

**Business description of**

**Database of Climbs**

**for a Mountaineering Club**



**CONTENTS**

1. **BUSINESS DESCRIPTION**  
   1.1. Business Background………………………………………………………………3  
   1.2. Problems. Current Situation………………………………………………………..3  
   1.3. The Benefits of Implementing a Database. Project Vision………………………...3
2. **MODEL DESCRIPTION**  
   2.1. Definitions & Acronyms……………………………………………………………4  
   2.2. Logical Scheme……………………………………………………………………..4  
   2.3. Objects………………………………………………………………………………5  
   2.4. Comments on Table Relationships………………………………………………...10

**1. BUSINESS DESCRIPTION**

**1.1. Business Background**

The climbing industry has grown significantly over the years, with many climbers venturing to various mountains around the world. In order to track the progress of mountaineers, record significant climbs, and ensure safety and success in these efforts, a robust system is needed to store detailed data. This information includes the start and end dates of climbs, the climbers’ details, mountain information (such as the mountain name, height, and location), and more.

**1.2. Problems. Current Situation**

At present, data about climbs may be stored in various formats: physical records, spreadsheets, or in some cases, informal databases. However, these systems are often prone to errors, lack accessibility for cross-referencing, and don't allow for easy updates or data mining. Key relationships, such as multiple climbers scaling the same mountain or a climber attempting multiple climbs, are difficult to track.

**1.3. The Benefits of Implementing a Database. Project Vision**

Implementing a centralized database will help address these issues by providing a structured and normalized approach to store and access information. This will allow users to easily track climbers, mountain details, climbing records, and the relationship between these entities, ultimately improving data accuracy and accessibility. The database will be designed to ensure scalability, ease of use, and ensure 3rd normal form (3NF) compliance.

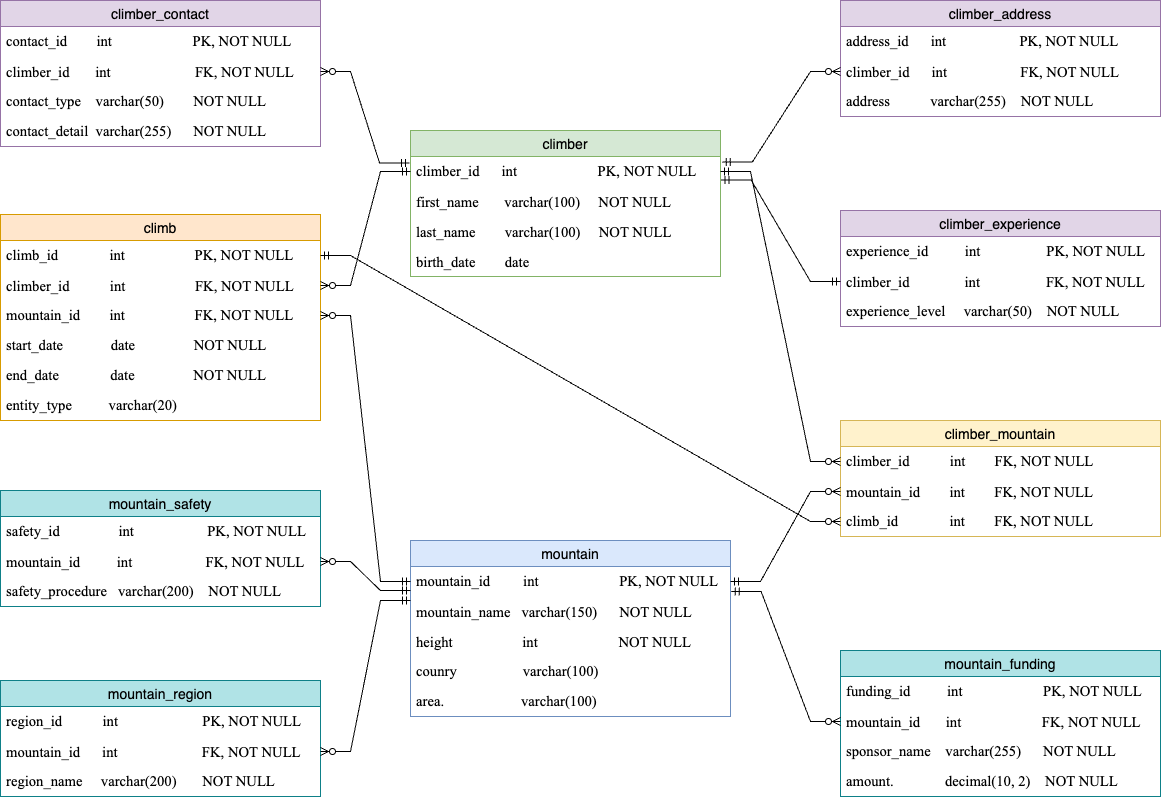
**2. MODEL DESCRIPTION**

**2.1. Definitions & Acronyms**

* **PK (Primary Key):** A unique identifier for each record in a table.
* **FK (Foreign Key):** A key that links one table to another, establishing a relationship between them.
* **3NF (Third Normal Form):** A normalization process to eliminate transitive dependency and ensure data integrity.
* **ER Diagram:** Entity-relationship Diagram is a visual representation of data within a database.
* **NOT NULL:** Ensures that a field must have a value; it cannot be left empty.
* **AUTO\_INCREMENT:** Automatically generates a unique value for the field (commonly used for primary keys).
* **UNIQUE:** Ensures that each value in the column is unique across the table.

**2.2. Logical Scheme**

**Logical Schema (ER Diagram)**



**2.3. Objects**

**Table 1: climber**

#### **Table Description:**

This table stores information about individual climbers, such as their names, contact details, and address. Each climber is uniquely identified by climber\_id, which serves as the primary key for the table.

#### **Table: climber**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Constraints** |
| climber\_id | Unique identifier for the climber (PK) | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| first\_name | First name of the climber | VARCHAR(100) | NOT NULL |
| last\_name | Last name of the climber | VARCHAR(100) | NOT NULL |
| birth\_date | Date of birth of the climber | DATE |  |

#### **Example with Data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **climber\_id** | **first\_name** | **last\_name** | **birth\_date** |
| 1223 | Jake | Simpson | 1982-10-23 |
| 1224 | Maria | Johnson | 1987-07-15 |

### ****Table 2: mountain****

#### **Table Description:**

This table stores information about each mountain, including its name, height, country, and area. Each mountain is uniquely identified by *mountain\_id*, which is the primary key for the table.

#### **Table: mountain**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Constraints** |
| mountain\_id | Unique identifier for the mountain (PK) | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| mountain\_name | Name of the mountain | VARCHAR(150) | NOT NULL |
| height | Height of the mountain (in meters) | INT | NOT NULL |
| country | Country where the mountain is located | VARCHAR(100) |  |
| area | Area or region within the country | VARCHAR(100) |  |

#### **Example with Data:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **mountain\_id** | **mountain\_name** | **height** | **country** | **area** |
| 1001 | Everest | 8848 | Nepal | Himalayas |
| 1002 | K2 | 8611 | Kazakhstan | Tien-Shan |

### ****Table 3: climb****

#### **Table Description:**

This table records each climbing event, including the start and end dates, as well as the climber and mountain involved.

#### **Table: climb**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Constraints** |
| climb\_id | Unique identifier for the climb | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| climber\_id | Links to the climber | INT | Foreign Key (FK) referencing climber(climber\_id), NOT NULL |
| mountain\_id | Links to the mountain | INT | Foreign Key (FK) referencing mountain(mountain\_id), NOT NULL |
| start\_date | Start date of the climb | DATE | NOT NULL |
| end\_date | End date of the climb | DATE | NOT NULL |
| entity\_type | Single or in group | VARCHAR(20) |  |

#### **Example with Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **climb\_id** | **climber\_id** | **mountain\_id** | **start\_date** | **end\_date** | **entity\_type** |
| 501 | 1223 | 1001 | 2023-05-10 | 2023-05-15 | group |
| 502 | 1224 | 1001 | 2023-05-10 | 2023-05-15 | group |
| 503 | 1224 | 1002 | 2023-08-01 | 2023-08-10 | single |

### ****Table 4: climber\_mountain****

#### **Table Description:**

This table represents the many-to-many relationship between climbers and mountains. It tracks which climbers have attempted which mountains and when.

#### **Table: climber\_mountain**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Constraints** |
| climber\_id | Links to the climber | INT | Foreign Key (FK) referencing climber(climber\_id), NOT NULL |
| mountain\_id | Links to the mountain | INT | Foreign Key (FK) referencing mountain(mountain\_id), NOT NULL |
| climb\_id | Links to the climb | INT | Foreign Key (FK) referencing climb(climb\_id), NOT NULL |

#### **Example with Data:**

|  |  |  |
| --- | --- | --- |
| **climber\_id** | **mountain\_id** | **climb\_id** |
| 1223 | 1001 | 501 |
| 1224 | 1001 | 502 |

### ****Table 5: climber\_address****

#### **Table Description:**

This table stores the address information for climbers. A climber can have multiple addresses.

#### **Table: climber\_address**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Data Type** | **Constraints** |
| address\_id | Unique identifier for the address (PK) | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| climber\_id | Foreign key linking to the climber | INT | Foreign Key (FK) referencing climber(climber\_id), NOT NULL |
| address | The address of the climber | VARCHAR(255) | NOT NULL |

#### **Example with Data:**

|  |  |  |
| --- | --- | --- |
| **address\_id** | **climber\_id** | **address** |
| 101 | 1223 | NY, 51 Street |
| 102 | 1224 | LA, 14 Avenue |

### ****Table 6: climber\_contact****

#### **Table Description:**

This table stores the contact details for climbers, including email addresses, phone numbers, etc. A climber can have multiple contact methods.

#### **Table: climber\_contact**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Constraints** |
| contact\_id | Unique identifier for the contact (PK) | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| climber\_id | Foreign key linking to the climber | INT | Foreign Key (FK) referencing climber(climber\_id), NOT NULL |
| contact\_type | Type of contact (e.g., email, phone) | VARCHAR(50) | NOT NULL |
| contact\_detail | Contact detail (e.g., phone number) | VARCHAR(255) | NOT NULL |

#### **Example with Data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **contact\_id** | **climber\_id** | **contact\_type** | **contact\_detail** |
| 201 | 1223 | email | jakesimpson@gmail.com |
| 202 | 1224 | phone | +1234567890 |

**Table 7: mountain\_region**

***Table Description:***

This table stores information about specific regions or areas within a country where mountains are located.

***Table: mountain\_region***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Constraints** |
| region\_id | Unique identifier for the region (PK) | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| mountain\_id | Foreign key linking to the mountain | INT | Foreign Key (FK) referencing mountain(mountain\_id), NOT NULL |
| region\_name | Name of the region | VARCHAR(200) | NOT NULL |

***Example with Data:***

|  |  |  |
| --- | --- | --- |
| **region\_id** | **mountain\_id** | **region\_name** |
| 301 | 1001 | Eastern Himalayas |
| 302 | 1002 | Northern Karakoram |

**Table 8: mountain\_safety**

***Table Description:***

This table tracks safety protocols and procedures specific to each mountain.

***Table: mountain\_safety***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Constraints** |
| safety\_id | Unique identifier for the safety procedure | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| mountain\_id | Links to the mountain | INT | Foreign Key (FK) referencing mountain(mountain\_id), NOT NULL |
| safety\_procedure | Safety protocol or guideline | VARCHAR(200) | NOT NULL |

***Example with Data:***

|  |  |  |
| --- | --- | --- |
| **safety\_id** | **mountain\_id** | **safety\_procedure** |
| 401 | 1001 | Use oxygen above 8000m |
| 402 | 1002 | Wear crampons in winter |

**Table 9: climber\_experience**

***Table Description:***

This table tracks the experience level of climbers, such as beginner, intermediate, and expert.

***Table: climber\_experience***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Constraints** |
| experience\_id | Unique identifier for the experience (PK) | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| climber\_id | Links to the climber | INT | Foreign Key (FK) referencing climber(climber\_id), NOT NULL |
| experience\_level | Level of experience (beginner, intermediate, expert) | VARCHAR(50) | NOT NULL |

***Example with Data:***

|  |  |  |
| --- | --- | --- |
| **experience\_id** | **climber\_id** | **experience\_level** |
| 501 | 1223 | Expert |
| 502 | 1224 | Intermediate |

**Table 10: mountain\_funding**

***Table Description:***

This table stores information about funding for mountain expeditions, such as sponsors, organizations, and budget information.

***Table: mountain\_funding***

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Constraints** |
| funding\_id | Unique identifier for the funding | INT | Primary Key (PK), NOT NULL, AUTO\_INCREMENT |
| mountain\_id | Links to the mountain | INT | Foreign Key (FK) referencing mountain(mountain\_id), NOT NULL |
| sponsor\_name | Name of the sponsoring organization | VARCHAR(255) | NOT NULL |
| amount | Amount of funding provided | DECIMAL(10, 2) | NOT NULL |

***Example with Data:***

|  |  |  |  |
| --- | --- | --- | --- |
| **funding\_id** | **mountain\_id** | **sponsor\_name** | **amount** |
| 601 | 1001 | Mountaineers Fund | 50000.00 |
| 602 | 1002 | Adventure Co. | 30000.00 |

## 2.4. Comments on Table Relationships

1. **Climber and Climb:**
   * **One-to-Many**: A climber can have multiple climbs, but each climb is associated with one climber. This is implemented by having climber\_id as a foreign key in the climb table.
2. **Mountain and Climb:**
   * **One-to-Many**: A mountain can be involved in multiple climbs, but each climb corresponds to only one mountain. This relationship is captured by the mountain\_id foreign key in the climb table.
3. **Climber and Mountain (Many-to-Many):**
   * **Many-to-Many**: A climber can climb multiple mountains, and a mountain can be climbed by multiple climbers. This many-to-many relationship is modeled by the climber\_mountain table, which links climber\_id and mountain\_id with start and end dates for each climb.
4. **Climber and Climber\_Address:**
   * **One-to-Many**: A climber can have multiple addresses (e.g., permanent, temporary), but each address belongs to one climber. This relationship is captured by the climber\_id foreign key in the climber\_address table.
5. **Climber and Climber\_Contact:**
   * **One-to-Many**: A climber can have multiple contact methods (e.g., email, phone number), but each contact belongs to one climber. The relationship is maintained by the climber\_id foreign key in the climber\_contact table.
6. **Mountain and Mountain\_Region:**
   * **One-to-Many**: A mountain can have multiple regions or areas it spans, but each region belongs to only one mountain. This relationship is tracked through the mountain\_id foreign key in the mountain\_region table.
7. **Mountain and Mountain\_Safety:**
   * **One-to-Many**: A mountain may have multiple safety procedures associated with it, but each safety procedure is tied to a specific mountain. The relationship is maintained by the mountain\_id foreign key in the mountain\_safety table.
8. **Climber and Climber\_Experience:**

* **One-to-Many**: A **climber** can have one row in the climber\_experience table, which holds their overall experience. The climber\_id field is a **foreign key** in the climber\_experience table, referencing the climber\_id in the climber table.

1. **Mountain and Mountain\_Funding:**
   * **Many-to-One**: A single **mountain** can have multiple **funding records** associated with it, as there can be multiple sponsors or sources of funding for different expeditions on the same mountain. The mountain\_id field is a **foreign key** in the mountain\_funding table, referencing the mountain\_id in the mountain table.