5.1 Each insertion of professor information, the data are inserted into not only professor table but also into faculty\_insurance table that credit\_limit value is calculated from 300% of his/her salary and ins\_plan is ”Group Insurance for Instructor”. (\*\*trigger name: new\_professor\_added)

CREATE TRIGGER new\_professor\_added

AFTER INSERT ON Professor

FOR EACH ROW

insert into faculty\_insurance (ref\_id,ins\_plan, credit\_limit,duedate,s\_timestamp,status)

values (new.pid,"Group Insurance for Instructor",new.salary\*3,DATE\_ADD(SYSDATE(),

INTERVAL 4

YEAR),SYSDATE(),'A');

INSERT INTO Professor (pid,pname,salary) VALUES ('123','foo',1000);

SELECT \* FROM Professor;

A screenshot of a cell phone

Description automatically generated

SELECT \* FROM faculty\_insurance;

A close up of text on a black background

Description automatically generated

5.2 Convert the number declared in a numerical data type to other currencies using function named “fn\_currency(input\_number, exchange\_rate, currency\_name)” and return the result as string.

DELIMITER $$

CREATE FUNCTION fn\_currency(input\_number DECIMAL, exchange\_rate DECIMAL,currency\_name VARCHAR (60))

RETURNS

varchar(50)

DETERMINISTIC

BEGIN

DECLARE currency varchar(50);

SET currency = CONCAT(CONCAT(input\_number/exchange\_rate,' '),currency\_name);

RETURN currency;

END$$

DELIMITER ;

select fn\_currency (70,35.00,'USD');

A screenshot of a cell phone

Description automatically generated

select \*, fn\_currency (70,35.00 ,'USD') from Professor;

A close up of a black background

Description automatically generated

5.3 Update salary of all professors who earns salary less than 30,000 up to 10% and update credit limit of insurance up to 400 % of new salary and alsoinsert log into system\_log table that stores the old salary,new salary,old credit limit and new credit limit. Finally, the data stored procedure has to print the name, old salary, new salary and credit limit of all professor informationthat areupdated. (\*\*procedure name: Proc\_cal\_professor\_upvel)

DELIMITER $$

DROP PROCEDURE IF EXISTS Proc\_cal\_professor\_upvel;

CREATE PROCEDURE Proc\_cal\_professor\_upvel()

DETERMINISTIC

BEGIN

if(select COUNT(salary) from Professor where salary<30000)>0 THEN

CREATE TEMPORARY TABLE IF NOT EXISTS TMP\_PROFESSOR(PID varchar(16),salary INT,credit\_limit DECIMAL(10,2));

TRUNCATE TABLE TMP\_PROFESSOR;

insert into TMP\_PROFESSOR (PID,salary,credit\_limit)

select DISTINCT pid,salary,credit\_limit from Professor JOIN faculty\_insurance ON pid=ref\_id where salary<30000;

update Professor set salary=salary\*1.1 where pid IN (select PID from TMP\_PROFESSOR);

update faculty\_insurance JOIN Professor ON pid=ref\_id set credit\_limit=salary\*4 WHERE ref\_id IN (SELECT PID FROM TMP\_PROFESSOR);

INSERT INTO system\_log (user\_log, remark,timestamp)

select t.PID, CONCAT('old salary ',t.salary,' new salary ',p.salary,' old credit ',t.credit\_limit,' new credit ',f.credit\_limit), SYSDATE() from TMP\_PROFESSOR t JOIN Professor p ON p.pid=t.PID JOIN faculty\_insurance f ON ref\_id=p.pid;

SELECT t.PID,pname,t.salary,p.salary,t.credit\_limit,f.credit\_limit,SYSDATE() FROM TMP\_PROFESSOR t JOIN Professor p ON t.PID=p.pid JOIN faculty\_insurance f ON ref\_id=p.pid;

ELSE

select ' <30000 is empty';

END IF;

END$$

DELIMITER ;

SELECT \* FROM Professor;

A screenshot of a cell phone

Description automatically generated

SELECT \* FROM faculty\_insurance;

A screenshot of a computer screen

Description automatically generated

Call Proc\_cal\_professor\_upvel();

A screenshot of a cell phone

Description automatically generated

SELECT \* FROM Professor;

A picture containing screenshot

Description automatically generated

SELECT \* FROM faculty\_insurance;

A picture containing screenshot, screen, monitor, black

Description automatically generated