```
create database hotel_project;
use hotel_project;
select * from data2018;
select * from data2019;
select * from data2020;

-- appending all 3 tables using union operator and creating it as hotel table
create table hotel as
select * from data2018
union
select * from data2019
union
select * from data2020;

select * from hotel;

-- changing datatype of reservation status date column to date
UPDATE hotel SET reservation_status_date =
STR_TO_DATE(reservation_status_date, '%d-%m-%Y');

-- handling null values and replacing them with NA in certain columns
update hotel set agent = 'NA' where agent is null;
update hotel set company = 'NA' where company is null;
```

booking_changes	deposit_type	agent	company	days_in_waiting_list	customer_type	adr
0	No Deposit	240	NA	0	Transient	82
0	No Deposit	15	NA	0	Transient	105.5
0	No Deposit	240	NA	0	Transient	123
0	No Deposit	240	NA	0	Transient	107
0	No Deposit	NA	NA	0	Transient	108.3
0	No Deposit	241	NA	0	Transient	108.8
0	No Deposit	241	NA	0	Transient	108.8
0	No Deposit	241	NA	0	Transient	108.8

-- To see if hotel's revenue increased year by yearS

-- For this, calculating cost of revenue of each visitor for each day

-- By multiplying daily rate with stays in weekend and week nights of visitors

```
select hotel,arrival_date_year,
round(sum((stays_in_weekend_nights+stays_in_week_nights)*adr),2) as
revenue

from hotel group by arrival_date_year

,hotel order by 1;
```

	hotel	arrival_date_year	revenue
▶	City Hotel	2018	1764060.07
	City Hotel	2019	9365478.77
	City Hotel	2020	6265358.52
	Resort Hotel	2018	3120849.49
	Resort Hotel	2019	9308524.65
	Resort Hotel	2020	6225393.01

-- By what % did hotel's revenue increased day by day?

```
with cte as(select hotel,arrival_date_year,
round(sum((stays_in_weekend_nights+stays_in_week_nights)*adr),2) as
revenue

from hotel group by arrival_date_year
```

,hotel order by 1),

cte2 as(

select hotel, arrival_date_year, revenue, lead(revenue) over (partition by hotel order by arrival_date_year) as next_year_rev

from cte)

select hotel, arrival_date_year, revenue, next_year_rev , round(((next_year_rev-revenue)/revenue)*100,2) as per_change from cte2

where next_year_rev is not null;

hotel	arrival_date_year	revenue	next_year_rev	per_change
City Hotel	2018	1764060.07	9365478.77	430.9
City Hotel	2019	9365478.77	6265358.52	-33.1
Resort Hotel	2018	3120849.49	9308524.65	198.27
Resort Hotel	2019	9308524.65	6225393.01	-33.12

-- market_segment table

select * from market_segment;

Discount	market_segment
0	Undefined
0.1	Direct
0.1	Groups
0.15	Corporate
0.2	Aviation
0.3	Offline TA/TO
0.3	Online TA
1	Complementary

-- meal table

select * from meals;

Cost	meal
0	Undefined
12.99	BB
17.99	HB
21.99	FB
35	SC

-- Joining meal and market_segment tables with hotel table to see discount on market_segment and cost of each meal in hotel data

```
select * from hotel left join market_segment on
hotel.market_segment=market_segment.market_segment left join meals on
hotel.meal=meals.meal;
```

waiting_list	customer_type	adr	required_car_parking_spaces	total_of_special_requests	reservation_status	reservation_status_date	Discount	market_segment	Cost	meal
	Transient	82	0	1	Canceled	2018-05-06	0.3	Online TA	12.99	BB
	Transient	105.5	0	0	Canceled	2018-04-22	0.3	Offline TA/TO	17.99	HB
	Transient	123	0	0	Canceled	2018-06-23	0.3	Online TA	12.99	BB
	Transient	107	0	2	Canceled	2018-05-11	0.3	Online TA	12.99	BB
	Transient	108.3	0	2	Canceled	2018-05-29	0.1	Direct	12.99	BB
	Transient	108.8	0	1	Canceled	2018-05-19	0.3	Online TA	12.99	BB
	Transient	108.8	0	1	Canceled	2018-06-19	0.3	Online TA	12.99	BB
	Transient	108.8	0	0	Canceled	2018-05-23	0.3	Online TA	12.99	BB
	Transient	117.81	0	0	Canceled	2018-05-18	0.3	Online TA	12.99	BB

-- Calculating revenue after discount of each visitor for each day

-- By multiplying daily rate with stays in weekend and week nights of visitors and discount column

```
select *,
round(((stays_in_weekend_nights+stays_in_week_nights)*adr*Discount),2) as
revenue

 from hotel left join market_segment on
hotel.market_segment=market_segment.market_segment left join meals on
hotel.meal=meals.meal;
```

customer_type	adr	required_car_parking_spaces	total_of_special_requests	reservation_status	reservation_status_date	Discount	market_segment	Cost	meal	revenue
Transient	82	0	1	Canceled	2018-05-06	0.3	Online TA	12.99	BB	73.8
Transient	105.5	0	0	Canceled	2018-04-22	0.3	Offline TA/TO	17.99	HB	94.95
Transient	123	0	0	Canceled	2018-06-23	0.3	Online TA	12.99	BB	147.6
Transient	107	0	2	Canceled	2018-05-11	0.3	Online TA	12.99	BB	224.7
Transient	108.3	0	2	Canceled	2018-05-29	0.1	Direct	12.99	BB	108.3
Transient	108.8	0	1	Canceled	2018-05-19	0.3	Online TA	12.99	BB	130.56
Transient	108.8	0	1	Canceled	2018-06-19	0.3	Online TA	12.99	BB	130.56
Transient	108.8	0	0	Canceled	2018-05-23	0.3	Online TA	12.99	BB	130.56

-- Booking Analysis by Year:

-- Count the number of bookings and cancellations for each year

```
SELECT arrival_date_year,
    COUNT(*) AS total_bookings,
    SUM(is_canceled) AS cancellations
FROM hotel
GROUP BY arrival_date_year;
```

arrival_date_year	total_bookings	cancellations
2018	13309	2700
2019	49909	13753
2020	27554	9824

-- Calculate the average lead time for bookings

```
SELECT AVG(lead_time) AS avg_lead_time
FROM hotel;
```

avg_lead_time
77.9277

-- Customer Segment Analysis:

-- Analyze the distribution of bookings across different customer types

```
SELECT customer_type, COUNT(*) AS bookings
FROM hotel
GROUP BY customer_type;
```

customer_type	bookings
Transient	73832
Contract	3951
Transient-Party	12398
Group	591

```
-- Calculate the average daily rate for each customer type
SELECT customer_type, round(AVG(adr),2) AS avg_daily_rate
FROM hotel
GROUP BY customer_type;
```

customer_type	avg_daily_rate
Transient	108.33
Contract	91.5
Transient-Party	85.09
Group	82.2

```
-- Room Type Analysis:
-- Examine the distribution of reserved and assigned room types
SELECT reserved_room_type, assigned_room_type, COUNT(*) AS bookings
FROM hotel
GROUP BY reserved_room_type, assigned_room_type;
```

	reserved_room_type	assigned_room_type	bookings
▶	A	A	47016
	D	D	16178
	E	E	5957
	G	G	2148
	F	F	2663
	H	H	644
	C	C	1000
	L	C	2
	A	D	6822

```
-- Calculate the percentage of bookings with room changes
SELECT SUM(booking_changes) / COUNT(*) * 100 AS
percentage_with_changes
FROM hotel;
```

	percentage_with_changes
▶	26.6448

-- Cancellation Analysis:

```
-- Calculate the cancellation rate
SELECT AVG(is_canceled) * 100 AS cancellation_rate
FROM hotel;
```

	cancellation_rate
▶	28.9484

-- Geographical Analysis:

```
-- Identify the top countries of origin for guests
SELECT country, COUNT(*) AS bookings
FROM hotel
GROUP BY country
```

```
ORDER BY bookings DESC
LIMIT 10;
```

country	bookings
PRT	31256
GBR	10753
FRA	8551
ESP	7660
DEU	5015
IRL	3253
ITA	3006
BRA	1942
BEL	1927
NLD	1855

```
-- Analyze booking patterns based on the country of origin
SELECT country, arrival_date_month, COUNT(*) AS bookings
FROM hotel
GROUP BY country, arrival_date_month;
```

country	arrival_date_month	bookings
PRT	July	3970
PRT	August	4423
PRT	September	2897
IRL	September	419
ESP	September	547
DEU	September	443
PRT	October	2865
CN	October	83
ESP	October	564
PRT	November	2303