1. Считалка.

Таблица PRODUCTS состоит из двух столбцов PRODUCT\_ID (number, PK) – идентификатор товара COST(number) – цена товара. Вывести в одну строку количество товаров, цена которых меньше 100, цена которых лежит в диапазоне 100-200 и цена которых больше 200.

Способ1:

SELECT DISTINCT

(SELECT count(product\_id)

FROM productcts

WHERE cost<100) AS costless\_100,

(SELECT count(product\_id)

FROM productcts

WHERE cost>=100 and cost<=200) AS costrange\_100\_200,

(SELECT count(product\_id)

FROM productcts

WHERE cost>200) AS cost\_more\_200

FROM productcts

Способ2:

SELECT  
 count(distinct(case when cost < 100 then product\_id end)) as costless\_100,  
 count(distinct(case when cost >= 100 and cost <= 200 then product\_id end)) as cost\_range\_100\_200,  
 count(distinct(case when cost > 200 then product\_id end)) as cost\_more\_200

FROM products

1. Календарь.

Таблица CALENDAR состоит из двух столбцов CALENDAR\_DATE(Date, PK) – дата IS\_DAYOFF (number) – признак выходного дня (0-рабочий 1-выходной). Таблица MESSAGE состоит из трех столбцов DOCUMENT\_ID (number, PK) – идентификатор сообщения(ответа), DOCUMENT\_DT (date) – дата отправки сообщения (получения ответа), CHILD\_DOCUMENT\_ID (number, nullable) – идентификатор ответа на сообщение. Сообщения и ответы на сообщения сохраняются в эту таблицу. Запрос должен отображать идентификаторы сообщений, по которым не получен ответ в течение 5 рабочих дней со дня отправки сообщения.

Табл. message

|  |  |  |
| --- | --- | --- |
| Document\_ID | Child\_Document\_ID | Document\_dt |
| 1 | 1.1 | 01.07.2022 |
| 2 | null | 04.07.2022 |
| 3 | 3.1 | 05.07.2022 |
| 3.1 | null | 06.07.2022 |
| 1.1 | null | 08.07.2022 |

Табл. new\_message

|  |  |  |  |
| --- | --- | --- | --- |
| T1.Document\_ID | T1.Child\_Document\_ID | T1.Document\_dt | T2.Document\_dt |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 |
| 2 | null | 08.07.2022 | 10.07.2022 |
| 3 | 3.1 | 05.07.2022 | 06.07.2022 |
| 3.1 | null | 06.07.2022 | 10.07.2022 |
| 1.1 | null | 08.07.2022 | 10.07.2022 |

Табл.calendar

|  |  |
| --- | --- |
| calendar\_date | is\_dayoff |
| 01.07.2022 | 0 |
| 02.07.2022 | 1 |
| 03.07.2022 | 1 |
| 04.07.2022 | 0 |
| 05.07.2022 | 0 |
| 06.07.2022 | 0 |
| 07.07.2022 | 0 |
| 08.07.2022 | 0 |
| 09.07.2022 | 1 |
| 10.07.2022 | 1 |

Табл. New\_message\_calendar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n.Document\_ID | n.Child\_Document\_ID | n.Document\_dt | n.child\_document\_dt | c.calendar\_date | c.is\_dayoff |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 01.07.2022 | 0 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 02.07.2022 | 1 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 03.07.2022 | 1 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 04.07.2022 | 0 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 05.07.2022 | 0 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 06.07.2022 | 0 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 07.07.2022 | 0 |
| 1 | 1.1 | 01.07.2022 | 08.07.2022 | 08.07.2022 | 0 |
|  |  |  |  |  |  |
| 2 | null | 08.07.2022 | 10.07.2022 | 08.07.2022 | 0 |
| 2 | null | 08.07.2022 | 10.07.2022 | 09.07.2022 | 1 |
| 2 | null | 08.07.2022 | 10.07.2022 | 10.07.2022 | 1 |
|  |  |  |  |  |  |
| 3 | 3.1 | 05.07.2022 | 06.07.2022 | 05.07.2022 | 0 |
| 3 | 3.1 | 05.07.2022 | 06.07.2022 | 06.07.2022 | 0 |
|  |  |  |  |  |  |
| 3.1 | null | 06.07.2022 | 10.07.2022 | 06.07.2022 | 0 |
| 3.1 | null | 06.07.2022 | 10.07.2022 | 07.07.2022 | 0 |
| 3.1 | null | 06.07.2022 | 10.07.2022 | 08.07.2022 | 0 |
| 3.1 | null | 06.07.2022 | 10.07.2022 | 09.07.2022 | 1 |
| 3.1 | null | 06.07.2022 | 10.07.2022 | 10.07.2022 | 1 |
|  |  |  |  |  |  |
| 1.1 | null | 08.07.2022 | 10.07.2022 | 08.07.2022 | 0 |
| 1.1 | null | 08.07.2022 | 10.07.2022 | 09.07.2022 | 1 |
| 1.1 | null | 08.07.2022 | 10.07.2022 | 10.07.2022 | 1 |

With  
new\_message as (  
SELECT  
 t1.\*,  
 ifnull(t2.document\_dt, getdate()) as child\_document\_dt

FROM message t1 LEFT JOIN message t2 on t1.child\_document\_id = t2. document\_id

),

New\_message\_calendar as (

SELECT

n.\*,

,c.is\_dayoff

FROM new\_message n LEFT JOIN calendar c on n.Document\_dt <= c.Calendar\_date and n child\_document\_dt >= c.Calendar\_date

),

Final\_table as (

SELECT

Document\_ID,

count(case when is\_dayoff = 0 then document\_id end) as counter\_0

FROM New\_message\_calendar

GROUP BY document\_id

),

SELECT

Document\_id

FROM final\_table

WHERE counter\_0 > 5

1. Создание объектов БД

Написать скрипты на создание таблиц из заданий 1 и 2

к заданию 1:

CREATE TABLE products (

product\_id int AUTO\_INCREMENT, --в зависимости в какой базе работать, так же можно

эту строку создать product\_id int identity(1,1)

нумерация будет идти автоматически, поочередно--

cost decimal(10,2) NOT NULL, --так же зависит от базы,можно тип данных задать money--

PRIMARY KEY (product\_id)

);

к заданию 2:

CREATE TABLE Calendar (

Calendar\_date date,

Is\_Dayoff int NOT NULL,

PRIMARY KEY (Calendar\_date)

);

CREATE TABLE Message(

Document\_id int,

Document\_dt date,

Child\_Document\_ID int,

PRIMARY KEY (Document\_id)

);

1. Тестовые данные

Разработать скрипты (процедуры) для генерации тестовых данных. Входным параметром процедуры должен быть параметр – необходимое количество строк с тестовыми данными

* Сгенерировали случайный данные:

CREATE TABLE Test\_Table (randnum SMALLINT);

DECLARE @num AS SMALLINT;

DECLARE @testcount AS SMALLINT = 10000;

WHILE @testcount > 0

BEGIN

SET @num = (SELECT FLOOR(RAND() \* (10)) + 1);

INSERT INTO Test\_Table (randnum)

VALUES (@num);

SET @testcount -= 1;

END

* Cоздали процедуру

CREATE PROCEDURE new\_procedure

@randnum nvarchar(100)

AS

SELECT count(\*) AS 'необходимое количество строк с тестовыми данными '

FROM Test\_Table

WHERE randnum = @randnum;

* Вызвать процедуру можно любым из этих способов

EXECUTE new\_procedure '1';

EXEC new\_procedure @randnum = '1';

* Удалить процедуру

DROP PROCEDURE new\_procedure