Assignment – 18

Maintaining the Integrity of your Data

1) Create a table called Cityorders. This will contain the same onum, amt and snum fields as the Orders table, and the same cnum and city fields as the Customers table, so that each customer's order will be entered into this table along with his or her city. Onum will be the primary key of Cityorders. All of the fields in Cityorders will be constrained to match the Customers and Orders tables. Assume the parent keys in these tables already have the proper constraints.

```
D2_92814_Krushna>CREATE TABLE Cityorders (
    ->     onum INT PRIMARY KEY,
    ->     amt DECIMAL(10,2) NOT NULL,
    ->     snum INT NOT NULL,
    ->     cnum INT NOT NULL,
    ->     city VARCHAR(20) NOT NULL,
    ->     CONSTRAINT fk_cityorders_orders FOREIGN KEY (onum)
    ->         REFERENCES Orders1(onum),
    ->            CONSTRAINT fk_cityorders_customers FOREIGN KEY (cnum)
    ->         REFERENCES Customers1(cnum)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

2) Redefine the Orders table as follows:- add a new column called prev, which will identify, for each order, the onum of the previous order for that current customer. Implement this with a foreign key referring to the Orders table itself. The foreign key should refer as well to the cnum of the customer, providing a definite enforced link between the current order and the one referenced.

```
D2_92814_Krushna>CREATE TABLE Orders3 (
    ->     onum INT,
    ->     amt DECIMAL(10,2),
    ->     odate DATE NOT NULL,
    ->     cnum INT NOT NULL,
    ->     snum INT NOT NULL,
    ->     prev INT,
    ->     CONSTRAINT pk_orders3 PRIMARY KEY (onum, cnum),
    ->     CONSTRAINT fk_orders3_prev FOREIGN KEY (prev, cnum)
    ->         REFERENCES Orders3(onum, cnum)
    -> );
Query OK, 0 rows affected (0.09 sec)
```