**Проект «Служебные телефоны»**

**Оглавление**

1. Задание…………………………………………………………….........................2
2. Инфологическая модель………………………………………………………….3
3. Простые запросы……………………………………………………….....…….3-4
4. Сложные запросы……………………………………………………………….4-5
5. Триггер и процедура……………………………………………………………6-7

**Задание:**

Служебные телефоны.

Компания хранит сведения о своих сотрудниках в формате: уникальный номер сотрудника, фамилия, адрес, дата рождения, название должности. Дата зачисления на должность, номер отдела, в котором работает сотрудник.

Каждому сотруднику компания предоставляет служебную сотовую связь (ССС) в формате лимита оплачиваемых компанией разговоров в месяц. Хранится код сотрудника, номер телефона и предоставленный ему лимит в рублях.

Каждый месяц компания получает счет от оператора мобильной связи. Каждый счет имеет номер, дату выставления, месяц и год, за который выставлен счет, общую сумму счета. В строках счета перечислены номера телефонов и сумма к оплате по каждому телефону за указанный период.

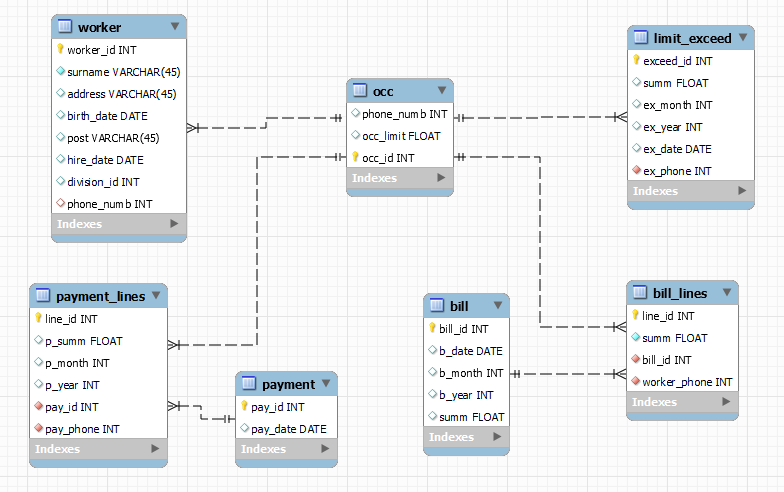
Компания сама оплачивает эти счета оператору мобильной связи.

Каждый месяц после получения счета в компании формируются данные о превышениях лимита. Заносится номер телефона, сумма превышения лимита и месяц, и год, в котором это превышение произошло.

Превышение лимита должно быть оплачено самим сотрудником. Данные об оплате также хранятся в превышениях лимита. Для этого предусмотрена дата погашения лимита превышения лимита. Если дата погашения отсутствует, то сотрудник еще не погасил задолженность.

Оплата производится сотрудниками через банк, из которого каждый месяц поступает платежка. Каждая платежка имеет номер и дату создания. В каждой строке платежки указаны: номер телефона, полученная сумма, месяц и год, за который сделан платеж.

**Инфологическая модель**

****

**Простые запросы**

1. Показать превышение лимита по каждому телефону за март 2020 года.

SELECT summ, exceed\_phone

FROM limit\_exceed

WHERE ex\_month=3 AND ex\_year=2020;

1. Показать общую сумму превышений лимита за март 2020 года.

SELECT SUM(summ)

FROM limit\_exceed

WHERE ex\_month=3 AND ex\_year=2020;

1. Показать все счета, выставленные за последние 60 дней.

SELECT bill\_id

FROM bill

WHERE datediff(now(),b\_date)<=60;

1. Показать дату рождения самого молодого сотрудника.

SELECT MAX(birth\_date)

FROM worker

1. Показать среднюю сумму счетов за первое полугодие 2020 года.

SELECT AVG(summ)

FROM bill

WHERE b\_year=2020 AND b\_month BETWEEN 1 AND 6;

1. Показать сколько служебных телефонов имеют префикс 903.

SELECT COUNT(\*)

FROM occ

WHERE phone\_numb LIKE '903%';

**Сложные запросы**

1. Сформировать отчет о превышениях лимита в марте 2020 года по форме: Фамилия сотрудника, номер телефона, величина превышения лимита.

SELECT surname, ex\_phone, summ

FROM worker JOIN limit\_exceed ON worker.phone\_numb=limit\_exceed.ex\_phone

WHERE ex\_month=3 AND ex\_year=2020

1. Показать сколько сотрудников каждого отдела превысили лимит разговоров в марте 2020 года.

SELECT post, COUNT(\*) as ex\_kol

FROM worker JOIN limit\_exceed ON worker.phone\_numb=limit\_exceed.ex\_phone

WHERE ex\_month=3 AND ex\_year=2020

GROUP BY post;

1. Показать все сведения о самом старом по возрасте сотруднике.

SELECT \*

FROM worker

WHERE birth\_date=(SELECT MIN(birth\_date) FROM worker);

1. Показать фамилии и номера отделов сотрудников, у которых нет служебных телефонов.

SELECT surname, division\_id

FROM worker LEFT JOIN occ USING(phone\_numb)

WHERE phone\_numb IS NULL;

1. Показать список сотрудников отдела ХХХ, которые в марте 2020 года не превышали лимита.

select surname from worker w left join

(select ex\_phone from limit\_exceed where ex\_month = 3 and ex\_year = 2020) march\_exceeded

on w.phone\_numb = march\_exceeded.ex\_phone

where

march\_exceeded.ex\_phone is NULL

1. Показать сведения о сотрудниках. Чаще всех превышавших лимит в 2020 (с помощью view).

CREATE VIEW exceed\_maxer (phone, ex\_kol) AS

SELECT ex\_phone, COUNT(\*)

FROM limit\_exceed

WHERE ex\_year = 2020

GROUP BY ex\_phone;

SELECT \*

FROM worker JOIN exceed\_maxer ON worker.phone\_numb=exceed\_maxer.phone

WHERE ex\_kol = (SELECT MAX(ex\_kol) FROM exceed\_maxer);

**Триггеры и хранимые процедуры**

Разработать процедуру, которая будет получать на вход номер счета от оператора связи. В процедуре реализовать вычисление и сохранение в БД превышений лимитов по каждому телефону за соответствующий месяц.

Реализовать триггер, который будет срабатывать при занесении в БД новой строки счета. В триггере реализовать вычисление превышения лимита для соответствующего телефона за соответствующий месяц.

CREATE DEFINER=`root`@`localhost` PROCEDURE `count\_exceed`(bill\_num integer)

BEGIN

DECLARE M, Y integer;

DECLARE done integer DEFAULT 0;

DECLARE phone integer;

DECLARE w\_sum, w\_limit float;

DECLARE C1 CURSOR FOR SELECT worker\_phone, summ, occ\_limit

FROM bill\_lines JOIN occ ON bill\_lines.worker\_phone = occ.phone\_numb

WHERE bill\_id = bill\_num;

DECLARE EXIT HANDLER FOR SQLSTATE '02000'

SET done = 1;

SELECT b\_month, b\_year INTO M, Y

FROM bill WHERE bill\_id = bill\_num;

OPEN C1;

WHILE done = 0 DO

FETCH C1 INTO phone, w\_sum, w\_limit;

IF w\_sum > w\_limit THEN

INSERT INTO limit\_exceed VALUES(NULL, w\_sum-w\_limit, M, Y, NULL, phone);

END IF;

END WHILE;

END

CREATE DEFINER=`root`@`localhost` TRIGGER `bill\_lines\_AFTER\_INSERT` AFTER INSERT ON `bill\_lines` FOR EACH ROW BEGIN

DECLARE M, Y integer;

DECLARE w\_limit float;

SELECT b\_month, b\_year INTO M, Y

FROM bill

WHERE bill\_id = new.bill\_id;

SELECT occ\_limit INTO w\_limit

FROM occ

WHERE phone\_numb = new.worker\_phone;

IF new.summ > w\_limit THEN

INSERT INTO limit\_exceed VALUES(NULL, new.summ-w\_limit, M, Y, NULL, new.worker\_phone);

END IF;

END