```
CREATE TABLE category_header (
   Cat_code VARCHAR(2) PRIMARY KEY,
   Cate_desc VARCHAR(20)
);

INSERT INTO category_header VALUES
('01', 'super delux'),
('02', 'delux'),
('03', 'super fast'),
('04', 'normal');

CREATE TABLE route_header (
   Route_id INT PRIMARY KEY,
   Route_no INT,
   Cate_code VARCHAR(2),
   Origin VARCHAR(20),
   Destination VARCHAR(20),
   Fare DECIMAL(5,2),
   Distance DECIMAL(6,2),
   Capacity INT,
   FOREIGN KEY (Cate_code) REFERENCES category_header(Cat_code)
);

INSERT INTO route_header VALUES
(101, 33, '01', 'Madurai', 'Madras', 35, 250, 50),
(102, 25, '02', 'Trichy', 'Madurai', 40, 159, 50),
(103, 15, '03', 'Thanjavur', 'Madurai', 59, 140, 50),
(104, 36, '04', 'Madras', 'Banglore', 79, 375, 50),
(105, 40, '01', 'Banglore', 'Madras', 80, 375, 50),
(106, 38, '02', 'Madras', 'Madurai', 39, 250, 50),
(107, 39, '03', 'Hydrabad', 'Madras', 50, 430, 50),
(108, 41, '04', 'Madras', 'Cochin', 47, 576, 50);

CREATE TABLE place_header (
   Place_id VARCHAR(2) PRIMARY KEY,
   Place_name VARCHAR(20),
   Place_address VARCHAR(50),
   Bus_station VARCHAR(20)
);

INSERT INTO place_header (Place_id, Place_name, Place_address, Bus_station) VALUES
('01', 'Madras', '10, ptc road', 'Parrys'),
('02', 'Madurai', '21, canal bank road', 'Kknagar'),
('03', 'Trichy', '11, first cross road', 'Bhel town');
```

```
CREATE TABLE fleet_header (
   Fleet_id VARCHAR(2),
   Day DATE,
   Route_id INT,
   Cat_code VARCHAR(2),
   PRIMARY KEY (Fleet_id, Day),
   FOREIGN KEY (Route_id) REFERENCES route_header(Route_id),
   FOREIGN KEY (Cat_code) REFERENCES category_header(Cat_code)
);

INSERT INTO fleet_header (Fleet_id, Day, Route_id, Cat_code) VALUES
('01', '1996-04-10', 101, '01'),
('02', '1996-04-10', 101, '01'),
('03', '1996-04-10', 101, '01'),
('04', '1996-04-10', 102, '02'),
('05', '1996-04-10', 102, '02'),
('06', '1996-04-10', 103, '03');

CREATE TABLE ticket_header (
   Fleet_id VARCHAR(2),
   Ticket_no VARCHAR(5),
   Doi DATE,
   Dot DATE,
   Time_travel TIME,
   Board_place VARCHAR(20),
   Origin VARCHAR(20),
   Destination VARCHAR(20),
   Adults INT,
   Children INT,
   Total_fare DECIMAL(6,2),
   Route_id INT,
   PRIMARY KEY (Fleet_id, Ticket_no),
   FOREIGN KEY (Fleet_id) REFERENCES fleet_header(Fleet_id),
   FOREIGN KEY (Route_id) REFERENCES route_header(Route_id)
);

INSERT INTO ticket_header (Fleet_id, Ticket_no, Doi, Dot, Time_travel, Board_place, Origin,
Destination, Adults, Children, Total_fare, Route_id) VALUES
('01', '01', '1996-04-10', '1996-05-10', '15:00:00', 'Parrys', 'Madras', 'Madurai', 1, 1, 60, 101),
('02', '02', '1996-04-12', '1996-05-05', '09:00:00', 'Kknagar', 'Madurai', 'Madras', 1, 1, 60, 102),
('03', '03', '1996-04-21', '1996-05-15', '21:00:00', 'Cubbon park', 'Banglore', 'Madras', 2, 0, 400,
101);
```

```
CREATE TABLE ticket_detail (
   Ticket_no VARCHAR(5),
   Name VARCHAR(20),
   Sex CHAR(1),
   Age INT,
   Fare DECIMAL(6,2),
   PRIMARY KEY (Ticket_no, Fleet_id, Name),
   FOREIGN KEY (Ticket_no, Fleet_id) REFERENCES ticket_header(Ticket_no, Fleet_id)
);

INSERT INTO ticket_detail (Ticket_no, Name, Sex, Age, Fare) VALUES
('01', 'Charu', 'F', 24, 14.00),
('01', 'Lathu', 'F', 10, 15.55),
('02', 'Anand', 'M', 28, 17.80),
('02', 'Guatham', 'M', 28, 16.00),
('03', 'Bala', 'M', 28, 17.65),
('05','Sandip', 'M', 30, 18.00);



CREATE TABLE route_details (
route_id INT,
place_id INT,
nonstop CHAR(1) NOT NULL CHECK (nonstop IN ('N', 'S')),
FOREIGN KEY (route_id) REFERENCES route_header(Route_id),
FOREIGN KEY (place_id) REFERENCES place_header(Place_id)
);

INSERT INTO route_details (route_id, place_id, nonstop) VALUES
(105, 1, 'N'),
(012, 2, 'S'),
(106, 1, 'S'),
(108, 5, 'N'),
(106, 2, 'N');
```

Here are the SQL queries for each of your requests:

1. Display only those routes that originate in “Madras” and terminate in “Cochin”.
```sql
SELECT *
FROM route_header
WHERE Origin = 'Madras' AND Destination = 'Cochin';
```

2. Display only those rows from `route_header` whose origin begins with ‘m’.
```sql
SELECT *
FROM route_header
WHERE Origin LIKE 'M%';
```

3. Display only those rows whose fare ranges between 30 and 50.
```sql
SELECT *
FROM route_header
WHERE Fare BETWEEN 30 AND 50;
```

4. Display the fare and the origin for `route_no` which are greater than 15.
```sql
SELECT Fare, Origin
FROM route_header
WHERE Route_no > 15;
```

5. Display those routes whose distance is in range of 200 and 400.
```sql
SELECT *
FROM route_header
WHERE Distance BETWEEN 200 AND 400;
```

6. Find out fleets which travel through route 102 or 103.
```sql
SELECT *
FROM fleet_header
WHERE Route_id IN (102, 103);
```

7. Find out routes which are non-stop.
```sql
SELECT *
FROM route_details
WHERE nonstop = 'S';
```

8. Arrange the `route_id` record in ascending order.
```sql
SELECT *
FROM route_header
ORDER BY Route_id ASC;
```

9. Find out category whose category description starts with ‘s’ and ends with ‘t’.
```sql
SELECT *
FROM category_header
WHERE Cate_desc LIKE 's%t';
```

10. Find out routes which have category code 1, 2, or 4.
```sql
SELECT *
FROM route_header
WHERE Cate_code IN ('01', '02', '04');
```

11. Display details of place with bus station “charminar”.
```sql
SELECT *
FROM place_header
WHERE Bus_station = 'Charminar';
```

12. Display details of those routes whose fare is less than 70 and distance greater than 120.
```sql
SELECT *
FROM route_header
WHERE Fare < 70 AND Distance > 120;
```

13. Find out details of tickets issued to female travelers and with age greater than 10.

```sql
SELECT *
FROM ticket_detail
WHERE Sex = 'F' AND Age > 10;
```

14. What will be fare of each route after incrementing fare by 10 percent?
```sql
SELECT Route_id, Fare, Fare * 1.10 AS New_Fare
FROM route_header;
```

15. Find out details of routes with `route_id` 101 or 105 or 107.
```sql
SELECT *
FROM route_header
WHERE Route_id IN (101, 105, 107);
```

16. Display those routes for which origin is “Madras” and distance is greater than 300 or destination is “Madras” and distance less than 300.
```sql
SELECT *
FROM route_header
WHERE (Origin = 'Madras' AND Distance > 300) OR (Destination = 'Madras' AND Distance < 300);
```

17. Write a query to display all `Bus_station` names in uppercase.
```sql
SELECT UPPER(Bus_station)
FROM place_header;
```

18. Write a query to print “mpstme” into uppercase.
```sql
SELECT UPPER('mpstme') AS UpperCaseValue;
```

19. Write a query to display category description of those categories for which `Cat_code` is either 01, 02, or 04. Display category description with the first character in uppercase and remaining characters in lowercase.
```sql
```

```sql
SELECT CONCAT(UPPER(SUBSTRING(Cate_desc, 1, 1)), LOWER(SUBSTRING(Cate_desc, 2))) AS Formatted_Cate_desc
FROM category_header
WHERE Cat_code IN ('01', '02', '04');
```

20. Write a query to concatenate and display `Place_name` and `Place_address` columns of `place_header` table.
```sql
SELECT CONCAT(Place_name, ', ', Place_address) AS Place_Details
FROM place_header;
```

21. Write a query to display `route_id` along with the substring “MAD” from its destination column.
```sql
SELECT Route_id, SUBSTRING(Destination, 1, 3) AS Substr_Destination
FROM route_header;
```

22. Write a query to display category code along with the total number of characters for category description.
```sql
SELECT Cat_code, LENGTH(Cate_desc) AS Description_Length
FROM category_header;
```

23. Write a query to display fare of `ticket_detail` table with a total 15-character space and padding of ‘*’ on the left side.
```sql
SELECT LPAD(Fare, 15, '*') AS Padded_Fare
FROM ticket_detail;
```

24. Write a query to display fare of `ticket_detail` table with a total 15-character space and padding of ‘*’ on the right side.
```sql
SELECT RPAD(Fare, 15, '*') AS Padded_Fare
FROM ticket_detail;
```

25. Write a query to round fare from `ticket_detail` column up to one decimal point.
```sql
SELECT ROUND(Fare, 1) AS Rounded_Fare
```

```
FROM ticket_detail;
```

26. Write a query to find the system date.
```sql
SELECT CURRENT_DATE() AS System_Date;
```

27. Write a query to display `fleet_id`, `ticket_id`, `origin`, `destination`, and `dot` column of `ticket_header` after adding 6 months to it for those records for which the number of adults traveling is more than one and the number of children traveling is greater than 0.
```sql
SELECT Fleet_id, Ticket_no, Origin, Destination, DATE_ADD(Dot, INTERVAL 6 MONTH) AS Updated_Dot
FROM ticket_header
WHERE Adults > 1 AND Children > 0;
```

28. Write a query to find the next “Tuesday” after `sysdate`.
```sql
SELECT DATE_ADD(CURRENT_DATE(), INTERVAL (9 - WEEKDAY(CURRENT_DATE())) % 7 DAY) AS Next_Tuesday;
```

These queries should help you retrieve and manipulate the data according to your requirements.