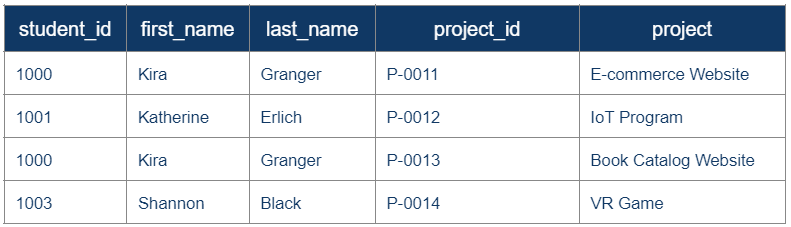
**Part 3**

1. **Task 6: Normalization Analysis from Students Table**

****

**Image 1 Students table**

#### First Normal Form (1NF)

Sebuah tabel dikatakan dalam **1NF** jika:

* Semua entri dalam kolom memiliki tipe data yang sama.
* Setiap kolom mengandung nilai yang atom (tidak bisa dibagi).
* Setiap kolom harus memiliki nama yang unik, **tidak boleh multivalue**.
* Urutan data yang disimpan tidak berpengaruh.

**Analisis:**

* Tabel student yang diberikan **memenuhi 1NF** karena semua kolom berisi nilai atom (misalnya, tidak ada daftar atau himpunan dalam satu kolom), dan setiap kolom memiliki nama yang unik.

#### Second Normal Form (2NF)

Sebuah tabel dikatakan dalam **2NF** jika:

* Tabel tersebut sudah dalam 1NF.
* Memenuhi ***full functional depedencies***. Semua *non-key attributes* (bukan PK) sepenuhnya bergantung pada *primary key* (PK).

**Analisis:**

* *Composite primary key* di sini dapat dianggap (**student\_id, project\_id**) karena seorang siswa dapat memiliki beberapa proyek.
* Kolom first\_name, last\_name, dan project tidak sepenuhnya bergantung pada *composite key* Misalnya, informasi siswa (first\_name dan last\_name) hanya bergantung pada student\_id, sedangkan project bergantung pada project\_id. Jadi, tabel ini **tidak memenuhi 2NF**.

**Update**:

* Mengubah ke bentuk 2NF dengan membagi table menjadi 2, yakni table students dan table projects.

1. **Tabel Students**

|  |  |  |
| --- | --- | --- |
| **student\_id** | **first\_name** | **last\_name** |
| 1000 | Kira | Granger |
| 1001 | Katherine | Erlich |
| 1003 | Shannon | Black |

**DDL**:

**CREATE** **TABLE** students (

student\_id **INT** **PRIMARY** **KEY**,

first\_name **VARCHAR**(50),

last\_name **VARCHAR**(50)

);

1. **Table Projects**

|  |  |  |
| --- | --- | --- |
| **project\_id** | **student\_id** | **project** |
| P-0011 | 1000 | E-commerce Website |
| P-0012 | 1001 | IoT Program |
| P-0013 | 1000 | Book Catalog Website |
| P-0014 | 1003 | VR Game |

**DDL**:

**CREATE** **TABLE** projects (

project\_id **VARCHAR**(10) **PRIMARY** **KEY**,

student\_id **INT**,

project **VARCHAR**(100),

**FOREIGN** **KEY** (student\_id) **REFERENCES** Students(student\_id)

);

#### Third Normal Form (3NF)

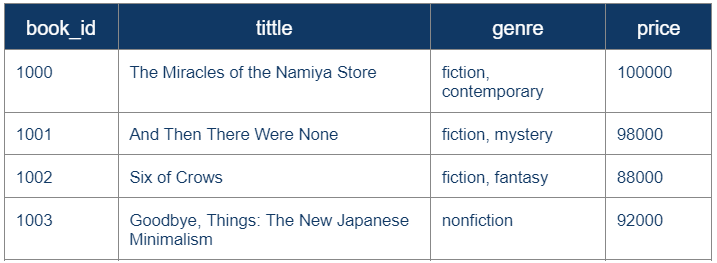
Sebuah tabel dikatakan dalam **3NF** jika:

* Tabel tersebut sudah dalam 2NF.
* Tidak ada ketergantungan transitif (tidak ada *non-key attribute* yang bergantung pada *non-key attribute* lainnya).

**Analisis:**

* Sebelumnya table students tidak memenuhi 2NF, maka tidak bisa memenuhi 3NF. Setelah dibagi lagi menjadi table students dan table pojects maka sudah memenuhi 2NF maka bisa dilakukan pengecekan ke 3NF.
* **Table students** **sudah dalam 3NF** karena tidak ada ketergantungan transitif. Semua kolom bergantung hanya kepada kolom **student\_id**. Dan **table projects** juga **sudah memenuhi 3NF** karena project bergantung hanya pada **project\_id** dan tidak ada ketergantungan transitif.
* Sudah tidak ada redundansi.

1. **Task 7 : Normalization Analysis from Students Table**

****

**Image 2 Books table**

1. **First Normal Form (1NF)**

**Analisis:**

* Tabel di atas **tidak memenuhi 1NF** karena kolom genre berisi nilai yang tidak atom (misalnya, "fiction, contemporary"). Seharusnya setiap nilai genre harus berada di baris terpisah agar memenuhi syarat 1NF.

**Update**:

* Kolom genre harus atom, jadi jika sebuah film masuk kategori lebih dari 1 genre kita buat di row berikutnya saja.

|  |  |  |  |
| --- | --- | --- | --- |
| book\_id | title | genre | price |
| 1000 | The Miracles of the Namiya Store | fiction | 100000 |
| 1000 | The Miracles of the Namiya Store | contemporary | 100000 |
| 1001 | And Then There Were None | fiction | 98000 |
| 1001 | And Then There Were None | mystery | 98000 |
| 1002 | Six of Crows | fiction | 88000 |
| 1002 | Six of Crows | fantasy | 88000 |
| 1003 | Goodbye, Things: The New Japanese Minimalism | nonfiction | 92000 |

1. **Second Normal Form (2NF)**

**Analisis:**

* Sebelum update, table books sudah otomatis tidak memenuhi 2NF. Setelah update kolom genre harus atom maka sudah memenuhi 1NF dan bisa dilakukan pengecekan 2NF.
* Primary key nya disini berupa composite key (book\_id, genre)
* Semua non\_key attribute (title, price) bergantung pada composite key.
* Karena title dan price bergantung hanya pada book\_id, maka ada ketergantungan parsial. Berarti table update di atas **tidak memenuhi 2NF**.

**Update**:

* Untuk mengubah tabel books menjadi 2NF, maka perlu menghilangkan ketergantungan parsial dengan memisahkan *non-key attributes* ke table lain.

1. **Table Books**

|  |  |  |
| --- | --- | --- |
| book\_id | title | price |
| 1000 | The Miracles of the Namiya Store | 100000 |
| 1001 | And Then There Were None | 98000 |
| 1002 | Six of Crows | 88000 |
| 1003 | Goodbye, Things: The New Japanese Minimalism | 92000 |

**DDL:**

**CREATE** **TABLE** books (

book\_id **INT** **PRIMARY** **KEY**,

title **VARCHAR**(100),

price **INT**

);

1. **Table Genres**

|  |  |  |
| --- | --- | --- |
| genre\_id | book\_id | genre |
| 1 | 1000 | fiction |
| 2 | 1000 | contemporary |
| 3 | 1001 | fiction |
| 4 | 1001 | mystery |
| 5 | 1002 | fiction |
| 6 | 1002 | fantasy |
| 7 | 1003 | nonfiction |

**DDL:**

**CREATE** **TABLE** genres (

genre\_id SERIAL **PRIMARY** **KEY**,

book\_id **INT**,

genre **VARCHAR**(50),

**FOREIGN** **KEY** (book\_id) **REFERENCES** books(book\_id)

);

#### **Third Normal Form (3NF)**

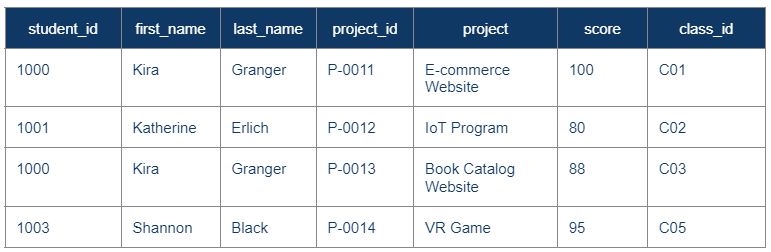
**Analisis:**

* **Table Books:** kolom title dan price bergantung hanya pada book\_id. Jadi tidak ada ketergantungan transitive. Table ini **sudah memenuhi 3NF.**
* **Table genres**: genre bergantung hanya pada genre\_id dan book\_id berfunsgi sebagai foreign key (FK). Tidak ada ketergantungan transitive, jadi tabel ini **sudah memenuhi 3NF**.

1. **Task 8 : Modify Table Structure of Students Table**

Table structure in number 6 should be modified so that it may record information about the **class** that provided the **project assignment as well as the score of the project the student worked on.** Implement the table structure that was created using the ddl syntax.

1. **Update Table Structure**



My assumption is that this is the table before normalization. So we need to add new columns, namely score and class\_id. The table structure will include the following columns:

* student\_id: The unique identifier for the student.
* first\_name: The first name of the student.
* last\_name: The last name of the student.
* project\_id: The unique identifier for the project.
* project: The name of the project.
* score: The score achieved for the project.
* class\_id: The identifier for the class associated with the project.

|  |  |  |
| --- | --- | --- |
| **students** | | |
| student\_id: int |  | PK |
| first\_name: varchar(100) | Not null |  |
| last\_name: varchar(100) | Not null |  |
| project\_id: varchar(10) | Not null | PK |
| project: varchar(100) | Not null |  |
| score : int | Between 0 and 100 |  |
| class\_id: varchar(10) | Not null | FK |

1. **DDL**

ALTER TABLE students

ADD COLUMN score INT NOT NULL CHECK (score >= 0 AND score <= 100),

ADD COLUMN class\_id VARCHAR(10) NOT NULL,

ADD CONSTRAINT fk\_class FOREIGN KEY (class\_id) REFERENCES classes(class\_id);