Практика по базам данных

ОТЧЕТ

Лившиц Лев

244 группа

Предметная область: «Конференция»

Реализация: MS SQLServer

Содержание

[ОПИСАНИЕ СИСТЕМЫ 2](#_heading=h.30j0zll)

[Требования 2](#_heading=h.1fob9te)

[Модель данных 2](#_heading=h.3znysh7)

[Функциональность 3](#_heading=h.2et92p0)

[Серверная часть 3](#_heading=h.tyjcwt)

[Клиентская часть 4](#_heading=h.3dy6vkm)

[СКРИПТЫ 5](#_heading=h.1t3h5sf)

[Серверная часть 5](#_heading=h.4d34og8)

[Хранимые процедуры и функции 5](#_heading=h.2s8eyo1)

[Триггеры 6](#_heading=h.17dp8vu)

[Представления 6](#_heading=h.3rdcrjn)

[Клиентская часть 7](#_heading=h.26in1rg)

[ПРИЛОЖЕНИЕ: Создание и заполнение базы данных 10](#_heading=h.lnxbz9)

ОПИСАНИЕ СИСТЕМЫ

# Требования

Система предназначена для обслуживания международной научной конференции.

Помимо регистрации участников, система должна хранить данные обо всех пленарных и стендовых докладах, о выходных данных сборника докладов по результатам конференции, и желательно - о культурной программе, организованной для участников.

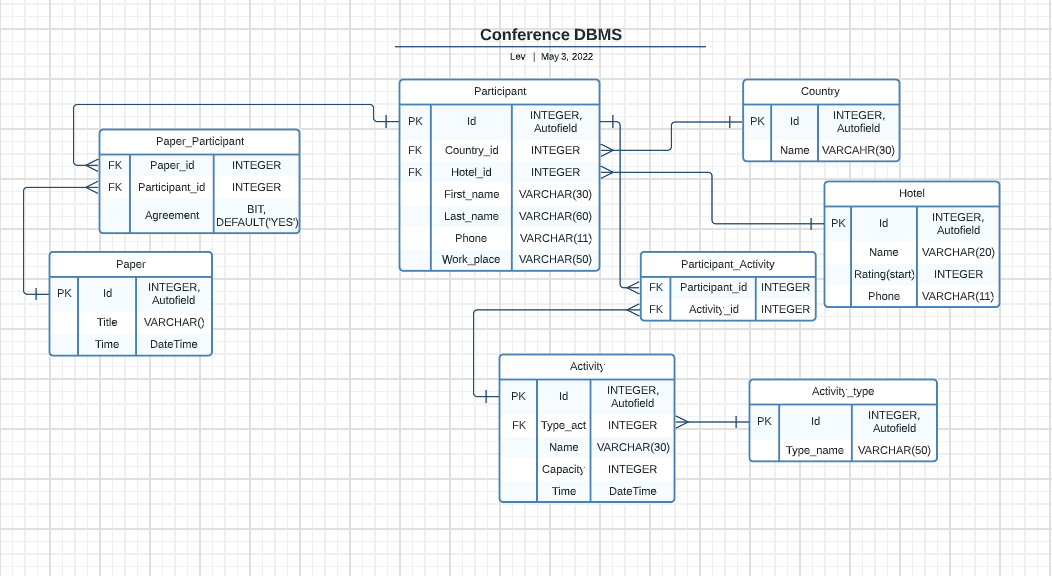
При регистрации участника сохраняются только самые основные его персональные и контактные данные, а также место работы. Обязательно следует вести реестр стран-участниц конференции.

Приезжие участники конференции будут проживать в двух отелях разных категорий, в системе полезно иметь информацию о номерах их комнат.

Информация о докладе должна включать данные о его типе, о названии, о времени и о секции, в которой он будет проходить.

Нужно иметь в виду, что при формировании сборника конференции в него вносятся только те доклады, авторы которых изъявили желание быть включенными.

# Модель данных



Функциональность

# Серверная часть

:

| ***Хранимые процедуры\функции*** | *Реализация* | *Комментарии* |
| --- | --- | --- |
| Регистрация нового участника |  | (фамилия, имя, телефон, …) |
| Добавление нового участника с докладом | AddNewParticipantWithPaper() | (фамилия, имя, телефон, …, наименование доклада) |
| Добавление нового типа активности |  |  |
| Добавление новой активности |  |  |
| Добавление отеля |  |  |
| Добавление страны |  |  |
| Удаление участника | DeleteParticipantWithAllEvents() | (Фамилия, имя), удаление происходит только ели участник с такими ФИ единственный |
| Сбор информации о загруженности отелей | GetNumberInHotel() | Возвращает таблицу (отель, количество постояльцев) |
| Проверка того, что все участники доклада дали согласие на его публикацию | PossibleToPublish() | (id доклада), возвращает: 1 - если все согласны, 0 - если хотя бы один нет |

| ***Триггеры*** | *Реализация* | *Комментарии* |
| --- | --- | --- |
| Запрет на удаление типа развлечения | TriggerNoDeleteActivity |  |
| Запрет на удаление связи доклада с единственным спикером | TriggerNoDeleteSpeaker | Иначе доклад может остаться бе спикера :) |
| Запрет на запись в группу, где уже набралось максимальное количество участников | TriggerNoOverBook | Чтобы в группу развлечения не могло записаться больше людей, чем указано. Остальные получают ошибку. |
| Контроль повторного добавления участника |  | ID формируется с помощью IDENTITY так что двух одинаковых строк получиться не может. |

| ***Представления*** | *Реализация* | *Комментарии* |
| --- | --- | --- |
| Имя и фамилия участника, из какой страны | v\_Participant\_Country | Все участники и страны из которых они приехали (Фамилия, Имя, страна) |
| Участники конференции |  | (фамилия,имя клиента, телефон, отель, страна, место работы) |
| Докладчик с докладом | v\_Participant\_Paper | (Фамилия (спикера), Имя (спикера), название доклада, время) |
| Реестр развлечений по видам |  | (вид, название) |
| Участники и развлечения | v\_Participant\_Activity | (Имя, фамилия, Название activity которое он выбрал, время) |
| Список всех докладов |  |  |
| Все доклады, которые войдут в сборник | v\_works | (id, название, время) |
| Список всех стран |  | (id, название страны) |
| Список всех отелей |  | (id, название отеля) |

# Клиентская часть

| *Экранные формы основные* | *дополнительные* | *Реализация*  *(запрос)* | *Что здесь можно использовать из серверной части* |
| --- | --- | --- | --- |
| **Реестр участников** |  | (01) Фамилия, имя, место работы | v\_Participant\_Country |
| Новый участник |  | AddNewParticipantWithPaper() |
| Изменить данные участника |  |  |
| Удалить участника |  | DeleteParticipantWithAllEvents()  TriggerNoDeleteSpeaker() |
| Добавить Участника |  |  |
| Фильтры | (03) Информация об участниках с одного места работы  (06) Информация об участниках докладов, запланированных на 23 мая 2022  (07) Информация о докаладах и их спикерах |  |
| **Список отелей** | Изменить данные |  |  |
| Добавить отель |  |  |
| Удалить отель |  |  |
| Фильтры | (02) Участники из России с информации об отеле.  (10) Информация о количесве поселенных по отелям участников из росcии. | FUNCTION GetNumberInHotel() |
| **Список типов мероприятий** | Добавить |  |  |
| Изменить |  |  |
| Удалить |  |  |
| Фильтры |  |  |
| **Реестр запланированных метроприятий** |  |  | v\_Participant\_Activity |
| Добавить |  |  |
| Изменить |  |  |
| Удалить |  |  |
| Фильтры | (09) ФИ участников, которые не записались ни на одну развлекательную программу. |  |
| **Список докладов** |  |  | v\_Participant\_Paper |
| Добавить |  |  |
| Изменить |  |  |
| Удалить |  |  |
| Фильтры | (08) Информация о докладах, которые войдут в сборник и их спикерах.  (11) Информация обо всех участниках, не являющимися спикерами ни в одном из докладов. | v\_works |
| **Список стран** | Добавить |  |  |
| Изменить |  |  |
| Удалить |  |  |
| Фильтры | (04) Информация о количестве участников из разных стран  (05) Страна, из которой больше всего спикеров |  |
| **Служебные запросы** |  | (12) Информация о том сколько человек из отеля записалось на каждую активность. |  |

СКРИПТЫ

# Серверная часть

## Хранимые процедуры и функции

-- 1. Добавление участника вместе с докладом, если доклада с таким названием еще не было зарегистрировано,

-- иначе - добавление этого участника к списку докладчиков.

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE AddNewParticipantWithPaper

@var\_Country\_id AS INTEGER,

@var\_Hotel\_id AS INTEGER,

@var\_First\_name AS VARCHAR(30),

@var\_Last\_name AS VARCHAR(60),

@var\_Phone AS VARCHAR(11),

@var\_Work\_place AS VARCHAR(50),

@var\_Title AS VARCHAR(250)

AS

BEGIN

DECLARE @var\_paper\_id INTEGER;

DECLARE @var\_participant\_id INTEGER;

INSERT INTO Participant(Country\_id, Hotel\_id, First\_name, Last\_name, Phone, Work\_place)

VALUES (@var\_Country\_id, @var\_Hotel\_id, @var\_First\_name, @var\_Last\_name, @var\_Phone, @var\_Work\_place);

IF ((SELECT count(\*) FROM Paper WHERE Title = @var\_Title) = 0)

BEGIN

INSERT INTO Paper(Title, Time) VALUES (@var\_Title, NULL);

END ;

SELECT @var\_paper\_id = Id FROM Paper WHERE Title = @var\_Title;

SELECT @var\_participant\_id = Max(Id) FROM Participant;

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (@var\_paper\_id, @var\_participant\_id);

END

;

--------------

-- Пример вызова.

EXEC AddNewParticipantWithPaper @var\_Country\_id=1, @var\_Hotel\_id = 3, @var\_First\_name = 'Ivan',

@var\_Last\_name = 'Moskalenko', @var\_Phone = '89123429134', @var\_Work\_place = 'Nothing', @var\_Title = 'Vk Parser';

--------------

-- Удаление процедуры.

DROP PROCEDURE AddNewParticipantWithPaper;

-- 2. Удаление участника по имени и фамилии (если такой один) вместе со всей информацией о докладах и активностях

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE PROCEDURE DeleteParticipantWithAllEvents

@var\_First\_name AS VARCHAR(30),

@var\_Last\_name AS VARCHAR(60)

AS

BEGIN

IF ((SELECT COUNT(\*) FROM Participant WHERE First\_name = @var\_First\_name AND Last\_name = @var\_Last\_name) = 1)

BEGIN

DECLARE @var\_Participant\_id INTEGER;

SELECT @var\_Participant\_id = Id FROM Participant WHERE First\_name = @var\_First\_name AND Last\_name = @var\_Last\_name;

DELETE FROM Paper\_Participant WHERE Participant\_id = @var\_Participant\_id;

DELETE FROM Participant\_Activity WHERE Participant\_id = @var\_Participant\_id;

DELETE FROM Participant WHERE Id = @var\_Participant\_id;

END;

END;

-- Пример вызова

EXEC DeleteParticipantWithAllEvents @var\_First\_name = 'Ivan', @var\_Last\_name = 'Moskalenko';

-- Удаление процедуры

DROP PROCEDURE DeleteParticipantWithAllEvents;

-- 3. Таблица с информацией о заргуженности отелей

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE FUNCTION GetNumberInHotel()

RETURNS @var\_Result TABLE(

Hotel\_id INTEGER,

Number\_of\_guests INTEGER

)

AS

BEGIN

INSERT @var\_Result(Hotel\_id, Number\_of\_guests)

SELECT Hotel\_id, COUNT(\*) AS Number\_of\_guests

FROM Participant

GROUP BY Hotel\_id;

RETURN

END;

-- Пример вызова

SELECT \* FROM dbo.GetNumberInHotel();

-- Удаление функции

DROP FUNCTION GetNumberInHotel;

-- 4. Проверка, можно ли публиковать доклад

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE FUNCTION PossibleToPublish(

@var\_Paper\_id INTEGER)

RETURNS BIT

AS

BEGIN

DECLARE @var\_result BIT;

IF

1 = ALL

(

SELECT Agreement FROM Paper\_Participant WHERE Paper\_Participant.Paper\_id = @var\_Paper\_id

)

SET @var\_result = 1;

ELSE

SET @var\_result = 0;

RETURN @var\_result

END;

-- Пример вызова

SELECT dbo.PossibleToPublish(1);

-- Удаление функции

DROP FUNCTION PossibleToPublish;

## Триггеры

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 1. Запрет на удаление типа развлечения

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TriggerNoDeleteActivity ON Activity\_type FOR DELETE

AS

ROLLBACK;

--- Пример вызова

DELETE FROM Activity\_type WHERE Id = 4;

--- Удалить триггер

DROP TRIGGER TriggerNoDeleteActivity;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 2. Запрет на удаление связи доклада с единственным спикером

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TriggerNoDeleteSpeaker ON Paper\_Participant INSTEAD OF DELETE

AS

BEGIN

DECLARE @var\_paper\_id INTEGER;

DECLARE @var\_participant\_id INTEGER;

DECLARE @var\_number\_speakers INTEGER;

SELECT @var\_paper\_id = Paper\_id FROM DELETED;

SELECT @var\_participant\_id = Participant\_id FROM DELETED;

IF ((SELECT COUNT(\*) FROM Paper\_Participant

WHERE Paper\_id = @var\_paper\_id

AND Is\_team\_leader = 1

AND Participant\_id <> @var\_participant\_id) > 0)

DELETE FROM Paper\_Participant WHERE Paper\_id = @var\_paper\_id AND Participant\_id = @var\_participant\_id;

ELSE

THROW 51001, 'You can not delete the only speaker', 1;

END;

--- Пример вызова

DELETE FROM Paper\_Participant WHERE Paper\_id = 1 AND Participant\_id = 2;

--- Удалить триггер

DROP TRIGGER TriggerNoDeleteSpeaker;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 3. Запрет на запись в группу, где уже набралось максимальное количество участников

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TRIGGER TriggerNoOverBook ON Participant\_Activity INSTEAD OF INSERT

AS

BEGIN

DECLARE @var\_Activity\_id INTEGER;

DECLARE @var\_Participant\_id INTEGER;

DECLARE @var\_number\_people\_in\_group INTEGER;

SELECT @var\_Participant\_id = Participant\_id FROM INSERTED;

SELECT @var\_Activity\_id = Activity\_id FROM INSERTED;

SELECT @var\_number\_people\_in\_group = COUNT(\*) FROM Participant\_Activity WHERE Activity\_id = @var\_Activity\_id;

IF ((SELECT Capacity FROM Activity WHERE Id = 2) > @var\_number\_people\_in\_group)

INSERT INTO Participant\_Activity(Participant\_id, Activity\_id) VALUES (@var\_Participant\_id, @var\_Activity\_id);

ELSE

THROW 51000, 'The group is full, choose another one, please', 1;

END;

--- Пример вызова

INSERT INTO Participant\_Activity VALUES (3, 2);

--- Удалить триггер

DROP TRIGGER TriggerNoOverBook;

## Представления

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 1. Имя и фамилия участника, из какой страны

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW v\_Participant\_Country

AS

SELECT First\_name, Last\_name, Name

FROM Participant

JOIN Country ON Participant.Country\_id = Country.Id;

-- Пример вызова

SELECT \* FROM v\_Participant\_Country;

-- Удаление

DROP VIEW v\_Participant\_Country;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 2. Имя и фамилия участника, название доклада и его время

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW v\_Participant\_Paper

AS

SELECT First\_name, Last\_name, Title, Time

FROM Participant

JOIN Paper\_Participant ON Participant.Id = Paper\_Participant.Participant\_id

JOIN Paper ON Paper\_Participant.Paper\_id = Paper.Id;

-- Пример вызова

SELECT \* FROM v\_Participant\_Paper;

-- Удаление

DROP VIEW v\_Participant\_Paper;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 3. Имя и фамилия участника, Название activity которое он выбрал и его время

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW v\_Participant\_Activity

AS

SELECT First\_name, Last\_name, Activity\_type.Type\_name, Time

FROM Participant

JOIN Participant\_Activity ON Participant.Id = Participant\_Activity.Participant\_Id

JOIN Activity ON Participant\_Activity.Activity\_id = Activity.Id

JOIN Activity\_type ON Activity.Type\_act = Activity\_type.Id;

-- Пример вызова

SELECT \* FROM v\_Participant\_Activity;

-- Удаление

DROP VIEW v\_Participant\_Activity;

-----------------------------------------------------------------------------------------------------------------------------------------------------

-- 4. Все доклады, которые войдут в сборник

-----------------------------------------------------------------------------------------------------------------------------------------------------

CREATE VIEW v\_works

AS

SELECT \* FROM Paper

WHERE (SELECT dbo.PossibleToPublish(Id)) = 1;

-- Пример вызова

SELECT \* FROM v\_works;

-- Удаление

DROP VIEW v\_works;

# Клиентская часть

(запросы для экранных форм и отчетов)

-- 1 Список участников

SELECT First\_name, Last\_name, Work\_place

FROM Participant

;

-- 2 Участники из России с информации об отеле

SELECT First\_name, Last\_name, Participant.Phone

FROM Participant

JOIN Hotel ON Participant.Hotel\_id = Hotel.Id

JOIN Country ON Participant.Country\_id = Country.Id

WHERE Country.Name = 'Russia'

;

-- 3 Информация об участниках с одного места работы

SELECT Work\_place, COUNT(\*) AS Number\_of\_workers

FROM Participant

GROUP BY Work\_place

;

-- 4 Информация о количестве участников из разных стран

SELECT Name, COUNT(\*) AS Number\_of\_participants

FROM Participant

JOIN Country ON participant.Country\_id = Country.Id

GROUP BY Name

;

-- 5 Страна, из которой больше всего спикеров

SELECT TOP 1 Country.Name, COUNT(\*) AS Number\_of\_participants

FROM participant

JOIN Country ON Participant.Country\_id = Country.Id

GROUP BY Country.Name

ORDER BY 2 DESC

;

-- 6 Информация об участниках докладов, запланированных на 23 мая 2022

SELECT First\_name, Last\_name, Title

FROM Participant

JOIN Paper\_Participant ON Participant.Id = Paper\_Participant.Participant\_id

JOIN Paper ON Paper\_Participant.Paper\_id = Paper.Id

WHERE Day(Time) = 23

;

-- 7 Информация о докаладах и их спикерах

SELECT First\_name, Last\_name, Title

FROM Paper

JOIN Paper\_Participant ON Paper.Id = Paper\_Participant.Paper\_id

JOIN Participant ON Paper\_Participant.Participant\_id = Participant.Id

WHERE Is\_team\_leader = 1

;

-- 8 Информация о докладах, которые войдут в сборник и их спикерах

SELECT First\_name, Last\_name, Title

FROM v\_works

JOIN Paper\_Participant ON v\_works.Id = Paper\_Participant.Paper\_id

JOIN Participant ON Paper\_Participant.Participant\_id = Participant.Id

WHERE Is\_team\_leader = 1

;

-- 9 ФИ участников, которые не записались ни на одну развлекательную программу

SELECT First\_name, Last\_name

FROM Participant

WHERE Id NOT IN

(SELECT Participant\_id FROM Participant\_Activity)

;

-- 10 Информация о количесве поселенных по отелям участников из Росcии

SELECT Hotel.Name, COUNT(\*) AS Number\_of\_people

FROM Hotel

JOIN Participant ON Participant.Hotel\_id = Hotel.Id

JOIN Country ON Participant.Country\_id = Country.Id

WHERE Country.Name = 'Russia'

GROUP BY Hotel.Name

ORDER BY COUNT(\*) ASC

;

-- 11 Информация обо всех участниках, не являющимися спикерами ни в одном из докладов

SELECT First\_name, Last\_name

FROM Participant

WHERE Participant.Id IN

(SELECT DISTINCT Participant\_id

FROM Paper\_Participant

WHERE Is\_team\_leader = 1)

;

-- 12 Информация о том сколько человек из отеля записалось на каждую активность

SELECT Hotel.Name, Activity\_type.Type\_name, COUNT(\*) AS Number\_of\_people

FROM Participant

JOIN Hotel ON Participant.Hotel\_id = Hotel.Id

JOIN Participant\_Activity ON Participant.Id = Participant\_Activity.Participant\_id

JOIN Activity ON Participant\_Activity.Activity\_id = Activity.Id

JOIN Activity\_type ON Activity.Type\_act = Activity\_type.Id

GROUP BY Hotel.Name, Activity\_type.Type\_name

;

-- 13 Информация о тех активностях, для которых запланирована группа.

SELECT \*

FROM Activity\_type

LEFT JOIN Activity ON Activity\_type.Id = Activity.Type\_act

;

-- 14 Информация о докладах и активностях

SELECT Id, Title

FROM Paper

UNION

SELECT Id, Name

FROM Activity

;

-- 15 Список стран, группа участников из которых заселилась в отели, набрав суммарно больше 25 звезд

SELECT Country.Name, SUM(Hotel.Rating) AS Hotel\_total\_rating FROM Participant

JOIN Country ON Participant.Country\_id = Country.Id

JOIN Hotel On Participant.Hotel\_id = Hotel.Id

GROUP BY (Country.Name)

HAVING SUM(Hotel.Rating) > 25

ORDER BY 2 DESC

;

# ПРИЛОЖЕНИЕ: Создание и заполнение базы данных

CREATE DATABASE conference;

GO

USE conference;

## Создание таблиц и PK

CREATE TABLE Participant(

Id INTEGER IDENTITY(1, 1) PRIMARY KEY,

Country\_id INTEGER NOT NULL,

Hotel\_id INTEGER NOT NULL,

First\_name VARCHAR(30) NOT NULL,

Last\_name VARCHAR(60) NOT NULL,

Phone VARCHAR (11),

Work\_place VARCHAR(50)

)

;

CREATE TABLE Hotel(

Id INTEGER IDENTITY(1, 1) PRIMARY KEY,

Name VARCHAR(20),

Rating INTEGER CHECK

(Rating >= 0 AND Rating <= 5),

Phone VARCHAR(11)

)

;

CREATE TABLE Country(

Id INTEGER IDENTITY(1, 1) PRIMARY KEY,

Name VARCHAR(30)

)

;

CREATE TABLE Paper(

Id INTEGER IDENTITY(1, 1) PRIMARY KEY,

Title VARCHAR(250) NOT NULL,

Time DateTime

)

;

CREATE TABLE Paper\_Participant(

Paper\_id INTEGER NOT NULL,

Participant\_id INTEGER NOT NULL,

Agreement BIT DEFAULT 1 NOT NULL,

Is\_team\_leader BIT DEFAULT 0 NOT NULL,

CONSTRAINT UPP UNIQUE (Paper\_id, Participant\_id)

)

;

CREATE TABLE Activity(

Id INTEGER IDENTITY(1, 1) PRIMARY KEY,

Type\_act INTEGER NOT NULL,

Name VARCHAR(30) NOT NULL,

Capacity INTEGER,

Time DateTime NOT NULL,

)

;

CREATE TABLE Participant\_Activity(

Participant\_id INTEGER NOT NULL,

Activity\_id INTEGER NOT NULL,

CONSTRAINT UPA UNIQUE (Participant\_id, Activity\_id)

)

;

CREATE TABLE Activity\_type(

Id INTEGER IDENTITY (1, 1) PRIMARY KEY,

Type\_name VARCHAR(50) NOT NULL

)

;

## Создание FK

ALTER TABLE Participant ADD CONSTRAINT FK\_Participant\_Country

FOREIGN KEY (Country\_id)

REFERENCES Country(Id)

;

ALTER TABLE Participant ADD CONSTRAINT FK\_Patricipant\_Hotel

FOREIGN KEY (Hotel\_id)

REFERENCES Hotel(Id)

;

ALTER TABLE Participant\_Activity ADD CONSTRAINT FK\_Participant\_Activity\_Participant

FOREIGN KEY (Participant\_id)

REFERENCES Participant(Id)

;

ALTER TABLE Participant\_Activity ADD CONSTRAINT FK\_Participant\_Participant\_Activity

FOREIGN KEY (Activity\_id)

REFERENCES Activity(Id)

;

ALTER TABLE Activity ADD CONSTRAINT FK\_Activity\_Activity\_type

FOREIGN KEY (Type\_act)

REFERENCES Activity\_type(Id)

;

ALTER TABLE Paper\_Participant ADD CONSTRAINT FK\_Participant\_Paper\_participant

FOREIGN KEY (Participant\_id)

REFERENCES Participant(Id)

;

ALTER TABLE Paper\_Participant ADD CONSTRAINT FK\_Paper\_Paper\_Participant

FOREIGN KEY (Paper\_id)

REFERENCES Paper(Id)

;

## Заполнение таблиц тестовыми данными

INSERT INTO Hotel VALUES ('Gazala Place', 4, 88001239999);

INSERT INTO Hotel VALUES('Plaza place', 5, 88009996666);

INSERT INTO Hotel VALUES('Home place', 4, 88005553535);

INSERT INTO Country VALUES('England');

INSERT INTO Country VALUES('Russia');

INSERT INTO Country VALUES('Germany');

INSERT INTO Country VALUES('Spain');

INSERT INTO Participant VALUES(1, 1, 'Oliver', 'Hughes', '45421234567', 'University of London');

INSERT INTO Participant VALUES(1, 3, 'Ovivia', 'Post', '45421234564', 'University of Manchester');

INSERT INTO Participant VALUES(1, 2, 'John', 'Inders', '45421233456', 'University of Manchester');

INSERT INTO Participant VALUES(1, 2, 'Harry', 'Potter', '45421234425', 'University of Manchester');

INSERT INTO Participant VALUES(1, 2, 'Tomas', 'Dep', '45421954567', 'University of London');

INSERT INTO Participant VALUES(2, 1, 'Ivan', 'Ivanov', '89121234567', '2nd RAS office');

INSERT INTO Participant VALUES(2, 2, 'Makar', 'Fedotov', '89121212546', '1st RAS office');

INSERT INTO Participant VALUES(2, 3, 'Victor', 'Petrov', '89412345678', 'University of Tomsk');

INSERT INTO Participant VALUES(2, 1, 'Petr', 'Markov', '89421234567', 'University of Omsk');

INSERT INTO Participant VALUES(2, 2, 'Vladimir', 'Krasnov', '89412345678', 'University of Moskow');

INSERT INTO Participant VALUES(2, 2, 'Iliya', 'Kostenkov', '89321235675', 'University of Moskow');

INSERT INTO Participant VALUES(2, 3, 'kirill', 'Dobronravov', '89425143462', 'University of Moskow');

INSERT INTO Participant VALUES(2, 3, 'Maxim', 'Kirillenko', '89124123526', 'University of Moskow');

INSERT INTO Participant VALUES(2, 3, 'Pavel', 'Maximov', '89346253322', 'University of Moskow');

INSERT INTO Participant VALUES(3, 2, 'Achim', 'Erdmann', '22345623444', 'University of Hamburg');

INSERT INTO Participant VALUES(3, 3, 'Adolf', 'Erich', '22345425754', 'University of Hamburg');

INSERT INTO Participant VALUES(3, 1, 'Alfons', 'Eugon', '21234625432', 'University of Hamburg');

INSERT INTO Participant VALUES(3, 1, 'Benno', 'Gerland', '26534553543', 'University of Berlin');

INSERT INTO Participant VALUES(3, 1, 'Carl', 'Gero', '22345452524', 'University of Berlin');

INSERT INTO Participant VALUES(3, 2, 'Claus', 'Gerulf', '22457894565', 'University of Hamburg');

INSERT INTO Participant VALUES(4, 1, 'Perez', 'Lopez', '33564625354', 'University of Barcelona');

INSERT INTO Participant VALUES(4, 1, 'Molina', 'Jove', '32345113433', 'University of Barcelona');

INSERT INTO Participant VALUES(4, 2, 'Tomas', 'Pons', '31353434535', 'University of Barcelona');

INSERT INTO Participant VALUES(4, 3, 'Vila', 'Gomez', '33345656734', 'University of Barcelona');

INSERT INTO Participant VALUES(4, 3, 'Ruiz', 'Moreno', '35676534325', 'University of Barcelona');

INSERT INTO Participant VALUES(4, 2, 'Ortega', 'Soler', '38523698521', 'University of Barcelona');

INSERT INTO Paper VALUES ('Computing Machinery and Intelligence', '2022-05-23T14:25:00');

INSERT INTO Paper VALUES ('A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence', '2022-05-23T15:25:00');

INSERT INTO Paper VALUES ('Fuzzy sets', '2022-05-23T16:25:00');

INSERT INTO Paper VALUES ('Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference', '2022-05-23T09:15:00');

INSERT INTO Paper VALUES ('Artificial Intelligence: A Modern Approach', '2022-05-23T12:45:00');

INSERT INTO Paper VALUES ('Language identification in the limit', '2022-05-24T09:15:00');

INSERT INTO Paper VALUES ('On the uniform convergence of relative frequencies of events to their probabilities', '2022-05-23T11:25:00');

INSERT INTO Paper VALUES ('A theory of the learnable', '2022-05-23T15:25:00');

INSERT INTO Paper VALUES ('Learning representations by back-propagating errors', '2022-05-23T16:25:00');

INSERT INTO Paper VALUES ('Induction of Decision Trees', '2022-05-23T18:00:00');

INSERT INTO Activity\_type VALUES ('Horse riding');

INSERT INTO Activity\_type VALUES ('Paragliding');

INSERT INTO Activity\_type VALUES ('jet skis');

INSERT INTO Activity VALUES (1, 'Park riding', 10, '2022-05-23T10:00:00');

INSERT INTO Activity VALUES (1, 'Sport riding', 5, '2022-05-23T12:00:00');

INSERT INTO Activity VALUES (1, 'forest riding', 15, '2022-05-23T14:00:00');

INSERT INTO Activity VALUES (2, 'loundge gliding', 15, '2022-05-23T18:00:00');

INSERT INTO Activity VALUES (2, 'high gliding excursion', 20, '2022-05-24T12:00:00');

INSERT INTO Activity VALUES (3, 'skis with sharks', 5, '2022-05-23T12:00:00');

INSERT INTO Activity VALUES (3, 'skis on the shutter island', 25, '2022-05-24T12:00:00');

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Agreement) VALUES (1, 1, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Is\_team\_leader) VALUES (1, 2, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Agreement) VALUES (1, 3, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Agreement) VALUES (1, 4, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Agreement) VALUES (1, 5, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Is\_team\_leader) VALUES (2, 6, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (2, 7);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (2, 8);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (2, 9);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (2, 10);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (3, 7);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (3, 8);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (2, 13);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (3, 11);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (3, 12);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (3, 13);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Is\_team\_leader) VALUES (3, 14, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Is\_team\_leader) VALUES (4, 15, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (4, 16);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (4, 17);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (4, 18);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (4, 19);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (4, 20);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Is\_team\_leader) VALUES (5, 21, 1);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (5, 22);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (5, 23);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (6, 23);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (6, 24);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id) VALUES (6, 25);

INSERT INTO Paper\_Participant(Paper\_id, Participant\_id, Is\_team\_leader) VALUES (6, 26, 1);

INSERT INTO Participant\_Activity VALUES (1, 1);

INSERT INTO Participant\_Activity VALUES (2, 2);

INSERT INTO Participant\_Activity VALUES (4, 3);

INSERT INTO Participant\_Activity VALUES (3, 4);

INSERT INTO Participant\_Activity VALUES (7, 5);

INSERT INTO Participant\_Activity VALUES (6, 6);

INSERT INTO Participant\_Activity VALUES (5, 7);

INSERT INTO Participant\_Activity VALUES (11, 1);

INSERT INTO Participant\_Activity VALUES (14, 4);

INSERT INTO Participant\_Activity VALUES (22, 7);

INSERT INTO Participant\_Activity VALUES (24, 2);

INSERT INTO Participant\_Activity VALUES (21, 1);

INSERT INTO Participant\_Activity VALUES (12, 1);

## Создание индексов

CREATE INDEX idx\_Activity\_Time ON Activity(Time);

CREATE INDEX idx\_Hotel\_Name ON Hotel(Rating);

CREATE UNIQUE INDEX idx\_Participant\_full\_name ON Participant(First\_name, Last\_name);

## Удаление таблиц

DROP TABLE Participant\_Activity;

DROP TABLE Activity;

DROP TABLE Activity\_type;

DROP TABLE Paper\_Participant;

DROP TABLE Paper;

DROP TABLE Participant;

DROP TABLE Country;

DROP TABLE Hotel;