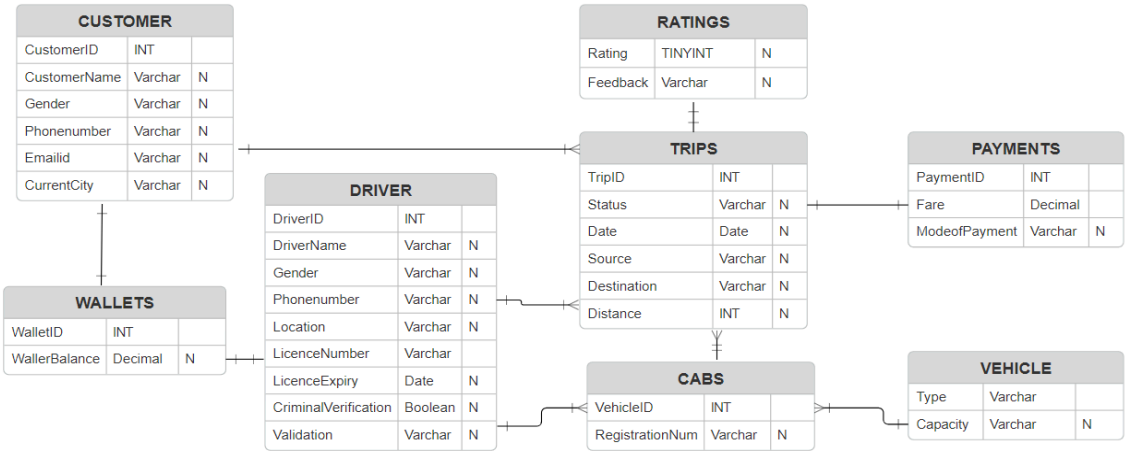
TEAM - SYNERGY

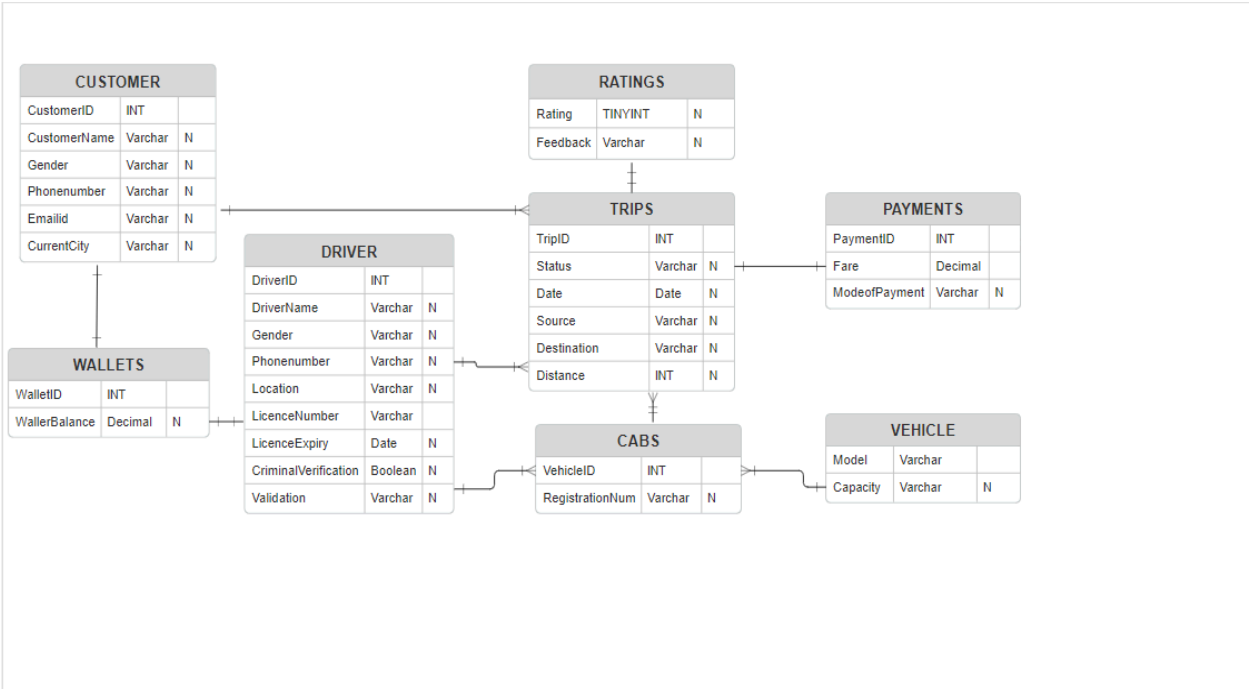
ASSIGNMENT

Task 1 :Create a conceptual and logical and physical data model using ER diagram for UBER (Ride sharing app).

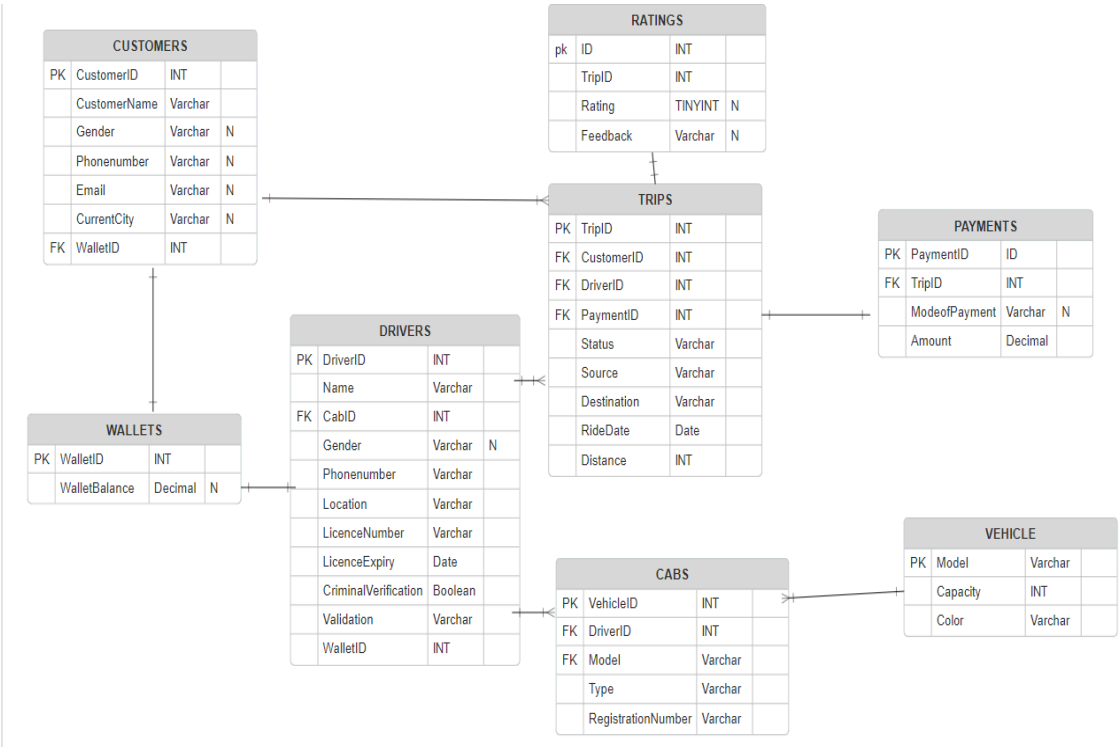
CONCEPTUAL MODEL:



LOGICAL MODEL:



PHYSICAL MODEL:



Task 2 : Create tables from the physical data model from above task.(Create appropriate constraints and keys)

RATING TABLE:

```
CREATE TABLE rating(
id INT PRIMARY KEY,
tripID INT not null,
rating TINYINT,
feedback VARCHAR(1000),
CHECK (rating<=5),
CHECK (rating>=0)
)
```

CABS:

```
CREATE TABLE cabs(
vehicleId INT PRIMARY KEY,
driverId INT not null,
model VARCHAR(30) not null,
TYPE VARCHAR(10) not null CHECK(TYPE in ('sedan','suv','hatchback')),
registrationNumber VARCHAR(10)
```

)

CUSTOMERS:

```
CREATE TABLE customers(
id INT PRIMARY KEY,
name VARCHAR(50) not null,
gender VARCHAR(15),
phoneNumber VARCHAR(15),
email VARCHAR(80),
currentCity VARCHAR(30),
walletId INT
)
```

PAYMENTS:

```
CREATE TABLE payments(
id INT PRIMARY KEY,
tripId INT not null,
modeOfPayment VARCHAR(10) CHECK(modeOfPayment in ('cash','upi','card')),
amount FLOAT(24) not null
)
```

DRIVERS:

```
CREATE TABLE drivers(
id INT PRIMARY KEY,
NAME VARCHAR(50) not null,
cabId INT not null,
gender VARCHAR(15),
phoneNumber VARCHAR(15) not null,
location VARCHAR(100) not null,
licenceNumber VARCHAR(20)UNIQUE not null,
licenceExpiry DATE not null,
validation VARCHAR(30) not null,
walletId INT
)
```

TRIPS:

```
CREATE TABLE trips(
id INT PRIMARY KEY,
customerId INT not null,
driverId INT not null,
paymentId INT not null,
status VARCHAR(10) not null CHECK(status in ('accepted','waiting','cancelled')),
source VARCHAR(100) not null,
destination VARCHAR(100) not null,
rideDate DATE not null,
distance INT not null
)
```

WALLET:

```
CREATE TABLE wallet(
id INT PRIMARY KEY,
balance FLOAT
)
```

VEHICLE:

```
CREATE TABLE vehicle(
model VARCHAR(30) PRIMARY KEY,
capacity INT not null,
color VARCHAR(20)
)

--making sure of the referential integrity
ALTER TABLE customers
ADD FOREIGN KEY(walletId) REFERENCES wallet(id)

ALTER TABLE drivers
ADD FOREIGN KEY(walletId) REFERENCES wallet(id)

ALTER TABLE cabs
ADD FOREIGN KEY (model) REFERENCES vehicle(model);


ALTER TABLE ratings
ADD FOREIGN KEY (tripID) REFERENCES trips(id);

ALTER TABLE cabs
ADD FOREIGN KEY (driverID) REFERENCES drivers(id);

ALTER TABLE trips
ADD FOREIGN KEY (customerId) REFERENCES customers(id);

ALTER TABLE trips
ADD FOREIGN KEY (driverId) REFERENCES drivers(id);

ALTER TABLE trips
ADD FOREIGN KEY (paymentId) REFERENCES payments(id);
```

Task 3: Populate all the tables created with sample data.

```
INSERT INTO wallet VALUES
(5000,50000),
(7000,14000),
(6000,12000),
(1500,30000),
(2000,40000),
```

(1800,360000);

INSERT INTO customers (id, name, gender, phoneNumber, email, currentCity, walletId) VALUES
(1, 'Aarav Sharma', 'Male', '9876543210', 'aarav.sharma@example.com', 'Mumbai', 5000),
(2, 'Diya Patel', 'Female', '8765432109', 'diya.patel@example.com', 'Delhi', 7000),
(3, 'Arjun Singh', 'Male', '7654321098', 'arjun.singh@example.com', 'Bangalore', 6000);

INSERT INTO drivers (id, name, cabId, gender, phoneNumber, location, licenceNumber, licenceExpiry, validation, walletId) VALUES
(1, 'Rohan Verma', 1, 'Male', '9876543210', 'Mumbai', 'MH123456789012', '2024-12-31', 'Valid', 1500),
(2, 'Anaya Gupta', 2, 'Female', '8765432109', 'Delhi', 'DL987654321098', '2023-11-30', 'Valid', 2000),
(3, 'Kabir Kumar', 3, 'Male', '7654321098', 'Bangalore', 'KA654321098765', '2025-05-31', 'Valid', 1800);

INSERT INTO vehicle (model,capacity,color) VALUES
('maruti swift',4,'black'),
('alto 500',3,'white'),
('s-class',2,'white');

INSERT INTO cabs (vehicleId, driverId, type, registrationNumber,model) VALUES
(1, 1, 'sedan', 'MH01AB1234','maruti swift'),
(2, 2, 'suv', 'KA02CD5678','alto 500'),
(3, 3, 'hatchback', 'DL03EF9101','s-class');

INSERT INTO payments (id, tripId, modeOfPayment, amount) VALUES
(1, 1, 'card', 300),
(2, 2, 'cash', 250),
(3, 3, 'upi', 200);

INSERT INTO trips (id, customerId, driverId, paymentId, status, source, destination, rideDate, distance) VALUES
(1, 1, 1, 1, 'accepted', 'Mumbai Central', 'Bandra Kurla Complex', '2024-02-08', 15),
(2, 2, 2, 2, 'waiting', 'Connaught Place', 'India Gate', '2024-02-09', 10),
(3, 3, 3, 3, 'cancelled', 'Electronic City', 'Koramangala', '2024-02-10', 8);

INSERT INTO rating (id, tripID, rating, feedback) VALUES
(1, 1, 4, 'Good service'),
(2, 2, 5, 'Excellent ride experience'),
(3, 3, 3, 'Average trip');

Task 4 : Write a stored routine to update the wallet balance of a user.(2 incoming argument i.e. user and amount. (Print or return) updated wallet balance)

```
Go
CREATE PROCEDURE AddMon4(
   @holder int,
   @fair int
)
AS
```

```
BEGIN
    -- Update Column1 by adding the specified value
    UPDATE wallet
    SET Balance = Balance + @fair
    WHERE id = (SELECT walletid from customers where id=@holder);
            select balance from wallet where id=(SELECT walletid from customers where id=@holder)
END;
```

OUTPUT:

EXEC [dbo].[AddMon4]1,-20

The id selected here is 1 which has the wallet id 5000

The balance before execution:

	id	balance
1	1500	30000
2	1800	360000
3	2000	40000
4	5000	49976
5	6000	12000
6	7000	14000

The Balance after execution:

	id	balance
1	1500	30000
2	1800	360000
3	2000	40000
4	5000	49956
5	6000	12000
6	7000	14000

Task 5 : Write the correct SQL execution order for the following SELECT query.

```
SELECT DISTINCT column, AGG_FUNC(column_or_expression), ...
FROM mytable
JOIN another_table ON mytable.column = another_table.column
WHERE constraint_expression
```

```
GROUP BY column
HAVING constraint_expression
ORDER BY column ASC/DESC
LIMIT count OFFSET COUNT;
```

The order of execution of an SQL query is as follows:
1)FROM
2)JOIN
3)WHERE
4)GROUP BY
5)HAVING
6)SELECT
7)DISTINCT
8)ORDER BY
9)LIMIT
10)OFFSET

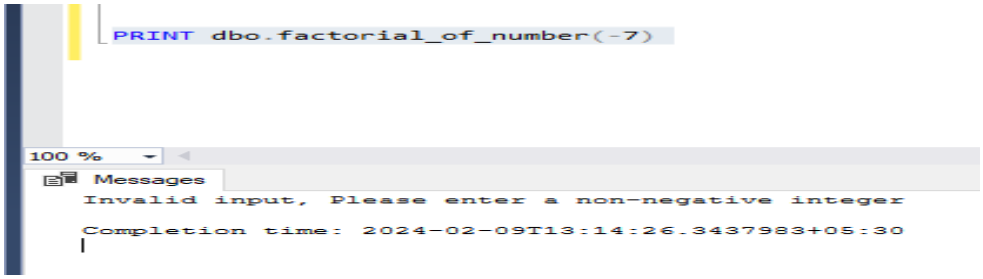
Task 6: Write a simple function to calculate factorial of any number. (function accepts an IN parameter and returns the factorial of that).

```
CREATE FUNCTION factorial_of_number(@numb BIGINT)
RETURNS VARCHAR(255)
AS
BEGIN
   IF @numb < 0
   BEGIN
      -- Return an error message for invalid input
      RETURN 'Invalid input, Please enter a non-negative integer';
   END
   DECLARE @ans BIGINT = 1;
   DECLARE @counter BIGINT = 2;
   DECLARE @fact VARCHAR(255);
   WHILE @counter <= @numb
   BEGIN
      SET @ans = @ans * @counter;
      SET @counter = @counter + 1;
   END
   -- Convert the result to VARCHAR before returning
   SET @fact = 'The factorial of the given number is: ' + CAST(@ans AS VARCHAR(255));
   RETURN @fact;
```

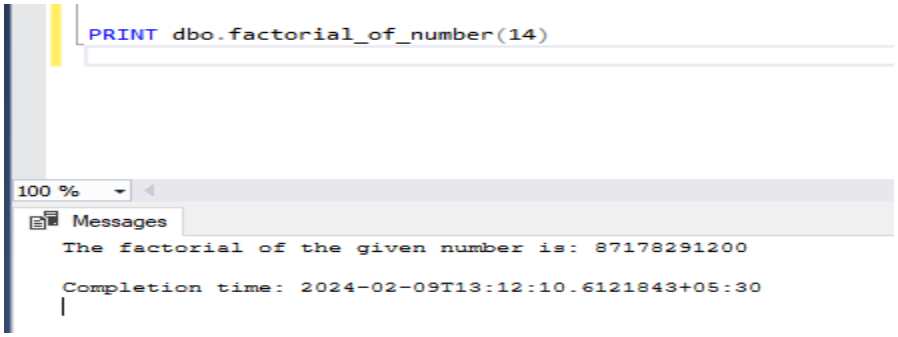
END
PRINT dbo.factorial_of_number(14)

Task 7: Execute the function with multiple values as a test case.

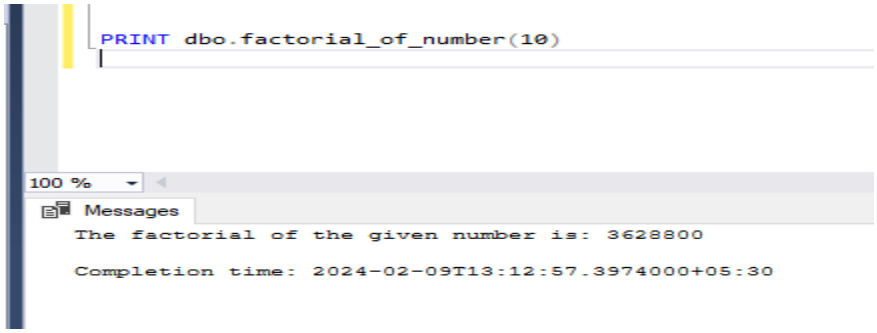
Testcase-1:



Testcase-2:



Testcase-3:



Testcase-4:

```
PRINT dbo.factorial_of_number(18)
```

```
100 %
Messages
The factorial of the given number is: 6402373705728000

Completion time: 2024-02-09T13:13:54.2858995+05:30
```

Testcase-5:

```
PRINT dbo.factorial_of_number(-7)
```

```
100 %
Messages
Invalid input, Please enter a non-negative integer

Completion time: 2024-02-09T13:14:26.3437983+05:30
```

Task 8:

1.) What is a tuple equivalent to in SQL?

Ans : A row in a Table

2.) How many NULL values can a Unique key can have?

Ans: 1

3.) Which Join is used to get only match tuples?

Ans: Inner Join

4.) Which is the kind of Aggregate function?

Ans : Min

5.) Which are the TRANSACTION control commands?

Ans: All the Above

6.) When a program is abnormally terminated in a transaction,which of the following commands occurs?

Ans: Rollback

7.) What is wrong with the following query?

Select V_ID,P_ID,P_DESC,P_RATE rate FROM TABLE1 GROUP BY V_ID

Ans: No Aggregate function is used

- AKSHAT KOTWALLA
- KUSAMPUDI LALITHA ARADHINI
- KOMMURI DIVYA PHANI SREE
- BILLA PRAHAS REDDY