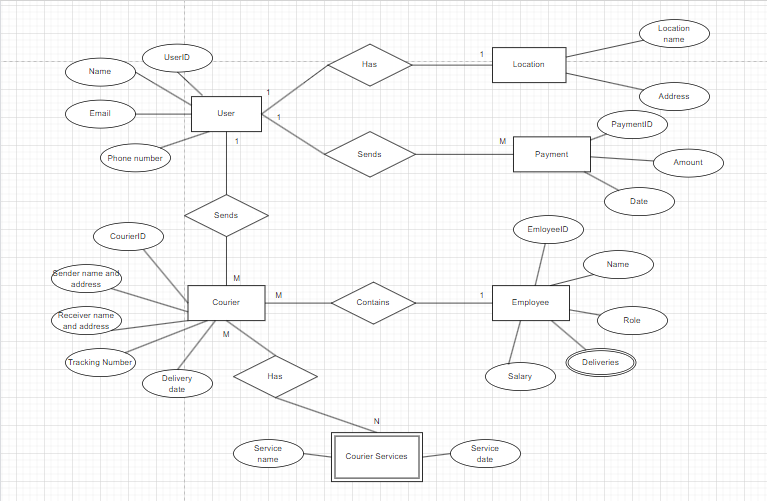
**COURIER MANAGEMENT SYSTEM**

TASK 1:

ER DIAGRAM:

SQL queries for creating tables:

CREATE TABLE user(

userid INT PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255) UNIQUE,

password VARCHAR(255),

contact VARCHAR(20),

address TEXT);

CREATE TABLE courier(

courierid INT PRIMARY KEY,

sender VARCHAR(255),

senderadrs TEXT,

receiver VARCHAR(255),

receiveradrs TEXT,

weight DECIMAL(5,2),

status VARCHAR(50),

trackingno VARCHAR(20) UNIQUE,

deliverydate DATE);

CREATE TABLE courierservices(

serviceid INT PRIMARY KEY,

servicename VARCHAR(100),

cost DECIMAL(8,2));

CREATE TABLE employee(

empid INT PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255) UNIQUE,

contact VARCHAR(20),

role VARCHAR(50),

salary DECIMAL(10,2));

CREATE TABLE location(

locationid INT PRIMARY KEY,

locationname VARCHAR(100),

address TEXT);

CREATE TABLE payment(

paymentid INT PRIMARY KEY,

courierid INT,

locationid INT,

amount DECIMAL(10,2),

paymentdate DATE,

FOREIGN KEY(courierid) REFERENCES courier(courierid),

FOREIGN KEY(locationid) REFERENCES location(locationid));

Entering data into tables:

INSERT INTO user VALUES

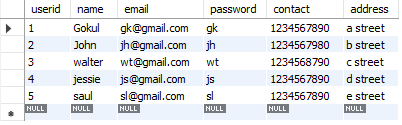
(1,'Gokul','gk@gmail.com','gk','1234567890','a street'),

(2,'John','jh@gmail.com','jh','1234567980','b street'),

(3,'walter','wt@gmail.com','wt','1234568790','c street'),

(4,'jessie','js@gmail.com','js','1234567890','d street'),

(5,'saul','sl@gmail.com','sl','1234567890','e street');



INSERT INTO courier VALUES

(101,'walter','c street','Gokul','a street',565,'undelivered','1234','2023-12-21'),

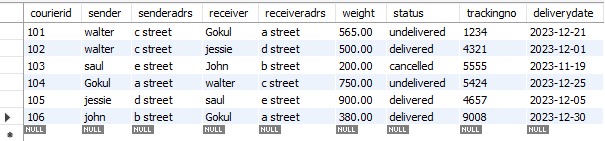
(102,'walter','c street','jessie','d street',500,'delivered','4321','2023-12-01'),

(103,'saul','e street','John','b street',200,'cancelled','5555','2023-11-19'),

(104,'Gokul','a street','walter','c street',750,'undelivered','5424','2023-12-25'),

('105', 'jessie', 'd street', 'saul', 'e street', '900.00', 'delivered', '4657', '2023-12-05'),

('106', 'john', 'b street', 'Gokul', 'a street', '380.00', 'delivered', '9008', '2023-12-30');

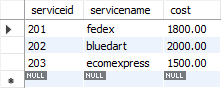


INSERT INTO courierservices VALUES

(201,'fedex',1800),

(202,'bluedart',2000),

(203,'ecomexpress',1500);



INSERT INTO employee VALUES

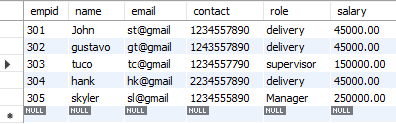
(301,'john','st@gmail','1234557890','delivery',45000),

(302,'gustavo','gt@gmail','1243557890','delivery',45000),

(303,'tuco','tc@gmail','1234557790','supervisor',150000),

(304,'hank','hk@gmail','2234557890','delivery',45000),

(305,'skyler','sl@gmail','1234555890','Manager',250000);



INSERT INTO location VALUES

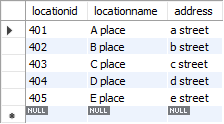
(401,'A place','a street'),

(402,'B place','b street'),

(403,'C place','c street'),

(404,'D place','d street'),

(405,'E place','e street');



INSERT INTO payment VALUES

(501,101,401,5650,'2023-12-21'),

(502,102,404,5000,'2023-12-01'),

(503,103,402,2000,'2023-11-19'),

(504,104,403,7500,'2023-12-25');

INSERT INTO payment (`paymentid`, `courierid`, `locationid`, `amount`, `paymentdate`) VALUES ('505', '106', '401', '3800.00', '2023-12-30');

UPDATE payment SET `courierid` = '104', `locationid` = '403', `amount` = '7500.00', `paymentdate` = '2023-12-25' WHERE (`paymentid` = '503');

UPDATE payment SET `courierid` = '105', `locationid` = '405', `amount` = '9000.00', `paymentdate` = '2023-12-05' WHERE (`paymentid` = '504');

ALTER TABLE payment ADD COLUMN paymenttime time;

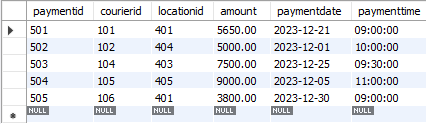
UPDATE `payment` SET `paymenttime` = '09:00:00' WHERE (`paymentid` = '501');

UPDATE `payment` SET `paymenttime` = '10:00:00' WHERE (`paymentid` = '502');

UPDATE `payment` SET `paymenttime` = '09:30:00' WHERE (`paymentid` = '503');

UPDATE `payment` SET `paymenttime` = '11:00:00' WHERE (`paymentid` = '504');

UPDATE `payment` SET `paymenttime` = '09:00:00' WHERE (`paymentid` = '505');



TASK 2:

1. List all customers:

SELECT name

FROM user;

1. List all orders for a specific customer:

SELECT courierid, receiver

FROM courier

WHERE sender='walter';

1. List all couriers:

SELECT courierid,sender,receiver,weight,status

FROM courier;

1. List all deliveries for a specific courier:

SELECT courierid,sender,receiver,status

FROM courier

WHERE status='delivered';

1. List all undelivered packages:

SELECT \* FROM courier

WHERE status='undelivered';

1. List all packages that are scheduled for delivery today:

SELECT \* FROM courier

WHERE deliverydate='2023-12-07';

1. List all packages with a specific status:

SELECT \* from courier

WHERE status !='delivered'

AND

status !='undelivered';

1. Calculate the total number of packages for each courier.

SELECT

courierid 'ID',sender 'User',weight 'No of Packages'

FROM courier;

1. Find the average delivery time for each courier

SELECT courierid,AVG(paymenttime) 'Average delivery time'

FROM payment

GROUP BY courierid;

1. List all packages with a specific weight range:

SELECT courierid,weight

FROM courier

WHERE weight>500;

1. Retrieve employees whose names contain 'John'

SELECT empid,name

FROM employee

WHERE name like'john';

1. Retrieve all courier records with payments greater than $500.

SELECT paymentid,amount

FROM payment

WHERE amount>500;

TASK 3:

14. Find the total number of couriers handled by each employee.

SELECT p.paymentid,p.courierid,e.empid,e.name,count(p.paymentid) 'Number of couriers'

FROM payment p INNER JOIN employee e

ON p.paymentid=e.empid

GROUP BY empid;

15. Calculate the total revenue generated by each location

SELECT locationid,sum(amount) 'total revenue'

FROM payment

GROUP BY locationid;

16. Find the total number of couriers delivered to each location.

SELECT courierid,locationid,COUNT(courierid) 'total couriers'

FROM payment

GROUP BY paymentid;

17. Find the courier with the highest average delivery time:

SELECT p1.paymentid,p1.courierid,p1.paymenttime

FROM payment p1 join payment p2

GROUP BY p1.paymentid

HAVING p1.paymenttime>AVG(p2.paymenttime);

18. Find Locations with Total Payments Less Than a Certain Amount

SELECT locationid,amount

FROM payment

HAVING amount < '5000';

19. Calculate Total Payments per Location

SELECT locationid,count(locationid) 'total payments'

FROM payment

GROUP BY locationid;

20. Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID = X):

SELECT courierid,locationid,amount

FROM payment

HAVING amount>'1000'

AND locationid='401'

ORDER BY couriered

21. Retrieve couriers who have received payments totaling more than $1000 after a certain date (PaymentDate > 'YYYY-MM-DD'):

SELECT courierid,paymentdate

FROM payment

WHERE amount>'1000'

AND paymentdate>'2023-12-01'

ORDER BY couriered.

22. Retrieve locations where the total amount received is more than $5000 before a certain date (PaymentDate > 'YYYY-MM-DD')

SELECT locationid,SUM(amount) 'total amount'

FROM payment WHERE paymentdate<'2023-12-01'

AND (SELECT SUM(amount) FROM payment)>'5000'

GROUP BY locationid;

TASK 4:

23. Retrieve Payments with Courier Information:

SELECT p.paymentid,p.amount,c.courierid,c.sender,c.receiver,c.weight,c.trackingno

FROM payment p INNER JOIN courier c

ON p.courierid =c.courierid

ORDER BY paymentid;

24. Retrieve Payments with Location Information:

SELECT p.paymentid,p.amount,l.locationid,l.locationname,l.address

FROM payment p INNER JOIN location l

ON p.locationid=l.locationid

ORDER BY locationid;

25. Retrieve Payments with Courier and Location Information:

SELECT p.paymentid,p.locationid,c.trackingno,c.weight,c.receiveradrs'Location', p.paymentdate'date'

FROM payment p LEFT JOIN courier c

ON c.courierid=p.courierid

GROUP BY paymentid;

26. List all payments with courier details:

SELECT p.paymentid,p.amount,c.courierid,c.sender,c.receiver,c.weight,c.trackingno

FROM payment p NATURAL JOIN courier c

ORDER BY paymentid;

27. Total payments received for each courier:

SELECT c.courierid,COUNT(p.paymentid) 'total payments'

FROM courier c INNER JOIN payment p

ON c.courierid=p.courierid

GROUP BY courierid;

28. List payments made on a specific date:

SELECT paymentid,amount,paymentdate

FROM payment WHERE paymentdate='2023-12-05'

29. Get Courier Information for Each Payment :

SELECT c.courierid,p.paymentid,c.sender,c.receiver,c.status,c.deliverydate

FROM courier c RIGHT JOIN payment p

ON c.courierid=p.courierid

ORDER BY paymentid;

30. Get Payment Details with Location:

SELECT p.paymentid,p.amount,p.paymentdate,p.paymenttime,l.locationid,l.locationname

FROM payment p NATURAL JOIN location l

ORDER BY locationid;

31. Calculating Total Payments for Each Courier:

SELECT c.courierid,c.sender,c.receiver,p.paymentid,COUNT(sender) 'total payments'

FROM payment p NATURAL JOIN courier c

GROUP BY paymentid;

32. List Payments Within a Date Range:

SELECT paymentid,paymenttime,paymentdate

FROM payment

WHERE paymentdate

BETWEEN '2023-12-01' AND '2023-12-31';

33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side:

SELECT u.userid,u.name,c.sender,c.receiver,c.status

FROM user u INNER JOIN courier c

ON u.name=c.sender

ORDER BY userid

34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side

SELECT c.courierid,s.serviceid,s.servicename,c.status

FROM courier c INNER JOIN courierservices s

ON c.courierid=s.serviceid

ORDER BY courierid;

35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side:

SELECT e.empid,p.paymentid,e.name'Employee name',p.amount,p.paymentdate

FROM employee e INNER JOIN payment p

ON e.empid=p.paymentid

ORDER BY empid;

36. List all users and all courier services, showing all possible combinations.

SELECT u.userid,u.name,s.serviceid,s.servicename

FROM user u INNER JOIN courierservices s

ORDER BY userid;

37. List all employees and all locations, showing all possible combinations:

SELECT e.empid,e.name,l.locationid,l.locationname

FROM employee e CROSS JOIN location l

ORDER BY empid;

38. Retrieve a list of couriers and their corresponding sender information (if available):

SELECT c.courierid,c.sender,c.senderadrs

FROM courier c

ORDER BY courierid;

39. Retrieve a list of couriers and their corresponding receiver information (if available):

SELECT c.courierid,c.receiver,c.receiveradrs

FROM courier c

ORDER BY courierid;

40. Retrieve a list of couriers along with the courier service details (if available):

SELECT c.courierid,c.sender,s.serviceid,s.servicename

FROM courier c INNER JOIN courierservices s

ON c.courierid=s.serviceid

ORDER BY courierid;

41. Retrieve a list of employees and the number of couriers assigned to each employee:

SELECT e.empid,e.name,p.courierid,COUNT(p.courierid)'Number of couriers'

FROM employee e JOIN payment p

ON e.empid=p.paymentid

GROUP BY empid

42. Retrieve a list of locations and the total payment amount received at each location:

SELECT l.locationid,p.paymentid,l.address,sum(p.amount)

FROM location l INNER JOIN payment p

ON p.locationid=l.locationid

GROUP BY p.paymentid

43. Retrieve all couriers sent by the same sender (based on SenderName).

SELECT c.courierid,c.receiver,c.trackingno,c.deliverydate

FROM courier c

WHERE c.sender='gokul';

44. List all employees who share the same role.

SELECT e.empid,e.name

FROM employee e

WHERE e.role='delivery';

45. Retrieve all payments made for couriers sent from the same location.

SELECT p.paymentid,p.courierid,l.locationid,l.locationname

FROM payment p NATURAL JOIN location l

WHERE l.address='a street';

46. Retrieve all couriers sent from the same location (based on SenderAddress).

SELECT c.courierid,c.sender,c.receiver

FROM courier c

WHERE c.senderadrs='c street';

47. List employees and the number of couriers they have delivered:

SELECT e.empid,e.name,COUNT(p.courierid)'No of couriers'

FROM payment p JOIN employee e

ON e.empid=p.paymentid

GROUP BY empid;

48. Find couriers that were paid an amount greater than the cost of their respective courier services

SELECT p.courierid,s.serviceid,s.servicename,s.cost,p.amount

FROM payment p INNER JOIN courierservices s

ON p.paymentid=s.serviceid

WHERE p.amount>s.cost

ORDER BY courierid;

TASK 5:

49. Find couriers that have a weight greater than the average weight of all couriers:

SELECT courierid,weight

FROM courier

WHERE weight>(SELECT AVG(weight) FROM courier);

50. Find the names of all employees who have a salary greater than the average salary:

SELECT empid,name

FROM employee

WHERE salary>(SELECT AVG(salary) FROM employee);

51. Find the total cost of all courier services where the cost is less than the maximum cost

SELECT serviceid,servicename

FROM courierservices

WHERE cost<(SELECT MAX(cost) FROM courierservices);

52. Find all couriers that have been paid for

SELECT courierid,status

FROM courier

WHERE status=ALL(SELECT status FROM courier

WHERE status='delivered')

53. Find the locations where the maximum payment amount was made

SELECT locationid,paymentid,amount

FROM payment

WHERE amount=ALL(SELECT MAX(amount) FROM payment);

54. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'):

SELECT courierid,weight

FROM courier

WHERE weight=ALL(SELECT MAX(weight) FROM courier)

AND sender='jessie';