

Project Assignment 2

-Database and SQL

Aleksandra Dorszewicz

* Creating database…………………………….
* Writing SQL query and execution……..
* SQL query debugging………………………..
* Creating EER Diagram……………………….
* Understanding relationships…………….
* Identifying primary and foreign key….

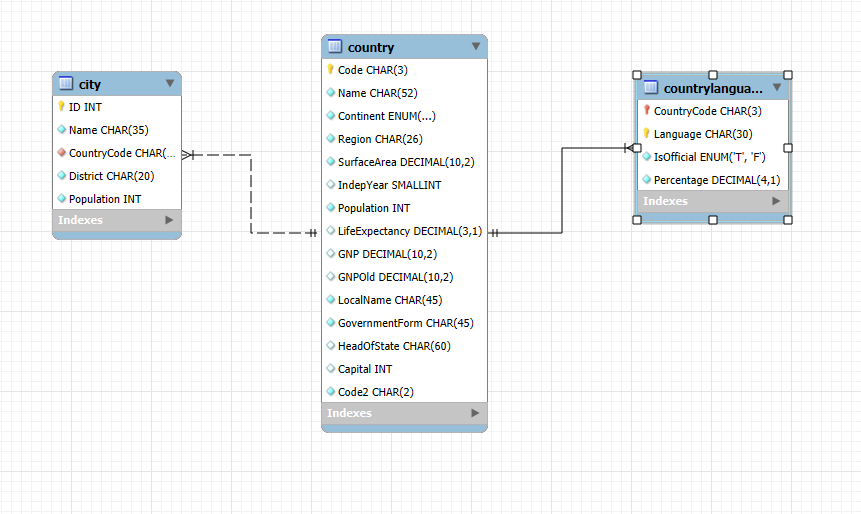
**Contents**

Task 1: List the different types of relationships in relational databases and provide examples………………….

Task 2: What is Normalization and why is it important to database development?....................................................

Task 26: Create a MS Word file name it as - Name for your answers………………………………………………………………….

* Identify the primary key in country table.
* Identify the primary key in city table.
* Identify the primary key in countrylanguage table.
* Identify the foreign key in city table.
* Identify the foreign key in countrylanguage table



Task 1: **List the different types of relationships in relational databases and provide examples**

Relational databases are powerful tools for organizing and managing information. One of the key concepts in relational database design is the establishment of relationships between different entities. These relationships can be categorized into three primary types: **one-to-one, one-to-many, and many-to-many.**

A **one-to-one** relationship exists when a single record in one table corresponds to exactly one record in another table. For instance, a person can only have one social security number, and conversely, a social security number can only be assigned to one person. This type of relationship is straightforward and often used to link closely related entities.

A **one-to-many** relationship, on the other hand, involves a single record in one table that can be associated with multiple records in another table. However, each record in the second table can only be linked to one record in the first table. A classic example of this is the relationship between a department and its employees. A department can have many employees, but each employee belongs to only one department.

Finally, **a many-to-many** relationship occurs when multiple records in one table can be associated with multiple records in another table. A common example is the relationship between students and courses. A student can enroll in many courses, and a course can be taken by many students. This type of relationship requires a junction table to effectively manage the connections between the two entities.

Understanding these different types of relationships is crucial for designing efficient and well-structured relational databases. By carefully considering the nature of the data and the desired relationships, database designers can create databases that are both flexible and easy to maintain.

Task 2: **What is Normalization and why is it important to database development?**

Normalization is a systematic process of organizing data in a database to reduce redundancy, improve data integrity, and enhance efficiency. By breaking down data into smaller, well-defined tables with specific relationships, normalization eliminates duplicate data, prevents anomalies, and optimizes storage and retrieval. This approach leads to more scalable, flexible, and maintainable databases, as it allows for easier data modification and addition without compromising the overall structure. Normalization levels, such as First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF), address specific data anomalies to achieve a well-structured database design.

**Data Redundancy**: Redundancy occurs when the same data is stored in multiple places. Normalization eliminates this by storing data only once, reducing storage space and the risk of inconsistencies

**Data Integrity**: Normalization ensures that data is accurate and consistent by enforcing data constraints and preventing anomalies like update, insertion, and deletion anomalies.

**Data Efficiency**: Normalized databases are more efficient in terms of storage and retrieval. Queries can be executed faster and more accurately, leading to improved performance.

**Scalability:** Normalized databases are easier to scale as the data grows. Adding new data or modifying existing data is simpler and less prone to errors.

**Flexibility:** Normalized databases are more flexible and adaptable to changes in data requirements. New fields or relationships can be added without significantly impacting the overall database structure.

Task 26: Create a MS Word file name it as - Name for your answers

the **primary** key in country table is **Code**

the **primary** key in city table is **ID int**

the **primary** key in CountryLanguage table is **Language CHAR (30)**

the **foreign** key in city table is **Country code**

the **foreign** key in CountryLanguage table is **Country Code**