

|  |
| --- |
| Iuliia Kaymak  **Database design for Climbing Club** |

|  |
| --- |
|  |

Contents

[1 Business Description 3](#_Toc164618543)

[1.1 Business background 3](#_Toc164618544)

[1.2 Problems. Current Situation 3](#_Toc164618545)

[1.3 the Benefits of implementing a database. Project Vision 3](#_Toc164618546)

[2 Approach to modelling 4](#_Toc164618547)

[2.1 Business requirements 4](#_Toc164618548)

[2.2 Technical realization 4](#_Toc164618549)

[3 Model description 5](#_Toc164618550)

[3.1 Definitions & Acronyms 5](#_Toc164618551)

[3.2 Logical Scheme 5](#_Toc164618552)

[3.3 Objects 6](#_Toc164618553)

[3.3.1 ‘Climbers’ 6](#_Toc164618557)

[3.3.2 ‘Enrollments’ 7](#_Toc164618558)

[3.3.3 ‘Climbers\_payments’ 8](#_Toc164618559)

[3.3.4 ‘Tours’ 9](#_Toc164618560)

[3.3.5 ‘Tour\_statuses’ 10](#_Toc164618561)

[3.3.6 ‘Transport\_companies’ 11](#_Toc164618562)

[3.3.7 ‘Accomodation\_companies’ 12](#_Toc164618563)

[3.3.8 ‘Routes’ 13](#_Toc164618564)

[3.3.9 ‘Levels’ 14](#_Toc164618565)

[3.3.10 ‘Mountains’ 15](#_Toc164618566)

[3.3.11 ‘Guides’ 15](#_Toc164618567)

# 

# Business Description

## Business background

**Climbing Club** is a fast-developing community that currently organizes climbing tours to mountains in multiple locations. From organizing tours only to Elbrus in Russia, the Climbing Club now offers tours in Türkiye, including climbing to Ararat. The community has pages on social media, including Facebook and Instagram, that attract people, mainly from Russia and Türkiye, to join its tours. Also, Climbing Сlub welcomes requests from businesses willing to order team-building event organization.

As for now, the company’s primary goals are:

1. Scale up their ads in European, American, and Asian markets.
2. Open new routes in already-represented countries and the ones that are not covered yet.
3. Expand the network of agreements with transport and accommodation companies.
4. Develop a flexible system of Referrals and Discounts.

## Problems. Current Situation

As the number of routes, clients, and guides is constantly increasing now, and considering Climbing Club's focus on growth, there is a strong need to implement a database solution that would cover essential aspects of working in the international community.

Currently, the documentation in various formats is stored partially in Russia and Türkiye. Hence, it is challenging for the Sales team to process and track client orders and for the Finance department to calculate the company's revenue and profit and employees' salaries.

Moreover, it is worth mentioning that Management lacks high-quality analytics, which prevents it from reacting appropriately to changing trends and affects the quality of decision-making processes.

## the Benefits of implementing a database. Project Vision

After investigating with the departments of Climbing Club, we formulate the following benefits of implementing a database:

1. The single client base that could be used for operating and marketing.
2. The possibility to track orders and enrollments to tours of all locations.
3. The possibility of tracking received payments from all clients.
4. The possibility to access and update the lists of partners providing accommodation and transportation services.
5. Significantly decreased time and human effort spent collecting all the necessary data and turning it into valuable insights.
6. For the Top Management team, it is an opportunity to make data-driven decisions about high-quality analytics.

# Approach to modelling

## Business requirements

After multiple kick-off sessions and analysis of processes, Business Analysts are forming acceptance criteria that should be covered by a database:

1. There should be a table of clients with their names, dates of birth, addresses, and contact details.
2. There should be a table/ multiple tables with routes, mountains, and tours.
3. There should be a table of guides – employees conducting tours.
4. A primitive status model should be implemented, allowing tours to be tracked by their level of completeness.
5. There should be a table on orders/enrollments to tours. In its turn, it should follow the logic:
6. A single client can be enrolled in multiple tours in different periods.
7. A single tour can have multiple clients enrolled.
8. It should be possible to track the amount of money clients transfer.
9. Determining the partnering transport and/or accommodation companies for a single tour should be possible.

## Technical realization

The model should be in 3rd normal form (3NF). This means:

1. Each table cell should contain an atomic value. In our case, it refers to addresses and areas, e.g., split countries, cities, street addresses, and zip codes into different columns.
2. All the columns in a table should have unique names.
3. Each row in a table must be unique (no rows may be duplicated).
4. Values stored in a column should be of the same data type and the same domain.
5. The order in which data is stored does not matter.
6. A table must have a unique primary key that is used to identify each record differently. In our case, primary keys for dimensional tables can be realized as serial numbers. For fact/operating tables, it is recommended to generate a unique primary key based on candidate keys (e.g., payment codes, enrollment\_ids)
7. Each non-key attribute must be fully functional, dependent on the primary key
8. No non-key attribute may be transitively dependent on a key candidate.

# Model description

## Definitions & Acronyms

We do not specify any definitions and acronyms in the suggested database schema.

Disclaimer: The data provided below in table mock-ups is fiction. Any similarities with reality are accidental.

## Logical Scheme

A screenshot of a computer screen

Description automatically generated

## Objects

### ‘Climbers’

This table contains information about every client who was/is registered for the tours.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| climbers | climber\_id | Serial number of a client. PK | INT |
| climber\_first\_name | Client’s first name | CHAR(100) |
| climber\_last\_name | Client’s last name | CHAR(100) |
| climber\_gender | Client’s gender. Only three options available to choose:  Man / Woman / Non-Binary | CHAR(50) |
| climber\_date\_of\_birth | Client’s date of birth | DATE |
| climber\_country | Client’s country of residence | CHAR(100) |
| climber\_city | Client’s city of residence | CHAR(100) |
| climber\_street\_address | Client’s street address | CHAR(250) |
| climber\_zip\_code | Zip code of street address | CHAR(25) |
| clmber\_email | Client’s email | CHAR(250) |
| climber\_phone\_number | Client’s phone number | CHAR(50) |

Data mock-up:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| climber\_id | climber\_first\_name | climber\_last\_name | climber\_gender | climber\_date\_of\_birth | climber\_country | climber\_city | climber\_street\_address | climber\_zip\_code | climber\_email | climber\_phone\_number |
| 1 | John | Smith | Man | 23/01/1996 | USA | Dallas | 720 W Mockingbird Ln | 75247 | john.smith@gmail.com | +12146880800 |
| 2 | Jenny | Ferguisson | Woman | 31/03/1985 | Sweden | Stockholm | Bryggargatan 10 | 11121 | jenny\_ferg01@gmail.com | +46735908857 |
| 3 | Polly | Miller | Non-Binary | 13/12/2004 | Germany | Munich | Blumenstraße 4 | 80331 | polly\_molly@icloud.com | +4989592122 |
| 4 | Sergei | Mikhailov | Man | 01/01/1990 | Russia | Moscow | Ulitsa Arhitektora Schuseva 1 | 115432 | serg\_m010190@mail.ru | +79993456782 |
| 5 | Ozan | Nigde | Man | 12/12/2001 | Türkiye | Istanbul | İnonu Cd. 83A | 34381 | ozee12@gmail.com | +905413416743 |
| 6 | Makbule | Nigde | Woman | 27/02/2002 | Türkiye | Istanbul | İnonu Cd. 83A | 34381 | makbule\_guzel@gmail.com | +905534867432 |

The table connects to:

1. ‘**enrollments’** table via ‘**climbers.climber\_id**’ and ‘**enrollments.climber\_id**’ using a one-to-many relationship.

### ‘Enrollments’

This is an intermediary linking table connecting climbers and the tours they are registered to. One climber can be registered for many tours, and one tour can have more than one registered climber.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| enrollments | enrollment\_id | Generated field that concatenates **tour\_id** and **climber\_id**. ID of enrollment of specific climbers to specific tours. PK | CHAR(100) |
| tour\_id | ID of a tour. FK1 | CHAR(50) |
| climber\_id | Serial number of a client. FK2 | INT |

Data mock-up:

|  |  |  |
| --- | --- | --- |
| Enrollment\_id | tour\_id | climber\_id |
| TR0000200002\_1 | TR0000200002 | 1 |
| TR0000200002\_2 | TR0000200002 | 2 |
| TR0000200002\_3 | TR0000200002 | 3 |
| TR0000200003\_4 | TR0000200003 | 4 |
| TR0000200003\_5 | TR0000200003 | 5 |
| TR0000200003\_6 | TR0000200003 | 6 |
| RU0000100002\_4 | RU0000100002 | 4 |
| RU0000100002\_1 | RU0000100002 | 1 |
| RU0000100002\_2 | RU0000100002 | 2 |

The table connects to:

1. **‘climbers’** table via **‘enrollments.climber\_id**’ and **‘climbers.climber\_id’** using a many-to-one relationship (mentioned above).
2. **‘climbers\_payments’** table via **‘enrollments.enrollment\_id’** and **‘climbers\_payments.enrollment\_id’** using a one-to-many relationship.
3. **‘tours’** table via **‘enrollments.tour\_id’** and **‘tours.tour\_id’** using a many-to-one relationship.

### ‘Climbers\_payments’

This table contains the data regarding received payments from clients to the company’s bank account.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| climbers\_payments | transaction\_id | Generated field that concatenates **enrollment\_id** and **payment\_code.** Identifies payment(s) belonging to specific enrollment\_id. PK | CHAR(250) |
| payment\_code | Encrypted code of payment. | CHAR(100) |
| enrollment\_id | Enrollment\_id of a specific climber to a specific tour. FK | CHAR(100) |
| transaction\_date | Date of transaction when money are received in the company’s bank account. | DATE |
| amount\_usd | Amount of transferred money in US Dollars. | FLOAT |

Data mock-up:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| transaction\_id | payment\_code | enrollment\_id | transaction\_date | amount\_usd |
| TR0000200002\_1#payment04ab34c45d3a35 | 04ab34c45d3a35 | TR0000200002\_1 | 25/03/2024 | 240.00 |
| TR0000200002\_2#payment04ab34c45d3a35 | 04ab34c45d3a35 | TR0000200002\_2 | 25/03/2024 | 240.00 |
| TR0000200002\_3#payment04ab34c45d3a35 | 04ab34c45d3a35 | TR0000200002\_3 | 25/03/2024 | 240.00 |
| RU0000100002\_4#payment1ab38c435bd42a | 1ab38c435bd42a | RU0000100002\_4 | 16/04/2024 | 170.00 |
| RU0000100002\_4#payment63bad401bd3241 | 63bad401bd3241 | RU0000100002\_4 | 18/04/2024 | 170.00 |
| RU0000100002\_1#payment2abg356gcv67fd | 2abg356gcv67fd | RU0000100002\_1 | 18/04/2024 | 340.00 |
| RU0000100002\_2#payment26aa7cd00e24df | 26aa7cd00e24df | RU0000100002\_2 | 20/04/2024 | 340.00 |

The table connects to:

1. **‘enrollments’** table via **‘climbers\_payments.enrollment\_id’** and **‘enrollments.enrollment\_id’** using a many-to-one relationship (mentioned above).

### ‘Tours’

This table contains the data regarding every tour announced and/or organized by the company.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| tours | tour\_id | Unique ID of organized tour. Contains route\_id and ordering number. PK | CHAR(50) |
| route\_id | ID of route. FK1 | CHAR(50) |
| price\_usd | Price of a tour in US Dollars | FLOAT |
| planned\_start\_date | Planned start date of a tour | DATE |
| planned\_end\_date | Planned end date of a tour | DATE |
| actual\_start\_date | Actual start date of a tour | DATE |
| actual\_end\_date | Actual end date of a tour | DATE |
| tour\_status\_id | ID of the current status of a tour | INT |
| tour\_status\_comments | Comments related to a tour | CHAR(250) |
| partner\_transport\_company | ID of a transport company with which the contract is concluded. | INT |
| partner\_accomodation\_company | ID of an accommodation company with which the contract is concluded. | INT |

Data mock-up:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| tour\_id | route\_id | price\_usd | planned\_start\_date | planned\_end\_date | actual\_start\_date | actual\_end\_date | tour\_status\_id | tour\_status\_comments | partner\_transport\_company | partner\_accomodation\_company |
| TR0000200001 | TR00002 | 220.00 | 25/03/2024 | 28/03/2024 |  |  | 4 | The group was not formed | 2 | 1 |
| TR0000200002 | TR00002 | 240.00 | 01/04/2024 | 04/04/2024 | 08/04/2024 | 12/04/2024 | 3 | Client organisation changed the period of team building | 2 | 1 |
| TR0000200003 | TR00002 | 240.00 | 15/04/2024 | 19/04/2024 | 15/04/2024 | 19/04/2024 | 3 |  | 1 | 1 |
| RU0000100001 | RU00001 | 310.00 | 25/03/2024 | 28/03/2024 | 25/03/2024 | 28/03/2024 | 3 |  | 4 | 4 |
| RU0000100002 | RU00001 | 340.00 | 01/05/2024 | 04/05/2024 |  |  | 1 |  |  |  |

The table connects to:

1. **‘enrollments’** table via **‘tours.tour\_id’** and **‘enrollments.tour\_id’** using a one-to-many relationship (mentioned above).
2. **’routes’** table via **‘tours.route\_id’** and **‘routes.route\_id’** using a many-to-one relationship.
3. **‘tour\_statuses’** table via **‘tours.tour\_status\_id’** and **‘tour\_statuses.tour\_status\_id’** using a many-to-one relationship.
4. **‘transport\_companies’** table via **‘tours.partner\_transport\_company’** and **‘transport\_companies.transport\_company\_id’** using a many-to-one relationship.
5. **‘accomodation\_companies’** table via **‘tours.partner\_accomodation\_company’** and **‘accomodation\_companies.accomodation\_company\_id’** using a many-to-one relationship.

### ‘Tour\_statuses’

This table contains the dictionary of available statuses of tours.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| tour\_statuses | tour\_status\_id | ID of a status. PK | INT |
| tour\_status\_name | Name of status. | CHAR(100) |

Data mock-up:

|  |  |
| --- | --- |
| tour\_status\_id | tour\_status\_name |
| 1 | Wait for Start |
| 2 | In Progress |
| 3 | Completed |
| 4 | Cancelled |

The table connects to:

1. **‘tours’** table via **‘tour\_statuses.tour\_status\_id’** and **‘tours.tour\_status\_id’** using a one-to-many relationship (mentioned above).

### ‘Transport\_companies’

This table lists transport companies that at least once had partner agreements with the Climbing Club.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| transport\_companies | transport\_company\_id | ID of a transport company. PK | INT |
| transport\_company\_name | Name of a transport company. | CHAR(250) |
| country | Country of where a transport company provides their services. | CHAR(100) |
| representative\_full\_name | Full name of a representative of a transport company. | CHAR(250) |
| representative\_email | Email address of a representative of a transport company. | CHAR(250) |
| representative\_phone\_number | Phone number of a representative of a transport company. | CHAR(250) |

Data mock-up:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| transport\_company\_id | transport\_company\_name | country | representative\_full\_name | representative\_email | representative\_phone\_number |
| 1 | İlkbahar | Türkiye | Ömer Kılıç | omer\_kilic@ilkbahar.com | +905416999319 |
| 2 | Kırmızı Bayrak | Türkiye | Hakan Aydın | aydin.hakan@krmzbyrk.tr | +905537110101 |
| 3 | Edelveis | Russia | Olga Khostova | olga\_khvostova@edelveis.ru | +79275435723 |
| 4 | Maksima Group | Russia | Maksim Maksimov | maksimov.maksim@maksimagr.com | +78006005510 |

The table connects to:

1. **‘tours’** table via **‘transport\_companies.transport\_company\_id’** and **‘tours.partner\_transport\_company’** using a one-to-many relationship (mentioned above).

### ‘Accomodation\_companies’

This table lists transport companies that at least once had partner agreements with the Climbing Club.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| accomodation\_companies | accomodation\_company\_id | ID of an accommodation company. PK | INT |
| accomodation\_company\_name | Name of an accommodation company. | CHAR(250) |
| accomodation\_company\_type | Type of an accommodation company. | CHAR(50) |
| country | Country of where an accommodation company provides their services. | CHAR(100) |
| representative\_full\_name | Full name of a representative of an accommodation company. | CHAR(250) |
| representative\_email | Email address of a representative of an accommodation company. | CHAR(250) |
| representative\_phone\_number | Phone number of a representative of an accommodation company. | CHAR(250) |

Data mock-up:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| accomodation\_company\_id | accomodation\_company\_name | accommodation\_company\_type | country | representative\_full\_name | representative\_email | representative\_phone\_number |
| 1 | Park Inn | Hotel | Türkiye | Necla Calman | necla.calman@parkinn.com | +905417842130 |
| 2 | Suda Hotels | Hostel | Türkiye | Mohammed Kizilagac | mkizilagac@suda.tr | +905517605624 |
| 3 | Aqua Light Hotel | Hotel | Russia | Ivan Svetlov | ivan\_svetlov@aqualight.ru | +79277093794 |
| 4 | Hostel Net | Hostel | Russia | Angelina Krasnova | mr@hostel.net | +78806793451 |

The table connects to:

1. **‘tours’** table via **‘accomodation\_companies.accomodation\_company\_id’** and **‘tours.partner\_accomodation\_company’** using a one-to-many relationship (mentioned above).

### ‘Routes’

This table lists all the routes that the Climbing Club refers to while organizing various tours.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| routes | route\_id | ID of a route. PK | CHAR(50) |
| route\_name | Name of a route. | CHAR(250) |
| route\_level\_id | ID of a difficulty level of a route. FK1 | INT |
| route\_mountain\_id | ID of a mountain where a route belongs to. FK2 | INT |
| route\_guide\_id | ID of a guide that conducts the route. FK3 | CHAR(50) |

Data mock-up:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| route\_id | route\_name | route\_level\_id | route\_mountain\_id | route\_guide\_id |
| TR00001 | Beautiful Ararat | 1 | 1 | 16TR001 |
| TR00002 | Ararat and History | 2 | 1 | 16TR001 |
| TR00003 | Symbolic Ararat | 3 | 1 | 10TR402 |
| TR00004 | Meet Suphan | 1 | 2 | 9TR346 |
| TR00005 | Suphan and its Legends | 2 | 2 | 9TR346 |
| TR00006 | Unknown Suphan | 3 | 2 | 10TR402 |
| RU00001 | Get to know Elbrus | 2 | 3 | RU7934 |
| RU00002 | PRO-climbing to Elbrus | 3 | 3 | RU7934 |

The table connects to:

1. **‘tours’** table via **‘routes.route\_id’** and **‘tours.route\_id’** using a one-to-many relationship (mentioned above).
2. ‘**levels’** table via **‘routes.route\_level\_id’** and **‘levels.level\_id’** using a many-to-one relationship.
3. ‘**mountains’** table via **‘routes.route\_mountain\_id’** and **‘mountains.mountain\_id’** using a many-to-one relationship.
4. **‘guides’** table via **‘routes.route\_guide\_id’** and **‘guides.guide\_id’** using a many-to-one relationship.

### ‘Levels’

This table is a dictionary of difficulty levels that are being set to the existing routes.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| levels | level\_id | ID of a difficulty level. PK | INT |
| level\_name | Name of a difficulty level | CHAR(50) |
| level\_age\_constraint | Age constraint related to a difficulty level. | INT |

Data mock-up:

|  |  |  |
| --- | --- | --- |
| level\_id | level\_name | level\_age\_constraint |
| 1 | Light | 16 |
| 2 | Medium | 18 |
| 3 | Hard | 18 |

The table connects to:

1. **‘routes’** table via **‘levels.level\_id’** and **‘routes.route\_level\_id’** using a one-to-many relationship (mentioned above).

### ‘Mountains’

This table is a list of mountains that the company includes in their routes.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| mountains | mountain\_id | ID of a mountain. PK | INT |
| mountain\_name | Name of a mountain. | CHAR(100) |
| mountain\_height\_meters | Height of a mountain in meters. | INT |
| mountain\_country | Country of where a mountain is located. | CHAR(100) |
| mountain\_province | Province of where a mountain is located. | CHAR(100) |

Data mock-up:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mountain\_id | mountain\_name | mountain\_height\_meter | mountain\_country | mountain\_province |
| 1 | Ararat | 5137 | Türkiye | Iğdır |
| 2 | Suphan | 4049 | Türkiye | Bitlis |
| 3 | Elbrus | 5642 | Russia | Kabardino-Balkaria |

The table connects to:

1. **‘routes’** table via **‘mountains.mountain\_id’** and **‘routes.route\_mountain\_id’** using a one-to-many relationship (mentioned above).

### ‘Guides’

This table lists guides (employees) who conduct the routes.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| guides | guide\_id | ID of a guide. PK | CHAR(50) |
| guide\_first\_name | Guide’s first name. | CHAR(100) |
| guide\_last\_name | Guide’s last name. | CHAR(100) |
| guide\_license\_number | Guide’s license number. | CHAR(50) |
| guide\_date\_of\_birth | Guide’s date of birth. | DATE |
| guide\_experience\_years | Guide’s experience years in climbing. | INT |

Data mock-up:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| guide\_id | guide\_first\_name | guide\_last\_name | guide\_license\_number | guide\_date\_of\_birth | guide\_experience\_years |
| 16TR001 | Mehmet | Karaoglu | 1453678956 | 01/04/1986 | 15 |
| 10TR402 | Ali | Kupas | 1234580734 | 10/01/1992 | 10 |
| 9TR346 | Haci Ali | Onder | 8902343562 | 13/08/1997 | 7 |
| RU7934 | Nikolay | Ivanov | 6754342454 | 12/12/1988 | 15 |

The table connects to:

1. **‘routes’** table via **‘guides.guide\_id’** and **‘routes.route\_guide\_id’** using a one-to-many relationship (mentioned above).