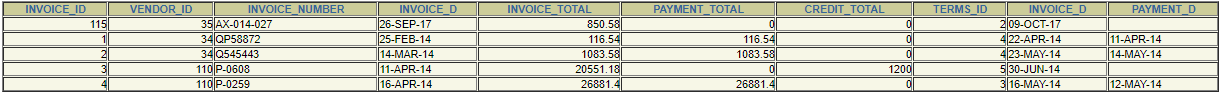
**In-class Exercise**

Write SQL statements for the following:

Submit the code + screenshots of the results (including the view table)

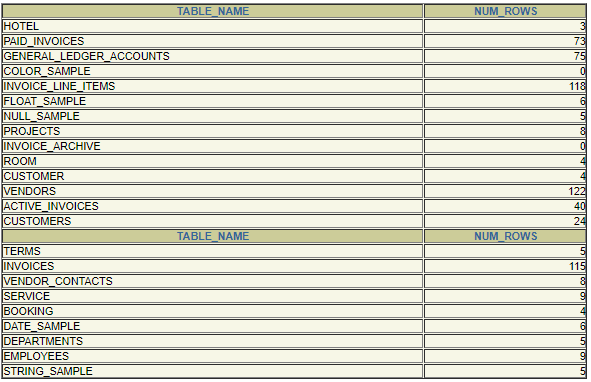
1. Write a SQL query to display the first 5 records in invoices table. (10 points)

“select \* from invoices where rownum <= 5;”



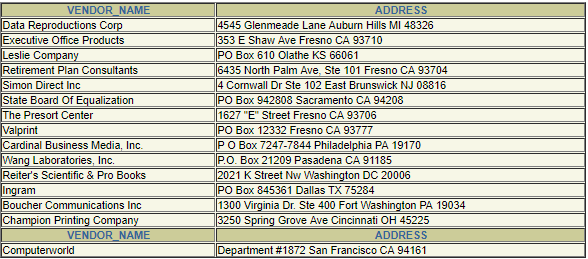
1. Write a SQL query to display all tables in the database. Display the name of the table, and number of rows only. (10 points)

“select distinct table\_name, num\_rows from user\_tables;”



1. Write a SQL query to list vendors and their address (vendor name, full address). The address should be in one column using the following format (“address1, city, state. zipcode”) list the first 15 rows only(20 points)

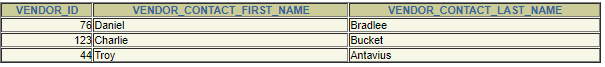
“Select vendor\_name, (vendor\_address1||' '|| vendor\_city ||' '|| vendor\_state ||' '||vendor\_zip\_code) as address from vendors where rownum <= 15;”



1. Write SQL statement to create a view match\_vendor\_name, containing vendor\_id, vendor first name, vendor last name from vendors table. For those vendors that their last name matches the last name in vendor\_contacts table. (20 points)

“ Create or replace view match\_vendor\_name as (select vendors.vendor\_id, vendors.vendor\_contact\_first\_name, vendors.vendor\_contact\_last\_name from vendors, vendor\_contacts where vendors.vendor\_contact\_last\_name = vendor\_contacts.last\_name);”

“select \* from match\_vendor\_name;”



1. Write SQL statement to create a view payed\_vendors, containing the vendor name, the count of invoices, the sum of the total invoices, and the sum of the total payments. For those that have 2 or more invoices. (20 points)

“Create or replace view payed\_vendor as (select (vendors.vendor\_contact\_first\_name||' '|| vendors.vendor\_contact\_last\_name) as vendor\_name, count(\*), sum(invoices.payment\_total), sum(invoices.invoice\_total) from vendors, invoices where vendors.vendor\_id = invoices.vendor\_id);”

“select \* from payed\_vendor;"

1. Create a PL/SQL anonymous block program unit that displays one of the following messages: ‘match’ or ‘not match’. For two numerical variables, if they are the same print ‘match’, otherwise print ‘not match. (20 points) (hint: you can hard-code the values of the two variables)

“set serveroutput on;

Declare

num\_one number(3);

num\_two number(3);

begin

num\_one := 10;

num\_two := 12;

if(num\_one = num\_two) then

dbms\_output.put\_line('Match');

else

dbms\_output.put\_line('Not Match');

end if;

end;”

