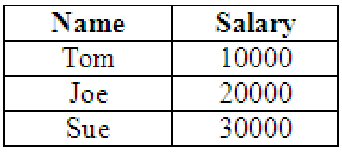
Structured Query Language

Q1. Suppose we have a relation declared by:

```
CREATE TABLE R
(
Name VARCHAR (50) PRIMARY KEY,
Salary INT CHECK (salary <= 40000)
);
```

Initially, the relation has three records:



Name	Salary
Tom	10000
Joe	20000
Sue	30000

We try to execute the following sequence of modifications.

```
(1) INSERT INTO R VALUES ('Fred', 12000);
(2) UPDATE R SET salary = 50000 WHERE name = 'Sue';
(3) INSERT INTO R VALUES (‘Tom’, 13000);
(4) DELETE FROM R WHERE name = 'Joe';
```

At the end of these statements, the sum of the salaries over all the tuples in R is:

(a) 52,000

(b) 62,000

(c) 65,000

(d) 72,000

Solution:

Name	Salary
Tom	10,000
Joe	20,000
Sue	30,000
Fred	12,000

(1)It will execute.
(2) Will not execute because salary ≤ 40,000
(3) It will also not execute because Name is primary key so we can’t add it again
(4) Delete from R where name = Joe

ne	ary
n	000
	000
d	000

than sum(salary) = 52000
= 10,000+30,000+12,000 = 52000
Answer:A

Q2. Here are three table declarations for P(A), Q(B) and R(C) with referential integrity in SQL.

```
Create table P (A primary key);
Create table Q (B primary key references P(A) on update cascade);
Create table R (C primary key references Q(B) on update cascade);
```

Initial contents of the tables are as following:

P(A) = {(1), (2), (3), (4), (5), (6)};

Q(B) = {(1), (2), (4), (6);

R(C) = {(1), (2), (6)};

Let suppose we run following modification command:

```
Update P set A = A+10 WHERE A<5;
```

What will be the output of following query assuming any referential integrity actions?

```
SELECT sum(C) FROM R;
```

(a) 39 (b) 29

(c) 19 (d) 9

Solution:

P	Q	R
A(pk)	B(pk refer P(A)	p(k). C(QB)
1	1	1
2	2	2
3	4	6
4	6	
5		
6		

Now update p set A= A+10 where A<5

p
11
12
13
14
5
6

So now in Q we need to update B and in R we need to update C.

B	C
11	11
12	12
14	6
6	

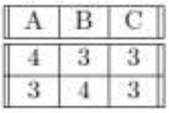
Now select sum(c) from R⇒ 11+12+6=29

Answer: B

Q3. Consider the following CREATE TABLE definition:

```
CREATE TABLE Midterm( A INT NOT NULL, B INT NOT NULL, C INT NOT
NULL,PRIMARY KEY (A), FOREIGN KEY (B) REFERENCES Midterm(A) ON
DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (C) REFERENCES
Midterm(A) ON DELETE CASCADE ON UPDATE RESTRICT)
```

Consider the following instance table Midterm:



A	B	C
4	3	3
3	4	3

What is the result of the following statement?

```
UPDATE Midterm SET B = B+1 WHERE B in (SELECT A FROM Midterm)
```

(a) Query will run successfully and update the value of B

(b) Error because foreign key constraint is violated

(c) Error because primary key constraint is violated

(d) Syntax error in query

(B)

In this

A	B	C
4	3	3
3	4	3

In this inner array return 3,4 and corresponding to this outer always in increment .value of B by 1.

So in column B, entry will be 4 and 5, but it is not possible as B is a foreign key reference to A

Thus here foreign constraint is violated

Ans :-(b)

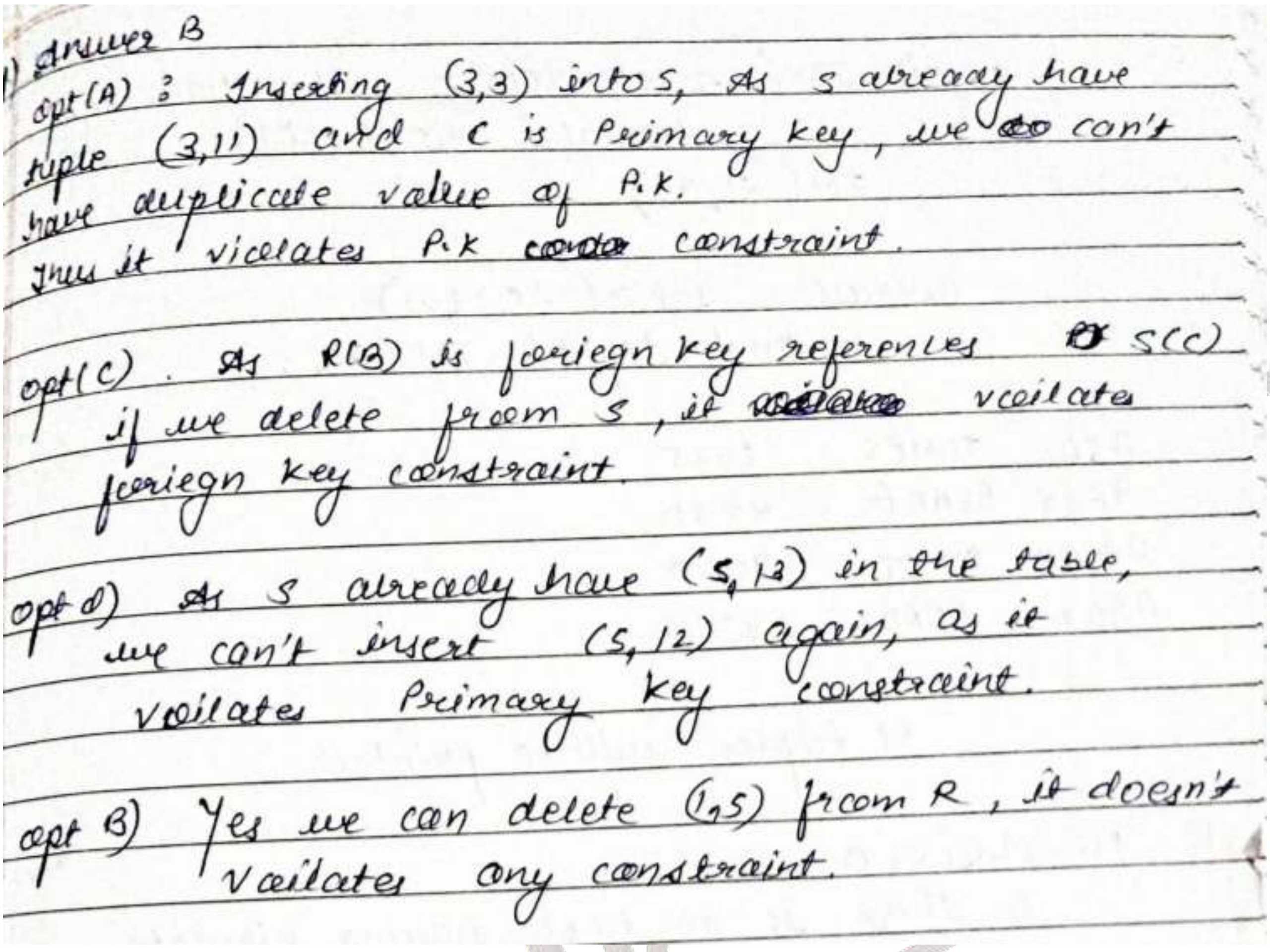
Q4. Here are declarations of two relations R and S:

```
CREATE TABLE S(C INT PRIMARY KEY, D INT);
CREATE TABLE R(A INT PRIMARY KEY, B INT FOREGIN KEY REFERENCES
S(C) );
```

R(A, B) currently contains the four tuples (0, 4), (1, 5), (2, 4), and (3, 5). S(C, D) currently contains the four tuples (2, 10), (3, 11), (4, 12), and (5, 13). As a result, certain insertions and deletions on R and S are illegal. Which of the following modifications will not violate any constraint?

(a) Inserting (3, 3) into S. (b) Deleting (1, 5) from R.

(c) Deleting (4, 12) from S. (d) Inserting (5, 12) into S.



1) Answer B

opt(A) : Inserting (3,3) into S, As S already have tuple (3,11) and C is Primary key, we ~~do~~ can't have duplicate value of P.K.

Thus it violates P.K ~~cons~~ constraint.

opt(C). As R(B) is foriegn key referenies ~~to~~ S(C) if we delete from S, it ~~violate~~ vcilates foriegn key constraint.

opt d) As S already have (5,13) in the table, we can't insert (5,12) again, as it vcilates Primary key constraint.

opt B) Yes we can delete (1,5) from R, it doesn't vcilates any constraint.

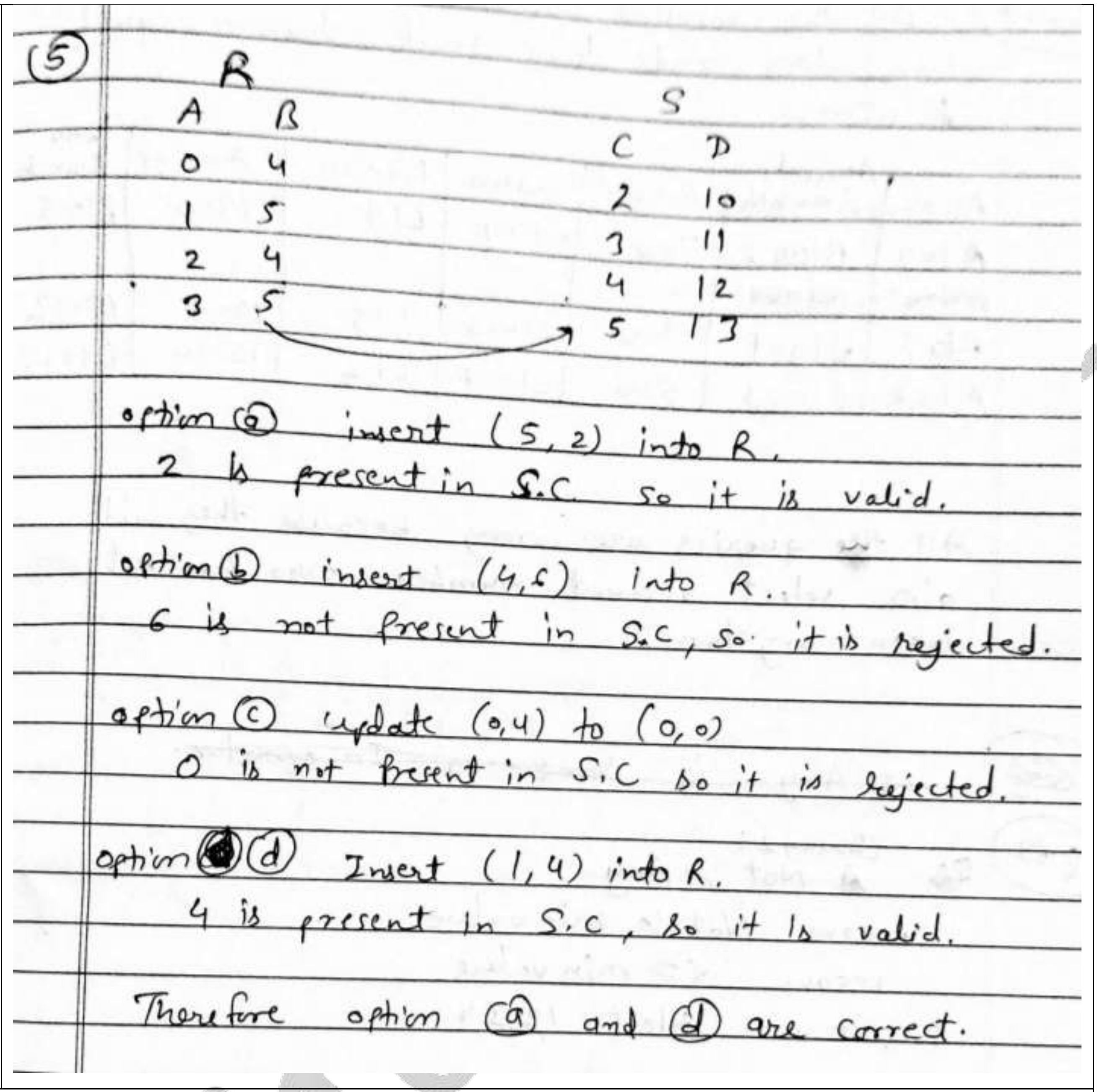
Q5. Consider the declarations of two relations R and S:

```
CREATE TABLE S(C INT PRIMARY KEY, D INT);
CREATE TABLE R(A INT PRIMARY KEY, B INT, CHECK(B IN (SELECT C FROM S)));
```

R(A, B) currently contains the four tuples (0, 4), (1, 5), (2, 4), and (3, 5). S(C, D) currently contains the four tuples (2, 10), (3, 11), (4, 12), and (5, 13). As a result, certain insertions and deletions on S are illegal, as are certain updates or insertions on R. Which of the following modifications will not be rejected because of a constraint violation?

(a) Inserting (5, 2) into R. (b) Inserting (4, 6) into R.

(c) Updating (0, 4) in R to be (0,0). (d) Inserting (1, 4) into R.



(5)

R

A	B
0	4
1	5
2	4
3	5

S

C	D
2	10
3	11
4	12
5	13

option (a) insert (5, 2) into R.
2 is present in S.C so it is valid.

option (b) insert (4,6) into R.
6 is not present in S.C, so it is rejected.

option (c) update (0,4) to (0,0)
0 is not present in S.C so it is rejected.

option (d) Insert (1, 4) into R.
4 is present in S.C, so it is valid.

Therefore option (a) and (d) are correct.

Data for the next twenty questions. Consider the following *EMP* and *DEPT* table form an organization

EMP Table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1993-06-13	800.00	0.00	20
7499	ALLEN	SALESMAN	7698	1998-08-15	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1996-03-26	1250.00	500.00	30
7566	JONES	MANAGER	7839	1995-10-31	2975.00		20
7698	BLAKE	MANAGER	7839	1992-06-11	2850.00		30
7782	CLARK	MANAGER	7839	1993-05-14	2450.00		10
7788	SCOTT	ANALYST	7566	1996-03-05	3000.00		20
7839	KING	PRESIDENT		1990-06-09	5000.00	0.00	10
7844	TURNER	SALESMAN	7698	1995-06-04	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1999-06-04	1100.00		20
7900	JAMES	CLERK	7698	2000-06-23	950.00		30
7934	MILLER	CLERK	7782	2000-01-21	1300.00		10
7902	FORD	ANALYST	7566	1997-12-05	3000.00		20
7654	MARTIN	SALESMAN	7698	1998-12-05	1250.00	1400.00	30