

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра информационных технологий

ПРОЕКТНАЯ РАБОТА

БАЗА ДАННЫХ ТУРИСТИЧЕСКОГО КЛУБА
(задача 13)

Работу выполнила _____ А.М. Слесаренко

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность Разработка баз данных

Руководитель
канд. физ. мат. наук, доц. _____ А.А. Евдокимов

Краснодар
2023

Содержание

1	Словарь терминов и сокращений.....	3
2	Описание предметной области.....	3
3	Варианты использования. Диаграмма Use Case.....	4
4	Потоки событий.....	5
5	Виды запросов к информационной системе.....	8
6	Логическая и физическая схемы базы данных. ER-диаграммы.....	9
7	Справочная информация по объектам ИС.....	10
8	Отношения (связи).....	28
9	SQL-код создания БД.....	33
10	Триггеры, джобсы, процедуры, функции.....	38
11	Ограничения на применение схемы.....	39
12	Запросы к БД.....	39

1 Словарь терминов и сокращений

Тренер — турист, тренирующий группу, административно относящийся к одной из секций.

Спортсмен — турист, занимающийся в секции.

Инструктор — тренер или спортсмен, возглавляющий поход.

Плановый поход — поход, у которого существует точный план, в котором указывается маршрут, расписание привалов и стоянок на каждый день.

Неплановый поход — поход, который имеет только маршрут и полное время его прохождения.

max_complexity – категория максимально сложного из пройденных туристом плановых походов.

time_tr – длительность тренировки.

hike_days – число дней, отведенных на поход.

2 Описание предметной области

Туристы, приходящие в туристический клуб, могут не только ходить в плановые походы, но и заниматься в различных секциях в течение всего года. Для этого они записываются в группы, относящиеся к определенным секциям.

Туристов можно условно разделить на любителей, спортсменов и тренеров. Каждая из перечисленных категорий может иметь свой набор характеристик-атрибутов. Секции клуба возглавляются руководителями, в функции которых входит контроль за работой секции. В работе секции участвуют тренеры, административно относящиеся к одной из секций. Руководитель секции назначает каждой группе тренера. Тренер может тренировать несколько групп, причем необязательно принадлежащих его секции. Спортсмены и тренеры могут участвовать в различных соревнованиях.

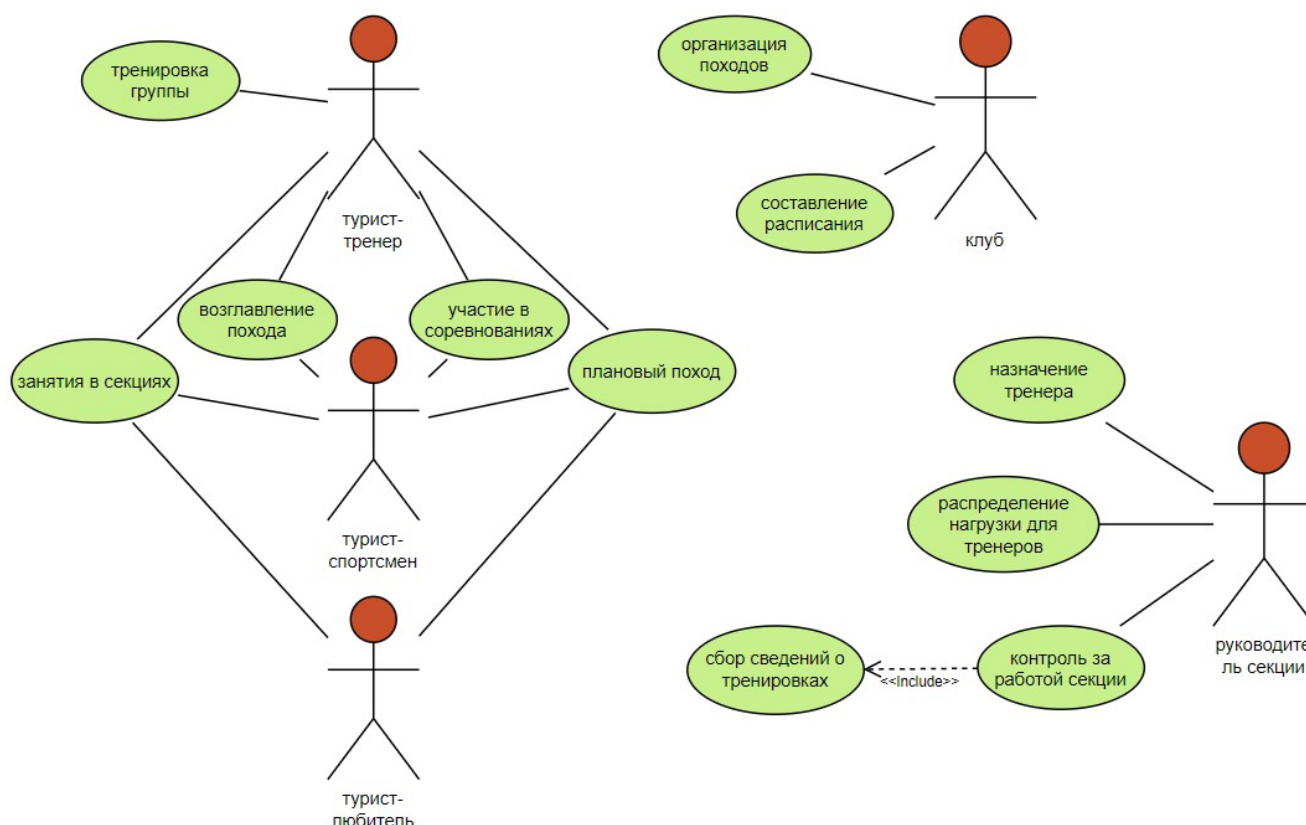
Каждый год составляется расписание работы секций. В нем указывается, какие будут проводиться тренировки и в каких секциях: их количество, место, время и т.д. В соответствии с этим руководители секций осуществляют распределение нагрузки для тренеров (с учетом их специальности). Сведения о проведенных тренировках и посещаемости тренировок собираются руководителями.

В течение года клуб организует различные походы. Каждый поход имеет свой маршрут, на который отводится определенное количество дней. По маршруту и количеству дней определяется категория сложности данного

похода. Поход возглавляет инструктор, которым может быть какой-либо тренер или спортсмен. Он набирает группу в количестве 5-15 человек для своего похода, исходя из типа похода (пеший, конный, водный, горный) и физических данных туристов (по их занятиям в секциях: водники, спелеологи, альпинисты и другие, с учетом специфики занятий - не умеющего плавать никогда не возьмут на сплав, а в пеший поход небольшой категории сложности могут взять любого туриста). Инструктор может водить в походы данной категории сложности, если он сам ее ранее уже прошел.

Походы могут быть плановыми и неплановыми. Для каждого планового похода существует точный план в котором указывается маршрут, расписание привалов и стоянок на каждый день. Во время планового похода ведется дневник. Неплановые походы имеют только маршрут и полное время его прохождения. Неплановый поход может быть переведен в категорию плановых. Каждому туристу присваивается категория максимально сложного из пройденных им плановых походов.

3 Варианты использования. Диаграмма Use Case



4 Потоки событий

Поток события «Тренировка группы»

Предусловие: руководитель назначил группе тренера

Основной поток:

1) Тренер приходит на тренировку

A1: Тренер заболел (или не смог прийти по иной уважительной причине)

1. Отменить или перенести тренировку / назначить временно другого тренера

2. Провести тренировку

E1: Тренер перестал быть тренером

1. Руководитель назначает группе нового тренера

2. Новый тренер проводит тренировку

2) Тренер отмечает присутствующих

3) Тренер проводит тренировку

Постусловие: тренировка проведена

Поток события «Возглавление похода»

Предусловие: турист является тренером или спортсменом

Основной поток:

1) Турист проходит поход и получает категорию

2) Турист становится инструктором

3) Инструктор набирает группу

A1: недостаточно желающих отправиться в поход

1. Поход переносится или меняется его план

2. Проводится повторный набор группы

4) Инструктор возглавляет группу в походе и ведет дневник, если поход плановый

E1: инструктор не смог возглавить группу

1. Поход переносится или назначается другой инструктор

2. Инструктор ведет группу в поход

Постусловие: поход пройден

Поток события «Участие в соревнованиях»

Предусловие: турист является тренером или спортсменом

Основной поток:

1) Турист сообщает, что хочет участвовать в соревнованиях

2) Турист готовится к соревнованиям

3) Турист приходит на соревнования

A1: соревнования перенесли по какой-либо причине

1. Участникам сообщается новая дата
2. Турист участвует в соревновании

E1: турист не пришел на соревнование

1. Турист выбывает из данного соревнования
2. Турист участвует в следующем соревновании (по желанию)

Постусловие: турист принял участие в соревновании

Поток события «Занятия в секциях»

Предусловие: турист пришел в туристический клуб

Основной поток:

- 1) Турист записался в группу, относящуюся к определенной секции
- 2) Турист посещает занятия

E1: турист не посещает занятия

1. Туристу напоминают о занятиях

Постусловие: турист занимается в секции

Поток события «Плановый поход»

Предусловие: турист пришел в туристический клуб

Основной поток:

- 1) Турист выбирает поход
- 2) Туриста берут в команду

A1: туриста не берут в этот поход (высокая сложность или иная причина)

1. Туристу предлагают подходящий поход
2. Туриста берут в команду

A2: команда набрана

1. Туристу предлагают другой поход
2. Туриста берут в команду

3) Турист отправляется в поход

E1: турист не пошел в поход

1. Туристу предлагают другой поход
2. Туриста берут в команду
3. Турист отправляется в поход

Постусловие: турист прошел поход и получил категорию, если она выше, чем у него была (если поход - плановый)

Поток события «Организация походов»

Предусловие: поход не организован

Основной поток:

- 1) Клуб выбирает маршрут похода и определяет число дней
- 2) Определяется категория сложности данного похода
- 3) Составляется план для планового похода

Постусловие: поход организован

Поток события «Составление расписания»

Предусловие: расписание не составлено

Основной поток:

- 1) Клуб определяет какие будут проводиться тренировки и в каких секциях
 - 2) Определяется количество, место, время проведения тренировок и т.д.
- Постусловие: расписание составлено

Поток события «Назначение тренера»

Предусловие: у группы нет тренера

Основной поток:

- 1) Руководитель выбирает более свободного тренера
- 2) Руководитель назначает тренера группе

Постусловие: тренер назначен

Поток события «Распределение нагрузки для тренеров»

Предусловие: составлено расписание

Основной поток:

- 1) Руководитель узнает специальности тренеров
- 2) В соответствии с этим распределяет нагрузки для тренеров

Постусловие: распределены нагрузки для тренеров

Поток события «Сбор сведений о тренировках»

Предусловие: тренера назначены, нагрузки для тренеров распределены

Основной поток:

- 1) Руководитель секции обращается к тренерам
- 2) Руководитель узнает были ли проведены тренировки
- 3) Руководитель собирает сведения о посещаемости тренировок

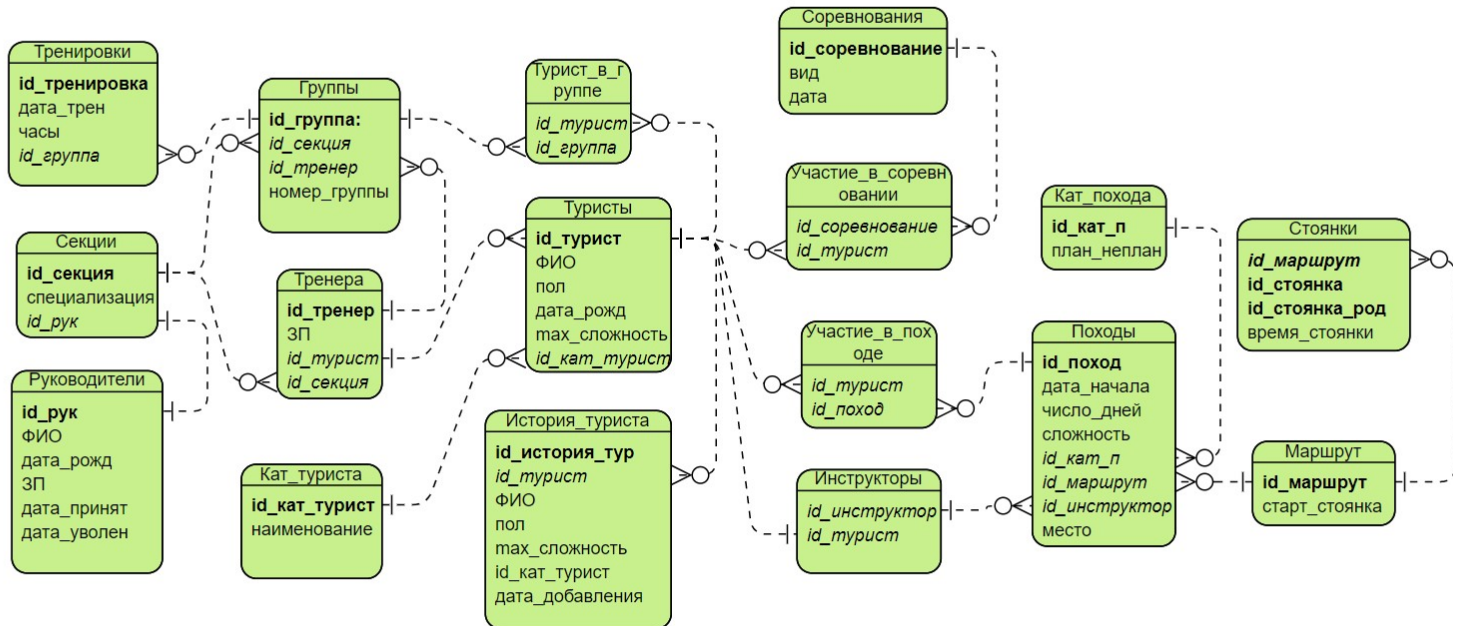
Постусловие: сведения собраны

5 Виды запросов к информационной системе

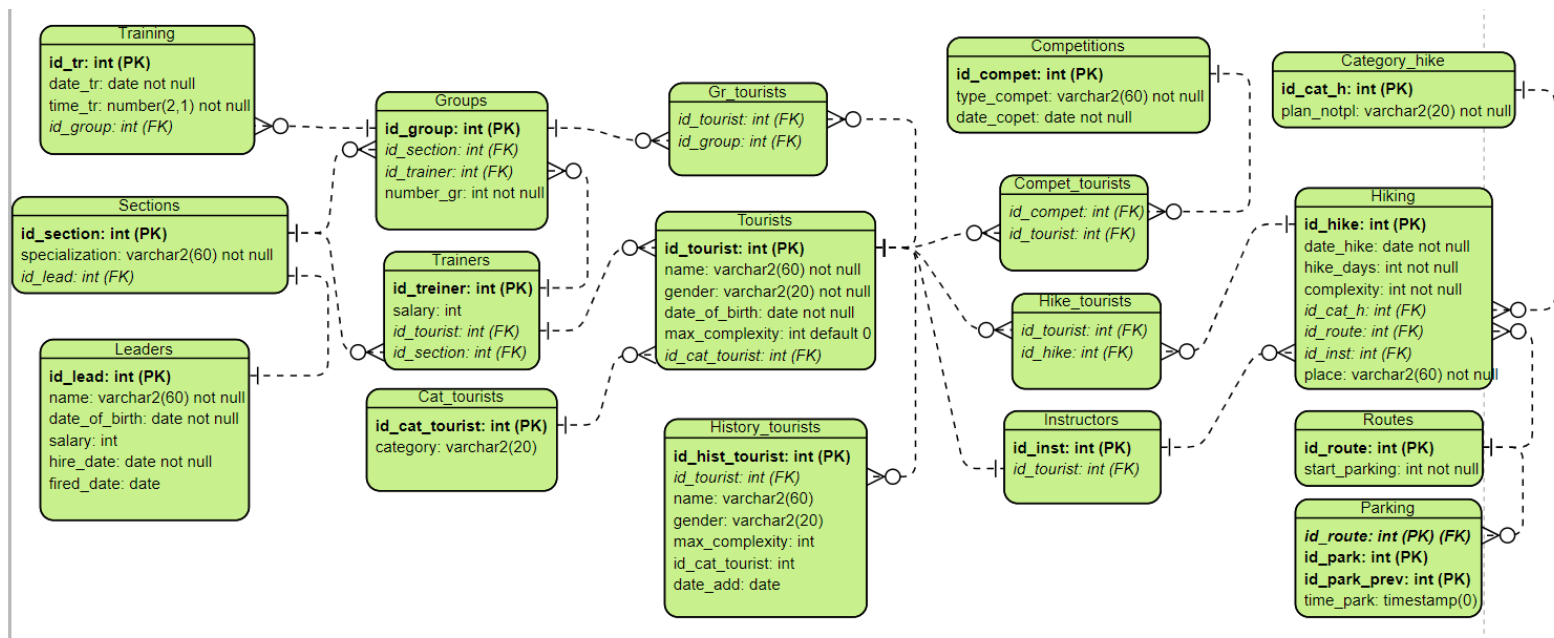
1. Получить список и общее число туристов, занимающихся в клубе, в указанной секции, группе, по половому признаку, году рождения, возрасту.
2. Получить список и общее число тренеров указанной секции, по всем секциям, по половому признаку, по возрасту, по размеру заработной платы, специализации.
3. Получить перечень и общее число соревнований, в которых участвовали спортсмены из указанной секции, по всем секциям.
4. Получить список тренеров, проводивших тренировки в указанной группе, за указанный период времени.
5. Получить список и общее число туристов из некоторой секции, группы, которые ходили в заданное количество походов, ходили в указанный поход, ходили в поход в обозначенное время, ходили по определенному маршруту, были в некоторой точке, имеют соответствующую категорию.
6. Получить перечень руководителей секций полностью, по размеру заработной платы, по году рождения, возрасту, году поступления на работу.
7. Получить нагрузку тренеров (вид занятий, количество часов), ее объем по определенным видам занятий и общую нагрузку за указанный период времени для данного тренера или указанной секции.
8. Получить перечень и общее число маршрутов, по которым ходили туристы из указанной секции, в обозначенный период времени, по которым водил свои группы данный инструктор, по которым прошло указанное количество групп.
9. Получить перечень и общее число маршрутов, которые проходят через некоторую точку, имеют длину больше указанной, могут удовлетворять заданной категории сложности.
10. Получить перечень и общее число туристов из указанной секции, группы, которые могут ходить в указанные типы походов.
11. Получить перечень и общее число инструкторов, инструкторов-спортсменов, инструкторов-тренеров, которые имеют определенную категорию, которые ходили в указанное количество походов, ходили в определенный поход, ходили по некоторому маршруту, были в указанной точке
12. Получить список туристов из указанной секции, группы, которые ходили в походы со своим тренером в качестве инструктора.
13. Получить список туристов из некоторой секции, группы, которые ходили по всем маршрутам, по указанным маршрутам.

6 Логическая и физическая схемы базы данных. ER-диаграммы

Логическая схема базы данных:



Физическая схема базы данных:



7 Справочная информация по объектам ИС

Объект/инструмент	Описание объекта
Таблица Tourists	<p>Назначение: содержит информацию о туристах, приходящих в туристический клуб</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Cat_tourists - таблица Trainers - таблица Gr_tourists - таблица Compet_tourists - таблица Hike_tourists - таблица Instructors - таблица History_tourists <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_tourist <p>Назначение: идентификатор туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_TOURIST_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Trainers, таблица Gr_tourists, таблица History_tourists, таблица Hike_tourists, таблица Compet_tourists, таблица Instructors</p> - столбец name <p>Назначение: ФИО туриста</p> <p>Тип: varchar2(60)</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> - столбец gender <p>Назначение: пол туриста</p> <p>Тип: varchar2(20)</p> <p>Ограничения целостности уровня столбца: not null</p>

	<p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец date_of_birth</p> <p>Назначение: дата рождения туриста</p> <p>Тип: date</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец max_complexity</p> <p>Назначение: максимальная сложность похода, пройденного туристом</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: default 0</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец id_cat_tourist</p> <p>Назначение: идентификатор туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_CAT_TOURIST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Cat_tourists</p>
Таблица Cat_tourists	<p>Назначение: содержит информацию о категориях туриста</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <p>- таблица Tourists</p> <p>Атрибуты:</p> <p>- столбец id_cat_tourist</p> <p>Назначение: идентификатор категории туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_CAT_TOURIST_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Tourists</p>

	<p>- столбец category Назначение: категория туриста Тип: varchar2(20) Ограничения целостности уровня столбца: отсутствуют Индексация: отсутствует Связанные объекты: отсутствуют</p>
Таблица Trainers	<p>Назначение: содержит информацию о тренерах Тип таблицы: heap Ограничения целостности уровня таблицы: отсутствуют Связанные объекты: - таблица Tourists - таблица Sections - таблица Groups Атрибуты: - столбец id_trainer Назначение: идентификатор тренера Тип: int Ограничения целостности уровня столбца: первичный ключ ID_TRAINER_PK Индексация: отсутствует Связанные объекты: таблица Groups - столбец salary Назначение: зарплата тренера Тип: int Ограничения целостности уровня столбца: отсутствуют Индексация: отсутствует Связанные объекты: отсутствуют - столбец id_tourist Назначение: идентификатор туриста Тип: int Ограничения целостности уровня столбца: внешний ключ ID_TRAINER_FK Индексация: отсутствует Связанные объекты: таблица Tourists</p>

	<p>- столбец id_section Назначение: идентификатор секции Тип: int Ограничения целостности уровня столбца: отсутствуют Индексация: отсутствует Связанные объекты: таблица Sections</p>
Таблица Leaders	<p>Назначение: содержит информацию о руководителях Тип таблицы: heap Ограничения целостности уровня таблицы: отсутствуют Связанные объекты: - таблица Sections Атрибуты: - столбец id_lead Назначение: идентификатор руководителя Тип: int Ограничения целостности уровня столбца: первичный ключ ID_LEAD_PK Индексация: отсутствует Связанные объекты: таблица Sections - столбец name Назначение: ФИО руководителя Тип: varchar2(60) Ограничения целостности уровня столбца: not null Индексация: отсутствует Связанные объекты: отсутствуют - столбец date_of_birth Назначение: дата рождения руководителя Тип: date Ограничения целостности уровня столбца: not null Индексация: отсутствует Связанные объекты: отсутствуют - столбец salary Назначение: зарплата руководителя Тип: int Ограничения целостности уровня столбца:</p>

	<p>отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец hire_date</p> <p>Назначение: дата найма руководителя</p> <p>Тип: date</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец fired_date</p> <p>Назначение: дата увольнения руководителя</p> <p>Тип: date</p> <p>Ограничения целостности уровня столбца:</p> <p>отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p>
Таблица Sections	<p>Назначение: содержит информацию о секциях</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы:</p> <p>отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Leaders - таблица Trainers - таблица Groups <p>Атрибуты:</p> <p>- столбец id_section</p> <p>Назначение: идентификатор секции</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>первичный ключ ID_SECTION_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Trainers, таблица Groups</p> <p>- столбец specialization</p> <p>Назначение: специализация секции</p> <p>Тип: varchar2(60)</p> <p>Ограничения целостности уровня столбца: not null</p>

	<p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец id_lead</p> <p>Назначение: идентификатор руководителя</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_LEAD_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Leaders</p>
Таблица Groups	<p>Назначение: содержит информацию о группах</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Sections - таблица Trainers - таблица Gr_tourists - таблица Training <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_group <p>Назначение: идентификатор группы</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_GROUP_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Gr_tourists</p> <ul style="list-style-type: none"> - столбец number_gr <p>Назначение: номер группы в секции</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <ul style="list-style-type: none"> - столбец id_section <p>Назначение: идентификатор секции</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_SECTION_FK</p>

	<p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Sections</p> <p>- столбец id_trainer</p> <p>Назначение: идентификатор тренера</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_TRAINER_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Trainers</p>
Таблица Gr_tourists	<p>Назначение: содержит информацию о туристах в группах</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Tourists - таблица Groups <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_tourist <p>Назначение: идентификатор туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_TOURIST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Tourists</p> <ul style="list-style-type: none"> - столбец id_group <p>Назначение: идентификатор группы</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_GROUP_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Groups</p>
Таблица History_tourists	<p>Назначение: содержит информацию об измененных данных туристов</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы:</p>

	<p>отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Tourists <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_hist_tourist <p>Назначение: идентификатор записи истории туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>первичный ключ ID_HIST_TOURIST_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <ul style="list-style-type: none"> - столбец id_tourist <p>Назначение: идентификатор туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>внешний ключ ID_TOURIST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Tourists</p> <ul style="list-style-type: none"> - столбец name <p>Назначение: ФИО туриста</p> <p>Тип: varchar2(60)</p> <p>Ограничения целостности уровня столбца:</p> <p>отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <ul style="list-style-type: none"> - столбец gender <p>Назначение: пол туриста</p> <p>Тип: varchar2(20)</p> <p>Ограничения целостности уровня столбца:</p> <p>отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <ul style="list-style-type: none"> - столбец max_complexity <p>Назначение: максимальная сложность похода, пройденного туристом</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p>
--	---

	<p>отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец id_cat_tourist</p> <p>Назначение: идентификатор категории туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец date_add</p> <p>Назначение: дата добавления записи</p> <p>Тип: date</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p>
Таблица Training	<p>Назначение: содержит информацию о тренировках</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы:</p> <p>отсутствуют</p> <p>Связанные объекты:</p> <p>- таблица Groups</p> <p>Атрибуты:</p> <p>- столбец id_tr</p> <p>Назначение: идентификатор тренировки</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>первичный ключ ID_TR_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец date_tr</p> <p>Назначение: дата тренировки</p> <p>Тип: date</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец time_tr</p>

	<p>Назначение: время (длительность) тренировки</p> <p>Тип: number(2,1)</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец id_group</p> <p>Назначение: идентификатор группы</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>внешний ключ ID_GROUP_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Groups</p>
<p>Таблица</p> <p>Compet_tourists</p>	<p>Назначение: содержит информацию о том, в каких соревнованиях участвовал турист</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы:</p> <p>отсутствуют</p> <p>Связанные объекты:</p> <p>- таблица Competitions</p> <p>- таблица Tourists</p> <p>Атрибуты:</p> <p>- столбец id_tourist</p> <p>Назначение: идентификатор туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>внешний ключ ID_TOURIST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Tourists</p> <p>- столбец id_compet</p> <p>Назначение: идентификатор соревнования</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца:</p> <p>внешний ключ ID_COMPET_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Competitions</p>
Таблица Competitions	<p>Назначение: содержит информацию соревнованиях</p>

	<p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты: - таблица CompetitionTourists</p> <p>Атрибуты: - столбец id_compet Назначение: идентификатор соревнования Тип: int Ограничения целостности уровня столбца: первичный ключ ID_COMPET_PK Индексация: отсутствует Связанные объекты: таблица Competition_tourists - столбец type_compet Назначение: тип соревнования Тип: varchar2(60) Ограничения целостности уровня столбца: not null Индексация: отсутствует Связанные объекты: отсутствуют - столбец date_compet Назначение: дата проведения соревнования Тип: date Ограничения целостности уровня столбца: not null Индексация: отсутствует Связанные объекты: отсутствуют</p>
Таблица Hike_tourists	<p>Назначение: содержит информацию о том, в каких походах участвовал турист</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты: - таблица Hiking - таблица Tourists</p> <p>Атрибуты: - столбец id_tourist Назначение: идентификатор туриста Тип: int</p>

	<p>Ограничения целостности уровня столбца: внешний ключ ID_TOURIST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Tourists</p> <p>- столбец id_hike</p> <p>Назначение: идентификатор похода</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_HIKE_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Hiking</p>
Таблица Hiking	<p>Назначение: содержит информацию о походах</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Hike_tourists - таблица Instructors - таблица Category_hike - таблица Routes <p>Атрибуты:</p> <p>- столбец id_hike</p> <p>Назначение: идентификатор похода</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_HIKE_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Hike_tourists</p> <p>- столбец date_hike</p> <p>Назначение: дата похода</p> <p>Тип: date</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец hike_days</p> <p>Назначение: дата похода</p> <p>Тип: int</p>

	<p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец place</p> <p>Назначение: место похода</p> <p>Тип: varchar2(60)</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец complexity</p> <p>Назначение: сложность похода</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец id_cat_h</p> <p>Назначение: индекс категории похода</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_CAT_H_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Category_hike</p> <p>- столбец id_route</p> <p>Назначение: индекс маршрута похода</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_ROUTE_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Routes</p> <p>- столбец id_inst</p> <p>Назначение: индекс инструктора</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_INST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Instructors</p>
Таблица Instructors	Назначение: содержит информацию об инструкторах

	<p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Hiking - таблица Tourists <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_inst <p>Назначение: идентификатор инструктора</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_INST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Hiking</p> <ul style="list-style-type: none"> - столбец id_tourist <p>Назначение: идентификатор туриста</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: внешний ключ ID_TOURIST_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Tourists</p>
Таблица Category_hike	<p>Назначение: содержит информацию о категориях похода</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Hiking <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_cat_h <p>Назначение: идентификатор категории инструктора</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_CAT_H_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Hiking</p> <ul style="list-style-type: none"> - столбец plan_notpl

	<p>Назначение: плановый/не плановый поход</p> <p>Тип: varchar2(20)</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p>
Таблица Routes	<p>Назначение: содержит информацию о маршрутах</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Hiking - таблица Parking <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_route <p>Назначение: идентификатор маршрута</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_ROUTE_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Hiking, таблица Parking</p> - столбец start_parking <p>Назначение: содержит информацию о стартовой стоянке</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: not null</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p>
Таблица Parking	<p>Назначение: содержит информацию о стоянках</p> <p>Тип таблицы: heap</p> <p>Ограничения целостности уровня таблицы: отсутствуют</p> <p>Связанные объекты:</p> <ul style="list-style-type: none"> - таблица Routes <p>Атрибуты:</p> <ul style="list-style-type: none"> - столбец id_route <p>Назначение: идентификатор маршрута</p>

	<p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_ROUTE_PK, внешний ключ ID_ROUTE_FK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: таблица Routes</p> <p>- столбец id_park</p> <p>Назначение: идентификатор стоянки</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_PARK_PK</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p> <p>- столбец id_park_prev</p> <p>Назначение: идентификатор предыдущей стоянки (стоянки-родителя)</p> <p>Тип: int</p> <p>Ограничения целостности уровня столбца: первичный ключ ID_PARK_PREV_PK</p> <p>- столбец time_park</p> <p>Назначение: дата и время стоянки</p> <p>Тип: timestamp(0)</p> <p>Ограничения целостности уровня столбца: отсутствуют</p> <p>Индексация: отсутствует</p> <p>Связанные объекты: отсутствуют</p>
Процедура/функция add_history_tourists	<p>Назначение: добавление данных в таблицу History_tourists</p> <p>Тип: процедура</p> <p>Связанные объекты: - триггер update_history_tourists, таблица History_tourists</p> <p>Входные параметры: -p_id_hist_tourist -p_id_tourist -p_name -p_gender -p_max_complexity</p>

	<p>-p_id_cat_tourist -p_date_add</p> <p>Реализуемый алгоритм: INSERT INTO history_tourists (id_hist_tourist, id_tourist, name, gender, max_complexity, id_cat_tourist, date_add) VALUES (p_id_hist_tourist, p_id_tourist, p_name, p_gender, p_max_complexity, p_id_cat_tourist, p_date_add);</p>
<p>Триггер update_history_tourists</p>	<p>Назначение: внесение предыдущих данных о туристах в таблицу History_tourists</p> <p>Установлен на таблицу: Tourists</p> <p>Тип: after update of table Tourists</p> <p>Для каждой строки: да</p> <p>Связанные объекты: - таблица Tourists - процедура add_history_tourists - последовательность hist_sequence</p> <p>Реализуемый алгоритм: add_history_tourists(hist_sequence.nextval, :old.id_tourist, :old.id_name, :old.gender, :old.max_complexity, :old.id_cat_tourist, sysdate);</p>
<p>Последовательность hist_sequence</p>	<p>Назначение: добавление последовательных значений в таблицу History_tourists, столбец id_hist_tourist</p> <p>Связанные объекты: триггер update_history_tourists</p> <p>Реализуемый алгоритм: start with 1 increment by 1;</p>
<p>Джобс del_old_history</p>	<p>Назначение: удаление каждый месяц данных о туристах, добавленных в таблицу History_tourists 10 или более лет назад</p> <p>Установлен на таблицу: History_tourists</p> <p>Тип: PLSQL_BLOCK</p> <p>Интервал: FREQ=MONTHLY; BYMONTHDAY=1 (в последний день каждого месяца)</p> <p>Дата старта: 18-FEBRUARY-33 07.00.00 PM</p> <p>Связанные объекты:</p>

	<p>- таблица History_tourists</p> <p>Реализуемый алгоритм:</p> <p>DELETE FROM history_tourists WHERE (month_between(date_add, sysdate)/12)>=10;</p>
--	---

8 Отношения (связи)

Связь	Описание связи
LEAD_SECTION_FK	<p>Связываемые объекты: таблица leaders и таблица sections</p> <p>Атрибуты связывания: id_lead(leaders) - id_lead(sections)</p> <p>Тип: строго один к строго одному</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы leaders не входит в первичный ключ таблицы sections)</p>
SECTION_TRAINER_FK	<p>Связываемые объекты: таблица sections и таблица trainers</p> <p>Атрибуты связывания: id_section(sections) - id_section(trainers)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы sections не входит в первичный ключ таблицы trainers)</p>
SECTION_GROUP_FK	<p>Связываемые объекты: таблица sections и таблица groups</p> <p>Атрибуты связывания: id_section(sections) - id_section(groups)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы sections не входит в первичный ключ таблицы groups)</p>
TRAINER_GROUP_FK	<p>Связываемые объекты: таблица trainers и таблица groups</p> <p>Атрибуты связывания: id_trainer(trainers) - id_trainer(groups)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы trainers не входит в первичный ключ таблицы groups)</p>
GROUP_TOURIST_FK	<p>Связываемые объекты: таблица groups и таблица</p>

	<p>gr_tourists</p> <p>Атрибуты связывания: id_group(groups) - id_group(gr_tourists)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы groups не входит в первичный ключ таблицы gr_tourists)</p>
TOURIST_GROUP_FK	<p>Связываемые объекты: таблица gr_tourists и таблица tourists</p> <p>Атрибуты связывания: id_tourist(tourists) - id_tourist(gr_tourists)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы tourists не входит в первичный ключ таблицы gr_tourists)</p>
TOURIST_TRAINER_FK	<p>Связываемые объекты: таблица tourists и таблица trainers</p> <p>Атрибуты связывания: id_tourist(tourists) - id_tourist(trainers)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы tourists не входит в первичный ключ таблицы trainers)</p>
CATEGORY_TOURIST_FK	<p>Связываемые объекты: таблица cat_tourists и таблица tourists</p> <p>Атрибуты связывания: id_cat_tourist(cat_tourists) - id_cat_tourist(tourists)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы cat_tourists не входит в первичный ключ таблицы tourists)</p>
TOURIST_HISTORY_FK	<p>Связываемые объекты: таблица tourists и таблица history_tourists</p> <p>Атрибуты связывания: id_tourist(tourists) - id_tourist(history_tourists)</p> <p>Тип: строго один к нестрогим</p>

	Идентифицирующая связь: нет (так как первичный ключ таблицы tourists не входит в первичный ключ таблицы history_tourists)
GROUP_TRAINING_FK	Связываемые объекты: таблица groups и таблица training Атрибуты связывания: id_group(groups) – id_group(training) Тип: строго один к нестрого многим Идентифицирующая связь: нет (так как первичный ключ таблицы groups не входит в первичный ключ таблицы training)
TOURIST_COMPET_FK	Связываемые объекты: таблица tourists и таблица compet_tourists Атрибуты связывания: id_tourist(tourists) - id_tourist(compet_tourists) Тип: строго один к нестрого многим Идентифицирующая связь: нет (так как первичный ключ таблицы tourists не входит в первичный ключ таблицы compet_tourists)
COMPET_TOURIST_FK	Связываемые объекты: таблица competitions и таблица compet_tourists Атрибуты связывания: id_id_compet(competitions) - id_compe(compet_tourists) Тип: строго один к нестрого многим Идентифицирующая связь: нет (так как первичный ключ таблицы competitions не входит в первичный ключ таблицы compet_tourists)
TOURIST_INSTRUCTOR_FK	Связываемые объекты: таблица tourists и таблица instructors Атрибуты связывания: id_tourist(tourists) - id_tourist(instructors) Тип: строго один к строго одному Идентифицирующая связь: нет (так как первичный ключ таблицы tourists не входит в первичный ключ таблицы instructors)
TOURIST_HIKE_FK	Связываемые объекты: таблица tourists и таблица

	<p>hike_tourists</p> <p>Атрибуты связывания: id_tourist(tourists) - id_tourist(hike_tourists)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы tourists не входит в первичный ключ таблицы hike_tourists)</p>
HIKE_TOURIST_FK	<p>Связываемые объекты: таблица hiking и таблица hike_tourists</p> <p>Атрибуты связывания: id_hike(hiking) - id_hike(hike_tourists)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы hiking не входит в первичный ключ таблицы hike_tourists)</p>
INSTRUCTOR_HIKE_FK	<p>Связываемые объекты: таблица instructors и таблица hiking</p> <p>Атрибуты связывания: id_inst(instructors) - id_inst(hiking)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы instructors не входит в первичный ключ таблицы hiking)</p>
CATEGORY_HIKE_FK	<p>Связываемые объекты: таблица category_hike и таблица hiking</p> <p>Атрибуты связывания: id_cat_h(category_hike) - id_cat_h(hiking)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: нет (так как первичный ключ таблицы category_hike не входит в первичный ключ таблицы hiking)</p>
ROUTE_HIKE_FK	<p>Связываемые объекты: таблица routes и таблица hiking</p> <p>Атрибуты связывания: id_route(routes) - id_route(hiking)</p> <p>Тип: строго один к строго одному</p>

	<p>Идентифицирующая связь: нет (так как первичный ключ таблицы routes не входит в первичный ключ таблицы hiking)</p>
ROUTE_PARK_FK	<p>Связываемые объекты: таблица routes и таблица parking</p> <p>Атрибуты связывания: id_route(routes) - id_route(parking)</p> <p>Тип: строго один к нестрогим</p> <p>Идентифицирующая связь: да (так как первичный ключ таблицы routes входит в первичный ключ таблицы parking)</p>

9 SQL-код создания БД

```
create table Cat_tourists
(
id_cat_tourist int primary key,
category varchar2(20)
);
```

```
create table Leaders
(
id_lead int primary key,
name varchar2(60) not null,
date_of_birth date not null,
salary int,
hire_date date not null,
fired_date date
);
```

```
create table Sections
(
id_section int primary key,
specialization varchar2(60) not null,
id_lead int,
foreign key (id_lead) references Leaders(id_lead)
);
```

```
create table Trainers
(
id_trainer int primary key,
salary int,
id_tourist int,
foreign key (id_section) references Sections(id_section)
);
```

```
create table Groups
(
id_group int primary key,
number_gr int not null,
```

```

id_section int,
id_trainer int,
foreign key (id_section) references Sections(id_section),
foreign key (id_trainer) references Trainers(id_trainer)
);

create table Tourists
(
id_tourist int primary key,
name varchar2(60) not null,
gender varchar2(20) not null,
date_of_birth date not null,
max_complexity int default 0,
id_cat_tourist int,
foreign key (id_cat_tourist) references Cat_tourists(id_cat_tourist)
);

alter table Trainers
add foreign key (id_tourist) references Tourists(id_tourist);

create table Gr_tourist
(
id_tourist int,
id_group int,
foreign key (id_tourist) references Tourists(id_tourist),
foreign key (id_group) references Groups(id_group)
);

create table Training
(
id_tr int primary key,
date_tr date not null,
time_tr number(2,1) not null,
id_group int,
foreign key (id_group) references Groups(id_group)
);

create table Competitions

```

```
(  
id_compet int primary key,  
type_compet varchar2(60) not null,  
date_compet date not null  
);
```

```
create table Compet_tourists  
(  
id_tourist int,  
id_compet int,  
foreign key (id_tourist) references Tourists(id_tourist),  
foreign key (id_compet) references Competitions(id_compet)  
);
```

```
create table Instructors  
(  
id_inst int primary key,  
id_tourist int,  
foreign key (id_tourist) references Tourists(id_tourist)  
);
```

```
create table Category_hike  
(  
id_cat_h int primary key,  
plan_notpl varchar2(20) not null  
);
```

```
create table Routes  
(  
id_route int primary key,  
start_parking int not null  
);
```

```
create table Parking  
(  
id_route int,  
id_park int,  
id_park_prev int,
```

```

time_park timestamp(0),
foreign key (id_route) references Routes(id_route),
primary key (id_route, id_park, id_park_prev)
);

```

```

create table Hiking
(
id_hike int primary key,
date_hike date not null,
hike_days int not null,
place varchar2(60) not null,
complexity int not null,
id_cat_h int,
id_route int,
id_inst int,
foreign key (id_cat_h) references Category_hike(id_cat_h),
foreign key (id_route) references Routes(id_route),
foreign key (id_inst) references Instructors(id_inst)
);

```

```

create table Hike_tourists
(
id_tourist int,
id_hike int,
foreign key (id_tourist) references Tourists(id_tourist),
foreign key (id_hike) references Hiking(id_hike)
);

```

```

create table History_tourists
(
id_hist_tourist int primary key,
id_tourist int,
name varchar2(60),
gender varchar2(20),
max_complexity int,
id_group int,
id_cat_tourist int,
date_add date,

```

```
foreign key (id_tourist) references Tourists(id_tourist)
);
```

10 Триггеры, джобсы, процедуры, функции

Добавление данных в таблицу History_tourists:

```
create sequence hist_sequence  
start with 1  
increment by 1;
```

```
create or replace PROCEDURE add_history_tourists  
( p_id_hist_tourist  history_tourists.id_hist_tourist%type  
  , p_id_tourist      history_tourists.id_tourist%type  
  , p_name            history_tourists.name%type  
  , p_gender          history_tourists.gender%type  
  , p_max_complexity history_tourists.max_complexity%type  
  , p_id_cat_tourist  history_tourists.id_cat_tourist%type  
  , p_date_add        history_tourists.date_add%type  
)  
IS  
BEGIN  
    INSERT INTO history_tourists (id_hist_tourist, id_tourist, name,  
                                   gender, max_complexity, id_cat_tourist, date_add)  
    VALUES (p_id_hist_tourist, p_id_tourist, p_name, p_gender,  
            p_max_complexity, p_id_cat_tourist, p_date_add);  
END add_history_tourists;
```

```
create or replace TRIGGER update_history_tourists  
AFTER UPDATE OF name, gender, max_complexity, id_cat_tourist ON  
Tourists  
FOR EACH ROW  
BEGIN  
    add_history_tourists(hist_sequence.nextval, :old.id_tourist, :old.id_name, :old  
.gender, :old.max_complexity, :old.id_cat_tourist, sysdate);  
END;
```

Удаление данных из таблицы History_tourists:

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'del_old_history',
JOB_TYPE => 'PLSQL_BLOCK',
JOB_ACTION => 'DELETE FROM history_tourists WHERE
(month_between(date_add, sysdate)/12)>=10;',
START_DATE => '18-FEBRUARY-33 07.00.00 PM',
REPEAT_INTERVAL => 'FREQ=MONTHLY; BYMONTHDAY=1',
COMMENTS => 'удаляет данные истории, которым 10 лет или более',
ENABLED => TRUE);
END;
```

11 Ограничения на применение схемы

- 1) Невозможно узнать победителей в соревновании
- 2) Максимальную сложность похода, пройденного туристом приходится вводить вручную
- 3) Невозможно определить вид соревнований

12 Запросы к БД

1. Получить список и общее число туристов, занимающихся в клубе, в указанной секции, группе, по половому признаку, году рождения, возрасту.

```
select name, gender, date_of_birth, round((sysdate-date_of_birth)/365.25, 5)
as age from Tourists
join Gr_tourist on Tourists.id_tourist = Gr_tourist.id_tourist
join Groups on Gr_tourist.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
where number = 1 and specialization = 'водный спорт'
union
select 'Всего', 'на', sysdate, count(name) from (select name from Tourists
join Gr_tourist on Tourists.id_tourist = Gr_tourist.id_tourist
join Groups on Gr_tourist.id_group = Groups.id_group
```

```
join Sections on Groups.id_section = Sections.id_section
where number = 1 and specialization = 'водный спорт')
order by gender, date_of_birth, age
```

2. Получить список и общее число тренеров указанной секции, по всем секциям, по половому признаку, по возрасту, по размеру заработной платы, специализации.

```
select name, gender, round((sysdate-date_of_birth)/365.25, 5) as age,
specialization, salary from Tourists
join Trainers on Tourists.id_tourist = Trainers.id_tourist
join Sections on Trainers.id_section = Sections.id_section
where Sections.specialization = 'водный спорт'
union
select 'Тренеров', 'из', 1, 'секции', count(name) from (select name from
Tourists
join Trainers on Tourists.id_tourist = Trainers.id_tourist
join Sections on Trainers.id_section = Sections.id_section
where Sections.specialization = 'водный спорт')
select 'Тренеров', 'из всех >', 1, 'секций', count(name) from (select name
from Tourists
join Trainers on Tourists.id_tourist = Trainers.id_tourist
join Sections on Trainers.id_section = Sections.id_section)
order by gender, age, salary, specialization
```

3. Получить перечень и общее число соревнований, в которых участвовали спортсмены из указанной секции, по всем секциям.

```
select type_compet, id_compet from Competitions
join Compet_tourists on Competitions.id_compet =
Compet_tourists.id_compet
join Tourists on Compet_tourists.id_tourist = Tourists.id_tourist
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
where Sections.specialization = 'водный спорт'
union
```



```

select 'Всего:', count(id_compet) from (select id_compet from Competitions
join Compet_tourists on Competitions.id_compet =
Compet_tourists.id_compet
join Tourists on Compet_tourists.id_tourist = Tourists.id_tourist
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
where Sections.specialization = 'водный спорт')
union
select 'Всего:', count(id_compet)

```

4. Получить список тренеров, проводивших тренировки в указанной группе, за указанный период времени.

```

select name from Tourists
join Trainers on Tourists.id_tourist = Trainers.id_tourist
join Groups on Trainers.id_trainer = Groups.id_trainer
join Training on Groups.id_group = Training.id_group
where number_gr = 1 and to_date('01-02-2020', 'DD-MM-YYYY')<=date_tr
and date_tr<=to_date('01-03-2020', 'DD-MM-YYYY')

```

5. Получить список и общее число туристов из некоторой секции, группы, которые ходили в заданное количество походов, ходили в указанный поход, ходили в поход в обозначенное время, ходили по определенному маршруту, были в некоторой точке, имеют соответствующую категорию.

```

select name from Tourists
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
join Hiking on Hike_tourists.id_hike = Hiking.id_hike
join Routes on Hiking.id_route = Routes.id_route
join Parking on Routes.id_route = Parking.id_route
where specialization = 'водный спорт' and number = 1 and count(id_hike) =
3 and place = 'Эверест' and date_hike = to_date('27-09-2007', 'DD-MM-YYYY') and
id_route = 4 and id_park = 7 and max_complexity = complexity

```

6. Получить перечень руководителей секций полностью, по размеру заработной платы, по году рождения, возрасту, году поступления на работу.

```
select id_lead, name, salary, extract(year from date_of_birth) as year_of_birth,  
round((sysdate-date_of_birth)/365.25, 5) as age, extract(year from hire_date) as  
hire_year from Leaders  
order by salary, year_of_birth, age, hire_year
```

7. Получить нагрузку тренеров (вид занятий, количество часов), ее объем по определенным видам занятий и общую нагрузку за указанный период времени для данного тренера или указанной секции.

```
select id_trainer, sum(time_tr) from Training  
join Groups on Training.id_group = Groups.id_group  
group by id_trainer
```

```
select sum(time_tr) from Training  
join Groups on Training.id_group = Groups.id_group  
where id_trainer = 2 and to_date('01-02-2020', 'DD-MM-YYYY')<=date_tr  
and date_tr<=to_date('01-03-2020', 'DD-MM-YYYY')
```

8. Получить перечень и общее число маршрутов, по которым ходили туристы из указанной секции, в обозначенный период времени, по которым водил свои группы данный инструктор, по которым прошло указанное количество групп.

```
select id_route from Routes  
join Hiking on Routes.id_route = Hiking.id_route  
join Hike_tourists on Hiking.id_hike = Hike_tourists.id_hike  
join Tourists on Hike_tourists.id_tourist = Tourists.id_tourist  
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist  
join Groups on Gr_tourists.id_group = Groups.id_group  
join Sections on Groups.id_section = Sections.id_section  
where specialization = 'водный спорт' and to_date('01-02-2020', 'DD-MM-YYYY')<=date_hike and date_hike<=to_date('01-03-2020', 'DD-MM-YYYY') and  
id_inst = 5, count(id_group)=24
```

```

union
select count(id_route) from (select id_route from Routes
join Hiking on Routes.id_route = Hiking.id_route
join Hike_tourists on Hiking.id_hike = Hike_tourists.id_hike
join Tourists on Hike_tourists.id_tourist = Tourists.id_tourist
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
where specialization = 'водный спорт' and to_date('01-02-2020', 'DD-MM-
YYYY')<=date_hike and date_hike<=to_date('01-03-2020', 'DD-MM-YYYY') and
id_inst = 5, count(id_group)=24)

```

9. Получить перечень и общее число маршрутов, которые проходят через некоторую точку, имеют длину больше указанной, могут удовлетворять заданной категории сложности.

```

select id_route from Routes
join Parking on Routes.id_route = Parking.id_route
join Hiking on Routes.id_route = Hiking.id_route
where id_park = 1 and count(id_park)>6 and complexity >= 1
union
select count(id_route) from (select id_route from Routes
join Parking on Routes.id_route = Parking.id_route
join Hiking on Routes.id_route = Hiking.id_route
where id_park = 1 and count(id_park)>6 and complexity >= 1)

```

10. Получить перечень и общее число туристов из указанной секции, группы, которые могут ходить в указанные типы походов.

```

select name, id_tourists from Tourists
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
join Hiking on Hike_tourists.id_hike = Hiking.id_hike
where max_complexity >= complexity-1
union
select 'Bcero', count(id_tourists) from (select Tourists
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
join Hiking on Hike_tourists.id_hike = Hiking.id_hike

```

where max_complexity >= complexity-1)

11. Получить перечень и общее число инструкторов, инструкторов-спортсменов, инструкторов-тренеров, которые имеют определенную категорию, которые ходили в указанное количество походов, ходили в определенный поход, ходили по некоторому маршруту, были в указанной точке

```
with inst as (select id_inst, id_cat_tourist from Instructors
join Tourists on Instructors.id_tourist = Tourists.id_tourist
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
join Hiking on Hike_tourists.id_hike = Hiking.id_hike
join Routes on Hiking.id_route = Routes.id_route
join Parking on Routes.id_route = Parking.id_route
where max_complexity = 2 and count(id_hike) = 3 and place = 'Эверест' and
id_route = 4 and id_park = 7)
select * from inst
union
select 1, count(id_inst) from inst
union
select * from inst
where id_cat_tourist = 2
union
select 2, count(id_inst) from inst where id_cat_tourist = 2
union
select * from inst
where id_cat_tourist = 3
union
select 3, count(id_inst) from inst where id_cat_tourist = 3
```

12. Получить список туристов из указанной секции, группы, которые ходили в походы со своим тренером в качестве инструктора.

```
select name from Tourists
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
```

```

join Hiking on Hike_tourists.id_hike = Hiking.id_hike
join Instructors on Tourists.id_tourist = Instructors.id_tourist
join Trainers on Tourists.id_tourist = Trainers.id_tourist
where number = 1 and specialization = 'водный спорт' and Hiking.id_inst =
Instuctor.id_tourist = Trainer.id_torist

```

13. Получить список туристов из некоторой секции, группы, которые ходили по всем маршрутам, по указанным маршрутам.

```

select name from Tourists
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
join Hiking on Hike_tourists.id_hike = Hiking.id_hike
join Routes on Hiking.id_route = Routes.id_route
where number = 1 and specialization = 'водный спорт' and
count(id_route)=(select count(id_route) from Routes)
union
select name from Tourists
join Gr_tourists on Tourists.id_tourist = Gr_tourists.id_tourist
join Groups on Gr_tourists.id_group = Groups.id_group
join Sections on Groups.id_section = Sections.id_section
join Hike_tourists on Tourists.id_tourist = Hike_tourists.id_tourist
join Hiking on Hike_tourists.id_hike = Hiking.id_hike
join Routes on Hiking.id_route = Routes.id_route
where number = 1 and specialization = 'водный спорт' and id_route =
ALL(2, 13, 7)

```