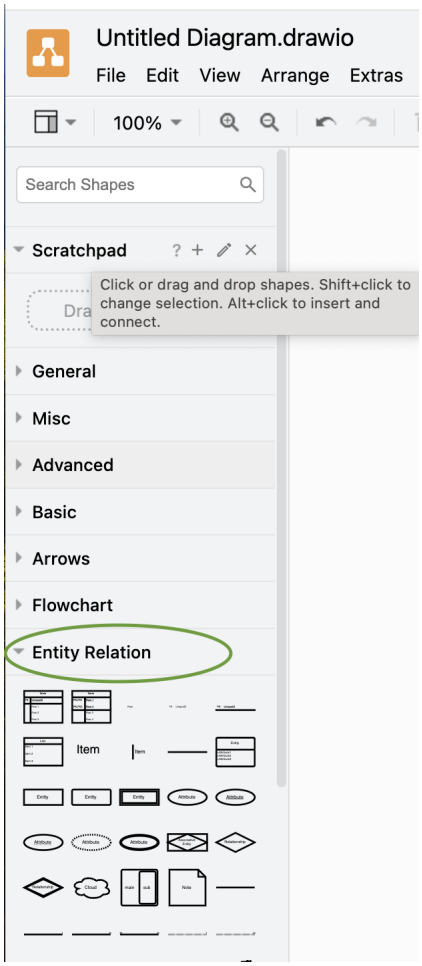
ASSIGNMENT 1

Part 1. Conceptual Data Modelling using Entity Relationship (ER) Diagrams (*45 points*)

Description
In Part 1, you will design two Entity-Relationship (ER) models and analyze one ER example to see your understanding of the database concepts introduced in Lecture 2.

Study Material: Lecture 2

Recommended Online Diagram Editor: https://app.diagrams.net/; select Entity Relation shapes as shown in the figure below. Use the notations introduced in Lecture 2



Task 1 Music Streaming Service Database (15 points)

You will design a database for a music streaming service (e.g., Spotify). The database contains information about:

- The user with his/her ID, name, email, password, birthday, and profile image. The user must subscribe with a subscription plan that has an ID, type (free, premium), subscription start date, and subscription due date. The database can not contain users with the same email address because it is used as a username for login purposes. Each user may create one or several playlists.
- The artists with their ID, name, genre, and image. Each artist can be associated with one or more genres. Some artists may have no albums, while others may have one or multiple albums. An album can have one or multiple artists.
- Each album is recorded with its ID, title, release date, and cover image. An album can include one or multiple tracks.
- Each track is recorded with its ID, title, and duration. Some tracks are released individually without being part of an album, some can be included in several albums, and some can belong to unknown (e.g., folk, anonymous artists) or multiple artists.
- Each playlist contains information about its ID, name, and creation date. A playlist can contain zero or many tracks.

1.1 Identify all entities and their attributes from the description of database requirements using the table format presented in the following table template (5 points):
Clarifications: *The Entity column is an entity name (usually in singular form, not plural), the Attribute column is an attribute name that describes the entity, Attribute Type is a type of attribute and can be simple, composite, or multivalued, "Is a key Attribute?" column defines a key of the entity (a key must be unique and static which will never change over time), and the last column contains the data type (string, number, boolean, etc.), and the constraints such as can be null, can not be null, or/and unique.*

Entity	**Attribute**	**Attribute Type**	**Is a Key Attribute?**	**The value type of attributes and constraints (data type, NULL/NOT NULL, Unique)**
Entity 1	Attribute 1	Simple	Yes	Unique, String, not null
	Attribute 2	Composite	No	Number, not null
	Attribute 3	Multivalued	No	String, null
Entity 2	…	….	…	….

1.2 Identifying the relationship between entities defined in Task 2.1 using the following table template (5 points):

Clarifications: Entity A and Entity B columns contain the entity name, Relationship name column is usually a verb that describes how these two entities are connected or associated. The naming of the relationship should be clear, and meaningful to understand the connection/interaction between the entities. The Cardinality Ratio column refers to one-to-one, one-to-many, and many-to-many relationships. The Attribute of Relationship column (which is optional and usually is used in N:M relationships) is an attribute assigned to the relationship, and can't be associated with any entity.

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**
Entity 1	Joined	Entity 2	N:M	Date	Entity 1 can join one or more Entity 2, and one Entity 2 can contain one or more Entity 1.

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. (5 points)

The ER schema should contain entities with their corresponding attributes, key attributes of each entity, only ***binary*** relationships, their corresponding cardinality ratio, and participation constraints (total/partial). You are free to make additional assumptions if you feel that some information is missing.

Task 2 University Social Network Database (15 points)

You will design a database for a university social network website. The database contains information about:

- The students with their ID, name (First name, Last name), email, password, birthday, hobbies, gender, study program (e.g., Computer Science, Design and Humanities, Business and Economics, etc.), date when they joined the network, and profile image. The email is used as a username to login to the network. The network allows registration only for students >=18 years old. Students can upload their own profile image or use the default avatar image.
- The friendship between students, its status (pending, accepted, declined), and the date they become friends.

- The study groups. Each student can create one or more study groups related to a specific subject. Each group contains information about the group ID, name, subject, and created_date. Students can join one or more study groups, and the joined date is recorded.
- Students can create events for meetups, group work, or any social event. An event contains information about its ID, name, description, address (street, postal code), date, and time.
- Students can create a post with a topic to discuss. The post contains ID, text, an image (optional), a location (optional), and a date.
- Students can comment on the posts with a text.

2.1 Identify all entities and their attributes from the description of database requirements using the following Table template: (5 points***)***

Entity	**Attribute**	**Attribute Type**	**Is a Key Attribute?**	**The value type of attributes and constraints (data type, NULL/NOT NULL, Unique)**
Entity 1	ID	Simple	Yes	Unique, Number, not null
	Name (First name, Last name)	Composite	No	String, not null
	Attribute 3	Multivalued	No	String, null
Entity 2	…	….	…	….

2.2 Identifying the relationship between entities defined in Task 1.1 using the following table template (5 points***):***

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**
Entity 1	Joined	Entity 2	N:M	Date	Entity 1 can join one or more Entity 2, and one Entity 2 can contain one or more of Entitiy1

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. (5 points***)***

The ER schema should contain entities with their corresponding attributes, key attributes of each entity, only ***binary*** relationships, their corresponding cardinality ratio, and participation constraints (total/partial). You are free to make additional assumptions if you feel that some information is missing.

Task 3 Airport Management Database (10 points)

3.1 Given the constraints in the ER schema below (Figure 1), respond to the following statements with True, False, or Maybe. Justify your answer (5 points).

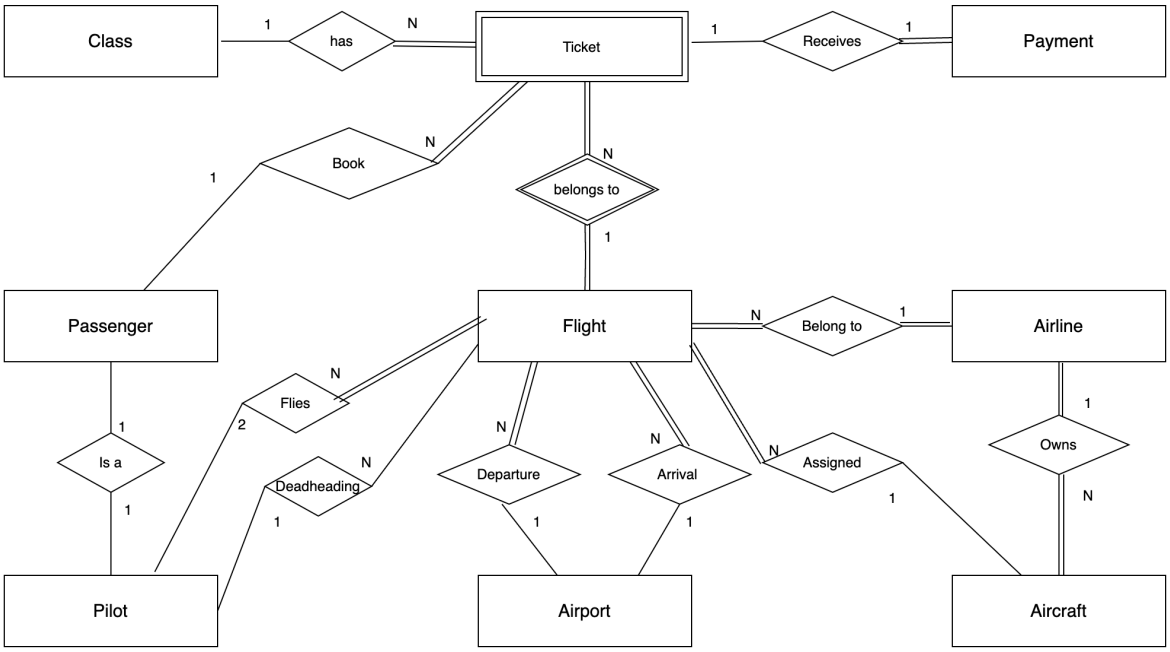


Figure 1 - ER diagram for Airport Management Database

Clarifications:

A Class entity represents the ticket type, such as Economy, Business, Flex class, etc.
A deadheading relationship means when the pilot is on duty flights as a passenger to another airport for work mid-trip or to return to their domicile city after a trip.

1. Every pilot has been a passenger on at least one flight.
2. Every flight has at least one deadheading pilot.
3. Every flight has 2 pilots.
4. Each flight must have an assigned departure and arrival airport.
5. Every pilot has flown at least 2 times.
6. Some tickets that do not belong to any flight.
7. Some airlines do not have flights.
8. A passenger can be a pilot.
9. Some tickets do not belong to any class type (Economy, Business, etc.).
10. Some tickets are without payment.

3.2 List the (min, max) constraint using the total/partial participation for the Ticket entity. Justify your answer (5 points).

Hint: The "min" value represents the type of participation: a single line indicates partial participation with a minimum value of 0, while a double line represents total participation with a minimum value of 1. The "max" value defines the cardinality, specifying whether the relationship is 1:N, N:M. For example, Flight (entity) and Flies(relation) will have min=1 (double line) and max =2 (cardinality ratio). Pilot (entity) and Flies (relation) will have min = 0 (single line), and max = N (cardinality ratio) flights.

Part 2 Relational Data Model and Normalisation (30 points)

Description

In this assignment, you will convert an Entity-Relationship model (Conceptual level) to the Relation model (Logical level) using the learning material provided in Lecture 3. Afterward, you will check to improve the design of the existing relational model to comply with the 3d normal form (3NF).

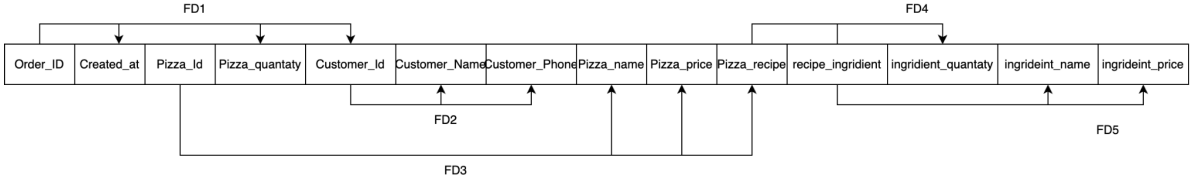
Study Material: Lecture 3 and Lecture 4

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

Using your ER diagram design from Task 1 in Part 1, convert it into a relational model. The process for converting an ER diagram to a relational model is explained in Lecture 3. Use the diagrams.net tool to create the relational model diagram. You are allowed to add any clarifications or explanations related to your proposed Relational Model design.

Task 5 Normalisation (15 points)

5.1 ***Considering unnormalized Pizzeria database with one table is shown below on image bellow, perform normalisaiton to achieve 3NF for all tables (10 points)***.



The following functional dependencies exist:

- FD1: {Order_ID,Pizza_Id} -> {Created_at,Pizza_quantaty, Customer_id}
- FD2: {Customer_ID}->{Customer_Name,Customer_Phone}
- FD3:{Pizza_Id}->{Pizza_name,Pizza_Price,Pizza_Recipe}

- FD4:{Pizza_recipe,recipe_ingridient} -> {ingridient_quantaty}
- FD5:{recipe_ingridient}->{ingridient_name,ingridient_price}

Each table must have a unique key, indicated by an underline, as well as foreign keys, which should be marked along with their references to the corresponding primary keys.

How can you check if the normalization is correct?
Join all tables using keys and foreign keys to get the original table with the exact amount of columns.

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 (5 points).

Hints: Pizza_price can not be uniquely identified by Pizza_Id

Part 3 Basic and Advanced SQL (30 points)

Description
In this assignment, you will create SQL queries to retrieve the data from an existing database. You will use an online SQL editor to load the existing database and execute SQL queries.

Software: Online SQL editor (https://sqliteonline.com/)

Study materials: Lecture 5

The use of AI support: It is recommended that you first write the query by yourself using the learning material provided in this course. Use the AI tool:

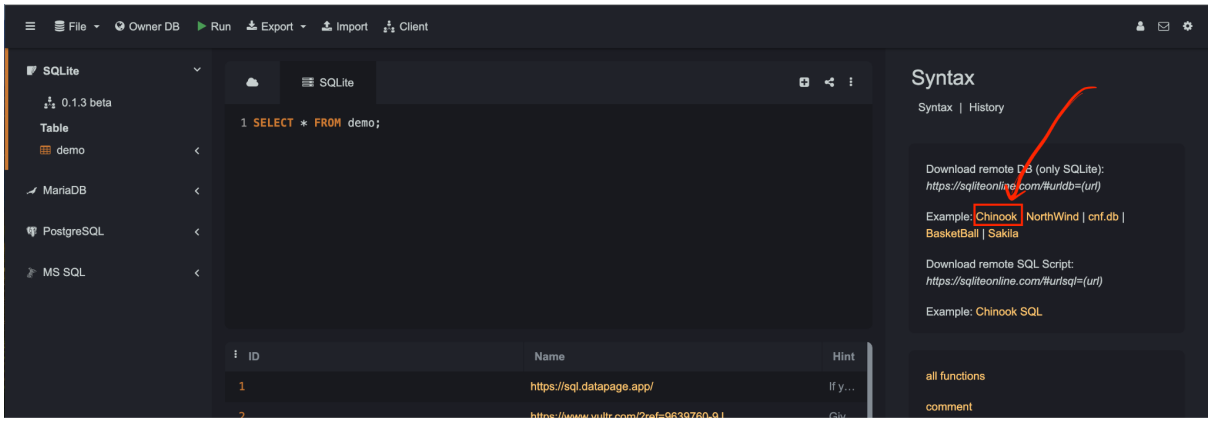
- to get better clarification of the SQL commands or the query
- To get explanation
- To check your solution
- To see if any other alternative solution exists

It's important to note that in the written exam, you will encounter similar SQL queries, and you will need to write them by hand on paper without the assistance of AI.

How do I check if my query is correct?
The only way to check it is to do it manually and verify the expected results.

Preparation

Navigate to the https://sqliteonline.com/ web site, and on the Syntax panel, click on Chinook as shown in the image bellow.



The Chinook database represents a digital media store, including tables for artists, albums, media tracks, invoices, and customers. The relational data model and detailed description can be found here https://github.com/lerocha/chinook-database .

Task 6 Basic SQL queries (20 points)

Write a query that:
1.1 retrieves all artists whose names start with the same letter as the first letter of your first name.
1.2 retrieves all tracks by “BackBeat” artists.
1.3 lists all album titles along with their respective artist name.
1.4 Lists the total number of tracks in each album, sorted by total in descending order.
1.5 finds tracks with “Protected AAC audio file” media type
1.6 Lists all tracks from “Big Ones” album
1.7 finds the total duration of tracks in each playlist. You'll need to use the *Playlist*, *PlaylistTrack*, and *Track* tables for this.
1.8 finds the 10 most expensive tracks.
1.9 finds the list of artists who do not have any albums.
1.10 lists all playlists and the number of tracks in each.

Task 7 Advanced SQL queries (10 points)

Write a query that

2.1 finds albums with multiple genres.
Clarification: Find all albums with tracks from multiple genres. It will be necessary to group tracks by album and genre and filter for albums that appear in multiple genres.

2.2 identifies customers with the highest average invoice value.

Clarification: Write a query to determine which customers have the highest average invoice value. Join the appropriate tables, calculate the average for each customer, and then rank or filter based on the highest values.

2.3 Find the Longest Playlist by Duration.

Write a query to identify which playlist has the longest total duration of all its tracks. In order to find which playlist has the maximum total duration, you will need to join the Playlist, PlaylistTrack, and Track tables.

2.4 finds customers with Maximum Purchase in Each Country.

Clarification: *Create a query that finds the customer with the highest total purchases in each country. Consider how to rank or group customers based on their country using the Customer, Invoice, and InvoiceLine tables.*

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

Clarification: The most popular album is the one with the highest number of purchases.

Submission

Submit a **PDF report** that includes solutions to **all tasks presented in Part 1, Part 2, Part 3, ~~and Part 4~~** using the assignment template provided in Moodle. The ER diagrams should NOT be drawn by hand. If the diagrams are too big to fit into a PDF file, please submit the diagrams as images. In addition, submit the script .**sql** file containing SQL queries from ***Part 3 ~~Part 4~~***.