DATABASE TECHNOLOGY (DBT) PROJECT

TEAM 115

Contractor Billing & Approval System

Team Members:

Name	PRN
1.Chaitali Barhate	**250840320046**
2.Dipti Akhade	**250840320061**

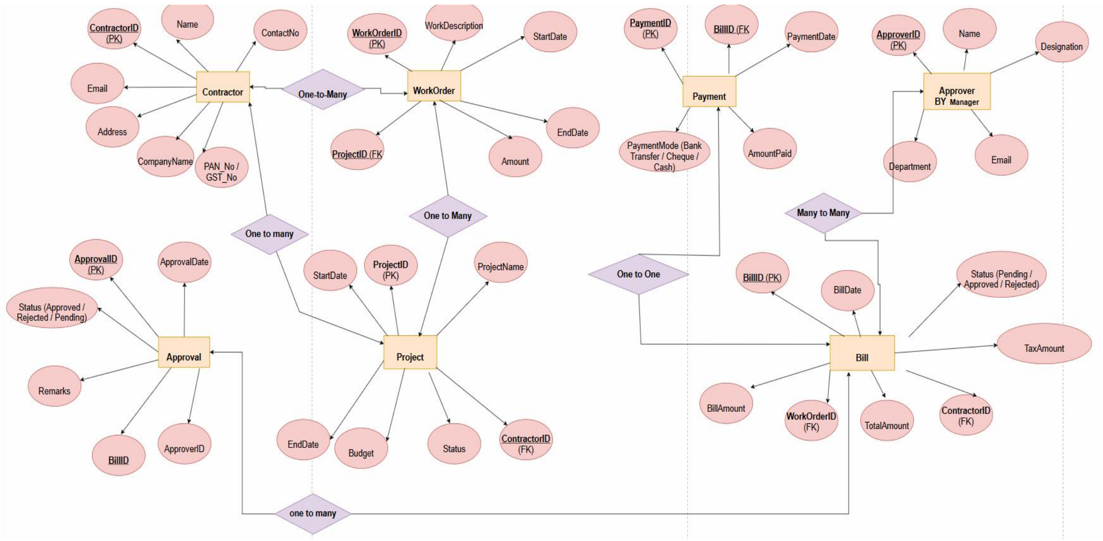
Submitted To: Shweta Bhere.

Under the Guidance of: Vipul Tembulwar.

Academic Year : 2025 – 2026

1) Draw an ER Diagram in *draw.io* showing entities, attributes, and relationships.

- Identify all major entities, their attributes, and primary keys.
- Show relationships (1–M, M–N, 1–1) with clear cardinalities.
- Include associative entities wherever M:N relationships exist.
- Indicate foreign keys and participation constraints clearly

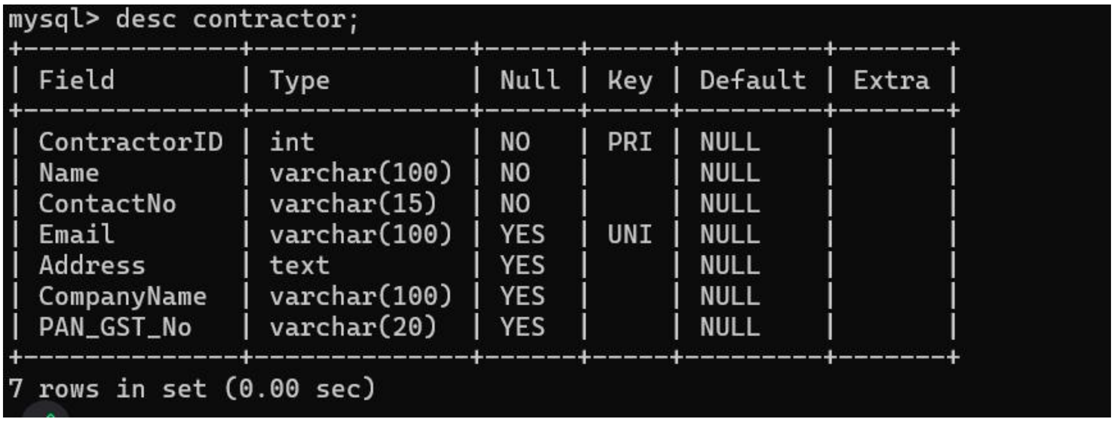


2) Create the database schema (DDL) with all required constraints and relationships.

- Appropriate data types and size definitions.
- Primary Keys and Foreign Keys for relationships.
- Unique, Check, and Not Null constraints.
- Use ENUM or SET data types where suitable (e.g., gender, status).
- Create indexes on key searchable fields.

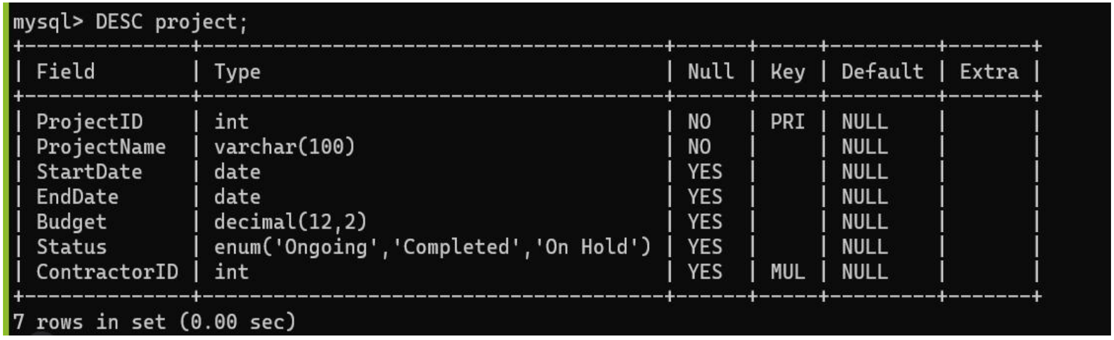
```
1.CREATE TABLE Contractor (
 ContractorID INT PRIMARY KEY,
 Name VARCHAR(100) NOT NULL,
 ContactNo VARCHAR(15) NOT NULL,
 Email VARCHAR(100) UNIQUE,
 Address TEXT,
 CompanyName VARCHAR(100),
 PAN_GST_No VARCHAR(20)
);
```

DESC Contractor;



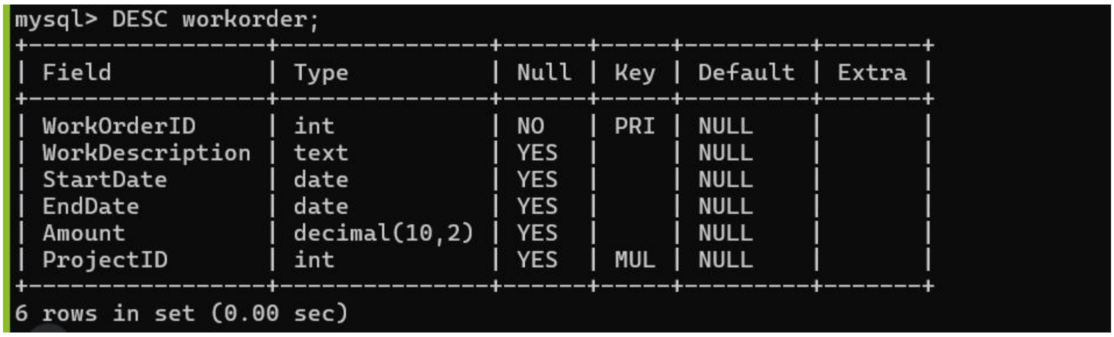
```
mysql> desc contractor;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| ContractorID | int          | NO   | PRI | NULL    |       |
| Name         | varchar(100) | NO   |     | NULL    |       |
| ContactNo    | varchar(15)  | NO   |     | NULL    |       |
| Email        | varchar(100) | YES  | UNI | NULL    |       |
| Address      | text         | YES  |     | NULL    |       |
| CompanyName  | varchar(100) | YES  |     | NULL    |       |
| PAN_GST_No   | varchar(20)  | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
2.CREATE TABLE Project (
  ProjectID INT PRIMARY KEY,
  ProjectName VARCHAR(100) NOT NULL,
  StartDate DATE,
  EndDate DATE,
  Budget DECIMAL(12,2) CHECK (Budget > 0),
  Status ENUM('Ongoing', 'Completed', 'On Hold'),
  ContractorID INT,
  FOREIGN KEY (ContractorID) REFERENCES Contractor(ContractorID)
);
DESC Project;
```



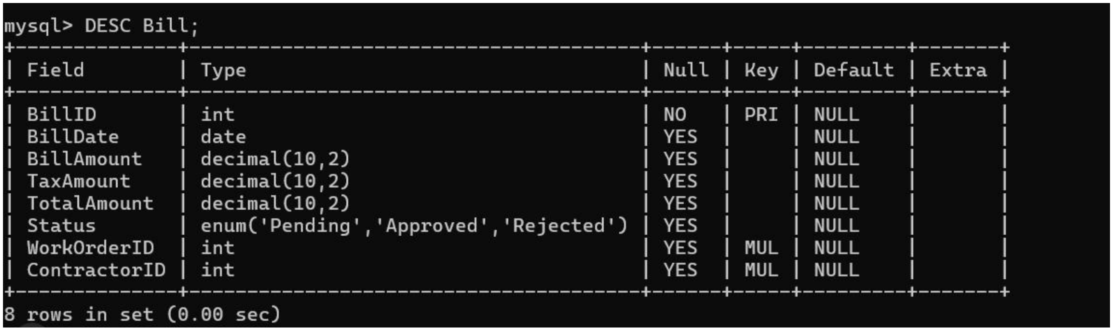
```
mysql> DESC project;
+--------------+-----------------------------------------+------+-----+---------+-------+
| Field        | Type                                    | Null | Key | Default | Extra |
+--------------+-----------------------------------------+------+-----+---------+-------+
| ProjectID    | int                                     | NO   | PRI | NULL    |       |
| ProjectName  | varchar(100)                            | NO   |     | NULL    |       |
| StartDate    | date                                    | YES  |     | NULL    |       |
| EndDate      | date                                    | YES  |     | NULL    |       |
| Budget       | decimal(12,2)                           | YES  |     | NULL    |       |
| Status       | enum('Ongoing','Completed','On Hold')   | YES  |     | NULL    |       |
| ContractorID | int                                     | YES  | MUL | NULL    |       |
+--------------+-----------------------------------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
3.CREATE TABLE WorkOrder (
  WorkOrderID INT PRIMARY KEY,
  WorkDescription TEXT,
  StartDate DATE,
  EndDate DATE,
  Amount DECIMAL(10,2),
  ProjectID INT,
  FOREIGN KEY (ProjectID) REFERENCES Project(ProjectID)
);
DESC WorkOrder;
```



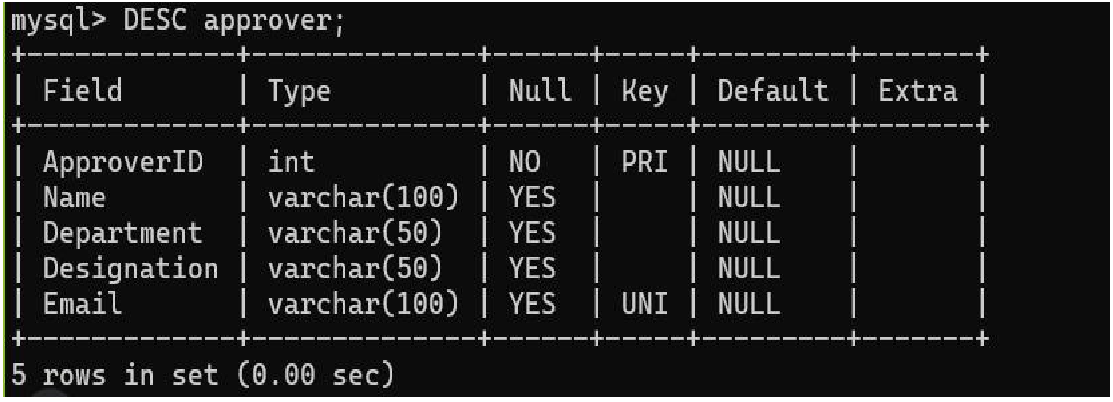
```
mysql> DESC workorder;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| WorkOrderID     | int           | NO   | PRI | NULL    |       |
| WorkDescription | text          | YES  |     | NULL    |       |
| StartDate       | date          | YES  |     | NULL    |       |
| EndDate         | date          | YES  |     | NULL    |       |
| Amount          | decimal(10,2) | YES  |     | NULL    |       |
| ProjectID       | int           | YES  | MUL | NULL    |       |
+-----------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
4.CREATE TABLE Bill (
  BillID INT PRIMARY KEY,
  BillDate DATE,
  BillAmount DECIMAL(10,2),
  TaxAmount DECIMAL(10,2),
  TotalAmount DECIMAL(10,2),
  Status ENUM('Pending', 'Approved', 'Rejected'),
  WorkOrderID INT,
  ContractorID INT,
  FOREIGN KEY (WorkOrderID) REFERENCES WorkOrder(WorkOrderID),
  FOREIGN KEY (ContractorID) REFERENCES Contractor(ContractorID)
);
DESC Bill;
```



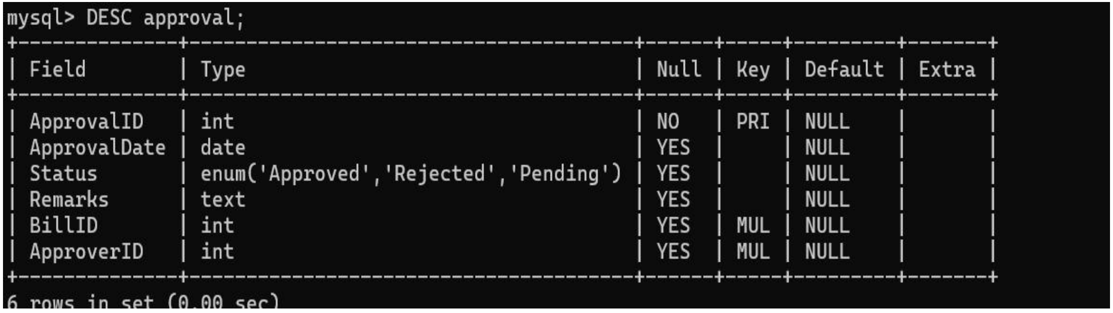
```
mysql> DESC Bill;
+--------------+-------------------------------------------+------+-----+---------+-------+
| Field        | Type                                      | Null | Key | Default | Extra |
+--------------+-------------------------------------------+------+-----+---------+-------+
| BillID       | int                                       | NO   | PRI | NULL    |       |
| BillDate     | date                                      | YES  |     | NULL    |       |
| BillAmount   | decimal(10,2)                             | YES  |     | NULL    |       |
| TaxAmount    | decimal(10,2)                             | YES  |     | NULL    |       |
| TotalAmount  | decimal(10,2)                             | YES  |     | NULL    |       |
| Status       | enum('Pending','Approved','Rejected')     | YES  |     | NULL    |       |
| WorkOrderID  | int                                       | YES  | MUL | NULL    |       |
| ContractorID | int                                       | YES  | MUL | NULL    |       |
+--------------+-------------------------------------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

```
5.CREATE TABLE Approver (
  ApproverID INT PRIMARY KEY,
  Name VARCHAR(100),
  Department VARCHAR(50),
  Designation VARCHAR(50),
  Email VARCHAR(100) UNIQUE
);
DESC Approver;
```



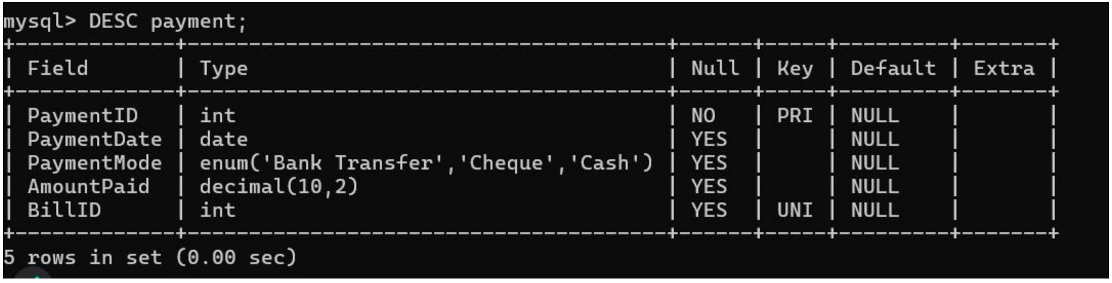
```
mysql> DESC approver;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ApproverID  | int          | NO   | PRI | NULL    |       |
| Name        | varchar(100) | YES  |     | NULL    |       |
| Department  | varchar(50)  | YES  |     | NULL    |       |
| Designation | varchar(50)  | YES  |     | NULL    |       |
| Email       | varchar(100) | YES  | UNI | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

6.CREATE TABLE Approval (
  ApprovalID INT PRIMARY KEY,
  ApprovalDate DATE,
  Status ENUM('Approved', 'Rejected', 'Pending'),
  Remarks TEXT,
  BillID INT,
  ApproverID INT,
  FOREIGN KEY (BillID) REFERENCES Bill(BillID),
  FOREIGN KEY (ApproverID) REFERENCES Approver(ApproverID)
);
DESC Approval;



```
mysql> DESC approval;
+--------------+----------------------------------------+------+-----+---------+-------+
| Field        | Type                                   | Null | Key | Default | Extra |
+--------------+----------------------------------------+------+-----+---------+-------+
| ApprovalID   | int                                    | NO   | PRI | NULL    |       |
| ApprovalDate | date                                   | YES  |     | NULL    |       |
| Status       | enum('Approved','Rejected','Pending') | YES  |     | NULL    |       |
| Remarks      | text                                   | YES  |     | NULL    |       |
| BillID       | int                                    | YES  | MUL | NULL    |       |
| ApproverID   | int                                    | YES  | MUL | NULL    |       |
+--------------+----------------------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

7.CREATE TABLE Payment (
  PaymentID INT PRIMARY KEY,
  PaymentDate DATE,
  PaymentMode ENUM('Bank Transfer', 'Cheque', 'Cash'),
  AmountPaid DECIMAL(10,2),
  BillID INT UNIQUE,
  FOREIGN KEY (BillID) REFERENCES Bill(BillID)
)
DESC Payment;



```
mysql> DESC payment;
+-------------+--------------------------------------+------+-----+---------+-------+
| Field       | Type                                 | Null | Key | Default | Extra |
+-------------+--------------------------------------+------+-----+---------+-------+
| PaymentID   | int                                  | NO   | PRI | NULL    |       |
| PaymentDate | date                                 | YES  |     | NULL    |       |
| PaymentMode | enum('Bank Transfer','Cheque','Cash') | YES  |     | NULL    |       |
| AmountPaid  | decimal(10,2)                        | YES  |     | NULL    |       |
| BillID      | int                                  | YES  | UNI | NULL    |       |
+-------------+--------------------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

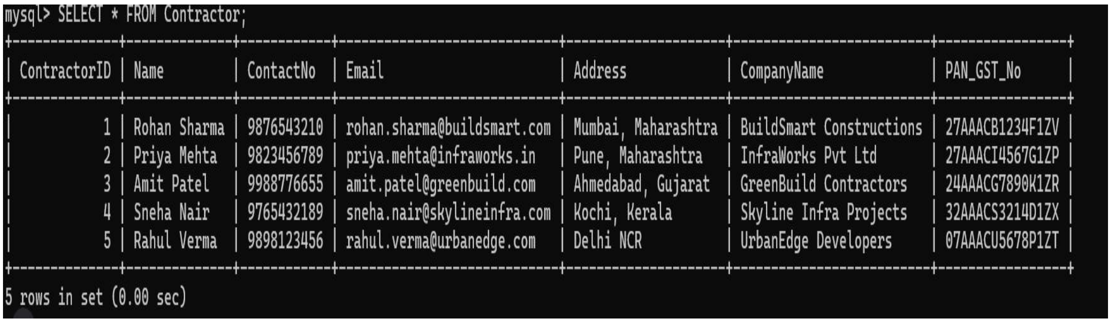
CREATE INDEX idx_bill_status ON Bill(Status);
CREATE INDEX idx_project_name ON Project(ProjectName);

3) Perform DML operations (Insert, Update, Delete) to populate sample data.

- Insert at least 5–10 records in each main table.
- Update some attribute (e.g., change contact info, modify price, update status).
- Delete one or more records safely (with `WHERE` condition).

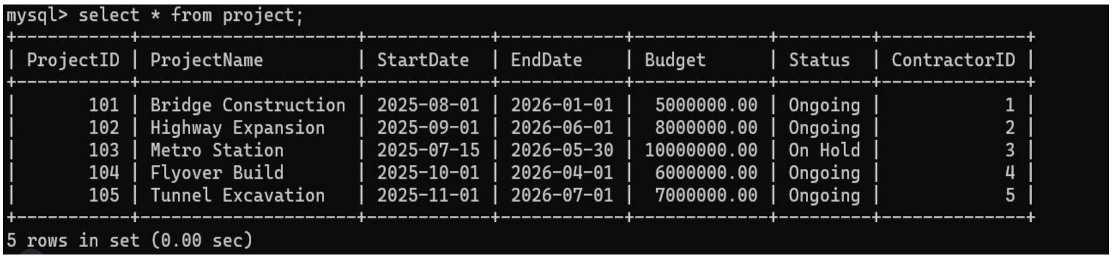
1.
INSERT INTO Contractor (ContractorID, Name, ContactNo, Email, Address, CompanyName, PAN_GST_No)
VALUES
(1, 'Rohan Sharma', '9876543210', 'rohan.sharma@buildsmart.com', 'Mumbai, Maharashtra', 'BuildSmart Constructions', '27AAACB1234F1ZV'),

(2, 'Priya Mehta', '9823456789', 'priya.mehta@infraworks.in', 'Pune, Maharashtra', 'InfraWorks Pvt Ltd', '27AAACI4567G1ZP'),
(3, 'Amit Patel', '9988776655', 'amit.patel@greenbuild.com', 'Ahmedabad, Gujarat', 'GreenBuild Contractors', '24AAACG7890K1ZR'),
(4, 'Sneha Nair', '9765432189', 'sneha.nair@skylineinfra.com', 'Kochi, Kerala', 'Skyline Infra Projects', '32AAACS3214D1ZX'),
(5, 'Rahul Verma', '9898123456', 'rahul.verma@urbanedge.com', 'Delhi NCR', 'UrbanEdge Developers', '07AAACU5678P1ZT');



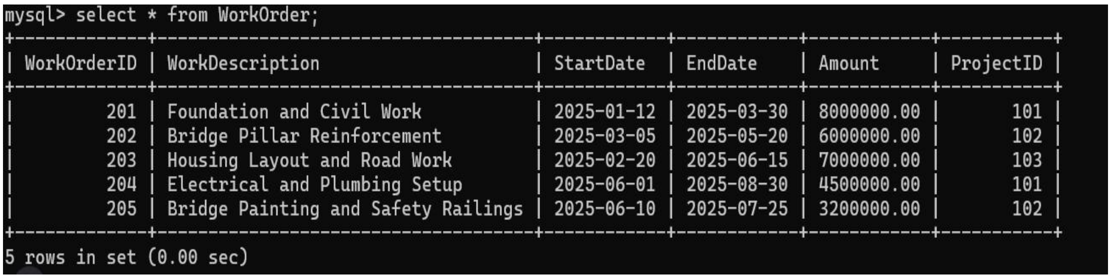
```
mysql> SELECT * FROM Contractor;
+--------------+--------------+------------+-----------------------------+---------------------+--------------------------+-----------------+
| ContractorID | Name         | ContactNo  | Email                       | Address             | CompanyName              | PAN_GST_No      |
+--------------+--------------+------------+-----------------------------+---------------------+--------------------------+-----------------+
|            1 | Rohan Sharma | 9876543210 | rohan.sharma@buildsmart.com | Mumbai, Maharashtra | BuildSmart Constructions | 27AAACB1234F1ZV |
|            2 | Priya Mehta  | 9823456789 | priya.mehta@infraworks.in   | Pune, Maharashtra   | InfraWorks Pvt Ltd       | 27AAACI4567G1ZP |
|            3 | Amit Patel   | 9988776655 | amit.patel@greenbuild.com   | Ahmedabad, Gujarat  | GreenBuild Contractors   | 24AAACG7890K1ZR |
|            4 | Sneha Nair   | 9765432189 | sneha.nair@skylineinfra.com | Kochi, Kerala       | Skyline Infra Projects   | 32AAACS3214D1ZX |
|            5 | Rahul Verma  | 9898123456 | rahul.verma@urbanedge.com   | Delhi NCR           | UrbanEdge Developers     | 07AAACU5678P1ZT |
+--------------+--------------+------------+-----------------------------+---------------------+--------------------------+-----------------+
5 rows in set (0.00 sec)
```

2.
INSERT INTO Project VALUES
(101, 'Bridge Construction', '2025-08-01', '2026-01-01', 5000000, 'Ongoing', 1),
(102, 'Highway Expansion', '2025-09-01', '2026-06-01', 8000000, 'Ongoing', 2),
(103, 'Metro Station', '2025-07-15', '2026-05-30', 10000000, 'On Hold', 3),
(104, 'Flyover Build', '2025-10-01', '2026-04-01', 6000000, 'Ongoing', 4),
(105, 'Tunnel Excavation', '2025-11-01', '2026-07-01', 7000000, 'Ongoing', 5);



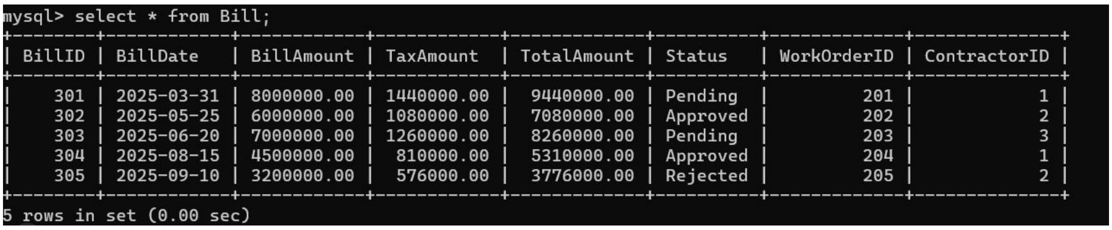
```
mysql> select * from project;
+-----------+---------------------+------------+------------+-------------+---------+--------------+
| ProjectID | ProjectName         | StartDate  | EndDate    | Budget      | Status  | ContractorID |
+-----------+---------------------+------------+------------+-------------+---------+--------------+
|       101 | Bridge Construction | 2025-08-01 | 2026-01-01 |  5000000.00 | Ongoing |            1 |
|       102 | Highway Expansion   | 2025-09-01 | 2026-06-01 |  8000000.00 | Ongoing |            2 |
|       103 | Metro Station       | 2025-07-15 | 2026-05-30 | 10000000.00 | On Hold |            3 |
|       104 | Flyover Build       | 2025-10-01 | 2026-04-01 |  6000000.00 | Ongoing |            4 |
|       105 | Tunnel Excavation   | 2025-11-01 | 2026-07-01 |  7000000.00 | Ongoing |            5 |
+-----------+---------------------+------------+------------+-------------+---------+--------------+
5 rows in set (0.00 sec)
```

3.
INSERT INTO WorkOrder (WorkOrderID, WorkDescription, StartDate, EndDate, Amount, ProjectID)
VALUES
(201, 'Foundation and Civil Work', '2025-01-12', '2025-03-30', 8000000.00, 101),
(202, 'Bridge Pillar Reinforcement', '2025-03-05', '2025-05-20', 6000000.00, 102),
(203, 'Housing Layout and Road Work', '2025-02-20', '2025-06-15', 7000000.00, 103),
(204, 'Electrical and Plumbing Setup', '2025-06-01', '2025-08-30', 4500000.00, 101),
(205, 'Bridge Painting and Safety Railings', '2025-06-10', '2025-07-25', 3200000.00, 102)



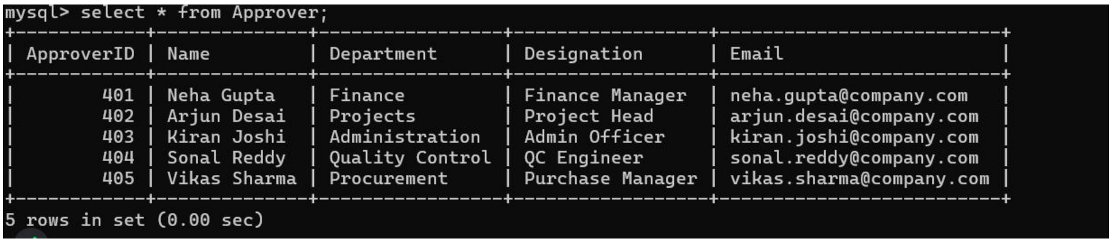
```
mysql> select * from WorkOrder;
+-------------+-------------------------------------+------------+------------+------------+-----------+
| WorkOrderID | WorkDescription                     | StartDate  | EndDate    | Amount     | ProjectID |
+-------------+-------------------------------------+------------+------------+------------+-----------+
|         201 | Foundation and Civil Work           | 2025-01-12 | 2025-03-30 | 8000000.00 |       101 |
|         202 | Bridge Pillar Reinforcement         | 2025-03-05 | 2025-05-20 | 6000000.00 |       102 |
|         203 | Housing Layout and Road Work        | 2025-02-20 | 2025-06-15 | 7000000.00 |       103 |
|         204 | Electrical and Plumbing Setup       | 2025-06-01 | 2025-08-30 | 4500000.00 |       101 |
|         205 | Bridge Painting and Safety Railings | 2025-06-10 | 2025-07-25 | 3200000.00 |       102 |
+-------------+-------------------------------------+------------+------------+------------+-----------+
5 rows in set (0.00 sec)
```

4.INSERT INTO Bill (BillID, BillDate, BillAmount, TaxAmount, TotalAmount, Status, WorkOrderID, ContractorID)
VALUES
(301, '2025-03-31', 8000000.00, 1440000.00, 9440000.00, 'Pending', 201, 1),
(302, '2025-05-25', 6000000.00, 1080000.00, 7080000.00, 'Approved', 202, 2),
(303, '2025-06-20', 7000000.00, 1260000.00, 8260000.00, 'Pending', 203, 3),
(304, '2025-08-15', 4500000.00, 810000.00, 5310000.00, 'Approved', 204, 1),
(305, '2025-09-10', 3200000.00, 576000.00, 3776000.00, 'Rejected', 205, 2);



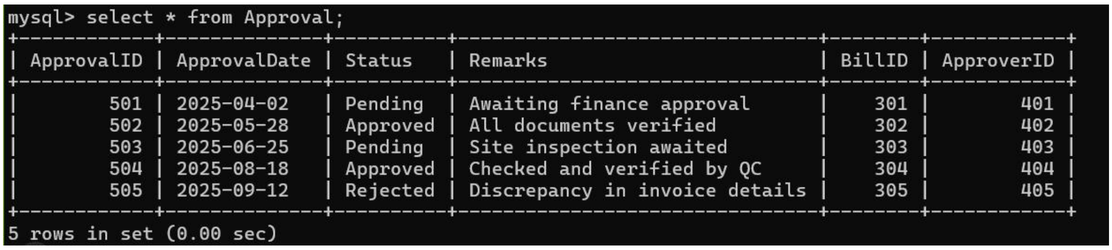
```
mysql> select * from Bill;
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
| BillID | BillDate   | BillAmount | TaxAmount  | TotalAmount | Status   | WorkOrderID | ContractorID |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
|    301 | 2025-03-31 | 8000000.00 | 1440000.00 |  9440000.00 | Pending  |         201 |            1 |
|    302 | 2025-05-25 | 6000000.00 | 1080000.00 |  7080000.00 | Approved |         202 |            2 |
|    303 | 2025-06-20 | 7000000.00 | 1260000.00 |  8260000.00 | Pending  |         203 |            3 |
|    304 | 2025-08-15 | 4500000.00 |  810000.00 |  5310000.00 | Approved |         204 |            1 |
|    305 | 2025-09-10 | 3200000.00 |  576000.00 |  3776000.00 | Rejected |         205 |            2 |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
5 rows in set (0.00 sec)
```

5.INSERT INTO Approver (ApproverID, Name, Department, Designation, Email)
VALUES
(401, 'Neha Gupta', 'Finance', 'Finance Manager', 'neha.gupta@company.com'),
(402, 'Arjun Desai', 'Projects', 'Project Head', 'arjun.desai@company.com'),
(403, 'Kiran Joshi', 'Administration', 'Admin Officer', 'kiran.joshi@company.com'),
(404, 'Sonal Reddy', 'Quality Control', 'QC Engineer', 'sonal.reddy@company.com'),
(405, 'Vikas Sharma', 'Procurement', 'Purchase Manager', 'vikas.sharma@company.com');



```
mysql> select * from Approver;
+------------+--------------+-----------------+------------------+--------------------------+
| ApproverID | Name         | Department      | Designation      | Email                    |
+------------+--------------+-----------------+------------------+--------------------------+
|        401 | Neha Gupta   | Finance         | Finance Manager  | neha.gupta@company.com   |
|        402 | Arjun Desai  | Projects        | Project Head     | arjun.desai@company.com  |
|        403 | Kiran Joshi  | Administration  | Admin Officer    | kiran.joshi@company.com  |
|        404 | Sonal Reddy  | Quality Control | QC Engineer      | sonal.reddy@company.com  |
|        405 | Vikas Sharma | Procurement     | Purchase Manager | vikas.sharma@company.com |
+------------+--------------+-----------------+------------------+--------------------------+
5 rows in set (0.00 sec)
```

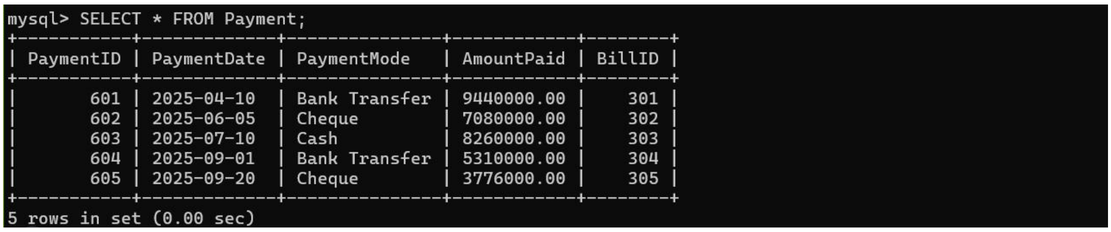
6.
INSERT INTO Approval (ApprovalID, ApprovalDate, Status, Remarks, BillID, ApproverID)
VALUES
(501, '2025-04-02', 'Pending', 'Awaiting finance approval', 301, 401),
(502, '2025-05-28', 'Approved', 'All documents verified', 302, 402),
(503, '2025-06-25', 'Pending', 'Site inspection awaited', 303, 403),
(504, '2025-08-18', 'Approved', 'Checked and verified by QC', 304, 404),
(505, '2025-09-12', 'Rejected', 'Discrepancy in invoice details', 305, 405);



```
mysql> select * from Approval;
+------------+--------------+----------+--------------------------------+--------+------------+
| ApprovalID | ApprovalDate | Status   | Remarks                        | BillID | ApproverID |
+------------+--------------+----------+--------------------------------+--------+------------+
|        501 | 2025-04-02   | Pending  | Awaiting finance approval      |    301 |        401 |
|        502 | 2025-05-28   | Approved | All documents verified         |    302 |        402 |
|        503 | 2025-06-25   | Pending  | Site inspection awaited        |    303 |        403 |
|        504 | 2025-08-18   | Approved | Checked and verified by QC     |    304 |        404 |
|        505 | 2025-09-12   | Rejected | Discrepancy in invoice details |    305 |        405 |
+------------+--------------+----------+--------------------------------+--------+------------+
5 rows in set (0.00 sec)
```

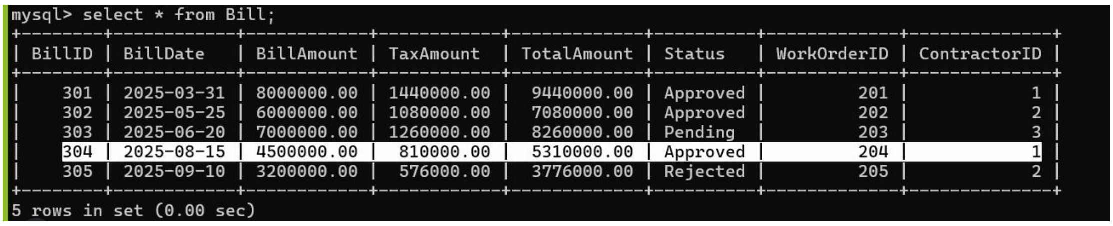
7.
INSERT INTO Payment (PaymentID, PaymentDate, PaymentMode, AmountPaid, BillID)
VALUES
(601, '2025-04-10', 'Bank Transfer', 9440000.00, 301),
(602, '2025-06-05', 'Cheque', 7080000.00, 302),

```
(603, '2025-07-10', 'Cash', 8260000.00, 303),
(604, '2025-09-01', 'Bank Transfer', 5310000.00, 304),
(605, '2025-09-20', 'Cheque', 3776000.00, 305);
```

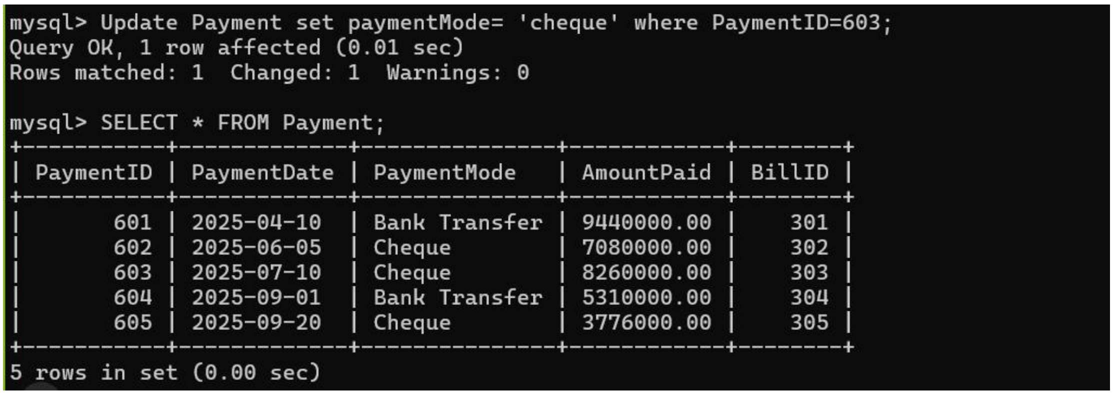


```
mysql> SELECT * FROM Payment;
+-----------+-------------+---------------+------------+--------+
| PaymentID | PaymentDate | PaymentMode   | AmountPaid | BillID |
+-----------+-------------+---------------+------------+--------+
|       601 | 2025-04-10  | Bank Transfer | 9440000.00 |    301 |
|       602 | 2025-06-05  | Cheque        | 7080000.00 |    302 |
|       603 | 2025-07-10  | Cash          | 8260000.00 |    303 |
|       604 | 2025-09-01  | Bank Transfer | 5310000.00 |    304 |
|       605 | 2025-09-20  | Cheque        | 3776000.00 |    305 |
+-----------+-------------+---------------+------------+--------+
5 rows in set (0.00 sec)
```

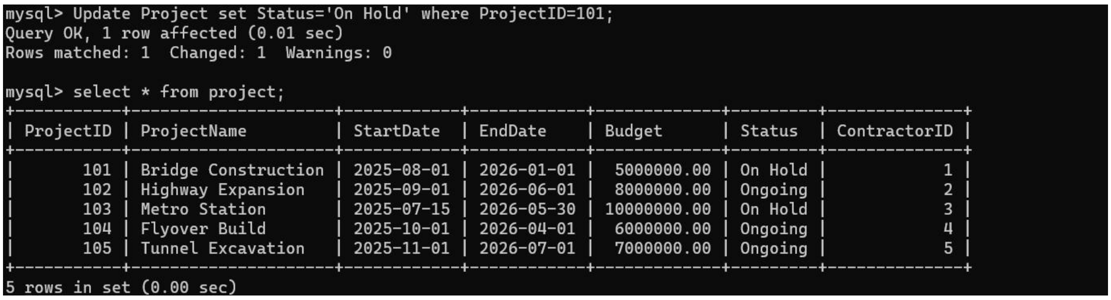
```
UPDATE Bill SET Status = 'Approved' WHERE BillID = 301;
```



```
mysql> select * from Bill;
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
| BillID | BillDate   | BillAmount | TaxAmount  | TotalAmount | Status   | WorkOrderID | ContractorID |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
|    301 | 2025-03-31 | 8000000.00 | 1440000.00 |  9440000.00 | Approved |         201 |            1 |
|    302 | 2025-05-25 | 6000000.00 | 1080000.00 |  7080000.00 | Approved |         202 |            2 |
|    303 | 2025-06-20 | 7000000.00 | 1260000.00 |  8260000.00 | Pending  |         203 |            3 |
|    304 | 2025-08-15 | 4500000.00 |  810000.00 |  5310000.00 | Approved |         204 |            1 |
|    305 | 2025-09-10 | 3200000.00 |  576000.00 |  3776000.00 | Rejected |         205 |            2 |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
5 rows in set (0.00 sec)
```



```
mysql> Update Payment set paymentMode= 'cheque' where PaymentID=603;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Payment;
+-----------+-------------+---------------+------------+--------+
| PaymentID | PaymentDate | PaymentMode   | AmountPaid | BillID |
+-----------+-------------+---------------+------------+--------+
|       601 | 2025-04-10  | Bank Transfer | 9440000.00 |    301 |
|       602 | 2025-06-05  | Cheque        | 7080000.00 |    302 |
|       603 | 2025-07-10  | Cheque        | 8260000.00 |    303 |
|       604 | 2025-09-01  | Bank Transfer | 5310000.00 |    304 |
|       605 | 2025-09-20  | Cheque        | 3776000.00 |    305 |
+-----------+-------------+---------------+------------+--------+
5 rows in set (0.00 sec)
```

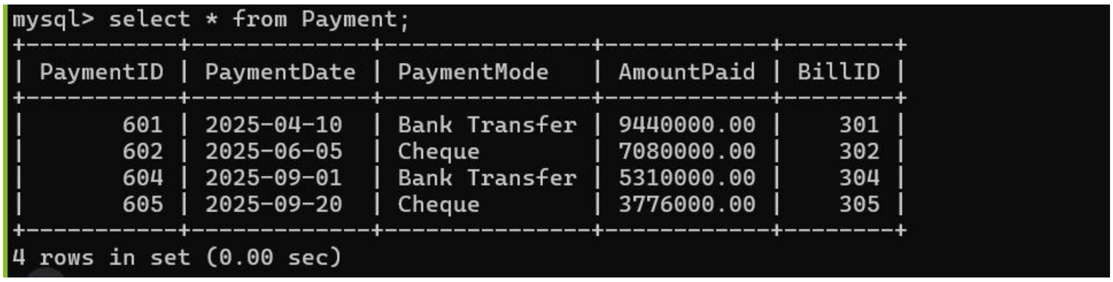


```
mysql> Update Project set Status='On Hold' where ProjectID=101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from project;
+-----------+--------------------+------------+------------+-------------+---------+--------------+
| ProjectID | ProjectName        | StartDate  | EndDate    | Budget      | Status  | ContractorID |
+-----------+--------------------+------------+------------+-------------+---------+--------------+
|       101 | Bridge Construction | 2025-08-01 | 2026-01-01 |  5000000.00 | On Hold |            1 |
|       102 | Highway Expansion  | 2025-09-01 | 2026-06-01 |  8000000.00 | Ongoing |            2 |
|       103 | Metro Station      | 2025-07-15 | 2026-05-30 | 10000000.00 | On Hold |            3 |
|       104 | Flyover Build      | 2025-10-01 | 2026-04-01 |  6000000.00 | Ongoing |            4 |
|       105 | Tunnel Excavation  | 2025-11-01 | 2026-07-01 |  7000000.00 | Ongoing |            5 |
+-----------+--------------------+------------+------------+-------------+---------+--------------+
5 rows in set (0.00 sec)
```

Delete Using Where clause:

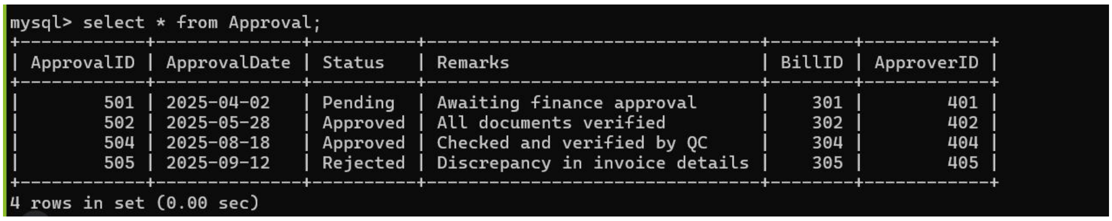
```
DELETE FROM Payment
WHERE BillID IN (
   SELECT BillID FROM Bill
   WHERE ContractorID IN (
      SELECT ContractorID FROM Contractor
      WHERE Name='Amit Patel'
   )
```

);



```
mysql> select * from Payment;
+-----------+-------------+---------------+------------+--------+
| PaymentID | PaymentDate | PaymentMode   | AmountPaid | BillID |
+-----------+-------------+---------------+------------+--------+
|       601 | 2025-04-10  | Bank Transfer | 9440000.00 |    301 |
|       602 | 2025-06-05  | Cheque        | 7080000.00 |    302 |
|       604 | 2025-09-01  | Bank Transfer | 5310000.00 |    304 |
|       605 | 2025-09-20  | Cheque        | 3776000.00 |    305 |
+-----------+-------------+---------------+------------+--------+
4 rows in set (0.00 sec)
```

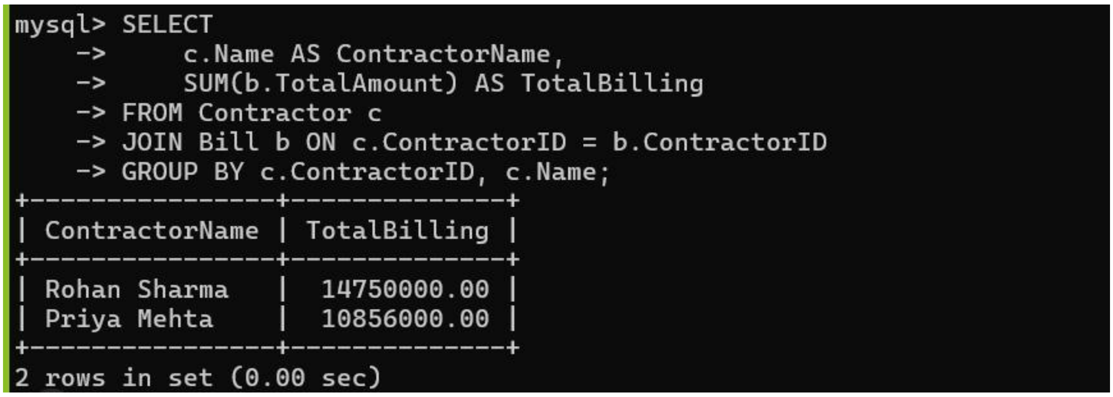
Delete using JOIN:
DELETE a
FROM Approval a
JOIN Bill b ON a.BillID = b.BillID
JOIN Contractor c ON b.ContractorID = c.ContractorID
WHERE c.Name = 'Amit Patel';



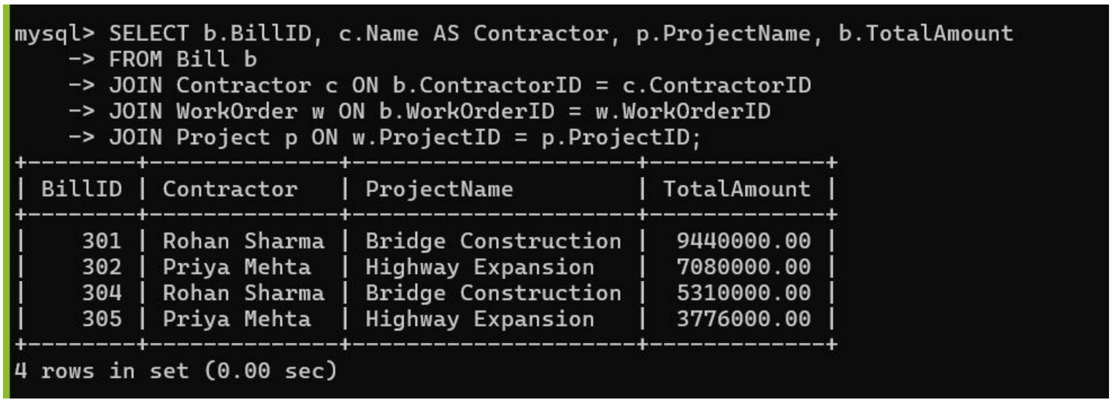
```
mysql> select * from Approval;
+------------+--------------+----------+------------------------------+--------+------------+
| ApprovalID | ApprovalDate | Status   | Remarks                      | BillID | ApproverID |
+------------+--------------+----------+------------------------------+--------+------------+
|        501 | 2025-04-02   | Pending  | Awaiting finance approval    |    301 |        401 |
|        502 | 2025-05-28   | Approved | All documents verified       |    302 |        402 |
|        504 | 2025-08-18   | Approved | Checked and verified by QC   |    304 |        404 |
|        505 | 2025-09-12   | Rejected | Discrepancy in invoice details |    305 |        405 |
+------------+--------------+----------+------------------------------+--------+------------+
4 rows in set (0.00 sec)
```

4) Write SQL Queries using Joins, Aggregate functions, Grouping, and Subqueries to retrieve meaningful
information.

JOIN :

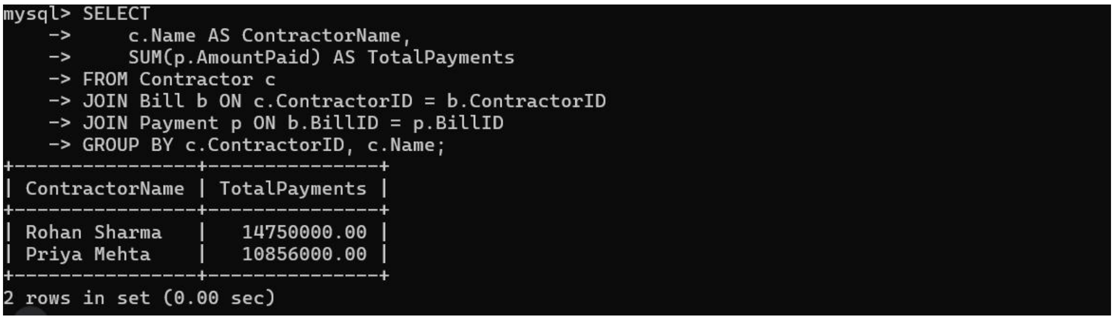


```
mysql> SELECT
    ->     c.Name AS ContractorName,
    ->     SUM(b.TotalAmount) AS TotalBilling
    -> FROM Contractor c
    -> JOIN Bill b ON c.ContractorID = b.ContractorID
    -> GROUP BY c.ContractorID, c.Name;
+----------------+--------------+
| ContractorName | TotalBilling |
+----------------+--------------+
| Rohan Sharma   |  14750000.00 |
| Priya Mehta    |  10856000.00 |
+----------------+--------------+
2 rows in set (0.00 sec)
```

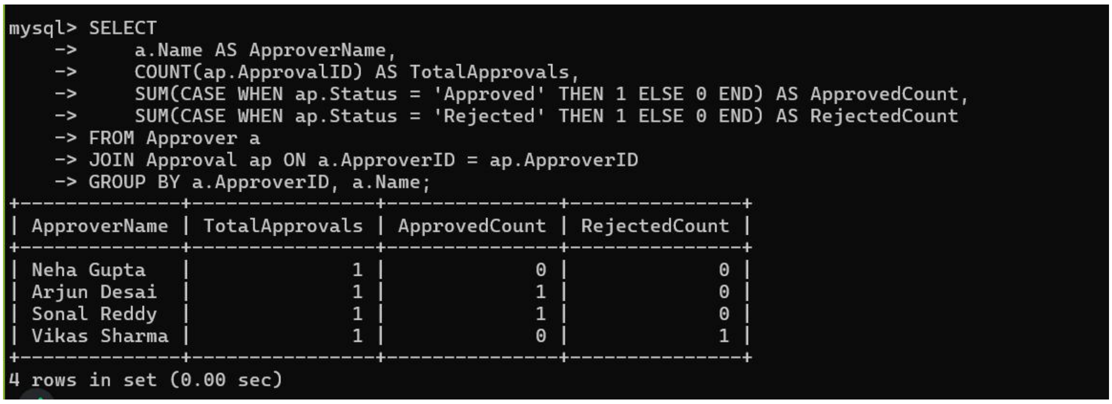


```
mysql> SELECT b.BillID, c.Name AS Contractor, p.ProjectName, b.TotalAmount
    -> FROM Bill b
    -> JOIN Contractor c ON b.ContractorID = c.ContractorID
    -> JOIN WorkOrder w ON b.WorkOrderID = w.WorkOrderID
    -> JOIN Project p ON w.ProjectID = p.ProjectID;
+--------+--------------+---------------------+-------------+
| BillID | Contractor   | ProjectName         | TotalAmount |
+--------+--------------+---------------------+-------------+
|    301 | Rohan Sharma | Bridge Construction |  9440000.00 |
|    302 | Priya Mehta  | Highway Expansion   |  7080000.00 |
|    304 | Rohan Sharma | Bridge Construction |  5310000.00 |
|    305 | Priya Mehta  | Highway Expansion   |  3776000.00 |
+--------+--------------+---------------------+-------------+
4 rows in set (0.00 sec)
```

JOIN + GROUP BY+, Aggregate functions:



```
mysql> SELECT
    ->     c.Name AS ContractorName,
    ->     SUM(p.AmountPaid) AS TotalPayments
    -> FROM Contractor c
    -> JOIN Bill b ON c.ContractorID = b.ContractorID
    -> JOIN Payment p ON b.BillID = p.BillID
    -> GROUP BY c.ContractorID, c.Name;
+----------------+---------------+
| ContractorName | TotalPayments |
+----------------+---------------+
| Rohan Sharma   |   14750000.00 |
| Priya Mehta    |   10856000.00 |
+----------------+---------------+
2 rows in set (0.00 sec)
```

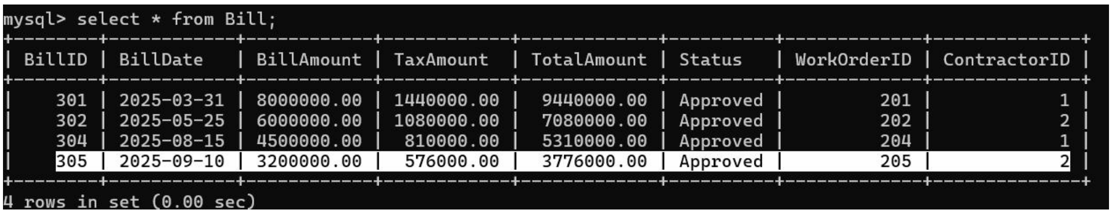


```
mysql> SELECT
    ->     a.Name AS ApproverName,
    ->     COUNT(ap.ApprovalID) AS TotalApprovals,
    ->     SUM(CASE WHEN ap.Status = 'Approved' THEN 1 ELSE 0 END) AS ApprovedCount,
    ->     SUM(CASE WHEN ap.Status = 'Rejected' THEN 1 ELSE 0 END) AS RejectedCount
    -> FROM Approver a
    -> JOIN Approval ap ON a.ApproverID = ap.ApproverID
    -> GROUP BY a.ApproverID, a.Name;
+--------------+----------------+---------------+---------------+
| ApproverName | TotalApprovals | ApprovedCount | RejectedCount |
+--------------+----------------+---------------+---------------+
| Neha Gupta   |              1 |             0 |             0 |
| Arjun Desai  |              1 |             1 |             0 |
| Sonal Reddy  |              1 |             1 |             0 |
| Vikas Sharma |              1 |             0 |             1 |
+--------------+----------------+---------------+---------------+
4 rows in set (0.00 sec)
```

5) Implement a Trigger, a Function, and a Stored Procedure relevant to your system's logic.

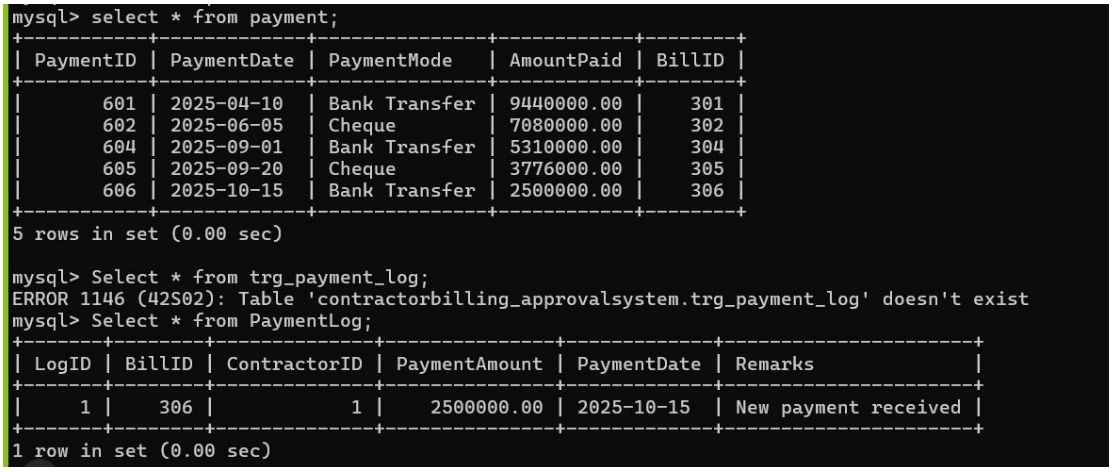
1.TRIGGER

```
-- Trigger: Auto-update bill status when approval is inserted
DELIMITER $$

CREATE TRIGGER trg_auto_approve
AFTER INSERT ON Approval
FOR EACH ROW
BEGIN
  IF NEW.Status = 'Approved' THEN
    UPDATE Bill
    SET Status = 'Approved'
    WHERE BillID = NEW.BillID;
  END IF;
END$$

DELIMITER ;


INSERT INTO Approval (ApprovalID, ApprovalDate, Status, Remarks, BillID, ApproverID)
VALUES
(509, '2025-05-21', 'Approved', 'checking for Approved', 305, 403);
```



```
mysql> select * from Bill;
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
| BillID | BillDate   | BillAmount | TaxAmount  | TotalAmount | Status   | WorkOrderID | ContractorID |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
|    301 | 2025-03-31 | 8000000.00 | 1440000.00 |  9440000.00 | Approved |         201 |            1 |
|    302 | 2025-05-25 | 6000000.00 | 1080000.00 |  7080000.00 | Approved |         202 |            2 |
|    304 | 2025-08-15 | 4500000.00 |  810000.00 |  5310000.00 | Approved |         204 |            1 |
|    305 | 2025-09-10 | 3200000.00 |  576000.00 |  3776000.00 | Approved |         205 |            2 |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
4 rows in set (0.00 sec)
```

```
CREATE PROCEDURE sp_ContractorPaymentReport(IN p_ContractorID INT)
BEGIN
  -- List all bills and payments for the contractor
  SELECT
    b.BillID,
    b.BillDate,
    b.TotalAmount AS BillAmount,
    IFNULL(p.AmountPaid, 0) AS PaymentMade,
    b.TotalAmount - IFNULL(p.AmountPaid, 0) AS Balance,
    b.Status
  FROM Bill b
  LEFT JOIN Payment p ON b.BillID = p.BillID
  WHERE b.ContractorID = p_ContractorID;

  -- Summary totals
  SELECT
    SUM(b.TotalAmount) AS TotalBilled,
    SUM(IFNULL(p.AmountPaid, 0)) AS TotalPaid,
    SUM(b.TotalAmount - IFNULL(p.AmountPaid, 0)) AS TotalBalance
  FROM Bill b
  LEFT JOIN Payment p ON b.BillID = p.BillID
  WHERE b.ContractorID = p_ContractorID;
END$$

DELIMITER ;

CALL sp_ContractorPaymentReport(1);



INSERT INTO Payment (PaymentID, PaymentDate, PaymentMode, AmountPaid, BillID)
VALUES (606, '2025-10-15', 'Bank Transfer', 2500000.00, 306);
```



```
mysql> select * from payment;
+-----------+-------------+---------------+------------+--------+
| PaymentID | PaymentDate | PaymentMode   | AmountPaid | BillID |
+-----------+-------------+---------------+------------+--------+
|       601 | 2025-04-10  | Bank Transfer | 9440000.00 |    301 |
|       602 | 2025-06-05  | Cheque        | 7080000.00 |    302 |
|       604 | 2025-09-01  | Bank Transfer | 5310000.00 |    304 |
|       605 | 2025-09-20  | Cheque        | 3776000.00 |    305 |
|       606 | 2025-10-15  | Bank Transfer | 2500000.00 |    306 |
+-----------+-------------+---------------+------------+--------+
5 rows in set (0.00 sec)

mysql> Select * from trg_payment_log;
ERROR 1146 (42S02): Table 'contractorbilling_approvalsystem.trg_payment_log' doesn't exist
mysql> Select * from PaymentLog;
+-------+--------+--------------+---------------+-------------+----------------------+
| LogID | BillID | ContractorID | PaymentAmount | PaymentDate | Remarks              |
+-------+--------+--------------+---------------+-------------+----------------------+
|     1 |    306 |            1 |    2500000.00 | 2025-10-15  | New payment received |
+-------+--------+--------------+---------------+-------------+----------------------+
1 row in set (0.00 sec)
```

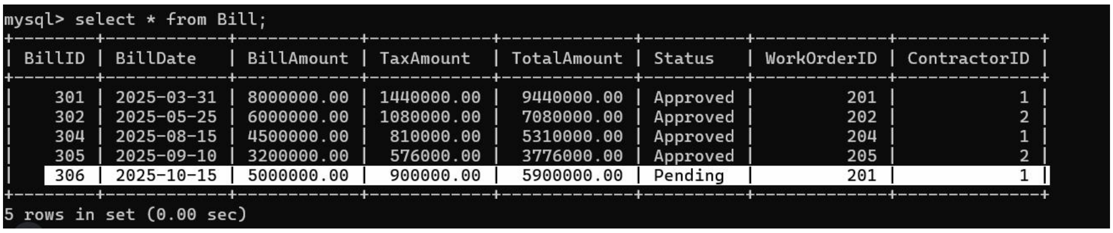
2.FUNCTION

```
DELIMITER $$

CREATE FUNCTION fn_CalculateTotal(billAmt DECIMAL(10,2), taxAmt DECIMAL(10,2))
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    RETURN billAmt + taxAmt;
END$$

DELIMITER ;

DELIMITER $$

CREATE FUNCTION fn_TotalPayments(contractorID INT)
RETURNS DECIMAL(15,2)
DETERMINISTIC
BEGIN
    DECLARE totalPaid DECIMAL(15,2);

    SELECT IFNULL(SUM(p.AmountPaid), 0)
    INTO totalPaid
    FROM Bill b
    LEFT JOIN Payment p ON b.BillID = p.BillID
    WHERE b.ContractorID = contractorID;

    RETURN totalPaid;
END$$

DELIMITER ;
```

3.Stored Procedure

```
DELIMITER $$
```

```
CREATE PROCEDURE sp_SubmitBill(
    IN p_BillID INT,
    IN p_BillDate DATE,
    IN p_BillAmount DECIMAL(10,2),
    IN p_TaxAmount DECIMAL(10,2),
    IN p_Status VARCHAR(10),
    IN p_WorkOrderID INT,
    IN p_ContractorID INT,
    IN p_ApproverID INT   -- Optional: initial approver for bill
)
BEGIN
    DECLARE total DECIMAL(10,2);
    SET total = fn_CalculateTotal(p_BillAmount, p_TaxAmount);

    -- Insert bill
    INSERT INTO Bill (BillID, BillDate, BillAmount, TaxAmount, TotalAmount, Status, WorkOrderID,
ContractorID)
    VALUES (p_BillID, p_BillDate, p_BillAmount, p_TaxAmount, total, p_Status, p_WorkOrderID,
p_ContractorID);

    -- Optionally, create initial approval record as Pending
    INSERT INTO Approval (ApprovalID, ApprovalDate, Status, Remarks, BillID, ApproverID)
    VALUES (p_BillID + 200, CURDATE(), 'Pending', 'Awaiting approval', p_BillID, p_ApproverID);

END$$

DELIMITER ;

CALL sp_SubmitBill(
    306,
    '2025-10-15',
    5000000,
    900000,
    'Pending',
    201,
    1,
    401
);
```



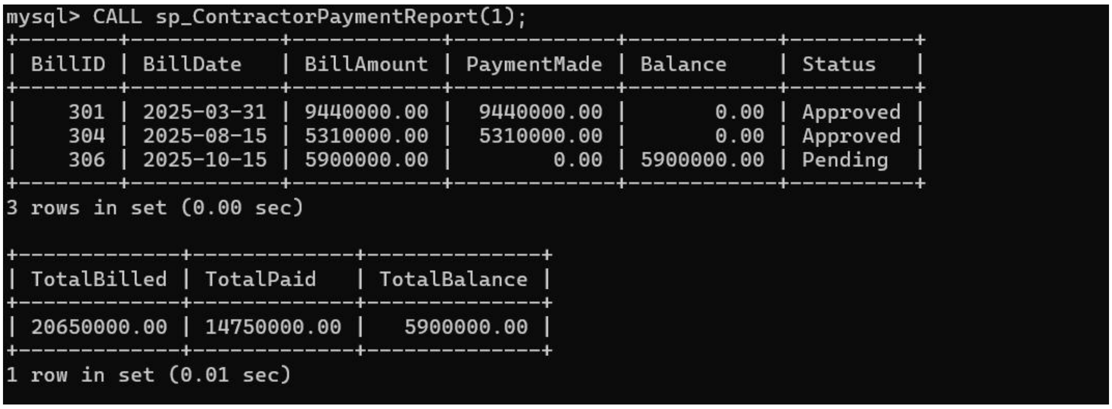
```
mysql> select * from Bill;
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
| BillID | BillDate   | BillAmount | TaxAmount  | TotalAmount | Status   | WorkOrderID | ContractorID |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
|    301 | 2025-03-31 | 8000000.00 | 1440000.00 |  9440000.00 | Approved |         201 |            1 |
|    302 | 2025-05-25 | 6000000.00 | 1080000.00 |  7080000.00 | Approved |         202 |            2 |
|    304 | 2025-08-15 | 4500000.00 |  810000.00 |  5310000.00 | Approved |         204 |            1 |
|    305 | 2025-09-10 | 3200000.00 |  576000.00 |  3776000.00 | Approved |         205 |            2 |
|    306 | 2025-10-15 | 5000000.00 |  900000.00 |  5900000.00 | Pending  |         201 |            1 |
+--------+------------+------------+------------+-------------+----------+-------------+--------------+
5 rows in set (0.00 sec)
```

2.

```
DELIMITER $$

CREATE PROCEDURE sp_ContractorPaymentReport(IN p_ContractorID INT)
BEGIN
    -- List all bills and payments for the contractor
    SELECT
```

```
        b.BillID,
        b.BillDate,
        b.TotalAmount AS BillAmount,
        IFNULL(p.AmountPaid, 0) AS PaymentMade,
        b.TotalAmount - IFNULL(p.AmountPaid, 0) AS Balance,
        b.Status
    FROM Bill b
    LEFT JOIN Payment p ON b.BillID = p.BillID
    WHERE b.ContractorID = p_ContractorID;

    -- Summary totals
    SELECT
        SUM(b.TotalAmount) AS TotalBilled,
        SUM(IFNULL(p.AmountPaid, 0)) AS TotalPaid,
        SUM(b.TotalAmount - IFNULL(p.AmountPaid, 0)) AS TotalBalance
    FROM Bill b
    LEFT JOIN Payment p ON b.BillID = p.BillID
    WHERE b.ContractorID = p_ContractorID;
END$$

DELIMITER ;
```

CALL sp_ContractorPaymentReport(1) ;



```
mysql> CALL sp_ContractorPaymentReport(1);
+--------+------------+------------+-------------+------------+----------+
| BillID | BillDate   | BillAmount | PaymentMade | Balance    | Status   |
+--------+------------+------------+-------------+------------+----------+
|    301 | 2025-03-31 | 9440000.00 |  9440000.00 |       0.00 | Approved |
|    304 | 2025-08-15 | 5310000.00 |  5310000.00 |       0.00 | Approved |
|    306 | 2025-10-15 | 5900000.00 |        0.00 | 5900000.00 | Pending  |
+--------+------------+------------+-------------+------------+----------+
3 rows in set (0.00 sec)

+-------------+-------------+--------------+
| TotalBilled | TotalPaid   | TotalBalance |
+-------------+-------------+--------------+
| 20650000.00 | 14750000.00 |   5900000.00 |
+-------------+-------------+--------------+
1 row in set (0.01 sec)
```

1NF (First Normal Form)

Definition:

A table is in **1NF** if:

Each column has **atomic (single) values**

Each row is **unique** (has a primary key)

No repeating groups or arrays

1. Contractor Table

Table:
Contractor (ContractorID, Name, ContactNo, Email, Address, CompanyName, PAN_GST_No)

Field	Atomic?	Explanation
ContractorID	YES	Single unique value identifying each contractor
Name	YES	Only one name per contractor
ContactNo	YES	Only one contact number (not multiple numbers in same cell)
Email	YES	Only one email address
Address	YES	Treated as one text string (not split into street, city, etc.)
CompanyName	YES	Single company name only
PAN_GST_No	YES	Single identifier (not multiple numbers)

All fields are atomic → Contractor table is in 1NF

2. Project Table

Table:
Project (ProjectID, ProjectName, StartDate, EndDate, Budget, Status, ContractorID)

Field	Atomic?	Explanation
ProjectID	YES	Unique ID per project
ProjectName	YES	One name per project
StartDate	YES	Single date
EndDate	YES	Single date
Budget	YES	Single numeric value
Status	YES	One status value (‘Ongoing’, ‘On Hold’, etc.)
ContractorID	YES	Refers to one contractor only

All atomic → 1NF satisfied

3. WorkOrder Table

Table:
WorkOrder (WorkOrderID, WorkDescription, StartDate, EndDate, Amount, ProjectID)

Field	Atomic?	Explanation
WorkOrderID	YES	Unique ID
WorkDescription	YES	Text field describing one scope of work
StartDate	YES	Single value
EndDate	YES	Single value

Field	Atomic?	Explanation
Amount	YES	One numeric value
ProjectID	YES	Refers to one project only

All atomic → 1NF satisfied

4. Bill Table

Table:
Bill (BillID, BillDate, BillAmount, TaxAmount, TotalAmount, Status, WorkOrderID, ContractorID)

Field	Atomic?	Explanation
BillID	YES	Unique per bill
BillDate	YES	Single date
BillAmount	YES	Single number
TaxAmount	YES	Single number
TotalAmount	YES	Single number
Status	YES	One value (‘Pending’, ‘Approved’, etc.)
WorkOrderID	YES	One foreign key reference
ContractorID	YES	One foreign key reference

All atomic → 1NF satisfied

5. Approver Table

Table:
Approver (ApproverID, Name, Department, Designation, Email)

Field	Atomic?	Explanation
ApproverID	YES	Unique ID
Name	YES	One name
Department	YES	Single department value
Designation	YES	One designation value
Email	YES	One email address

All atomic → 1NF satisfied

6. Approval Table

Table:
Approval (ApprovalID, ApprovalDate, Status, Remarks, BillID, ApproverID)

Field	Atomic?	Explanation
ApprovalID	YES	Unique ID
ApprovalDate	YES	One date
Status	YES	Single value ('Pending', etc.)
Remarks	YES	One comment string
BillID	YES	Single bill reference
ApproverID	YES	Single approver reference

All atomic → 1NF satisfied

7. Payment Table

Table:
Payment (PaymentID, PaymentDate, PaymentMode, AmountPaid, BillID)

Field	Atomic?	Explanation
PaymentID	YES	Unique ID
PaymentDate	YES	One date
PaymentMode	YES	Single value ('Cheque', 'Bank Transfer', etc.)
AmountPaid	YES	Single numeric value
BillID	YES	Single bill reference

All atomic → 1NF satisfied

What is 2NF (Second Normal Form)?

A table is in **2NF** if:

a. It is **already in 1NF**

b. b.**Every non-key attribute depends on the whole primary key**, not just part of it.
→ This matters **only when the primary key is composite** (made up of more than one column).

So, 2NF removes **partial dependency** — when a column depends on *part* of a composite key instead of the whole key.

1. Contractor Table

Primary Key: ContractorID
Other Columns: Name, ContactNo, Email, Address, CompanyName, PAN_GST_No

Each non-key attribute (Name, ContactNo, etc.) depends entirely on ContractorID.
There is **no composite key**, so **no partial dependency** exists.

Contractor table is in 2NF

2. Project Table

Primary Key: ProjectID
Foreign Key: ContractorID

All other columns (ProjectName, StartDate, EndDate, Budget, Status) depend on

ProjectID only.
No field depends only on part of a key (there's no composite key).

Project table is in 2NF

3. WorkOrder Table

Primary Key: WorkOrderID
Foreign Key: ProjectID

Non-key attributes (WorkDescription, StartDate, EndDate, Amount) depend fully on WorkOrderID.

They do not depend on ProjectID alone.
WorkOrder table is in 2NF

4. Bill Table

Primary Key: BillID
Foreign Keys: WorkOrderID, ContractorID

All non-key attributes (BillDate, BillAmount, TaxAmount, TotalAmount, Status) depend on the whole key BillID.

There's **no composite primary key**, so **no partial dependency**.

Bill table is in 2NF

5. Approver Table

Primary Key: ApproverID

All other attributes (Name, Department, Designation, Email) depend on the entire primary key.

Approver table is in 2NF

6. Approval Table

Primary Key: ApprovalID
Foreign Keys: BillID, ApproverID

Attributes (ApprovalDate, Status, Remarks) depend entirely on ApprovalID.
No field depends only on part of a composite key.

Approval table is in 2NF

7. Payment Table

Primary Key: PaymentID
Foreign Key: BillID

All non-key fields (PaymentDate, PaymentMode, AmountPaid) depend on PaymentID fully.
No partial dependency.

Payment table is in 2NF

What is 3NF (Third Normal Form)?

A table is in **3NF** if:
a.It is already in **2NF**, and
b.It has **no transitive dependencies** — meaning:

No non-key column depends on another non-key column.

1. Contractor Table

Primary Key: ContractorID
Attributes: Name, ContactNo, Email, Address, CompanyName, PAN_GST_No

All attributes depend directly on ContractorID.
There is no column like "City" that depends on "Address" or "CompanyName."

Contractor is in 3NF

2. Project Table

Primary Key: ProjectID
Foreign Key: ContractorID

All other columns (ProjectName, StartDate, EndDate, Budget, Status) depend only on ProjectID.
No column depends on another non-key attribute.

Project is in 3NF

3. WorkOrder Table

Primary Key: WorkOrderID
Foreign Key: ProjectID

Columns (WorkDescription, StartDate, EndDate, Amount) depend directly on WorkOrderID.
No non-key attribute depends on another non-key attribute.

WorkOrder is in 3NF

4. Bill Table

Primary Key: BillID
Foreign Keys: WorkOrderID, ContractorID
Other Fields: BillDate, BillAmount, TaxAmount, TotalAmount, Status

Check for **transitive dependency**:

TotalAmount = BillAmount + TaxAmount → derived value.

To make it **fully 3NF**, you should **not store derived/ calculated values** in a column.
So ideally, remove TotalAmount and calculate it when needed.

After removing or treating TotalAmount as a derived field → Bill is in **3NF**.

Bill (in 3NF after removing derived field)

5. Approver Table

Primary Key: ApproverID
Attributes: Name, Department, Designation, Email

All depend only on ApproverID.
No attribute depends on another non-key attribute.

Approver is in 3NF

6. Approval Table

Primary Key: ApprovalID
Foreign Keys: BillID, ApproverID
Other Columns: ApprovalDate, Status, Remarks

All depend on ApprovalID only.
No field depends on another non-key field.

Approval is in 3NF

7. Payment Table

Primary Key: PaymentID
Foreign Key: BillID
Attributes: PaymentDate, PaymentMode, AmountPaid

✅ All depend directly on PaymentID.
No field depends on another non-key field.

Payment is in 3NF