PL/SQL – 1

```
DECLARE
  CURSOR dc IS
    SELECT * FROM dependent;
  dep dc%ROWTYPE;

BEGIN
  FOR dep IN dc LOOP
    DBMS_OUTPUT.PUT_LINE(dep.dependent_name
        ||'   '|| dep.relationship);
  END LOOP;
END;
/
```

PL/SQL – 2

```
CREATE OR REPLACE PROCEDURE listit(pn NUMBER)
AS
  CURSOR wc IS
    SELECT * FROM works_on WHERE pno=pn;
  works wc%ROWTYPE;
  zero EXCEPTION;

BEGIN
  FOR works IN wc LOOP
    IF works.hours<8.0 THEN
      RAISE zero;
    END IF;
    DBMS_OUTPUT.PUT_LINE(works.essn ||'  '|| works.hours);
  END LOOP;

EXCEPTION
  WHEN zero THEN
    DBMS_OUTPUT.PUT_LINE(' Found a zero');
END;
/
```

TRIGGERS – 2 required

```
CREATE OR REPLACE TRIGGER tt1
BEFORE INSERT OR UPDATE OR DELETE OF salary ON e1
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    UPDATE d1
      SET tot_sal = tot_sal + :NEW.salary
```

```
            WHERE dnumber = :NEW.dno;
    ELSIF UPDATING THEN
      UPDATE d1
        SET tot_sal = tot_sal + :NEW.salary - :OLD.salary
          WHERE dnumber = :OLD.dno;
    ELSE -- deleting
      UPDATE d1
        SET tot_sal = tot_sal - :OLD.salary
          WHERE dnumber = :OLD.dno;
    END IF;
END;
/

CREATE OR REPLACE TRIGGER tt2
BEFORE UPDATE OF dno ON e1
FOR EACH ROW
BEGIN
  UPDATE d1
    SET tot_sal = tot_sal + :NEW.salary
      WHERE dnumber = :NEW.dno;
  UPDATE d1
    SET tot_sal = tot_sal - :OLD.salary
      WHERE dnumber = :OLD.dno;
END;
/
```