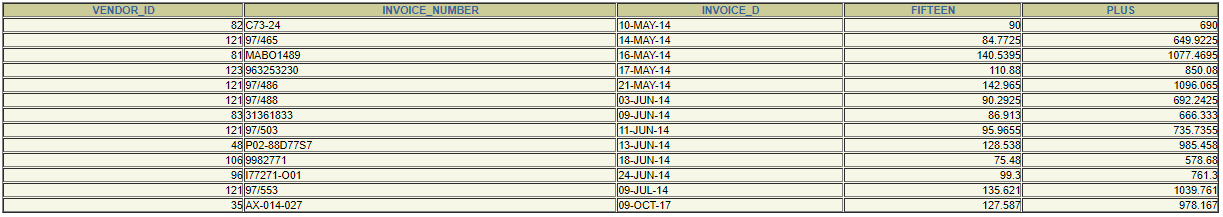
(invoice\_total \* .15) as fifteen,

(invoice\_total + (invoice\_total \* .15)) as plus

from invoices

where invoice\_total <= 1000 and invoice\_total >= 500

order by invoice\_due\_date;”



1. (20pts) Write a select statement that return 5 columns from three tables:
   1. Vendor name: vendor\_name
   2. Invoice date: invoice\_date
   3. Invoice number: invoice\_number
   4. Invoice sequence line item: invoice\_sequence from the invoice\_line\_item table
   5. Line item amount: line\_item\_amt from the invoice\_line\_item table

Use the following aliases for the tables ‘vd’ for vendors table, ‘inv’ for invoices table, ‘invli’ for invoice\_line\_item table.

Filter the results to only show those vendors who have the abbreviation “Inc” in their name.

Sort your result set by: vendor name, invoice date, and invoice sequence

“select vd.vendor\_name, inv.invoice\_date, inv.invoice\_number,

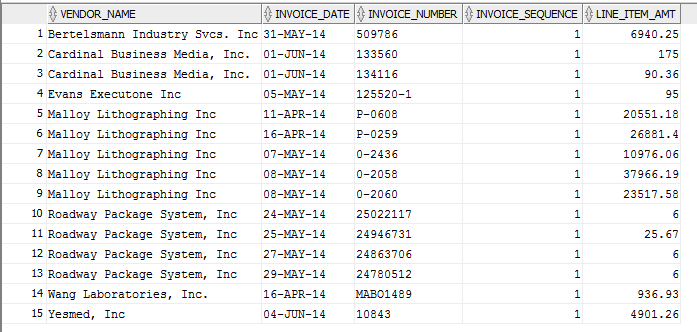
invli.invoice\_sequence,

invli.line\_item\_amt

from invoices inv, vendors vd, invoice\_line\_items invli

where inv.vendor\_id = vd.vendor\_id and invli.invoice\_id = inv.invoice\_id and vd.vendor\_name like '%Inc%'

order by vd.vendor\_name, inv.invoice\_date, inv.invoice\_number;”



1. (20 pts) Write a select statement that return one row for each vendor\_state that contain three columns:
   1. Vendor state: vendor\_state
   2. The count of the vendors in each state
   3. The sum of the invoices total for all invoices for the vendors of each state.

Ex.

|  |  |  |
| --- | --- | --- |
| Vendor state | Vendors count | Total sum of invoices |
| MD | 14 | 36820 |
| .. | .. | .. |
| .. | .. | .. |

Filter the result set to include only those rows with a sum less than 5000.

Then, sort the result set in descending order by the count of vendors.

“

select vd.vendor\_state, count(\*), sum(inv.invoice\_total)

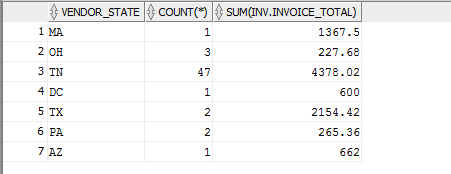
from vendors vd, invoices inv

where vd.vendor\_id = inv.vendor\_id

group by vd.vendor\_state

having sum(inv.invoice\_total) <= 50000;

“



1. (20pts) Write a select statement that return 4 column from invoices table named, vendor id, invoices id, invocie total, invoice category. These columns should contain the following
   1. Vendor id : vendor\_id
   2. Invoice id : invoice\_id
   3. Invoice total: invoice\_total
   4. Invoice category: a case function that categorize the invoice\_total into 4 categories as following:
      1. If invoice total is greater than (3\*stddev(invoice\_total)+avg(invoice\_total)) print ‘HIGH’
      2. If invoice total is greater than (avg(invoice\_total)) then print ‘ ABOVE AVERAGE’
      3. If invoice total is greater than (avg(invoice\_total)/2) then print ‘BELOW AVERAGE’
      4. Else print ‘ low’

Hint :

\*stddev is the function for standard deviation in oracle

\* each condition in the case function should be in subqueries:

Ex. “>(select 3\*stddev(invoice\_total)+avg(invoice\_total) from invoices)”

“

select vendor\_id, invoice\_id, invoice\_total, case

when invoice\_total > (select 3 \* stddev(invoice\_total) + avg(invoice\_total) from invoices) then 'HIGH'

when invoice\_total > (select avg(invoice\_total) from invoices) then 'ABOVE AVERAGE'

when invoice\_total > (select avg(invoice\_total)/2 from invoices) then 'BELOW AVERAGE'

else 'LOW'

end as invoice\_category

from invoices

where rownum <= 15

group by vendor\_id, invoice\_id, invoice\_total;

“

