# df\_ticketing System - Date-wise Changes Documentation

This document provides a detailed overview of the schema changes implemented in the df\_ticketing database, categorized by date.

## 08/03/2025

### Database df\_ticketing created

**Summary:** df\_ticketing database created

## 15/04/2025

### Table: budget\_type (New Table)

CREATE TABLE budget\_type (  
 budget\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 budget\_name VARCHAR(255) NOT NULL,  
 status\_id INT,  
 FOREIGN KEY (status\_id) REFERENCES master\_status(status\_id)  
);

**Summary:** Created a new table budget\_type to store different budget categories, linked to master\_status.

### Table: expense (New Table)

CREATE TABLE expense (  
 expense\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 expense\_name VARCHAR(255) NOT NULL,  
 status\_id INT,  
 FOREIGN KEY (status\_id) REFERENCES master\_status(status\_id)  
);

**Summary:** Created a new table expense to categorize various expenses, linked to master\_status.

### Table: contribution\_type (New Table)

CREATE TABLE contribution\_type (  
 contribution\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 contribution\_name VARCHAR(255) NOT NULL,  
 status\_id INT,  
 FOREIGN KEY (status\_id) REFERENCES master\_status(status\_id)  
);

**Summary:** Created a new table contribution\_type to define different types of contributions, linked to master\_status.

### Table: ticket\_history (Alter Table)

ALTER TABLE ticket\_history  
ADD COLUMN expense\_id INT AFTER cost\_center\_id,  
ADD COLUMN budget\_id INT AFTER expense\_id,  
ADD COLUMN contribution\_id INT AFTER budget\_id;

**Summary:** Added three new columns (expense\_id, budget\_id, contribution\_id) to ticket\_history to link tickets to expense, budget, and contribution types.

### Table: ledgers (Alter Table)

ALTER TABLE ledgers  
ADD COLUMN category\_id INT AFTER ledger\_name,  
ADD FOREIGN KEY (category\_id) REFERENCES categories(category\_id);

**Summary:** Added a category\_id column to ledgers to categorize ledger entries and established a foreign key relationship with the categories table.

### Table: reimb\_others (New Table)

CREATE TABLE reimb\_others (  
 reimb\_other\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 reimb\_dtls\_id INT,  
 date datetime,  
 remarks text,  
 FOREIGN KEY (reimb\_dtls\_id) REFERENCES re\_ticket\_details(reimb\_dtls\_id)  
);

**Summary:** Created a new table reimb\_others to store additional reimbursement details, linked to re\_ticket\_details.

## 21/04/2025

### Table: organization\_bank (New Table)

CREATE TABLE organization\_bank (  
 org\_bank\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 org\_id INT DEFAULT NULL,  
 entity\_id INT DEFAULT NULL,  
 org\_bank\_name VARCHAR(100) DEFAULT NULL,  
 org\_bank\_account\_no VARCHAR(50) DEFAULT NULL,  
 org\_name\_on\_bank VARCHAR(100) DEFAULT NULL,  
 org\_bank\_ifsc VARCHAR(50) DEFAULT NULL,  
 org\_bank\_format TEXT DEFAULT NULL,  
 status\_id INT DEFAULT NULL,  
 created\_at DATETIME DEFAULT NULL,  
 updated\_at DATETIME DEFAULT NULL,  
 FOREIGN KEY (status\_id) REFERENCES master\_status(status\_id),  
 FOREIGN KEY (org\_id) REFERENCES organization(org\_id),  
 FOREIGN KEY (entity\_id) REFERENCES entities(entity\_id)  
);

**Summary:** Created a new table organization\_bank to store bank details for organizations and entities, with foreign keys to master\_status, organization, and entities.

### Table: payment\_type (New Table)

CREATE TABLE payment\_type (  
 pay\_type\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 pay\_type VARCHAR(30) DEFAULT NULL  
);

**Summary:** Created a new table payment\_type to define different payment types.

### Table: payments (Alter Table)

ALTER TABLE payments   
ADD COLUMN pay\_type\_id INT DEFAULT NULL AFTER route\_id,  
ADD CONSTRAINT payments\_ibfk\_pay\_type\_id FOREIGN KEY (pay\_type\_id) REFERENCES payment\_type(pay\_type\_id);  
  
ALTER TABLE payments   
ADD COLUMN org\_bank\_id INT DEFAULT NULL AFTER pay\_type\_id,  
ADD CONSTRAINT payments\_ibfk\_org\_bank\_id FOREIGN KEY (org\_bank\_id) REFERENCES organization\_bank(org\_bank\_id);  
  
ALTER TABLE payments   
ADD COLUMN user\_bank\_id INT DEFAULT NULL AFTER org\_bank\_id,  
ADD CONSTRAINT payments\_ibfk\_user\_bank\_id FOREIGN KEY (user\_bank\_id) REFERENCES user\_bank(bank\_id);  
  
ALTER TABLE payments   
ADD COLUMN vendor\_bank\_id INT DEFAULT NULL AFTER user\_bank\_id,  
ADD CONSTRAINT payments\_ibfk\_vendor\_bank\_id FOREIGN KEY (vendor\_bank\_id) REFERENCES vendor\_bank(vendor\_bank\_id);  
  
ALTER TABLE payments  
MODIFY COLUMN paid\_from INT DEFAULT NULL AFTER pay\_type\_id,  
MODIFY COLUMN paid\_to INT DEFAULT NULL AFTER paid\_from;  
  
ALTER TABLE payments  
MODIFY COLUMN amount varchar(30) Default Null;

**Summary:** Added new columns (pay\_type\_id, org\_bank\_id, user\_bank\_id, vendor\_bank\_id) to the payments table, establishing foreign key relationships to their respective tables. Also modified paid\_from, paid\_to, and amount columns.

## 22/04/2025

### Table: tickets (Alter Table)

ALTER TABLE tickets   
ADD COLUMN entity\_id INT DEFAULT NULL AFTER ticket\_id,  
ADD CONSTRAINT ticketss\_ibfk\_entity\_id FOREIGN KEY (entity\_id) REFERENCES entities(entity\_id);

**Summary:** Added an entity\_id column to tickets and created a foreign key to the entities table.

### Table: organization\_bank (Alter Table)

ALTER TABLE organization\_bank   
ADD COLUMN entity\_id INT DEFAULT NULL AFTER org\_id,  
ADD CONSTRAINT organization\_bank\_ibfk\_entity\_id FOREIGN KEY (entity\_id) REFERENCES entities(entity\_id);  
  
ALTER TABLE organization\_bank   
ADD COLUMN org\_name\_on\_bank INT DEFAULT NULL AFTER org\_bank\_account\_no;  
  
ALTER TABLE organization\_bank  
MODIFY COLUMN org\_name\_on\_bank varchar(100) DEFAULT NULL;

**Summary:** Added an entity\_id column to organization\_bank with a foreign key to entities. Also modified the org\_name\_on\_bank column's data type.

### Table: tickets (Data Update)

SET SQL\_SAFE\_UPDATES = 0;  
UPDATE tickets  
SET entity\_id = 1;  
SET SQL\_SAFE\_UPDATES = 1;  
  
SET SQL\_SAFE\_UPDATES = 0;  
UPDATE tickets  
SET entity\_id = NULL;  
SET SQL\_SAFE\_UPDATES = 1;

**Summary:** Updated the entity\_id column in the tickets table, first setting it to 1 and then to NULL. SQL\_SAFE\_UPDATES was temporarily disabled for these updates.

## 23/04/2025

### Table: employees (Alter Table)

ALTER TABLE employees  
CHANGE COLUMN current\_address current\_address\_line\_1 TEXT;  
  
ALTER TABLE employees  
ADD COLUMN current\_address\_line\_2 TEXT AFTER current\_address\_line\_1,  
ADD COLUMN current\_address\_city TEXT AFTER current\_address\_line\_2,  
ADD COLUMN current\_address\_state TEXT AFTER current\_address\_city,  
ADD COLUMN current\_address\_zip TEXT AFTER current\_address\_state,  
ADD COLUMN current\_address\_country TEXT AFTER current\_address\_zip;  
  
ALTER TABLE employees  
CHANGE COLUMN permanent\_address permanent\_address\_line\_1 TEXT;  
  
ALTER TABLE employees  
ADD COLUMN permanent\_address\_line\_2 TEXT AFTER permanent\_address\_line\_1,  
ADD COLUMN permanent\_address\_city TEXT AFTER permanent\_address\_line\_2,  
ADD COLUMN permanent\_address\_state TEXT AFTER permanent\_address\_city,  
ADD COLUMN permanent\_address\_zip TEXT AFTER permanent\_address\_state,  
ADD COLUMN permanent\_address\_country TEXT AFTER permanent\_address\_zip;

**Summary:** Refactored address columns in the employees table by renaming current\_address and permanent\_address to \_line\_1 and adding more granular address fields (\_line\_2, \_city, \_state, \_zip, \_country).

### Table: employee\_history (Alter Table)

ALTER TABLE employee\_history  
CHANGE COLUMN current\_address current\_address\_line\_1 TEXT;  
  
ALTER TABLE employee\_history  
ADD COLUMN current\_address\_line\_2 TEXT AFTER current\_address\_line\_1,  
ADD COLUMN current\_address\_city TEXT AFTER current\_address\_line\_2,  
ADD COLUMN current\_address\_state TEXT AFTER current\_address\_city,  
ADD COLUMN current\_address\_zip TEXT AFTER current\_address\_state,  
ADD COLUMN current\_address\_country TEXT AFTER current\_address\_zip;  
  
ALTER TABLE employee\_history  
CHANGE COLUMN permanent\_address permanent\_address\_line\_1 TEXT;  
  
ALTER TABLE employee\_history  
ADD COLUMN permanent\_address\_line\_2 TEXT AFTER permanent\_address\_line\_1,  
ADD COLUMN permanent\_address\_city TEXT AFTER permanent\_address\_line\_2,  
ADD COLUMN permanent\_address\_state TEXT AFTER permanent\_address\_city,  
ADD COLUMN permanent\_address\_zip TEXT AFTER permanent\_address\_state,  
ADD COLUMN permanent\_address\_country TEXT AFTER permanent\_address\_zip;

**Summary:** Similar to employees, address columns in employee\_history were refactored for more detailed address information.

### Table: role\_actions (New Table)

CREATE TABLE role\_actions (  
 role\_action\_id INT(11) NOT NULL AUTO\_INCREMENT,  
 role\_id INT(11) DEFAULT NULL,  
 action VARCHAR(100) DEFAULT NULL,  
 status\_id INT(11) DEFAULT NULL,  
 PRIMARY KEY (role\_action\_id),  
 KEY role\_id (role\_id),  
 KEY status\_id (status\_id),  
 CONSTRAINT role\_action\_ibfk\_1 FOREIGN KEY (role\_id) REFERENCES roles (role\_id),  
 CONSTRAINT role\_action\_ibfk\_2 FOREIGN KEY (status\_id) REFERENCES master\_status (status\_id)  
 );

**Summary:** Created a new table role\_actions to manage actions associated with specific roles, with foreign keys to roles and master\_status.

## 25/04/2025

### Table: login\_details (Alter Table)

ALTER TABLE login\_details   
ADD COLUMN login\_type VARCHAR(30) AFTER user\_id;

**Summary:** Added a login\_type column to login\_details to specify the type of login.

### Table: payments (Alter Table)

ALTER TABLE payments CHANGE transaction\_id UTR\_number VARCHAR(100);  
  
ALTER TABLE payments   
ADD COLUMN payment\_number VARCHAR(45) AFTER ticket\_id;  
  
ALTER TABLE payments   
ADD COLUMN created\_by int(11) AFTER payment\_date,  
ADD CONSTRAINT payments\_ibfk\_created\_by FOREIGN KEY (created\_by) REFERENCES users(user\_id);  
  
ALTER TABLE payments CHANGE payment\_number transaction\_id VARCHAR(45);

**Summary:** Renamed transaction\_id to UTR\_number, added a payment\_number column, then renamed payment\_number back to transaction\_id. Also added created\_by column with a foreign key to users.

## 26/04/2025

### Table: payments (Alter Table)

ALTER TABLE payments DROP FOREIGN KEY payments\_ibfk\_org\_bank\_id;  
ALTER TABLE payments DROP FOREIGN KEY payments\_ibfk\_user\_bank\_id;  
ALTER TABLE payments DROP FOREIGN KEY payments\_ibfk\_vendor\_bank\_id;  
  
ALTER TABLE payments  
DROP COLUMN org\_bank\_id,  
DROP COLUMN user\_bank\_id,  
DROP COLUMN vendor\_bank\_id;  
  
ALTER TABLE payments  
ADD COLUMN paid\_from\_bank INT(11) AFTER paid\_to,  
ADD COLUMN paid\_to\_bank INT(11) AFTER paid\_from\_bank;

**Summary:** Dropped foreign keys and columns related to org\_bank\_id, user\_bank\_id, and vendor\_bank\_id from the payments table, and added paid\_from\_bank and paid\_to\_bank columns.

### Table: organization\_bank (Alter Table)

ALTER TABLE organization\_bank  
CHANGE org\_bank\_name entity\_bank\_name VARCHAR(100),  
CHANGE org\_bank\_account\_no entity\_bank\_account\_no VARCHAR(50),  
CHANGE org\_name\_on\_bank entity\_name\_on\_bank VARCHAR(100),  
CHANGE org\_bank\_ifsc entity\_bank\_ifsc VARCHAR(50),  
CHANGE org\_bank\_format entity\_bank\_format TEXT;

**Summary:** Renamed several org\_bank\_ prefixed columns to entity\_bank\_ prefixed columns in organization\_bank for consistency.

### Table: payments (Data Update)

SET SQL\_SAFE\_UPDATES = 0;  
  
UPDATE payments  
SET paid\_from\_bank = 1,  
 paid\_to\_bank = 1;  
  
SET SQL\_SAFE\_UPDATES = 1;

**Summary:** Updated paid\_from\_bank and paid\_to\_bank columns in the payments table to '1'. SQL\_SAFE\_UPDATES was temporarily disabled.

## 02/05/2025

### Table: re\_ticket\_details (Alter Table)

ALTER TABLE re\_ticket\_details  
CHANGE granted\_amount m\_granted\_amount VARCHAR(100);  
  
ALTER TABLE re\_ticket\_details  
ADD f\_granted\_amount VARCHAR(100) AFTER m\_granted\_amount;  
  
ALTER TABLE re\_ticket\_details  
CHANGE granted\_by f\_granted\_by INT(11);

**Summary:** Renamed granted\_amount to m\_granted\_amount and added f\_granted\_amount in re\_ticket\_details. Also, granted\_by was renamed to f\_granted\_by.

### Table: reimbursement\_history (Alter Table)

ALTER TABLE reimbursement\_history  
CHANGE granted\_amount m\_granted\_amount VARCHAR(100);  
  
ALTER TABLE reimbursement\_history  
ADD f\_granted\_amount VARCHAR(100) AFTER m\_granted\_amount;

**Summary:** Renamed granted\_amount to m\_granted\_amount and added f\_granted\_amount in reimbursement\_history.

## 03/05/2025

### Table: reports (Constraint Add)

ALTER TABLE df\_ticketing.reports  
ADD CONSTRAINT unique\_report\_code UNIQUE (report\_code);

**Summary:** Added a unique constraint on the report\_code column in the reports table to ensure no duplicate report codes.

### Table: tickets (Constraint Add)

ALTER TABLE df\_ticketing.tickets  
ADD CONSTRAINT unique\_ticket\_number UNIQUE (ticket\_number);

**Summary:** Added a unique constraint on the ticket\_number column in the tickets table to ensure no duplicate ticket numbers.

### Table: payments (Constraint Add)

ALTER TABLE df\_ticketing.payments  
ADD CONSTRAINT unique\_transaction\_id UNIQUE (transaction\_id);

**Summary:** Added a unique constraint on the transaction\_id column in the payments table to ensure no duplicate transaction IDs.

## 08/05/2025

### Tables: employee\_history, employees, user\_history, users (Alter Table)

ALTER TABLE employee\_history MODIFY COLUMN secondary\_job\_title VARCHAR(100);  
ALTER TABLE employees MODIFY COLUMN secondary\_job\_title VARCHAR(100);  
ALTER TABLE user\_history MODIFY COLUMN secondary\_job\_title VARCHAR(100);  
ALTER TABLE users MODIFY COLUMN secondary\_job\_title VARCHAR(100);  
  
ALTER TABLE employee\_history MODIFY COLUMN job\_title VARCHAR(100);  
ALTER TABLE employees MODIFY COLUMN job\_title VARCHAR(100);  
ALTER TABLE user\_history MODIFY COLUMN job\_title VARCHAR(100);  
ALTER TABLE users MODIFY COLUMN job\_title VARCHAR(100);

**Summary:** Modified the secondary\_job\_title and job\_title columns in multiple tables (employee\_history, employees, user\_history, users) to increase their VARCHAR length to 100.

### Table: organization\_bank (Alter Table)

ALTER TABLE organization\_bank   
ADD COLUMN email VARCHAR(100)   
after entity\_bank\_IFSC;

**Summary:** Added an email column to organization\_bank after entity\_bank\_IFSC.

## 09/05/2025

### Table: organization\_bank (Alter Table)

ALTER TABLE organization\_bank  
ADD COLUMN account\_type VARCHAR(50) AFTER entity\_id,  
ADD COLUMN bank\_address TEXT AFTER entity\_bank\_IFSC,  
ADD COLUMN bank\_contact\_no VARCHAR(30) AFTER bank\_address,  
ADD COLUMN bank\_contact\_person VARCHAR(100) AFTER bank\_contact\_no;  
  
ALTER TABLE organization\_bank AUTO\_INCREMENT = 1;

**Summary:** Added account\_type, bank\_address, bank\_contact\_no, and bank\_contact\_person columns to organization\_bank and reset its auto-increment value.

### Tables: employee\_history, employees, user\_history, users (Alter Table)

ALTER TABLE employee\_history   
MODIFY COLUMN employee\_number VARCHAR(50);  
  
ALTER TABLE employees   
MODIFY COLUMN employee\_number VARCHAR(50);  
  
ALTER TABLE user\_history   
MODIFY COLUMN employee\_number VARCHAR(50);  
  
ALTER TABLE users   
MODIFY COLUMN employee\_number VARCHAR(50);

**Summary:** Modified the employee\_number column in multiple tables (employee\_history, employees, user\_history, users) to increase its VARCHAR length to 50.

### Table: bill\_history (Table Drop and Recreate - Commented Out)

DROP TABLE IF EXISTS `bill\_history`;  
/\* /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\* /\*!50503 SET character\_set\_client = utf8mb4 \*/;  
/\* CREATE TABLE bill\_history (  
 `bill\_hst\_id` INT(11) NOT NULL AUTO\_INCREMENT,  
 `bill\_id` INT(11) DEFAULT NULL,  
 `ticket\_id` INT(11) DEFAULT NULL,  
 `exp\_catg\_id` INT(11) DEFAULT NULL,  
 `ticket\_dtls\_id` INT(11) DEFAULT NULL,  
 `bill\_path` TEXT DEFAULT NULL,  
 `bill\_number` VARCHAR(45) DEFAULT NULL,  
 `bill\_amount` VARCHAR(100) DEFAULT NULL,  
 `bill\_date` DATETIME DEFAULT NULL,  
 `status\_id` INT(11) DEFAULT NULL,  
 `created\_at` DATETIME DEFAULT NULL,  
 `created\_by` INT(11) DEFAULT NULL,  
 `updated\_at` DATETIME DEFAULT NULL,  
 `updated\_by` INT(11) DEFAULT NULL,  
 PRIMARY KEY (`bill\_hst\_id`),  
 KEY `bill\_id` (`bill\_id`),  
 KEY `ticket\_id` (`ticket\_id`),  
 KEY `exp\_catg\_id` (`exp\_catg\_id`),  
 KEY `status\_id` (`status\_id`),  
 CONSTRAINT `bill\_hst\_ibfk\_1` FOREIGN KEY (`bill\_id`) REFERENCES `bills`(`bill\_id`),  
 CONSTRAINT `bill\_hst\_ibfk\_2` FOREIGN KEY (`ticket\_id`) REFERENCES `tickets`(`ticket\_id`) ,  
 CONSTRAINT `bill\_hst\_ibfk\_3` FOREIGN KEY (`exp\_catg\_id`) REFERENCES `expense\_category`(`expense\_category\_id`) ,  
 CONSTRAINT `bill\_hst\_ibfk\_4` FOREIGN KEY (`status\_id`) REFERENCES `master\_status`(`status\_id`)   
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
 /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
/\* LOCK TABLES `bill\_history` WRITE;  
 /\*!40000 ALTER TABLE `bill\_history` DISABLE KEYS \*/;  
/\* /\*!40000 ALTER TABLE `bill\_history` ENABLE KEYS \*/;  
/\* UNLOCK TABLES;  
  
DROP TABLE `df\_ticketing`.`bill\_history`

**Summary:** The bill\_history table was dropped and then a CREATE TABLE statement for it was included, though it appears to be commented out or intended for a re-creation. Finally, another DROP TABLE statement for bill\_history is present. This indicates a cycle of dropping and potentially redefining the bill\_history table, but the final state based on this script is that it's dropped.

## 12/05/2025

### Tables: employee\_history, employees (Alter Table)

ALTER TABLE employee\_history   
MODIFY COLUMN reporting\_manager\_en VARCHAR(50);  
  
ALTER TABLE employees   
MODIFY COLUMN reporting\_manager\_en VARCHAR(50);

**Summary:** Modified the reporting\_manager\_en column in employee\_history and employees to increase its VARCHAR length to 50.

### Table: edit\_history (Table Drop and Recreate - Commented Out)

DROP TABLE IF EXISTS `edit\_history`;  
/\* /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\* /\*!50503 SET character\_set\_client = utf8mb4 \*/;  
/\* CREATE TABLE `edit\_history` (  
 `edit\_id` int(11) NOT NULL AUTO\_INCREMENT,  
 `report\_id` int(11) DEFAULT NULL,  
 `ticket\_id` int(11) DEFAULT NULL,  
 `updated\_by` int(11) DEFAULT NULL,  
 `table\_name` varchar(50) DEFAULT NULL,  
 `column\_name` varchar(60) DEFAULT NULL,  
 `before` varchar(50) DEFAULT NULL,  
 `after` varchar(50) DEFAULT NULL,  
 `updated\_date` datetime DEFAULT NULL,  
 `description` text,  
 PRIMARY KEY (`edit\_id`),  
 KEY `report\_id` (`report\_id`),  
 KEY `ticket\_id` (`ticket\_id`),  
 CONSTRAINT `edit\_history\_ibfk\_1` FOREIGN KEY (`report\_id`) REFERENCES `reports` (`report\_id`),  
 CONSTRAINT `edit\_history\_ibfk\_2` FOREIGN KEY (`ticket\_id`) REFERENCES `tickets` (`ticket\_id`)  
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
 /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
/\* LOCK TABLES `edit\_history` WRITE;  
 /\*!40000 ALTER TABLE `edit\_history` DISABLE KEYS \*/;  
/\* /\*!40000 ALTER TABLE `edit\_history` ENABLE KEYS \*/;  
/\* UNLOCK TABLES;  
  
DROP TABLE `df\_ticketing`.`edit\_history`

**Summary:** The edit\_history table was dropped and then a CREATE TABLE statement for it was included, though it appears to be commented out or intended for a re-creation. Finally, another DROP TABLE statement for edit\_history is present. This indicates a cycle of dropping and potentially redefining the edit\_history table, but the final state based on this script is that it's dropped.

### Tables: employee\_history, employees, users, user\_history (Alter Table)

ALTER TABLE employee\_history   
MODIFY COLUMN personal\_email VARCHAR(100);   
ALTER TABLE employee\_history   
MODIFY COLUMN work\_email VARCHAR(100);  
  
ALTER TABLE employees   
MODIFY COLUMN personal\_email VARCHAR(100);  
ALTER TABLE employees   
MODIFY COLUMN work\_email VARCHAR(100);  
  
ALTER TABLE users   
MODIFY COLUMN work\_email VARCHAR(100);  
  
ALTER TABLE users   
MODIFY COLUMN personal\_email VARCHAR(100);  
  
ALTER TABLE user\_history   
MODIFY COLUMN work\_email VARCHAR(100);  
  
ALTER TABLE user\_history   
MODIFY COLUMN personal\_email VARCHAR(100);  
  
ALTER TABLE employee\_history   
MODIFY COLUMN display\_name VARCHAR(100);   
  
ALTER TABLE employees   
MODIFY COLUMN display\_name VARCHAR(100);   
  
ALTER TABLE users   
MODIFY COLUMN user\_name VARCHAR(100);   
  
ALTER TABLE user\_history   
MODIFY COLUMN user\_name VARCHAR(100);   
  
ALTER TABLE user\_bank   
MODIFY COLUMN name\_on\_bank VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN first\_name VARCHAR(100);   
ALTER TABLE employee\_history   
MODIFY COLUMN middle\_name VARCHAR(100);   
ALTER TABLE employee\_history   
MODIFY COLUMN last\_name VARCHAR(100);   
ALTER TABLE employee\_history   
MODIFY COLUMN full\_name VARCHAR(100);   
  
ALTER TABLE employees   
MODIFY COLUMN first\_name VARCHAR(100);   
ALTER TABLE employees   
MODIFY COLUMN middle\_name VARCHAR(100);   
ALTER TABLE employees   
MODIFY COLUMN last\_name VARCHAR(100);   
ALTER TABLE employees   
MODIFY COLUMN full\_name VARCHAR(100);   
  
ALTER TABLE employees   
MODIFY COLUMN entity VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN entity VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN employee\_name\_on\_bank VARCHAR(100);   
  
ALTER TABLE employees   
MODIFY COLUMN employee\_name\_on\_bank VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN employees\_name VARCHAR(100);   
  
ALTER TABLE employees   
MODIFY COLUMN employees\_name VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN pay\_group VARCHAR(100);   
  
ALTER TABLE employees   
MODIFY COLUMN pay\_group VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN employee\_IFSC VARCHAR(50);   
  
ALTER TABLE employees   
MODIFY COLUMN employee\_IFSC VARCHAR(50);   
  
ALTER TABLE user\_bank   
MODIFY COLUMN IFSC VARCHAR(50);   
  
ALTER TABLE vendor\_bank   
MODIFY COLUMN IFSC VARCHAR(50);   
  
ALTER TABLE employees   
MODIFY COLUMN reporting\_manager VARCHAR(100);   
  
ALTER TABLE employee\_history   
MODIFY COLUMN reporting\_manager VARCHAR(100);   
  
ALTER TABLE vendor\_bank   
MODIFY COLUMN name\_on\_bank VARCHAR(100);

**Summary:** This extensive set of modifications increases the VARCHAR length for numerous columns across multiple tables (employee\_history, employees, users, user\_history, user\_bank, vendor\_bank) to 100 or 50, primarily for name, email, and bank-related fields.

## 15/05/2025

### Tables: employees, employee\_history (Alter Table)

ALTER TABLE employees  
CHANGE employee\_Account\_number employee\_account\_number VARCHAR(20);  
  
ALTER TABLE employee\_history  
CHANGE employee\_Account\_number employee\_account\_number VARCHAR(20);

**Summary:** Renamed employee\_Account\_number to employee\_account\_number and changed its VARCHAR length to 20 in both employees and employee\_history tables.

## 23/05/2025

### Table: tickets (Alter Table)

ALTER TABLE tickets  
ADD description text AFTER process\_status\_id;

**Summary:** Added a description column (TEXT type) to the tickets table after process\_status\_id.

### Table: ticket\_history (Alter Table)

ALTER TABLE ticket\_history  
ADD description text AFTER contribution\_id;

**Summary:** Added a description column (TEXT type) to the ticket\_history table after contribution\_id.

### Table: travels (Alter Table)

ALTER TABLE travels  
MODIFY COLUMN `from` VARCHAR(100) AFTER vehicle\_id,  
MODIFY COLUMN `to` VARCHAR(100) AFTER `from`;

**Summary:** Modified the from and to columns in travels to VARCHAR(100) and reordered them.

### Table: tickets (Alter Table)

ALTER TABLE tickets  
ADD COLUMN granted\_amount VARCHAR(100) AFTER exp\_catg\_id;

**Summary:** Added a granted\_amount column (VARCHAR(100)) to the tickets table after exp\_catg\_id.

### Table: organization\_bank (Alter Table)

ALTER TABLE organization\_bank  
ADD COLUMN client\_code VARCHAR(100) AFTER account\_type;

**Summary:** Added a client\_code column (VARCHAR(100)) to the organization\_bank table after account\_type.

## 31/05/2025

### Table: organization\_bank (Alter Table)

ALTER TABLE organization\_bank  
ADD bank\_ledger text AFTER entity\_bank\_IFSC;

**Summary:** Added a bank\_ledger column (TEXT type) to the organization\_bank table after entity\_bank\_IFSC.

### Tables: employees, employee\_history (Alter Table)

ALTER TABLE employees  
MODIFY cost\_center VARCHAR(225);  
  
ALTER TABLE employee\_history  
MODIFY cost\_center VARCHAR(225);

**Summary:** Modified the cost\_center column in employees and employee\_history to increase its VARCHAR length to 225.

### Table: users (Alter Table)

ALTER TABLE `df\_ticketing`.`users`  
ADD COLUMN `sended\_email` INT(11) AFTER `work\_location`;

**Summary:** Added a sended\_email column (INT(11)) to the users table after work\_location.

## 05/06/2025

### Table: ledgers (Index/Constraint Drop, Data Update, Column Add, Constraint Add)

ALTER TABLE ledgers  
DROP FOREIGN KEY ledgers\_ibfk\_2;  
  
ALTER TABLE `ledgers` DROP INDEX `entity\_id`;  
  
SET SQL\_SAFE\_UPDATES = 0;  
  
UPDATE ledgers  
SET entity\_id = NULL;  
  
SET SQL\_SAFE\_UPDATES = 1;  
  
SET SQL\_SAFE\_UPDATES = 0;  
  
UPDATE ledgers  
SET status\_id = 2;  
  
SET SQL\_SAFE\_UPDATES = 1;

**Summary:** Dropped the foreign key and index related to entity\_id in the ledgers table. Updated entity\_id to NULL and status\_id to 2, temporarily disabling SQL\_SAFE\_UPDATES.

### Table: tally\_booking (Alter Table)

alter table tally\_booking ADD COLUMN `dr/cr` varchar(10) AFTER `payment\_id`;  
  
ALTER TABLE `tally\_booking`  
 ADD COLUMN `tally\_pay\_id` INT(11) AFTER `payment\_id`,  
 ADD CONSTRAINT `tally\_booking\_ibfk\_5` FOREIGN KEY (`tally\_pay\_id`) REFERENCES `tally\_payment` (`tally\_pay\_id`);

**Summary:** Added a dr/cr column (VARCHAR(10)) and tally\_pay\_id column (INT(11)) to tally\_booking, with tally\_pay\_id having a foreign key to tally\_payment.

## 11/06/2025

### Table: tally\_pay\_bank (Alter Table)

ALTER TABLE tally\_pay\_bank  
ADD COLUMN route\_id INT AFTER tally\_pay\_id,  
ADD COLUMN bank\_id INT AFTER route\_id,  
ADD CONSTRAINT fk\_tally\_pay\_bank\_route\_id  
 FOREIGN KEY (route\_id) REFERENCES payment\_route(route\_id);

**Summary:** Added route\_id and bank\_id columns to tally\_pay\_bank, with route\_id having a foreign key to payment\_route.