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.

```
D3_93068_Pankaj>create table cityorders
    -> select onum,
    -> amt,
    -> snum,
    -> cnum,
    -> city
    -> from orders
    -> natural join customer;
Query OK, 5 rows affected, 1 warning (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 1
```

```
D3_93068_Pankaj>desc cityorders;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| onum  | int         | YES  |     | NULL    |       |
| amt   | float(7,2)  | YES  |     | NULL    |       |
| snum  | int         | YES  |     | NULL    |       |
| cnum  | int         | YES  |     | NULL    |       |
| city  | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
5 rows in set (0.00 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.

ALTER TABLE cityorders **ADD**
FOREIGN key(cnum) **REFERENCES** customers(cnum);

-- Table altered.

ALTER TABLE copy_orders **ADD**
FOREIGN key(prev) **REFERENCES** copy_orders(onum);

-- Table altered.