
Normalization Analysis

To effectively perform the process of normalization, let's populate the relations with some sample tuples.

Package

packageName	cost
Beginner	100
Intermediate	125

Trainer

trainerNumber	fname	lname	phoneNumber
1	Umid	Muzrapov	5202555555
2	Lola	Ahmedova	5207777777

Course

className	maxParticipant	currentParticipant	startDate	endDate	trainerNumber	day	hour	minute	duration
Yoga 001	60	35	05/11/2023	07/11/2023	1	1	13	30	60
Strength 002	55	30	05/12/2023	07/11/2023	2	2	13	30	60
Yoga 001	60	35	05/11/2023	07/11/2023	1	3	16	30	60
Strength 002	55	30	05/12/2023	07/11/2023	2	4	13	30	60

Member

memberNumber	fname	lname	phoneNumber	levelId
1	Pika	Chu	5201112233	1
2	Smith	John	5208888888	2

Transaction

transactionNumber	total	transactionDate	type	memberNumber
1	100	05/06/2023	Paid	1
2	125	05/03/202	Unpaid	2

Equipment

itemNumber	description	quantity
1	football	20
2	basketball	10

Level

LevelId	minSpending	discount
Gold Member	1000	20
Diamon Member	500	30

CoursePackage

packageName	className	startDate
Beginner	Yoga 001	05/11/2023
Intermediate	Strength 002	07/11/2023

Purchase

memberNumber	courseName	transactionNumber
1	Yoga 001	1
2	Strength 002	2

EquipmentLoan

loanNumber	itemNumber	memberItem	quantity	borrowedDate	returnedDate
11	1	1	2	06/05/2023	Null
12	2	2	1	05/20/2023	06/20/2023

First Normal Form (1NF)

In the first normal form, the intersection of each row and column contains one and only one value. None of the intersections contains several values. Hence, our relations are in the first normal form.

Second Normal Form.

A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key is in the second normal form.

First let's determine functional dependencies:

fd1 packageName -> cost

fd2 trainerNumber -> fname, lname, phoneNumber

fd3 className, startDate, day -> maxParticipant, currentParticipant, endDate, trainerNumber, day, hour, minute, duration

fd4 memberNumber -> fName, lName, phoneNumber, levelId

fd5 transactionNumber -> total, transactionDate, type, memberNumber

fd6 itemNumber -> description, quantity

fd7 levelId -> minSpending, discount

fd8 memberNumber, courseName, transactionNumber

fd9 transactionNumber -> memberNumber

fd10 loanNumber -> itemNumber, memberItem, quantity, borrowedDate, returnedDate

fd11 className, startDate -> maxParticipant, currentParticipant, endDate, trainerNumber

We can skip relations with a single-attribute primary key because they are automatically in at least 2NF. We attach our attention to fd11 and fd3. We can see some non-key attributes (maxParticipant, currentParticipant, endDate, trainerNumber) are partially dependent on primary key. Hence, course should be split into two relations:

Course

Course (className, maxParticipant, currentParticipant, startDate, endDate, trainerNumber)

Primary Key: className, startDate

Schedule (className, startDate day, hour, minute, duration)

Primary Key: className, startDate, day

There are no partial dependencies. Hence we can stop here.

Third Normal Form

Package (packageName, cost)

This relation is in 3NF since there are no transitive dependencies on the primary key.

Tainer (trainerNumber, fName, lName, phoneNumber)

This relation is in 3NF. Each non-prime attribute is directly dependent on the primary key.

Course (className, maxParticipant, currentParticipant, startDate, endDate, trainerNumber)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Schedule (className, startDate, day, hour, minute, duration)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Member (memberNumber, fName, lName, phoneNumber, levelId)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Transaction (transactionNumber, total, transactionDate, type, memberNumber)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Equipment (itemNumber, description, quantity)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Level (levelId, minSpending, discount)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

CoursePackage (packageName, className, startDate)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Purchase (memberNumber, courseName, transactionNumber)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

EquipmentLoan (loanNumber, itemNumber, memberItem, quantity, borrowedDate, returnedDate)

This relation is in 3NF. All non-prime attributes are directly dependent on the primary key.

Hence, the final relations after the analysis are:

Package (packageName, cost)

Primary Key packageName

Tainer (trainerNumber, fname, lname, phoneNumber)

Primary Key trainerNumber

Course (className, maxParticipant, currentParticipant, startDate, endDate, trainerNumber)

Primary Key: className, startDate

Schedule (className, startDate day, hour, minute, duration)

Primary Key: className, startDate, day

Member (memberNumber, fname, lname, phoneNumber, levelId)

Primary Key memberNumber; Foreign Key: levelId

Transaction (transactionNumber, total, transactionDate, type, memberNumber)

Primary Key transactionNumber; Foreign Key: memberNumber

Equipment (itemNumber, description, quantity)

Primary Key itemNumber

Level (levelId, minSpending, discount)

Primary Key levelId

CoursePackage (packageName, className, startDate)

Primary Key: packageName className; Foreign Key: packageName, className, startDate

Purchase (memberNumber, courseName, transactionNumber)

Primary Key: memberNumber, courseName, transactionNumber;

Foreign Key: memberNumber, courseName, transactionNumber;

EquipmentLoan (loanNumber, itemNumber, memberItem, quantity, borrowedDate, returnedDate)

Primary key: loanNumber; Foreign Key: itemNumber, memberItem

Further Domain Analysis

Further domain and requirements analysis reveals that a member purchases a package not a course. Hence,

Purchase (packageName, transactionNumber)

Primary Key: courseName, startDate

Foreign Key: courseName, startDate

To keep track of enrollment, we should formulate a student enrolls in a course' relationship. Each student can enroll in 0 or more classes, and a class can contain zero or more students. It is many-to-many relationship.

Enrollment (memberNumber, courseName, startDate)

Primary Key: memberNumber, courseName, startDate

Foreign Key: memberNumber, courseName, startDate

For both relations, no non-candidate attribute partially depends on the primary key and no non-primary-key attribute is transitively dependent on the primary key. Hence, all relations are still in Third Normal Form (3NF).