

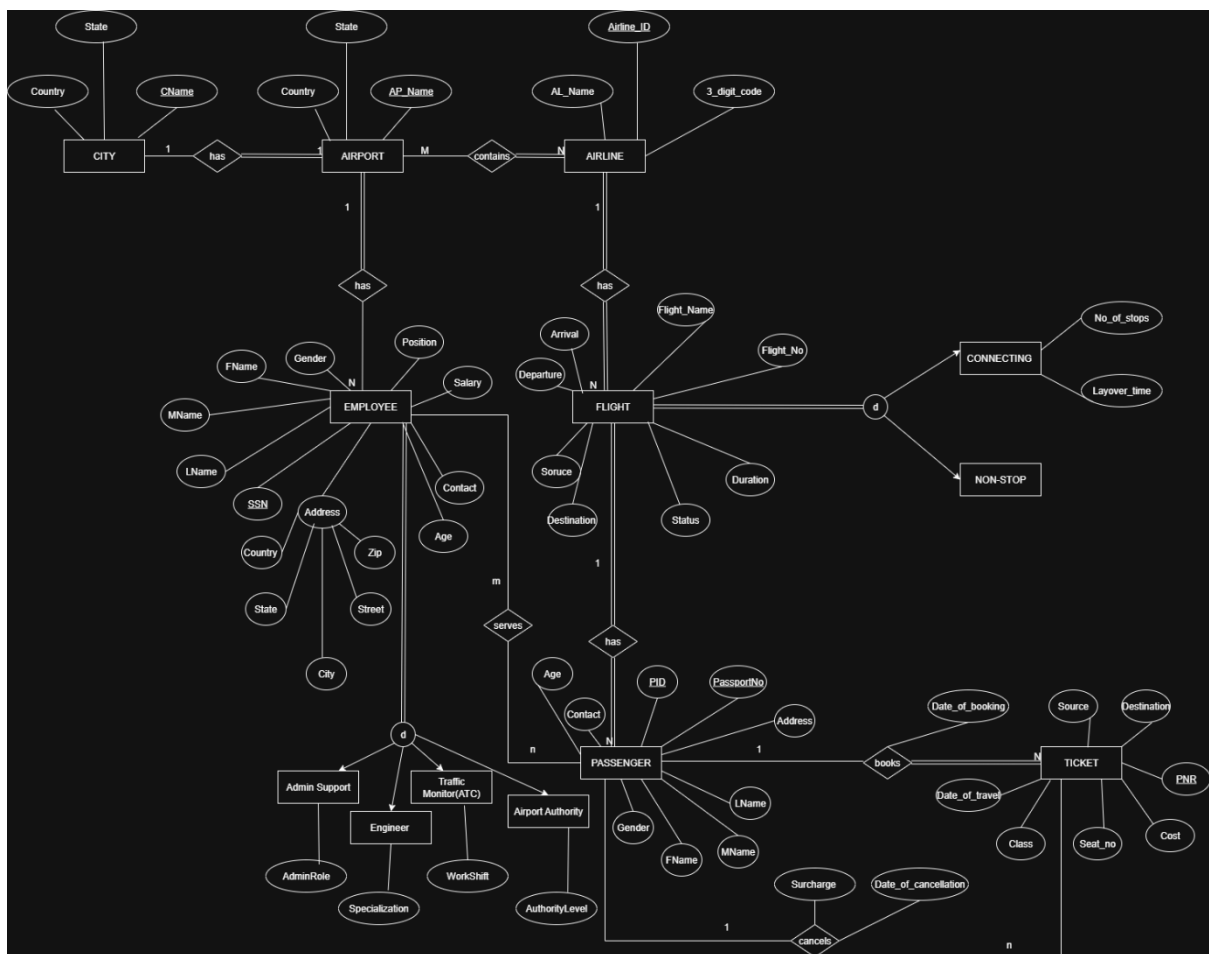
DBMS MINI PROJECT

AIRPORT MANAGEMENT SYSTEM

TEAM MEMBERS:

- Sanjana Pulla (PES2UG23CS529)
- Sharon A (PES2UG23CS544)

ER DIAGRAM:



RELATIONAL SCHEMA:



DDL, DML AND TRIGGERS, PROCEDURES AND FUNCTIONS:

```
CREATE DATABASE IF NOT EXISTS airport_demo;  
USE airport_demo;
```

```
CREATE TABLE IF NOT EXISTS city (  
    city_id INT AUTO_INCREMENT PRIMARY KEY,  
    city_name VARCHAR(100) NOT NULL UNIQUE,  
    state VARCHAR(100),  
    country VARCHAR(100)  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS airport (  
    airport_id INT AUTO_INCREMENT PRIMARY KEY,  
    airport_name VARCHAR(100) NOT NULL,  
    city_id INT,  
    airport_code VARCHAR(10) UNIQUE,  
    CONSTRAINT fk_airport_city FOREIGN KEY (city_id)  
        REFERENCES city(city_id)  
        ON DELETE SET NULL ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS airline (  
    airline_id INT AUTO_INCREMENT PRIMARY KEY,  
    airline_name VARCHAR(100),  
    iata_code CHAR(2),  
    icao_code CHAR(3)  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS airport_airline (  

```

```
airline_id INT,  
airport_id INT,  
operating_since DATE,  
PRIMARY KEY (airline_id, airport_id),  
FOREIGN KEY (airline_id) REFERENCES airline(airline_id)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (airport_id) REFERENCES airport(airport_id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS flight (  
    flight_id VARCHAR(20) PRIMARY KEY,  
    source_id INT,  
    destination_id INT,  
    airline_id INT,  
    flight_date DATE,  
    status VARCHAR(50),  
    d_time TIME,  
    a_time TIME,  
    departure_time DATETIME,  
    arrival_time DATETIME,  
    tot_seat INT DEFAULT 180,  
    base_price DECIMAL(10,2),  
    price DECIMAL(10,2),  
    flight_type VARCHAR(30),  
    layover_time INT DEFAULT 0,  
    no_of_stops TINYINT DEFAULT 0,  
    FOREIGN KEY (airline_id) REFERENCES airline(airline_id),
```

```
FOREIGN KEY (source_id) REFERENCES airport(airport_id),  
FOREIGN KEY (destination_id) REFERENCES airport(airport_id)  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS passenger (  
    passport_no VARCHAR(30) PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(150),  
    phone VARCHAR(15),  
    address VARCHAR(255),  
    gender CHAR(1),  
    dob DATE  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS bookings (  
    booking_id INT AUTO_INCREMENT PRIMARY KEY,  
    flight_id VARCHAR(20),  
    passport_no VARCHAR(30),  
    booking_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
    status ENUM('booked','cancelled','refunded') DEFAULT 'booked',  
    seat_no VARCHAR(10),  
    class VARCHAR(20),  
    FOREIGN KEY (flight_id) REFERENCES flight(flight_id),  
    FOREIGN KEY (passport_no) REFERENCES passenger(passport_no)  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS ticket (  
    ticket_no VARCHAR(20) PRIMARY KEY,
```

```
flight_id VARCHAR(20),  
passport_no VARCHAR(30),  
price DECIMAL(10,2),  
seat_no VARCHAR(10),  
class VARCHAR(20),  
FOREIGN KEY (flight_id) REFERENCES flight(flight_id),  
FOREIGN KEY (passport_no) REFERENCES passenger(passport_no)  
) ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS payment (  
payment_id VARCHAR(50) PRIMARY KEY,  
ticket_no VARCHAR(20),  
payment_mode VARCHAR(30),  
payment_date DATE,  
amount DECIMAL(12,2),  
status VARCHAR(30),  
CONSTRAINT fk_payment_ticket FOREIGN KEY (ticket_no)  
REFERENCES ticket(ticket_no)  
ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
INSERT IGNORE INTO city (city_name, state, country) VALUES  
( 'Mumbai', 'Maharashtra', 'India'),  
( 'Dubai', 'Dubai Emirate', 'UAE'),  
( 'Delhi', 'Delhi', 'India'),  
( 'Singapore', 'Central Region', 'Singapore'),  
( 'London', 'England', 'United Kingdom');
```

```
INSERT IGNORE INTO airport (airport_name, city_id, airport_code) VALUES
('Chhatrapati Shivaji Maharaj International Airport', 1, 'BOM'),
('Dubai International Airport', 2, 'DXB'),
('Indira Gandhi International Airport', 3, 'DEL'),
('Singapore Changi Airport', 4, 'SIN'),
('Heathrow Airport', 5, 'LHR');
```

```
INSERT IGNORE INTO airline (airline_name, iata_code, icao_code) VALUES
('Emirates', 'EK', 'UAE'),
('Air India', 'AI', 'AIC'),
('Singapore Airlines', 'SQ', 'SIA'),
('British Airways', 'BA', 'BAW'),
('Vistara', 'UK', 'VTI');
```

```
INSERT IGNORE INTO airport_airline (airline_id, airport_id, operating_since) VALUES
(1, 1, '2010-04-05'),
(1, 2, '2005-09-10'),
(2, 3, '2000-01-01'),
(3, 4, '2013-05-18'),
(4, 5, '2008-11-22');
```

```
INSERT IGNORE INTO flight (
    flight_id, source_id, destination_id, airline_id, flight_date,
    status, d_time, a_time, departure_time, arrival_time, base_price, price, flight_type,
    layover_time, no_of_stops
) VALUES
('EK-507', 1, 2, 1, '2025-10-25', 'On Time', '21:30:00', '23:59:00', '2025-10-25 21:30:00',
'2025-10-25 23:59:00', 15000, 16200, 'Domestic', 0, 0),
```

```
('AI-102', 3, 5, 2, '2025-10-26', 'Delayed', '10:00:00', '16:00:00', '2025-10-26 10:00:00',  
'2025-10-26 16:00:00', 18000, 19000, 'Domestic', 0, 0),  
  
('SQ-423', 4, 1, 3, '2025-10-27', 'On Time', '07:15:00', '09:45:00', '2025-10-27 07:15:00',  
'2025-10-27 09:45:00', 12000, 12500, 'Domestic', 0, 0),  
  
('BA-118', 5, 3, 4, '2025-10-28', 'On Time', '13:20:00', '23:00:00', '2025-10-28 13:20:00',  
'2025-10-28 23:00:00', 25000, 26500, 'Domestic', 0, 0),  
  
('UK-220', 1, 3, 5, '2025-10-29', 'Cancelled', '15:00:00', '17:25:00', '2025-10-29 15:00:00',  
'2025-10-29 17:25:00', 9500, 9800, 'Domestic', 0, 0);
```

```
INSERT IGNORE INTO passenger (passport_no, name, email, phone, address, gender,  
dob) VALUES
```

```
('P990123', 'Aaradhya Nair', 'aaradhya.nair@example.com', '9876501234', 'Mumbai', 'F',  
'2001-08-19'),
```

```
('P775612', 'Kabir Khanna', 'kabir.khanna@example.com', '9856723456', 'Delhi', 'M',  
'1998-11-04'),
```

```
('P553400', 'Meera Reddy', 'meera.reddy@example.com', '9812334567', 'Hyderabad', 'F',  
'1995-03-12'),
```

```
('P220908', 'Liam Carter', 'liam.carter@example.co.uk', '447821334500', 'London', 'M',  
'1992-09-10'),
```

```
('P110777', 'Sophia Lim', 'sophia.lim@example.sg', '6589123456', 'Singapore', 'F', '1999-  
02-28');
```

```
INSERT IGNORE INTO bookings (flight_id, passport_no, booking_date, status, seat_no,  
class) VALUES
```

```
('EK-507', 'P990123', '2025-10-22 10:45:00', 'booked', '14A', 'Business'),
```

```
('AI-102', 'P775612', '2025-10-22 11:00:00', 'booked', '19C', 'Economy'),
```

```
('SQ-423', 'P553400', '2025-10-22 12:30:00', 'booked', '06B', 'Premium Economy'),
```

```
('BA-118', 'P220908', '2025-10-22 14:20:00', 'cancelled', '02A', 'First Class'),
```

```
('UK-220', 'P110777', '2025-10-22 15:10:00', 'refunded', '21C', 'Economy');
```

```
DROP TRIGGER IF EXISTS trg_ticket_after_insert;
```



```
DELIMITER $$

CREATE TRIGGER trg_ticket_after_insert
AFTER INSERT ON ticket
FOR EACH ROW
BEGIN
    DECLARE v_exists INT DEFAULT 0;

    SELECT COUNT(*) INTO v_exists FROM payment WHERE ticket_no = NEW.ticket_no;

    IF v_exists = 0 THEN

        INSERT INTO payment (payment_id, ticket_no, payment_mode, payment_date,
amount, status)

        VALUES (

            CONCAT('PAY', UNIX_TIMESTAMP(), FLOOR(RAND()*1000)),

            NEW.ticket_no,

            'AUTO',

            CURDATE(),

            NEW.price,

            'completed'

        );

    END IF;

END$$

DELIMITER ;
```

```
INSERT IGNORE INTO ticket (ticket_no, flight_id, passport_no, price, seat_no, class)
VALUES

('T2001', 'EK-507', 'P990123', 16200, '14A', 'Business'),

('T2002', 'AI-102', 'P775612', 19000, '19C', 'Economy'),

('T2003', 'SQ-423', 'P553400', 12500, '06B', 'Premium Economy'),

('T2004', 'BA-118', 'P220908', 26500, '02A', 'First Class'),

('T2005', 'UK-220', 'P110777', 9800, '21C', 'Economy');
```

```
DROP PROCEDURE IF EXISTS sp_update_flight_schedule;

DELIMITER $$

CREATE PROCEDURE sp_update_flight_schedule(
    IN p_flight_id VARCHAR(20),
    IN p_flight_date DATE,
    IN p_departure_time TIME,
    IN p_arrival_time TIME
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'sp_update_flight_schedule failed:
transaction rolled back due to an internal error.';
    END;

    START TRANSACTION;

    IF NOT EXISTS (SELECT 1 FROM flight WHERE flight_id = p_flight_id) THEN
        ROLLBACK;

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Flight not found.';
    END IF;

    IF p_flight_date < CURDATE() THEN
        ROLLBACK;

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot set flight_date to a past
date.';
    END IF;
```

```
UPDATE flight
SET flight_date = p_flight_date,
    departure_time = p_departure_time,
    arrival_time = p_arrival_time
WHERE flight_id = p_flight_id;
```

```
COMMIT;
```

```
SELECT CONCAT('Updated schedule for ', p_flight_id, ' -> ',
DATE_FORMAT(p_flight_date, '%Y-%m-%d')) AS info;
END$$
DELIMITER ;
```

```
DROP PROCEDURE IF EXISTS sp_recalculate_price;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE sp_recalculate_price(
    IN p_flight_id VARCHAR(20)
)
```

```
BEGIN
```

```
    DECLARE v_base DECIMAL(10,2);
```

```
    DECLARE v_stops INT;
```

```
    DECLARE v_type VARCHAR(30);
```

```
    DECLARE v_new_price DECIMAL(12,2);
```

```
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
BEGIN
```

```
    ROLLBACK;
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'sp_recalculate_price failed:  
transaction rolled back due to an internal error.';
```

```
END;
```

```
START TRANSACTION;
```

```
IF NOT EXISTS (SELECT 1 FROM flight WHERE flight_id = p_flight_id) THEN
```

```
    ROLLBACK;
```

```
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Flight not found.';
```

```
END IF;
```

```
SELECT base_price, no_of_stops, flight_type
```

```
INTO v_base, v_stops, v_type
```

```
FROM flight
```

```
WHERE flight_id = p_flight_id
```

```
LIMIT 1;
```

```
IF v_base IS NULL THEN SET v_base = 0.00; END IF;
```

```
IF v_stops IS NULL THEN SET v_stops = 0; END IF;
```

```
IF v_type IS NULL THEN SET v_type = 'Domestic'; END IF;
```

```
SET v_new_price = v_base + (v_stops * 500);
```

```
IF UPPER(v_type) = 'INTERNATIONAL' THEN
```

```
    SET v_new_price = v_new_price + (v_base * 0.20);
```

```
END IF;
```

```
IF v_new_price < v_base THEN SET v_new_price = v_base; END IF;
```

UPDATE flight

SET price = ROUND(v_new_price, 2)

WHERE flight_id = p_flight_id;

COMMIT;

SELECT ROUND(v_new_price,2) AS new_price;

END\$\$

DELIMITER ;

DROP FUNCTION IF EXISTS fn_estimated_revenue;

DELIMITER \$\$

CREATE FUNCTION fn_estimated_revenue(p_flight_id VARCHAR(20))

RETURNS DECIMAL(14,2)

DETERMINISTIC

READS SQL DATA

BEGIN

DECLARE v_total DECIMAL(14,2) DEFAULT 0.00;

SELECT COALESCE(SUM(t.price), 0.00) INTO v_total

FROM ticket t

LEFT JOIN bookings b

ON b.flight_id = t.flight_id

AND b.passport_no = t.passport_no

WHERE t.flight_id = p_flight_id

AND (b.status IS NULL OR b.status <> 'cancelled');

```
RETURN v_total;
```

```
END$$
```

```
DELIMITER ;
```

```
SELECT 'schema ready' AS status;
```

```
SELECT airline_id, airline_name, iata_code, icao_code FROM airline;
```

```
SELECT flight_id, flight_date, departure_time, arrival_time, base_price, price, flight_type  
FROM flight LIMIT 10;
```