Normalisation – Example

Version 1.0

Examples

Example 1 - One to many relationship

Scenario:

Consider an educational institution where the students are divided into many batches. Every batch has approximately 30 students each. A student can be a part of only one batch. Create a data model to store the following information.

Batch(Batch id, Batch name, Batch strength, Batch Start date, Batch end date)
Student(Student id, student name, Student age, Student address)

Step 1

Decide the relationships and cardinality

Batch to student --> One to many

Step 2

Bring all the required Attributes together

Batch id, Batch name, Batch strength, Batch Start date, Batch end date Student id, student name, Student age, Student address

Step 3

1st normal form

Lets assume that Student address is a multivalued column. So lets divide it to
[Student add line1, Student add line2, Student city]

No Repeating Groups (many to many relationships)

1NF Table

Batch id, Batch name, Batch strength, Batch Start date, Batch end date, Student id, student name, Student age, Student add line1, Student add line2, Student city

Primary key - Student id

Note:- Student id will not repeat. So it is considered as the primary key.

Step 4

2nd normal form

Partial dependency should be removed.

The above 1NF table has no partial dependency.

2NF Table

Batch id, Batch name, Batch strength, Batch Start date, Batch end date, Student id, student name, Student age, Student add line1, Student add line2, Student city

Primary key - Student id

Step 5

3rd normal form

Transitive dependency should be removed.

Batch details depend only on Batch id and not on student id. Hence they should be separated.

3NF Tables

Table name :- **Batch_Details**

[Batch id<PK>, Batch name, Batch strength, Batch Start date, Batch end date]

Primary key - Batch id

Table name :- **Student_Details**

[Student id<PK>, student name, Student age, Student add line1, Student add line2, Student city, Batch id<FK>]

Primary key - Student id
Foreign key - Batch id (references batch table)

Note 1:- Whenever you remove a set of columns to a separate table, always keep a foreign key in the table to keep the relationship.

Note 2:- If you are separating two entities that are in a one to many relationship, the foreign key should always be placed in the table on the many side of the relationship.

Note 3:- Once you create all the 3NF tables, make sure that all the relationships (determined in step1) are existing as foreign keys in the tables.

Example 2 - Many to many relationship

Scenario:

Consider the examination centre of an online educational institution. The students can come and attend the examinations for their various courses they are learning. Each student may write many examinations. Each examination will have many participants. Create a data model to store the following information.

Student details (Student id, student name, Student age)
Exam details (Exam id, Examination name, Exam start time , exam duration, exam marks)

Step 1

Decide the relationships and cardinality

Student to Exam --> many to many

Step 2

Bring all the required Attributes together

Student id, Student name, Student age, Exam id, Examination name, Exam start time , Exam duration, Exam marks

Step 3

1st normal form

There are no non atomic values

Repeating groups --> Student to exam has a many to many relationship. So separate the two entities.

Table 1:

Student id, Student name, Student age

Primary key - Student id

Table 2:

Student id, Exam id, Examination name, Exam start time , Exam duration, Exam marks

Primary key - Student id, Exam id
Foreign key - Student id (references table 1)

Note 1:- Whenever you remove a set of columns to a separate table, always keep a foreign key in the table to keep the relationship.

Note 2:- In the second table, the Primary key is a combination of Student id, Exam id because of the many to many relationship.

Step 4

2nd normal form

Partial dependency should be removed.

The columns [Examination name, Exam start time , Exam duration] depend only on the Exam id. These details are independent of the student id. So we have to remove the columns to a separate table.

2NF tables
Table 1:

Student id, Student name, Student age

Primary key - Student id

Table 2:

Exam id, Examination name, Exam start time , Exam duration

Primary key - Exam id

Table 3:

Student id, Exam id, Exam marks

Primary key - Student id, Exam id
Foreign key - Student id(references Table 1), Exam id(references Table 2)

Step 5

3rd normal form

Transitive dependency should be removed.

There is no Transitive dependency

3NF tables

Table Name:- **Student_details**

Student id<PK>, Student name, Student age

Primary key - Student id

Table Name:- **Exam_Details**

Exam id<PK>, Examination name, Exam start time , Exam duration

Primary key - Exam id

Table Name:- **Student_Marks**

Student id<PK, FK>, Exam id<PK, FK>, Exam marks

Primary key - Student id, Exam id
Foreign key - Student id(references Student_Details)
Exam id(references Exam_details)

Note 3:- If you are separating two tables that is in a many to many relationship, there will always be a relationship table which has foreign keys pointing to both the tables.

Example 3 –

Consider the computer centre of a university. The computer centre has many Rooms and each room has many computers. The university students can use any computer at any time. The student information, the computer they have used , the start time and end time gets logged in the online register. Create a data model to store the following information.

Student details(Student id, Student name, Student DOB)
Usage details(Usage start time, usage end time)
Computer details(Computer asset id, IP address)
Room details(Room number, Room capacity)

Step 1

Decide the relationships and cardinality

Room to Computer → one to many
Computer to student → many to many

Step 2

Bring all the required Attributes together

Student id, Student name, Student DOB, Usage start time, Usage end time, Computer asset id, IP address, Room number, Room capacity

Step 3

1st normal form

There are no non atomic values

Repeating groups --> Student to computer is a many to many relationship. So separate the two entities.

Table 1:

Student id, Student name, Student DOB

Primary key - Student id

Table 2:

Student id, Usage start time, Usage end time, Computer asset id, IP address, Room number, Room capacity

Primary key - Student id, Computer asset id
Foreign key - Student id (references table 1)

Step 4

2nd normal form

Partial dependency should be removed.

The columns [IP address, Room number, Room capacity] are independent of the student id. So we have to remove the columns to a separate table.

2NF tables

Table 1:

Student id, Student name, Student DOB

Primary key - Student id

Table 2:

Computer asset id, IP address, Room number, Room capacity

Primary key - Computer asset id

Table 3:

Student id , Computer asset id , Usage start time, Usage end time

Primary key - Student id, Computer asset id
Foreign key - Student id(references Table 1)
Computer asset id(references Table 2)

Step 5

3rd normal form

Transitive dependency should be removed.

In Table 2: Room capacity depends only on the room no and not on the computer asset id. So it should be moved to a separate table

3NF tables

Table Name:- **Student_details**

Student_id<PK>, Student_name, Student_DOB

Primary key - Student id

Table Name:- **Computer_Usage**

Student_id<PK, FK> , Computer_asset id <PK, FK>, Usage_start_time, Usage_end_time

Primary key - Student_id, Computer_asset_id
Foreign key - Student_id(references Student_details)
Computer_asset_id(references Computer_details)

Table Name:- **Computer_Details**

Computer_asset_id<PK>, IP_address, Room_no<FK>

Primary key - Computer_asset_id
Foreign key – Room_No(references Room_details)

Table Name:- **Room_Details**

Room_no<PK>, Room_capacity
Primary key – Room_no