Entities

University:
The university entity represents each university. It has an associated UId which serves as the primary key, and has attributes for the university's name and location.

People:
The person entity represents a researcher. It stores a PersonId as the primary key, as well as their first and last names, UId of the university where they did their PhD.
ROutput:
The ROutput table describes researcher's outcomes, such papers, posters, etc. ROutput table takes the RId as the primary key to uniquely identify each outcome. ROutput also contains publisher, year, and Title of the outcome as the attributes.

Topics:
Topics represent possible topics of research outputs. There is a TopicId which serves as the primary key, and the name and categories are the attributes of topics.

User:
The User entity represents each user of our program. A user will have a username and email which will serve in combination as the primary key, as well as the user's name and password.

Music:
Music table represents the background music which will be played when the user reads the information of the outputs. Music table takes MusicId as the primary key. It also takes Genre, MusicPath, MusicName, as attributes.

Relationships

Student/Employee: The people are connected to the university through the UIds in the researchers' PhdUId and EmployeeId attributes. A person has to be associated with one university through their PhD Id. A university can be associated with many people.

ROutput/Topics: This relationship is one-to-many relationship for both directions. Because for each output, it can be involved in multiple topics. For each topic, there should be multiple papers in the field as well.

Paper Topics: The relationship is one-to-many relationship for both directions. Because a research output can contain multiple topics and for each topics there are multiple research outputs associated with.

Topic Music: Each topic has an associated genre of music, but genres can be shared between different topics.

User/Topics: This relationship is one-to-many relationship. For each user, they can be interested in many topics. Also for each topic, there could be multiple users who are interested in the topic.

Normalization Procedure:
TopicId -> TopicId, TopicName, Category
MusicId -> MusicId, Genre, MusicPath, TopicId
UserId -> UserId, UserName, Password, Email
Uid -> UId, UniName, Location
RId -> RId, Title, Year, Publisher
PersonId -> PersonId, FirstName, LastName, PhdUId

(RId, PersonId) -> RId, PersonId, FirstName, LastName, PhdUId, EmployeeUId, Title, Year, Publisher
(TopicId, RId) -> TopicId, TopicName, Category, RId, Title, Year, Publisher
(TopicId, UserId) -> TopicId, TopicName, Category, UserId, UserName, Password, Email
(TopicId, MusicId) -> TopicId, TopicName, Category, MusicId, Genre, MusicPath, TopicId
(PhdUId, PersonId) -> PersonId, FirstName, LastName, PhdUId, UId, UniName, Location

There are no problematic functional dependencies, thus there is no need to proceed with the BCNF or 3NF algorithms.

Relational Schema

Your relational schema should be formatted as follows:
Table-Name(Column1:Domain [PK], Column2:Domain [FK to table.column], Column3:Domain,...)

PK: Indicates that the column is a primary key for the table
FK: Indicates that the column is a foreign key referencing the primary key of table.column.
Domain: INT, Decimal, VARCHAR(X),....

Person(PersonId:INT [PK] [FK to ResearcherOuput.PersonId, FirstName:VARCHAR(15), LastName:VARCHAR(15), UId:INT [FK to Universities.UId])

Universities(UId:INT [PK], Name:VARCHAR(50), Location:VARCHAR(100))

ROutput(RId:INT [PK] [FK to ResearchOutput.RId] [FK to PaperTopics.RId], Title:VARCHAR(100), Year:INT, Publisher:VARCHAR(100), TopicsList:VARCHAR(100), PeopleList:VARCHAR(200))

ResearcherOutput(PersonID:INT [PK] [FK to Person.PersonId], RId:INT [PK] [FK to ROutput.RId])

Topics(TopicId:INT [PK] [FK to PaperTopics.TopicId] [FK tp Interest.TopicId] [FK to TopicMusic.TopicId], TopicName:VARCHAR(15), Category:VARCHAR(20))

PaperTopics(TopicId:INT [PK] [FK to Topics.TopicId], RId:INT [PK] [FK to ROutput.RId])

Interest(TopicId:INT [PK] [FK to Topics.TopicId], UserId:INT [PK] [FK to User.UserId])

Users(UserId:INT [PK] [FK to Interest.UserId], UserName:VARCHAR(50), Password:VARCHAR(30), Email:VARCHAR(30) [PK])

TopicMusic(TopicId:INT [PK] [FK to Topics.TopicId], MusicId[INT] [FK to Music.MusicId])

Music(MusicId:INT [PK] [FK to TopicMusic.MusicId], Genre:VARCHAR(30), MusicPath:VARCHAR(30))