COMP- 8115-M50 Database systems

Assignment – 4

1. [20 pts]
Write SQL update statements to do the following on the database schema shown in Figure 1.2.

(a) [5 pts] Insert a new student <'Johnson', 25, 1, 'MATH'> in the database.

```
insert into student values('johnson',25,1,'Math');
```

(b) [5 pts] Change the class of student 'Smith' to 2.

```
update student set class=2 where sname='Smith';
```

(c) [5 pts] Insert a new course <'Knowledge Engineering','COSC4390', 3,'COSC'>.

```
insert into course values('Knowledge Engineering','CS4390',3,'CS');
```

(d) [5 pts] Delete the record for the student whose name is 'Smith' and student number is 17.

```
delete from student where sname='Smith' AND sno=17;
```

2. [55 pts]
For each update below, write SQL queries for the COMPANY database in Figure 5.6. Notice that
some of these updates may violate integrity constraints as discussed before.

[5 pts] Insert < 'Robert', 'F', 'Scott', '943775543', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1 > into EMPLOYEE.

Insert into Employee values ('Robert', 'F', 'Scott', '943775543', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1)

------No violation

(b) [5 pts] Insert < 'ProductA', 4, 'Bellaire', 2 > into PROJECT.

Insert into Project values ('ProductA', 4, 'Bellaire', 2)

-------Referential integrity constraint violation DNYM=2. NO such department exists

(c) [5 pts] Insert < 'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT.

insert in to Department values ('Production', 4, '943775543', '01-OCT-88')

------- Key constraint violation on DNUMBER = 4.
DNUMBER = 4 already exists

-------Referential integrity constraint violation on MGRSSN = ‘943775543’.No such employee exists

(d) [5 pts] Insert < '677678989', null, '40.0' > into WORKS_ON.

insert into WORKS_ON values (‘677678989', null, '40.0');

-------Referential integrity constraint violation on ESSSN = ‘677678989’. No such employee exists

------- Entity integrity constraint violation on PNO = NULL

(e) [5 pts] Insert < '453453453', 'John', M, '12-DEC-60', 'SPOUSE' > into DEPENDENT.

Insert into dependent values (‘453453453', 'John', M, '12-DEC-60', 'SPOUSE')

---------No violation

(f) [5 pts] Delete the WORKS_ON tuples with ESSN= '333445555'.

Delete from WORKS_ON where ESSN= '333445555'

---------- No violation

(g) [5 pts] Delete the tuple with SSN= '987654321'.

Delete from Employee where SSN= '987654321'

------- Referential integrity constraint violations. Foreign keys EMPLOYEE(SUPERSSN), DEPARTMENT(MGRSSN)< WORKS_ON(ESSN) and DEPENDENT(ESSN)
referred to the deleting tuple

(h) [5 pts] Delete the PROJECT tuple with PNAME= 'ProductX'.

Delete from PROJECT where PNAME = ‘ProductX’

---------- Referential integrity constraint violation. Foreign keys WORKS_ON
(PNO) referred to the deleting tuple

[5 pts] Modify the MGRSSN and MGRSTARTDATE of the DEPARTMENT tuple with DNUMBER=5 to '123456789' and '01-OCT-88', respectively.

Update DEPARTMENT set MGRSSN = ‘123456789' and MGRSTARTDATE= '01-OCT-88'where MGRSSN = 5

-------- No violation

(j) [5 pts] Modify the SUPERSSN attribute of the EMPLOYEE tuple with SSN= '999887777' to
'943775543'.

Update EMPLOYEE set SUPERSSN= '943775543' where SUPERSSN= '943775543'.

--------Referential integrity constraint violation on SUPERSSN = ‘943775543’. No such employee exists

(k) [5 pts] Modify the HOURS attribute of the WORKS_ON tuple with ESSN= '999887777' and
PNO= 10 to '5.0'.

Update WORKS_ON set HOURS where ESSN= '999887777' and
PNO= 10 to '5.0'.

-------- NO violation

3. [25pts]
Write SQL statements to create a table EMPOLEE_BACKUP to back up the EMPLOYEE table
shown in Figure 5.6.

As asked, we need to back up the data of the Employee it can be done as follows

1.To create the EMPOLEE_BACKUP table

Create table EMPOLEE_BACKUP LIKE EMPLOYEE.

so, this SQL statement will create the table employee EMPLOYEE_BACKUP with the same structure as the table employee

Create table is the command to create the table

LIKE is the keyword used to copy the structure of the table.

The SQL statement to insert the data into EMPLOYEE_BACKUP

Insert into EMPLOYEE_BACKUP values (select * from EMPLOYEE);

Explanation:

The SQL statement will insert the data in the table EMPLOYEE_BACKUP into the table EMPLOYEE_BACKUP

select * from EMPLOYEE_BACKUP;

Explanation:

It will display all the records from the table employee

Figure 5.6