**LAB 4**

**Chapter 8 exercises**

**Exercise 1**:

Write a SELECT statement that returns these columns from the Invoices table:

The invoice\_total column

A column that uses the FORMAT function to return the invoice\_total column with 1 digit

to the right of the decimal point

A column that uses the CONVERT function to return the invoice\_total column as an

integer

A column that uses the CAST function to return the invoice\_total column as an integer

use ap;

SELECT invoice\_total,

FORMAT(invoice\_total, 1) AS total\_format,

CONVERT(invoice\_total, SIGNED) AS total\_convert,

CAST(invoice\_total AS SIGNED) AS total\_cast

FROM invoices;

A screenshot of a computer

Description automatically generated

**Exercise 2:**

Write a SELECT statement that returns these columns from the Invoices table:

The invoice\_date column A column that uses the CAST function to return the invoice\_date column with its full date and time

A column that uses the CAST function to return the invoice\_date column with just the

year and the month

SELECT invoice\_date,

CAST(invoice\_date AS DATETIME) AS invoice\_datetime,

CAST(invoice\_date AS CHAR(7)) AS invoice\_char7

FROM invoices

**A screenshot of a computer

Description automatically generated**

**Chapter 9 exercises**

**Exercise 1**:

Write a SELECT statement that returns these columns from the Invoices table:

The invoice\_total column A column that uses the ROUND function to return the

invoice\_total column with 1 decimal digit

A column that uses the ROUND function to return the invoice\_total column with

no decimal digits

A column that uses the TRUNCATE function to return the invoice\_total column

with no decimal digits

SELECT invoice\_total, ROUND(invoice\_total, 1) AS one\_digit,

ROUND(invoice\_total, 0) AS zero\_digits\_round,

TRUNCATE(invoice\_total, 0) AS zero\_digits\_truncate

FROM invoices

A screenshot of a computer

Description automatically generated

**Exercise 2**:

Write a SELECT statement that returns these columns from the Date\_Sample table in the **EX database**:

The start\_date column

A column that uses the DATE\_FORMAT function to return the start\_date column

with its month name abbreviated and its month, day, and two-digit year separated by slashes

A column that uses the DATE\_FORMAT function to return the start\_date column

with its month and day returned as integers with no leading zeros, a two-digit

year, and all date parts separated by slashes

A column that uses the DATE\_FORMAT function to return the start\_date column

with only the hours and minutes on a 12-hour clock with an am/pm indicator

SELECT start\_date,

DATE\_FORMAT(start\_date, '%b/%d/%y') AS format1,

DATE\_FORMAT(start\_date, '%c/%e/%y') AS format2,

DATE\_FORMAT(start\_date, '%l:%i %p') AS twelve\_hour,

DATE\_FORMAT(start\_date, '%c/%e/%y %l:%i %p') AS format3

FROM date\_sample

A screenshot of a computer

Description automatically generated

**Exercise 3:**

Write a SELECT statement that returns these columns from the Invoices table:

The invoice\_number column

The invoice\_date column

The invoice\_date column plus 30 days

The payment\_date column

A column named days\_to\_pay that shows the number of days between the

invoice date and the payment date

The number of the invoice date’s month The four-digit year of the invoice date

When you have this working, add a WHERE clause that retrieves just the invoices for the month of May based on the invoice date, not the number of the invoice month.

SELECT invoice\_number,

invoice\_date,

DATE\_ADD(invoice\_date, INTERVAL 30 DAY) AS date\_plus\_30\_days,

payment\_date,

DATEDIFF(payment\_date, invoice\_date) AS days\_to\_pay,

MONTH(invoice\_date) AS "month",

YEAR(invoice\_date) AS "year"

FROM invoices

WHERE invoice\_date > '2018-04-30' AND invoice\_date < '2018-06-01'

A screenshot of a computer

Description automatically generated