Database Schema Documentation for

DF Ticketing System

This document outlines the tables, their columns, and the relationships within the database schema. The database appears to manage various organisational operations, including financial reporting, employee data, expense management, and user access control, specifically tailored for a **DF Ticketing system**.

### 1. Core Entities

#### employees

Stores comprehensive information about individual employees.

* **employee\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each employee.
* **employee\_number** (VARCHAR(50)): Unique employee identification number.
* **first\_name** (VARCHAR(100)): Employee's first name.
* **middle\_name** (VARCHAR(100)): Employee's middle name.
* **last\_name** (VARCHAR(100)): Employee's last name.
* **display\_name** (VARCHAR(100)): Employee's preferred display name.
* **full\_name** (VARCHAR(100)): Employee's full name.
* **work\_email** (VARCHAR(100)): Employee's work email address.
* **DOB** (VARCHAR(30)): Date of birth.
* **gender** (VARCHAR(30)): Employee's gender.
* **marital\_status** (VARCHAR(30)): Marital status.
* **marriage\_date** (VARCHAR(30)): Date of marriage.
* **blood\_group** (VARCHAR(30)): Blood group.
* **physically\_handicapped** (VARCHAR(30)): Indicates if physically handicapped.
* **nationality** (VARCHAR(30)): Employee's nationality.
* **mobile** (VARCHAR(20)): Personal mobile number.
* **work\_mobile** (VARCHAR(20)): Work mobile number.
* **home\_mobile** (VARCHAR(20)): Home mobile number.
* **personal\_email** (VARCHAR(100)): Employee's personal email address.
* **skype** (VARCHAR(30)): Skype ID.
* **current\_address\_line\_1** (TEXT): Current address line 1.
* **current\_address\_line\_2** (TEXT): Current address line 2.
* **current\_address\_city** (TEXT): Current address city.
* **current\_address\_state** (TEXT): Current address state.
* **current\_address\_zip** (TEXT): Current address ZIP/postal code.
* **current\_address\_country** (TEXT): Current address country.
* **permanent\_address\_line\_1** (TEXT): Permanent address line 1.
* **permanent\_address\_line\_2** (TEXT): Permanent address line 2.
* **permanent\_address\_city** (TEXT): Permanent address city.
* **permanent\_address\_state** (TEXT): Permanent address state.
* **permanent\_address\_zip** (TEXT): Permanent address ZIP/postal code.
* **permanent\_address\_country** (TEXT): Permanent address country.
* **father\_name** (VARCHAR(30)): Father's name.
* **mother\_name** (VARCHAR(30)): Mother's name.
* **spouse\_name** (VARCHAR(30)): Spouse's name.
* **children\_name** (VARCHAR(30)): Children's names.
* **attendance\_number** (VARCHAR(30)): Attendance tracking number.
* **location** (VARCHAR(100)): Work location.
* **location\_country** (VARCHAR(30)): Work location country.
* **business\_unit** (VARCHAR(30)): Business unit.
* **department** (VARCHAR(30)): Department.
* **sub\_department** (VARCHAR(30)): Sub-department.
* **job\_title** (VARCHAR(100)): Employee's job title.
* **secondary\_job\_title** (VARCHAR(100)): Secondary job title.
* **reporting\_manager\_en** (VARCHAR(50)): Reporting manager's employee number.
* **reporting\_manager** (VARCHAR(100)): Reporting manager's name.
* **dotted\_line\_manager** (VARCHAR(30)): Dotted line manager.
* **date\_of\_joining** (VARCHAR(30)): Date of joining the organization.
* **leave\_plan** (VARCHAR(100)): Leave plan applicable to the employee.
* **band** (VARCHAR(30)): Employee band/level.
* **pay\_grade** (VARCHAR(30)): Employee's pay grade.
* **time\_type** (VARCHAR(30)): Time type (e.g., full-time, part-time).
* **worker\_type** (VARCHAR(30)): Worker type (e.g., permanent, contract).
* **shift\_policy\_name** (VARCHAR(30)): Shift policy applicable.
* **week\_off\_policy\_name** (VARCHAR(30)): Week off policy applicable.
* **tracking\_policy\_name** (VARCHAR(100)): Tracking policy applicable.
* **attendance\_capture\_scheme** (VARCHAR(30)): Attendance capture scheme.
* **holiday\_list\_name** (VARCHAR(30)): Holiday list applicable.
* **expense\_policy\_name** (VARCHAR(30)): Expense policy applicable.
* **notice\_period** (VARCHAR(30)): Notice period.
* **PAN\_number** (VARCHAR(30)): PAN number.
* **aadhaar\_number** (VARCHAR(30)): Aadhaar number.
* **pf\_number** (VARCHAR(30)): PF number.
* **UAN\_number** (VARCHAR(30)): UAN number.
* **employment\_status** (VARCHAR(30)): Current employment status.
* **exit\_date** (VARCHAR(30)): Date of exit from the organization.
* **comments** (VARCHAR(30)): Comments related to employment.
* **exit\_status** (VARCHAR(30)): Status of exit.
* **termination\_type** (VARCHAR(30)): Type of termination.
* **termination\_reason** (VARCHAR(30)): Reason for termination.
* **resignation\_note** (VARCHAR(30)): Resignation note.
* **cost\_center** (VARCHAR(225)): Associated cost center.
* **entity** (VARCHAR(100)): Associated entity.
* **emergency\_name** (VARCHAR(30)): Emergency contact name.
* **emergency\_number** (VARCHAR(30)): Emergency contact number.
* **emergency\_relation** (VARCHAR(30)): Emergency contact relationship.
* **emergency\_address** (TEXT): Emergency contact address.
* **employees\_name** (VARCHAR(100)): Redundant, possibly for clarity or historical data.
* **degree** (VARCHAR(30)): Highest degree obtained.
* **specialization** (VARCHAR(30)): Specialization of degree.
* **year\_of\_completion** (VARCHAR(30)): Year of degree completion.
* **CGPA\_or\_percentage** (VARCHAR(100)): CGPA or percentage in degree.
* **university\_or\_college** (VARCHAR(100)): University or college.
* **company\_name** (VARCHAR(100)): Previous company name.
* **previous\_job\_title** (VARCHAR(100)): Previous job title.
* **job\_duration** (VARCHAR(30)): Duration of previous job.
* **certification** (VARCHAR(30)): Certifications held.
* **year\_of\_passing** (VARCHAR(30)): Year of passing certification.
* **institute** (VARCHAR(30)): Institute of certification.
* **course\_appraisal** (VARCHAR(30)): Course appraisal details.
* **pay\_group** (VARCHAR(100)): Pay group.
* **work\_location** (TEXT): Detailed work location.
* **employee\_name\_on\_bank** (VARCHAR(100)): Name on bank account.
* **employee\_bank\_name** (VARCHAR(100)): Bank name.
* **employee\_Account\_number** (VARCHAR(20)): Bank account number.
* **employee\_IFSC** (VARCHAR(100)): Bank IFSC code.
* **created\_at** (DATETIME): Timestamp of record creation.
* **created\_by** (VARCHAR(50)): User who created the record.
* **status\_id** (INT, Index): Foreign key to master\_status for record status.
* **updated\_at** (DATETIME): Timestamp of last update.
* **updated\_by** (INT): User who last updated the record.

#### users

Manages user accounts with links to employee information and access controls.

* **user\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each user.
* **employee\_id** (INT, Index): Foreign key to employees table.
* **employee\_number** (VARCHAR(50)): Employee identification number.
* **user\_name** (VARCHAR(100)): User's full name or display name.
* **work\_email** (VARCHAR(100)): User's work email, used for login.
* **password** (TEXT): Hashed password for the user.
* **mobile** (VARCHAR(20)): User's mobile number.
* **gender** (VARCHAR(10)): User's gender.
* **PAN\_number** (VARCHAR(50)): User's PAN number.
* **aadhaar\_number** (VARCHAR(30)): User's Aadhaar number.
* **enrollment\_number** (INT): Enrollment number.
* **DOB** (VARCHAR(30)): Date of Birth.
* **entity\_id** (INT, Index): Foreign key to entities table.
* **cost\_center\_id** (INT, Index): Foreign key to cost\_center table.
* **department\_id** (INT, Index): Foreign key to departments table.
* **sub\_department\_id** (INT, Index): Foreign key to sub\_departments table.
* **job\_title** (VARCHAR(100)): User's job title.
* **secondary\_job\_title** (VARCHAR(100)): User's secondary job title.
* **manager\_user\_id** (INT, Index): Self-referencing foreign key to users for reporting manager.
* **personal\_email** (VARCHAR(100)): User's personal email.
* **father\_name** (VARCHAR(50)): Father's name.
* **mother\_name** (VARCHAR(50)): Mother's name.
* **blood\_group** (VARCHAR(20)): Blood group.
* **nationality** (VARCHAR(30)): Nationality.
* **bank\_id** (INT, Index): Foreign key to user\_bank for bank details.
* **pf\_id** (INT, Index): Foreign key to pf\_details for PF details.
* **pay\_group** (VARCHAR(100)): Pay group.
* **band** (VARCHAR(30)): User's band/level.
* **pay\_grade** (VARCHAR(100)): User's pay grade.
* **work\_location** (TEXT): User's work location.
* **sended\_email** (INT(11)): Flag for email sending status.
* **status\_id** (INT, Index): Foreign key to master\_status for user status.
* **timestamp** (DATETIME): Timestamp of record creation or last significant action.

### 2. Organizational Structure

#### organization

Defines the top-level organizational units.

* **org\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each organization.
* **org\_name** (VARCHAR(50)): Name of the organization.
* **org\_code** (VARCHAR(50)): Code for the organization.
* **address** (TEXT): Organization's address.
* **status\_id** (INT, Index): Foreign key to master\_status for organization status.
* **logo** (VARCHAR(255)): Path to the organization's logo.

#### entities

Represents entities within an organization.

* **entity\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each entity.
* **org\_id** (INT, Index): Foreign key to organization table.
* **entity\_name** (VARCHAR(255)): Name of the entity.
* **address** (TEXT): Entity's address.
* **entity\_code** (VARCHAR(50)): Code for the entity.
* **status\_id** (INT, Index): Foreign key to master\_status for entity status.
* **logo** (VARCHAR(50)): Path to the entity's logo.
* **description** (TEXT): Description of the entity.

#### departments

Manages different departments within the organization.

* **department\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each department.
* **department\_name** (VARCHAR(50)): Name of the department.
* **status\_id** (INT, Index): Foreign key to master\_status for department status.
* **description** (TEXT): Description of the department.

#### sub\_departments

Manages sub-departments nested under departments.

* **sub\_department\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each sub-department.
* **department\_id** (INT, Index): Foreign key to departments table.
* **sub\_department\_name** (VARCHAR(50)): Name of the sub-department.
* **status\_id** (INT, Index): Foreign key to master\_status for sub-department status.
* **description** (TEXT): Description of the sub-department.

#### cost\_center

Defines cost centers for financial tracking.

* **cost\_center\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each cost center.
* **cost\_center\_name** (VARCHAR(255)): Name of the cost center.
* **status\_id** (INT, Index): Foreign key to master\_status for cost center status.
* **description** (TEXT): Description of the cost center.

#### academic\_year

Defines academic or financial years for reporting and categorization.

* **year\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each academic year.
* **year\_name** (VARCHAR(20)): Name of the academic year (e.g., "2023-2024").
* **status\_id** (INT, Index): Foreign key to master\_status for academic year status.
* **start\_date** (DATETIME): Start date of the academic year.
* **end\_date** (DATETIME): End date of the academic year.

### 3. Reporting and Tickets

#### reports

Main table for reports, likely high-level financial or operational summaries.

* **report\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each report.
* **user\_id** (INT, Index): Foreign key to users indicating the report creator.
* **year\_id** (INT, Index): Foreign key to academic\_year for the report's year.
* **org\_id** (INT, Index): Foreign key to organization for the associated organization.
* **entity\_id** (INT, Index): Foreign key to entities for the associated entity.
* **report\_code** (VARCHAR(30), UNIQUE): Unique code for the report.
* **manager\_id** (INT, Index): ID of the manager associated with the report.
* **name** (VARCHAR(255)): Name of the report.
* **status\_id** (INT, Index): Foreign key to master\_status for report status.
* **process\_status\_id** (INT, Index): Foreign key to master\_status for report processing status.
* **created\_at** (DATETIME): Timestamp of report creation.
* **start\_date** (DATETIME): Start date of the report period.
* **end\_date** (DATETIME): End date of the report period.
* **description** (TEXT): Description of the report.
* **updated\_at** (DATETIME): Timestamp of last update.

#### report\_history

Stores historical changes made to reports.

* **history\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for report history entries.
* **report\_id** (INT, Index): Foreign key to reports table.
* **name** (VARCHAR(255)): Name of the report at the time of history entry.
* **start\_date** (DATETIME): Start date of the report at the time of history entry.
* **description** (TEXT): Description of the report at the time of history entry.
* **end\_date** (DATETIME): End date of the report at the time of history entry.
* **updated\_by** (INT): User who made the update.
* **updated\_at** (DATETIME): Timestamp of the update.
* **edit\_description** (TEXT): Description of the changes made.

#### report\_logs

Logs actions and status changes related to reports.

* **log\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for report log entries.
* **report\_id** (INT, Index): Foreign key to reports table.
* **pre\_status\_id** (INT, Index): Previous status of the report (FK to master\_status).
* **aft\_status\_id** (INT, Index): After status of the report (FK to master\_status).
* **pre\_process\_id** (INT, Index): Previous process status of the report (FK to master\_status).
* **aft\_process\_id** (INT, Index): After process status of the report (FK to master\_status).
* **level\_id** (INT, Index): Foreign key to levels for the approval level.
* **action\_id** (INT, Index): Foreign key to master\_status (likely for action type).
* **created\_by** (INT): User who initiated the action.
* **created\_at** (DATETIME): Timestamp of the log entry.
* **remarks** (TEXT): Remarks about the action.

#### tickets

Central table for various types of tickets (e.g., expense, budget, contribution).

* **ticket\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each ticket.
* **entity\_id** (INT, Index): Foreign key to entities table.
* **report\_id** (INT, Index): Foreign key to reports table, if part of a report.
* **year\_id** (INT, Index): Foreign key to academic\_year for the ticket's year.
* **user\_id** (INT, Index): Foreign key to users indicating the ticket creator.
* **ticket\_number** (VARCHAR(45), UNIQUE): Unique ticket number.
* **cost\_center\_id** (INT, Index): Foreign key to cost\_center table.
* **expense\_id** (INT, Index): Foreign key to expense table.
* **budget\_id** (INT, Index): Foreign key to budget\_type table.
* **contribution\_id** (INT, Index): Foreign key to contribution\_type table.
* **exp\_catg\_id** (INT, Index): Foreign key to expense\_category for the expense category.
* **granted\_amount** (VARCHAR(100)): Amount granted for the ticket.
* **status\_id** (INT, Index): Foreign key to master\_status for ticket status.
* **process\_status\_id** (INT, Index): Foreign key to master\_status for ticket processing status.
* **description** (TEXT): Description of the ticket.
* **created\_at** (DATETIME): Timestamp of ticket creation.
* **updated\_by** (INT): User who last updated the ticket.
* **updated\_at** (DATETIME): Timestamp of last update.

#### ticket\_history

Stores historical changes made to tickets.

* **ticket\_hst\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for ticket history entries.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **cost\_center\_id** (INT): Cost center ID at the time of history entry.
* **expense\_id** (INT): Expense ID at the time of history entry.
* **budget\_id** (INT): Budget ID at the time of history entry.
* **contribution\_id** (INT): Contribution ID at the time of history entry.
* **description** (TEXT): Description of the ticket at the time of history entry.
* **updated\_by** (INT): User who made the update.
* **updated\_at** (DATETIME): Timestamp of the update.
* **edit\_description** (TEXT): Description of the changes made.

#### ticket\_logs

Logs actions and status changes related to tickets.

* **log\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for ticket log entries.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **pre\_status\_id** (INT, Index): Previous status of the ticket (FK to master\_status).
* **aft\_status\_id** (INT, Index): After status of the ticket (FK to master\_status).
* **pre\_process\_id** (INT, Index): Previous process status of the ticket (FK to master\_status).
* **aft\_process\_id** (INT, Index): After process status of the ticket (FK to master\_status).
* **level\_id** (INT, Index): Foreign key to levels for the approval level.
* **action\_id** (INT, Index): Foreign key to master\_status (likely for action type).
* **created\_by** (INT): User who initiated the action.
* **created\_at** (DATETIME): Timestamp of the log entry.
* **remarks** (TEXT): Remarks about the action.

### 4. Financial and Expense Management

#### expense\_category

Defines categories for expenses.

* **expense\_category\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each expense category.
* **category** (VARCHAR(30)): Name of the expense category.
* **status\_id** (INT, Index): Foreign key to master\_status for category status.
* **description** (TEXT): Description of the expense category.

#### categories

General categories, possibly for ledgers.

* **category\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each category.
* **ledger\_code** (VARCHAR(30)): Code for the ledger category.
* **category\_type** (VARCHAR(30)): Type of category.
* **status\_id** (INT, Index): Foreign key to master\_status for category status.

#### ledgers

Manages ledger accounts.

* **ledger\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each ledger.
* **entity\_id** (INT, Index): Foreign key to entities table.
* **ledger\_code** (VARCHAR(50)): Code for the ledger.
* **ledger\_name** (VARCHAR(100)): Name of the ledger.
* **ledger\_type** (VARCHAR(50)): Type of the ledger.
* **category\_id** (INT, Index): Foreign key to categories table.
* **status\_id** (INT, Index): Foreign key to master\_status for ledger status.

#### re\_ticket\_details

Details for reimbursement tickets.

* **reimb\_dtls\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for reimbursement details.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **category\_id** (INT, Index): Foreign key to categories table.
* **ledger\_id** (INT, Index): Foreign key to ledgers table.
* **total\_expense** (VARCHAR(100)): Total expense amount.
* **m\_granted\_amount** (VARCHAR(100)): Manager granted amount.
* **f\_granted\_amount** (VARCHAR(100)): Finance granted amount.
* **f\_granted\_by** (INT): User who granted the finance amount.
* **exp\_pay\_date** (DATETIME): Expected payment date.
* **reimbursed** (VARCHAR(100)): Reimbursed amount.
* **status\_id** (INT, Index): Foreign key to master\_status for reimbursement status.
* **created\_at** (DATETIME): Timestamp of record creation.
* **description** (TEXT): Description of the reimbursement.
* **updated\_by** (INT): User who last updated the record.
* **updated\_at** (DATETIME): Timestamp of last update.

#### reimbursement\_history

History of reimbursement details.

* **reimb\_hst\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for reimbursement history entries.
* **reimb\_dtls\_id** (INT, Index): Foreign key to re\_ticket\_details.
* **ticket\_id** (INT, Index): Foreign key to tickets.
* **category\_id** (INT, Index): Foreign key to categories.
* **ledger\_id** (INT, Index): Foreign key to ledgers.
* **total\_expense** (VARCHAR(100)): Total expense.
* **m\_granted\_amount** (VARCHAR(100)): Manager granted amount.
* **f\_granted\_amount** (VARCHAR(100)): Finance granted amount.
* **reimbursed** (VARCHAR(100)): Reimbursed amount.
* **hotel\_name** (VARCHAR(200)): Hotel name (for accommodation).
* **address** (TEXT): Address (for accommodation).
* **check\_in\_date** (DATETIME): Check-in date (for accommodation).
* **check\_out\_date** (DATETIME): Check-out date (for accommodation).
* **stay\_days** (INT): Number of stay days (for accommodation).
* **vehicle\_id** (INT, Index): Foreign key to vehicles (for travel).
* **start\_odo** (VARCHAR(100)): Start odometer reading (for travel).
* **end\_odo** (VARCHAR(100)): End odometer reading (for travel).
* **start\_odo\_path** (TEXT): Path to start odometer image (for travel).
* **end\_odo\_path** (TEXT): Path to end odometer image (for travel).
* **distance** (INT): Distance traveled (for travel).
* **from** (VARCHAR(100)): Travel from location.
* **to** (VARCHAR(100)): Travel to location.
* **travel\_date** (DATETIME): Travel date.
* **status\_id** (INT, Index): Foreign key to master\_status for history entry status.
* **updated\_by** (INT): User who updated the record.
* **updated\_at** (DATETIME): Timestamp of the update.
* **description** (TEXT): Description of the reimbursement history.
* **edit\_description** (TEXT): Description of changes made.

#### adv\_ticket\_details

Details for advance tickets.

* **adv\_dtls\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for advance ticket details.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **amount** (INT): Advance amount.
* **manager\_file** (TEXT): Path to manager-related file.
* **description** (TEXT): Description of the advance.

#### pro\_ticket\_details

Details for procurement tickets.

* **pro\_dtls\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for procurement details.
* **ticket\_id** (INT(11), Index): Foreign key to tickets table.
* **procurement\_type** (VARCHAR(30)): Type of procurement.
* **particulars** (VARCHAR(30)): Particulars of the procurement.
* **description** (TEXT): Description of the procurement.
* **upload\_file** (TEXT): Path to uploaded procurement file.

#### bills

Records of bills associated with tickets.

* **bill\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each bill.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **exp\_catg\_id** (INT, Index): Foreign key to expense\_category for the expense category.
* **ticket\_dtls\_id** (INT): ID of the related ticket detail (e.g., reimb\_dtls\_id).
* **bill\_number** (VARCHAR(45)): Bill number.
* **bill\_amount** (VARCHAR(100)): Amount of the bill.
* **bill\_date** (DATETIME): Date of the bill.
* **status\_id** (INT, Index): Foreign key to master\_status for bill status.
* **bill\_path** (TEXT): Path to the scanned bill image.
* **created\_at** (DATETIME): Timestamp of record creation.
* **created\_by** (INT): User who created the record.
* **updated\_at** (DATETIME): Timestamp of last update.
* **updated\_by** (INT): User who last updated the record.

#### bill\_history

History of changes to bills.

* **bill\_hst\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for bill history entries.
* **bill\_id** (INT, Index): Foreign key to bills table.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **exp\_catg\_id** (INT, Index): Foreign key to expense\_category for the expense category.
* **ticket\_dtls\_id** (INT): ID of the related ticket detail.
* **bill\_path** (TEXT): Path to the scanned bill image at the time of history entry.
* **bill\_number** (VARCHAR(45)): Bill number at the time of history entry.
* **bill\_amount** (VARCHAR(100)): Amount of the bill at the time of history entry.
* **bill\_date** (DATETIME): Date of the bill at the time of history entry.
* **status\_id** (INT, Index): Foreign key to master\_status for history entry status.
* **created\_at** (DATETIME): Timestamp of record creation.
* **created\_by** (INT): User who created the record.
* **updated\_at** (DATETIME): Timestamp of last update.
* **updated\_by** (INT): User who last updated the record.

### 5. Payments and Banking

#### payments

Records individual payments made.

* **payment\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each payment.
* **ticket\_id** (INT, Index): Foreign key to tickets table.
* **transaction\_id** (VARCHAR(45), UNIQUE): Unique transaction identifier.
* **route\_id** (INT, Index): Foreign key to payment\_route table.
* **pay\_type\_id** (INT, Index): Foreign key to payment\_type table.
* **paid\_from** (INT): User or entity from whom payment was made.
* **paid\_to** (INT): User or entity to whom payment was made.
* **paid\_from\_bank** (INT): Bank ID from which payment was made (likely from organization\_bank or user\_bank).
* **paid\_to\_bank** (INT): Bank ID to which payment was made (likely from organization\_bank or user\_bank or vendor\_bank).
* **amount** (VARCHAR(30)): Payment amount.
* **UTR\_number** (VARCHAR(100)): Unique Transaction Reference number.
* **status\_id** (INT, Index): Foreign key to master\_status for payment status.
* **payment\_mode\_id** (INT, Index): Foreign key to payment\_mode table.
* **payment\_date** (DATETIME): Date of the payment.
* **created\_by** (INT, Index): User who created the payment record.
* **created\_at** (DATETIME): Timestamp of record creation.
* **level\_id** (INT, Index): Foreign key to levels for the approval level.
* **description** (TEXT): Description of the payment.

#### payment\_mode

Defines different modes of payment (e.g., UPI, Bank Transfer).

* **mode\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for payment mode.
* **mode\_name** (VARCHAR(50)): Name of the payment mode.
* **status\_id** (INT, Index): Foreign key to master\_status for status.

#### payment\_route

Defines payment routes (e.g., internal transfer, external transfer).

* **route\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for payment route.
* **route\_from** (VARCHAR(100)): Origin of the payment route.
* **route\_to** (VARCHAR(100)): Destination of the payment route.
* **status\_id** (INT, Index): Foreign key to master\_status for status.
* **description** (TEXT): Description of the payment route.

#### payment\_type

Categorizes types of payments.

* **pay\_type\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for payment type.
* **pay\_type** (VARCHAR(30)): Name of the payment type.

#### user\_bank

Stores bank account details for users.

* **bank\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for user bank details.
* **user\_id** (INT, Index): Foreign key to users table.
* **name\_on\_bank** (VARCHAR(100)): Name on the bank account.
* **bank\_name** (VARCHAR(100)): Bank name.
* **account\_number** (VARCHAR(20)): Bank account number.
* **IFSC** (VARCHAR(100)): IFSC code.
* **status\_id** (INT, Index): Foreign key to master\_status for status.
* **created\_at** (DATETIME): Timestamp of record creation.
* **updated\_at** (DATETIME): Timestamp of last update.

#### pf\_details

Stores Provident Fund (PF) details for employees.

* **pf\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for PF details.
* **pf\_establishment\_id** (INT(11)): PF establishment ID.
* **pf\_details\_available** (VARCHAR(255)): Indicates if PF details are available.
* **pf\_number** (VARCHAR(50)): PF account number.
* **pf\_joining\_date** (DATE): Date of joining PF.
* **name\_on\_pf\_account** (VARCHAR(50)): Name registered on the PF account.
* **UAN** (VARCHAR(50)): Universal Account Number (UAN).
* **status\_id** (INT, Index): Foreign key to master\_status for status.
* **created\_at** (DATETIME): Timestamp of record creation.
* **updated\_at** (DATETIME): Timestamp of last update.

#### organization\_bank

Stores bank account details for organizations and entities.

* **org\_bank\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for organization bank details.
* **org\_id** (INT, Index): Foreign key to organization table.
* **entity\_id** (INT, Index): Foreign key to entities table.
* **account\_type** (VARCHAR(50)): Type of account (e.g., Savings, Current).
* **client\_code** (VARCHAR(100)): Client code for the bank.
* **entity\_bank\_name** (VARCHAR(100)): Bank name.
* **entity\_bank\_account\_no** (VARCHAR(50)): Bank account number.
* **entity\_name\_on\_bank** (VARCHAR(100)): Name on the bank account.
* **entity\_bank\_ifsc** (VARCHAR(50)): IFSC code.
* **bank\_address** (TEXT): Bank's address.
* **bank\_contact\_no** (VARCHAR(30)): Bank's contact number.
* **bank\_contact\_person** (VARCHAR(100)): Bank's contact person.
* **entity\_bank\_format** (TEXT): Bank format details.
* **email** (VARCHAR(100)): Bank's email.
* **status\_id** (INT, Index): Foreign key to master\_status for status.
* **created\_at** (DATETIME): Timestamp of record creation.
* **updated\_at** (DATETIME): Timestamp of last update.

### 6. Vendor Management

#### vendors

Information about external vendors.

* **vendor\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each vendor.
* **vendor\_name** (VARCHAR(30)): Name of the vendor.
* **phone** (INT(10)): Vendor's phone number.
* **email** (VARCHAR(30)): Vendor's email address.
* **address** (TEXT): Vendor's address.
* **GST** (VARCHAR(50)): Vendor's GST number.
* **vendor\_type** (VARCHAR(30)): Type of vendor.

#### vendor\_bank

Bank details for vendors.

* **vendor\_bank\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for vendor bank details.
* **vendor\_id** (INT, Index): Foreign key to vendors table.
* **name\_on\_bank** (VARCHAR(100)): Name on the bank account.
* **bank\_name** (VARCHAR(30)): Bank name.
* **bank\_branch** (VARCHAR(20)): Bank branch.
* **account\_number** (VARCHAR(20)): Bank account number.
* **IFSC** (VARCHAR(100)): IFSC code.

### 7. Travel, Accommodation, and Food Details

#### vehicles

Defines types of vehicles for travel expenses.

* **vehicle\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each vehicle.
* **vehicle\_type** (VARCHAR(30)): Type of vehicle (e.g., "Car", "Bike").
* **per\_km\_amount** (INT): Cost per kilometer for the vehicle type.
* **description** (TEXT): Description of the vehicle type.

#### travels

Detailed records of travel expenses.

* **travel\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each travel record.
* **reimb\_dtls\_id** (INT, Index): Foreign key to re\_ticket\_details table.
* **vehicle\_id** (INT, Index): Foreign key to vehicles table.
* **from** (VARCHAR(100)): Origin of travel.
* **to** (VARCHAR(100)): Destination of travel.
* **start\_odo** (VARCHAR(100)): Start odometer reading.
* **end\_odo** (VARCHAR(100)): End odometer reading.
* **start\_odo\_path** (TEXT): Path to start odometer image.
* **end\_odo\_path** (TEXT): Path to end odometer image.
* **distance** (VARCHAR(10)): Distance traveled.
* **travel\_date** (DATETIME): Date of travel.

#### accommodation

Detailed records of accommodation expenses.

* **accommodation\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each accommodation record.
* **reimb\_dtls\_id** (INT, Index): Foreign key to re\_ticket\_details table.
* **hotel\_name** (VARCHAR(200)): Name of the hotel.
* **address** (TEXT): Hotel address.
* **check\_in\_date** (DATETIME): Check-in date.
* **check\_out\_date** (DATETIME): Check-out date.
* **stay\_days** (INT): Number of days stayed.

#### food

Detailed records of food expenses.

* **food\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each food expense record.
* **reimb\_dtls\_id** (INT, Index): Foreign key to re\_ticket\_details table.
* **date** (DATETIME): Date of the food expense.

#### reimb\_others

Other reimbursement details not covered by specific categories.

* **reimb\_other\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for other reimbursement details.
* **reimb\_dtls\_id** (INT, Index): Foreign key to re\_ticket\_details table.
* **date** (DATETIME): Date of the expense.
* **remarks** (TEXT): Remarks about the expense.

### 8. User Access and System Settings

#### master\_status

A lookup table for various statuses used across the database.

* **status\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each status.
* **status** (VARCHAR(50)): Name of the status (e.g., "Active", "Inactive", "Approved", "Pending").
* **description** (TEXT): Description of the status.

#### roles

Defines different user roles.

* **role\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each role.
* **role\_name** (VARCHAR(50)): Name of the role (e.g., "Admin", "Employee", "Manager").
* **created\_at** (DATETIME): Timestamp of role creation.
* **create\_by** (VARCHAR(50)): User who created the role.
* **status\_id** (INT, Index): Foreign key to master\_status for role status.
* **description** (TEXT): Description of the role.

#### user\_roles

Assigns roles to users.

* **user\_role\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for user-role assignment.
* **user\_id** (INT, Index): Foreign key to users table.
* **role\_id** (INT, Index): Foreign key to roles table.
* **created\_at** (DATETIME): Timestamp of assignment.
* **created\_by** (VARCHAR(50)): User who made the assignment.
* **status\_id** (INT, Index): Foreign key to master\_status for assignment status.

#### menus

Defines the menu structure for the application.

* **menu\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each menu item.
* **parent\_id** (INT(11)): Self-referencing ID for parent menu item (for hierarchical menus).
* **menu\_name** (VARCHAR(50)): Name of the menu item.
* **icon** (VARCHAR(255)): Icon associated with the menu item.
* **sort** (VARCHAR(50)): Sorting order for menu items.
* **create\_by** (INT(11)): User who created the menu item.
* **status\_id** (INT(11), Index): Foreign key to master\_status for menu status.
* **action** (VARCHAR(30)): Action associated with the menu item.
* **description** (TEXT): Description of the menu item.

#### role\_menu

Links roles to accessible menu items.

* **role\_menu\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for role-menu mapping.
* **role\_id** (INT, Index): Foreign key to roles table.
* **menu\_id** (INT, Index): Foreign key to menus table.
* **status\_id** (VARCHAR(50), Index): Status of the role-menu mapping (likely FK to master\_status, but defined as VARCHAR).

#### role\_action

Defines actions allowed for specific roles.

* **role\_action\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for role-action mapping.
* **role\_id** (INT, Index): Foreign key to roles table.
* **action** (VARCHAR(100)): Specific action allowed (e.g., "create", "edit", "delete").
* **status\_id** (INT, Index): Foreign key to master\_status for status.

#### levels

Defines different approval levels.

* **level\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each level.
* **level\_name** (VARCHAR(100)): Name of the level (e.g., "Level 1 Approval", "Final Approval").
* **status\_id** (INT(11), Index): Foreign key to master\_status for level status.
* **description** (TEXT): Description of the level.

#### login\_details

Logs user login and logout times.

* **login\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each login session.
* **user\_id** (INT, Index): Foreign key to users table.
* **login\_type** (VARCHAR(30)): Type of login (e.g., "Web", "Mobile").
* **login\_time** (DATETIME): Timestamp of login.
* **logout\_time** (DATETIME): Timestamp of logout.
* **status\_id** (INT(11), Index): Foreign key to master\_status for login status.

### 9. Academic Year & Organizational Associations

#### academic\_organization

Links academic years to organizations.

* **id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for this mapping.
* **org\_id** (INT, Index): Foreign key to organization table.
* **year\_id** (INT, Index): Foreign key to academic\_year table.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### academic\_entity

Links academic years to entities.

* **id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for this mapping.
* **entity\_id** (INT, Index): Foreign key to entities table.
* **year\_id** (INT, Index): Foreign key to academic\_year table.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### academic\_expense\_category

Links academic years to expense categories.

* **id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for this mapping.
* **expence\_category\_id** (INT, Index): Foreign key to expense\_category table.
* **year\_id** (INT, Index): Foreign key to academic\_year table.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### academic\_cost\_center

Links academic years and entities to cost centers.

* **id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for this mapping.
* **cost\_center\_id** (INT, Index): Foreign key to cost\_center table.
* **year\_id** (INT, Index): Foreign key to academic\_year table.
* **entity\_id** (INT, Index): Foreign key to entities table.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

### 10. Budget and Contribution Types

#### budget\_type

Defines types of budgets.

* **budget\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for budget type.
* **budget\_name** (VARCHAR(255)): Name of the budget type.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### expense

Defines general expense types.

* **expense\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for expense type.
* **expense\_name** (VARCHAR(255)): Name of the expense type.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### contribution\_type

Defines types of contributions.

* **contribution\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for contribution type.
* **contribution\_name** (VARCHAR(255)): Name of the contribution type.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

### 11. Tally Integration

#### tally\_booking

Records entries for Tally integration (accounting software).

* **tally\_booking\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for Tally booking.
* **voucher\_type** (VARCHAR(100)): Type of voucher (e.g., "Payment", "Receipt").
* **ticket\_id** (INT(11), Index): Foreign key to tickets table.
* **expense\_category\_id** (INT(11), Index): Foreign key to expense\_category table.
* **ticket\_dtls\_id** (INT(11)): ID of related ticket details (e.g., reimb\_dtls\_id).
* **payment\_id** (INT(11), Index): Foreign key to payments table.
* **paid\_amount** (VARCHAR(100)): Amount paid.
* **tally\_pay\_id** (INT(11), Index): Foreign key to tally\_payment table.
* **dr/cr** (VARCHAR(10)): Debit or Credit indicator.
* **status\_id** (INT(11), Index): Foreign key to master\_status for booking status.
* **process\_status\_id** (INT(11), Index): Foreign key to master\_status for processing status.
* **created\_by** (INT(11)): User who created the booking.
* **created\_at** (DATETIME): Timestamp of creation.
* **updated\_by** (INT(11)): User who last updated the booking.
* **updated\_at** (DATETIME): Timestamp of last update.
* **update\_remarks** (TEXT): Remarks about the update.

#### tally\_payment

General payment entries for Tally.

* **tally\_pay\_id** (INT(11), PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for Tally payment.
* **voucher\_type** (VARCHAR(100)): Type of voucher.
* **ledger** (VARCHAR(100)): Ledger account name.
* **amount** (VARCHAR(225)): Amount.
* **dr/cr** (VARCHAR(10)): Debit or Credit indicator.
* **led\_narration** (TEXT): Ledger narration.
* **narration** (TEXT): General narration.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.
* **created\_by** (INT(11)): User who created the record.
* **created\_at** (DATETIME): Timestamp of creation.
* **updated\_by** (INT(11)): User who last updated the record.
* **updated\_at** (DATETIME): Timestamp of last update.
* **update\_remarks** (TEXT): Remarks about the update.

#### tally\_pay\_bank

Bank-specific payment entries for Tally.

* **tally\_pay\_bank\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for Tally bank payment.
* **voucher\_type** (VARCHAR(100)): Type of voucher.
* **tally\_pay\_id** (INT(11), Index): Foreign key to tally\_payment table.
* **route\_id** (INT(11), Index): Foreign key to payment\_route table.
* **bank\_id** (INT(11), Index): Foreign key to bank tables (e.g., organization\_bank, user\_bank).
* **ledger** (VARCHAR(100)): Ledger account name.
* **amount** (VARCHAR(225)): Amount.
* **dr/cr** (VARCHAR(10)): Debit or Credit indicator.
* **led\_narration** (TEXT): Ledger narration.
* **narration** (TEXT): General narration.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.
* **created\_by** (INT(11)): User who created the record.
* **created\_at** (DATETIME): Timestamp of creation.
* **updated\_by** (INT(11)): User who last updated the record.
* **updated\_at** (DATETIME): Timestamp of last update.
* **update\_remarks** (TEXT): Remarks about the update.

### 12. Support & History

#### faq

Frequently Asked Questions or support queries.

* **faq\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for each FAQ/query.
* **user\_id** (INT(11), Index): Foreign key to users for the user who raised the query.
* **faq\_description** (TEXT): Description of the FAQ or query.
* **created\_at** (DATETIME): Timestamp of creation.
* **solved\_description** (TEXT): Description of the solution.
* **solved\_by** (INT(11)): User who solved the query.
* **solved\_at** (DATETIME): Timestamp when the query was solved.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### faq\_files

Files attached to FAQs.

* **faq\_file\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for FAQ files.
* **faq\_id** (INT(11), Index): Foreign key to faq table.
* **file\_path** (TEXT): Path to the attached file.
* **faq\_type** (VARCHAR(50)): Type of FAQ file.

#### query\_chat

Chat logs for support queries.

* **query\_chat\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for chat messages.
* **user\_id** (INT(11), Index): Foreign key to users table.
* **faq\_id** (INT(11), Index): Foreign key to faq table.
* **query\_chat** (TEXT): Content of the chat message.
* **created\_at** (DATETIME): Timestamp of the chat message.
* **status\_id** (INT(11), Index): Foreign key to master\_status for status.

#### employee\_history

Historical records of employee data.

* **employee\_history\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for employee history entries.
* **employee\_id** (INT, Index): Foreign key to employees table.
* *(All columns from employees are duplicated here to capture historical states)*
* **updated\_at** (DATETIME): Timestamp of the update.
* **updated\_by** (INT): User who updated the record.

#### user\_history

Historical records of user data.

* **user\_history\_id** (INT, PK, NOT NULL, AUTO\_INCREMENT): Unique identifier for user history entries.
* **user\_id** (INT, Index): Foreign key to users table.
* *(All columns from users are duplicated here to capture historical states)*
* **updated\_at** (DATETIME): Timestamp of the update.
* **updated\_by** (INT): User who updated the record.

### Key Relationships (Foreign Keys)

The Ref statements clearly define the foreign key relationships between tables. Here's a summary of the most important relationships, highlighting how data flows and connects across the system:

* **User Management:**
  + users links to employees, cost\_center, departments, sub\_departments, entities, user\_bank, pf\_details, and master\_status.
  + user\_roles links users to roles.
  + login\_details tracks user logins from users.
* **Organizational Hierarchy:**
  + entities link to organization.
  + sub\_departments link to departments.
* **Reporting Flow:**
  + reports link to users, academic\_year, organization, entities, and master\_status (for status and process status).
  + report\_history and report\_logs track changes and actions on reports.
* **Ticket Processing:**
  + tickets link to reports, academic\_year, users, cost\_center, expense, budget\_type, contribution\_type, expense\_category, entities, and master\_status (for status and process status).
  + ticket\_history and ticket\_logs track changes and actions on tickets.
* **Expense Details:**
  + re\_ticket\_details links to tickets, categories, ledgers, and master\_status.
  + travels, accommodation, food, and reimb\_others provide specific details for re\_ticket\_details.
  + reimbursement\_history tracks changes to re\_ticket\_details.
* **Financial Transactions:**
  + payments link to tickets, payment\_route, payment\_mode, payment\_type, levels, users, and master\_status.
  + user\_bank, organization\_bank, and vendor\_bank store banking details.
  + pf\_details stores provident fund information.
* **Master Data:**
  + master\_status is a central lookup table used by many other tables for status tracking.
* **Tally Integration:**
  + tally\_booking links to tickets, expense\_category, payments, tally\_payment, and master\_status.
  + tally\_pay\_bank links to tally\_payment, payment\_route, and bank tables.

This comprehensive schema provides a robust framework for managing various business processes within an organization.