

1. Среднее количество ночей

```
0: jdbc:hive2://localhost:10000> select year, month, round(avg(week_nights + weekend_nights), 2) as avg_nights from logs_hotel where status = 'Not_Canceled' group by year, month order by year, month;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+-----+-----+-----+
| year | month | avg_nights |
+-----+-----+-----+
| 2017 | 7     | 3.02       |
| 2017 | 8     | 2.72       |
| 2017 | 9     | 2.66       |
| 2017 | 10    | 2.7        |
| 2017 | 11    | 2.72       |
| 2017 | 12    | 3.04       |
| 2018 | 1     | 2.74       |
| 2018 | 2     | 2.69       |
| 2018 | 3     | 3.04       |
| 2018 | 4     | 2.92       |
| 2018 | 5     | 2.81       |
| 2018 | 6     | 2.6        |
| 2018 | 7     | 3.19       |
| 2018 | 8     | 3.15       |
| 2018 | 9     | 2.79       |
| 2018 | 10    | 2.89       |
| 2018 | 11    | 2.98       |
| 2018 | 12    | 3.25       |
+-----+-----+-----+
18 rows selected (4.862 seconds)
0: jdbc:hive2://localhost:10000>
```

```
select year, month, round(avg(week_nights + weekend_nights), 2) as avg_nights
from logs_hotel
where status = 'Not_Canceled'
group by year, month
order by year, month;
```

2. ТОП-3 месяца по отменам

```
0: jdbc:hive2://localhost:10000> with monthly_bookings as (select month, count(*) as total_bookings, sum(case when status = 'Canceled' then 1 else 0 end) as canceled_bookings from logs_hotel where year = 2018 group by month) select month, round((canceled_bookings * 100.0 / total_bookings), 2) as percent_cancellations from monthly_bookings order by percent_cancellations desc limit 3;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+-----+-----+
| month | percent_cancellations |
+-----+-----+
| 8     | 46.55                 |
| 10    | 46.36                 |
| 9     | 45.78                 |
+-----+-----+
3 rows selected (3.642 seconds)
```

```
with monthly_bookings as (
  select month, count(*) as total_bookings, sum(case when status = 'Canceled' then 1
  else 0 end) as canceled_bookings
  from logs_hotel where year = 2018 group by month
)
select month, round((canceled_bookings * 100.0 / total_bookings), 2) as
percent_cancellations
from monthly_bookings
order by percent_cancellations desc
limit 3;
```

3. Ср. время м/у бронирование м и заездом

```
0: jdbc:hive2://localhost:10000> select year, month, round(avg(lead_time), 2) as avg_lead_time_days from logs_hotel where status = 'Not_Canceled' group by year, month order by year, month;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+-----+-----+-----+
| year | month | avg_lead_time_days |
+-----+-----+-----+
| 2017 | 7     | 130.73             |
| 2017 | 8     | 35.08              |
| 2017 | 9     | 51.72              |
| 2017 | 10    | 55.89              |
| 2017 | 11    | 33.28              |
| 2017 | 12    | 46.75              |
| 2018 | 1     | 34.87              |
| 2018 | 2     | 30.53              |
| 2018 | 3     | 43.19              |
| 2018 | 4     | 62.49              |
| 2018 | 5     | 60.99              |
| 2018 | 6     | 70.64              |
| 2018 | 7     | 86.88              |
| 2018 | 8     | 83.09              |
| 2018 | 9     | 63.32              |
| 2018 | 10    | 73.24              |
| 2018 | 11    | 44.25              |
| 2018 | 12    | 69.75              |
+-----+-----+-----+
18 rows selected (3.211 seconds)
0: jdbc:hive2://localhost:10000>
```

```
select year, month, round(avg(lead_time), 2) as avg_lead_time_days
```

from logs_hotel

where status = 'Not_Canceled'

group by year, month

order by year, month;

4.1. Общая
средняя
выручка

10 rows selected (3.211 seconds)

0: jdbc:hive2://localhost:10000> select year, month, round(avg(case when market_segment = 'Online' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_online, round(avg(case when market_segment = 'Offline' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_offline, round(avg(case when market_segment = 'Aviation' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_aviation, round(avg(case when market_segment = 'Corporate' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_corporate, round(avg(case when market_segment = 'Complementary' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_complementary from logs_hotel where status = 'Not_Canceled' group by year, month order by year, month;

WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.

year	month	avg_revenue_online	avg_revenue_offline	avg_revenue_aviation	avg_revenue_corporate	avg_revenue_complementary
2017	7	290.56	228.95	NULL	113.75	22.4
2017	8	284.21	235.54	NULL	156.42	0.32
2017	9	348.55	236.65	NULL	177.83	16.89
2017	10	311.47	223.24	NULL	180.26	1.09
2017	11	240.52	198.36	NULL	102.97	14.81
2017	12	258.93	253.86	NULL	141.11	0.25
2018	1	236.09	210.51	NULL	113.03	2.27
2018	2	238.07	251.85	352.0	115.06	1.39
2018	3	301.71	233.39	118.33	142.39	38.17
2018	4	320.08	236.44	321.81	108.42	0.0

select year, month, round(avg(case when market_segment = 'Online' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_online, round(avg(case when market_segment = 'Offline' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_offline, round(avg(case when market_segment = 'Aviation' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_aviation, round(avg(case when market_segment = 'Corporate' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_corporate, round(avg(case when market_segment = 'Complementary' then avg_room_price * (week_nights + weekend_nights) else null end), 2) as avg_revenue_complementary

from logs_hotel

where status = 'Not_Canceled'

group by year, month

order by year, month;

4.2.
окончание
таблицы

2017	10	311.47	223.24	NULL	180.26	1.09
2017	11	240.52	198.36	NULL	102.97	14.81
2017	12	258.93	253.86	NULL	141.11	0.25
2018	1	236.09	210.51	NULL	113.03	2.27
2018	2	238.07	251.85	352.0	115.06	1.39
2018	3	301.71	233.39	118.33	142.39	38.17
2018	4	320.08	236.44	321.81	108.42	0.0
2018	5	352.34	274.55	262.5	229.5	0.0
2018	6	335.03	251.98	247.0	148.13	0.0
2018	7	390.05	310.36	79.0	146.99	5.38
2018	8	391.22	318.36	NULL	206.75	0.0
2018	9	367.61	319.4	361.67	201.12	21.92
2018	10	358.81	276.64	313.04	145.33	0.26
2018	11	332.04	226.33	159.75	153.9	8.39
2018	12	349.46	254.97	NULL	138.07	4.0
-----+						
18 rows selected (5.042 seconds)						
0: jdbc:hive2://localhost:10000>						

5. Топ-5
постоянных
гостей

```
0: jdbc:hive2://localhost:10000> with guest_revenue as (select id, sum(avg_room_price * (week_nights + weekend_nights)) as total_revenue from logs_hotel where status = 'Not_Canceled' and repeated_guest = 1 group by id) select id, total_revenue, round((total_revenue * 100.0 / sum(total_revenue) over ()), 2) as revenue_percent from guest_revenue order by total_revenue desc limit 5;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
```

id	total_revenue	revenue_percent
INN19235	1754.3999999999999	1.55
INN05222	690.0	0.61
INN14189	665.0	0.59
INN09923	660.0	0.58
INN25479	650.0	0.57

```
5 rows selected (5.199 seconds)
```

with guest_revenue as (
 select id, sum(avg_room_price * (week_nights + weekend_nights)) as total_revenue
 from logs_hotel **where** status = 'Not_Canceled' **and** repeated_guest = 1 **group by** id
)
select id, total_revenue, round((total_revenue * 100.0 / sum(total_revenue) over ()), 2)
as revenue_percent
from guest_revenue
order by total_revenue desc
limit 5;

6.1. Общее
кол-во гостей
в день и
загрузка
отеля

```
0: jdbc:hive2://localhost:10000> with logs_prepared as (select *, cast(cast(year as string) || '-' || (case when month < 10 then '0' || cast(month as string) else cast(month as string) end) || '-' || (case when 'date' < 10 then '0' || cast('date' as string) else cast('date' as string) end) as string) as full_date from logs_hotel where status = 'Not_Canceled') select c.calendar_dt, coalesce(sum(lp.n_adults + lp.n_children), 0) as guests_count, round(coalesce(sum(lp.n_adults + lp.n_children), 0) * 100.0 / 400, 2) as occupancy from calendar c left join logs_prepared lp on c.calendar_dt = lp.full_date group by c.calendar_dt order by c.calendar_dt desc;
```

WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.

SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Execution log at: /tmp/root/20251208213417_e2c1233a-6ede-4595-9742-3a34f8ff80ff.log

2025-12-08 21:34:23 Starting to launch local task to process map join; maximum memory = 477626368

2025-12-08 21:34:24 Dump the side-table for tag: 1 with group count: 1 into file: file:/tmp/root/47d3c0a9-6d46-47f2-9de9-902a17b099cb/hive_2025-12-08_21-34-17_736_7054566840503966050-1/-local-10006/HashTable-Stage-2/MapJoin-mapfile01--.hashtable

2025-12-08 21:34:25 Uploaded 1 File to: file:/tmp/root/47d3c0a9-6d46-47f2-9de9-902a17b099cb/hive_2025-12-08_21-34-17_736_7054566840503966050-1/-local-10006/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (171836 bytes)

2025-12-08 21:34:25 End of local task; Time Taken: 1.74 sec.

c.calendar_dt	guests_count	occupancy
2018-12-31	0	0.00
2018-12-30	0	0.00
2018-12-29	0	0.00
2018-12-28	0	0.00
2018-12-27	0	0.00
2018-12-26	0	0.00
2018-12-25	0	0.00
2018-12-24	0	0.00
2018-12-23	0	0.00
2018-12-22	0	0.00
2018-12-21	0	0.00

with logs_prepared as (
 select *, cast(cast(year as string) || '-' || (case when month < 10 then '0' || cast(month as string) else cast(month as string) end) || '-' || (case when 'date' < 10 then '0' || cast('date' as string) else cast('date' as string) end) as string) as full_date
 from logs_hotel **where** status = 'Not_Canceled')
 select c.calendar_dt, coalesce(sum(lp.n_adults + lp.n_children), 0) as guests_count,
 round(coalesce(sum(lp.n_adults + lp.n_children), 0) * 100.0 / 400, 2) as occupancy
 from calendar c **left join** logs_prepared lp **on** c.calendar_dt = lp.full_date
 group by c.calendar_dt
 order by c.calendar_dt desc;

```

select *, cast(cast(year as string) || '-' ||
(case when month < 10 then '0' || cast(month as string) else cast(month as string) end)
|| '-' ||
(case when 'date' < 10 then '0' || cast('date' as string) else cast('date' as string) end) as
string) as full_date
from logs_hotel
where status = 'Not_Canceled'
)
select c.calendar_dt, coalesce(sum(lp.n_adults + lp.n_children), 0) as guests_count,
round(coalesce(sum(lp.n_adults + lp.n_children), 0) * 100.0 / 400, 2) as occupancy
from calendar c
left join logs_prepared lp on c.calendar_dt = lp.full_date
group by c.calendar_dt
order by c.calendar_dt desc;

```

6.2.
окончание
таблицы

c.calendar_dt	guests_count	occupancy
2017-01-30	0	0.00
2017-01-29	0	0.00
2017-01-28	0	0.00
2017-01-27	0	0.00
2017-01-26	0	0.00
2017-01-25	0	0.00
2017-01-24	0	0.00
2017-01-23	0	0.00
2017-01-22	0	0.00
2017-01-21	0	0.00
2017-01-20	0	0.00
2017-01-19	0	0.00
2017-01-18	0	0.00
2017-01-17	0	0.00
2017-01-16	0	0.00
2017-01-15	0	0.00
2017-01-14	0	0.00
2017-01-13	0	0.00
2017-01-12	0	0.00
2017-01-11	0	0.00
2017-01-10	0	0.00
2017-01-09	0	0.00
2017-01-08	0	0.00
2017-01-07	0	0.00
2017-01-06	0	0.00
2017-01-05	0	0.00
2017-01-04	0	0.00
2017-01-03	0	0.00
2017-01-02	0	0.00
2017-01-01	0	0.00

730 rows selected (10.708 seconds)
0: jdbc:hive2://localhost:10000> |