CHARLES DARWIN UNIVERSITY

HIT234 Database Concepts

Week 8

Outline of today's lecture

- Quick revision
- Normalisation
- Mapping ERDs into logical model

Normalisation

- Normalisation is a technique for producing a set of relations with desirable properties, given the data requirements of an enterprise:
  - Developed by E.F. Codd (1972)
  - Often performed as a series of tests on a relation to determine whether it satisfies or violates the requirements of a given normal form
- Four most commonly used normal forms are:
  - First (1NF),
  - Second (2NF),
  - Third (3NF),
  - Boyce-Codd (BCNF)

  then
  - 4NF, …. etc (required by some very specialised applications)

- Based on concept of functional dependencies among the attributes of a relation
- Normalization
  - Reorganisation and evaluation of table structures.
  - Produces a set of stable, well structured tables.
  - Remove repeating/redundant data.
- The normalization process helps to identify the existence of potential problems – update anomalies.

Why Normalisation is required?

If a table is not properly normalized and have data redundancy then it will not only take extra memory space but will also make it difficult to handle and update the database, without facing data loss.

Insertion, Updation and Deletion Anamolies are very frequent if database is not normalized.

Data Redundancy

Student table.

rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

Student table

rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

Insertion Anomaly

If we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students.

Updation Anomaly

In any update in (Branch, hod, offive-tel) all the student records will have to be updated as well, Another, and if by mistake we miss any record, it will lead to data **inconsistency**.

Deletion Anomaly

if student records are deleted, we will also lose the branch information

Normalization process

Normalization rules are divided into the following normal forms:

- First normal form – 1NF
- Second normal form – 2NF
- Third normal form – 3NF
- Boyce-Codd Normal Form. Not covered.
- Fourth Normal Form. Not covered.
- Fifth Normal Form... Not covered.

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Rule 1: Single Valued Attributes

- Each column of your table should be single valued which means they should not contain multiple values.

roll_no	name	subject
101	Akon	OS, CN
103	Ckon	Java
102	Bkon	C, C++

break the values into atomic values.

roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

Rule 2: Attribute Domain should not change

each column the values stored must be of the same kind or type.

roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

Rule 3: Unique name for Attributes/Columns

Eeach column in a table should have a unique name. This is to avoid confusion at the time of retrieving data or performing any other operation on the stored data.

roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

Rule 4: Order doesn't matters

This rule says that the order in which you store the data in your table doesn't matter.

roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

Second Normal Form (2NF)

For a table to be in the Second Normal Form, it must satisfy two conditions:

1. The table should be in the First Normal Form.
2. There should be no Partial Dependency.

What is Dependency?

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat

student_id is the primary key and will be unique for every row, in another word, we can use student_id to fetch any row of data from this table

This is **Dependency** or **Functional Dependency**.

Partial Dependency

Subject

subject_id	subject_name
1	Java
2	C++
3	Php

Stdents

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat

Score

score_id	student_id	subject_id	marks	teacher
1	10	1	70	Java Teacher
2	10	2	75	C++ Teacher
3	11	1	80	Java Teacher

column names teacher which is only dependent on the subject

Get me marks of student with student_id 10, can you get it from Score table? No, because you don't know for which subject. And if I give you subject_id, you would not know for which student. Hence we need student_id + subject_id to uniquely identify any row.

How to remove Partial Dependency?

One way, remove columns teacher from Score table and add it to the Subject table

subject_id	subject_name
1	Java
2	C++
3	Php

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat

Score

score_id	student_id	subject_id	marks	teacher
1	10	1	70	Java Teacher
2	10	2	75	C++ Teacher
3	11	1	80	Java Teacher

Quick Recap

1. For a table to be in the Second Normal form, it should be in the First Normal form and it should not have Partial Dependency.
2. Partial Dependency exists, when for a composite primary key, any attribute in the table depends only on a part of the primary key and not on the complete primary key.
3. To remove Partial dependency, we can divide the table, remove the attribute which is causing partial dependency, and move it to some other table where it fits in well.

Third Normal Form (3NF)

Student Table

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat
12	Bkon	09-WY	IT	Rajasthan

Subject Table

subject_id	subject_name	teacher
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher

Score Table

score_id	student_id	subject_id	Marks
1	10	1	70
2	10	2	75
3	11	1	80

exam_name	Total_marks
1	10
2	10
3	11

Third Normal Form (3NF)

For a table to be in the Third Normal Form, it must satisfy two conditions:

1. The table should be in the Second Normal Form.
2. There should be no Transitive Dependency.

Transitive Dependen

Student Table

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat
12	Bkon	09-WY	IT	Rajasthan

Subject Table

subject_id	subject_name	teacher
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher

Score Table

score_id	student_id	subject_id	Marks	exam_name	Total_marks
1	10	1	70	1	10
2	10	2	75	2	10
3	11	1	80	3	11

Well, the column total_marks depends on exam_name as with exam type the total score changes. For example, practicals are of less marks while theory exams are of more marks.

But, exam_name is just another column in the score table. It is not a primary key or even a part of the primary key, and total_marks depends on it.

This is **Transitive Dependency**. When a non-prime attribute depends on other non-prime attributes rather than depending upon the prime attributes or primary key.

How to remove Transitive Dependency?

Take out the columns exam_name and total_marks from Score table and put them in an Exam table and use the exam_id wherever required.

Student Table

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat
12	Bkon	09-WY	IT	Rajasthan

Subject Table

subject_id	subject_name	teacher
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher

The new Exam table

exam_id	exam_name	total_marks
1	Workshop	200
2	Mains	70
3	Practicals	30

Score Table

score_id	student_id	subject_id	Marks	exam_id
1	10	1	70	1
2	10	2	75	2
3	11	1	80	3

CHARLES DARWIN UNIVERSITY

Advantage of removing Transitive Dependency

- Amount of data duplication is reduced.
- Data integrity achieved.

Normalization Revision

- All attributes in every table must be determined by the whole key
- Normal Forms
  - UNF – a table that contains one or more repeating groups
  - 1NF – A relation is in 1NF if all value are atomic and no rows are repeated.
  - 2NF – A relation is said to be in 2NF if it is in 1NF and every non-key attribute is fully functionally dependent on the primary key
  - 3NF – A relation is said to be in 3NF if it is in 2NF and every non-key is non-transitively dependent on the primary key

CHARLES DARWIN UNIVERSITY

Thank you