**Представления для просмотра данных:**

1. Список корпусов гостиницы.
2. Перечень номеров с указанием корпуса и характеристик.
3. Бронь на будущее время.
4. Список истории проживания.
5. Список дополнительных услуг с указанием цены.
6. Список клиентов с общим количеством дней проживания, общей стоимостью проживания, датой окончания последнего проживания.
7. Сделать отбор Топ 5 клиентов / Топ 5 партнеров с количеством проживаний и общим количеством дней проживания.
8. Список партнеров с общим количеством дней проживания, общей стоимостью проживания, датой окончания последнего проживания (для всех сотрудников партнера).
9. Список клиентов-должников.
10. Перечень жалоб.

**Представления со смысловой нагрузкой:**

1. Список номеров с количеством проживаний и общим количеством дней проживания
2. Сделать отборы по корпусу, ТОП 3 лучших номеров (наиболее посещаемых).
3. Список клиентов, ни разу не воспользовавшимися дополнительными услугами.
4. Список клиентов, пользовавшихся дополнительными услугами.
5. Список клиентов, ни разу не оставлявших жалобы.
6. Список клиентов, оставивших хотя бы одну жалобу.

**Функции:**

1. Получить перечень и общее число клиентов, заселявшихся в номера с указанными характеристиками за некоторый период.
2. Получить список жалоб за период: всех, на проживание в определенном корпусе, от определенного клиента, по звездности корпуса.
3. Получить список и количество свободных номеров и количество свободных мест на данный момент, на определенную дату, для номеров с определенными характеристиками.
4. Получить сведения о количестве свободных номеров: по корпусам; по звездности; по количеству комнат; по количеству мест в номере.
5. Вывести сведения о занятости и свободном времени для указанного номера. (пока без свободного времени)
6. Получить список занятых сейчас номеров, которые освобождаются к указанному сроку. (Добавить заезд и выезд текущего клиента в rooms)
7. Получить данные об объеме бронирования номеров конкретным партнером за указанный период: в целом и с разбивкой по звездности и номерам, с указанием характеристик номеров.
8. Получить сведения о партнерах, с которыми заключены договора о брони на указанный период.
9. Получить сведения о новых клиентах за указанный период. (Добавить дату регистрации клиента)
10. Получить сведения о конкретном клиенте гостиницы: сколько раз он посещал гостиницу, в каких номерах и в какой период останавливался, какими дополнительными услугами пользовался.
11. Получить сведения о конкретном номере: кем он был занят в определенный период.
12. Получить процентное отношение всех номеров к номерам, бронируемым партнерами, за определенный период.
13. Вывести график работы конкретного сотрудника.
14. Вывести все брони, оформленные указанным сотрудником.
15. Получить данные о рентабельности номеров с определенными характеристиками: соотношение об объеме продаж номеров к накладным расходам за указанный период.
16. С разбивкой по сотрудникам получить количество отработанных дней и часов за определенный период.

**Процедуры:**

1. Установить график работы сотруднику на неделю. (alter table set)
2. Увеличить цены всех услуг на заданное количество процентов.

**Триггеры:**

1. Изменять статус номера при вселении и выселении клиента.
2. Автоматически изменять статус номера при создании / отмене брони и въезде / выезде постояльцев.
3. Автоматически изменять поле “Задолженность” при внесении платежа.

**Виды запросов в информационной системе:**

1. Получить перечень и общее число фирм, забронировавших места в объеме, не менее указанного, за весь период сотрудничества, либо за некоторый период.
2. Получить перечень и общее число постояльцев, заселявшихся в номера с указанными характеристиками за некоторый период.
3. Получить количество свободных номеров на данный момент.
4. Получить сведения о количестве свободных номеров с указанными характеристиками.
5. Получить сведения о конкретном свободном номере: в течение какого времени он будет пустовать и о его характеристиках.
6. Получить список занятых сейчас номеров, которые освобождаются к указанному сроку.
7. Получить данные об объеме бронирования номеров данной фирмой за указанный период, и каким номерам отдавались предпочтения.
8. Получить список недовольных клиентов и их жалобы.
9. Получить данные о рентабельности номеров с определенными характеристиками: соотношение об объеме продаж номеров к накладным расходам за указанный период.
10. Получить сведения о постояльце из заданного номера: его счет гостинице за дополнительные услуги, поступавшие от него жалобы, виды дополнительных услуг, которыми он пользовался.
11. Получить сведения о фирмах, с которыми заключены договора о брони на указанный период.
12. Получить сведения о наиболее часто посещающих гостиницу постояльцах по [всем корпусам гостиниц](http://topuch.com/lekcii-1-klassifikaciya-gostinic-za-rubejom/index.html), по определенному зданию.
13. Получить сведения о новых клиентах за указанный период.
14. Получить сведения о конкретном человеке, сколько раз он посещал гостиницу, в каких номерах и в какой период останавливался, какие счета оплачивал.
15. Получить сведения о конкретном номере: кем он был занят в определенный период.
16. Получить процентное отношение всех номеров к номерам, бронируемым партнерами.

**Представления для просмотра данных:**

1. select \* from hotel
2. select \* from rooms
3. select \* from reservations

where in\_data > now()

1. select \* from history
2. select \* from services
3. SELECT c.id,surname, name, patronymic, SUM(check\_out\_data - (check\_in\_data-1)) as days,

Sum(payment) FROM clients c

left join history h on c.id=h.client\_id

group by c.id, surname, name, patronymic, client\_id

order by id

1. CREATE TEMPORARY TABLE temp\_table AS (SELECT CONCAT(surname, ' ', name, ' ', patronymic) AS "Top 5 partners/clients", COUNT(check\_in\_data) AS check\_ins, SUM(check\_out\_data - (check\_in\_data-1)) as days FROM clients c LEFT JOIN history h ON c.id=h.client\_id GROUP BY c.id, client\_id HAVING SUM(check\_out\_data - (check\_in\_data-1)) > 0 ORDER BY days DESC LIMIT 5)

UNION

(SELECT organization::text, COUNT(check\_in\_data) AS check\_ins, SUM(check\_out\_data - (check\_in\_data-1)) as days FROM partners p LEFT JOIN history h ON p.id=h.partner\_id GROUP BY organization HAVING SUM(check\_out\_data - (check\_in\_data-1)) > 0 ORDER BY days DESC LIMIT 5);

SELECT \* FROM temp\_table;

1. SELECT organization::text, SUM(check\_out\_data - (check\_in\_data-1)) as days, Sum(payment), (select check\_out\_data) as "last day" FROM partners p

LEFT JOIN history h ON p.id=h.partner\_id

GROUP BY organization,check\_out\_data

HAVING SUM(check\_out\_data - (check\_in\_data-1)) > 0

ORDER BY days

1. select \* from clients

where id in (select client\_id from history where debt - payment > 0)

1. select \* from complaints\_list

**Представления со смысловой нагрузкой:**

1. SELECT room\_number, COUNT(check\_in\_data) AS number\_of\_stays, SUM(check\_out\_data - (check\_in\_data - 1)) AS total\_stay\_days FROM history

GROUP BY room\_number

order by room\_number

1. --------------------
2. select CONCAT(surname, ' ', name, ' ', patronymic) AS clients from clients

where id in (select client\_id from history

where id not in (select history\_id from services\_list group by history\_id))

1. select CONCAT(surname, ' ', name, ' ', patronymic) AS clients from clients

where id in (select client\_id from history

where id in (select history\_id from services\_list group by history\_id))

1. select CONCAT(surname, ' ', name, ' ', patronymic) AS clients from clients

where id in (select client\_id from history

where id not in (select history\_id from complaints\_list group by history\_id))

1. select CONCAT(surname, ' ', name, ' ', patronymic) AS clients from clients

where id in (select client\_id from history

where id not in (select history\_id from complaints\_list group by history\_id))

**Функции:**

1. CREATE OR REPLACE FUNCTION Rooms\_info(start\_date DATE, end\_date DATE, room\_size INT, room\_build INT)

RETURNS varchar (100)

AS $$

DECLARE

curs1 cursor for SELECT CONCAT(surname, ' ', name, ' ', patronymic) FROM history h

JOIN rooms r ON h.room\_number = r.id

JOIN clients c ON c.id = h.client\_id

WHERE h.check\_in\_data BETWEEN '2024-07-01' AND '2024-07-08' and

r.capacity = 1 and

r.building = 3

group by h.client\_id, surname, name, patronymic;

quantity INT := 0;

cl varchar(50);

BEGIN

open curs1;

fetch curs1 into cl;

loop

EXIT WHEN NOT FOUND;

RAISE NOTICE '%', cl;

quantity:= quantity + 1;

fetch curs1 into cl;

end loop;

RAISE NOTICE 'Количество клиентов заселявшихся в номера с указанными характеристиками: %', quantity;

close curs1;

END;

$$ LANGUAGE plpgsql;

select \* from Rooms\_info('2024-07-01','2024-07-08',1,3)

5. CREATE OR REPLACE FUNCTION rooms\_get\_free (data DATE)

RETURNS TABLE (

room\_number INT

)

AS $$

BEGIN

RETURN QUERY

SELECT id FROM rooms

WHERE out\_data <= data and booking\_id is null or (select out\_data from reservations where id = booking\_data) <= data;

END;

$$ LANGUAGE plpgsql;

select \* from partner\_booking('2024-07-10','2024-07-20')

2. CREATE OR REPLACE FUNCTION partner\_booking(start\_date DATE, end\_date DATE)

RETURNS TABLE (

partner\_name VARCHAR(100),

contract\_number VARCHAR(100),

booking\_start\_date DATE,

booking\_end\_date DATE)

AS $$

BEGIN

RETURN QUERY

SELECT p.organization, p.email, r.in\_data, r.out\_data FROM reservations r

JOIN partners p ON r.partner\_id = p.id

WHERE r.in\_data >= start\_date AND r.in\_data <= end\_date;

END;

$$ LANGUAGE plpgsql;

select \* from partner\_booking('2024-07-10','2024-07-20')

3. CREATE OR REPLACE FUNCTION concerned\_room (nomer INT, start\_data DATE, end\_data DATE)

RETURNS TABLE (

client VARCHAR(100))

AS $$

BEGIN

RETURN QUERY

SELECT CONCAT(surname, ' ', name, ' ', patronymic) AS client FROM from clients c

JOIN history h ON c.id = h.client\_id

WHERE h. room\_number = nomer and h. check\_in\_data >= start\_data AND h.check\_out\_data <= end\_data;

END;

$$ LANGUAGE plpgsql;

select \* from concerned\_room (3101, '2024-07-10', '2024-07-20')



**Процедуры:**



**Триггеры:**