Course_Book

Course	Text-Book
Physics	Mechanics
Physics	Optics
Chemistry	Organic_Chemistry
Chemistry	Inorganic_chemistry
English	English_litlerature
English	English_grammer

1.5.7 Fifth Normal Form (5NF)

A relation is said to be on 5NF, if and only if, it is in 4NF and if we can decompose table further to eliminate redundancy and anomaly and when we rejoin the decomposed table buy means of candidate keys, we should not be losing the original data or any new record set should not-arise. In simple word, joining two or more decomposed table should. not lose record nor create new records.

Consider an example of different subjects taught by differents lecturers and the lecturers taking classes for different semester.

Semester 1 will have three subject mathematics, physic and chemistry and semester 2 has only mathematics subject.

Course
Subject
Lecturer
Class

Subject	Subject	Lecture
Mathematics	Amit	Semester1
Mathematics	Ajay	Semester1
Physics	Ajay	Semester1
Physics	Summet	Semester2
Chemistry	Ravi	Semester1

In the above table, Ajay takes both mathematics and physic class for semester 1, but she does not take physic class for semester 2. In this case, combination of all these 3 fields is required to identify a valid data. If are what to add new class - semester 3 but do not known which subject and who will be taking that subject. We would be simply insetting a new entry with class as semester 3 and leaving lecturer and subject as NULL. As all the three column acts a primary key, we cannot leave other two column black.

Hence we have to decompose the table in such a way that it satisfies all the rules of 4NF and when join them by using keys, it should yield correct record.

Here / we can represent each lecture's subject area and their classes in into three P_1, P_2, P_3. (Subject, Lecture - P_1 table) (lecturer, class - P_2 table), (subject, class - P_3 table).

Course	Student-Name	Text-Book
Physics	Ajay	Mechanics
Physics	Ajay	Optics
Physics	Ravi	Mechanics
Physics	Ravi	Optics
Chemistry	Ajay	Organic_chemistry
Chemistry	Ajay	Inorganic_literature
English	Raj	English_literature
English	Raj	English_grammer

In the above table the attributes 'student_name' and 'text_book' are multiva facts about the attribute 'course'. Thus the table contains an multivalued Depende Multi-value facts are represented by → →

In the above table following MVDs (multi-valued dependency) exists

Course –> –> student–name

Course –> –> Text_book

Here, student_name and Text_book are independent of each other

Anomalies of table with MVDS

This form of the table is obuiouslly full of anomalies. If a new student join physics course then are have to make two insertions for that student in the datab which is equal to the number of physic text book. Consider the problem if there hundred text books for a subject. Similarly if a new text book is introduced f course then again we have to make multiple insertions in the table / which is eq to number of students for that course. So, there is a high degree of redundancy in table, which will lead to updates problem.

To convert the table into 4NF we have to decompose course_sudent_Book as it contains multi-valued dependency.

To put it into 4NF, two separate table are formed as shown below.

course _ student (course, student_name)

course _ Book (course, text-Book)

Course_student

Course	Student-Name
Physics	Ankit
Physics	Rahat
Chemistry	Ankit
English	Raj

- Name is functionally dependent on E code.
- Ecode is functionally dependent on Name
- Line to be added and put in next para

This table has

- Multiple candidate Keys, that is E code + proj code and name + proj code
- The condidate Keys are composite
- The condidate Keys overlap line the attribute proj code in common.

This is a situation that requires conversion to BCNF. The table is essentially in the 3NF. The only non-Key item is Hours, which is dependent on the whole keys, that is Ecode + proj code or Name + proj code.

Ecode and name are determinants since they are functionally dependent on each other. However they are not candidate keys by themselves. As per BCNF, the determinants have to be candidate keys.

To Convert table to BCNF

- Find and remove the overlapping candidate Keys. Place the part of the candidate key and attribute it is functionally dependent on in a different table.
- Group the remaining item into a table

Hence, remove Name and Ecode and place them in different table.

Employee

Employee

Ecode	Name
E 1	Ajay
E 2	Ravi
E 3	Sumit
E 4	Yash

Project

Ecode	Projcode	Dept
E 1	P2	40
E 2	P5	50
E 3	P6	15
E 4	P2	45
E 4	P5	40
E 1	P5	30

1.5.6 Fourth Normal From (4NF)

A relation is in Fourth Normal Form (4NF) if it is third normal form or BCNF and if it contains no multi-valued dependencies

Multi-valued Dependency (MVD) is the dependency where one attribute value is potentially a 'Multi-valued fact' about another.

Consider a table course_student _Book

Employee

Ecode	Dept
E 101	System
E 305	Finance
E 402	Sales
E 508	Admin
E 607	Finance
E 608	Finance

Department

Dept.	Dept. Head
System	E 901
Sales	E 906
Admin	E 908
Finance	E 909

1.5.5 BOYCE - CODD Normal Form (BCNF)

The original definition of 3NF was inadequate in some situation. It was satisfactory for the tables:

1) That had multiple candidate key
2) Where the multiple condidats keys were composite.
3) Where the multiple candidate keys were overlapped.

Therefore, a new normal form, the Boyce-codd normal Form.

A relation is in the Boyce-codd normal Form (BCNF) if and only if ev determinant is a candidate key. To test whether a relation is in BCNF, we identify the determinants and make sure that they are candidate keys. A determinant is attribute or a group of attributes on which some other attributes is fully functiona dependent.

Consider the table project given below.

Ecode	Name	Projcode	Hours
E 1	Ajay	P2	40
E 2	Ravi	P5	50
E 3	Sumit	P6	15
E 4	Yash	P2	45
E 4	Yash	P5	10
E 1	Ajay	P5	30

This table has redundancy Ecode + proj code is the primary key we can no that name + proj code would be chosen as the primary Key and hence, is a candid Key.

- Hours is functionally dependent on Ecode + proj code.
- Hours is also functionally dependent on name + proj code.

- Nam
- Eco
- Line

This tal

- Mul
- The
- The

This is in the 3NF. Th that is Ecode

Ecode each other. H determinants

To Convert t

- Find a candid table.
- Group

Hence,

Employee

Emplo

Ecod
E 1
E 2
E 3
E 4

1.5.6 Fourth

A relat and if it cont

Multi-v is potentially

Consid

1.5.4 Third Normal Form (3NF)

A relation is in third normal form if it is in second normal form and no transitive dependences exist. A transitive dependency in a relation is a functional dependency between two or more non - key attributes.

Consider the table Employee

Ecode	Dept	DeptHead
E 101	System	E 901
E 305	Finance	E 909
E 402	Sales	E 906
E 508	Admin	E 908
E 607	Finance	E 909
E 608	Finance	E 909

In above table Ecode in the primary key, so that all the remaining attributes are functionally dependent on this attribute. However there is a transitive dependency Dept Head in dependent on Dept and Dept is functionally dependent on Ecode. This table is not in 3NF since Dept Head is functionally not dependent on the key value and should be removed to another table.

As a result by the transitive dependency, there are anomalies in Employee table.

Insertion

The department head of a new department that does not have any employee at present cannot be entered in the Dept. Head column.

Updation

For given department the code for a particular head (Dept. Head) is repeated several times. Hence if a department head moves to another department, the changes will have to be made consistently across the table.

Deletion

If the record of an employee is deleted the information regarding the head of the department will also be deleted. Hence, there will be a loss of information.

To convert the table employee in 3NF, we must remove the column DeptHead and place if in another table called Department along with the attribute Dept that is functionally dependent on.

2. Updation

For a given employee the employee code, department name and departmen head are repeated several times. Hence if an employee is transferred to anothe department, this charge will have to be recorded in every row of the employee tabl pertaining to that employee. Any omission will lead to in consistencies.

3. Deletion

When an employee completes work on a project, the employee record is deleted The information regarding the department to which the employee belongs will als be lost.

The table satisfies the definition of 1 NF. Now we have to check if it satisfies NF.

In table for each value of Ecode, there is more then one value of Hours for E Ecode E 101 there are three values. Hence hours in not functionally dependent o Ecode, similarly for each value of proj code there is more then one value of hour However, for a combination of the Ecode and projcode value there in exactly on value of Hours. Hence Hours is functionally dependent on the composite key Ecod projcode. Dept. is functionally dependent on the whole key, Ecode + projcode. Fo each value of Ecode there is exactly one value of Dept. How ever for each value of pro code there is more than one value of Dept. Hence Dept is not functionally depend o projcode. Dept is therefore functionally dependent on part of the key and full dependent on whole key. Similar depending is true for the Dept Head attribute.

To convert the table project into 2NF. We must remove the attributes that ar not functionally dependent on the composite (whole key) and place them in differen table along with the attributes that it is functionally dependent on. In above example since Dept is not functionally dependent on the whole key Ecode + projcode, w place Dept. along with Ecode in a separate table called. Employee Dept. Also mov Dept Head to Employee next table.

Project Table After Normalization

Employeedept

Ecode	Dept	Dept Head
E 101	System	E 901
E 305	Finance	E 909
E 508	Admin	E 908

Project

Ecode	Projcode	Dept
E 101	P27	90
E 101	P51	101
E 101	P20	60
E 305	P27	10
E 508	P51	40
E 508	P27	72

Given a relation (table R, attribute A is functionally dependent on attribute B if each value of A in R is associated with precisely one value of B. In other word attribute. A is functionally dependent on B if and only if, for each value of B there is exactly one value of A.

Attribute B is called determinant consider the following table Employee.

Code	Name	City
E_1	Ravi	Delhi
E_2	Raj	Bombay
E_3	Sumeet	Madras

Given particular value of code there is precisely one corresponding value for name. For example for code E_1 there in exactly one value of Name, Ravi. Hence Name is functionally dependent on code. Similarly, there is exactly one value of city for each value of code. Hence attribute city in functionally dependent on the attributes code. The attribute code is the determinant you can also say that code determines city and name.

1.5.3 Second Normal Form (2NF)

A relation is in second normal form if it is first normal form and every non-key attribute is fully dependent on the primary key.

Consider a relation called project with attributes Ecode, proj code, Dept. Dept Head and Hours. The primary key for this relation is composite key Ecode + proj code.

Project has the following rows.

Ecode	Projcode	Dept	Dept. Head	Hours
E 101	P27	System	E 901	90
E 305	P27	Finance	E 909	10
E 508	P51	Admin	E 908	40
E 101	P51	Systems	E 901	101
E 101	P20	System	E 901	60
E 508	P27	Admin	E 908	72

This situation could lead to the following anomalies.

1. Insertion

The department of particular employee cannot be recorded until the employee is assigned. a project.

1.5.2 First Normal Form

A relation is said to be in first Normal Form (1NF) when every entry of the relation (table) has at most a single value.

In other words a relation is in First normal form if and only if when the intersection of each row and column contains precisely one value or atomic value.

The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most a single value.

Consider the following table project

Project

Ecode	Dept.	Dept Head	Projcode	Hours
E 101	System	E901	P27 P51 P20	90 101 60
E 305	Sales	E 906	P27	10
E 508	Admin	E 908	P27 P27	40 72

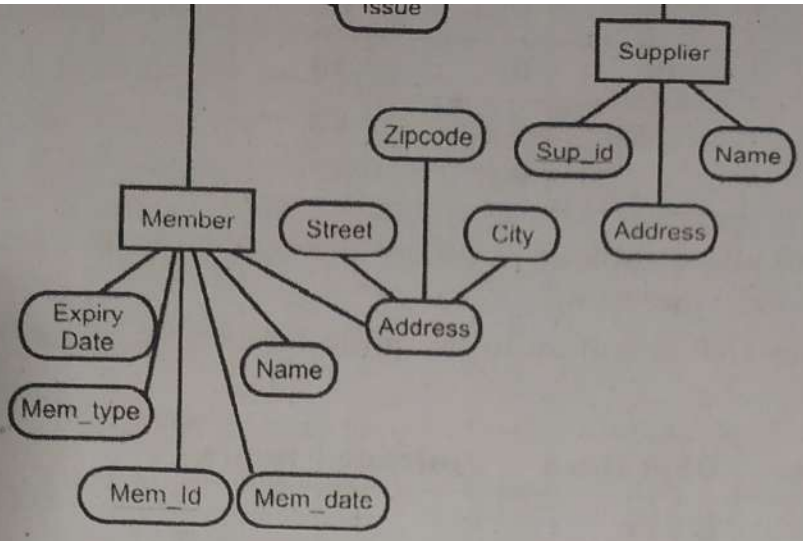
The data in the above table is not normalized because the cell in proj code and hours have more than one value.

By applying the 1NF definition to the project table, we will get the following table.

Ecode	Dept.	Dept Head	Projcode	Hours
E 101	System	E 901	P27	90
E 101	System	E 901	P51	101
E 101	System	E 901	P20	60
E 305	Sale	E 906	P27	10
E 508	Admin	E 908	P51	40
E 508	Admin	E 908	P27	72

Functional Dependency

A functional dependency is an association between two attributes of the same relational database table one of the attributes is called a determinant and the other attribute is called the determined. For each value of the determinant there is associated one and only one value of the determined.



1.5 DATA NORMALIZATION

1.5.1 Introduction

Normalization is a design technique that is widely used in designing relational model.

It is the method of organizing or structuring data into tables.

Normalization of data can be defined as a process of organizing fields and tables by a relational database to minimize redundancy and dependency. It also involues dividing larger tables into smaller and less redundant tables defining relation ship been them. Normalization serves as tool for validating and improving logical design of the data based before physical design of the data base.

11. **Database description rule:** The description of database is stored and maintained in the form of tables. This allows the users with appropriate authority to query information using similar ways and using the same languages. This implies that a data dictionary should be present within the RDBMS that is constructed of tables and views and can be examined using SQL.

12. **Distribution Independence:** The RDBMS package must have distribution independency. Thus RDBMS package must makes it possible for the database to be distributed across multiple computers even though they are having different platform both for hardware and operating system. This is one of the most attractive aspects of the RDBMS.

1.2.6 Codd's Rule:

Dr. E.F. Codd, the founder of the relational database systems, framed twelve rule to assist a database product to qualify as relational. An RDBMS product has to satisfy at least six of the 12 rules of Codd to be accepted as a full fledged RDBMS.

1. **Information Rule:** All information in a relational database including Table names,column names is represented in the form of tables. User productivity is improved since knowledge of only one language is necessary to access all data such as description of the table, attribute definitions and integrity constraints.
2. **Guaranteed Access Rule:** Every piece of data in a relational database, can be accessed by a primary key. It provides data independence and makes it possible to retrieve each individual piece of data by specifying the name of table, the column and the primary key.
3. **Comprehensive Data sub-language Rule**: the RDBMS may support many languages. But at least one of them should allow the user to define tables and views ,query and update data, set integrity constraints etc. User productivity is improved since there is only one approach that can be used for all database operations, such as Create, delete, update and query a table.
4. **View updating Rule:** Any view that can be updated theoretically can be updated using the RDBMS. Data consistency is ensured since the changes made in the view are transmitted to the base table and vice-versa.
5. **High level Insert Update and Delete:** The RDBMS supports insertion, updating and deletion at a table level. The performance is improved since the commands act on a set of records rather than one record at a time.
6. **Physical data Independency:** The execution of ad hoc requests and application programs is not affected by changes in the physical data access and storage methods. Database administrators can make changes to physical access and storage methods, to improve performance of database. These changes do not affect the user's application programs.
7. **Logical data independency:** Logical changes in tables and views such as adding or deleting columns or changing fields lengths need not necessitate modifications in the programs or in the format of requests.
8. **Integrity independence:** Integrity constraints are stored in the on line catalog or data dictionary and therefore can changed without affecting the application programs.
9. **Non subversion Rule:** If the RDBMS has a language that accesses the information of a record at a time, this language should not be used to bypass the integrity constraints.
10. **Systematic Treatment of Null Values:** In relational database null values should be supported for the representation of missing and inapplicable information. The system must have a consistent method for representing null values for example, Null values for numeric data must be distinct from zero or any other numeric value and for character data , it must be different from a string of blanks.

- **Uniqueness:** At any given time no two distinct tuples of R have the same valu for Ai,Aj and An.
- **Minimality:** No proper subset of K has uniqueness property. Means if candidate key is consist of more than one attribute ,then no individua attribute of candidate key is unique. For example if the combination of (name class) is unique then it can be called a candidate key if and only if name an class individually are not unique.

SUPER KEY

A super key has uniqueness property but not necessarily minimality property A candidate key is a special super key.

For example if Roll_number is unique in Student relation then the set o attribute (Roll_number , Name , Class) is a super key but not candidate key becaus Roll_number field can uniquely identify the record.

PRIMARY KEY

A primary key is an attribute or set of attributes that uniquely identify th record. Eevery relation must have a primary key. Primary key is chosen from candidat key the remaining keys of candidate key are called alternate keys.

Properties of primary key

1) **Stable :** The value of primary key must not change or should not become Null
2) **Minimal :** The primary key should be composed of minimum number Minima the primary key should be composed of minimum number of fields.
3) **Definitive-** A value must exist for every record at creation time
4) **Accessible-** Anyone who want to create , read or delete a record must be abl to see primary key.

COMPOSITE KEYS

Sometimes it requires more than one attribute to uniquely identify an entity A primary key that is made up of more than one attribute is known as composite key

FOREIGN KEYS

Foreign keys are attribute of a table which refers to the primary key of some other table. Foreign keys are used to link to different tables , which have some form o relationship with each other.

Example: Two table are EMP and DEPT

Degree:

The degree of a relation is the number of attributes it contain. The Student relation has four attributes, this means each row contains four value so the degree of relation is four. If a relation has one attribute its degree is one and called unary relation. A relation with two attribute is called binary, with three attributes is called ternary.

Cardinality:

The cardinality of a relation is the number of records (tuples) it contains. cardinality changes as the number of records added or deleted.

1.2.4 Relational Data Integrity

Data integrity ensures that entered data is accurate, for this some integrity rules are applied. Relational keys are used to enforce integrity. A key is used to identify a record uniquely in a table. A key is a single field or combination of fields.

Example of Student table:

Roll_number	Name	Class	Marks
1	Ajay	MCM	550
2	Rohan	MCA	450
3	Rajesh	BBA	600
4	Kamal	MCM	500

Table student containing 4 records and each containing four fields.in this table, more than one student may have same name, class and marks but must have different roll number. So we can distinguish one record from other with Roll_number column. Here Roll_number column is called a key field.A key is the relational means of specifying uniqueness. there must be at least one key field in each table sometimes a record may contain more than one key fields.

Types of keys

- Candidate key
- Super key
- Primary key
- Composite keys
- Foreign keys

Candidate key:

Most of the relations have an attribute(field) which can uniquely identify each tuple. In some cases there can be more than one attribute, this attribute is called the candidate key. Let R be a relation with attributes A1, A2,....An, the set of attribute K=(Ai,Aj...An) of R is said to be candidate key of R if and only if the following two properties are satisfied.

Primary key	Unique Identifier
Domain	Set of legal value

1.2.3 Relational Data Structure

Relation: A relation is a table with row and column.All data and relationships are represented by a two dimensional table called a relation. The two dimensions are rows and columns.

Attribute: An attribute is a named column of a relation. Attributes can appear in any order and the relation will be not affected.

Domain: a domain is a set of allowed values for one or more attributes. For example in Student table Domain for Roll_Number column consists of range of valid roll number of student.

Each attributes in the model should be assigned domain information that includes:

- **Data type :** Basic data types are integer, decimal or character. Some database also support date and time data type
- **Length :** This is the number of digit or character in the value. For example value of 5 digit or 40 character.
- **Date format :** The format for date value such as dd/mm/yy or mm/dd//yy or yy/mm/dd.
- **Range :** The range specifies the lower and upper boundaries of the values of attributes may legally have. for example in the Marks field the range is 0 to 800. Marks more than 800 is the invalid entry.
- **Constraints :** Constraints are special restrictions applied on values. For example month in a date can never exceed 12 and day of a month can never exceed 31.
- **Null support :** Indicate whether the attribute can have null or unknown value. For example, in phone number field it can have null value if student don't have phone.
- **Default value :** Defcult value is the value that an attribute will have if a value is not entered in that field.

Tuple: A tuple is a row of a relation. For example in Student table each row contains four value,one for each attribute. tuples can appear in any order and the relation will still be the same meaning.

ternary.

Cardinality:

The cardinality of a relation is the cardinality changes as the number of reco

1.2.4 Relational Data Integrity

Data integrity ensures that entere rules are applied. Relational keys are us identify a record uniquely in a table. A ke

Example of Student table:

Roll_number	Name	Class	M
1	Ajay	MCM	5
2	Rohan	MCA	4
3	Rajesh	BBA	
4	Kamal	MCM	

Table student containing 4 rec table, more than one student may have different roll number. So we can distin column. Here Roll_number column is of specifying uniqueness. there must b a record may contain more than one

Types of keys

- Candidate key
- Super key
- Primary key
- Composite keys
- Foreign keys

Candidate key:

Most of the relations have an tuple. In some cases there can be m candidate key. Let R be a relation K=(Ai,Aj...An) of R is said to be ca properties are satisfied.