

Helps for Chapter 3 SQL Assignments

Problem #6

Clauses

- SELECT
- FROM
- WHERE
- ORDER BY *expression* [ASC | DESC]
- LIMIT

Tables (FROM clause)

- Vendors

Joins (FROM clause)

- none

(SELECT clause)	
Columns	Aliases
vendor_name	
vendor_contact_last_name	
vendor_contact_first_name	

Filter (WHERE clause)

- none

Functions (SELECT, FROM, ORDER BY clause)

- none

Sort (ORDER BY clause)

- vendor_contact_last_name, vendor_contact_first_name
- ascending order

Problem #7

Clauses

- SELECT
- FROM
- WHERE
- ORDER BY *expression* [ASC|DESC]
- LIMIT

Tables (FROM clause)

- vendors

Joins (FROM clause)

- none

(SELECT clause)	
Columns	Aliases
vendor_name	AS full_name
vendor_contact_last_name	
vendor_contact_first_name	

Filter (used in WHERE clause)

- last name begins with the letter A, B, C, or E

Functions (SELECT, FROM, ORDER BY clause)

- CONCAT(string1[string2]...) (refer to Murach pg. 86)
Ex., **CONCAT(vendor_contact_last_name, ', ', vendor_contact_first_name)**
- REGEX (refer to Murach pg. 101)
- Ex., **REGEXP '^[A,B,C,E]'**

Sort (ORDER BY clause)

- last_name
- first_name

Problem #8

Clauses

- SELECT
- FROM
- WHERE
- ORDER BY *expression* ASC|DESC
- LIMIT

Tables

- invoices

Joins

- none

(SELECT clause)	
Columns	Aliases
invoice_due_date	AS "Due Date"
invoice_total	As "Invoice Total"
invoice_total * .1	As "10%"
invoice_total + (invoice_total * .1)	AS "Plus 10%"

Filter (used in WHERE clause)

- One way: invoice_total >= 500 AND invoice_total <=1000
- Another way: invoice_total BETWEEN 500 and 1000

Functions

- none

Sort

- invoice_due_date

Problem #9

Clauses

- SELECT
- FROM
- WHERE
- ORDER BY *expression* ASC|DESC
- LIMIT

Tables (FROM clause)

- invoices

Joins (FROM)

- none

(SELECT clause)	
Columns	Aliases
invoice number	
invoice_total	
payment_total + credit_total	AS "payment_credit_total"
Invoice_total – (payment_total + credit_total)	AS "balance_due"

Filter (WHERE clause)

- Invoice_total – (payment_total + credit_total) > 50

Functions (SELECT, WHERE, ORDER BY)

- none

Sort (ORDER BY clause)

- invoice_due_date DESC

LIMIT

- LIMIT 5

Problem #10

Clauses

- SELECT
- FROM
- WHERE
- ORDER BY *expression* ASC|DESC
- LIMIT

Tables (FROM clause)

- invoices

Joins (FROM)

- none

(SELECT clause)	
Columns	Aliases
invoice number	
invoice_date	
invoice_total - payment_total - credit_total	AS "balance_due"
payment_date	

Filter (WHERE clause)

- payment_date IS NULL

Functions (SELECT, WHERE, ORDER BY)

- IS NULL

Sort (ORDER BY clause)

- none

Problem #11

Clauses

- **SELECT**
- FROM
- WHERE
- ORDER BY *expression* ASC|DESC
- LIMIT

Tables (FROM clause)

- none

Joins (FROM)

- none

(SELECT clause)	
Columns	Aliases
DATE_FORMAT(CURRENT_DATE, '%m-%d-%Y')	AS "current_date"

Filter (WHERE clause)

- none

Functions (SELECT, WHERE, ORDER BY)

- CURRENT_DATE – returns today's date
- DATE_FORMAT(date, format_string)
Ex., DATE_FORMAT(CURRENT_DATE, '%m-%d-%Y')

Sort (ORDER BY clause)

- none

Problem #12

Clauses

- **SELECT**
- FROM
- WHERE
- ORDER BY *expression* ASC|DESC
- LIMIT

Tables (FROM clause)

- none

Joins (FROM)

- none

(SELECT clause)	
Columns	Aliases
50000	AS "starting_principle"
50000 * .065	AS "interest"
50000 + (50000 * .065)	AS "principle_plus_interest"

Filter (WHERE clause)

- none

Functions (SELECT, WHERE, ORDER BY)

- none

Sort (ORDER BY clause)

- none