**Task 1**

Nothing to submit

**Task 2**

The if statement does not have a “THEN” after the condition

`enum` needs to be added as a parameter

The else statement does not need a “THEN”

`employee\_name` is not used, but that may or may not be an error

The example input has 5 parameters, assuming the 5th one is “shipped” date, though it is unused

Correct script:

Delimiter $$  
create procedure *add\_order*(IN onum int, IN cnum int, IN enum int, IN receive date, IN shipped date)  
begin  
 DECLARE employee\_name varchar(255);  
 if (receive is null) THEN  
 insert into ORDERS values (onum, cnum, enum, *CURDATE*(), null);  
 else  
 insert into ORDERS values (onum, cnum, enum, receive, null);  
 end if;  
end;  
$$



**Task 3**

Delimiter $$  
create procedure *ship\_order*(IN onum int, IN ship date)  
begin  
 if (ship is null) THEN  
 UPDATE ORDERS SET shipped = *CURDATE*() WHERE ono = onum;  
 else  
 UPDATE ORDERS SET shipped = ship WHERE ono = onum;  
 end if;  
end;  
$$



**Task 4**

Delimiter ||  
CREATE FUNCTION *cancel\_order*(onum INT) RETURNS varchar(255)  
begin  
 #DELETE FROM ORDERS WHERE ono = onum;  
 DECLARE message varchar(255);  
 IF not *exists*((SELECT ono FROM ORDERS where ono = onum)) THEN #no row in ORDERS  
 SET message = 'Order Details Do Not Exist! Order Does Not Exist!';  
 return message;  
 ELSEIF not *exists*((SELECT ono FROM ODETAILS where ono = onum)) THEN #no row in ODETAILS  
 DELETE FROM ORDERS WHERE ono = onum;  
 SET message = 'Order Details Do Not Exist! Order Is Cancelled Successfully!';  
 return message;  
 ELSE  
 DELETE FROM ODETAILS WHERE ono = onum;  
 DELETE FROM ORDERS WHERE ono = onum;  
 SET message = 'Order Details Removed Successfully! Order Is Cancelled Successfully!';  
 return message;  
 END IF;  
end;  
||

cancel\_order(315):

cancel\_order(666):

A close up of a screen

Description automatically generated

SELECT \* FROM ORDERS:



SELECT \* FROM ODETAILS:



**Task 5**

Delimiter ||  
CREATE procedure *add\_order\_details*(IN onum int, IN pnum int, IN qty int)  
begin  
 SELECT qoh INTO @quantity FROM PARTS WHERE pno = pnum;  
 IF qty > @quantity THEN  
 INSERT into ORDERS\_ERRORS VALUES (*CURDATE*(), onum, 'Do not have enough quantity to sell!');  
 CALL *cancel\_order*(onum);  
 ELSE  
 UPDATE PARTS SET qoh = qoh - qty WHERE pno = pnum;  
 INSERT into ODETAILS VALUES (onum, pnum, qty);  
 IF (SELECT qoh FROM PARTS WHERE pno = pnum) < (SELECT olevel FROM PARTS where pno = pnum) THEN  
 INSERT into RESTOCK VALUES (*CURDATE*(), pnum);  
 UPDATE PARTS set qoh = olevel \* 2 where pno = pnum;  
 end if;  
 end if;  
end;  
||