**SRS Documentation**

**Notown Music Record Management System**

**(Assignment 3)**

Prepared by,

**DEBOPRIYO GHOSH Roll: 35**

**SAIKAT JANA Roll: 50**

**MOUSUMI MONDAL Roll: 40**

**SARADINDU RANA Roll: 28**

**B.Tech., Sem - V, Group- A**

**Dated: 07/04/2021**

**Contents**

**Assignment 3**

1. **Data Requirement**
2. **Assumptions**
3. **Entity - Relationship Diagram**
4. **Relationships**

**5. Relational Schema**

**6. Normalization of Relational Schema**

**7. Data Dictionary**

**8. User Interface of Implementation**

**Assignment 3**

**Draw the ER diagram; Design the relational schema with minimum redundancy.**

**Notown Records has decided to store information about musicians who perform on its albums (as well as other company data) in a database. The company has wisely chosen to hire you as a database designer (at your usual consulting fee of $2,500/day).**

**Each musician that records at Notown has an SSN, a name, an address, and a phone number. Poorly paid musicians often share the same address, and no address has more than one phone.**

**Each instrument that is used in songs recorded at Notown has a name (e.g., guitar, synthesizer, flute) and a musical key (e.g., C, B-flat, E-flat).**

**Each album that is recorded on the Notown label has a title, a copyright date, a format (e.g., CD or MC), and an album identifier.**

**Each song recorded at Notown has a title and an author.**

**Each musician may play several instruments, and a given instrument may be played by several musicians.**

**Each album has a number of songs on it, but no song may appear on more than one album.**

**Each song is performed by one or more musicians, and a musician may perform a number of songs.**

**Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.**

**1.Data Requirement**

* **Songs:** Every song has a unique title. Each song is written by specific author and in a specific album.
* **Albums:** Every album has a unique id and respective album name and a producer. Each one has specific format (e.g. CD or MC, or DVD), date of copyright.
* **Musicians:** Every musician has an unique Ssn number and their respective name.
* **Instruments:** Each instrument is identified by an unique name (e.g. guiter, synthesizer, flute ) and its musical key (e.g. C,B-flat ,E-flat ).
* **Addresses:** Each address of musicians identified by a contact phone number. Each one has full address.

**2. Assumption**

1. An album may contain multiple songs. But the recorded albums has at least one song.
2. A musician may or may not perform songs. Musician may produce one or more albums.
3. A musician can play one or more instruments.
4. A musician lives only in one address identified by it’s phone number.
5. In one address there may be more than one poorly paid musicians.
6. An instrument can be played by one or more than one musicians.
7. Every album has a producer.
8. A song can be performed by one or more musicians.

**3. Entity Relationship Diagram**

**A. Diagram**



**B. Description**

1. In this diagram the entities are Albums, Musicians, Songs , Instruments , Addresses.
2. Albums contains songs so they are connected by the relationship ‘Contains’.
3. Musicians performs songs. They are connected by the relationship ‘Performs’.
4. Musicians produces album, so they are connected by the relationship ‘Produces’.
5. A musician lives in an addresses. So musician and addresses are connected by the relationship ‘Lives’.
6. Musician plays instruments, so musician and instruments is connected by the relationship ‘Plays’.

**C. Attributes**

Albums: {A\_Id, A\_Name, Format, Copyright\_Date}

Songs: {Title, Author}

Musicians: {Ssn, Name, Existance}

Instruments: {Ins\_Name, MusicalKey}

Addresses: {Ph\_no, FullAddress}

**4.Relationships**

a. Musicians-Addresses (N:1) comprises.

b. Musicians-Instruments (N:M) comprises.

c. Musicians-Albums (1:N) comprises.

d. Musicians-Songs (N:M) comprises.

e. Albums-Songs (1:N) holds.

**5.Relational Schema**

**Albums:**

Albums

<A\_Id>

Album title: <A\_Name>

Date of copyright : <Copyright\_Date>

Album format : <Format>

Producer id : <Ssn> (Forgien key from ‘Musicians’ table)

**Songs:**

Songs

<Title>

Author name : <Author>

Album id : <A\_Id> (Forgien key from ‘Albums’ table)

**Musicians:**

Musicians

<Ssn>

Musician name : <Name>

Musician contact number : <Ph\_No>(Forgien key from ‘Addresses’ table)

**Instruments:**

Instruments

<Ins\_Name>

Musical key type : <Musicalkey>

**Addresses:**

Addresses

<Ph\_No>

Musician address : <FullAddress>

**Plays:**

Plays

Instrument name : <Ins\_name> (Forgien key from ‘Instruments’ table)

Musician id : <Ssn> (Forgien key from ‘Musicians’ table)

**Performs:**

Performs

Musician id : <Ssn> (Forgien key from ‘Musicians’ table)

Song title : <Title> (Forgien key from ‘Songs’ table)

**6. Normalization of Relational Schema**

1. **Musicians {Ssn, Name,** **Ph\_No }**

Ssn -> { Name,Ph\_No }

Ssn is the candidate key.

There is no multi valued attribute so the table is in **1NF**.

Ssn is the primary key and there is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

1. **Songs { Title, Author, A\_Id }**

Title -> { **Author, A\_Id** }

Title is the candidate key.

There is no multi valued attribute so the table is in **1NF**.

Title is the primary key and there is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

1. **Albums{A\_id, A\_Name, Format, Copyright\_Date, Ssn}**

A\_Id -> { A\_name, Format, Copyright\_Date ,Ssn}

A\_Id is the candidate key.

There is no multi valued attribute so the table is in **1NF**.

A\_Id is the primary key and there is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

1. **Instruments { Ins\_Name, Musicalkey}**

(Ins\_Name, MusicalKey} is the composite primary key.

There is no multi valued attribute so the table is in **1NF**.

There is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

1. **Addresses {Ph\_No, FullAddress}**

Ph\_No -> { FullAddress }

Ph\_No is the candidate key.

There is no multi valued attribute so the table is in **1NF.**

Ph\_No is the primary key and there is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

1. **Performs{Ssn,Title}**

{Ssn,title} is a composite primary key.

There is no multi valued attribute so the table is in **1NF**.

There is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

1. **Plays{Ssn,Ins\_Name}**

{Ssn,Ins\_Name} is a composite primary key.

There is no multi valued attribute so the table is in **1NF**.

There is no partial dependency, so the table is in **2NF**.

The table is in **3NF** as there is no transitive dependency.

**7.Data Dictionary:**

Table: albums

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***A\_Id*** | int(4) | No |  | Album id |
| A\_Name | text | No |  | Name of the album |
| Copyright\_Date | date | No |  | Album’s copyright date |
| Format | varchar(255) | No |  | Format of album |
| Ssn | int(4) | No |  | Producer id |

Table: songs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***Title*** | varchar(255) | No |  | Song title or name |
| Author | text | No |  | Author of the song |
| A\_Id | int(4) | No |  | Id of the album in which song belongs |

Table: instruments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***Ins\_Name*** | varchar(255) | No |  | Musical instrument name |
| Musicalkey | varchar(255) | No |  | Keys of musical instrument |

Table: musicians

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***Ssn*** | int(4) | No |  | Musician id |
| Name | text | No |  | Full name of musician |
| Ph\_No | varchar(12) | No |  | Phone number of musician’s permanent address |
| Existance | text | No | 'Exist' | Musician connected with Notown or not |

Table: addresses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***Ph\_No*** | varchar(12) | No |  | Phone number for a particular address |
| FullAddress | text | No |  | Full address |

l

Table: plays

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***Ins\_name*** | varchar(255) | No |  | Musical instrument name |
| ***Ssn*** | int(4) | No |  | Musician id |

Table: performs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| ***Ssn*** | int(4) | No |  | Musician id |
| ***Title*** | varchar(255) | No |  | Song title |

**8 . User Interface of Implementation:**















