**Hypothetical Modifying tuple in differential Table**

Let a1 = tuple old (tpo) exist in Parent Table

a2 = tuple old (tpo) does not exist in Parent Table

b1 = tuple new (tpn) exist in Parent Table

b2 = tuple new (tpn) does not exist in Parent Table

c1 = tuple old (tpo) exist in Differential Table with action ‘i’

c2 = tuple old (tpo) exist in Differential Table with action‘d’

c3 = tuple old (tpo) exist in Differential Table with action ‘m’

c4= tuple old (tpo) does not exist in Differential Table

d1 = tuple new (tpn) exist in Differential Table with action ‘i’

d2 = tuple new (tpn) exist in Differential Table with action‘d’

d3 = tuple new (tpn) exist in Differential Table with action ‘m’

d4= tuple new (tpn) does not exist in Differential Table

There are 64 combinations. We have divided these combinations in four cases which will be discussed one by one.

Legal condition of modifying tuple (LCMT)

To check the legality of the tuples (old and new) first consider the requirements of old tuple, if it is legal then check the requirements of new tuple, if it is legal then modify the required tuple. If old tuple condition is illegal then the whole combination is illegal

**Case 1:** **Tuple old and tuple new exists in Parent Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tpo & tpn exist in PT  (a1 and b1) | tpn exist in DT with action ‘i’  (d1) | tpn exist in DT with action ‘d’  (d2) | tpn exist in DT with action ‘m’  (d3) | tpn does not exist in DT  (d4) |
| tpo exist in DT with action ‘i’  (c1) | Malformed | Malformed | Malformed | Malformed |
| tpo exist in DT with action ‘d’  (c2) | Malformed | Malformed | Malformed | Malformed |
| tpo exist in DT with action ‘m’  (c3) | Malformed | Malformed | Malformed | Malformed |
| tpo does not exist in DT (c4) | Malformed | Malformed | Malformed | Malformed |

There are sixteen combinations of the said case. Every condition of case 1 is given in the form of table or matrix below. Also all combinations are described as flow diagram.

Table 5.3

**Result of case 1**

The results of all the sixteen combinations are malformed. In this scenario, tester faces abnormal hypothetical table. The tester modifies those tuples (old and new) which are already present in parent table and the old tuple is also present in differential table (DT) with actions, either insert or delete or modify but does not exist in DT respectively. Further the new tuple is also present in DT with actions, either insert or delete or modify but does not exist in DT respectively

**Case 2:** **Tuple old and tuple new does not exists in Parent Table**

There are sixteen combinations of the said case. Every condition of case 2 is given in the form of table or matrix below. Also all combinations are described as flow diagram 5.4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tpo & tpn does not exist in PT  (a2 and b2) | tpn exist in DT with action ‘i’  (d1) | tpn exist in DT with action ‘d’  (d2) | tpn exist in DT with action ‘m’  (d3) | tpn does not exist in DT  (d4) |
| tpo exist in DT with action ‘i’  (c1) | Bad request | Malformed | Malformed | Legal |
| tpo exist in DT with action ‘d’  (c2) | Bad request | Bad request | Bad request | Bad request |
| tpo exist in DT with action ‘m’  (c3) | Bad request | Bad request | Bad request | Bad request |
| tpo does not exist in DT (c4) | Bad request | Bad request | Bad request | Bad request |

**Table 5.4**

**Result of case 2**

Total conditions = 16, Legal = 1. The remaining conditions are illegal (i.e. some are abnormal request and the rest are bad request sent by tester). In this scenario tuple old and tuple new both do not exist in parent table but tpo exists in DT with action ‘i’ and tpn does not exist in DT, which is the legal condition to modify the tuple, what to do now? First delete the tuple old from DT and update the tuple old as per requirement sent by tester. After update it is called tuple new, insert tpn into DT with action ‘i’. Remaining conditions as per LCMT requirements are defined in case 5.3.11 above. As such tpo does not meet the legal requirements of modified tuples. For example in condition 5, second row first column, the result show **Bad request** why? Because tpo does not exist in PT and the same exist in DT with action‘d’ As we know the tuple which we have to modify must exist in PT. Similar are the reasons for remaining conditions.

**Case 3:** **Tuple old exist in PT and tuple new does not exists in Parent Table**

There are sixteen combinations of the said case. Every condition of the case 3 is given in the form of table or matrix below. Also all combinations are described as flow diagram 5.5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tpo exist in PT & tpn does not exist in PT  (a1 and b2) | tpn exist in DT with action ‘i’  (d1) | tpn exist in DT with action ‘d’  (d2) | tpn exist in DT with action ‘m’  (d3) | tpn does not exist in DT  (d4) |
| tpo exist in DT with action ‘i’ (c1) | Malformed | Malformed | Malformed | Malformed |
| tpo exist in DT with action ‘d’ (c2) | Bad request | Bad request | Bad request | Bad request |
| tpo exist in DT with action ‘m’ (c3) | Bad request | Bad request | Bad request | Legal |
| tpo does not exist in DT (c4) | Bad request | Bad request | Bad request | Legal |

**Table 5.5**

**Result of case 3**

Total conditions = 16, legal = 2. The remaining condition are illegal (i.e. some are abnormal request and the rest are bad request sent by tester). The locations of two legal conditions are third row fourth column called condition 12 and fourth row froth column called condition 16. In condition 12, tpo exists in PT and tpn does not exist in PT, but tpo exists in DT with action ‘m’ and tpn does not exist in DT, which is legal but requires preparation, because tpo exists in DT with action modify, so delete tpo from DT and insert tpn into DT with tuple id equal to id old and action ‘m’. In condition 16, which is purely a legal condition, tpo exist in PT, tpn does not exist in PT, tpo and tpn both not exist in DT, retrieved the desired tuple from PT and insert into DT with action column ‘m’, and applying query 5.3.8 (Query for Established new HypotheticalState). The rest of conditions throw error, in other words abnormal or bad request sent by tester. For example condition 1, shows malformed hypothetical state. Because tpo exists in PT as well as in DT with action ‘i’, abnormal request sent by tester. Because the tuple which we have to modify is already exists in DT with action insert. According to legal condition to modify tuple, if tpo is illegal then whole combination is illegal.

**Case 4:** **Tuple old does not exist in PT and tuple new exists in Parent Table**

There are sixteen combinations of the said case. Every condition of the case 1 is given in the form of table or matrix below. Also all combinations are described as flow diagram 5.6.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tpo does not exist in PT & tpn exist in PT (a2 and b1) | tpn exist in DT with action ‘i’  (d1) | tpn exist in DT with action ‘d’  (d2) | tpn exist in DT with action ‘m’  (d3) | tpn does not exist in DT  (d4) |
| tpo exist in DT with action ‘i’ (c1) | Malformed | Legal | Malformed | Malformed |
| tpo exist in DT with action ‘d’ (c2) | Malformed | Malformed | Malformed | Malformed |
| tpo exist in DT with action ‘m’ (c3) | Malformed | Malformed | Malformed | Malformed |
| tpo does not exist in DT (c4) | Bad request | Bad request | Bad request | Bad request |

**Table 5.6**

**Result of case 4**

Total conditions = 16, Legal = 1. The remaining condition are illegal (i.e. some are abnormal request and the rest are bad request sent by tester). The location of legal condition is first row second column which is called condition 2, is not a purely legal condition. It requires some preparation to be legal. The tpo does not exist in PT but exists in DT with action ‘i’ and tpn exists in PT as well as in DT with action‘d’. To make it legal, delete tpo and tpn from DT. The remaining conditions are abnormal request. For examples location 16, fourth row fourth column. In this situation tester sent ‘modify the tuple’ which is already present in modified form in PT. So we can say that it is the case of duplication or the tester sends ‘bad request’.