

National University of Computer and Emerging Sciences FAST



Database Systems Lab Manual

Course Instructor	Ms. Aleena Ahmad
Lab Instructor	Muhammad Kamran
Lab TA	Abdullah Naeem
Section	BCS-4F
Semester	Spring 2025

Department of Computer Science

FAST-NU, Lahore, Pakistan

Lab 10 – DB Modelling

Objectives

- Learn how to design a relational database by mapping relationships between entities, including **one-to-one**, **one-to-many**, and **many-to-many** relationships.
 - Gain hands-on experience using database design tools like **ERWIN** to visually represent and create Relational Schema.
 - Understand how to structure complex data models for real-world applications.
-



Instructions

- Use ERWIN to design the EER and ER Diagrams for the Need for Speed game scenario, ensuring that the relationships between entities (like cars, characters, and game editions) are accurately mapped.
 - Refer to the attached ERWIN manual for guidance on how to use the tool for creating and managing Relational Schema.
 - Focus on identifying and mapping one-to-one, one-to-many, and many-to-many relationships within the context of the game's data, such as cars and game editions or characters and roles.
-



Database Design for "Need for Speed" Game Series

In this lab scenario, we are tasked with designing a database for the **Need for Speed** (NFS) game series, which will store detailed information about various cars, characters, game editions, and missions. The goal is to capture the relationships between these entities, especially focusing on the fact that both cars and characters can appear in multiple editions of the game, with varying roles and attributes in each edition.

The **Cars** table will include essential details like the model, make, year of introduction, top speed, and acceleration. Each car will be associated with a specific **Car Company**, which provides information such as the company's name and the year it was founded. This structure allows us to group cars by their manufacturers and easily track which cars belong to which companies.

Next, we have the **Game Editions** table, which stores information about the various versions of the NFS games released over the years. This table includes attributes like the edition's name, release date, and platform availability. The **Characters** table will store details about the characters featured in the game, including the character's name and role in the game. For instance, a character like **Sam Harper** from **Need for Speed: Carbon**, who plays the role of the player's ally and racing companion, might appear in different editions, but his role and importance could vary depending on the edition.

Characters can appear in multiple game editions, but their roles might differ in each edition. For example, **Sam Harper** could be a key ally in **Need for Speed: Carbon**, but in another edition like **Need for Speed: Most Wanted**, he might only appear briefly or in a different capacity. This flexibility is important for the database design, as it allows for a dynamic assignment of roles to characters across different editions.

The **Missions** table will store details about the missions associated with each game edition, including the mission's name and description. Missions are specific to each game edition and will be linked to the corresponding edition in the database. This allows players to track the objectives and challenges available in different editions of the game.

In addition to the core data tables, the database must handle the complex relationships between these entities. For example, cars may appear in multiple editions, and each edition can feature several cars. Similarly, characters can be part of multiple editions, each time with potentially different roles and significance. This structure gives us the flexibility to handle scenarios where the same car or character appears in multiple game editions, but with distinct attributes, roles, or availability depending on the edition.

-----Good Luck-----



-----END-----