

(You may use compound triggers as required to solve any of these problems as required.
Hint: you may find associative arrays handy...)

1. We wish that commissions provided to employees (by updates or insertions into the emp table) are also entered into the bonus table. The employee information is added to Bonus, unless it already is present, at which time only the commission column is updated. However, the company has a peculiar rule that commissions granted outside of working hours (8AM to 5PM) are NOT reflected in the Bonus table. Write a trigger to implement this policy.
2. In an effort to reduce long-term debt, our company has decided to cap salary increases. Any salary increase over the cap is to be regarded as a commission and hence the company does not incur additional long-term salary commitment.

The salary caps are based on job classification; the caps are:

Analyst	\$4,000
Clerk	\$1,500
Manager	\$3,500
Salesman	\$2,000

There is no cap for the president.

The first business rule to be instituted is that any salary modification that exceeds the cap, the difference between the new salary and the cap is to be regarded as a commission and added to the current commission.

A second business rule that will be instituted will prevent the circumvention of the salary cap. Employees may not change jobs and thereby obtain a new position with a higher salary than allowed at their previous position (we're a tough company!) This only applies when the employee's new job is different than their old job AND their new salary exceeds what they could have made at their old job. There are obviously ways to avoid this – such as changing jobs and then later getting a bump, but for the purposes of this exercise we will keep it simple, and only prevent *simultaneous* modifications of both job and salary, and then only when the new salary is above the cap.

Write a single trigger that will accomplish these two business rules. One method for creating user-defined errors is to call the procedure raise_application_error with parameters: error_number and message where error numbers in the range from -20000 to -20999 are reserved for user defined errors. For example:

```
raise_application_error(-20101, 'Salary is missing');
```

If an attempt to change jobs (with an unpermitted salary raise) is encountered, cause the error with error number –20100 to occur with error message ‘Job modification not permitted.’ (NOTE: RAISING AN EXCEPTION AND HANDLING AN EXCEPTION ARE NOT THE SAME. YOUR TRIGGER SHOULD RAISE AN EXCEPTION ONLY.)

3. Write a ***before row*** trigger on the p table that implements the business rule that if a part's weight in an insert or update exceeds 10 units, the color of the part must be RED to flag it as a "heavy" item. This should only be applied to changes and additions: this does NOT apply to parts and weights currently in the database.
4. Write an ***after row*** trigger on the p table that implements the business rule that if a part's weight in an insert or update is less than 8 units, the color of the part must be BLUE to flag it as a "light" item. This should only be applied to changes and additions: this does NOT apply to parts and weights currently in the database.

Submit a listing of your trigger source code. Also, make sure your emp, bonus, s, p and sp tables have the original set of data. Recognize that I will likely test your triggers in your databases.