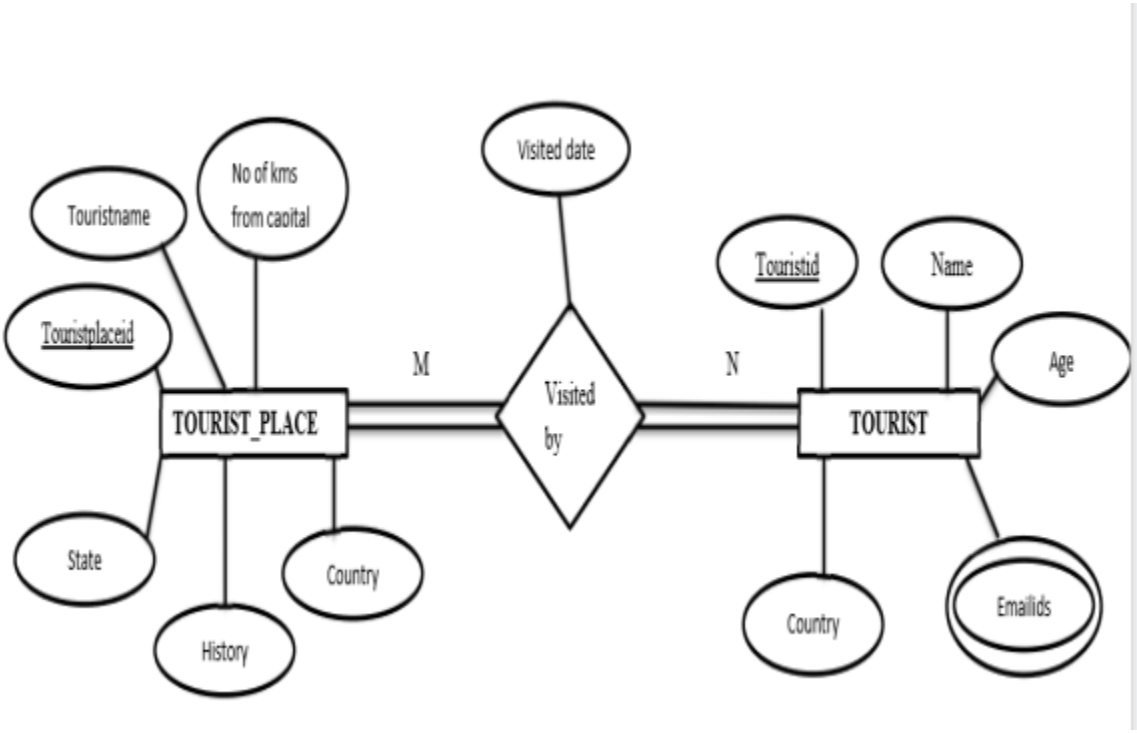
Exercise 4

Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries. A country can have many Tourist places . Each Tourist place is identified by using tourist_place_id, having a name, belonging to a state, Number of kilometers away from the capital city of that state,history. There are many Tourists visiting tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple email ids. A tourist visits many Tourist places, it is also required to record the visted_date in the database. A tourist can visit a Tourist place many times on different dates. A Tourist place can be visited by many tourists either on the same date or at different dates.

Queries:
1 List the state name which has the maximum number of tourist places.
2 List details of Tourist places where the maximum number of tourists visited.
3 List the details of tourists visiting all tourist places of the state “KARNATAKA”.
4 Display the details of the tourists who visited at least one tourist place of the state, but visited all state tourist places.
5 Display the details of the tourist place visited by the tourists of all country.



```
-- Table creation for Tourist places
CREATE TABLE TouristPlace (
    place_id INT PRIMARY KEY,
    name VARCHAR(255),
    state VARCHAR(255),
    distance_from_capital INT,
    history TEXT
);

-- Table creation for Tourists
CREATE TABLE Tourist (
    tourist_id INT PRIMARY KEY,
    name VARCHAR(255),
    age INT,
    country VARCHAR(255)
);

-- Table creation for Tourist visits
CREATE TABLE TouristVisits (
    visit_id INT PRIMARY KEY,
    tourist_id INT,
    place_id INT,
    visit_date DATE,
    FOREIGN KEY (tourist_id) REFERENCES Tourist(tourist_id),
    FOREIGN KEY (place_id) REFERENCES TouristPlace(place_id)
);



-- Inserting records into TouristPlace table
INSERT INTO TouristPlace (place_id, name, state, distance_from_capital, history)
VALUES (1, 'Tourist Place 1', 'KARNATAKA', 100, 'Historical Site 1');

INSERT INTO TouristPlace (place_id, name, state, distance_from_capital, history)
VALUES (2, 'Tourist Place 2', 'KARNATAKA', 150, 'Historical Site 2');

INSERT INTO TouristPlace (place_id, name, state, distance_from_capital, history)
```

```
VALUES (3, 'Tourist Place 3', 'KERALA', 200, 'Historical Site 3');

-- Inserting records into Tourist table
INSERT INTO Tourist (tourist_id, name, age, country)
VALUES (1, 'Tourist A', 25, 'India');

INSERT INTO Tourist (tourist_id, name, age, country)
VALUES (2, 'Tourist B', 30, 'USA');

-- Inserting records into TouristVisits table
INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (1, 1, 1, '2023-01-15');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (2, 1, 2, '2023-02-20');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (3, 2, 1, '2023-03-10');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (4, 2, 3, '2023-04-05');

-- Additional Tourist Places in Karnataka
INSERT INTO TouristPlace (place_id, name, state, distance_from_capital, history)
VALUES (4, 'Tourist Place 4', 'KARNATAKA', 120, 'Historical Site 4');

-- Additional Tourists
INSERT INTO Tourist (tourist_id, name, age, country)
VALUES (3, 'Tourist C', 22, 'India');

-- Additional Visits
INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (5, 3, 1, '2023-05-12');

-- Another State with Tourist Places
INSERT INTO TouristPlace (place_id, name, state, distance_from_capital, history)
VALUES (5, 'Tourist Place 5', 'TAMIL NADU', 180, 'Historical Site 5');
```

```
-- More Visits to Cover Different States
INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (6, 1, 5, '2023-06-20');

-- Another Tourist Visiting All Places in Karnataka
INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (7, 3, 4, '2023-07-05');

-- Inserting records into TouristVisits table to satisfy the 3rd query
-- Tourist C visiting all tourist places in KARNATAKA
INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (8, 3, 1, '2023-08-10');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (9, 3, 2, '2023-08-12');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (10, 3, 4, '2023-08-15');

-- Inserting records into TouristVisits table to satisfy the 4th query
-- Tourist A visiting all places in KARNATAKA
```

1: List the state name which has the maximum number of tourist places.

```
SELECT state FROM TouristPlace GROUP BY state ORDER BY COUNT(place_id) DESC
LIMIT 1;
```

mysql> select state,count(*) from TouristPlace group by state order by count(*) desc limit 1;

```
INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
```

```
VALUES (11, 1, 1, '2023-08-18');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (12, 1, 2, '2023-08-20');

INSERT INTO TouristVisits (visit_id, tourist_id, place_id, visit_date)
VALUES (13, 1, 4, '2023-08-25');
```

2: List details of tourist place where maximum number of tourist Visited.

```
select tv.place_id, count(*) ,tp.state from TouristVisits tv,TouristPlace tp where
tv.place_id=tp.place_id group by place_id having count(*) >= all(select count(*) from
TouristVisits group by place_id);
```

3:List details of tourists visiting all tourist places of the KARNATAKA. List out the tourist places in karnataka

-- List the details of tourists visiting all tourist places of the state “KARNATAKA” using ALL clause

```
SELECT tourist_id, name, age, country
FROM Tourist
WHERE tourist_id IN (
   SELECT DISTINCT TV.tourist_id
   FROM TouristVisits TV
     WHERE TV.place_id IN (SELECT place_id FROM TouristPlace WHERE state =
'KARNATAKA')
   GROUP BY TV.tourist_id
        HAVING COUNT(DISTINCT TV.place_id) = (SELECT COUNT(*) FROM
TouristPlace WHERE state = 'KARNATAKA')
);
```

4. Display the details of the tourists who visited atleast one tourist place of the state, but visited all states tourist places.

-- Display the details of the tourists who visited at least one tourist place of the state but visited all state tourist places.

```
SELECT distinct tv.tourist_id, t.name, t.age, t.country
FROM Tourist t
JOIN TouristVisits tv ON t.tourist_id = tv.tourist_id
WHERE EXISTS (
   SELECT tp.place_id
   FROM TouristPlace tp
   WHERE tp.state = 'KARNATAKA'
   EXCEPT
   SELECT tv2.place_id
   FROM TouristVisits tv2
   WHERE tv2.tourist_id = t.tourist_id
);
```

mysql> select touristid,count(distinct tplaceid) from tourist_visited_places group by touristid having count(distincttplaceid)>=(select count(distinct state) from touristplace);

5.Display the details of the tourist place visited by the tourists of all country.

```
SELECT tp.place_id, tp.name, tp.state FROM TouristPlace tp WHERE NOT EXISTS ( SELECT DISTINCT country FROM Tourist WHERE NOT EXISTS ( SELECT tv.tourist_id FROM TouristVisits tv WHERE tv.tourist_id = Tourist.tourist_id AND tv.place_id = tp.place_id ) );
```