|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |
| --- |
| **по лабораторной работе №10** |

**Название:**

Spark, Scala

**Дисциплина:** Языка программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | Д.Д. Капитонов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Вариант:**

Спроектировать системы записи в библиотеку в соответствии с заданием.

**Код программы:**

*-- http://sqlfiddle.com/ SQLite (SQL js)*

*PRAGMA foreign\_keys = ON;*

*CREATE TABLE Department (*

*ID Integer PRIMARY KEY,*

*Name varchar(100)*

*);*

*CREATE TABLE BookingAllowance (*

*ID Integer PRIMARY KEY,*

*DepartmentID Integer,*

*FromTime datetime NOT NULL CHECK (FromTime < ToTime AND DATE(FromTime) == DATE(ToTime)),*

*ToTime datetime NOT NULL CHECK (ROUND((JulianDay(ToTime) - JulianDay(FromTime))\*24\*60)%MinutesPerPerson =0),*

*MinutesPerPerson Integer NOT NULL CHECK (MinutesPerPerson > 0),*

*SimultaneousPersons Integer NOT NULL CHECK (SimultaneousPersons > 0),*

*FOREIGN KEY (DepartmentID) REFERENCES Department(ID)*

*);*

*CREATE TABLE Reader(*

*ID Integer PRIMARY KEY,*

*Name varchar(200) NOT NULL);*

*CREATE TABLE Booking(*

*ID Integer PRIMARY KEY,*

*AllowanceID Integer,*

*ReaderID Integer,*

*BookingTime datetime NOT NULL,*

*IsCancelled boolean NOT NULL,*

*FOREIGN KEY (AllowanceID) REFERENCES BookingAllowance(ID),*

*FOREIGN KEY (ReaderID) REFERENCES Reader(ID)*

*);*

*CREATE TRIGGER Check\_zap BEFORE INSERT*

*ON Booking*

*BEGIN*

*SELECT CASE*

*WHEN ((SELECT FromTime FROM BookingAllowance WHERE (BookingAllowance.FromTime <= NEW.BookingTime)AND*

*( (JulianDay(BookingAllowance.ToTime) - JulianDay(NEW.BookingTime))\*24\*60 >= BookingAllowance.MinutesPerPerson ) AND (BookingAllowance.ID = NEW.AllowanceID) ) ISNULL)*

*THEN RAISE(ABORT, 'Проверка интервала времени на то, что запись можно совершить на какое-то время с учётом конца')*

*END;*

*END;*

*CREATE TRIGGER Check\_time\_and\_count BEFORE INSERT*

*ON Booking*

*BEGIN*

*SELECT CASE*

*WHEN ((SELECT SimultaneousPersons FROM BookingAllowance WHERE BookingAllowance.ID = NEW.AllowanceID) <*

*(SELECT count(ID) FROM Booking WHERE Booking.AllowanceID = NEW.AllowanceID AND Booking.BookingTime = NEW.BookingTime)+1)*

*THEN RAISE(ABORT, 'Проверка возможноси создания записи по кол-во уже записанных человек на это время')*

*END;*

*END;*

*CREATE TRIGGER Check\_kratnost BEFORE INSERT*

*ON Booking*

*BEGIN*

*SELECT CASE*

*WHEN ((SELECT MinutesPerPerson FROM BookingAllowance WHERE*

*ROUND((JulianDay(BookingAllowance.ToTime) - JulianDay(NEW.BookingTime))\*24\*60)%BookingAllowance.MinutesPerPerson = 0 AND*

*BookingAllowance.ID = NEW.AllowanceID) ISNULL)*

*THEN RAISE(ABORT, 'Проверка на то, что запись в указанном временном интрвал 15 30 45 60 по 15 минут')*

*END;*

*END;*

*CREATE TRIGGER Check\_update BEFORE UPDATE*

*ON BookingAllowance*

*BEGIN*

*SELECT CASE*

*WHEN ((SELECT BookingTime FROM Booking WHERE NEW.ID = Booking.AllowanceID AND NEW.FromTime <= Booking.BookingTime AND NEW.ToTime >= Booking.BookingTime) ISNULL)*

*THEN RAISE(ABORT, 'Проверка обновления таблицы BookingAllowance')*

*END;*

*END;*

*CREATE VIEW BookingReport AS*

*SELECT DISTINCT DATE(B.BookingTime) AS Дата, D.Name AS Отдел,*

*ROUND(((JulianDay(BK.ToTime) - JulianDay(BK.FromTime))\*24\*60)/BK.MinutesPerPerson)*

*\*BK.SimultaneousPersons AS Всего, (SELECT count(\*) FROM BookingAllowance BK2, Booking B2*

*WHERE B2.AllowanceID = BK2.ID AND BK.ID = BK2.ID) AS Записалось,*

*ROUND(((JulianDay(BK.ToTime) - JulianDay(BK.FromTime))\*24\*60)/BK.MinutesPerPerson)*

*\*BK.SimultaneousPersons - (SELECT count(\*) FROM BookingAllowance BK2, Booking B2*

*WHERE B2.AllowanceID = BK2.ID AND BK.ID = BK2.ID) AS Осталось*

*FROM Booking B, Department D, BookingAllowance BK*

*WHERE B.AllowanceID = BK.ID AND BK.DepartmentID = D.ID*

*ORDER BY B.BookingTime, D.Name;*

**Результат выполнения программы:**

 

**Вывод:** научились работать со Spark и Scala.