Click to verify

Third normal form example

Normalization in database design is an important process for organizing data to reduce redundancy, maintain data integrity and improve accuracy. The Third Normal Form (3NF) builds on the First (1NF) and Second (2NF) Normal Forms. Achieving 3NF ensures that the database structure is free of transitive dependencies, reducing the chances of data anomalies.Even though tables in 2NF have reduced redundancy compared to 1NF, they may still encounter issues like update anomalies. For example, if one row is updated and another one is not, this can lead to inconsistent data. This happens due to transitive dependencies, which 3NF resolves by removing such dependencies, making the database more reliable.What is Third Normal Form (3NF)? A relation is in Third Normal Form (3NF) if it satisfies the following two conditions:It is in Second Normal Form (2NF): This means the table has no partial dependencies (i.e., no non-prime attribute is dependent on a part of a candidate key).There is no transitive dependency for non-prime attributes: In simpler terms, no non-key attribute should depend on another non-key attribute. Instead, all non-key attributes should depend directly on the primary key.Understanding Transitive Dependency To fully grasp 3NF, it’s essential to understand transitive dependency. A transitive dependency occurs when one non-prime attribute depends on another non-prime attribute rather than depending directly on the primary key. This can create redundancy and inconsistencies in the database.For example, if we have the following relationship between attributes:A -> B (A determines B)B -> C (B determines C)This means that A indirectly determines C through B, creating a transitive dependency. 3NF eliminates these transitive dependencies to ensure that non-key attributes are directly dependent only on the primary key.Conditions for a Table to be in 3NFA table is in Third Normal Form (3NF) if, for every non-trivial functional dependency X→Y, at least one of the following holds:X is a superkey: This means that the attribute(s) on the left-hand side of the functional dependency (X) must be a superkey (a key that uniquely identifies a tuple in the table).Y is a prime attribute: This means that every element of the attribute set Y must be part of a candidate key (i.e., a prime attribute).Example 1: Third Normal Form (3NF)Consider the following relation for a Candidate table with the following attributes and functional dependencies:1. Functional dependency Set:The set of functional dependencies is as follows:CAND_NO → CAND_NAMECAND_NO → CAND_STATECAND_STATE → CAND_COUNTRYCAND_NO → CAND_AGEThe candidate key for this relation is {CAND_NO}, since CAND_NO uniquely identifies all other attributes in the table.3. Identifying Transitive Dependency:The issue here arises from the transitive dependency between CAND_NO and CAND_COUNTRY:CAND_NO → CAND_STATECAND_STATE → CAND_COUNTRYThis means that CAND_COUNTRY is transitively dependent on CAND_NO via CAND_STATE, which violates the Third Normal Form (3NF) rule that states that no non-prime attribute (non-key attribute) should be transitively dependent on the primary key.Converting the Relation into 3NFTo remove the transitive dependency and ensure the relation is in 3NF, we decompose the original CANDIDATE relation into two separate relations:CANDIDATE: This will store information about the candidates, including their CAND_NO, CAND_NAME, CAND_STATE, and CAND_AGE:\text{CANDIDATE (CAND_NO, CAND_NAME, CAND_STATE, CAND_AGE)}STATE_COUNTRY: This relation will store information about the states and their respective countries:\text{STATE_COUNTRY (CAND_STATE, CAND_COUNTRY)}Final Decomposed Relations:CANDIDATE (CAND_NO, CAND_NAME, CAND_STATE, CAND_AGE)STATE_COUNTRY (CAND_STATE, CAND_COUNTRY)Why This Decomposition Works:The CANDIDATE relation now no longer has a transitive dependency. CAND_STATE no longer determines CAND_COUNTRY within this relation.The STATE_COUNTRY relation handles the CAND_STATE → CAND_COUNTRY dependency separately, ensuring that all data is now organized in a way that satisfies 3NF.Example 2: Relation R(A, B, C, D, E) Consider the relation R(A, B, C, D, E) with the following functional dependencies:A → BCCD → EB → DE → AA candidate key is a minimal set of attributes that can uniquely identify a tuple (row) in the relation. In this case, the possible candidate keys for the relation are {A, E, CD, BC}. This means that any of these sets of attributes can uniquely identify all other attributes in the relation.Step 2: Check Functional DependenciesLet's analyze the given functional dependencies:A → BC: This means that knowing A allows us to determine both B and C.CD → E: Knowing CD allows us to determine E.B → D: Knowing B allows us to determine D.E → A: Knowing E allows us to determine A.We observe that all attributes on the right-hand side of the functional dependencies are prime attributes (i.e., they are part of some candidate key). This means no non-prime attribute is dependent on another non-prime attribute (which would be a transitive dependency).Step 3: Check for Transitive DependenciesIn 3NF, a relation must be free of transitive dependencies, where a non-prime attribute depends on another non-prime attribute indirectly via the primary key.Here, A → BC and B → D, so B is a non-prime attribute that determines D, and A determines B. However, since B is part of a candidate key, this does not introduce a transitive dependency.E → A and A → BC, meaning E determines A, and then A determines B and C. Again, no transitive dependency is formed because A is part of a candidate key.Since there are no transitive dependencies, the relation R satisfies the condition of 3NF.Step 4: ConclusionRelation R(A, B, C, D, E) is already in Third Normal Form (3NF) because:There are no transitive dependencies.All non-prime attributes are functionally dependent only on candidate keys.Why is 3NF Important?1. Eliminates Redundancy: 3NF helps to remove unnecessary duplication of data by ensuring that non-prime attributes (attributes not part of any candidate key) depend directly on the primary key, not on other non-prime attributes.2. Prevents Anomalies: A table in 3NF is free from common anomalies such as:Insertion Anomaly: The inability to insert data without having to insert unwanted or redundant data.Update Anomaly: The need to update multiple rows of data when a change occurs in one place.Deletion Anomaly: The unintended loss of data when a record is deleted.3. Preserves Functional Dependencies: 3NF ensures that all functional dependencies are preserved, meaning that the relationships between attributes are maintained.4. Lossless Decomposition: When decomposing a relation to achieve 3NF, the decomposition should be lossless, meaning no information is lost in the process of normalization. Conclusion In conclusion, a crucial stage in database normalization is Third Normal Form (3NF). It deals with transitive dependencies and improves data integrity through effective information organization. In the case of Relation R(A, B, C, D, E), the relation is already in 3NF because it avoids transitive dependencies and ensures that all functional dependencies are directly related to the candidate keys. Therefore, this relation is well-structured and free from insertion, update, and deletion anomalies. Our website is made possible by displaying ads to our visitors. Please supporting us by whitelisting our website. Normalization in database design is an important process for organizing data to reduce redundancy, maintain data integrity and improve accuracy. The Third Normal Form (3NF) builds on the First (1NF) and Second (2NF) Normal Forms. Achieving 3NF ensures that the database structure is free of transitive dependencies, reducing the chances of data anomalies.Even though tables in 2NF have reduced redundancy compared to 1NF, they may still encounter issues like update anomalies. For example, if one row is updated and another one is not, this can lead to inconsistent data. This happens due to transitive dependencies, which 3NF resolves by removing such dependencies, making the database more reliable.What is Third Normal Form (3NF)? A relation is in Third Normal Form (3NF) if it satisfies the following two conditions:It is in Second Normal Form (2NF): This means the table has no partial dependencies (i.e., no non-prime attribute is dependent on a part of a candidate key).There is no transitive dependency for non-prime attributes: In simpler terms, no non-key attribute should depend on another non-key attribute. Instead, all non-key attributes should depend directly on the primary key.Understanding Transitive Dependency To fully grasp 3NF, it’s essential to understand transitive dependency. A transitive dependency occurs when one non-prime attribute depends on another non-prime attribute rather than depending directly on the primary key. This can create redundancy and inconsistencies in the database.For example, if we have the following relationship between attributes:A -> B (A determines B)B -> C (B determines C)This means that A indirectly determines C through B, creating a transitive dependency. 3NF eliminates these transitive dependencies to ensure that non-key attributes are directly dependent only on the primary key.Conditions for a Table to be in 3NFA table is in Third Normal Form (3NF) if, for every non-trivial functional dependency X→Y, at least one of the following holds:X is a superkey: This means that the attribute(s) on the left-hand side of the functional dependency (X) must be a superkey (a key that uniquely identifies a tuple in the table).Y is a prime attribute: This means that every element of the attribute set Y must be part of a candidate key (i.e., a prime attribute).Example 1: Third Normal Form (3NF)Consider the following relation for a Candidate table with the following attributes and functional dependencies:1. Functional dependency Set:The set of functional dependencies is as follows:CAND_NO → CAND_NAMECAND_NO → CAND_STATECAND_STATE → CAND_COUNTRYCAND_NO → CAND_AGEThe candidate key for this relation is {CAND_NO}, since CAND_NO uniquely identifies all other attributes in the table.3. Identifying Transitive Dependency:The issue here arises from the transitive dependency between CAND_NO and CAND_COUNTRY:CAND_NO → CAND_STATECAND_STATE → CAND_COUNTRYThis means that CAND_COUNTRY is transitively dependent on CAND_NO via CAND_STATE, which violates the Third Normal Form (3NF) rule that states that no non-prime attribute (non-key attribute) should be transitively dependent on the primary key.Converting the Relation into 3NFTo remove the transitive dependency and ensure the relation is in 3NF, we decompose the original CANDIDATE relation into two separate relations:CANDIDATE: This will store information about the candidates, including their CAND_NO, CAND_NAME, CAND_STATE, and CAND_AGE:\text{CANDIDATE (CAND_NO, CAND_NAME, CAND_STATE, CAND_AGE)}STATE_COUNTRY: This relation will store information about the states and their respective countries:\text{STATE_COUNTRY (CAND_STATE, CAND_COUNTRY)}Final Decomposed Relations:CANDIDATE (CAND_NO, CAND_NAME, CAND_STATE, CAND_AGE)STATE_COUNTRY (CAND_STATE, CAND_COUNTRY)Why This Decomposition Works:The CANDIDATE relation now no longer has a transitive dependency. CAND_STATE no longer determines CAND_COUNTRY within this relation.The STATE_COUNTRY relation handles the CAND_STATE → CAND_COUNTRY dependency separately, ensuring that all data is now organized in a way that satisfies 3NF.Example 2: Relation R(A, B, C, D, E) Consider the relation R(A, B, C, D, E) with the following functional dependencies:A → BCCD → EB → DE → AA candidate key is a minimal set of attributes that can uniquely identify a tuple (row) in the relation. In this case, the possible candidate keys for the relation are {A, E, CD, BC}. This means that any of these sets of attributes can uniquely identify all other attributes in the relation.Step 2: Check Functional DependenciesLet's analyze the given functional dependencies:A → BC: This means that knowing A allows us to determine both B and C.CD → E: Knowing CD allows us to determine E.B → D: Knowing B allows us to determine D.E → A: Knowing E allows us to determine A.We observe that all attributes on the right-hand side of the functional dependencies are prime attributes (i.e., they are part of some candidate key). This means no non-prime attribute is dependent on another non-prime attribute (which would be a transitive dependency).Step 3: Check for Transitive DependenciesIn 3NF, a relation must be free of transitive dependencies, where a non-prime attribute depends on another non-prime attribute indirectly via the primary key.Here, A → BC and B → D, so B is a non-prime attribute that determines D, and A determines B. However, since B is part of a candidate key, this does not introduce a transitive dependency.E → A and A → BC, meaning E determines A, and then A determines B and C. Again, no transitive dependency is formed because A is part of a candidate key.Since there are no transitive dependencies, the relation R satisfies the condition of 3NF.Step 4: ConclusionRelation R(A, B, C, D, E) is already in Third Normal Form (3NF) because:There are no transitive dependencies.All non-prime attributes are functionally dependent only on candidate keys.Why is 3NF Important?1. Eliminates Redundancy: 3NF helps to remove unnecessary duplication of data by ensuring that non-prime attributes (attributes not part of any candidate key) depend directly on the primary key, not on other non-prime attributes.2. Prevents Anomalies: A table in 3NF is free from common anomalies such as:Insertion Anomaly: The inability to insert data without having to insert unwanted or redundant data.Update Anomaly: The need to update multiple rows of data when a change occurs in one place.Deletion Anomaly: The unintended loss of data when a record is deleted.3. Preserves Functional Dependencies: 3NF ensures that all functional dependencies are preserved, meaning that the relationships between attributes are maintained.4. Lossless Decomposition: When decomposing a relation to achieve 3NF, the decomposition should be lossless, meaning no information is lost in the process of normalization. Conclusion In conclusion, a crucial stage in database normalization is Third Normal Form (3NF). It deals with transitive dependencies and improves data integrity through effective information organization. In the case of Relation R(A, B, C, D, E), the relation is already in 3NF because it avoids transitive dependencies and ensures that all functional dependencies are directly related to the candidate keys. Therefore, this relation is well-structured and free from insertion, update, and deletion anomalies. A database table is in 3rd normal form if the following two conditions are true; A database table is in second normal form and is in first normal form. There is no transitive dependency. If A->B, then A can be a super key or candidate key. What is a transitive dependency? Transitive dependency means that a non-prime attribute A(other than candidate key) depending on another non-prime attribute B and that B is entirely dependent on the candidate key. In the left figure, the primary key is only on “A” and on the right figure, in the first relation, the primary key is on “A” and in 2nd relation, the primary key is on B. A transitive dependency is an indirect dependency. Let’s see one example of transitive dependency. X depends on Y Y depends on Z So we can say that X depends on Z A transitive dependency exists in how many tables? A Transitive dependency in a database is a relationship between values in the same table. How many attributes are required for Transitive dependency? By nature, a transitive dependency requires three or more attributes. Transitive dependency causes what kind of anomalies? Transitive dependency creates deletion, updating, and insertion anomalies in the database and is considered as a bad database design. Examples of 3rd Normal Form Normalize the following table into 3NF. SPECIALIZATION CODE SPECIALIZATION NAME COURSE CODE COURSE NAME 1 Computing T4TUTORIALS-CS1 BATABASE 2 Programming T4TUTORIALS-CS2 C++ 3 Computing T4TUTORIALS-CS3 OPERATING SYSTEM 4 Programming T4TUTORIALS-CS4 OOP SPECIALIZATION CODE SPECIALIZATION NAME COURSE CODE 1 Computing T4TUTORIALS-CS1 2 Programming T4TUTORIALS-CS2 3 Computing T4TUTORIALS-CS3 4 Programming T4TUTORIALS-CS4 COURSE CODE COURSE NAME T4TUTORIALS-CS1 BATABASE T4TUTORIALS-CS2 C++ Example 2: from 1NF to 3NF Normalize the following table into 1NF, 2NF and 3NF. Com id Com Name Prod id Prod Name Prod Quantity 1. New Electronics T4Tutorials1 LCD 333 T4Tutorials2 LED 100 2. Khan Electronic T4Tutorials3 Monitor 140 3. Neon Electronics T4Tutorials3 UPS 565 FIRST NORMAL FORM In first normal form, the duplicate columns are removed. First Normal form Com id Com Name Prod id Prod Name Prod Quantity 1. New Electronics T4Tutorials1 LCD 333 1. New Electronics T4Tutorials2 LED 100 2. Khan Electronic T4Tutorials3 Monitor 140 3. Neon Electronics T4Tutorials3 UPS 565 SECOND NORMAL FORM In case of second normal form, it contains step of first normal form in addition to removal of duplicate data which is placed in a child table COMPANY TABLE Company Table Com id Com Name 1. New Electronics 1. New Electronics 2. Khan Electronic 3. Neon Electronics PRODUCT TABLE Product Table Prod id Prod Name Prod Quantity T4Tutorials1 LCD 333 T4Tutorials2 LED 100 T4Tutorials3 Monitor 140 T4Tutorials3 UPS 565 Company Table Com id Prod id 1. T4Tutorials1 1. T4Tutorials2 2. T4Tutorials3 3. T4Tutorials3 THIRD NORMAL FORM The third normal form include 2nd normal form and further steps are carried out. In this form the columns are removed which are not dependent on primary key columns COMPANY TABLE Company Table Com id Com Name 1. New Electronics 1. New Electronics 2. Khan Electronic 3. Neon Electronics PRODUCT TABLE Product Table Prod id Prod Name Prod Quantity T4Tutorials1 LCD 333 T4Tutorials2 LED 100 T4Tutorials3 Monitor 140 T4Tutorials3 UPS 565 COMPANY_PRODUCT TABLE Company_Poduct Table Com id Prod id 1. T4Tutorials1 1. T4Tutorials2 2. T4Tutorials3 3. T4Tutorials3 Example 4: 3NF Student ID Student Name Subject ID Subject Address 18-Uni-205 Ali 01 DATABASE California 19-Uni-335 Rashid 02 DLD Pakistan 17-Uni-832 Zafar 03 C++ United States 18-Uni-192 Asad 04 DLD United Kingdom Subject ID Subject 01 DATABASE 02 DLD 03 C++ 04 DLD Student ID Student Name Student ID Address 18-Uni-205 Ali 01 California 19-Uni-335 Rashid 02 Pakistan 17-Uni-832 Zafar 03 United States 18-Uni-192 Asad 04 United Kingdom Example 5: 3NF Id Name Region Country 1 T4Tutorials A Region A US 2 T4Tutorials B Region B UK 3 T4Tutorials C Region C France 4 T4Tutorials D Region D Pakistan 5 T4Tutorials E Region E Pakistan Id Name Region 1 T4Tutorials A Region A 2 T4Tutorials B Region B 3 T4Tutorials C Region C 4 T4Tutorials D Region D 5 T4Tutorials E Region E Region Country Region A US Region B UK Region C France Region D Pakistan Region E Pakistan Quiz of 3NF 1. If there exists transitive dependency, then the database is in 3rd normal form? No 2. Dependency shown below is the transitive dependency? Yes, No X depends on Y, Y depends on Z. Z depends on Y. so X depends on Z Yes What are characteristics of third normal form? A database table is in second normal form and there is no transitive dependency. Third normal form addresses which dependency? Third normal form addresses the transitive dependency. Third normal form is based on what concept? Third normal form is based on the concept of removing transitive dependencies. What is the difference between third normal form and BCNF? In third normal, the concept is to remove the transitive dependency and in BCNF Non-prime attribute never determines prime attribute. Converting to third normal form always avoids the problems related to dependencies. True/False Answer: F ___ __ __ converts a table that is in third normal form to a table no longer in third normal form. A. Conversion B. Denormalization C. Normalization D. Replication Answer: B. De-normalization Data in third normal form (3NF) contains which of the following? A. partial dependencies B. transitive dependencies C. repeating groups D. none of the above Answer: D. none of the above SQL third normal form and third normal form are same or different? Answer: Both are same. Converting to third normal form always avoids the problems related to dependencies. A.True B.False Answer: B.False Comparison of 3NF and 3.5 NF BCNF Properties 3NF BCNF Introduced by Edgar F. Codd Raymond F. Boyce and Edgar F. Codd jointly proposed Pre-requisite Tables must be in the 2nd Normal Form. Tables must be in the Third Normal Form. Ensure Avoids transitive dependency. If A determines B, then A must be a super key. Quality of the tables Less More Non-key Determinants Can have non-key attributes as determinants Cannot have non-key attributes as determinants Decomposition Loss-less join decomposition can be achieved Sometimes loss-less join decomposition cannot be achieved Achievability Always achievable Not always Always achievable Topic Covered Why is the third normal form important? | 3rd normal form examples | 3nf in DBMS with examples | explanation of the third normal form | how to identify transitive dependency | transitive dependency diagram | transitive dependency in DBMS | transitive dependency in database normalization. Read Tutorials about Normalization in DBMS All Copyrights Reserved 2025 Reserved by T4Tutorials Updated July 6, 2023 What is Third Normal Form? Third normal form is a form used to normalize the database design to avoid duplication of data. In a Relational Database Management System, a huge amount of data gets stored across multiple tables and the storing as well as the retrieval and manipulation of data becomes easier with the introduction of the concept of the key which establishes the relationship among the tables. As we store a huge amount of data in the tables, there might be cases where duplicate records may get stored in the table which will result in inconsistent data. Also, the redundancy in the data will lead to various issues such as insert, update and delete anomalies. How does Third Normal Form Work? The update, insert and delete anomalies are seen because of the redundancy of data. In update anomaly, the data does not get updated correctly and the data becomes inconsistent due to existing redundancy of data. In case of delete anomaly, if the record that is to be deleted is present in more than one rows of a table, deleting one record creates inconsistency. In insert anomaly, a table does not allow the null values to be inserted for a specific column. The data independency is ensured by the normalization of data and the redundant data gets removed. A relation is in Third Normal Form if the relation is in First and Second Normal Form and the non-primary key attributes are not transitively dependent upon the primary key. A super key can be defined as a group of single or multiple keys which will identify the rows of a table. A candidate key is a column or set of columns in a table that can identify the record uniquely. The Transitive Dependency in a table or relation comes into picture when one non-prime attributes are dependent upon another non-prime attribute instead of it being dependent upon the primary key. So removing the transitive dependency ensures data integrity as well as less duplication of data. A relation is in Third Normal Form if one of the below conditions are true for every non-trivial functional dependency A -> B. A is a super key. B is a prime attribute where each element of B is part of any candidate key. The normalization from Second Normal Form to the Third Normal Form requires the transitive dependencies to be removed. The transitive dependencies are removed by eliminating the transitive attributes from the relation by locating these attributes in a new relation. The steps for achieving Third Normal Form are as below: A table or relation should be in Second Normal Form. The table or relation should not contain any transitive partial dependency. Example of Third Normal Form Let us consider the below table ‘TEACHER_DETAILS’ to understand the Third Normal Form better. ID NAME SUBJECT STATE COUNTRY 29 Lalita English Gujrat INDIA 33 Ramesh Geography Punjab INDIA 49 Sarita Mathematics Maharashtra INDIA 78 Zayed History Bihar INDIA The candidate key in the above table is ID. The functional dependency set can be defined as ID->NAME, ID->SUBJECT, ID->STATE, STATE->COUNTRY. If A->B and B->C are the two functional dependencies, then A->C is called the Transitive Dependency. For the above relation, ID->STATE, STATE->COUNTRY is true. So we deduce that COUNTRY is transitively dependent upon ID. This does not satisfy the conditions of the Third Normal Form. So in order to transform it into Third Normal Form, we need to break the table into two tables in total and we need to create another table for STATE and COUNTRY with STATE as the primary key. Below are the tables after normalization to the Third Normal Form. TEACHER_DETAILS: ID NAME SUBJECT STATE 29 Lalita English Gujrat 33 Ramesh Geography Punjab 49 Sarita Mathematics Maharashtra 78 Zayed History Bihar STATE_COUNTRY: STATE COUNTRY Gujrat INDIA Punjab INDIA Maharashtra INDIA Bihar INDIA Advantages of Third Normal Form Below are the advantages of Third Normal Form: Normalization increases the data quality as the unwanted data is reduced from the database. Even though the redundancy of the Second Normal Form is less as compared to the First Normal Form, it is still possible to have update anomalies. For example, if one tuple is updated only while others remains unchanged, the inconsistency of data will be there in the database. The transitive dependency creates the update anomalies and they can be removed by the usage of the Third Normal Form. The Third Normal Form is also considered to be the ample requirement to build a database as the tables in the Third Normal Form are devoid of insert, update or delete anomalies. The Third Normal Form removes the redundancy effectively so the data becomes consistent as well as maintains the data integrity. As the redundancy is reduced, the database becomes less in size and also the duplication of data is reduced which also improves the performance. Conclusion The design of a robust and effective database is crucial. The role of Normalization is also very critical as it ensures both data consistency and integrity. But normalization has certain disadvantages too such as having more tables and joining of the multiple tables taking more time in development. Also, a fully normalized database takes more time in analyzing and understanding critical business logic. So the understanding of the Third Normal Form is very important for the database designer and the usage of this normal form should be done according to the business requirement. Recommended Articles This is a guide to Third Normal Form. Here we discuss how does it Work, and Advantages of Third Normal Form along with its Examples. You can also go through our suggested articles to learn more –