Assignment 3

Name ____________Christopher Holmes________________________________

How to retrieve data using multiple tables and perfrom summaries of data

- Read chapter 4 on how to retrieve data from two or more tables using join and union operations
- Do exercises 1-7 at end of chapter 4
- Read chapter 6 on how to code summary queries using aggregation function AVG, SUM, MIN, MAX and COUNT, and use the GROUP BY and HAVING clauses
- Do exercises 1-7 at end of chapter 6
- Read chapter 7 on how to code subqueries
- Do exercies at end of chapter 7
- Read chapter 8 about data types and how to convert or cast data from one data type to another and use the FORMAT and CHAR functions
- Read chapter 9 about how to work with string and date-time data types in SQL statements

---

Copy and paste your working select statement from chapter 4 exercises here.

1-

```
select *
from vendors inner join invoices
   on vendors.vendor_id = invoices.vendor_id
order by invoice_number;
```

2-

```
select vendor_name as V, invoice_number as I, invoice_date, invoice_total - payment_total -
credit_total as balance_due
from vendors inner join invoices
   on vendors.vendor_id = invoices.vendor_id
where invoice_total - payment_total - credit_total > 0
order by vendor_name asc;
```

3-

```
select vendor_name, default_account_number as default_account, account_description as
description
from vendors inner join general_ledger_accounts
          on vendors.default_account_number = general_ledger_accounts.account_number
order by account_description, vendor_name
```

4-

```
select vendor_name, invoice_date, invoice_number,
       invoice_sequence as li_sequence,
       line_item_amount as li_amount
from vendors v join invoices i
  on v.vendor_id = i.vendor_id
 join invoice_line_items li
   on i.invoice_id = li.invoice_id
order by vendor_name, invoice_date, invoice_number, invoice_sequence
```

5-

```
select v1.vendor_id, v1.vendor_name, concat(v1.vendor_contact_first_name, ' ',
v1.vendor_contact_last_name) as contact_name
from vendors v1 join vendors v2
on v1.vendor_id <> v2.vendor_id and
       v1.vendor_contact_last_name = v2.vendor_contact_last_name
order by v1.vendor_contact_last_name
```

6-

```
select g.account_number, account_description, invoice_id
from general_ledger_accounts g left join invoice_line_items i
             on g.account_number = i.account_number
where invoice_id is null
order by g.account_number
```

7-

```
select vendor_name, vendor_state
from vendors
where vendor_state = 'CA'
union
select vendor_name, 'Outside CA'
from vendors
where vendor_state <> 'CA'
order by vendor_name
```

Copy and paste your working select statements from chapter 6 exercises here.

1-

```
select vendor_id, sum(invoice_total) as invoice_total_sum
from invoices
group by vendor_id
```

2-

```
select vendor_name, sum(payment_total) as payment_total_sum
from vendors join invoices
        on vendors.vendor_id = invoices.vendor_id
group by vendor_name
order by payment_total_sum desc
```

3-

```
select vendor_name, count(*) as invoice_count, sum(invoice_total) as invoice_total_sum
from vendors join invoices
        on vendors.vendor_id = invoices.vendor_id
group by vendor_name
order by invoice_count desc
```

4-

```
select account_description, count(*) as count_invoices, sum(line_item_amount) as sum_line
from general_ledger_accounts gl join invoice_line_items il
on gl.account_number = il.account_number
group by gl.account_description
having count(*) > 1
order by sum_line desc
```

5-

```
select account_description, count(*) as count_invoices, sum(line_item_amount) as sum_line
from general_ledger_accounts gl join invoice_line_items il
        on gl.account_number = il.account_number
join invoices i
        on il.invoice_id = i.invoice_id
where invoice_date between '2011/04/01' and '2011/06/30'
group by gl.account_description
having count(*) > 1
order by sum_line desc
```

6-

```
select il.account_number, sum(line_item_amount) as sum_line
from invoice_line_items il join general_ledger_accounts gl
where il.account_number = gl.account_number
group by il.account_number
with rollup
```

7-

```
select vendor_name, count(distinct il.account_number) as 'Number of Accounts'
from vendors v join invoices i
            on v.vendor_id = i.vendor_id
join invoice_line_items il
            on i.invoice_id = il.invoice_id
group by vendor_name
having count(distinct il.account_number)>1
order by vendor_name
```

Copy and paste your working select statement with subqueries from chapter 7 here.

1-

```
select vendor_name
from vendors
where vendor_id in (select vendor_id from invoices)
order by vendor_name
```

2-

```
select invoice_number, invoice_total
from invoices
where payment_total  > (select avg(payment_total) from invoices where payment_total > 0)
order by invoice_total desc
```

3-

```
select account_number, account_description
from general_ledger_accounts
where not exists (select * from invoice_line_items where invoice_line_items.account_number =
general_ledger_accounts.account_number)
order by account_number
```

4-

```
select vendor_name, i.invoice_id, invoice_sequence, line_item_amount
from vendors v join invoices i
            on v.vendor_id = i.vendor_id
join invoice_line_items il
            on i.invoice_id = il.invoice_id
where i.invoice_id in (select invoice_id from invoice_line_items where invoice_sequence > 1)
order by vendor_name, i.invoice_id, invoice_sequence
```

5-

6-

```
select min(v.vendor_name) as vendor_name , v.vendor_city , v.vendor_state
from vendors v
group by v.vendor_city, v.vendor_state
having count(*) = 1
order by vendor_state, vendor_city
```

7-

```
select vendor_name, invoice_number as oldest_invoice, invoice_date, invoice_total
from invoices i join vendors v
on i.vendor_id = v.vendor_id
where invoice_date =
(select min(invoice_date)
from invoices isub
where isub.vendor_id = i.vendor_id)
order by vendor_name
```

8-

```
select vendor_name, invoice_number as oldest_invoice, oi.oldest_invoice_date, invoice_total
from invoices i join
(select vendor_id, min(invoice_date) as oldest_invoice_date
from invoices
group by vendor_id) oi
on (i.vendor_id = oi.vendor_id and
i.invoice_date = oi.oldest_invoice_date)
join vendors v
on i.vendor_id = v.vendor_id
order by vendor_name
```