**Assignment 1**

**​​1DV503 Database Technology**

**Kaja Gülzau**

School of Computer Science, Physics

and Mathematics, Linnaeus University, Sweden

[kg222tj@student.lnu.se](mailto:your.email@address.cm)

# Part 1

## **Task 1 Music Streaming Service Database (15 points)**

***1.1******Identify all entities and their attributes from the description of database requirements using the following Table template:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attribute** | **Attribute Type** | **Is a Key Attribute?** | **The value type of attributes and constraints (data type, NULL/NOT NULL, Unique)** |
| User | User-ID | Simple | Yes | Unique, String, not null |
| Birthday | Simple | No | Date (YYYY-MM-DD), null |
| Profile Image | Simple | No | String (url path), null |
| Password | Simple | No | String, not null |
| Email | Simple | Yes | Unique, String, not null |
| Full Name  (FirstName and LastName) | Composite | No | String, not null |
| Subscription (weak) | Subscription-ID | Simple | Yes | Unique, String, not null |
| Type | Simple | No | String, free or premium, not null |
| Playlist | Playlist creation Date | Simple | No | Date (YYYY-MM-DD), not null |
| Playlist-ID | Simple | Yes | Unique, String, not null |
| Playlist Name | Simple | No | String, not null |
| Artist | Artist-ID | Simple | Yes | Unique, String, not null |
| Artist Name | Simple | No | String, not null |
| Genre | Multi-valued | No | String, null |
| Artist Image | Simple | No | String (url path), null |
| Album | Album-ID | Simple | Yes | Unique, String, not null |
| Album Title | Simple | No | String, not null |
| Album Release Date | Simple | No | Date (YYYY-MM-DD), not null |
| Album Image | Simple | No | String (url path), null |
| Track | Track-ID | Simple | Yes | Unique, String, not null |
| Track title | Simple | No | String, not null |
| Duration | Simple | No | Int, not null |

***1.2 Identifying the relationship between entities defined in Task 1.1 using the following table template:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity A** | **Relationship name** | **Entity B** | **Cardinality Ration**  **(1:1, 1:N, N:1, M:N)** | **Attribute of Relationship**  **(optional)** | **Justify your decision for the cardinality ratio**  **(mandatory)** |
| User | Has | Subscription | 1:1 | Subscription Start Date, Subscription End Date | Every user can only have one subscription at a time and every subscription can only belong to one user |
| User | Creates | Playlist | 1:N |  | A user can create one or several playlists and each playlist only belongs to one user, however this relationship could also be M:N if the music streaming service allows several users to create a playlist together |
| Playlist | Includes | Track | M:N |  | Playlist can include zero or several tracks, Track can be in several or no playlists |
| Track | Belongs\_To | Album | M:N |  | Track can be part of several or no albums, Albums can include one or more tracks |
| Artist | Releases | Album | M:N |  | Artist can release zero or several albums, Albums belong to one or more artists |
| Artist | Is | User | 1:1 |  | Every artist is also a user and every user account can only belong to one artist |

***1.3 Design an ER schema for the music streaming service database based on information provided in task 1, and entities defined in 1.1 with relationships defined in 1.2.***

Note: Artist-ID and Playlist-ID are key attributes and Subscription-ID a partial key attribute, but the website kept removing the underline, despite me selecting the “key attribute” / ”weak key attribute” type. I tried several times to fix this issue, but even me manually marking the text and selecting the “underline” option did not work, the website would immediately remove it. When selected, the attribute still shows up underlined, it just changes as soon as I click away from it. I have added a screenshot of an example of this in order to clear up any confusion.

A black background with blue dots and white text

Description automatically generated

A diagram of a company

Description automatically generated

## **Task 2 University Social Network Database (15 points)**

***2.1******Identify all entities and their attributes from the description of database requirements using the following Table template:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Attribute** | **Attribute Type** | **Is a Key Attribute?** | **The value type of attributes and constrains (data type, NULL/NOT NULL, Unique)** |
| Student | Student-ID | Simple | Yes | Unique, String, not null |
| StudentName  (FirstName, LastName) | Composite | No | String, not null |
| E-Mail | Simple | Yes | Unique, String, not null |
| Password | Simple | No | String, not null |
| Birthday | Simple | No | Date (YYYY-MM-DD), not null |
| Hobby | Multivalued | No | String, null |
| Gender | Simple | No | String, not null |
| StudyProgram | Simple | No | String, not null |
| JoinDate | Simple | No | Date (YYYY-MM-DD) |
| StudentImage | Simple | No | String (url path), null |
| Study\_Group | Group-ID | Simple | Yes | String, Unique, not null |
| GroupName | Simple | No | String, not null |
| GroupSubject | Simple | No | String, not null |
| CreationDate | Simple | No | Date (YYYY-MM-DD), not null |
| Event | Event-ID | Simple | Yes | String, Unique, not null |
| EventName | Simple | No | String, not null |
| EventDescription | Simple | No | String, not null |
| EventAddress (Street, PostalCode) | Composite | No | String, not null |
| EventDate | Simple | No | Date (YYYY-MM-DD), not null |
| EventTime | Simple | No | String (Time), not null |
| Post | Post-ID | Simple | Yes | String, Unique, not null |
| PostText | Simple | No | String, not null |
| PostImage | Simple | No | String (url path), null |
| PostLocation | Simple | No | String, null |
| PostDate | Simple | No | Date (YYYY-MM-DD), not null |

***2.2 Identifying the relationship between entity sets using the following table template:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity A** | **Relationship name** | **Entity B** | **Cardinality Ration**  **(1:1,1:N,N:1,M:N)** | **Attribute of Relationship**  **(optional)** | **Justify your decision for the cardinality ratio**  **(mandatory)** |
| Student | Befriends | Student | N:M | Status, Start\_Date | Every student can have several friends |
| Student | Creates | Study\_Group | 1:N |  | Student can create more than one study group, but a study group can’t be created by more than one student |
| Student | Joins | Study\_Group | M:N | GroupJoinDate | Students can join several study groups and study groups can have more than one student in them |
| Student | Hosts | Event | 1:N |  | Students can host several events, but there can only be one host per event  Note: This could also be M:N if the university social network allows several students to host events together, I simply decided on 1:N to make it easier |
| Student | Writes | Post | 1:N |  | Students can create more than one post, but posts can’t have more than one author |
| Student | Comments\_On | Post | M:N | CommentText | Students can comment on more than one post and a post can have comments from more than one student |

***2.3 Design an ER schema for the university social network database based on information provided in task 2, and entities defined in 2.1 with relationships defined in 2.2.***

Note: I ran into the same problem as in task 1.3, Group-ID, Event-ID and Post-ID are supposed to be underlined key attributes.

A diagram of a company

Description automatically generated

## **Task 3 Airport Management Database (10 points)**

***3.1******Given the constraints shown in the ER schema below, respond to the following statements with True, False, or Maybe.***

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **Statement** | **True/False/Maybe** | **Justify your answer (mandatory)** |
| **1** | Every pilot has been a passenger on at least one flight. | Maybe  (or False) | The pilot only has partial participation in being a passenger or deadheading a flight, which points to the answer being “False”. However, from a realistic standpoint, it is very likely that every pilot has been a passenger on at least one flight, even just as part of their (or another pilot’s) training. Due to this reason, I decided to answer “Maybe”. |
| **2** | Every flight has at least one deadheading pilot | False | The “Deadheading” relationship only has partial participation from both the pilot and the flight. |
| **3** | Every flight has 2 pilots. | False/Maybe | The flight has total participation in the “Flies” relationship and therefore has at least one pilot and a maximum of two.  Legal requirements call for two pilots per flight for most airplanes, but in case of a very small aircraft it would be allowed to have only one pilot (<https://flygv.com/blog/the-importance-and-benefits-of-utilizing-dual-pilot-operations>), so the answer depends on what kind of aircrafts are being handled by this database. Assuming any size of aircraft is included, the answer would be “False” but considering the fact that all the aircrafts are owned by airlines (see the circular dependency of the “owns” relationship) instead of also being private jets, which tend to be smaller, it is highly probable that all the aircrafts are big enough to require two pilots.  Without having more information on the type of aircrafts, it is not possible to give a definitive answer, which is why I also included “Maybe” as an answer. |
| **4** | Each flight must have an assigned departure and arrival airport. | True | The flight has total participation in both the “Arrival” and “Departure” relationship. |
| **5** | Every pilot has flown at least 2 times | False | There is no information on the minimum amount of flights a pilot has to have made, only the maximum (n --> no limit), a pilot could be doing their first flight |
| **6** | Some tickets that do not belong to any flight | False | Tickets have total participation in the “Belongs To” relationship, so they have to belong to a flight |
| **7** | Some airlines do not have flights. | False | Airlines have total participation in the “Belong To” relationship, so they have to have at least one flight |
| **8** | A passenger can be a pilot. | True | The “Is A” relationship has partial participation for both the pilot and the passenger, so a passenger CAN (but doesn’t have to be) a pilot and the same is true in reverse. |
| **9** | Some tickets do not belong to any class type (Economy, Business, etc.) | False | Tickets have to have a class due to having total participation in their relationship to “Class” |
| **10** | Some tickets are without payment | True | Tickets have partial participation in the “Receives” relationship and therefore don’t necessarily have to have a payment |

***3.2 List the (min, max) constraint using the total/partial participation for the Ticket entity. Justify your answer.***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity | Relation | (min,max) | Explanation |
| Payment | Receives | (1,1) | The payment has total participation (min: 1) in the 1:1 (max: 1) relationship with the ticket |
| Ticket | Receives | (0,1) | The ticket only has partial participation (min: 0), the max constraint stays the same |
| Flight | Belongs To | (1, N) | The flight has total participation (min: 1) and a 1:N cardinality ratio (max: N) |
| Ticket | Belongs To | (1, 1) | The ticket has full participation (min: 1), but is on the other side of the relationship, so the “1” from the cardinality applies (max: 1) |
| Class | Has | (0, N) | The class has partial participation (min: 0) and a 1:N cardinality (max: N) |
| Ticket | Has | (1, 1) | The ticket has full participation (min: 1) and is on the other side of the cardinality (max: 1) |
| Passenger | Book | (0, N) | The passenger has partial participation (min: 0) and a 1:N cardinality (max: N) |
| Ticket | Book | (1, 1) | The ticket has full participation (min: 1) and is on the other side of the cardinality (max: 1) |

Part 2

## **Task 4. Conver ER model to Relational Model (15 points)**

Note: I combined “User” and “Subscription” into one table according to slide 12 of lecture 4, since it was a circular dependency. I tried to follow the example from the slide, which is why I wrote “SUBSCRIPTION” and left out the primary key (“Subscription-ID”) of the subscription entity.

A diagram of a computer program

Description automatically generated with medium confidence

## **Task 5 Normalisation (15 points)**

**5.1** ***Considering unnormalized Pizzeria database with one table is shown below on image below, perform normalisation to achieve 3NF for all tables***.

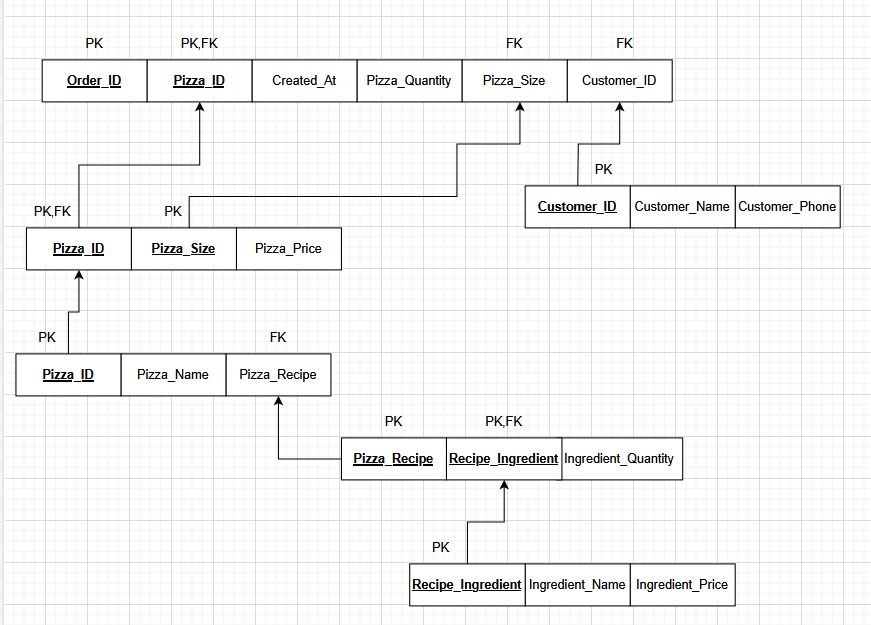
A diagram of a project

Description automatically generated with medium confidenceUnnormalized database:

A diagram of a company

Description automatically generatedNormalized version:

***5.2 Consider adding pizza size (small, medium, large) attribute to the Order table, and the pizza price will be different for each pizza size. Make sure that changes and new tables satisfy the 3NF.***



# Part 3

## **Task 6. Basic SQL queries (20 points)**

|  |  |
| --- | --- |
| **Query** | **Answer** |
| 1.1 retrieves all artists whose names start with the same letter as the first letter of your first name. | SELECT Artist.Name  FROM Artist  WHERE Artist.Name LIKE 'K%';  Rows: 4 |
| 1.2 retrieves all tracks by “BackBeat” artists. | SELECT Track.Name, Artist.Name AS artist  FROM Track  JOIN Album ON Track.AlbumId=Album.AlbumId JOIN Artist ON Album.ArtistId=Artist.ArtistId  WHERE Artist.Name='BackBeat';  Rows: 12 |
| 1.3 lists all album titles along with their respective artist name. | SELECT Album.Title, Artist.Name  FROM Album  JOIN Artist ON Album.ArtistId=Artist.ArtistId;  Rows: 347 |
| 1.4 Lists the total number of tracks in each album, sorted by total in descending order. | SELECT Album.Title, Count(Track.Name) AS Total\_Tracks  FROM Album  JOIN Track ON Album.AlbumId=Track.AlbumId  GROUP BY Album.Title  ORDER BY Total\_Tracks DESC;  Rows: 347 |
| 1.5 finds tracks with “Protected AAC audio file” media type | SELECT Track.Name, MediaType.Name  FROM Track  JOIN MediaType ON Track.MediaTypeId=MediaType.MediaTypeId  WHERE MediaType.Name='Protected AAC audio file';  Rows: 237 |
| 1.6 Lists all tracks from “Big Ones” album | SELECT Track.Name, Album.Title  FROM Track  JOIN Album ON Track.AlbumId=Album.AlbumId  WHERE Album.Title='Big Ones';  Rows: 15 |
| 1.7 finds the total duration of tracks in each playlist. You'll need to use the *Playlist*, *PlaylistTrack*, and *Track* tables for this. | SELECT Playlist.Name, SUM(Track.Milliseconds) AS Total\_Duration  FROM Playlist  JOIN PlaylistTrack ON Playlist.PlaylistId=PlaylistTrack.PlaylistId JOIN Track ON PlaylistTrack.TrackId=Track.TrackId  GROUP BY Playlist.Name  ORDER BY Total\_Duration DESC;  Rows: 12 |
| 1.8 finds the 10 most expensive tracks. | SELECT Track.Name, Track.UnitPrice  FROM Track  ORDER BY Track.UnitPrice DESC  LIMIT 10;  Rows: 10 |
| 1.9 finds the list of artists who do not have any albums. | SELECT Artist.Name  FROM Artist  WHERE NOT EXISTS (SELECT \* FROM Album WHERE Artist.ArtistId=Album.ArtistId);  Rows: 71 |
| 1.10 lists all playlists and the number of tracks in each. | SELECT Playlist.Name, Count(PlaylistTrack.TrackId) AS Total\_Tracks  FROM Playlist  JOIN PlaylistTrack ON Playlist.PlaylistId=PlaylistTrack.PlaylistId  GROUP BY Playlist.Name;  Rows: 12 |

## 

## 

## 

## **Task 7. Advanced SQL queries (10 points)**

|  |  |
| --- | --- |
| **Query** | **Answer** |
| 2.1 finds albums with multiple genres. |  |
| 2.2 identifies customers with the highest average invoice value. | SELECT Customer.FirstName, Customer.LastName, AVG(Invoice.Total) AS Average\_Invoice  FROM Customer  JOIN Invoice ON Customer.CustomerId=Invoice.CustomerId  GROUP BY Customer.LastName  ORDER BY Average\_Invoice DESC;  Rows: 1 |
| 2.3 Find the Longest Playlist by Duration. | SELECT Playlist.Name, SUM(Track.Milliseconds) AS Total\_Duration  FROM Playlist  JOIN PlaylistTrack ON Playlist.PlaylistId=PlaylistTrack.PlaylistId JOIN Track ON PlaylistTrack.TrackId=Track.TrackId  GROUP BY Playlist.Name  ORDER BY Total\_Duration DESC  LIMIT 1;  Rows: 1 |
| 2.4 finds customers with Maximum Purchase in Each Country. |  |
| 2.5 find the most popular album purchased by each country. Order the results by number of Sales in descending order. Print the country, album name, and maximum sales |  |

OBS! DO NOT FORGET ALSO TO SUBMIT THE .SQL SCRIPT FILE CONTAINING ALL SQL QUERRIES