# Отчет по главе “Триггеры” Шелудько А.А. ИТ2202

**Задача 1**

Вам доступна база данных, принадлежащая социальной сети "ВКонтакте". Она включает таблицу Users, которая хранит информацию о пользователях сети.

Напишите запрос, создающий триггер, который перед изменением номера телефона пользователя переводит его в следующий формат:

+7dddddddddd

Также запрос должен включать создание триггера, который при добавлении информации о новом пользователе переводит его номер телефона в формат, приведенный выше.

Гарантируется, что номер телефона пользователя всегда указывается в одном из следующих форматов:

* +7dddddddddd
* +8dddddddddd
* +7 dddddddddd
* +8 dddddddddd

где d — цифра в диапазоне [0; 9].

DROPTABLEIFEXISTSusers;

CREATETABLEUsers(idINTAUTO\_INCREMENT,nameVARCHAR(20),surnameVARCHAR(20),phone\_numberVARCHAR(20),PRIMARYKEY(id));INSERTINTOUsers(name,surname,phone\_number)VALUES('Matt','Damon','+79087333025'),('Edward','Norton','+79642218964'),('Nicolas','Cage','+79808814813'),('Ben','Affleck','+79042778299'),('John','Travolta','+79640950623');

DELIMITER //

CREATETRIGGERBeforeUserUpdateBEFOREUPDATEONUsers

FOREACHROW

BEGIN

SETNEW.phone\_number=CONCAT('+7',REPLACE(REPLACE(NEW.phone\_number,' ',''),'+8',''));

END//

CREATETRIGGERBeforeUserInsertBEFOREINSERTONUsers

FOREACHROW

BEGIN

SETNEW.phone\_number=CONCAT('+7',REPLACE(REPLACE(NEW.phone\_number,' ',''),'+8',''));

END//

DELIMITER;

SELECT\*FROMusers

**Задача 2**

Вам доступна база данных, принадлежащая социальной сети "ВКонтакте". Она включает таблицы Users и UsersEmailHistory. Первая таблица хранит информацию о пользователях сети, вторая — историю изменения адресов электронных почт пользователей.

Напишите запрос, создающий триггер, который после изменения пользователем адреса электронной почты фиксирует данное изменение и добавляет соответствующую информацию в таблицу UsersEmailHistory.

DROPTABLEIFEXISTSUsersEmailHistory;DROPTABLEIFEXISTSUsers;CREATETABLEUsers(idINTAUTO\_INCREMENT,nameVARCHAR(20),surnameVARCHAR(20),emailVARCHAR(40),PRIMARYKEY(id));INSERTINTOUsers(name,surname,email)VALUES('Matt','Damon','matt@gmail.com'),('Edward','Norton','ENorton.@outlook.com'),('Nicolas','Cage','ghostrider@outlook.com'),('Ben','Affleck','thebestbat@gmail.com'),('John','Travolta','WhereAmI@cloud.com');CREATETABLEUsersEmailHistory(log\_idINTAUTO\_INCREMENT,user\_idINT,old\_emailVARCHAR(40),new\_emailVARCHAR(40),updated\_onDATE,PRIMARYKEY(log\_id));INSERTINTOUsersEmailHistory(user\_id,old\_email,new\_email,updated\_on)VALUES(3,'cagecage2@gmail.com','ghostrider@outlook.com','2023-08-19'),(1,'mattie@cloud.com','matt@gmail.com','2023-08-20');

DELIMITER //

CREATETRIGGERAfterEmailUpdate

AFTERUPDATEONUsers

FOREACHROW

BEGIN

IFOLD.email!=NEW.emailTHEN

INSERTINTOUsersEmailHistory(user\_id,old\_email,new\_email,updated\_on)

VALUES(OLD.id,OLD.email,NEW.email,CURDATE());

ENDIF;

END//

DELIMITER;

SELECT\*FROMUsersEmailHistory

**Задача 3**

Вам доступна база данных, принадлежащая онлайн-кинотеатру "Окко". Она включает таблицы Films, Users и Purchases. Первая таблица хранит информацию о доступных в кинотеатре фильмах, вторая — информацию о пользователях кинотеатра, третья — информацию о покупках пользователей.

Напишите запрос, создающий триггер, который после покупки пользователем очередного фильма прибавляет к его сумме, потраченной на покупку фильмов, стоимость только что купленного фильма.

DROPTABLEIFEXISTSPurchases;DROPTABLEIFEXISTSUsers;DROPTABLEIFEXISTSFilms;CREATETABLEFilms(idINTPRIMARYKEYAUTO\_INCREMENT,titleVARCHAR(20),directorVARCHAR(20),priceDECIMAL(5,2));INSERTINTOFilms(title,director,price)VALUES('Toy Story 2','John Lasseter',2.99),('WALL-E','Andrew Stanton',4.99),('Ratatouille','Brad Bird',4.99),('Up','Pete Docter',4.99),('Brave','Brenda Chapman',7.99),('Monsters University','Dan Scanlon',7.99),('Cars 2','John Lasseter',7.99),('Finding Nemo','Andrew Stanton',4.99),('Toy Story','John Lasseter',2.99),('The Incredibles','Brad Bird',4.99);CREATETABLEUsers(idINTPRIMARYKEYAUTO\_INCREMENT,nameVARCHAR(40),surnameVARCHAR(40),total\_spendingDECIMAL(5,2));INSERTINTOUsers(name,surname,total\_spending)VALUES('Matt','Damon',23.96),('Edward','Norton',10.98),('Nicolas','Cage',10.98),('Ben','Affleck',7.98),('John','Travolta',0.0);CREATETABLEPurchases(idINTPRIMARYKEYAUTO\_INCREMENT,film\_idINT,user\_idINT);INSERTINTOPurchases(film\_id,user\_id)VALUES(1,1),(1,3),(2,4),(1,2),(9,4),(6,1),(7,2),(6,3),(5,1),(10,1);

DELIMITER //

CREATETRIGGERAfterPurchase

AFTERINSERTONPurchases

FOREACHROW

BEGIN

DECLAREfilmPriceDECIMAL(5,2);

SELECTpriceINTOfilmPrice

FROMFilms

WHEREid=NEW.film\_id;

UPDATEUsers

SETtotal\_spending=total\_spending+filmPrice

WHEREid=NEW.user\_id;

END//

DELIMITER;

SELECT\*FROMUsers

**Представления.**

**Задача 1**

Вам доступна база данных, принадлежащая социальной сети "ВКонтакте". Она состоит из одной таблицы с именем Users, которая хранит информацию о пользователях сети.

Напишите запрос, создающий представление с именем UsersWithEmail. Представление должно включать информацию (идентификатор, имя, фамилия, адрес электронной почты) о тех пользователях, адрес электронной почты которых известен.

DROPTABLEIFEXISTSUsers;CREATETABLEUsers(idINTPRIMARYKEYAUTO\_INCREMENT,nameVARCHAR(40),surnameVARCHAR(40),emailVARCHAR(40));INSERTINTOUsers(name,surname,email)VALUES('Christopher','Nolan',NULL),('Steven','Spielberg',NULL),('Quentin','Tarantino','queen@gmail.com'),('Martin','Scorsese',NULL),('David','Fincher','dfincher@outlook.com'),('Ridley','Scott','riddler@gmail.com'),('Stanley','Kubrick','bigstan@icloud.com'),('Clint','Eastwood','cowboy@yahoo.com'),('James','Cameron',NULL),('Tim','Burton','timmy@icloud.com');

CREATEVIEWUsersWithEmailAS

SELECTid,name,surname,email

FROMUsers

WHEREemailISNOTNULL;

SELECT\*FROMUsersWithEmail

**Задача 2**

Вам доступна база данных, принадлежащая средней школе Мидтауна. Она включает таблицу Students, которая хранит информацию об учениках школы.

Напишите запрос, извлекающий из предложенной базы данных идентификаторы групп, в которых нет ни максимального, ни минимального количества учеников.

DROPTABLEIFEXISTSStudents;CREATETABLEStudents(idINTPRIMARYKEYAUTO\_INCREMENT,nameVARCHAR(60),group\_idINT);INSERTINTOStudents(name,group\_id)VALUES('Alice',101),('Bob',102),('Charlie',101),('David',103),('Eva',102),('Frank',104),('Grace',101),('Hannah',105),('Ivy',103),('Jack',102),('Kate',104),('Liam',105),('Mia',101),('Noah',103),('Olivia',102),('Peter',102),('Quinn',101),('Riley',104),('Sophia',103),('Thomas',102);

WITHGroupCountsAS(

SELECTgroup\_id,COUNT(\*)asstudent\_count

FROMStudents

GROUPBYgroup\_id

),

MinMaxAS(

SELECTMIN(student\_count)asmin\_count,MAX(student\_count)asmax\_count

FROMGroupCounts

)

SELECTgc.group\_id

FROMGroupCountsgc,MinMaxmm

WHEREgc.student\_count>mm.min\_countANDgc.student\_count<mm.max\_count;

SELECT\*FROMStudents

**Задача 3**

Вам доступна база данных, принадлежащая социальной сети "ВКонтакте". Она включает таблицу Relations, которая хранит информацию о подписчиках пользователей.

Напишите запрос, который определяет все пары пользователей с максимальным количеством общих подписчиков. Другими словами, если максимальное количество общих подписчиков между любыми двумя пользователями равно max, то запрос должен определить все пары пользователей, у которых есть max общих подписчиков.

Поле с идентификатором первого пользователя должно иметь псевдоним user1\_id, поле с идентификатором второго пользователя — user2\_id. Также для каждой записи результирующей таблицы должно выполняться user1\_id < user2\_id.

DROPTABLEIFEXISTSRelations;CREATETABLERelations(user\_idINT,follower\_idINT);INSERTINTORelations(user\_id,follower\_id)VALUES(1,3),(2,3),(7,3),(1,4),(2,4),(7,4),(1,5),(2,6),(7,5),(3,1),(4,1),(3,2),(4,2),(5,1),(6,2),(5,7),(6,7),(5,4),(7,6),(6,5);

WITHCommonFollowersAS(

SELECT

r1.user\_idASuser1\_id,

r2.user\_idASuser2\_id,

COUNT(\*)AScommon\_followers\_count

FROMRelationsr1

JOINRelationsr2ONr1.follower\_id=r2.follower\_idANDr1.user\_id<r2.user\_id

GROUPBYr1.user\_id,r2.user\_id

)

SELECTuser1\_id,user2\_id

FROMCommonFollowers

WHEREcommon\_followers\_count=(

SELECTMAX(common\_followers\_count)FROMCommonFollowers

);

SELECT\*FROMRelations

**Задача 4**

Вам доступна база данных, принадлежащая интернет-магазину Amazon. Она включает таблицу Orders, которая хранит информацию о заказах покупателей.

Напишите запрос, который определяет уникальные идентификаторы товаров, которые были заказаны три или более раз в течение двух лет подряд.

DROPTABLEIFEXISTSOrders;CREATETABLEOrders(idINTPRIMARYKEYAUTO\_INCREMENT,product\_idINT,quantityINT,purchase\_dateDATE);INSERTINTOOrders(product\_id,quantity,purchase\_date)VALUES(1,7,'2020-03-16'),(1,4,'2020-05-02'),(1,7,'2020-07-10'),(1,6,'2021-02-23'),(1,5,'2021-05-21'),(1,6,'2021-10-11'),(3,5,'2019-08-08'),(8,1,'2021-06-03'),(2,5,'2023-12-25'),(7,7,'2022-07-10'),(3,5,'2019-09-24'),(8,5,'2022-07-29'),(7,5,'2023-01-16'),(3,7,'2022-12-21'),(3,3,'2021-06-16'),(3,7,'2019-06-11'),(8,3,'2021-06-13'),(8,1,'2020-05-25'),(3,3,'2023-06-14'),(8,5,'2022-01-13'),(7,9,'2023-07-19'),(5,3,'2023-12-01'),(4,8,'2022-02-15'),(3,7,'2020-09-06'),(8,3,'2020-11-23'),(4,9,'2023-10-05'),(7,2,'2023-10-12'),(8,5,'2020-01-03'),(6,6,'2022-06-21'),(8,6,'2021-07-01'),(3,4,'2018-12-26'),(3,5,'2019-08-11'),(3,7,'2022-02-21'),(3,5,'2018-09-04'),(7,3,'2022-03-07'),(6,5,'2021-01-01'),(3,4,'2022-03-18'),(5,2,'2023-01-31'),(8,7,'2022-07-19'),(7,3,'2022-10-27'),(2,6,'2023-10-16'),(8,5,'2020-08-04'),(4,7,'2022-08-28');

WITHYearlyOrdersAS(

SELECTproduct\_id,YEAR(purchase\_date)asyear,COUNT(\*)asorder\_count

FROMOrders

GROUPBYproduct\_id,YEAR(purchase\_date)

HAVINGorder\_count>=3

)

SELECTDISTINCTa.product\_id

FROMYearlyOrdersa

JOINYearlyOrdersbONa.product\_id=b.product\_idANDa.year=b.year-1;

SELECT\*FROMOrders

**Обобщенные табличные выражения.**

**Задача 1**

Вам доступна база данных, принадлежащая средней школе Мидтауна. Она включает таблицу Math, которая хранит информацию об оценках учеников выпускного класса за итоговый экзамен по математике.

У каждой буквенной оценки имеется текстовый эквивалент, который определяется согласно следующей таблице:



Напишите запрос, который разбивает студентов на группы в зависимости от текстового эквивалента их оценки, определяет количество студентов в каждой группе и отображает полученный результат в виде таблицы из двух полей:

1. result — текстовый эквивалент оценки (Great, Well или Bad)
2. students — количество студентов, получивших оценку, соответствующую этому текстовому эквиваленту

WITHGradeEquivalentAS(

SELECT

CASE

WHENgradeIN('A','B')THEN'Great'

WHENgradeIN('C')THEN'Well'

ELSE'Bad'

ENDASresult

FROMMath

)

SELECT

result,

COUNT(\*)asstudents

FROMGradeEquivalent

GROUPBYresult;

**Задача 2**

Вам доступна база данных, принадлежащая интернет-магазину Amazon. Она включает таблицы Customers и Orders. Первая таблица хранит информацию о покупателях магазина, вторая — информацию о заказах покупателей.

Напишите запрос, который определяет имя покупателя с наибольшей средней суммой заказа и указывает полученный результат в поле с псевдонимом customer.

WITHAverageOrdersAS(

SELECT

customer\_id,

AVG(amount)asavg\_amount

FROMOrders

GROUPBYcustomer\_id

)

SELECT

c.nameAScustomer

FROMCustomersc

JOINAverageOrdersaoONc.id=ao.customer\_id

ORDERBYao.avg\_amountDESC

LIMIT1;

**Задача 3**

Вам доступна база данных, принадлежащая маркетплейсуOzon. Она включает таблицу Orders, которая хранит информацию о заказах покупателей, совершенных в июне и июле 2023 года.

Напишите запрос, определяющий название магазина, который заработал наибольшее количество денег в июне, а также название магазина, который заработал наибольшее количество денег в июле.

Полученные значения должны быть указаны в полях с псевдонимами best\_in\_june и best\_in\_july соответственно.

WITHMonthlyEarningsAS(

SELECT

store,

SUM(amount)astotal\_amount,

MONTH(order\_date)asmonth

FROMOrders

WHEREMONTH(order\_date)IN(6,7)ANDYEAR(order\_date)=2023

GROUPBYstore,month

)

SELECT

(SELECTstoreFROMMonthlyEarningsWHEREmonth=6ORDERBYtotal\_amountDESCLIMIT1)asbest\_in\_june,

(SELECTstoreFROMMonthlyEarningsWHEREmonth=7ORDERBYtotal\_amountDESCLIMIT1)asbest\_in\_july;

**Задача 4**

Вам доступна база данных, принадлежащая маркетплейсуOzon. Она включает таблицу Orders, которая хранит информацию о заказах покупателей.

Напишите запрос, который извлекает из предложенной базы данных всю информацию о каждом заказе. Также рядом с данными о каждом заказе должна быть указана средняя сумма заказа в том магазине, в котором был совершен заказ.

Поле со средней суммой заказа в магазине должно иметь псевдоним avg\_for\_store. Значения в поле avg\_for\_store должны быть округлены до 1 знака после запятой.

WITHStoreAveragesAS(

SELECT

store,

ROUND(AVG(amount),1)asavg\_for\_store

FROMOrders

GROUPBYstore

)

SELECT

o.\*,

sa.avg\_for\_store

FROMOrderso

JOINStoreAveragessaONo.store=sa.store;

**Задача 5**

Вам доступна база данных, принадлежащая платформе для поиска авиабилетов Aviasales. Она включает таблицу Flights, которая хранит информацию о количестве доступных авиарейсов между различными аэропортами.

Трафиком аэропорта называется общее количество авиарейсов, в которых участвует аэропорт (в качестве аэропорта отправления или аэропорта прибытия). Например, трафик аэропорта с идентификатором 1 равен 19 + 1 = 20.

Напишите запрос, который определяет идентификаторы аэропортов с наибольшим трафиком.

Поле с идентификатором аэропорта должно иметь псевдоним airport\_id.

WITHAirportTrafficAS(

SELECT

departure\_airportASairport\_id,

SUM(flights\_count)AStraffic

FROMFlights

GROUPBYdeparture\_airport

UNIONALL

SELECT

arrival\_airportASairport\_id,

SUM(flights\_count)AStraffic

FROMFlights

GROUPBYarrival\_airport

)

SELECT

airport\_id

FROMAirportTraffic

GROUPBYairport\_id

ORDERBYSUM(traffic)DESC

LIMIT3;

**Задача 6**

Вам доступна база данных, принадлежащая налоговой службе. Она включает таблицу Salaries, которая хранит информацию о зарплатах сотрудников различных организаций.

Для каждой организации налоговая служба определяет собственную налоговую ставку:

* 0%, если максимальная зарплата любого сотрудника организации меньше 1000
* 24%, если максимальная зарплата любого сотрудника организации находится в диапазоне [1000; 10000] ​​​​​​
* 49%, если максимальная зарплата любого сотрудника организации больше 10000

Напишите запрос, который извлекает из предложенной базы данных идентификаторы и имена сотрудников организаций, а также зарплату каждого сотрудника после уплаты налогов.

Поле с идентификатором сотрудника должно иметь псевдоним id, поле с именем — name. Значения в поле salary должны быть округлены до ближайшего целого числа.

WITHTaxRatesAS(

SELECT

company\_id,

CASE

WHENMAX(salary)<1000THEN0

WHENMAX(salary)BETWEEN1000AND10000THEN0.24

ELSE0.49

ENDAStax\_rate

FROMSalaries

GROUPBYcompany\_id

)

SELECT

s.employee\_idASid,

s.employee\_nameASname,

ROUND(s.salary\*(1-tr.tax\_rate))ASsalary

FROMSalariess

JOINTaxRatestrONs.company\_id=tr.company\_id;

**Задача 7**

Вам доступна база данных, принадлежащая организации по проведению выборов в городе БрайтФоллс. Она включает таблицу Votes, которая хранит информацию о голосах избирателей.

Каждый избиратель имеет 1 голос, который он может отдать одному или нескольким кандидатам. Если избиратель голосует за нескольких кандидатов, его голос распределяются между ними поровну. Например, если избиратель голосует за 2кандидатов, каждый из них получит по 0.5 голосов.

Напишите запрос, который определяет имя кандидата, набравшего наибольшее количество голосов.

WITHVoteCountsAS(

SELECT

candidate,

SUM(1.0/COUNT(\*))OVER(PARTITIONBYvoter)asvotes

FROMVotes

GROUPBYvoter,candidate

)

SELECT

candidate

FROMVoteCounts

GROUPBYcandidate

ORDERBYSUM(votes)DESC

LIMIT1;

**Задача 8**

Вам доступна база данных, принадлежащая автовокзалу города БрайтФоллс. Она включает таблицы Buses и Passengers. Первая таблица хранит информацию об автобусах, вторая — информацию о пассажирах.

Пассажиры прибывают на автовокзал в определенное время и из доступных автобусов выбирают тот, который уезжает раньше всех.

Напишите запрос, который определяет, со скольким количеством пассажиров выедет из автовокзала каждый автобус, и отображает полученный результат в виде таблицы из двух полей:

* bus\_id — идентификатор автобуса
* passenger\_count — количество пассажиров в этом автобусе

Записи в результирующей таблице должны быть расположены в порядке возрастания значения поля bus\_id.

WITHBusPassengersAS(

SELECT

b.idASbus\_id,

p.idASpassenger\_id,

ROW\_NUMBER()OVER(PARTITIONBYp.idORDERBYb.departure\_time)asrn

FROMBusesb

JOINPassengerspONb.departure\_time>p.arrival\_time

)

SELECT

bus\_id,

COUNT(\*)ASpassenger\_count

FROMBusPassengers

WHERErn=1

GROUPBYbus\_id

ORDERBYbus\_id;

**Задача 1**

Напишите запрос, который извлекает таблицу, содержащую квадраты всех целых чисел от 1 до 10 включительно.

Поле с квадратом целого числа должно иметь псевдоним number.

WITHRECURSIVEnumbers(n,number)AS(

SELECT1,1

UNIONALL

SELECTn+1,(n+1)\*(n+1)

FROMnumbers

WHEREn<10

)

SELECT\*FROMnumbers;

**Задача 2**

Напишите запрос, который извлекает таблицу, содержащую все даты с 31 декабря 2024 года по 1 декабря 2024 включительно.

Поле с датой должно иметь псевдоним day.

WITHRECURSIVEdates(day)AS(

SELECTDATE('2024-12-31')

UNIONALL

SELECTDATE\_SUB(day,INTERVAL1DAY)

FROMdates

WHEREday>'2024-12-01'

)

SELECT\*FROMdates;

**Задача 3**

В SQL с помощью функции ORD() можно получить код символа в таблице ASCII.

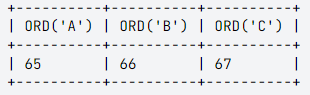
Результатом приведенного ниже запроса:

SELECTORD('A'),

ORD('B'),

ORD('C');

является:



С помощью функции CHAR() можно выполнить обратную операцию: получить символ по его коду.

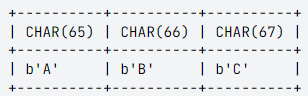
Результатом приведенного ниже запроса:

SELECTCHAR(65),

CHAR(66),

CHAR(67);

является:



Обратите внимание, что возвращаемым значением функции CHAR() является байтовая строка, а не обычная. При необходимости получить результат в виде обычной строки достаточно выполнить дополнительную конвертацию.

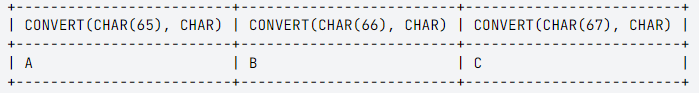
Результатом приведенного ниже запроса:

SELECTCONVERT(CHAR(65), CHAR),

CONVERT(CHAR(66), CHAR),

CONVERT(CHAR(67), CHAR);

является:



Применив предложенные функции, напишите запрос, который извлекает таблицу, содержащую все заглавные латинские буквы от A до Z включительно.

Поле с латинской буквой должно иметь псевдоним letter.

WITHRECURSIVEalphabet(letter,ascii\_value)AS(

SELECTCHAR(65),65

UNIONALL

SELECTCHAR(ascii\_value+1),ascii\_value+1

FROMalphabet

WHEREascii\_value<90

)

SELECTletterFROMalphabet;

**Задача 4**

Напишите запрос, который извлекает таблицу, содержащую названия всех 12 месяцев на английском языке.

Поле с названием месяца должно иметь псевдоним month.

WITHRECURSIVEmonths(n,month)AS(

SELECT1,DATE\_FORMAT(DATE(CONCAT('2024-',LPAD(1,2,'0'),'-01')),'%M')

UNIONALL

SELECTn+1,DATE\_FORMAT(DATE(CONCAT('2024-',LPAD(n+1,2,'0'),'-01')),'%M')

FROMmonths

WHEREn<12

)

SELECTmonthFROMmonths;

**Задача 5**

Напишите запрос, который извлекает таблицу, содержащую все целые числа от 1 до 20 включительно, а также их факториалы.

Поле с целым числом должно иметь псевдоним number, поле с факториалом числа — factorial.

WITHRECURSIVEfactorials(number,factorial)AS(

SELECT1,1

UNIONALL

SELECTnumber+1,factorial\*(number+1)

FROMfactorials

WHEREnumber<20

)

SELECT\*FROMfactorials;

**Задача 6**

Напишите запрос, который извлекает таблицу, содержащую 20 случайных целых чисел в диапазоне [10; 50].

Поле со случайным числом должно иметь псевдоним number.

WITHRECURSIVErandom\_numbers(n,number)AS(

SELECT1,FLOOR(10+RAND()\*41)

UNIONALL

SELECTn+1,FLOOR(10+RAND()\*41)

FROMrandom\_numbers

WHEREn<20

)

SELECTnumberFROMrandom\_numbers;

**Задача 7**

Вам доступна база данных, принадлежащая математическому ресурсу CleverStudents. Она включает таблицу Numbers, которая хранит различные целые числа.

Напишите запрос, который извлекает из предложенной базы данных все целые числа, располагая их в порядке возрастания и добавляя между ними отсутствующие значения (например, число 4 между числами 3 и 5).

WITHRECURSIVEsequenceAS(

SELECTMIN(num)ASnum

FROMNumbers

UNIONALL

SELECTnum+1

FROMsequence

WHEREnum+1<=(SELECTMAX(num)FROMNumbers)

)

SELECTnumFROMsequence;

**Задача 8**

Вам доступна база данных, принадлежащая маркетплейсуOzon. Она включает таблицу Orders, которая хранит информацию о заказах покупателей.

Напишите запрос, который извлекает из предложенной базы данных все даты, располагая их в порядке возрастания и добавляя между ними отсутствующие даты (например, дату 2023-12-30 между датами 2023-12-29 и 2023-12-31). Помимо этого, для каждой даты должно быть указано количество заказов, совершенных в эту дату.

Поле с количеством заказов, совершенных в определенную дату, должно иметь псевдоним orders\_count.

WITHRECURSIVEdate\_sequenceAS(

SELECTMIN(order\_date)ASdate

FROMOrders

UNIONALL

SELECTDATE\_ADD(date,INTERVAL1DAY)

FROMdate\_sequence

WHEREDATE\_ADD(date,INTERVAL1DAY)<=(SELECTMAX(order\_date)FROMOrders)

)

SELECT

ds.date,

IFNULL(COUNT(o.order\_date),0)ASorders\_count

FROMdate\_sequenceds

LEFTJOINOrdersoONds.date=o.order\_date

GROUPBYds.date;

**Задача 1**

Вам доступна база данных, принадлежащая операционной системе Windows. Она включает таблицу Files, которая хранит информацию о расположении файлов и папок внутри системы.

Напишите запрос, извлекающий из предложенной базы данных идентификаторы всех файлов и папок, а также указывающий для каждого файла или папки путь до него в следующем формате:

.../<название родительской папки>/<название файла или папки>

Поле с путем до файла или папки должно иметь псевдоним path.

SELECTid,name,CAST(nameASCHAR(1000))ASpath

FROMFiles

WHEREparent\_directory\_idISNULL

UNIONALL

SELECTf.id,f.name,CONCAT(fp.path,'/',f.name)

FROMFilesf

JOINfile\_pathsfpONf.parent\_directory\_id=fp.id

)

SELECTid,pathFROMfile\_paths;

**Задача 2**

Вам доступна база данных, принадлежащая компании Google. Она включает таблицу Investment, хранящую информацию о суммах, которые могут быть инвестированы в компанию.

Каждая инвестиционная сумма может быть инвестирована одним, двумя или тремя инвесторами. Если сумма инвестируется несколькими инвесторами, она делится между ними поровну.

Напишите запрос, который разделяет каждую инвестиционную сумму на равные доли в зависимости от количества инвесторов и отображает полученный результат в виде таблицы из трех полей:

* investors\_number — количество инвесторов
* investment\_amount — инвестиционная сумма
* individual\_amount — доля каждого инвестора

WITHRECURSIVEinvestment\_splits(investors\_number,investment\_amount,individual\_amount)AS(

SELECT1,investment\_amount,investment\_amount

FROMInvestment

UNIONALL

SELECTinvestors\_number+1,investment\_amount,investment\_amount/(investors\_number+1)

FROMinvestment\_splits

WHEREinvestors\_number<3

)

SELECT\*FROMinvestment\_splits;

**Задача 3**

Вам доступна база данных, принадлежащая платформе для поиска авиабилетов Aviasales. Она включает таблицу Routes, которая хранит информацию об авиарейсах между различными городами.

Напишите запрос, извлекающий из предложенной базы данных названия городов, в которые можно добраться из города NewYork (напрямую или с помощью промежуточных рейсов), а также указывающий расстояние маршрута до каждого города.

WITHRECURSIVEreachable\_citiesAS(

SELECTdestination\_city,distance

FROMRoutes

WHEREsource\_city='New York'

UNIONALL

SELECTr.destination\_city,r.distance+rc.distance

FROMRoutesr

JOINreachable\_citiesrcONr.source\_city=rc.destination\_city

)

SELECTdestination\_cityAScity,MIN(distance)ASdistance

FROMreachable\_cities

GROUPBYdestination\_city;

**Задача 4**

Вам доступна база данных, принадлежащая компании Google. Она включает таблицу Tasks, которая хранит информацию о задачах, которые должны быть выполнены сотрудниками компании.

Напишите запрос, извлекающий из предложенной базы данных названия всех задач, а также указывающий для каждой задачи время, необходимое для ее выполнения, с учетом времени решения всех задач, от которых она зависит.

Поле с конечным временем, необходимым для выполнения задачи, должно иметь псевдоним total\_time.

WITHRECURSIVEtask\_timesAS(

SELECTid,name,time\_requiredAStotal\_time

FROMTasks

WHEREdepends\_on\_task\_idISNULL

UNIONALL

SELECTt.id,t.name,t.time\_required+tt.total\_timeAStotal\_time

FROMTaskst

JOINtask\_timesttONt.depends\_on\_task\_id=tt.id

)

SELECTname,total\_timeFROMtask\_times;

**Задача 5**

Вам доступна база данных, принадлежащая платформе для поиска авиабилетов Aviasales. Она включает таблицу Routes, которая хранит информацию об авиарейсах между различными городами.

Напишите запрос, определяющий все возможные маршруты, которыми можно добраться из города Groningen в город Haarlem, и указывающий полученные результаты в поле с псевдонимом route в следующем формате:

Groningen -><название промежуточного города> -> ... -><название промежуточного города> -> Haarlem

WITHRECURSIVEtravel\_routesAS(

SELECTsource\_city,destination\_city,CONCAT(source\_city,' -> ',destination\_city)ASroute

FROM(SELECT\*FROMRoutes)ASstart

WHEREsource\_city='Groningen'

UNIONALL

SELECTtr.source\_city,r.destination\_city,CONCAT(tr.route,' -> ',r.destination\_city)

FROMtravel\_routestr

JOIN(SELECT\*FROMRoutes)ASrONtr.destination\_city=r.source\_city

WHEREtr.destination\_city!='Haarlem'

)

SELECTroute

FROMtravel\_routes

WHEREdestination\_city='Haarlem';

**Задача 6**

Вам доступна база данных, принадлежащая неизвестной семье. Она включает таблицу Genealogy, которая хранит информацию о родительских отношениях между определенными членами семьи.

Напишите запрос, извлекающий из предложенной базы данных имена всех членов семьи, а также указывающий для каждого члена его родственное отношение к члену семьи по имени James:

* mother — мать
* father — отец
* grandmother — бабушка
* grandfather — дедушка

Сам James по отношению к самому себе должен являться сыном — son.

Поле с родственным отношением члена семьи должно иметь псевдоним relationship.

WITHRECURSIVEfamily\_treeAS(

SELECTid,name,father\_id,mother\_id,CAST('son'ASCHAR(20))ASrelationship

FROMGenealogy

WHEREname='James'

UNIONALL

SELECTg.id,g.name,g.father\_id,g.mother\_id,

CASE

WHENft.relationship='son'ANDg.id=ft.father\_idTHENCAST('father'ASCHAR(20))

WHENft.relationship='son'ANDg.id=ft.mother\_idTHENCAST('mother'ASCHAR(20))

WHENft.relationship='father'ANDg.id=ft.father\_idTHENCAST('grandfather'ASCHAR(20))

WHENft.relationship='father'ANDg.id=ft.mother\_idTHENCAST('grandmother'ASCHAR(20))

WHENft.relationship='mother'ANDg.id=ft.father\_idTHENCAST('grandfather'ASCHAR(20))

WHENft.relationship='mother'ANDg.id=ft.mother\_idTHENCAST('grandmother'ASCHAR(20))

ENDASrelationship

FROMGenealogyg

JOINfamily\_treeftONg.id=ft.father\_idORg.id=ft.mother\_id

)

SELECTname,relationshipFROMfamily\_treeWHERErelationshipISNOTNULL;