Normalization Forms

* There are mainly 5 Normalization forms in RDBMS
  + First Normal Form
  + Second Normal Form
  + Third Normal Form
  + Boyce-Codd Normal Form (BCNF)
  + Fourth Normal Form
  + Fifth Normal Form

For Each Normal Form the Table should undergo certain criteria .

### The First Normal Form – 1NF

* No repeating values in a group
* No repeating groups
* Delete the Columns which causes repeating values but gives same meaning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student\_Id** | **Name** | **Class** | **Section** | **Phone\_Number** | **Phone\_Number** |
| 102 | Rahul | CSE | A | 123 | 963 |
| 103 | Adithya | CSE | B | 456 |  |
| 104 | Varun | CSM | A | 789 | 369 |
| 105 | Rithvik | ECE | B | 741 |  |
| 106 | Uday | CSO | C | 852 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_Id** | **Name** | **Class** | **Section** |
| 102 | Rahul | CSE | A |
| 103 | Adithya | CSE | B |
| 104 | Varun | CSM | A |
| 105 | Rithvik | ECE | B |
| 106 | Uday | CSO | C |

By Using Primary Key connect the tables to avoid conflicts



|  |  |
| --- | --- |
| **Student\_Id** | **Phone\_Number** |
| 102 | 123 |
| 102 | 963 |
| 103 | 456 |
| 104 | 789 |
| 104 | 369 |
| 105 | 741 |
| 106 | 852 |

### The Second Normal Form – 2NF

* Should be in 1NF
* No Partial Dependency
* Occurs when there is composite key

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student\_Id** | **Name** | **Subject** | **Marks** | **Teacher** |
| 101 | Carlos | Mathematics | 91 | M |
| 102 | Jill | Physics | 92 | P |
| 103 | Chris | Chemistry | 87 | C |
| 104 | Leon | English | 89 | E |
| 101 | Carlos | Social | 85 | S |
| 102 | Jill | History | 87 | S |

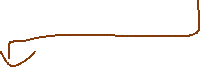
Here from the table we observe the values of primary keys are repeated and if we take question as Get the marks of 101 ..The conflict occurs that in which subject because there are two values for 101..

In such cases we take another column as composite key which describes another important group..

* Here Student\_Id is Primary key
* And Subject is Composite Key

Teacher Column is Partial Dependency for the Course so we divide the partial dependency column from the actual table

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_Id** | **Name** | **Subject** | **Marks** |
| 101 | Carlos | Mathematics | 91 |
| 102 | Jill | Physics | 92 |
| 103 | Chris | Chemistry | 87 |
| 104 | Leon | English | 89 |
| 101 | Carlos | Social | 85 |
| 102 | Jill | History | 87 |



|  |  |
| --- | --- |
| **Subject** | **Teacher** |
| Mathematics | M |
| Physics | P |
| Chemistry | C |
| English | E |
| Social | S |
| History | S |

### The Third Normal Form – 3NF

* Should be in 2NF
* No Transitive Dependency

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_Id** | **Name** | **Exam\_Type** | **MaxMarks** |
| 101 | Sagar | Viva | 5 |
| 102 | Shiva | Theory | 40 |
| 103 | Sam | Coding | 20 |
| 104 | Suresh | Viva | 5 |

Here MaxMarks Transitively depends on Exam\_type which violates 3NF so break the MaxMarks from the actual table.

|  |  |  |
| --- | --- | --- |
| **Student\_Id** | **Name** | **Exam\_Type** |
| 101 | Sagar | Viva |
| 102 | Shiva | Theory |
| 103 | Sam | Coding |
| 104 | Suresh | Viva |



|  |  |
| --- | --- |
| **Exam\_Type** | **MaxMarks** |
| Viva | 5 |
| Theory | 40 |
| Coding | 20 |
| Viva | 5 |

Boyce-Codd Normal Form (BCNF):

* Should be in 3NF
* Left hand side of Each Functional dependency should be either candidate key or super key

|  |  |  |  |
| --- | --- | --- | --- |
| Roll\_no | Name | Voter\_Id | Age |
| 101 | Chandrashekar | HD1 | 19 |
| 102 | Rajesh | HD5 | 20 |
| 103 | Tyson | HD7 | 18 |

Left hand side of Each Functional dependency should be either candidate key or super key

Candidate Keys: Functional Dependency:

{Roll\_no,Voter\_id,} (valid)Roll\_no🡪Name

(valid)Roll\_no🡪Voter\_id

(valid)Voter\_id🡪Age

(valid)Voter\_id🡪Name

Hence it satisfies the criteria of BCNF.

ACID in RDBMS

A – (ATOMICITY):

This means that the entire transaction takes place at once or doesn’t happen at all.

EXAMPLE FOR ATOMICITY:

Partial Transaction

Receiver

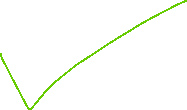
Sender



Successful Transaction

Receiver

Sender



C – (CONSISTENCY):



This means that integrity constraints must be maintained so that database is consistent before and after transaction.

EXAMPLE FOR CONSISTENCY:

Total Count Before Transaction should equals to Total Count After Transaction.

Total Money before Transaction = 800 + 200 =1000

Total Money after Transaction = 600 + 400 =1000

I – (ISOLATION):

This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of database state.

EXAMPLE FOR ISOLATION:

Suppose there are 3 users A,B,C and A sends 300 to B and B sends 50 to C while C sends 20 to B simultaneously,In this case maybe C’s transaction shows error and it fails and there can be chance of loss of 20.So Isolation between transactions is compulsory.

D – (DURABILITY):

This property ensures that once the transaction has completed execution,the updates and modifications to the database are stored in and written to disk.

EXAMPLE FOR DURABILITY:

There are 2 users A and B

A has 100 in his account and sends 20 to B and B has 150 in account and receives 20.

These updates has to be saved permanently for accurate transactions..

The modifications done are

A has 80 in his account and B has 170 in his account.