Normalization Rules

And Denormalization Scenarios

Database for Software Developers

Anis Uddin Ahmad

Normalization

In Database Schema Design

Normalization is a process used to organize data efficiently in a database.

Normalization

In Database Schema Design

Normalization is a process used to organize data efficiently in a database.

- ☑ Avoid Anomalies
- ☑ Reduce Redundancy

Let’s take an Example Scenario

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2)
);
```

StudentCourses

student_id	*student_name*	*course_code*	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

Let’s take an Example Scenario

```sql
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2)
);
```

StudentCourses

student_id	*student_name*	*course_code*	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

- Redundancy
- Update Anomalies
- Deletion Anomalies

First Normal Form (1NF)

```sql
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2)
);
```

StudentCourses

student_id	*student_name*	*course_code*	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

First Normal Form (1NF)

1. The table has a primary key (Uniquely identify the records)
2. All attributes contain atomic (indivisible) values

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2)
);
```

StudentCourses

student_id	*student_name*	*course_code*	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

First Normal Form (1NF)

1. The table has a primary key (Uniquely identify the records)
2. All attributes contain atomic (indivisible) values

Action: make primary key with *student_id* and *course_code*.

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2)
);
```

StudentCourses

student_id	*student_name*	*course_code*	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

First Normal Form (1NF)

1. The table has a primary key (Uniquely identify the records)
2. All attributes contain atomic (indivisible) values

Action: make primary key with *student_id* and *course_code*.

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2),
    PRIMARY KEY
        (student_id, course_code)
);
```

StudentCourses

student_id	*student_name*	***course_code***	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

Second Normal Form (2NF)

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2),
    PRIMARY KEY
        (student_id, course_code)
);
```

StudentCourses

student_id	*student_name*	***course_code***	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

Second Normal Form (2NF)

1. The table is already in 1NF
2. All non-key attributes are fully functionally dependent on the entire primary key

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2),
    PRIMARY KEY
        (student_id, course_code)
);
```

StudentCourses

student_id	*student_name*	***course_code***	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

Second Normal Form (2NF)

1. The table is already in 1NF
2. All non-key attributes are fully functionally dependent on the entire primary key

Action: Move *student_name* and *course_name* to *students* and *courses* tables

```
CREATE TABLE StudentCourses (
    student_id INT,
    student_name VARCHAR(100),
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2),
    PRIMARY KEY
        (student_id, course_code)
);
```

StudentCourses

student_id	*student_name*	***course_code***	*course_name*	*instructor_name*	*instructor_phone*	*grade*
1	Zayd	101	Mathematics	Ahmad	17222	A
1	Zayd	102	Literature	Jahid	17333	B
2	Anas	101	Mathematics	Ahmad	17222	C
2	Anas	103	Physics	Khairul	17444	B

Second Normal Form (2NF)

1. The table is already in 1NF
2. All non-key attributes are fully functionally dependent on the primary key

Action: Move *student_name* and *course_name* to *students* and *courses* tables

```
CREATE TABLE StudentCourses (
    student_id INT,
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2),
    PRIMARY KEY (student_id, course_code)
);
```

```
CREATE TABLE students (
    id INT PRIMARY KEY,
    name VARCHAR(100),
);
```

```
CREATE TABLE courses (
    code INT PRIMARY KEY,
    name VARCHAR(100),
);
```

* Not defining foreign keys for simplifying the example

Second Normal Form (2NF)

1. The table is already in 1NF
2. All non-key attributes are fully functionally dependent on the primary key

Action: Move *student_name* and *course_name* to *students* and *courses* tables

StudentCourses

student_id	***course_code***	*instructor_name*	*instructor_phone*	*grade*
1	101	Ahmad	17222	A
1	102	Jahid	17333	B
2	101	Ahmad	17222	C
2	103	Khairul	17444	B

Students

id	*name*
1	Zayd
2	Anas

Courses

code	*name*
101	Mathematics
102	Literature
103	Physics

Third Normal Form (3NF)

1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

Third Normal Form (3NF)

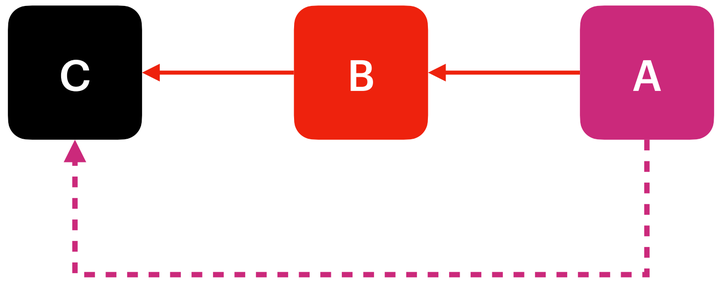
1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

Transitive dependency?

Third Normal Form (3NF)

1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

Transitive dependency?



One attribute in a table depends on another attribute through a third attribute.

Third Normal Form (3NF)

1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

StudentCourses

student_id	***course_code***	*instructor_name*	*instructor_phone*	*grade*
1	101	Ahmad	17222	A
1	102	Jahid	17333	B
2	101	Ahmad	17222	C
2	103	Khairul	17444	B

Students

id	*name*
1	Zayd
2	Anas

Courses

code	*name*
101	Mathematics
102	Literature
103	Physics

Third Normal Form (3NF)

1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

Action: Move *instructor_name* and *instructor_phone* to *instructors* tables

StudentCourses

student_id	***course_code***	*instructor_name*	*instructor_phone*	*grade*
1	101	Ahmad	17222	A
1	102	Jahid	17333	B
2	101	Ahmad	17222	C
2	103	Khairul	17444	B

Students

id	*name*
1	Zayd
2	Anas

Courses

code	*name*
101	Mathematics
102	Literature
103	Physics

Third Normal Form (3NF)

1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

Action: Move *instructor_name* and *instructor_phone* to *instructors* tables

```
CREATE TABLE StudentCourses (
    student_id INT,
    course_code INT,
    course_name VARCHAR(100),
    insructor_name VARCHAR(50),
    insructor_phone CHAR(15),
    grade CHAR(2),
    PRIMARY KEY (student_id, course_code)
);
```

```
CREATE TABLE students (
    id INT PRIMARY KEY,
    name VARCHAR(100),
);
```

```
CREATE TABLE courses (
    code INT PRIMARY KEY,
    name VARCHAR(100),
);
```

* Not defining foreign keys for simplifying the example

Third Normal Form (3NF)

1. The table is already in 2NF
2. All the attributes are not only dependent on the primary key but also non-transitively dependent

Action: Move *instructor_name* and *instructor_phone* to *instructors* tables

```
CREATE TABLE StudentCourses (
    student_id INT,
    course_code INT,
    course_name VARCHAR(100),
    instructor_id INT,
    grade CHAR(2),
    PRIMARY KEY (student_id, course_code)
);
```

* Not defining foreign keys for simplifying the example

```
CREATE TABLE students (
    id INT PRIMARY KEY,
    name VARCHAR(100),
);
```

```
CREATE TABLE courses (
    code INT PRIMARY KEY,
    name VARCHAR(100),
);
```

```
CREATE TABLE instructors (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    phone VARCHAR(100),
);
```

Questions?