TELANGANA TOURISM ANALYSIS

Provide Insights for Telangana Government Tourism Department.

Task1:
--Merge all individual CSV files in "domestic_visitors" and "foreign_visitors" folders using a data integration tool such as Pandas or PowerBI, and name the resulting files "domestic_visitors.csv" and "foreign_visitors.csv", respectively, containing all data from 2016 to 2019.

Task2:
--Once the merged data is obtained, you can use it to answer the questions listed in the file 'research_questions_and_recommendations.pdf'. You can use any tool of your choice (Python, SQL, PowerBI, Tableau, Excel) to answer these questions.

Telangana Tourism Analysis.

Task1:

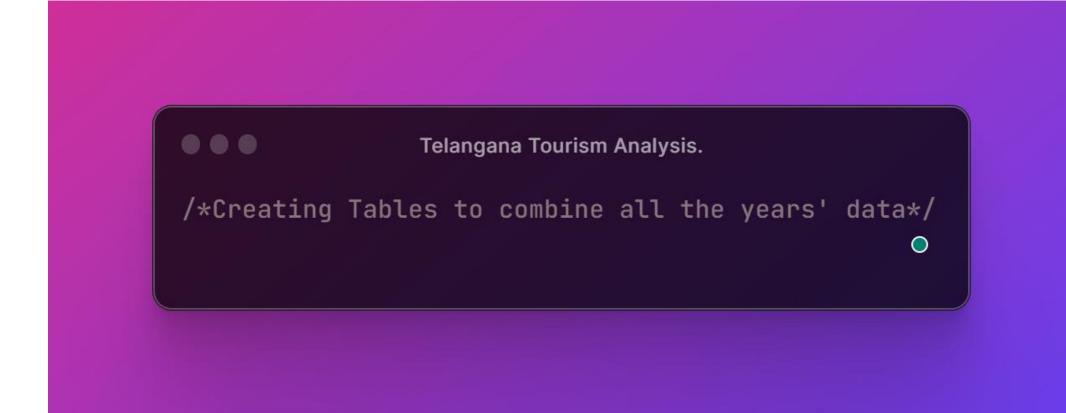--Getting an overview of the already provided tables.

Telangana Tourism Analysis.

```sql
/*domestic_visitor_tables*/

SELECT COLUMN_NAME,
       DATA_TYPE,
       IS_NULLABLE,
       COLUMN_DEFAULT,
       CHARACTER_MAXIMUM_LENGTH,
       NUMERIC_PRECISION,
       NUMERIC_SCALE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'domestic_visitors_2016';
```

	COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFA...	CHARACTER_M...	NUMERIC_PRE...	NUMERIC_SCALE
1	district	nvarchar	NO	NULL	50	NULL	NULL
2	date	date	NO	NULL	NULL	NULL	NULL
3	month	nvarchar	NO	NULL	50	NULL	NULL
4	year	smallint	NO	NULL	NULL	5	0
5	visitors	int	YES	NULL	NULL	10	0

Telangana Tourism Analysis.

```sql
/*foreigh_visitors_tables*/

SELECT COLUMN_NAME,
       DATA_TYPE,
       IS_NULLABLE,
       COLUMN_DEFAULT,
       CHARACTER_MAXIMUM_LENGTH,
       NUMERIC_PRECISION,
       NUMERIC_SCALE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'foreign_visitors_2016';
```
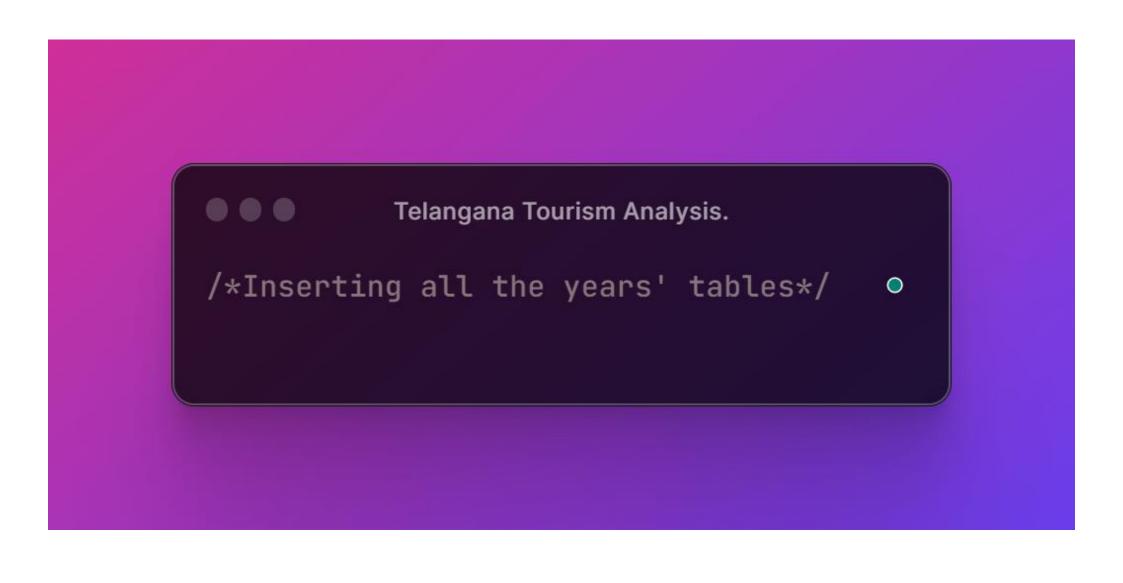
▲ RESULTS

	COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFA...	CHARACTER_M...	NUMERIC_PRE...	NUMERIC_SCALE
1	district	nvarchar	NO	*NULL*	50	*NULL*	*NULL*
2	date	date	NO	*NULL*	*NULL*	*NULL*	*NULL*
3	month	nvarchar	NO	*NULL*	50	*NULL*	*NULL*
4	year	smallint	NO	*NULL*	*NULL*	5	0
5	visitors	smallint	YES	*NULL*	*NULL*	5	0

Telangana Tourism Analysis.
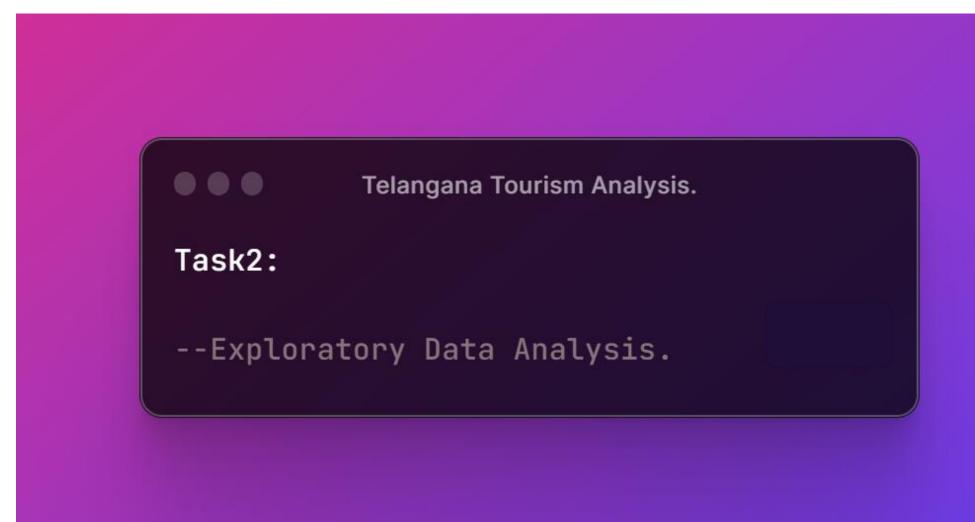
```
/*Creating Tables to combine all the years' data*/
```

```sql
/*domestic_visitors_table*/

CREATE TABLE domestic_visitors
(
    district NVARCHAR(50),
    date DATE,
    month NVARCHAR(50),
    year SMALLINT,
    visitors INT NULL
);

/*foreigh_visitors_table*/

CREATE TABLE foreign_visitors
(
    district NVARCHAR(50),
    date DATE,
    month NVARCHAR(50),
    year SMALLINT,
    visitors INT NULL
);
```

Telangana Tourism Analysis.

```
/*Inserting all the years' tables*/
```

```sql
/*domestic_visitor_tables*/

INSERT INTO domestic_visitors
(
    district,
    [date],
    [month],
    [year],
    visitors
)
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM domestic_visitors_2016
UNION ALL
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM domestic_visitors_2017
UNION ALL
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM domestic_visitors_2018
UNION ALL
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM domestic_visitors_2019;
```

```sql
/*foreigh_visitors_tables*/

INSERT INTO foreign_visitors
(
    district,
    [date],
    [month],
    [year],
    visitors
)
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM foreign_visitors_2016
UNION ALL
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM foreign_visitors_2017
UNION ALL
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM foreign_visitors_2018
UNION ALL
SELECT district,
        [date],
        [month],
        [year],
        visitors
FROM foreign_visitors_2019;

--Task1: Merging of files: Completed--
```

Task2:

--Exploratory Data Analysis.

```sql
/*Getting an overview of the two new tables with total data*/


/*domestic_visitors*/


SELECT *
FROM domestic_visitors;


/*foreigh_visitors*/


SELECT *
FROM foreign_visitors;
```

RESULTS (Left)

	district	date	month	year	visitors
1	Adilabad	2016-01-01	January	2016	792136
2	Adilabad	2016-01-02	February	2016	937820
3	Adilabad	2016-01-03	March	2016	582946
4	Adilabad	2016-01-04	April	2016	341948
5	Adilabad	2016-01-05	May	2016	252887
6	Adilabad	2016-01-06	June	2016	368237
7	Adilabad	2016-01-07	July	2016	447562
8	Adilabad	2016-01-08	August	2016	614285
9	Adilabad	2016-01-09	September	2016	491279
10	Adilabad	2016-01-10	October	2016	94184
11	Adilabad	2016-01-11	November	2016	99148
12	Adilabad	2016-01-12	December	2016	53125
13	Bhadradri Koth...	2016-01-01	January	2016	NULL
14	Bhadradri Koth...	2016-01-02	February	2016	NULL
15	Bhadradri Koth...	2016-01-03	March	2016	NULL
16	Bhadradri Koth...	2016-01-04	April	2016	NULL
17	Bhadradri Koth...	2016-01-05	May	2016	NULL
18	Bhadradri Koth...	2016-01-06	June	2016	NULL
19	Bhadradri Koth...	2016-01-07	July	2016	NULL
20	Bhadradri Koth...	2016-01-08	August	2016	NULL

RESULTS (Right)

	district	date	month	year	visitors
1	Adilabad	2016-01-01	January	2016	2
2	Adilabad	2016-01-02	February	2016	0
3	Adilabad	2016-01-03	March	2016	2
4	Adilabad	2016-01-04	April	2016	0
5	Adilabad	2016-01-05	May	2016	0
6	Adilabad	2016-01-06	June	2016	0
7	Adilabad	2016-01-07	July	2016	4
8	Adilabad	2016-01-08	August	2016	2
9	Adilabad	2016-01-09	September	2016	0
10	Adilabad	2016-01-10	October	2016	0
11	Adilabad	2016-01-11	November	2016	0
12	Adilabad	2016-01-12	December	2016	0
13	Bhadradri Koth...	2016-01-01	January	2016	NULL
14	Bhadradri Koth...	2016-01-02	February	2016	NULL
15	Bhadradri Koth...	2016-01-03	March	2016	NULL
16	Bhadradri Koth...	2016-01-04	April	2016	NULL
17	Bhadradri Koth...	2016-01-05	May	2016	NULL
18	Bhadradri Koth...	2016-01-06	June	2016	NULL
19	Bhadradri Koth...	2016-01-07	July	2016	NULL
20	Bhadradri Koth...	2016-01-08	August	2016	NULL

```
/*Both the queries result in 1512 rows. There is no missing data. Hence no
discrepancy.*/
```

```sql
--List of the top 10 districts that have the highest number of domestic visitors
overall (2016-2019)--

SELECT TOP 10
    district,
    SUM(visitors) AS 'total_visitors'
FROM domestic_visitors
GROUP BY district
ORDER BY total_visitors DESC;
```

	district	total_visitors
1	Hyderabad	83900960
2	Rajanna Sircilla	41763276
3	Warangal (Urba...	30726603
4	Yadadri Bhongir	26893080
5	Bhadradri Koth...	21600962
6	Medak	20542639
7	Jayashankar Bh...	19632865
8	Mahbubnagar	17180118
9	Nirmal	13315796
10	Jagtial	11303514

```sql
--What are the peak and low seasons for Hyderabad based on the data from 2016 to
2019 for Hyderabad district?--

/*domestic_visitors*/

/*peak seasons*/

SELECT [month],
       SUM(visitors) AS total_visitors
FROM domestic_visitors
WHERE district = 'HYDERABAD'
GROUP BY [month]
ORDER BY total_visitors DESC;

/*We can see that the highest peak is in the month of June (Summer) and December
(Winter)*/
```

RESULTS

	month	total_visitors
1	June	16897783
2	December	9338637
3	October	6552397
4	January	6452101
5	April	6126839
6	May	6049214
7	August	5750967
8	November	5626156
9	July	5552527
10	September	5312283
11	March	5227626
12	February	5014430

```sql
--What are the peak and low seasons for Hyderabad based on the data from 2016 to
2019 for Hyderabad district?--


/*domestic_visitors*/


/*low seasons*/


SELECT [month],
        SUM(visitors) AS total_visitors
FROM domestic_visitors
WHERE district = 'HYDERABAD'
GROUP BY [month]
ORDER BY total_visitors;


/*We can see that the low is in the months of February, March and September.*/
```

◢ RESULTS

	month	total_visitors
1	February	5014430
2	March	5227626
3	September	5312283
4	July	5552527
5	November	5626156
6	August	5750967
7	May	6049214
8	April	6126839
9	January	6452101
10	October	6552397
11	December	9338637
12	June	16897783

```sql
--What are the peak and low seasons for Hyderabad based on the data from 2016 to
2019 for Hyderabad district?--

/*foreigh_visitors*/

/*peak seasons*/

SELECT [month],
        SUM(visitors) AS total_visitors
FROM foreign_visitors
WHERE district = 'HYDERABAD'
GROUP BY [month]
ORDER BY total_visitors DESC;

/*We can see that the highest peak is in the months of December, January and
February (Winter)*/
```

	month	total_visitors
1	December	119995
2	January	106450
3	February	103778
4	October	97954
5	September	94080
6	November	93503
7	August	83769
8	July	80616
9	March	76358
10	June	67524
11	April	60495
12	May	60376

◢ RESULTS

Telangana Tourism Analysis.

```sql
--What are the peak and low seasons for Hyderabad based on the data from 2016 to
2019 for Hyderabad district?--


/*foreigh_visitors*/

/*low seasons*/


SELECT [month],
       SUM(visitors) AS total_visitors
FROM foreign_visitors
WHERE district = 'HYDERABAD'
GROUP BY [month]
ORDER BY total_visitors;

/*We can see that the low is in the months of May, April and June (Summer).*/
```

◢ RESULTS

	month	total_visitors
1	May	60376
2	April	60495
3	June	67524
4	March	76358
5	July	80616
6	August	83769
7	November	93503
8	September	94080
9	October	97954
10	February	103778
11	January	106450
12	December	119995

```sql
--Find the top ranking districts for each year using window functions--

/*domestic_visitors*/


SELECT *
FROM
(
    SELECT year,
            district,
            SUM(visitors) AS total_visitors,
            RANK() OVER (PARTITION BY year ORDER BY SUM(visitors) DESC) AS rank
    FROM domestic_visitors
    GROUP BY district,
            [year]
) ranking
WHERE ranking.rank <= 5;
```

	year	district	total_visitors	rank
1	2016	Warangal (Urba...	25788035	1
2	2016	Hyderabad	23394705	2
3	2016	Karimnagar	9167468	3
4	2016	Mahbubnagar	8304766	4
5	2016	Nalgonda	5858461	5
6	2017	Hyderabad	27160242	1
7	2017	Rajanna Sircilla	11919347	2
8	2017	Medak	7726869	3
9	2017	Yadadri Bhongir	7001728	4
10	2017	Nirmal	4405083	5
11	2018	Hyderabad	19543651	1
12	2018	Jayashankar Bh...	16895925	2
13	2018	Yadadri Bhongir	13673378	3
14	2018	Rajanna Sircilla	10834231	4
15	2018	Nirmal	4177325	5
16	2019	Rajanna Sircilla	16832897	1
17	2019	Hyderabad	13802362	2
18	2019	Bhadradri Koth...	12817737	3
19	2019	Medak	5452570	4
20	2019	Sangareddy	4553160	5

Telangana Tourism Analysis.

```sql
--Find the top ranking districts for each year using window functions--

/*foreign_visitors*/


SELECT *
FROM
(
    SELECT year,
            district,
            SUM(visitors) AS total_visitors,
            RANK() OVER (PARTITION BY year ORDER BY SUM(visitors) DESC) AS rank
    FROM foreign_visitors
    GROUP BY district,
            [year]
) ranking
WHERE ranking.rank <= 5;
```

	year	district	total_visitors	rank
1	2016	Hyderabad	163631	1
2	2016	Warangal (Urba...	1899	2
3	2016	Mahbubnagar	868	3
4	2016	Jayashankar Bh...	86	4
5	2016	Jogulamba Gad...	45	5
6	2017	Hyderabad	247179	1
7	2017	Warangal (Urba...	2630	2
8	2017	Jayashankar Bh...	582	3
9	2017	Mahbubnagar	520	4
10	2017	Nagarkurnool	311	5
11	2018	Hyderabad	314788	1
12	2018	Warangal (Urba...	1842	2
13	2018	Jayashankar Bh...	539	3
14	2018	Mahbubnagar	454	4
15	2018	Jogulamba Gad...	300	5
16	2019	Hyderabad	319300	1
17	2019	Warangal (Urba...	2450	2
18	2019	Mulugu	575	3
19	2019	Mahbubnagar	440	4
20	2019	Jogulamba Gad...	295	5

Telangana Tourism Analysis.

```sql
--Find the top ranking months for each year using window functions--

/*domestic_visitors*/

SELECT *
FROM
(
    SELECT year,
           [month],
           SUM(visitors) AS total_visitors,
           RANK() OVER (PARTITION BY year ORDER BY SUM(visitors) DESC) AS rank
    FROM domestic_visitors
    GROUP BY [month],
             [year]
) ranking
WHERE ranking.rank <= 3;
```

RESULTS

	year	month	total_visitors	rank
1	2016	February	25675243	1
2	2016	June	15406150	2
3	2016	January	7773492	3
4	2017	December	11772522	1
5	2017	February	7352530	2
6	2017	November	7219814	3
7	2018	February	18591596	1
8	2018	January	16188552	2
9	2018	March	8428964	3
10	2019	June	15913901	1
11	2019	January	10782269	2
12	2019	March	6824552	3

Telangana Tourism Analysis.

```sql
--Find the top ranking months for each year using window functions--

/*foreign_visitors*/

SELECT *
FROM
(
    SELECT year,
           [month],
           SUM(visitors) AS total_visitors,
           RANK() OVER (PARTITION BY year ORDER BY SUM(visitors) DESC) AS rank
    FROM foreign_visitors
    GROUP BY [month],
             [year]
) ranking
WHERE ranking.rank <= 3;
```

	year	month	total_visitors	rank
1	2016	February	30163	1
2	2016	December	17839	2
3	2016	July	17262	3
4	2017	December	39353	1
5	2017	November	29603	2
6	2017	September	28294	3
7	2018	January	35696	1
8	2018	August	31500	2
9	2018	February	30709	3
10	2019	January	36256	1
11	2019	December	34411	2
12	2019	September	34384	3

```sql
--What are the top & bottom 3 districts with high domestic to foreign tourist ratio?
--


/*Top 3*/

/*Method 1: Using Subqueries*/

SELECT TOP 3
    district,
    domestic_tourist,
    foreign_tourist,
    domestic_tourist / foreign_tourist AS ratio
FROM
(
    SELECT district AS domestic_district,
            SUM(visitors) AS domestic_tourist
    FROM domestic_visitors
    GROUP BY district
    HAVING SUM(visitors) <> 0
) AS domestic_tourists
    JOIN
    (
        SELECT district,
                SUM(visitors) AS foreign_tourist
        FROM foreign_visitors
        GROUP BY district
        HAVING SUM(visitors) <> 0
    ) AS foreign_tourists
        ON domestic_tourists.domestic_district = foreign_tourists.district
ORDER BY ratio;
```

RESULTS

	district	domestic_tourist	foreign_tourist	ratio
1	Hyderabad	83900960	1044898	80
2	Warangal (Rural)	819162	306	2677
3	Mulugu	1819800	575	3164

```sql
--What are the top & bottom 3 districts with high domestic to foreign tourist ratio?
--


/*Top 3*/

/*Method 2: Using CTE's or Subquery factoring*/

WITH CTE_Domestic_Tourists
AS (SELECT district,
            SUM(visitors) AS domestic_tourist
    FROM domestic_visitors
    GROUP BY district
    HAVING SUM(visitors) <> 0
    ),
    CTE_Foreign_Tourists
AS (SELECT district,
            SUM(visitors) AS foreign_tourist
    FROM foreign_visitors
    GROUP BY district
    HAVING SUM(visitors) <> 0
    )
SELECT TOP 3
    CTE_Domestic_Tourists.district,
    domestic_tourist,
    foreign_tourist,
    domestic_tourist / foreign_tourist AS ratio
FROM CTE_Domestic_Tourists
    JOIN CTE_Foreign_Tourists
        ON CTE_Domestic_Tourists.district = CTE_Foreign_Tourists.district
ORDER BY ratio;


/* Method 2 is better in terms of optimization and speed because CTEs are pre-
compiled and stored in memory, which can improve the performance of queries that use
them. Subqueries, on the other hand, are not pre-compiled and are evaluated each
time they are executed, which can make them slower.
Overall, CTEs are a better choice than subqueries for most queries. They are more
efficient, more readable, and more maintainable. */
```

	district	domestic_tourist	foreign_tourist	ratio
1	Hyderabad	83900960	1044898	80
2	Warangal (Rural)	819162	306	2677
3	Mulugu	1819800	575	3164

```sql
--What are the top & bottom 3 districts with high domestic to foreign tourist ratio?
--

/*Bottom 3*/

WITH CTE_Domestic_Tourists
AS (SELECT district,
            SUM(visitors) AS domestic_tourist
    FROM domestic_visitors
    GROUP BY district
    HAVING SUM(visitors) <> 0
    ),
     CTE_Foreign_Tourists
AS (SELECT district,
            SUM(visitors) AS foreign_tourist
    FROM foreign_visitors
    GROUP BY district
    HAVING SUM(visitors) <> 0
    )
SELECT TOP 3
    CTE_Domestic_Tourists.district,
    domestic_tourist,
    foreign_tourist,
    domestic_tourist / foreign_tourist AS ratio
FROM CTE_Domestic_Tourists
    JOIN CTE_Foreign_Tourists
        ON CTE_Domestic_Tourists.district = CTE_Foreign_Tourists.district
ORDER BY ratio DESC;
```

	district	domestic_tourist	foreign_tourist	ratio
1	Nirmal	13315796	2	6657898
2	Jangaon	826280	2	413140
3	Adilabad	7321575	32	228799

```sql
--What are the top 5 and bottom 5 districts based on 'Population to Tourisst
Footfall Ratio' in 2019?--

/*Getting an overview of the demographics table*/

SELECT *
FROM demographics;

/*Need to add a new column for population in 2019*/

ALTER TABLE demographics ADD population_2019 INT;

/*Estimating and inserting data into the newly added column*/
/*Esmitating the population in 2019 using the census of 2011 and the estimated
values of 2023*/

UPDATE demographics
SET population_2019 = As_per_2011_census
                      + ((Estimated_Population_in_2023 - As_per_2011_census) / (2023
- 2011) * (2019 - 2011));

/*In this query, we use the formula for linear interpolation:
EstimatedPopulation2019 = Population2011 + ((Population2023 - Population2011) /
(2023-2011) * (2019 - 2011)). This formula calculates the estimated population for
the year 2019 based on the population data we have for the years 2011 and 2023.*/
```

Districts	Males	Females	Sex_Ratio_Fem...	Rural	Urban	Rural_Populatio...	Urban_Populati...	Households	As_per_2011_c...	Estimated_Pop...	population_2019
ADILABAD	356407	352565	989	541226	167746	76.3399963378...	23.6599998474...	156683	708972	768667	748764
BHADRADRI KO...	532390	536871	1008	730178	339083	68.2900009155...	31.7099990844...	279190	1069261	1159293	1129277
HYDERABAD	2018575	1924748	954	0	3943323	0	100	849051	3943323	4275351	4164675
JAGTIAL	484079	501338	1036	764081	221336	77.5400009155...	22.4599990844...	253619	985417	1068389	1040729
JANGAON	283648	282728	997	495019	71357	87.4000015258...	12.6000003814...	139238	566376	614065	598168
JAYASHANKAR ...	354203	357231	1009	657554	53880	92.4300003051...	7.57000017166...	189622	416763	451854	440155
JOGULAMBA G...	309274	300716	972	546813	63177	89.6399993896...	10.3599996566...	132261	609990	661351	644230
KAMAREDDY	478389	494236	1033	849003	123622	87.2900009155...	12.7100000381...	222513	972625	1054520	1027217
KARIMNAGAR	504620	501091	993	696727	308984	69.2799987792...	30.7199993133...	258485	1005711	1090392	1062159
KHAMMAM	699124	702515	1005	1084811	316828	77.4000015258...	22.6000003814...	382929	1401639	1519657	1480311
KOMARAM BH...	258197	257615	998	428828	86984	83.1399993896...	16.8600006103...	120420	515812	559243	544764
MAHABUBABAD	388058	386491	996	698173	76376	90.1399993896...	9.85999965667...	195889	774549	839766	818021
MAHABUBNAG...	745101	741676	995	1178574	308203	79.2699966430...	20.7299995422...	300508	919903	997359	971535
MANCHERIAL	408272	398765	977	453190	353847	56.1500015258...	43.8499984741...	206983	807037	874990	852333
MEDAK	378654	388774	1027	708574	58854	92.3300018310...	7.67000007629...	168677	767428	832045	810500
MEDCHAL	1246666	1193407	957	209828	2230245	8.60000038146...	91.4000015258...	593697	2440073	2645527	2577041
NAGARKURNO...	437986	423780	968	773936	87830	89.8099975585...	10.1899995803...	196261	893308	968525	943452
NALGONDA	818306	800110	978	1250113	368303	77.2399978637...	22.7600002288...	401728	1618416	1754687	1709256
NIRMAL	346721	362697	1046	557736	151682	78.6200027465...	21.3799991607...	165763	709418	769151	749234
NIZAMABAD	768477	802545	1044	1106272	464750	70.4199981689...	29.5799999237...	369031	1571022	1703302	1659206

Telangana Tourism Analysis.

```sql
/*Top 5 districts based on 'Population to Tourisst Footfall Ratio'*/

WITH CTE_Domestic_Tourists
AS (SELECT district,
           SUM(visitors) AS domestic_tourist
    FROM domestic_visitors
    WHERE domestic_visitors.[year] = 2019
    GROUP BY district
    ),
     CTE_Foreign_Tourists
AS (SELECT district,
           SUM(visitors) AS foreign_tourist
    FROM foreign_visitors
    WHERE foreign_visitors.[year] = 2019
    GROUP BY district
    )
SELECT TOP 5
    CTE_Domestic_Tourists.district,
    demographics.population_2019,
    domestic_tourist,
    foreign_tourist,
    domestic_tourist + foreign_tourist AS total_visitors,
    (domestic_tourist + foreign_tourist) / demographics.population_2019 AS ratio
FROM CTE_Domestic_Tourists
    JOIN CTE_Foreign_Tourists
        ON CTE_Domestic_Tourists.district = CTE_Foreign_Tourists.district
    JOIN demographics
        ON demographics.Districts = CTE_Foreign_Tourists.district
ORDER BY ratio DESC;
```

	district	population_2019	domestic_tourist	foreign_tourist	total_visitors	ratio
1	Rajanna Sircilla	583021	16832897	0	16832897	28
2	Bhadradri Koth...	1129277	12817737	0	12817737	11
3	Medak	810500	5452570	0	5452570	6
4	Nirmal	749234	3816778	0	3816778	5
5	Yadadri Bhongir	780952	4489374	0	4489374	5

Telangana Tourism Analysis.

```sql
/*Bottom 5 districts based on 'Population to Tourisst Footfall Ratio'*/

WITH CTE_Domestic_Tourists
AS (SELECT district,
           SUM(visitors) AS domestic_tourist
    FROM domestic_visitors
    WHERE domestic_visitors.[year] = 2019
    GROUP BY district
    ),
     CTE_Foreign_Tourists
AS (SELECT district,
           SUM(visitors) AS foreign_tourist
    FROM foreign_visitors
    WHERE foreign_visitors.[year] = 2019
    GROUP BY district
    )
SELECT TOP 5
    CTE_Domestic_Tourists.district,
    demographics.population_2019,
    domestic_tourist,
    foreign_tourist,
    domestic_tourist + foreign_tourist AS total_visitors,
    (domestic_tourist + foreign_tourist) / demographics.population_2019 AS ratio
FROM CTE_Domestic_Tourists
    JOIN CTE_Foreign_Tourists
        ON CTE_Domestic_Tourists.district = CTE_Foreign_Tourists.district
    JOIN demographics
        ON demographics.Districts = CTE_Foreign_Tourists.district
ORDER BY ratio;
```

RESULTS

	district	population_2019	domestic_tourist	foreign_tourist	total_visitors	ratio
1	Medchal	2577041	*NULL*	*NULL*	*NULL*	*NULL*
2	Ranga Reddy	2583577	*NULL*	*NULL*	*NULL*	*NULL*
3	Suryapet	1161280	*NULL*	0	*NULL*	*NULL*
4	Vikarabad	979180	*NULL*	*NULL*	*NULL*	*NULL*
5	Jangaon	598168	328890	0	328890	0

Telangana Tourism Analysis.

```sql
/*To find ratios that are not null*/

WITH CTE_Domestic_Tourists
AS (SELECT district,
            SUM(visitors) AS domestic_tourist
    FROM domestic_visitors
    WHERE domestic_visitors.[year] = 2019
    GROUP BY district
    ),
     CTE_Foreign_Tourists
AS (SELECT district,
            SUM(visitors) AS foreign_tourist
    FROM foreign_visitors
    WHERE foreign_visitors.[year] = 2019
    GROUP BY district
    )
SELECT TOP 5
    CTE_Domestic_Tourists.district,
    demographics.population_2019,
    domestic_tourist,
    foreign_tourist,
    domestic_tourist + foreign_tourist AS total_visitors,
    (domestic_tourist + foreign_tourist) / demographics.population_2019 AS ratio
FROM CTE_Domestic_Tourists
    JOIN CTE_Foreign_Tourists
        ON CTE_Domestic_Tourists.district = CTE_Foreign_Tourists.district
    JOIN demographics
        ON demographics.Districts = CTE_Foreign_Tourists.district
WHERE (domestic_tourist + foreign_tourist) / demographics.population_2019 IS NOT
NULL
ORDER BY ratio;
```

RESULTS

	district	population_2019	domestic_tourist	foreign_tourist	total_visitors	ratio
1	Jangaon	598168	328890	0	328890	0
2	Kamareddy	1027217	534	0	534	0
3	Karimnagar	1062159	77491	0	77491	0
4	Khammam	1480311	1413440	0	1413440	0
5	Komaram Bhee...	544764	19189	0	19189	0

```sql
--What will be the projected number of domestic and foreign tourists in Hyderabad in
2025? --

/*Projected domestic tourists in 2025 */

WITH CombinedVisitors
AS (SELECT 'domestic' AS visitor_type,
           district,
           year,
           visitors
    FROM domestic_visitors
    WHERE district = 'Hyderabad'
    UNION ALL
    SELECT 'foreign' AS visitor_type,
           district,
           year,
           visitors
    FROM foreign_visitors
    WHERE district = 'Hyderabad'
    ),
    ProjectedVisitors
AS (SELECT district,
           '2025' AS year,
           SUM(   CASE
                      WHEN visitor_type = 'domestic' THEN
                          visitors
                      ELSE
                          0
                  END
               ) AS projected_domestic_visitors,
           SUM(   CASE
                      WHEN visitor_type = 'foreign' THEN
                          visitors
                      ELSE
                          0
                  END
               ) AS projected_foreign_visitors
    FROM CombinedVisitors
    GROUP BY district
    )
SELECT district,
       projected_domestic_visitors,
       projected_foreign_visitors
FROM ProjectedVisitors;
```

	district	projected_dom...	projected_forei...
1	Hyderabad	83900960	1044898

THANK
YOU