GALGOTIAS UNIVERSITY

School of Computing Science and Engineering

Program: BCA

Course Code:BCAC2102

Course Name: Database Management System

Lecture-19

Topic- 2NF

Faculty:-Dr. Satyajee Srivastava

Lecture-18(RECAP)

Topic- 1NF

Objective :

To acquire knowledge about 1-Normal Forms

Lecture-18

Normalization is a process that "improves" a database design by generating relations that are of higher normal forms.

The *objective* of normalization:
"*to create relations where every dependency is on the key, the whole key, and nothing but the key*".

Lecture-18

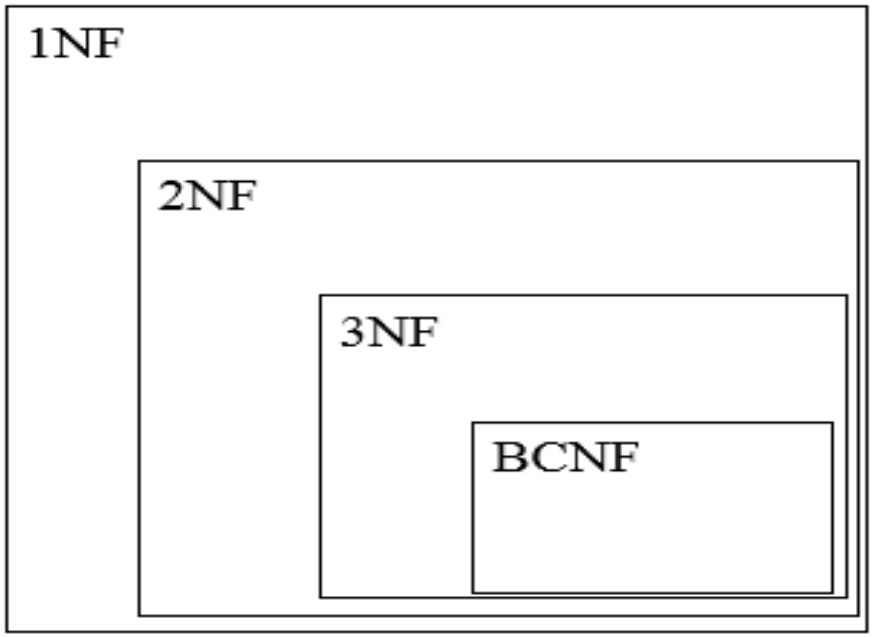
Levels of Normalization

- Levels of normalization based on the amount of redundancy in the database.
- Various levels of normalization are:
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce-Codd Normal Form (BCNF)
  - Fourth Normal Form (4NF)
  - Fifth Normal Form (5NF)
  - Domain Key Normal Form (DKNF)

Most databases should be 3NF or BCNF in order to avoid the database anomalies.

Lecture-18

Normalization



a relation in BCNF, is also in 3NF

a relation in 3NF is also in 2NF

a relation in 2NF is also in 1NF

Lecture-18

- According to Date's definition of 1NF, a table is in 1NF if and only if it is "isomorphic to some relation", which means, specifically, that it satisfies the following five conditions:

1. There's no top-to-bottom ordering to the rows.
2. There's no left-to-right ordering to the columns.
3. There are no duplicate rows.
4. Every row-and-column intersection contains exactly one value from the applicable domain (and nothing else).
5. All columns are regular [i.e. rows have no hidden components such as row IDs, object IDs, or hidden timestamps].

Lecture-18

First Normal Form

We say a relation is in **1NF** if all values stored in the relation are single-valued and atomic.

1NF places restrictions on the structure of relations. Values must be simple.

Lecture-18

First Normal Form

The following in **not** in 1NF

EmpNum	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1231	BSc, MSc

EmpDegrees is a multi-valued field:

employee 679 has two degrees: *BSc* and *MSc*

employee 333 has three degrees: *BA, BSc, PhD*

Lecture-18

To obtain 1NF relations we must, without loss of information, replace the above with two relations

Lecture-18

First Normal Form

Employee

EmpNum	EmpPhone
123	233-9876
333	233-1231
679	233-1231

EmployeeDegree

EmpNum	EmpDegree
333	BA
333	BSc
333	PhD
679	BSc
679	MSc

An outer join between Employee and EmployeeDegree will produce the information we saw before

Lecture-18

A relation is in 1NF if and only if all underlying domains contain atomic or scalar values.

For example

An attribute **customer-id** has a atomic domain but **address** attribute may not have a atomic domain as it can be divided further into city, street and road number etc.

Lecture-18

Example 2.

Unnormalized relation:

Defn: An unnormalized relation contains non atomic values.

Each row may contain multiple set of values for some of the columns, these multiple values in a single row are also called non atomic values.

Order no.	*Order date*	*Item code*	Quantity	*Price/unit*
1456	260289	3687	52	50.40
		4627	38	60.20
		3214	20	17.50
1886	040389	4629	45	20.25
		4627	30	60.20
1788	040489	4627	40	60.20

Lecture-18

Order no.	Order date	Item code	Quantity	Price/unit
1456	260289	3687	52	50.40
1456	260289	4627	38	60.20
1456	260289	3214	20	17.50
1886	040389	4629	45	20.25
1886	040389	4627	30	60.20
1788	040489	4627	40	60.20

Lecture-19

Topic- 2NF

Objective :

To acquire knowledge about 2-Normal Forms

Lecture-19

First Normal Form (1NF) does not eliminate redundancy, but rather, it's that it eliminates repeating groups.

Instead of having multiple columns of the same kind of data in a record, (0NF or Unnormalized form) you remove the repeated information into a separate relation and represent them as rows. This is what constitutes 1NF.

Second Normal Form (2NF):

Second Normal Form (2NF) is based on the concept of full functional dependency. Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes. A relation with a single-attribute primary key is automatically in at least 2NF. A relation that is not in 2NF may suffer from the update anomalies.

Lecture-19

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

In other words,

Lecture-19

A relation that is in First Normal Form and every non-primary-key attribute is fully functionally dependent on the primary key, then the relation is in Second Normal Form (2NF).

Note – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

The normalization of 1NF relations to 2NF involves the **removal of partial dependencies**. If a partial dependency exists, we remove the partially dependent attribute(s) from the relation by placing them in a new relation along with a copy of their determinant.

Lecture-19

Example-1:

Consider table as following below.

STUD_NO	COURSE_NO	COURSE_FEE
1	C1	1000
2	C2	1500
1	C4	2000
4	C3	1000
4	C1	1000
2	C5	2000

{Note that, there are many courses having the same course fee. }

Here,

COURSE_FEE cannot alone decide the value of COURSE_NO or STUD_NO;

COURSE_FEE together with STUD_NO cannot decide the value of COURSE_NO;

COURSE_FEE together with COURSE_NO cannot decide the value of STUD_NO;

Lecture-19

Hence,

COURSE_FEE would be a non-prime attribute, as it does not belong to the one only candidate key {STUD_NO, COURSE_NO} ;

But, COURSE_NO -> COURSE_FEE, i.e., COURSE_FEE is dependent on COURSE_NO, which is a proper subset of the candidate key. Non-prime attribute COURSE_FEE is dependent on a proper subset of the candidate key, which is a partial dependency and so this relation is not in 2NF.

To convert the above relation to 2NF,

we need to split the table into two tables such as :

Table 1: STUD_NO, COURSE_NO

Table 2: COURSE_NO, COURSE_FEE

Lecture-19

Table 1

STUD_NO	COURSE_NO
1	C1
2	C2
1	C4
4	C3
4	C1
2	C5

Table 2

COURSE_NO	COURSE_FEE
C1	1000
C2	1500
C3	1000
C4	2000
C5	2000

Note – 2NF tries to reduce the redundant data getting stored in memory. For instance, if there are 100 students taking C1 course, we dont need to store its Fee as 1000 for all the 100 records, instead once we can store it in the second table as the course fee for C1 is 1000.

Lecture-19

Example-2:

Consider following functional dependencies in relation R (A, B, C, D)

```
AB -> C  [A and B together determine C]
BC -> D  [B and C together determine D]
```

In the above relation, AB is the only candidate key and there is no partial dependency, i.e., any proper subset of AB doesn't determine any non-prime attribute.

Thank You