**Normalization:-**

**“E.F. Codd”**

**Normalization is a database design technique that organizes tables in a manner that reduces “Redundancy and Dependency” of data.**

**To avoid insertion, update and deletion anomaly/Problems.**

**Normalization divides larger tables into smaller tables and links them using relationships. (Primary key and Foreign Key).**

**Anomaly (Insertion, Update , Delete ):-**

**Anomaly: Something that is not according to standard or normal.**



**Not NULL**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Address** | **Dept** |
| **101** | **Joshi** | **Noida** | **IT** |
| **102** | **Saurabh** | **Delhi** | **CS** |
| **101** | **Dinesh** | **Hyderabad** | **IT** |
| **105** | **Kamlesh** | **Bangalore** | **CS** |
| **102** | **Ram** | **Kanpur** | **IT** |

**Insert:-**

**Insert into emp(id,name,address)values(106,’hari’,’goa’)**

**Result: dept doesn’t not allow null value.**

**Update:-**

**Update emp set dept=’Sales’ where id=101**

**Result: update two rows**

**Delete:**

**delete from emp where id=101**

**Result: Delete two Rows.**

**Types of Normalization:-**

**1. INF.**

**2. 2NF. -> Fulfill the criteria of 1NF.**

**3. 3NF. -> Fulfill the criteria of 1NF & 2NF.**

**4. BCNF (Boyce & Codd Normal Form) -> Fulfill the criteria of 1NF,2NF & 3NF.**

**5. 4NF.**

**6. 5NF.**

**First Normal Form:-**

**As per the rule of first normal form:-**

1. **An attribute(column) of a table can’t hold multiple values.**
2. **It should hold only atomic (forming a single unit) values.**
3. **Each record need to be unique.**

**Atomic Value: An atomic value is a value that can’t be divided.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Mobile** | **Dept** |
| **101** | **Hari** | **89898,78783** | **IT** |
| **102** | **Shivam** | **898938** | **Sales** |
| **103** | **Rohit** | **39898,89898** | **SE** |
| **104** | **Adarsh** | **38893,98983** | **IT** |
| **105** | **Raj** | **98989** | **Mrkt** |

**Mobile column is incorrect.**

**Second Normal Form:-**

1. **It should be in the first normal form.**
2. **All Non-key attributes are fully functional Dependent on the P.K. In simple words it should not have Partial Dependency.**
3. **Composite P.K. -> Empid+Qualification**
4. **Non-Key Attribute: Age**

**Non-Key attribute which is not a part of composite key.**

|  |  |  |
| --- | --- | --- |
| **Id** | **Qualification** | **Age** |
| **101** | **MA** | **23** |
| **101** | **CA** | **23** |
| **102** | **BA** | **28** |
| **102** | **MCA** | **28** |
| **103** | **MS** | **27** |
| **103** | **MBA** | **27** |

**How to find age of Employee?**

**To make the table compiled with 2NF we can break it in two tables like this:**

|  |  |
| --- | --- |
| **ID (P.K)** | **AGE** |
| **101** | **23** |
| **102** | **28** |
| **103** | **27** |

**One To Many:-**

|  |  |
| --- | --- |
| **Id (F.K)** | **Qualification** |
| **101** | **MA** |
| **101** | **CA** |
| **102** | **BA** |
| **102** | **MCA** |
| **103** | **MS** |
| **103** | **MBA** |

**Third Normal Form:- A/B/C [ A -> B , B->C -> [A->C] ]**

1. **It is in Second Normal Form.**
2. **There is no Transitive Functional Dependency.**
3. **A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id (P.K)** | **Name** | **Pincode** | **City** |
| **100** | **Ritesh** | **8989** | **Noida** |
| **101** | **Sameer** | **3422** | **Punjab** |
| **102** | **Ajay** | **50494** | **Delhi** |
| **103** | **Bablu** | **30493** | **Goa** |
| **104** | **Tikam** | **49595** | **Banglore** |

**Note: Change in pincode require change in city.**

**Id->pincode->city => city dependent on pincode dependent on id.**

**Id -> city -> city dependent on id.**

**City indirectly dependent on id is called transitive dependency.**

**Solution: create new table for pincode and city.**

**EmpTable:-**

|  |  |  |
| --- | --- | --- |
| **Id (P.K)** | **Name** | **Pincode (F.K)** |
| **100** | **Ritesh** | **8989** |
| **101** | **Sameer** | **3422** |
| **102** | **Ajay** | **50494** |
| **103** | **Bablu** | **30493** |
| **104** | **Tikam** | **49595** |

**PinCode Table:-**

|  |  |
| --- | --- |
| **Pincode (P.K.)** | **City** |
| **8989** | **Noida** |
| **3422** | **Punjab** |
| **50494** | **Delhi** |
| **30493** | **Goa** |
| **49595** | **Banglore** |

**Boyce-Codd Normal Form(BCNF):-**

1. **BCNF is an extension to the 3NF, and is also known as 3.5 Normal form.**
2. **In BCNF if every functional dependency A->B then A has to be the Super Key (is a Key that uniquely identifies rows in a table) of that particular table.**
3. **One Student can enroll multiple subjects.**
4. **For each subject, a professor is assigned to the student.**
5. **And there can be multiple professors teaching one subject.**

**Student\_id + Subject (P.K,subject as Prime arribute).**

**Professor -> subject**

|  |  |  |
| --- | --- | --- |
| **Stu\_id** | **Subject** | **Professor** |
| **101** | **Java** | **Rajat** |
| **101** | **C++** | **Kamlesh** |
| **102** | **Java** | **Rajat2** |
| **103** | **C#** | **Chandra** |
| **104** | **Java** | **Rajat** |

**Here professor dependent upon subject.**

**Student Table**

|  |  |
| --- | --- |
| **Sid** | **Pid** |
| **101** | **1** |
| **101** | **2** |
| **102** | **3** |
| **103** | **4** |
| **104** | **1** |

**Professor Table:-**

|  |  |  |
| --- | --- | --- |
| **Pid** | **Professor** | **Subject** |
| **1** | **Rajat** | **java** |
| **2** | **Kamlesh** | **C++** |
| **3** | **Rajat2** | **Java** |
| **4** | **Chandra** | **C#** |

**Fourth Normal Form:-**

1. **A relation will be in 4NF if it is in Boyce Codd normal form.**
2. **It has No Multi-valued dependency.**
3. **For a dependency A->B, if for a single value of A, Multiple values of B exists, then the relation will be in multi-valued dependency.**

**What is Multi-Valued Dependency?**

|  |  |  |
| --- | --- | --- |
| **Sid** | **Course** | **Hobby** |
| **21** | **Computer** | **Dancing** |
| **21** | **Math** | **Singing** |
| **34** | **Chemistry** | **Dancing** |
| **74** | **Biology** | **Cricket** |
| **59** | **Physics** | **Hockey** |

**Sid->Course**

**Sid -> Hobby**

1. **Course and Hobby are two independent entities, hence there is no relationship between Course and Hobby.**
2. **To remove Multi-Valued Dependency decompose it into two tables.**

**Student\_Course Table:-**

|  |  |
| --- | --- |
| **Sid** | **Course** |
| **21** | **Computer** |
| **21** | **Math** |
| **34** | **Chemistry** |
| **74** | **Biology** |
| **59** | **Physics** |

**Student\_Hobby table:-**

|  |  |
| --- | --- |
| **Sid** | **Hobby** |
| **21** | **Dancing** |
| **21** | **Singing** |
| **34** | **Dancing** |
| **74** | **Cricket** |
| **59** | **Hockey** |

**Fifth Normal Form:-**

1. **A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.**
2. **5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.**
3. **A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.**
4. **5NF is also known as Project-join normal form (PJ/NF).**

|  |  |  |
| --- | --- | --- |
| **SUBJECT** | **LECTURER** | **SEMESTER** |
| Computer | Anshika | Semester 1 |
| Computer | John | Semester 1 |
| Math | John | Semester 1 |
| Math | Akash | Semester 2 |
| Chemistry | Praveen | Semester 1 |

**Solution:**

**P1 Table**

|  |  |
| --- | --- |
| **SEMESTER** | **SUBJECT** |
| Semester 1 | Computer |
| Semester 1 | Math |
| Semester 1 | Chemistry |
| Semester 2 | Math |

**P2 Table**

|  |  |
| --- | --- |
| **SUBJECT** | **LECTURER** |
| Computer | Anshika |
| Computer | John |
| Math | John |
| Math | Akash |
| Chemistry | Praveen |

**P3 Table**

|  |  |
| --- | --- |
| **SEMSTER** | **LECTURER** |
| Semester 1 | Anshika |
| Semester 1 | John |
| Semester 1 | John |
| Semester 2 | Akash |
| Semester 1 | Praveen |

**1NF -> No Atomicity /Non-Unique.**

**2NF -> No Partial Dependency/Functionally Dependent .**

**3NF -> No Transitive Dependency.**

**BCNF -> No Functional Dependency.**

**4NF -> Multi-Valued Dependency.**

**5NF -> Join Dependency.**