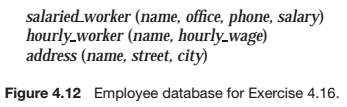
Solve question 1 using standard SQL. Solve question 2 using either standard SQL or Oracle. Please write clearly and explain your solution where needed. Please turn in stapled hard copy in class on the due date. If you use Oracle to solve question 2, please also upload an electronic copy of your solution for that question to trace.

1. [20 points] Consider the database given in Figure 4.12, page 155 of the text. Create an SQL assertion that ensures that every name that appears in the relation address appears in either salaried\_worker or hourly\_worker, but not necessarily in both.



CREATE ASSERTION worker\_address

CHECK (SELECT name

FROM address

WHERE address.name IN ((SELECT hW.name

FROM hourly\_worker as hW)

OR IN (SELECT sW.name

FROM salaried\_worker as sW)));

1. [30 points] Suppose you are given a relation emp(empid, dept, salary) and wish to maintain a materialized view deptsalary(dept, totalsalary) which stores the total salary for all employees of each department. Suppose the system does not support materialized views but supports triggers. Write SQL trigger(s) on insert on emp to keep the relation deptsalary up to date. Do not worry about deletes or updates. You can assume that there is already a tuple for each department in deptsalary so you do not need to worry about new deparments.  (Note: if you solve this question using Oracle, do not create a view for deptsalary, just create a table for it.)

CREATE TRIGGER update\_deptsalary

AFTER INSERT ON emp

REFERENCING NEW ROW as nrow

FOR EACH STATEMENT

BEGIN ATOMIC

UPDATE deptsalary

SET totalsalary = totalsalary + (SELECT salary

FROM emp

WHERE emp.salary = nrow.salary)

WHERE emp.dept = nrow.dept;

END;