

# Хранимые процедуры

## 1. Счёт за проживание и услуги

На вход подается ID проживания и складывается стоимость проживания с суммой за оказанные услуги. Если ID проживания не существует, возвращается 0, если же существует, но не присутствует в таблице оказанных услуг, то возвращается стоимость проживания(цена номера).

Код.

```
CREATE PROCEDURE BillForAccommodationAndServices
@AccommodationID INT
AS
BEGIN
IF (NOT EXISTS (SELECT * FROM ServicesRendered WHERE AccommodationID =
@AccommodationID) OR
NOT EXISTS (SELECT * FROM AccommodationInfo WHERE ID = @AccommodationID))
RETURN 0
DECLARE @SumServices INT = (SELECT SUM(Price) FROM ServicesRendered WHERE
AccommodationID = @AccommodationID)
DECLARE @AccommodationCost INT = (SELECT Cost FROM AccommodationInfo
WHERE ID = @AccommodationID)
RETURN @SumServices + @AccommodationCost
END
```

### Пример 1.

```
DECLARE @Sum INT
EXEC @Sum = BillForAccommodationAndServices @AccommodationID = 2
SELECT Bill = @Sum
```

	Bill
1	4200

Действительно, за проживание клиент заплатил 3000:

	ID	ClientID	RoomID	Booking...	ArrivalDate	DepartureDate	Cost
1	2	1	35	1	2022-02-25 12:10:00.000	2022-02-28 11:41:00.000	3000

За услуги в сумме 1200:

	ID	Accommodatio...	ServiceDate	Service...	Employee...	Price
1	11	2	2022-02-25 12:10:00.000	1	1	0
2	12	2	2022-02-25 12:10:00.000	2	2	0
3	13	2	2022-02-26 12:10:00.000	1	1	0
4	14	2	2022-02-26 12:10:00.000	2	2	0
5	15	2	2022-02-26 12:10:00.000	8	4	200
6	16	2	2022-02-26 12:10:00.000	7	3	1000
7	17	2	2022-02-27 12:10:00.000	1	1	0
8	18	2	2022-02-27 12:10:00.000	2	2	0
9	19	2	2022-02-28 12:10:00.000	2	2	0

## Пример 2.

Клиент не получал услуги, выводится стоимость проживания:

```
DECLARE @Sum INT
```

```
EXEC @Sum = BillForAccommodationAndServices @AccommodationID = 3
```

```
SELECT Bill = @Sum
```

	Bill
1	6000

## Пример 3.

Такого ID проживания нет:

```
DECLARE @Sum INT
```

```
EXEC @Sum = BillForAccommodationAndServices @AccommodationID = 9
```

```
SELECT Bill = @Sum
```

	Bill
1	0

## 2. Список проживавших на момент времени

На вход подается дата, на момент которой выводится список проживающих клиентов.

Код.

```
CREATE PROCEDURE AccommodationListAtTheTime
```

```
@CurrentDate date
```

```
AS
```

```
BEGIN
```

```

DECLARE @AccommodationID INT, @ClientID INT, @ClientFirstName varchar(255),
@ClientLastName varchar(255),
@ArrivalDate datetime, @DepartureDate datetime, @RoomID INT
DECLARE myCursor CURSOR FOR
SELECT acc.ID, acc.ClientID, cl.FirstName, cl.LastName, acc.ArrivalDate,
acc.DepartureDate, acc.RoomID
FROM AccommodationInfo AS acc
INNER JOIN Client AS cl ON cl.ID = acc.ClientID
WHERE ArrivalDate <= @CurrentDate AND DepartureDate >= @CurrentDate
OPEN myCursor
FETCH NEXT FROM myCursor INTO @AccommodationID, @ClientID, @ClientFirstName,
@ClientLastName,
@ArrivalDate, @DepartureDate, @RoomID
WHILE @@FETCH_STATUS = 0
BEGIN
SELECT
AccommodationID = @AccommodationID, ClientID = @ClientID,
ClientFirstName = @ClientFirstName, ClientLastName = @ClientLastName,
ArrivalDate = @ArrivalDate, DepartureDate = @DepartureDate, RoomID = @RoomID
FETCH NEXT FROM myCursor INTO @AccommodationID, @ClientID, @ClientFirstName,
@ClientLastName,
@ArrivalDate, @DepartureDate, @RoomID
END
CLOSE myCursor
DEALLOCATE myCursor
END

```

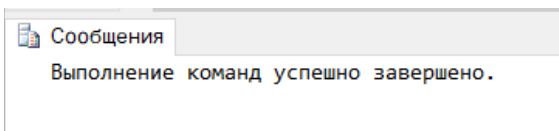
## Пример 1.

EXEC AccommodationListAtTheTime @CurrentDate = '2022-02-28'

Результаты							
	Accommodatio...	ClientID	ClientFirstNa...	ClientLastNa...	ArrivalDate	DepartureDate	RoomID
1	2	1	Маргарита	Киреева	2022-02-25 12:10:00.000	2022-02-28 11:41:00.000	35
	Accommodatio...	ClientID	ClientFirstNa...	ClientLastNa...	ArrivalDate	DepartureDate	RoomID
1	3	2	Алина	Загитова	2022-02-27 12:13:56.000	2022-03-03 10:21:42.590	31

## Пример 2.

EXEC AccommodationListAtTheTime @CurrentDate = '2022-02-02'



### 3. Списки броней на определенный срок

Выводится список броней на указанный срок(входные параметры – дата прибытия и дата отбытия). Если дата прибытия позже даты отбытия выводится ошибка.

Код.

```
CREATE PROCEDURE GetBookingListForPeriod
@ArrivalDate date, @DepartureDate date
AS
BEGIN
IF (@ArrivalDate > @DepartureDate)
RAISERROR ('Дата прибытия не может быть позже даты отбытия', 16, 1)
DECLARE myCursor CURSOR FOR
SELECT * FROM BookingInfo
WHERE ArrivalDate >= @ArrivalDate AND DepartureDate <= @DepartureDate
OPEN myCursor
FETCH NEXT FROM myCursor
WHILE @@FETCH_STATUS = 0
BEGIN
FETCH NEXT FROM myCursor
END
CLOSE myCursor
DEALLOCATE myCursor
END
```

#### Пример 1.

Таблица BookingInfo(все брони)

	ID	ClientID	RoomTyp...	ArrivalDate	DepartureDate	BookingDate	HotelID	RealizationInfo
1	1	1	3	2022-02-25 12:00:00.000	2022-02-28 12:00:00.000	2022-02-20 12:56:09.560	1	Реализовано
2	2	2	7	2022-02-27 12:00:00.000	2022-03-03 12:00:00.000	2022-02-23 12:56:09.560	4	Реализовано
3	3	3	4	2022-03-02 12:00:00.000	2022-03-14 12:00:00.000	2022-02-28 12:56:09.560	1	Реализовано
4	4	4	1	2022-03-05 12:00:00.000	2022-03-12 12:00:00.000	2022-03-01 12:56:09.560	2	Реализовано
5	5	5	2	2022-03-15 12:00:00.000	2022-03-27 12:00:00.000	2022-02-28 12:56:09.560	1	Не реализовано
6	6	6	4	2022-04-05 12:00:00.000	2022-04-13 12:00:00.000	2022-03-20 12:56:09.560	1	Реализовано
7	7	7	3	2022-04-25 12:00:00.000	2022-04-30 12:00:00.000	2022-04-10 12:56:09.560	1	Не реализовано
8	8	8	6	2022-04-27 12:00:00.000	2022-05-05 12:00:00.000	2022-04-15 12:56:09.560	3	Не реализовано
9	11	1	3	2022-04-25 12:00:00.000	2022-04-28 12:00:00.000	2022-04-20 12:56:09.560	1	Не реализовано

EXEC GetBookingListForPeriod

@ArrivalDate = '2022-02-26', @DepartureDate = '2022-03-15'

Срок от 26.02.2022 до 15.03.2022:

Результаты Сообщения									
	ID	ClientID	RoomTyp...	ArrivalDate	DepartureDate	BookingDate	HotelID	Realization...	
1	2	2	7	2022-02-27 12:00:00.000	2022-03-03 12:00:00.000	2022-02-23 12:56:09.560	4	Реализовано	
1	3	3	4	2022-03-02 12:00:00.000	2022-03-14 12:00:00.000	2022-02-28 12:56:09.560	1	Реализовано	
1	4	4	1	2022-03-05 12:00:00.000	2022-03-12 12:00:00.000	2022-03-01 12:56:09.560	2	Реализовано	

## Пример 2.

EXEC GetBookingListForPeriod

@ArrivalDate = '2022-03-15', @DepartureDate = '2022-02-26'

Срок от 15.03.2022 до 26.02.2022:

Результаты Сообщения									
Сообщение 50000, уровень 16, состояние 1, процедура GetBookingListForPeriod, строка 40 Дата прибытия не может быть позже даты отбытия									

## 4. Расчет общей прибыли гостиниц за заданный период

На вход подаются дата начала периода и дата его конца. Прибыль рассчитывается как в цикле как сумма всех счетов за проживание и услуги для каждого ID проживания(то есть вызывается функция BillForAccommodationAndServices). Если дата начала периода позже даты конца выводится ошибка.

### Код.

```
CREATE PROCEDURE HotelSumProfitForPeriod
```

```
@BeginDate date, @EndDate date
```

```
AS
```

```
BEGIN
```

```
IF (@BeginDate > @EndDate)
```

```
RAISERROR ('Дата начала периода не может быть позже даты конца', 16, 1)
```

```
DECLARE @Profit INT = 0, @AccommodationID INT, @CurrentBill INT
```

```
DECLARE myCursor CURSOR FOR
```

```
SELECT ID
```

```
FROM AccommodationInfo
```

```
OPEN myCursor
```

```

FETCH NEXT FROM myCursor INTO @AccommodationID
WHILE @@FETCH_STATUS = 0
BEGIN
EXEC @CurrentBill = BillForAccommodationAndServices @AccommodationID =
@AccommodationID
SET @Profit = @Profit + @CurrentBill
FETCH NEXT FROM myCursor INTO @AccommodationID
END
CLOSE myCursor
DEALLOCATE myCursor
RETURN @Profit
END

```

## Пример 1.

```

DECLARE @HotelProfit INT
EXEC @HotelProfit = HotelSumProfitForPeriod @BeginDate = '2022-03-01', @EndDate =
'2022-03-31'
SELECT 'Прибыль гостиницы' = @HotelProfit

```

Результаты		Сообщения	
Прибыль гостиницы			
1	25200		

## Пример 2.

```

DECLARE @HotelProfit INT
EXEC @HotelProfit = HotelSumProfitForPeriod @BeginDate = '2022-04-01', @EndDate =
'2022-03-01'
SELECT 'Прибыль гостиницы' = @HotelProfit

```

Результаты		Сообщения	
		Сообщение 50000, уровень 16, состояние 1, процедура HotelProfitForPeriod, строка 133 Дата начала периода не может быть позже даты конца	
		(строк обработано: 1)	

# 5. Расчет прибыли заданной гостиницы за заданный период

На вход подаются ID отеля, дата начала периода и дата его конца. Прибыль рассчитывается в цикле как сумма счетов за проживание и услуги для каждого ID проживания(то есть вызывается функция BillForAccommodationAndServices), где ID

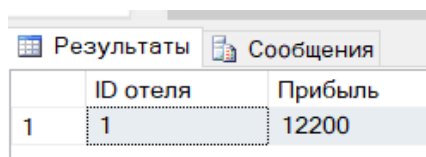
отеля равен входному параметру. Если дата начала периода позже даты конца выводится ошибка.

## Код.

```
CREATE PROCEDURE HotelProfitForPeriod
@HotelID int, @BeginDate date, @EndDate date
AS
BEGIN
IF (@BeginDate > @EndDate)
RAISERROR ('Дата начала периода не может быть позже даты конца', 16, 1)
DECLARE @Profit INT = 0, @AccommodationID INT, @CurrentBill INT
DECLARE myCursor CURSOR FOR
SELECT ID
FROM AccommodationInfo
WHERE RoomID IN (SELECT ID FROM Room WHERE HotelID = @HotelID)
OPEN myCursor
FETCH NEXT FROM myCursor INTO @AccommodationID
WHILE @@FETCH_STATUS = 0
BEGIN
EXEC @CurrentBill = BillForAccommodationAndServices @AccommodationID =
@AccommodationID
SET @Profit = @Profit + @CurrentBill
FETCH NEXT FROM myCursor INTO @AccommodationID
END
CLOSE myCursor
DEALLOCATE myCursor
RETURN @Profit
END
```

## Пример 1.

```
DECLARE @HotelProfit INT
EXEC @HotelProfit = HotelProfitForPeriod @BeginDate = '2022-03-01', @EndDate =
'2022-03-31', @HotelID = 1
SELECT 'ID отеля' = 1, 'Прибыль' = @HotelProfit
```

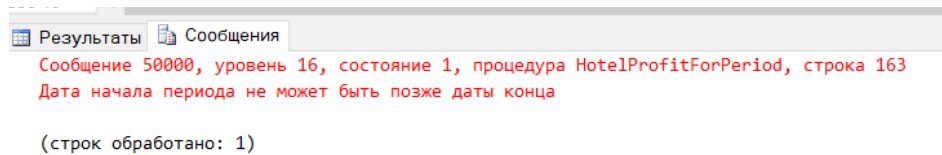


	ID отеля	Прибыль
1	1	12200

## Пример 2.

```
DECLARE @HotelProfit INT
```

```
EXEC @HotelProfit = HotelProfitForPeriod @BeginDate = '2022-04-01', @EndDate =  
'2022-03-31', @HotelID = 1  
SELECT 'ID отеля' = 1, 'Прибыль' = @HotelProfit
```



## 6. Вывод прибыли для каждой гостиницы за указанный период с указанием города и страны, в которой она находится

На вход подаются дата начала периода и дата его конца. В цикле для каждого ID отеля вызывается процедура для расчета прибыли заданной гостиницы за заданный период. Если дата начала периода позже даты конца выводится ошибка.

### Код.

```
CREATE PROCEDURE EveryHotelProfitForPeriod  
@BeginDate date, @EndDate date  
AS  
BEGIN  
IF (@BeginDate > @EndDate)  
RAISERROR ('Дата начала периода не может быть позже даты конца', 16, 1)  
ELSE  
BEGIN  
DECLARE @Profit INT = 0, @AccommodationID INT, @CurrentBill INT, @HotelID INT,  
@CityName varchar(255), @CountryName varchar(255)  
DECLARE myCursor2 CURSOR FOR  
SELECT e.ID, d.Name City, p.Name Country  
FROM Hotel AS e  
LEFT JOIN City AS d ON d.ID=e.CityID  
LEFT JOIN Country AS p ON p.ID=d.CountryID  
OPEN myCursor2  
FETCH NEXT FROM myCursor2 INTO @HotelID, @CityName, @CountryName  
WHILE @@FETCH_STATUS = 0  
BEGIN  
EXEC @Profit = HotelProfitForPeriod @BeginDate = @BeginDate, @EndDate =  
@EndDate, @HotelID = @HotelID  
SELECT HotelID = @HotelID, Profit = @Profit, City = @CityName, Country =  
@CountryName  
FETCH NEXT FROM myCursor2 INTO @HotelID, @CityName, @CountryName  
END
```



```
CLOSE myCursor2
DEALLOCATE myCursor2
END
END
```

### Пример 1.

EXEC EveryHotelProfitForPeriod @BeginDate = '2022-03-01', @EndDate = '2022-03-31'

	HotelID	Profit	City	Country
1	1	12200	Москва	Россия

	HotelID	Profit	City	Country
1	2	7000	Анталья	Турция

	HotelID	Profit	City	Country
1	3	0	Сочи	Россия

	HotelID	Profit	City	Country
1	4	6000	Сочи	Россия

### Пример 2.

EXEC EveryHotelProfitForPeriod @BeginDate = '2022-04-01', @EndDate = '2022-03-31'

Сообщение 50000, уровень 16, состояние 1, процедура EveryHotelProfitForPeriod, строка 197  
Дата начала периода не может быть позже даты конца