Database Management Systems
Practice 3

Exercise 1

IMP

EMPNO	DEPTNO	ENAME	JOB	SAL
7000	10	SMITH	CLERK	850
7010	10	SCOTT	ANALYST	1000
7020	10	BLAKE	SALESMAN	1400
7030	10	SMITH	MANAGER	2500
7040	20	SMITH	CLERK	800
7050	20	SCOTT	ANALYST	1600
7060	30	ADAMS	CLERK	000
7070	30	JAMES	CLERK	1000
7080	40	ALLEN	CLERK	850

DIP

DEPTNO	DNAME	LOC	MINSAL	MAXSAL
10	ACCOUNTING	NEW YORK	100	2500
20	RESEARCH	DALLAS	150	3000
30	SALES	CHICAGO	120	2500
40	OPERATIONS	BOSTON	200	2100

<u>Trigger</u>
```
CREATE OR REPLACE TRIGGER UP_SAL AFTER UPDATE OF DNAME ON DIP FOR EACH ROW WHEN
(OLD.DNAME='ACCOUNTING' AND NEW.DNAME='SALES') BEGIN
  ---update employee salary when the department changes UPDATE IMP SET SAL=SAL+100
  WHERE DEPTNO=:OLD.DEPTNO;
END;
```

<u>Update statement</u>
```
UPDATE DIP SET DNAME = 'SALES' WHERE DNAME='ACCOUNTING';
```

EMPNO	DEPTNO	ENAME	JOB	SAL
7000	10	SMITH	CLERK	950
7010	10	SCOTT	ANALYST	1700
7020	10	BLAKE	SALESMAN	1500
7030	10	SMITH	MANAGER	2600
7040	20	SMITH	CLERK	800
7050	20	SCOTT	ANALYST	1600
7060	30	ADAMS	CLERK	000
7070	30	JAMES	CLERK	1000
7080	40	ALLEN	CLERK	850

DEPTNO	DNAME	LOC	MINSAL	MAXSAL
10	SALES	NEW YORK	100	2500
20	RESEARCH	DALLAS	150	3000
30	SALES	CHICAGO	120	2500
40	OPERATIONS	BOSTON	200	2100

EXERCISE 2

CARDS

CARDNO	NAME	STATUS
10	JOHN	SILVER
20	KEN	GOLD
30	LUCY	PREMIUM
40	BRUCE	SILVER
50	BILL	SILVER

FLIGHTS

FLIGHTID	DEPARTURETIME	DEPARTURECITY	ARRIVALCITY	MILES
RN12K	00:45	ROME	NEW YORK	12000
BT18K	12:34	BERLIN	TOKYO	18000
PS22K	23:59	PARIS	SIDNEY	22000
LT08K	23:59	LONDON	TORONTO	8000
TS01K	11:00	TURIN	S.MARIA DI LEUCA	1000

TICKETS

TICKETID	FLIGHTID	FLIGHTDATE	NAME	CARDNO
T01	LT08K	01-APR-07	JOHN	10

CREDITS

TICKETID	CARDNO	MILES
T01	10	8000

At the beginning the **NOTIFY** table is empty.

STEP 1

Trigger at step 1
```
CREATE OR REPLACE TRIGGER NEWTICKET AFTER INSERT ON TICKETS FOR EACH ROW WHEN
(NEW.CARDNO IS NOT NULL) DECLARE
  FLIGHTMILES NUMBER;
BEGIN ---STEP 1 ---find the miles for the flight of the ticket SELECT MILES INTO FLIGHTMILES
  FROM FLIGHTS WHERE FLIGHTID = :NEW.FLIGHTID;

 ---insert into credits the miles INSERT INTO CREDITS(TICKETID, CARDNO, MILES)
 VALUES(:NEW.TICKETID, :NEW.CARDNO, FLIGHTMILES);
 END;
```

Trace of the trigger execution at step 1
INSERT INTO TICKETS (TICKETID, FLIGHTID, FLIGHTDATE, NAME, CARDNO) VALUES ('T02', 'RN12K', '01-MAR-07', 'PIPPO', NULL);

INSERT INTO TICKETS (TICKETID, FLIGHTID, FLIGHTDATE, NAME, CARDNO) VALUES ('T03', 'RN12K', '02-APR-07', 'BILL', 50);

TICKETS

TICKETID	FLIGHTID	FLIGHTDATE	NAME	CARDNO
T01	LT08K	01-APR-07	JOHN	10
T02	RN12K	01-MAR-07	PIPPO	-
T03	RN12K	02-APR-07	BILL	50

CREDITS

TICKETID	CARDNO	MILES
T01	10	8000
T03	50	12000

STEP 2

Trigger at step 2
CREATE OR REPLACE TRIGGER NEWTICKET AFTER INSERT ON TICKETS FOR EACH ROW WHEN (NEW.CARDNO IS NOT NULL) DECLARE
 FLIGHTMILES NUMBER; TOTMILES NUMBER; CURRENTSTATUS CHAR(10); NEXTSTATUS
 CHAR(10);
BEGIN ---STEP 1 ---find the miles for the flight of the ticket SELECT MILES INTO FLIGHTMILES
 FROM FLIGHTS WHERE FLIGHTID = :NEW.FLIGHTID;

 ---insert into the credits table the miles for the customer INSERT INTO CREDITS(TICKETID,
 CARDNO, MILES) VALUES(:NEW.TICKETID, :NEW.CARDNO, FLIGHTMILES);

 ---STEP 2 ----compute the total number of miles for the customer SELECT SUM(MILES) INTO
 TOTMILES FROM CREDITS WHERE CARDNO = :NEW.CARDNO;

 ---read the current status of the customer to check if his status will change SELECT STATUS INTO
 CURRENTSTATUS FROM CARDS WHERE CARDNO = :NEW.CARDNO;

```
IF (TOTMILES >30000 AND TOTMILES <50000 AND CURRENTSTATUS <>'GOLD') THEN ---change the
 customer status to GOLD NEXTSTATUS := 'GOLD';
   ELSE IF (TOTMILES >50000 AND CURRENTSTATUS <>'PREMIUM') THEN ---change the customer
     status to PREMIUM
       NEXTSTATUS := 'PREMIUM';
   END IF; END IF;

  IF (CURRENTSTATUS <>NEXTSTATUS) THEN UPDATE CARDS SET STATUS = NEXTSTATUS WHERE
    CARDNO = :NEW.CARDNO;
  END IF; END;
```

Trace of the trigger execution at step 2

INSERT INTO TICKETS (TICKETID, FLIGHTID, FLIGHTDATE, NAME, CARDNO) VALUES ('T04',
'RN12K', '03-MAG-07', 'BILL', 50);

INSERT INTO TICKETS (TICKETID, FLIGHTID, FLIGHTDATE, NAME, CARDNO) VALUES ('T05',
'RN12K', '03-MAG-07', 'BILL', 50);

TICKETS

TICKETID	FLIGHTID	FLIGHTDATE	NAME	CARDNO
T04	RN12K	03-MAG-07	BILL	50
T01	LT08K	01-APR-07	JOHN	10
T02	RN12K	01-MAR-07	PIPPO	-
T03	RN12K	02-APR-07	BILL	50
T05	RN12K	03-MAG-07	BILL	50

CREDITS

TICKETID	CARDNO	MILES
T04	50	12000
T01	10	0000
T03	50	12000
T05	50	12000

CARDS

CARDNO	NAME	STATUS
10	JOHN	SILVER
20	KEN	GOLD
30	LUCY	PREMIUM
40	BRUCE	SILVER
50	BILL	GOLD

STEP 3

Trigger at step 3

```
CREATE OR REPLACE TRIGGER NEWTICKET AFTER INSERT ON TICKETS FOR EACH ROW WHEN
(NEW.CARDNO IS NOT NULL) DECLARE
  FLIGHTMILES NUMBER; TOTMILES NUMBER; CURRENTSTATUS CHAR(10); NEXTSTATUS
  CHAR(10); NOTIFYID NUMBER;
BEGIN ---STEP 1 ---find the miles for the flight of the ticket SELECT MILES INTO FLIGHTMILES
  FROM FLIGHTS WHERE FLIGHTID = :NEW.FLIGHTID;

  ---insert into the credits table the miles for the customer INSERT INTO CREDITS(TICKETID,
  CARDNO, MILES) VALUES(:NEW.TICKETID, :NEW.CARDNO, FLIGHTMILES);

  ---STEP 2 ----compute the total number of miles for the customer SELECT SUM(MILES) INTO
  TOTMILES FROM CREDITS WHERE CARDNO = :NEW.CARDNO;

  ---read the current status of the customer to check if his status will change SELECT STATUS INTO
  CURRENTSTATUS FROM CARDS WHERE CARDNO = :NEW.CARDNO;

  NEXTSTATUS := CURRENTSTATUS ;

IF (TOTMILES >30000 AND TOTMILES <50000 AND CURRENTSTATUS <>'GOLD') THEN ---change the
 customer status to GOLD NEXTSTATUS := 'GOLD'; ELSE IF (TOTMILES >50000 AND
 CURRENTSTATUS <>'PREMIUM') THEN ---change the customer status to PREMIUM

NEXTSTATUS := 'PREMIUM'; END IF; END IF;

 ---check if the customer status changed with respect to the previous one
  IF (CURRENTSTATUS <>NEXTSTATUS) THEN UPDATE CARDS SET STATUS = NEXTSTATUS WHERE
    CARDNO = :NEW.CARDNO;

   ---STEP 3  ---insert a new record into the NOTIFY table  ---read the last NOTIFYNO associated to
    the CARDNO of the customer
    SELECT MAX(NOTIFYNO) INTO NOTIFYID FROM NOTIFY WHERE CARDNO = :NEW.CARDNO;

   IF (NOTIFYID IS NOT NULL) THEN
      ---it is not the first notification for the customer INSERT INTO NOTIFY(CARDNO, NOTIFYNO,
      NOTIFYDATE,  OLDSTATUS,  NEWSTATUS,  TOTALMILES)  VALUES(:NEW.CARDNO,
      NOTIFYID+1, :NEW.FLIGHTDATE, CURRENTSTATUS , NEXTSTATUS, TOTMILES);
   ELSE ---it is the first notification for the customer
    INSERT  INTO  NOTIFY(CARDNO,  NOTIFYNO,  NOTIFYDATE,  OLDSTATUS,  NEWSTATUS,
      TOTALMILES)  VALUES(:NEW.CARDNO,  1,  :NEW.FLIGHTDATE,  CURRENTSTATUS  ,
      NEXTSTATUS, TOTMILES);
   END IF;
END IF;
END;
```

Trace of the trigger execution at step 3

```
INSERT  INTO  TICKETS  (TICKETID,  FLIGHTID,  FLIGHTDATE,  NAME,
CARDNO) VALUES ('T06', 'RN12K', '03-MAG-07', 'BILL', 50);

INSERT  INTO  TICKETS  (TICKETID,  FLIGHTID,  FLIGHTDATE,  NAME,
CARDNO) VALUES ('T07', 'RN12K', '03-MAG-07', 'BILL', 50);
```

TICKETS

TICKETID	FLIGHTID	FLIGHTDATE	NAME	CARDNO
T02	RN12K	01-MAR-07	PIPPO	-
T04	RN12K	03-MAG-07	BILL	50
T06	RN12K	03-MAG-07	BILL	50
T01	LT08K	01-APR-07	JOHN	10
T03	RN12K	02-APR-07	BILL	50
T05	RN12K	03-MAG-07	BILL	50
T07	RN12K	03-MAG-07	BILL	50

CARDS

CARDNO	NAME	STATUS
10	JOHN	SILVER
20	KEN	GOLD
30	LUCY	PREMIUM
40	BRUCE	SILVER
50	BILL	PREMIUM

CREDITS

TICKETID	CARDNO	MILES
T04	50	12000
T06	50	12000
T01	10	0000
T03	50	12000
T05	50	12000
T07	50	12000

NOTIFY

CARDNO	NOTIFYNO	NOTIFYDATE	OLDSTATUS	NEWSTATUS	TOTALMILES
50	1	03-MAG-07	GOLD	PREMIUM	60000

EXERCISE 3

IMP

EMPNO	ENAME	JOB	SAL
1	VERDI	SEGRETARIA	800
2	ROSSI	BANCHIERE	900
3	BIANCHI	BANCHIERE	1100

SUMMARY

JOB	NUM
SEGRETARIA	1
BANCHIERE	2

Trigger to manage the insertion of a new record into the IMP table

```
CREATE OR REPLACE TRIGGER INSERTIMP AFTER INSERT ON IMP FOR EACH ROW DECLARE
 N NUMBER;
BEGIN ---count how many employees have the job equal to the new job

SELECT COUNT(*) INTO N FROM IMP WHERE JOB = :NEW.JOB;

 IF (N=0) THEN ---the new employee is the first employee for the new job
 INSERT INTO SUMMARY(JOB, NUM) VALUES(:NEW.JOB, 1);
 ELSE ---there is at least one employee for the new job
 UPDATE SUMMARY SET NUM= NUM +1 WHERE JOB= :NEW.JOB;
 END IF; END;
```
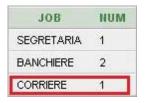
Insert statement
```
INSERT INTO IMP(EMPNO, ENAME, JOB, SAL) VALUES(4, 'NERI', 'CORRIERE', 750);
```

Database after the trigger execution

IMP

EMPNO	ENAME	JOB	SAL
1	VERDI	SEGRETARA	800
2	ROSSI	BANCHIERE	900
3	BIANCHI	BANCHIERE	1100
4	NERI	CORRIERE	750

SUMMARY

JOB	NUM
SEGRETARIA	1
BANCHIERE	2
CORRIERE	1

Trigger to manage the update of JOB on the IMP table

```
CREATE OR REPLACE TRIGGER UPD_IMP AFTER UPDATE OF JOB ON IMP FOR EACH ROW DECLARE
  N NUMBER; M NUMBER;
BEGIN ---count how many employees have the new job

SELECT COUNT(*) INTO N FROM IMP WHERE JOB=:NEW.JOB;

---increase the number of employees for the new job
 IF (N=0) THEN ---the inserted employee is the first employee for the new job
 INSERT INTO SUMMARY(JOB, NUM) VALUES(:NEW.JOB, 1);
 ELSE ---there is at least one employee for the new job
 UPDATE SUMMARY SET NUM = NUM +1 WHERE JOB = :NEW.JOB;
 END IF;

 ---decrease the number of employees for the old job

 SELECT NUM INTO M FROM SUMMARY WHERE JOB = :OLD.JOB;

 IF (M=1)THEN ---there was only one employee for the old job. Delete the record from the
   SUMMARY table DELETE FROM SUMMARY WHERE JOB = :OLD.JOB;
 ELSE ---decrease NUM in the corresponding record in the SUMMARY table
   UPDATE SUMMARY SET NUM = NUM -1 WHERE JOB = :OLD.JOB;
 END IF;
END;
```

6) Update statement
```
UPDATE IMP SET JOB='CORRIERE' WHERE EMPNO=2;
```

IMP

EMPNO	ENAME	JOB	SAL
1	VERDI	SEGRETARIA	800
2	ROSSI	CORRIERE	900
3	BIANCHI	BANCHIERE	1100
4	NERI	CORRIERE	750

SUMMARY

JOB	NUM
SEGRETARIA	1
BANCHIERE	1
CORRIERE	2