1. **Write a SELECT statement that shows invoice\_number and invoice\_total from the invoices table.**

USE ap;

SELECT invoice\_number, invoice\_total

FROM invoices

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

1. **Write a SELECT statement that shows invoice\_id and invoice\_total where the payment is made past the due date.**

Table

Description automatically generatedUSE ap;

SELECT invoice\_id, invoice\_total

FROM invoices

WHERE payment\_date > invoice\_due\_date

Graphical user interface

Description automatically generated

**3. Write a SELECT statement that shows invoice\_number, invoice\_date, and invoice\_total where the payment is made on or before July 2018.**

USE ap;

SELECT invoice\_number, invoice\_date, invoice\_total

FROM invoices

WHERE payment\_date <= '2018-07-31'

Graphical user interface, text, application, table

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

1. **Write a SELECT statement that shows invoice\_id, vendor\_id, and invoice\_total where the payment is not made. Use the LIMIT clause so the result set contains only the rows with the 5 largest invoice\_total.**

USE ap;

SELECT invoice\_id, vendor\_id, invoice\_total

FROM invoices

WHERE payment\_date is null

ORDER BY invoice\_total DESC

LIMIT 5

Table

Description automatically generated



**5. Write a SELECT statement that shows invoice\_id, invoice\_total, and payment\_total where the payment is made partial, that is, the payment is made but not made in full.**

USE ap;

SELECT invoice\_id, invoice\_total, payment\_total

FROM invoices

WHERE payment\_total != 0

AND invoice\_total - payment\_total != 0

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

**6. Write a SELECT statement that returns invoice\_number, invoice\_date, balance\_due, and payment\_date from the Invoices table. The balance\_due is the invoice\_total column minus the payment\_total and credit\_total columns. Return only the rows where the payment\_date column contains a null value.**

USE ap;

SELECT invoice\_number, invoice\_date, invoice\_total - payment\_total - credit\_total

AS balance\_due, payment\_date

FROM invoices

WHERE payment\_Date IS Null

Text

Description automatically generated with medium confidenceTable, Excel

Description automatically generated

**7. Write a SELECT statement to show the invoice number, invoice\_total, and balance due where the balance due is greater than 100. Sort the result by the balance due in descending order.**

USE ap;

SELECT invoice\_number, invoice\_total, invoice\_total - payment\_total - credit\_total

AS balance\_due

FROM invoices

WHERE invoice\_total - payment\_total - credit\_total > 100

ORDER BY balance\_due DESC

Table

Description automatically generated



**8. Write a SELECT statement without a FROM clause that uses the CURRENT\_DATE function to return the current date in DD-Mon-YYYY format. Use the DATE\_FORMAT function. This displays the day, month, and four‐digit year of the current date. Give this column an alias of ‘Current Date’.**

USE ap;

SELECT DATE\_FORMAT(CURRENT\_DATE, '%d-%b-%y') AS 'Current Date'

Graphical user interface, application, table

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

9. **Write a SELECT statement without a FROM clause that creates a row with these columns:**

**starting\_principal Starting principal of $200,000**

**interest 2.5% of the principal**

**principal\_plus\_interest The principal plus the interest**

**To calculate the third column, add the expressions you used for the first two columns.**

USE ap;

SELECT 200000

AS starting\_principal, (.025 \* 200000)

AS interest, (200000 + (.025\*200000))

AS principal\_plus\_interest

Graphical user interface

Description automatically generated with medium confidence



**10. Write a SELECT statement to show all the states where vendors are located from the vendors**

**table. Avoid duplicate state names.**

USE ap;

SELECT DISTINCT vendor\_state

FROM vendors

Table

Description automatically generated with medium confidence

**11. Write a SELECT statement that returns vendor\_name, vendor\_contact\_last\_name, and vendor\_contact\_first\_name from the vendors table, with the result ordered by last name and then first name.**

USE ap;

SELECT vendor\_name, vendor\_contact\_last\_name, vendor\_contact\_first\_name

FROM vendors

ORDER BY vendor\_contact\_last\_name, vendor\_contact\_first\_name

Table

Description automatically generated



**12. Write a SELECT statement to show the vendor name, vendor city, vendor state, and vendor zipcode where the vendor zip code starts with 9.**

USE ap;

SELECT vendor\_name, vendor\_city, vendor\_state, vendor\_zip\_code

FROM vendors

WHERE vendor\_zip\_code LIKE '9%'

Graphical user interface, application

Description automatically generated



**13. Write a SELECT statement to show the vendor name and vendor city where there is no phone number for the vendor.**

USE ap;

SELECT vendor\_name, vendor\_city

FROM vendors

WHERE vendor\_phone IS null

Graphical user interface, application

Description automatically generated



**14. Write a SELECT statement that returns one column from the Vendors table named full\_name that joins the vendor\_contact\_first\_name and vendor\_contact\_last\_name columns. Format this column with the first name, a space, and the last name like this: “Jane Doe”Sort the result set by last name and then first name. Return only the contacts whose last name starts with A through H.**

USE ap;

SELECT CONCAT(vendor\_contact\_first\_name, ' ', vendor\_contact\_last\_name)

AS full\_name

FROM vendors

WHERE vendor\_contact\_last\_name >= 'A'

AND vendor\_contact\_last\_name < 'I'

ORDER BY vendor\_contact\_last\_name, vendor\_contact\_first\_name

Graphical user interface, application

Description automatically generated/\*I tried <= H but it did not include H so I changed it to < I. Why does it not work with <=H?\*/



**15. Write a SELECT statement that returns these column names and data from the Invoices table:**

**Due Date The invoice\_due\_date column**

**Invoice Total The invoice\_total column**

**10% 10% of the value of invoice\_total**

**Plus 10% The value of invoice\_total plus 10%**

USE ap;

SELECT invoice\_due\_date AS 'Due Date', invoice\_total AS 'Invoice Total',

(invoice\_total \* .1) AS '10%', (invoice\_total \* 1.1) AS 'Plus 10%'

FROM invoices

Table

Description automatically generated



**16. Write a SELECT statement that shows the invoice\_id, line\_item\_amount, and line\_item\_description from the invoice\_line\_items table where the line\_item\_amount is greater than 100.**

USE ap;

SELECT invoice\_id, line\_item\_amount, line\_item\_description

FROM invoice\_line\_items

Table

Description automatically generatedWHERE line\_item\_amount > 100

