
Data Modeling - KU

Exercise 1

Lucas Iacono - 2025

Exercise 1 (25 points)

- Published Mon 10th of March 2025 (TeachCenter)
 - Deadline Tue 1st of April 2025 (23:59)
-

Exercise 1 (25 points)

- StackExchange
 - Q&A platform to collect questions, answers, ratings, etc.
 - E.g., **StackOverflow** for programming-related questions
 - Datasets are shared for research and teaching purposes

 - **Cleaned dataset will be shared for Exercise 2**
-

Exercise 1 (25 points)

- 1-1: ER Modeling (15 points)
 - Deliverable: ERDiagram.pdf (1 page PDF)

 - 1-2: Mapping ER Model to Relational Model (10 points)
 - Deliverable: CreateSchema.sql
-

1-1 ER Modeling

- Create an ER diagram in Modified Chen notation for managing data stored on a Q&A (Question and Answer) platform.
 - Entity types
 - Relationship types
 - Attributes
 - Cardinalities (M, MC, C, 1)
 - Keys (create surrogate keys if natural unique identifiers are missing)
 - Use **draw.io** for **designing** the **ER diagram**.
 - Why draw.io -> toolbar for ER elements (and other benefits!!).
-

1-1 ER Modeling

The diagram should capture the essential information of the **following discourse**:

- The database contains data on users registered on the platform. Each user is described by a unique ID, name displayed on the Web site and country. Additionally, the platform stores the registration date of every user and a link to their profile image (if they decide to submit one). The platform also saves the number of upvotes and the number of downvotes each user receives. On this basis, a user's reputation score is calculated.
 - Users can collect badges on the platform. For each badge there are an ID, title and date of being granted to a particular user stored in the database. The number of badges per user is unlimited.
 - Users can publish posts on the platform if they wish. The main elements of a post are its title, body and tags. The database also includes data about the creation date, view count and score of each post. Only a single user can author a post.
-

1-1 ER Modeling

- A post on the platform can be linked to another post. There is no upper or lower limit to a number of posts linked to an original post.
 - Users are allowed to comment on posts on the platform. The database stores the text of a comment, its creation date and its score. A comment can correspond to a single post only.
 - Users may also share their opinions on a post in the form of votes. The platform does not limit the number of users interacting with a post in this way. However, there can be a maximum of one vote given to a post per user. A vote is described by its ID and the date of submission.
-

1-2 Mapping ER Diagrams into the Relational Model

- Create a relational schema for the ER diagram designed in Task 1.1 and bring it into **third normal form**.
 - This schema should include the relations and typed attributes, as well as all primary and foreign keys.
 - Then, use the PostgreSQL Data Base Management System to create a SQL script (CreateSchema.sql) with CREATE TABLE statements.
-

Example

Disclaimer: This section contains an example. The exercise to be solved to approve the course can be found in the TeachCenter.

Task 1.1

Task 1.2

PostgreSQL & pgAdmin

Example T1.1 (similar to our exercise)

Create an ER diagram in Modified Chen notation for managing data stored on the Formula 1 Championship Management System

The database should capture the essential information of the following statements:

- The database contains data on **drivers** participating in the F1 Championship. Each driver has a unique ID (e.g., 44 for Lewis Hamilton), name, nationality, and date of birth.
- **Teams** participate in the championship, and each team has exactly two drivers. A team is described by its name, country, and foundation year.
- Each **Grand Prix** is an event held in a specific country and has attributes such as circuit name, date, and number of laps. Each Grand Prix can have multiple drivers competing, and each driver may participate in multiple Grand Prix events.



Example T1.1 (similar to our exercise)

Create an ER diagram in Modified Chen notation for managing data stored on the Formula 1 Championship Management System

Step 1: Identify entities and attributes

Entity 1: Driver

- ID (Primary Key)
- Name
- Nationality
- Date of Birth

Entity 2: Teams

- Team Name (Primary Key)
- Country
- Foundation Year

Entity 3: Grand Prix

- Grand Prix ID (Primary Key)
- Circuit Name
- Country
- Date
- Number of Laps

Example T1.1 (similar to our exercise)

Create an ER diagram in Modified Chen notation for managing data stored on the Formula 1 Championship Management System

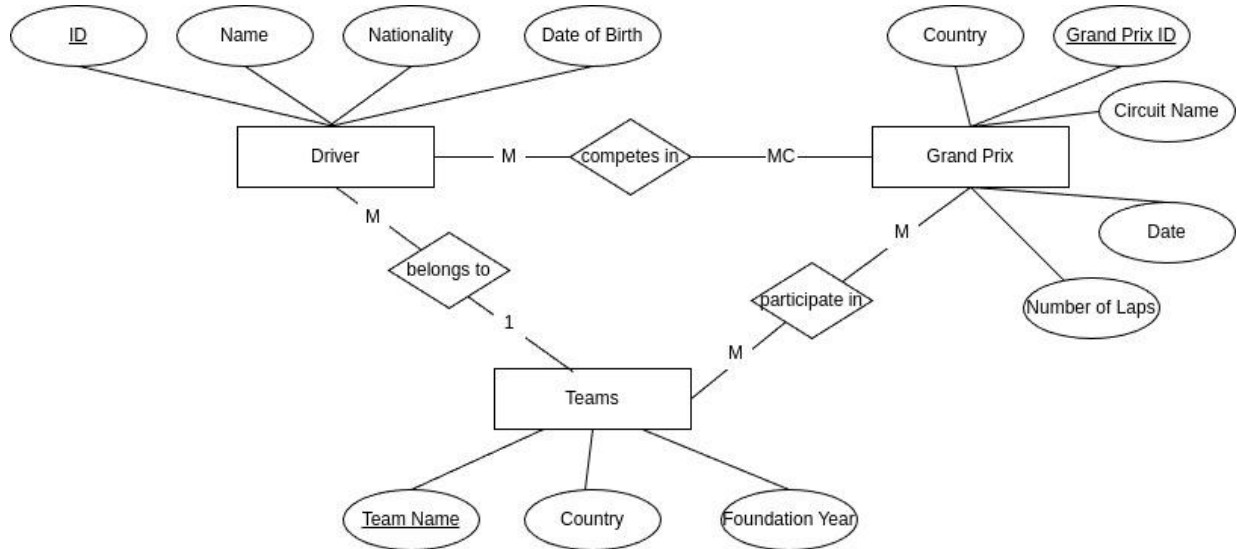
Step 2: Relationships and cardinalities

Relationship	Cardinality	Explanation
Driver - Belongs To - Team	(1:M)	Each driver belongs to one team, and each team has exactly two drivers.
Driver - Competes In - Grand Prix	(MC:M)	A Grand Prix has multiple drivers competing, and a driver may participate in multiple Grand Prix races (or none).
Team - Participates In - Grand Prix	(M:M)	Each team participates in multiple Grand Prix races, and each Grand Prix has multiple teams competing.

Example T1.1 (similar to our exercise)

Create an ER diagram in Modified Chen notation for managing data stored on the Formula 1 Championship Management System

Step 3: ER Map



Example T1.2 (similar to our exercise)

Create a relational schema for the ER diagram designed in Task 1.1 and bring it into third normal form. This schema should include the relations and typed attributes, as well as all primary and foreign keys. Then, use the PostgreSQL Data Base Management System to create a SQL script (CreateSchema.sql) with CREATE TABLE statements.

Step 1:

- Download installer via: <https://www.postgresql.org/download/>
- Setup PostgreSQL in your Laptop (See Tutorial in TeachCenter)

Example T1.2 (similar to our exercise)

Step 2: create your schema and normalize (1NF, 2NF, 3NF)

```
4 -- Table: Teams
5 CREATE TABLE Teams (
6     team_id SERIAL PRIMARY KEY,
7     team_name VARCHAR(100) NOT NULL,
8     country VARCHAR(50) NOT NULL,
9     foundation_year INT NOT NULL
10 );
11
12 -- Table: Drivers
13 CREATE TABLE Drivers (
14     driver_id SERIAL PRIMARY KEY,
15     name VARCHAR(100) NOT NULL,
16     nationality VARCHAR(50) NOT NULL,
17     date_of_birth DATE NOT NULL,
18     team_id INT NOT NULL -- Reference to Teams (1:M relationship)
19 );
20
21 -- Table: Grand Prix
22 CREATE TABLE GrandPrix (
23     grand_prix_id SERIAL PRIMARY KEY,
24     circuit_name VARCHAR(100) NOT NULL,
25     country VARCHAR(50) NOT NULL,
26     date DATE NOT NULL,
27     number_of_laps INT NOT NULL
28 );
```

Example T1.2 (similar to our exercise)

Step 3: Implement your schema in PostgreSQL



Example T1.2 (similar to our exercise): Create DB

The screenshot shows the pgAdmin 4 interface. The 'Object Explorer' on the left shows a tree view with 'Servers (1)' expanded to 'DM2025', which contains 'Databases (3)'. The 'Databases (3)' folder is selected, and a context menu is open with the 'Create' option highlighted. A sub-menu is visible with the option 'Database...'. The main window shows the 'Activity' tab with a 'Server sessions' table and two line graphs for 'Tuples in' and 'Tuples out'.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Servers (1)

DM2025

Databases (3)

- data_manager
- formula_1
- postgres
- Login/Group Roles
- Tablespaces

Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x

Activity State Configuration Logs System

Server sessions

Total Active

Tuples in

Inserts Updates Deletes

Tuples out

Server sessions	Total	Active
1		
2		
3		
4		
5		

Tuples in	Tuples out
1	1.5K
0.75	1K
0.5	500
0.25	0
0	0

Example T1.2 (similar to our exercise): Create DB

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows a server named 'DM2025' with three databases: 'data_management', 'formula_1', and 'postgres'. The 'Server sessions' activity monitor is open, showing a graph of 'Transactions per second' with a legend for Total (blue), Active (orange), and Idle (green) sessions. The 'Create - Database' dialog box is the central focus, with the following fields:

- Database:** formula-1-dmsl
- Owner:** postgres
- Comment:** (empty text area)

The dialog box has tabs for 'General', 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

Example T1.2 (similar to our exercise): Create Schema

The screenshot displays the PostgreSQL Enterprise Manager interface. On the left, the Object Explorer shows a tree view of the database structure. The 'formula-1-dms' database is selected, and its 'Schemas' folder is expanded. A context menu is open over the 'Schemas' folder, with 'CREATE Script' highlighted. The main panel shows the 'Activity' tab with several performance charts: 'Database sessions', 'Transactions per second', 'Tuples in', 'Tuples out', and 'Block I/O'. The 'Database sessions' chart shows a single active session. The 'Transactions per second' chart shows a peak of 8 transactions. The 'Tuples in' chart shows a peak of 100 tuples. The 'Tuples out' chart shows a peak of 600 tuples. The 'Block I/O' chart shows a peak of 800 blocks.

Object Explorer

- Servers (1)
 - DM2025
 - Databases (4)
 - data_management
 - formula-1-dms
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - formula_1
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x

Activity State Configuration Logs System

Database sessions

Total Active Idle

Transactions per second

Tuples in

Inserts Updates Deletes

Tuples out

Fetches Returned

Block I/O

Create

- Delete
- Delete (Force)
- Refresh...
- Restore...
- Backup...
- CREATE Script
- Disconnect from database
- ERD For Database
- Maintenance...
- Grant Wizard...
- Search Objects...
- PSQL Tool
- Query Tool
- Properties...

Example T1.2 (similar to our exercise): Create Schema

The screenshot displays a database management tool interface. On the left, the Object Explorer shows a tree view of servers and databases. The 'formula-1-dms' database is selected. The main window shows a SQL query editor with the following code:

```
1  -- Drop tables if they already exist
2  DROP TABLE IF EXISTS Drivers, Teams, GrandPrix, TeamGrandPrix, DriverGrandPrix CASCADE;
3
4  -- Table: Teams
5  CREATE TABLE Teams (
6      team_id SERIAL PRIMARY KEY,
7      team_name VARCHAR(100) NOT NULL,
8      country VARCHAR(50) NOT NULL,
9      foundation_year INT NOT NULL
10 );
11
12 -- Table: Drivers
13 CREATE TABLE Drivers (
14     driver_id SERIAL PRIMARY KEY,
15     name VARCHAR(100) NOT NULL,
16     nationality VARCHAR(50) NOT NULL,
17     date_of_birth DATE NOT NULL,
18     team_id INT NOT NULL -- Reference to Teams (1:M relationship)
19 );
20
21 -- Table: Grand Prix
22 CREATE TABLE GrandPrix (
```

The interface includes a top toolbar with various icons for file operations and a bottom panel with tabs for 'Data Output', 'Messages', and 'Notifications'.

Vielen Dank!
