

```
CREATE TABLE vendors6 AS
```

```
SELECT *
```

```
FROM vendors;
```

```
CREATE TABLE invoices6 AS
```

```
SELECT *
```

```
FROM invoices;
```

```
alter table vendors6
```

```
add CONSTRAINT vendors6_pk
```

```
PRIMARY KEY (vendor_id);
```

```
alter table vendors6
```

```
add CONSTRAINT vendors6_vendor_name_uq
```

```
UNIQUE (vendor_name);
```

```
alter table vendors6
```

```
add CONSTRAINT vendors6_fk_terms
```

```
FOREIGN KEY (default_terms_id)
```

```
REFERENCES terms (terms_id);
```

```
alter table vendors6  
add CONSTRAINT vendors6_fk_accounts  
FOREIGN KEY (default_account_number)  
REFERENCES general_ledger_accounts (account_number);
```

```
alter table invoices6  
add CONSTRAINT invoices6_pk  
PRIMARY KEY (invoice_id);
```

```
alter table invoices6  
add CONSTRAINT invoices6_fk_vendors  
FOREIGN KEY (vendor_id)  
REFERENCES vendors6 (vendor_id);
```

```
alter table invoices6  
add CONSTRAINT invoices6_fk_terms  
FOREIGN KEY (terms_id)  
REFERENCES terms (terms_id);
```

```
alter table invoices6
```

```
add Balance_Due Number(9,2);
```

```
UPDATE invoices6
```

```
SET BALANCE_DUE = invoice_total - payment_total - CREDIT_TOTAL;
```

```
---QUESTION 2 STARTS-----
```

```
DELETE FROM VENDORS6 WHERE VENDOR_ID = 101;
```

```
/
```

```
---END OF QUESTION 2-----
```

```
-----3.1-----
```

```
BEGIN
```

```
EXECUTE IMMEDIATE
```

```
'CREATE OR REPLACE FUNCTION get_calculate '
```

```
|| '(first_num NUMBER, second_num NUMBER, third_num NUMBER) '
```

```
|| 'RETURN NUMBER '
```

```
|| 'AS final_num NUMBER; '
```

```

|| 'BEGIN '
|| 'RETURN (first_num - (second_num + third_num)); '
|| 'END; '
;
END;
/

```

-----3.2-----TRIGGER----STARTS-----

BEGIN

EXECUTE IMMEDIATE Q'[CREATE OR REPLACE TRIGGER

BAL_TRIGGER BEFORE INSERT OR UPDATE ON INVOICES6

FOR EACH ROW

BEGIN :NEW.BALANCE_DUE := get_calculate(:NEW.INVOICE_TOTAL, :NEW.PAYMENT_TOTAL, :NEW.CREDIT_TOTAL); END;

]; END;

----TRIGGER ENDS-----=-----

/

DECLARE

new_vendor_id vendors6.vendor_id%TYPE;

new_invoice_id invoices6.invoice_id%TYPE;

BEGIN

SELECT MAX(vendor_id)+1 INTO new_vendor_id

FROM vendors6;

INSERT INTO vendors6

VALUES (new_vendor_id,'Advait R. Patel, LLC','Wabash Ave.',NULL,'Chicago','IL',60604,8475084136,'Patel','Advait',3,400);

FOR I IN 1..3 LOOP

SELECT MAX(invoice_id)+1 INTO new_invoice_id

FROM invoices6;

INSERT INTO invoices6 VALUES

(new_invoice_id,new_vendor_id,'AAA-7654-ZZZ',SYSDATE, 6400.59, 0, 0, 3, SYSDATE+30, NULL, 6400.59);

END LOOP;

UPDATE invoices6 SET credit_total = ROUND(invoice_total*0.15,2), balance_due = invoice_total-(Payment_Total + credit_total)

WHERE vendor_id = new_vendor_id;

END;

/

BEGIN

FOR I IN 1..3 LOOP

UPDATE invoices6 set balance_due = 0 WHERE invoice_id = (select MAX(invoice_id) from invoices6 where balance_due != 0);

END LOOP;

END;

/

SET LINESIZE 1000

COLUMN VENDOR_DETAILS FORMAT A30;

COLUMN INVOICE_TOTAL FORMAT \$999,999,999.00;

COLUMN CREDIT_TOTAL FORMAT \$999,999,999.00;

COLUMN PAYMENT_TOTAL FORMAT \$999,999,999.00;

COLUMN BALANCE_DUE FORMAT \$999,999,999.00;

COLUMN VENDOR_NAME FORMAT A25

```
SELECT INVOICES6.VENDOR_ID, VENDORS6.VENDOR_NAME, INVOICE_ID, TRIM(TO_CHAR(INVOICE_DATE,
'Month')) || ' ' || TRIM(TO_CHAR(INVOICE_DATE, 'DD, YYYY')) AS INVOICE_DATE , TRIM(TO_CHAR(INVOICE_DUE_DATE, 'Month')) || ' ' ||
TRIM(TO_CHAR(INVOICE_DUE_DATE, 'DD, YYYY')) AS
INVOICE_DUE_DATE , INVOICE_TOTAL, CREDIT_TOTAL, BALANCE_DUE
FROM INVOICES6
JOIN VENDORS6 ON VENDORS6.VENDOR_ID = INVOICES6.VENDOR_ID
WHERE INVOICES6.VENDOR_ID = (SELECT MAX(vendors6.vendor_id) FROM vendors6);
```

-----5.

DROP FUNCTION get_calculate;

DROP TRIGGER BAL_TRIGGER;

DROP TABLE INVOICES6;

DROP TABLE VENDORS6;

Table VENDORS6 created.

Table INVOICES6 created.

Table VENDORS6 altered.

Table VENDORS6 altered.

Table VENDORS6 altered.

Table VENDORS6 altered.

Table INVOICES6 altered.

Table INVOICES6 altered.

Table INVOICES6 altered.

Table INVOICES6 altered.

114 rows updated.

1 row deleted.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

VENDOR_ID	VENDOR_NAME	INVOICE_ID	INVOICE_DATE	INVOICE_DUE_DATE	INVOICE_TOTAL	CREDIT_TOTAL	BALANCE_DUE
124	Advait R. Patel, LLC	115	November 02, 2015	December 02, 2015	\$6,400.59	\$960.09	\$5,440.50
124	Advait R. Patel, LLC	116	November 02, 2015	December 02, 2015	\$6,400.59	\$960.09	\$5,440.50
124	Advait R. Patel, LLC	117	November 02, 2015	December 02, 2015	\$6,400.59	\$960.09	\$5,440.50

Function GET_CALCULATE dropped.

Trigger BAL_TRIGGER dropped.

Table INVOICES6 dropped.

Table VENDORS6 dropped.