SQL QUERIES FOR HOSPITALITY PROJECT

```
USE PROJECT;
select * from dim_date;
select * FROM dim_hotels;
select * from dim_rooms;
select  * from fact_bookings;
select * from fact_aggbookings;
-------------------------------------------------------------
/* REVENUE */
CREATE VIEW REVENUE AS
SELECT CONCAT(SUM(revenue_realized)," Rs.") AS REVENUE FROM FACT_BOOKINGS;
-------------------------------------------------------------
/* TOTAL BOOKINGS */
CREATE VIEW TOTAL_BOOKINGS AS
SELECT BOOKING_STATUS,COUNT(booking_id) AS TOTAL_BOOKINGS FROM
FACT_BOOKINGS
GROUP BY BOOKING_STATUS;

SELECT BOOKING_STATUS,COUNT(booking_id) FROM FACT_BOOKINGS
GROUP BY BOOKING_STATUS;

-------------------------------------------------------------
/* TOTAL CAPACITY */

CREATE VIEW TOTAL_CAPACITY AS
SELECT SUM(CAPACITY) AS TOTAL_CAPACITY FROM FACT_AGGBOOKINGS;

-------------------------------------------------------------
/* AVERAGE RATING*/

CREATE VIEW AVERAGE_RATING AS
SELECT ROUND(AVG(ratings_given),2) AS AVERAGE_RATING FROM FACT_BOOKINGS
WHERE ratings_given <> '' AND RATINGS_GIVEN IS NOT NULL;
-------------------------------------------------------------
/* NO. OF DAYS*/

CREATE VIEW NUMBER_OF_DAYS AS
SELECT DISTINCT(COUNT(DATE)) AS NUMBER_OF_DAYS FROM DIM_DATE;

-------------------------------------------------------------
/* Total cancelled bookings */
select * from fact_bookings;

CREATE VIEW TOTAL_CANCELLED_BOOKINGS AS
SELECT COUNT(booking_status) AS TOTAL_CANCELLED_BOOKINGS FROM
FACT_BOOKINGS
WHERE booking_status="Cancelled";

-------------------------------------------------------------
```

```
/* Total CHECKED OUT */
CREATE VIEW TOTAL_CHECKED_OUTS AS
SELECT COUNT(booking_status) AS TOTAL_CHECKED_OUT FROM FACT_BOOKINGS
WHERE booking_status="Checked Out";

------------------------------------------------------------
/* Total NO SHOW */
CREATE VIEW TOTAL_NO_SHOWS AS
SELECT COUNT(booking_status) AS TOTAL_NO_SHOW FROM FACT_BOOKINGS
WHERE booking_status="NO SHOW";

------------------------------------------------------------

/* CANCELLATION RATE */
CREATE VIEW CANCELLATION_RATE AS
with ctec as(
SELECT COUNT(booking_status) AS TOTC FROM FACT_BOOKINGS
WHERE BOOKING_STATUS="CANCELLED"
), CTET AS(
SELECT COUNT(BOOKING_ID) AS TOT FROM FACT_BOOKINGS
)
SELECT CONCAT(ROUND((SELECT * FROM CTEC) /(SELECT * FROM CTET)*100,2)," %")
AS CANCELLATION_RATE;

------------------------------------------------------------
/* NO SHOW RATE*/
CREATE VIEW NO_SHOW_RATE AS
with cteN as(
SELECT COUNT(booking_status) AS TOTC FROM FACT_BOOKINGS
WHERE BOOKING_STATUS="NO SHOW"
), CTENT AS(
SELECT COUNT(BOOKING_ID) AS TOT FROM FACT_BOOKINGS
)
SELECT CONCAT(ROUND((SELECT * FROM CTEN) /(SELECT * FROM CTENT)*100,2),"
%") AS NOSHOW_RATE;

------------------------------------------------------------
/* ADR */
CREATE VIEW ADR AS
WITH REV AS(
SELECT SUM(revenue_realized) AS REVENUE FROM FACT_BOOKINGS
), TB AS(
SELECT COUNT(booking_id) AS TOTAL_BOOKINGS FROM FACT_BOOKINGS
)
SELECT CONCAT(ROUND((SELECT * FROM REV)/(SELECT * FROM TB),0),"  RS") AS
ADR;


------------------------------------------------------------
/* REVPAR */
```

```
CREATE VIEW REV_PAR AS
WITH REVPAR AS(
SELECT SUM(revenue_realized) AS R FROM FACT_BOOKINGS
), TC AS(
SELECT SUM(CAPACITY) AS C FROM FACT_AGGBOOKINGS
)
SELECT CONCAT(ROUND((SELECT * FROM REVPAR)/(SELECT * FROM TC),0)," RS") AS
REVPAR;

------------------------------------------------------------
/*REALISATION RATE */
DROP VIEW REALISATION_RATE;
CREATE VIEW REALISATION_RATE AS
with CTECHOUT as(
SELECT COUNT(booking_status) AS TOTC FROM FACT_BOOKINGS
WHERE BOOKING_STATUS="CHECKED OUT"
), CTETOTAL AS(
SELECT COUNT(BOOKING_ID) AS TOT FROM FACT_BOOKINGS
)
SELECT CONCAT(ROUND((SELECT * FROM CTECHOUT) /(SELECT * FROM
CTETOTAL)*100,2)," %") AS REALISATION_RATE;

------------------------------------------------------------
SELECT COUNT(DISTINCT (BOOKING_PLATFORM)) AS BOKKING_PLATFORMS FROM
FACT_BOOKINGS;
select BOOKING_PLATFORM,COUNT(BOOKING_ID) AS BOOKINGS,
COUNT(BOOKING_ID) OVER(ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) AS TOTAL from fact_bookings
GROUP BY BOOKING_PLATFORM;

 ------------------------------------------------------------
 /* BOOKINGS % BY BOOKING_PLATFORMS */
 CREATE VIEW PERCENTAGE_BOOKINGS_BY_BOOKING_PLATFORMS AS
 SELECT
  booking_platform,
  bookings,
  SUM(bookings) OVER () AS total_bookings,
  CONCAT(ROUND(100.0 * bookings / SUM(bookings) OVER (), 2)," %") AS pct_of_total
FROM (
  SELECT booking_platform,
         COUNT(*) AS bookings
  FROM fact_bookings
  GROUP BY booking_platform
) AS grp;

 ------------------------------------------------------------
 /* BOOKINGS % BY ROOM CLASS */
 select * from fact_bookings;
 CREATE VIEW PERCENTAGE_BOOKINGS_BY_ROOM_CATEGORY AS
 SELECT
  ROOM_CATEGORY,
  bookings,
  SUM(bookings) OVER () AS total_bookings,
  CONCAT(ROUND(100.0 * bookings / SUM(bookings) OVER (), 2)," %") AS pct_of_total
FROM (SELECT ROOM_CATEGORY,COUNT(BOOKING_ID) AS BOOKINGS FROM
FACT_BOOKINGS
 GROUP BY ROOM_CATEGORY) AS GROUP_;

  ------------------------------------------------------------
  SELECT * FROM FACT_BOOKINGS;
 /* WOW CHANGE IN REVENUE */

 WITH CTECW AS(
 SELECT DISTINCT WEEK(CHECK_IN_DATE)AS WEEKNO,COUNT(BOOKING_ID) AS
BOOKINGS)
  GROUP BY WEEK(CHECK_IN_DATE)
 ,CTENW AS (
 LEAD(COUNT(BOOKING_ID)) OVER() AS NEXT_VALUE FROM FACT_BOOKINGS
 GROUP BY WEEK(CHECK_IN_DATE))
 SELECT(SELECT * FROM CTEW)/(SELECT * FROM CTENW)


 WITH CTEWOW(
 SELECT DISTINCT WEEK(CHECK_IN_DATE)AS WEEKNO,COUNT(BOOKING_ID) AS
BOOKINGS,
 LEAD(COUNT(BOOKING_ID)) OVER() AS NEXT_VALUE
 FROM FACT_BOOKINGS
 GROUP BY WEEK(CHECK_IN_DATE)
 )
  SELECT * FROM CTEWOW
  GROUP BY WEEKNO;

--------------------------------------------------------------
/* WEEK OVER WEEK CHANGE REVENUE*/
CREATE VIEW WOW_REVENUE AS
WITH CTEWOW AS (
  SELECT
    WEEK(CHECK_IN_DATE, 1) AS WEEKNO,
    COUNT(BOOKING_ID) AS BOOKINGS
  FROM FACT_BOOKINGS
  GROUP BY WEEK(CHECK_IN_DATE, 1)
)
SELECT
  WEEKNO,
  BOOKINGS,
  LEAD(BOOKINGS) OVER (ORDER BY WEEKNO) AS NEXT_VALUE,
CONCAT(ROUND((BOOKINGS/LEAD(BOOKINGS) OVER (ORDER BY WEEKNO))*100,2),"
%") AS WOW_REV_CHANGE_
FROM CTEWOW;
```

```
------------------------------------------------------------------------------------------------------------------------
/* FLITER REVENUE WISE */
select distinct property_name from dim_hotels;
select *from FACT_BOOKINGS;

SELECT * FROM DIM_HOTELS JOIN FACT_BOOKINGS ON
DIM_HOTELS.PROPERTY_ID = FACT_BOOKINGS.PROPERTY_ID;

/* city wise revenu */
SELECT DIM_HOTELS.CITY,CONCAT(SUM(FACT_BOOKINGS.REVENUE_REALIZED), "
Rs.") AS REVENUE  FROM DIM_HOTELS JOIN FACT_BOOKINGS ON
DIM_HOTELS.PROPERTY_ID = FACT_BOOKINGS.PROPERTY_ID
GROUP BY DIM_HOTELS.CITY;

/* propert name wise */
SELECT dim_hotels.property_name AS
PROPERTY_NAME,concat(sum(fact_bookings.revenue_realized)," Rs.") as REVENUE
FROM DIM_HOTELS JOIN FACT_BOOKINGS ON
DIM_HOTELS.PROPERTY_ID = FACT_BOOKINGS.PROPERTY_ID
group by dim_hotels.property_name;


SELECT * FROM DIM_ROOMS JOIN FACT_BOOKINGS ON
DIM_ROOMS.ROOM_ID = FACT_BOOKINGS.ROOM_CATEGORY;

/* room class wise revenue */
SELECT
DIM_ROOMS.ROOM_CLASS,CONCAT(SUM(FACT_BOOKINGS.REVENUE_REALIZED),"
Rs.") AS REVENUE FROM DIM_ROOMS JOIN FACT_BOOKINGS ON
DIM_ROOMS.ROOM_ID = FACT_BOOKINGS.ROOM_CATEGORY
GROUP BY DIM_ROOMS.ROOM_CLASS;

/* booking platform wise revenue */


SELECT booking_platform,CONCAT(sum(revenue_realized)," Rs.") as REVENUE FROM
FACT_BOOKINGS
GROUP BY BOOKING_PLATFORM;



------------------------------------------------------------------------------------------------------------------------
/* FILTER BOOKINGS WIESE */

SELECT * FROM DIM_HOTELS JOIN FACT_AGGBOOKINGS ON
DIM_HOTELS.PROPERTY_ID = FACT_AGGBOOKINGS.PROPERTY_ID;
```

```
SELECT DIM_HOTELS.CITY,SUM(FACT_AGGBOOKINGS.successful_bookings) AS
BOOKINGS FROM DIM_HOTELS JOIN FACT_AGGBOOKINGS ON
DIM_HOTELS.PROPERTY_ID = FACT_AGGBOOKINGS.PROPERTY_ID
GROUP BY  DIM_HOTELS.CITY;

SELECT DIM_HOTELS.CATEGORY,SUM(FACT_AGGBOOKINGS.successful_bookings)
AS BOOKINGS FROM DIM_HOTELS JOIN FACT_AGGBOOKINGS ON
DIM_HOTELS.PROPERTY_ID = FACT_AGGBOOKINGS.PROPERTY_ID
GROUP BY  DIM_HOTELS.CATEGORY;

SELECT DIM_ROOMS.ROOM_CLASS,
SUM(FACT_AGGBOOKINGS.successful_bookings) AS BOOKINGS FROM DIM_ROOMS
JOIN FACT_AGGBOOKINGS ON
DIM_ROOMS.ROOM_ID = FACT_AGGBOOKINGS.room_category
GROUP BY ROOM_CLASS;

/* FLITER SECTION */

/* REVENUE AND BOOKINGS : FILTERS : CITY | PROPERTY NAME | ROOM CLASS */

DELIMITER $$
CREATE FUNCTION ROOMCLASSREV(CATEGORY VARCHAR(100))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
   DECLARE totalRevenue BIGINT;

   SELECT SUM(B.REVENUE_REALIZED)
   INTO totalRevenue
   FROM fact_bookings B
   JOIN DIM_ROOMS R
     ON B.ROOM_CATEGORY = R.ROOM_ID
   WHERE R.ROOM_CLASS = CATEGORY;

   RETURN CONCAT(totalRevenue, ' Rs.');
END $$

DELIMITER ;
/*------------------------------------------------------------------------------------------*/
SELECT * FROM DIM_HOTELS;
SELECT * FROM DIM_ROOMS;
SELECT * FROM fact_AGGbookings;

DELIMITER $$
CREATE FUNCTION CITYREV(CITY VARCHAR(100))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
   DECLARE totalRevenue BIGINT;
```

```
    SELECT SUM(B.REVENUE_REALIZED)
    INTO totalRevenue
    FROM fact_bookings B
    JOIN DIM_HOTELS H
      ON B.property_id = H.property_id
    WHERE H.CITY = CITY;

    RETURN CONCAT(totalRevenue, ' Rs.');
END $$

DELIMITER ;
/*------------------------------------------------------------------------------------------*/

DELIMITER $$
CREATE FUNCTION PROPREV(PROPERTY_NAME VARCHAR(255))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
    DECLARE totalRevenue BIGINT;

    SELECT SUM(B.REVENUE_REALIZED)
    INTO totalRevenue
    FROM fact_bookings B
    JOIN DIM_HOTELS H
      ON B.property_id = H.property_id
    WHERE H.PROPERTY_NAME = PROPERTY_NAME;

    RETURN CONCAT(totalRevenue, ' Rs.');
END $$

DELIMITER ;
/*------------------------------------------------------------------------------------------*/
/* BOOKINGS */

SELECT * FROM DIM_HOTELS;
SELECT * FROM DIM_HOTELS JOIN fact_aggbookings ON
DIM_HOTELS.PROPERTY_ID=fact_aggbookings.PROPERTY_ID;

SELECT DIM_HOTELS.CITY,SUM(FACT_AGGBOOKINGS.successful_bookings) AS
BOOKINGS FROM DIM_HOTELS JOIN fact_aggbookings ON
DIM_HOTELS.PROPERTY_ID=fact_aggbookings.PROPERTY_ID
GROUP BY DIM_HOTELS.CITY;

DELIMITER $$
CREATE FUNCTION CITY_BOOKINGS(CITY VARCHAR(100))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
    DECLARE BOOKINGS BIGINT;
```

```
   SELECT SUM(B.successful_bookings)
   INTO BOOKINGS
   FROM fact_AGGbookings B
   JOIN DIM_HOTELS H
     ON B.PROPERTY_ID = H.PROPERTY_ID
   WHERE H.CITY = CITY;

   RETURN BOOKINGS;
END $$

DELIMITER ;

/*----------------------------------------------------------------------------------------*/
SELECT * FROM DIM_ROOMS;
SELECT * FROM fact_aggbookings;


DELIMITER $$
CREATE FUNCTION ROOMCLASS_BOOKINGS(ROOM_CLASS VARCHAR(100))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
   DECLARE BOOKINGS BIGINT;

   SELECT SUM(B.successful_bookings)
   INTO BOOKINGS
   FROM fact_AGGbookings B
   JOIN DIM_ROOMS R
     ON B.room_category = R.ROOM_ID
   WHERE R.ROOM_CLASS = ROOM_CLASS;

   RETURN BOOKINGS;
END $$

DELIMITER ;

/*----------------------------------------------------------------------------------------*/
SELECT H.property_name,SUM(B.successful_bookings) AS BOOKINGS FROM
DIM_HOTELS H JOIN fact_aggbookings B ON
H.PROPERTY_ID=B.PROPERTY_ID
GROUP BY  H.property_name;

DELIMITER $$
CREATE FUNCTION PROP_BOOKINGS(PROPERTY_NAME VARCHAR(255))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
   DECLARE BOOKINGS BIGINT;

   SELECT SUM(B.successful_bookings)
   INTO BOOKINGS
   FROM fact_AGGbookings B
   JOIN DIM_HOTELS H
     ON B.PROPERTY_ID = H.PROPERTY_ID
   WHERE H.property_name = PROPERTY_NAME;

   RETURN BOOKINGS;
END $$

DELIMITER ;
/*-------------------------------------------------------------------------------------------*/
/* EXTRA POINTERS */

/* 1. MOST / LEAST BOOKED CITY | PROPERTY*/

SELECT * FROM DIM_HOTELS;
SELECT
DIM_HOTELS.CITY,DIM_HOTELS.PROPERTY_NAME,FACT_BOOKINGS.PROPERTY_ID,C
OUNT(FACT_BOOKINGS.property_id) AS BOOKINGS FROM FACT_BOOKINGS JOIN
DIM_HOTELS ON
DIM_HOTELS.PROPERTY_ID=FACT_BOOKINGS.PROPERTY_ID
GROUP BY property_id
ORDER BY COUNT(property_id) DESC
LIMIT 1;

/* 2. LEAST BOOKED CITY | PROPERTY*/
SELECT
DIM_HOTELS.CITY,DIM_HOTELS.PROPERTY_NAME,FACT_BOOKINGS.PROPERTY_ID,C
OUNT(FACT_BOOKINGS.property_id) AS BOOKINGS FROM FACT_BOOKINGS JOIN
DIM_HOTELS ON
DIM_HOTELS.PROPERTY_ID=FACT_BOOKINGS.PROPERTY_ID
GROUP BY property_id
ORDER BY COUNT(property_id)
LIMIT 1;

/*-------------------------------------------------------------------------------------------*/
/* 1. MOST / LEAST REVENUE CREATING CITY | PROPERTY*/
SELECT
DIM_HOTELS.CITY,DIM_HOTELS.PROPERTY_NAME,FACT_BOOKINGS.PROPERTY_ID,
CONCAT(SUM(FACT_BOOKINGS.REVENUE_REALIZED)," Rs.") AS REVENUE
FROM FACT_BOOKINGS JOIN DIM_HOTELS ON
DIM_HOTELS.PROPERTY_ID=FACT_BOOKINGS.PROPERTY_ID
GROUP BY property_id
ORDER BY  SUM(FACT_BOOKINGS.REVENUE_REALIZED) DESC
LIMIT 1;

SELECT
DIM_HOTELS.CITY,DIM_HOTELS.PROPERTY_NAME,FACT_BOOKINGS.PROPERTY_ID,
CONCAT(SUM(FACT_BOOKINGS.REVENUE_REALIZED)," Rs.") AS REVENUE
FROM FACT_BOOKINGS JOIN DIM_HOTELS ON
```

```
DIM_HOTELS.PROPERTY_ID=FACT_BOOKINGS.PROPERTY_ID
GROUP BY property_id
ORDER BY  SUM(FACT_BOOKINGS.REVENUE_REALIZED)
LIMIT 1;

/*------------------------------------------------------------------------------------------*/
/* STORED PROCEDURES FOR TOP AND BOTTOM VALUES */

/* 1. N TOP HOTELS REVENUE WISE*/
DELIMITER $$

CREATE PROCEDURE GetTopHotels(IN limit_value INT)
BEGIN
    SELECT
        DIM_HOTELS.CITY,
        DIM_HOTELS.PROPERTY_NAME,
        FACT_BOOKINGS.PROPERTY_ID,
        CONCAT(SUM(FACT_BOOKINGS.REVENUE_REALIZED), ' Rs.') AS REVENUE
    FROM FACT_BOOKINGS
    JOIN DIM_HOTELS
        ON DIM_HOTELS.PROPERTY_ID = FACT_BOOKINGS.PROPERTY_ID
    GROUP BY FACT_BOOKINGS.PROPERTY_ID
    ORDER BY SUM(FACT_BOOKINGS.REVENUE_REALIZED) DESC
    LIMIT limit_value;
END$$

DELIMITER ;
/*------------------------------------------------------------------------------------------*/
/* 2. N BOTTOM HOTELS REVENUE WISE*/
DELIMITER $$

CREATE PROCEDURE GetBottomHotels(IN limit_value INT)
BEGIN
    SELECT
        DIM_HOTELS.CITY,
        DIM_HOTELS.PROPERTY_NAME,
        FACT_BOOKINGS.PROPERTY_ID,
        CONCAT(SUM(FACT_BOOKINGS.REVENUE_REALIZED), ' Rs.') AS REVENUE
    FROM FACT_BOOKINGS
    JOIN DIM_HOTELS
        ON DIM_HOTELS.PROPERTY_ID = FACT_BOOKINGS.PROPERTY_ID
    GROUP BY FACT_BOOKINGS.PROPERTY_ID
    ORDER BY SUM(FACT_BOOKINGS.REVENUE_REALIZED)
    LIMIT limit_value;
END$$

DELIMITER ;
/*------------------------------------------------------------------------------------------*/
/* 2. N TOP / BOTTOM HOTELS BOOKINGS WISE*/
DELIMITER //
```

```
CREATE PROCEDURE GetBottomHotels_BOOKINGS(IN p_limit INT)
BEGIN
   SELECT h.CITY,
        h.PROPERTY_NAME,
        f.PROPERTY_ID,
        COUNT(f.PROPERTY_ID) AS BOOKINGS
   FROM FACT_BOOKINGS f
   JOIN DIM_HOTELS h
      ON h.PROPERTY_ID = f.PROPERTY_ID
   GROUP BY f.PROPERTY_ID
   ORDER BY COUNT(f.PROPERTY_ID) DESC
   LIMIT p_limit;
END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE GetBottomHotels_BOOKINGS(IN p_limit INT)
BEGIN
   SELECT h.CITY,
        h.PROPERTY_NAME,
        f.PROPERTY_ID,
        COUNT(f.PROPERTY_ID) AS BOOKINGS
   FROM FACT_BOOKINGS f
   JOIN DIM_HOTELS h
      ON h.PROPERTY_ID = f.PROPERTY_ID
   GROUP BY f.PROPERTY_ID
   ORDER BY COUNT(f.PROPERTY_ID)
   LIMIT p_limit;
END //

DELIMITER ;

/*--------------------------------------------------------------------------------------------*/
LANDING PAGE

DELIMITER $$

CREATE PROCEDURE SelectFromCustomView(IN viewName VARCHAR(64))
BEGIN
   -- Declare SQL variable
   SET @sql = CONCAT('SELECT * FROM ', viewName, ' LIMIT 50;');

   -- Prepare and execute the dynamic SQL
   PREPARE stmt FROM @sql;
   EXECUTE stmt;
   DEALLOCATE PREPARE stmt;
END$$
```

DELIMITER ;

ROOM FOR IMPROVEMENTS

1. **DESPITE OF BEING CAPITAL OF COUNTRY DELHI HAS RECEIVED LEAST BOOKINGS RESULTING IN LEAST REVENUE GENERATING STATE.**

- **Company can work on tie-ups with political parties during sessions of parliament such as Budget session, Monsoon session and Winter session**
- **The Company can promote their business in the most relevant regions where tourist visits are followed by a halt.**
  Most tourists from abroad start with Delhi–Agra–Jaipur (Golden Triangle).
  Nature lovers go to Himachal or Uttarakhand.
  Spiritual seekers visit Varanasi, Haridwar, Rishikesh, or Amritsar.

2. **MINIMIZING DIFFERENCE BETWEEN EACH CLASS'S BUSINESS**

- **% Difference between each class's revenue collected is app 21-22 %**
- **The company can work on reducing it up to 10 % by deploying dynamic pricing.**
- **Highest revenue share by ELITE class highlights clear focus of the company**
  On the luxury side, the company can think of attracting the general masses for the STANDARD class at discounted price.

3. **IMPROVISING BOOKING CHANNELS**

- **The company can use SEO for improving DIRECT OFFLINE bookings, Strategic commission based tie-ups with local tour operators, travel agencies , cab drivers, and restaurants can help to bring more DIRECT OFFLINE customers.**
- **Digital marketing and influencer based marketing can help online channels to attract more customers.**

4. **OFF SEASON BOOKINGS**

- **As we are moving away from the vacation period, bookings are decreasing.**
- **Promotional activities,effective marketing and targeting local events and occasions can help the company to become season independent.**

5. **AVERAGE RATING (3.62)**

- **The company can implement programs to enhance the user experience of customers.**

- **Improvising on service factors and keeping all amenities in order can help to get more ratings.**

6. MOST BOOKED AND LEAST BOOKED HOTELS

- **MUMBAI’S Atliq exotica receiving maximum bookings (7338)**
- **DELHI’S Atliq grands received least bookings (3153)**