Business Template

SUBJECT AREAS

Contents

1 Business Description

1.1 Business background

The business revolves around organizing and documenting mountain climbing expeditions. It keeps track of various mountains, the actions taken during expeditions, groups of climbers

1.2 Problems. Current Situation

Currently, there is a lack of a centralized system to efficiently manage and retrieve detailed information about different aspects of the climbing expeditions.

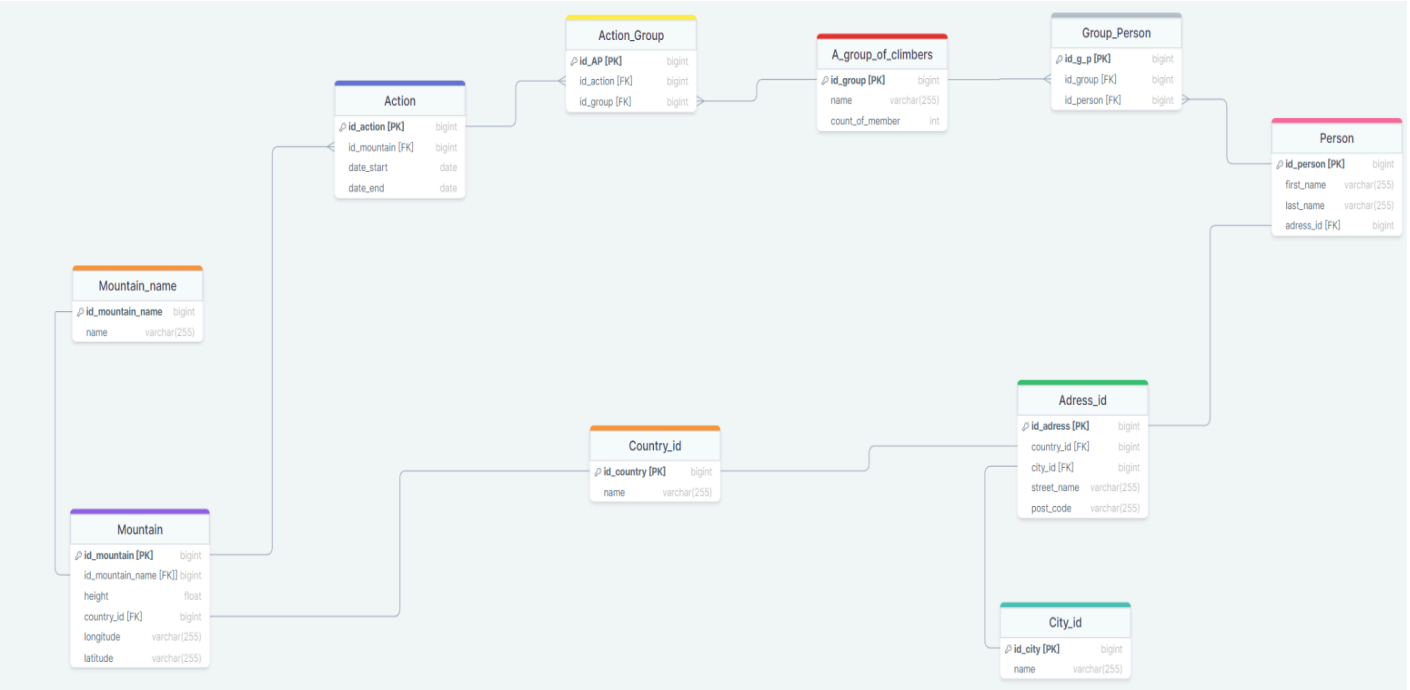
1.3 The Benefits of implementing a database. Project Vision

2 Model description

2.1 Definitions & Acronyms

Implementing a database will streamline the process of storing, managing, and retrieving detailed information about each expedition including participants' details and their actions.

2.2 Logical Scheme



2.3 Objects

Table Description

This table, therefore, holds information about people, including their first and last names, with a unique identifier for each person.

Table Name	Field name	Field Description	Data Type
Person	id_person	PK	bigint
	first_name		varchar(255)
	last_name		varchar(255)
	adress_id	FK	bigint

It references another table through the adress_id field, to link each person to their address.

Example with data

id_person	first_name	last_name	adress_id
1	Dzianis	Lukashevich	666

Table Description

This table captures details about addresses, including the street name, postal code, and references to associated country and city information.

Table Name	Field name	Field Description	Data Type
Adress_id	id_adress	PK	bigint
	country_id	FK	bigint
	city_id	FK	bigint
	street_name		varchar(255)
	post_code		varchar(255)

Country_id and city_id fields are foreign keys, referencing the primary keys of tables storing country and city information.

Example with data

id_adress	country_id	city_id	street_name	post_code
1	12	23	Prityckogo	220073

Table Description

This table is designed to avoid issues with variations in city name spellings. By assigning unique identifiers to each city (id_city) and storing their respective names in the "name" field, the system ensures standardization and uniformity in the data, preventing potential errors due to differences in spelling or formatting of city names.

Table Name	Field name	Field Description	Data Type
City_id	id_city	PK	bigint
	name		varchar(255)

Example with data

id_city	name
1	Warsaw

Table Description

This table is designed to maintain consistency in the representation of country names, ensuring that variations or discrepancies in spellings are avoided. By assigning a unique identifier to each country and storing its name, the table facilitates standardized data management and retrieval processes.

Table Name	Field name	Field Description	Data Type
Country_id	id_country	PK	bigint
	name		varchar(255)

Example with data

id_country	name
1	Poland

Table Description

This table captures details about mountains, including their height, location coordinates (latitude and longitude).

Table Name	Field name	Field Description	Data Type

Mountain	id_mountain	PK	bigint
	id_mountain_name	FK	bigint
	height		float
	country_id	FK	bigint
	longitude		varchar(255)
	latitude		varchar(255)

References to associated mountain names and countries.

Example with data

id_mountain	id_mountain_name	height	country_id	longitude	longitude
1	1	8,848	1	27.988056°	86.925278°

Table Description

This table is designed to store names of mountains, ensuring consistency and standardized representation of mountain names within the system. Each mountain name is uniquely identified by a numeric ID (id_mountain_name) and stored as a string in the "name" field.

Table Name	Field name	Field Description	Data Type
Mountain_name	id_mountain_name	PK	bigint
	name		varchar(255)

Example with data

id_mountain_name	name
1	Everest

Table Description

This table captures details about climbing activities related to mountains, including the mountain involved, start and end dates of the climb. The id_action field serves as a unique identifier for each climbing record, while the id_mountain field establishes a relationship between climbs and the respective mountains they are associated with.

Table Name	Field name	Field Description	Data Type
Action	id_action	PK	bigint
	id_mountain	FK	bigint
	date_start		date
	date_end		date

Each record in the "Action" table, representing a particular climb or activity, is associated with a specific mountain by referencing its unique identifier (id_mountain).

Example with data

id_action	id_mountain	date_start	date_end
1	1	17.07.2024 15:00	19.07.2024 11:00

Table Description

This table, named "Action_Group," acts as a linking or junction table to establish a many-to-many relationship between two other tables, likely "Action" and "Group."

Table Name	Field name	Field Description	Data Type
Action_Group	id_AG	PK	bigint
	id_action	FK	bigint
	id_group	FK	bigint

This table facilitates the implementation of a many-to-many relationship between actions and groups. Each record in the "Action_Group" table represents a combination of an action and a group. It allows for actions to be associated with multiple groups and vice versa, by storing pairs of action IDs and group IDs.

Example with data

id_AG	id_action	id_group
1	1	1

Table Description

This table represents groups of climbers, with each group uniquely identified by an ID (id_group). The group's name is stored as a string in the "name" field, while the number of members in the group is stored as an integer in the "count_of_members" field.

Table Name	Field name	Field Description	Data Type
A_group_of_climbers	id_group	PK	bigint
	name		varchar(255)
	count_of_members		int

Example with data

id_group	name	count_of_members
1	Stars	8

Table Description

This table, named "Group_Person," serves as a linking or junction table to establish a many-to-many relationship between two other tables, likely "A_group_of_climbers" and "Person."

Table Name	Field name	Field Description	Data Type
Group_Person	id_g_p	PK	bigint
	id_group	FK	bigint
	id_person	FK	bigint

This table facilitates the implementation of a many-to-many relationship between groups of climbers and persons. Each record in the "Group_Person" table represents a connection between a group and a person. It allows for persons to belong to multiple groups and for groups to include multiple persons by storing pairs of group IDs and person IDs.

Example with data

id_g_p	id_group	id_person
1	1	1