TECHNICAL REPORT

AIRLINE DATABASE MANAGEMENT SYSTEM

**BUAN 6320**

*by Group 4*

**JSOM - UTD**

*Anuj Gupta [AXG230072]*

*Keerthana Munnangi [KXM220017]*

*Naga Harshath Medisetti [NXM220059]*

*Shreevershith Kollabettu [SXK220508]*

**INTRODUCTION AND ASSUMPTIONS:**

The Airline Database Management System's architecture is meticulously crafted to handle and organize information within the aviation sector. It incorporates airports, cities, passengers, alliances, airlines, airplanes, and airport-airline relationships, each with specific attributes to capture relevant details. The system's design is structured to streamline data storage, retrieval, and maintenance associated with various facets of air travel.

The interconnections between these entities are thoughtfully defined, establishing meaningful links for the smooth flow of data interactions. The incorporation of primary and foreign keys enhances data integrity, facilitating accurate associations across different elements of the aviation system.

This comprehensive database design thoroughly addresses the nuanced requirements of the aviation industry and offers a robust framework for managing airport infrastructure, airline operations, passenger data, and alliance dynamics. The inclusion of triggers, sequences, and views adds depth to the system, ensuring a well-structured and effective solution for handling diverse data associated with aviation operations.

**BUSINESS RULES:**

**City - Airport Relationship:**

*Assumption:* Each city can have only one airport at the most.

*Cardinality:* Airport (1): City (1) | *Mandatory:* Airport | *Optional:* City

*Rule:* Each airport is exclusively associated with one city, creating a one-to-one relationship. Conversely, as per the assumption, a city can accommodate only one airport, making it a one-to-one relationship.

**City - Air Alliance Relationship:**

*Assumption:* None.

*Cardinality:* City (1): Air Alliance (M) | *Mandatory:* Air Alliance | *Optional:* City

*Rule:* Each air alliance designates one city as its headquarters, forming a one-to-one relationship. Meanwhile, a city can serve as the base headquarters for multiple air alliances, establishing a one-to-many relationship.

**Airport - Passenger Relationship:**

*Assumption:*

1. An airport will cease to operate without passengers.
2. Each layover creates a new trip. Hence, a Passenger can only fly to a single airport on a single trip.

*Cardinality:* Airport (1): Passenger (M) | *Mandatory:* Airport, Passenger

*Rule:* A Passenger can choose a single airport to fly to. An airport can have Multiple Passengers. This establishes a one-to-many relationship.

**Passenger - Airline Relationship:**

*Assumptions:*

1. Airlines should have passengers to run; the airlines would cease to exist without passengers.
2. Each layover creates a new trip. Hence, a Passenger can only fly a single airline on a single trip.

*Cardinality:* Airline (1): Passenger (M) | *Mandatory:* Airline, Passenger

*Rule:* Passengers can choose a single airline for a trip, while an airline serves multiple passengers. This establishes a one-to-many relationship.

**Airport - Airline Relationship:**

*Assumption:* None.

*Cardinality:* Airport (M): Airline (M) | *Mandatory:* Airline | *Optional:* Airport

*Rule:* Airports can permit multiple airlines, creating a one-to-many relationship. Conversely, airlines can fly to various airports, forming another many-to-many (M: M) relationship.

To solve the M: M relationship, we will be using a bridge entity “Airport\_Airline” here:

**Airport - Airport\_Airline Entity Relationship:**

*Cardinality:* Airport (1): (Airport\_Airline) (M)

*Optional:* Airport

*Rule:* An airport can permit multiple airlines to operate. This creates a 1:M relationship between Airport and Airport\_airline bridge entity. It is optional for the airport to permit an airline to operate.

**Airport\_Airline - Airline Entity Relationship:**

*Cardinality:* (Airport\_Airline) (M): Airline (1)

*Mandatory:* Airport\_Airline, Airline

*Rule:* The "Airport\_Airline" represents a specific arrangement between an airport and an airline, allowing for a one-to-many relationship with airlines. Airlines are mandatory in this context, meaning each Airport\_Airline corresponds to a specific airline's operation at an airport.

**Air Alliance - Airline Relationship:**

*Assumption:* An Air Alliance has to have a minimum of 1 Airline as member.

*Cardinality:* Air Alliance (1): Airline (M) | *Mandatory:* Air Alliance | *Optional:* Airline

*Rule:* An Airline can be a part of only One Air Alliance. An Air Alliance can have Multiple Airlines as its members. An airline choosing an Air\_Alliance is Optional. However, an Air\_Alliance having Airlines as its members is mandatory.

**Airline-Airplane Relationship:**

*Assumption***:** None.

*Cardinality***:** Airline (1): Airplane (M) Mandatory: Airline, Airplane

*Rule:* An airline is mandatory and can have multiple airplanes, creating a one-to-many relationship. Each aircraft can belong to only one airline at a time, making it mandatory for airplanes. This relationship signifies that each plane is exclusively associated with a single airline, ensuring that an airline can operate multiple planes while each is operated by a single airline.

**ENTITY DESCRIPTIONS:**

1. **Entity Name: PASSENGER**

**Entity Description:** A "PASSENGER" is an individual who utilizes transportation services and holds attributes that help identify and classify them for travel purposes.

**Attributes of PASSENGER:**

* **Passport\_No (Primary Key):** A unique identifier associated with the passenger, typically provided by a government or relevant authority. The passport number serves as a primary identification document for international travel. Passport\_No format is unique to each country. In the Passenger entity, this is used as a primary key, which acts as a unique identifier of this entity.
* **FName:** The first name of the passenger. This attribute is crucial for identification and booking purposes.
* **LName:** The last name of the passenger. This attribute is crucial for identification and booking purposes.
* **Age:** The age of the passenger, expressed in years. Age information is essential for various purposes, including fare calculation, eligibility for specific discounts, and travel restrictions.
* **Gender:** The gender of the passenger, which can be categorized as male, female, or other. This attribute helps provide personalized services and adhere to gender-specific regulations if applicable.
* **Citizenship:** The nationality or citizenship status of the passenger, indicating the country to which the passenger belongs. This information is significant for immigration and customs processes and compliance with travel restrictions and visa requirements.

1. **Entity Name: AIR ALLIANCE**

**Entity Description:** An "AIR ALLIANCE" represents a collaborative group or alliance of multiple airlines working together to offer seamless travel worldwide.

**Attributes of AIR ALLIANCE:**

* **Alliance\_Code (Primary Key):** A unique code or identifier distinguishing the air alliance from others. This code serves as a primary key for database management and differentiation of partnerships.
* **Alliance\_Name:** The official name or title of the air alliance. This attribute provides a recognizable identity for the partnership in the industry.
* **Age:** The age of the Air Alliance indicates the number of years since its establishment. This attribute helps us understand the alliance's history and experience in the aviation sector.
* **Founding\_Airline:** The airline that initiated or founded the Air Alliance. This attribute identifies the key airline responsible for creating the alliance and is essential for historical context.
* **No\_of\_Members:** The total count of airlines that are part of the air alliance. This attribute quantifies the size and membership of the coalition, reflecting its scale and influence.
* **FK\_City\_Name (Foreign Key):** The location or city where the Air Alliance has its central administrative office or headquarters. This attribute serves as a reference point for the alliance's operational center.

1. **Entity Name: AIRPORT**

**Entity Description:** An “Airport” is an airfield with a site and installation for the takeoff and landing of an aircraft.

* **Airport Code (Primary Key):** A unique code assigned to the airport, such as the International Air Transport Association (IATA) code or the International Civil Aviation Organization (ICAO) code.
* **Airport\_Name:** The name of the airport.
* **Age:** The age of the airport indicates the number of years since its establishment. This attribute helps us understand the airport's history and experience in the aviation sector.
* **Capacity:** This attribute represents the maximum number of passengers the airport can accommodate at any given time, which may vary based on the size and facilities of the airport.
* **No\_of\_Terminals:** This attribute represents the total count of terminal buildings or facilities at the airport. It is typically an integer value indicating the airport's number of separate terminals.
* **FK\_City\_Name (Foreign Key):** This attribute represents the city’s name where the airport is located. It is typically a text or string data type.

4. **Entity Name: AIRLINE**

**Entity Description:** An airline is a commercial entity that offers freight and passenger air transportation services. Airlines are businesses that move people and cargo locally and abroad by using a fleet of aircraft, including airplanes and helicopters.

* **Airline Code (Primary Key):** A unique identifier for the airline, often using an industry-standard code.
* **Airline\_Name:** The official name of the airline.
* **Age:** The age of the airline indicates the number of years since its establishment. This attribute helps us understand the airline's history and experience in the aviation sector.
* **Country:** The country where the airline is based or registered.
* **Airline\_Fleet\_Size:** The total number of aircraft in the airline's fleet.
* **FK\_Alliance\_Code (Foreign Key):** The unique code of the air alliance that the airline is a member of. This attribute provides a recognizable identity for the partnership in the industry.

1. **Entity Name: CITY**

**Entity Description:** A “CITY” in this context is a pivotal airport location, providing a geographical reference point for air travel operations.

* **Country:** The country to which the city belongs, indicating its national affiliation. This information is significant for international travel, regulatory compliance, and demographic analysis.
* **City Name (Primary Key):** The city’s name, identifying its specific location. This attribute is crucial for establishing associations with airports and providing context for air travel operations.
* **FK Airport\_Code (FK\_Primary Key):** A unique code assigned to the airport, such as the International Air Transport Association (IATA) code or the International Civil Aviation Organization (ICAO) code.
* **Foundation\_Date:** The date the city was officially established or founded. This attribute provides historical context and allows for analysis of the city's development over time.
* **Population:** The total number of people residing in the city. This demographic information is significant to each town.
* **Area:** Size of the city in which there is an airport.

1. **Entity Name: AIRPLANE**

**Entity Description:** An "AIRPLANE" refers to a powered flying vehicle. In the context of this database, airplanes are essential assets owned and operated by airlines for air transportation services.

* **FK Airline Code (FK\_Primary Key):** The unique identifier of the airline that owns or operates the airplane. This attribute establishes a direct relationship between the aircraft and the airline, indicating ownership and operational responsibility.
* **Tail No (Primary Key):** A unique identifier assigned to each airplane, typically displayed on the aircraft's tail. This identifier serves as a primary means of identifying and tracking individual aircraft.
* **Manufacturer:** The company or entity responsible for designing and producing the airplane. This attribute provides information about the origin and manufacture of the aircraft.
* **Model:** The specific model or type of the airplane, indicating its design characteristics and capabilities.
* **Airplane\_Age:** The age of the airplane denotes the number of years since its initial manufacture or entry into service.
* **Capacity:** The number of people an airplane can accommodate.

1. **Entity Name: AIRPORT\_AIRLINE**

**Entity Description:** The entity is a bridge/intersect entity as the AIRPORT and AIRLINE relationship is M: N.

* **ROUTE (Primary Key):** The unique identifier of the AIRPORT\_AIRLINE that signifies the route associated between the airport and the airline.
* **FK\_Airport\_Code:** A unique code assigned to the airport, such as the International Air Transport Association (IATA) code or the International Civil Aviation Organization (ICAO) code. This references the primary key of the AIRPORT entity.
* **FK\_Airline\_Code:** A unique identifier for the airline, often using an industry-standard code. This references the primary key of the AIRLINE entity.

**ER DIAGRAM:**

A diagram of a computer

Description automatically generated

**DDL – DATA DEFINITION LANGUAGE:**

set search\_path to public;

/\* DROP STATEMENTS TO CLEAN UP OBJECTS FROM PREVIOUS RUN \*/

-- Triggers

DROP TRIGGER IF EXISTS aa1\_trig ON airport\_airline;

DROP TRIGGER IF EXISTS tail\_no\_trig ON airplane;

DROP TRIGGER IF EXISTS fleet\_trig ON airplane;

DROP TRIGGER IF EXISTS airline\_code\_trig ON airline;

DROP TRIGGER IF EXISTS alliance\_code\_trig ON air\_alliance;

DROP TRIGGER IF EXISTS airport\_code\_trig ON airport;

-- Sequences

DROP SEQUENCE IF EXISTS alliance\_seq;

DROP SEQUENCE IF EXISTS airline\_seq;

DROP SEQUENCE IF EXISTS airport\_code\_seq;

DROP SEQUENCE IF EXISTS tail\_no\_seq;

DROP SEQUENCE IF EXISTS aa\_seq;

-- Views

DROP VIEW IF EXISTS airplaneinfo;

DROP VIEW IF EXISTS routeinfo;

-- Drop the existing schema if it exists

DROP SCHEMA IF EXISTS airdb CASCADE;

-- Create a new schema with the same name (Run this first time)

CREATE SCHEMA airdb;

set search\_path to airdb;

-- Tables

DROP TABLE IF EXISTS airplane CASCADE;

DROP TABLE IF EXISTS airline CASCADE;

DROP TABLE IF EXISTS air\_alliance CASCADE;

DROP TABLE IF EXISTS passenger CASCADE;

DROP TABLE IF EXISTS city CASCADE;

DROP TABLE IF EXISTS airport CASCADE;

DROP TABLE IF EXISTS airport\_airline CASCADE;

/\* CREATE TABLES BASED ON ENTITIES \*/

--CREATION OF AIRPORT TABLE

CREATE TABLE airport (

airport\_code VARCHAR(10) PRIMARY KEY NOT NULL,

airport\_name VARCHAR,

no\_of\_terminals INT,

age INT,

capacity INT

);

--CREATION OF CITY TABLE

CREATE TABLE City (

city\_name varchar(50) NOT NULL,

fk\_airport\_code varchar NOT NULL,

country varchar(50),

area int,

foundation\_date date,

population int,

PRIMARY KEY (city\_name, fk\_airport\_code),

FOREIGN KEY (fk\_airport\_code) REFERENCES airport(airport\_code),

CONSTRAINT UC\_City\_City\_Name UNIQUE (City\_name)

);

--CREATION OF PASSENGER TABLE

CREATE TABLE Passenger (

passport\_no varchar(9) PRIMARY KEY NOT NULL,

fname char(100),

lname char(100),

age int,

gender char(20),

citizenship char(20)

);

--CREATION OF AIR\_ALLIANCE TABLE

CREATE TABLE Air\_Alliance (

Alliance\_Code VARCHAR(10) NOT NULL,

Alliance\_Name VARCHAR(40) NOT NULL,

Founding\_Airline VARCHAR(50) NOT NULL,

Age INTEGER NOT NULL,

No\_of\_Members INTEGER NOT NULL,

City\_Name VARCHAR(50) NOT NULL,

CONSTRAINT PK\_Air\_Alliance PRIMARY KEY (Alliance\_Code),

FOREIGN KEY (city\_name) REFERENCES City (city\_name)

);

--CREATION OF AIRLINE TABLE

CREATE TABLE Airline (

Airline\_Code VARCHAR(7) NOT NULL,

Airline\_Name VARCHAR(40) NOT NULL,

Country VARCHAR(50) NOT NULL,

Age INTEGER NOT NULL,

Airline\_Fleet\_Size INTEGER NOT NULL,

Alliance\_Code VARCHAR(10),

CONSTRAINT PK\_Airline PRIMARY KEY (Airline\_Code),

FOREIGN KEY (Alliance\_Code) REFERENCES Air\_Alliance

);

-- CREATION OF AIRPLANE TABLE

CREATE TABLE Airplane (

Tail\_No VARCHAR(10) NOT NULL,

Airline\_Code VARCHAR(7) NOT NULL,

Manufacturer VARCHAR(50) NOT NULL,

Model VARCHAR(50) NOT NULL,

Airplane\_Age INTEGER NOT NULL,

Capacity INTEGER NOT NULL,

PRIMARY KEY (Tail\_No, Airline\_Code),

FOREIGN KEY (Airline\_Code) REFERENCES Airline(Airline\_Code)

);

--CREATION OF AIRPORT\_AIRLINE BRIDGE TABLE

CREATE TABLE airport\_airline (

ROUTE INT PRIMARY KEY,

airport\_code varchar(10),

airline\_code varchar(7),

FOREIGN KEY (airport\_code) REFERENCES airport(airport\_code),

FOREIGN KEY (airline\_code) REFERENCES airline(airline\_code)

);

/\* CREATE SEQUENCES \*/

--Creating Sequence for Airport\_Code

CREATE SEQUENCE airport\_code\_seq

START 100

INCREMENT 1;

--Creating Sequence for Air\_Alliance\_Code

CREATE SEQUENCE alliance\_seq

start with 100

increment by 100;

--Creating Sequence for Airline\_Code

CREATE SEQUENCE airline\_seq

start with 101

increment by 10;

--Creating Sequence for Tail\_No

CREATE SEQUENCE tail\_no\_seq

START WITH 1001

INCREMENT BY 1;

--Creating Sequence for Airport\_Airline\_Seq

CREATE SEQUENCE aa\_seq

START WITH 1

INCREMENT BY 1;

/\*CREATE TRIGGERS\*/

--CREATING TRIGGER FOR AIRPORT\_CODE

--Function

CREATE OR REPLACE FUNCTION airport\_insert\_trigger()

RETURNS TRIGGER AS $$

BEGIN

NEW.airport\_code = 'ARPT' || nextval('airport\_code\_seq');

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Trigger

CREATE TRIGGER airport\_code\_trig

BEFORE INSERT ON airport

FOR EACH ROW

EXECUTE FUNCTION airport\_insert\_trigger();

--CREATING TRIGGER FOR ALLIANCE\_CODE

--Function

CREATE OR REPLACE FUNCTION alliance\_insert\_trigger()

RETURNS TRIGGER AS $$

BEGIN

NEW.alliance\_code = 'ALL'||nextval('alliance\_seq');

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

--Trigger

CREATE TRIGGER alliance\_code\_trig

BEFORE INSERT ON Air\_Alliance

FOR EACH ROW

EXECUTE FUNCTION alliance\_insert\_trigger();

--CREATING FUNCTION AND TRIGGER FOR AIRLINE\_CODE

CREATE OR REPLACE FUNCTION airline\_insert\_trigger()

RETURNS TRIGGER AS $$

BEGIN

NEW.airline\_code = 'AIR'||nextval('airline\_seq');

--Updating Air\_Alliance Table

IF NOT EXISTS (SELECT FROM Air\_Alliance WHERE NEW.airline\_name=Air\_Alliance.Founding\_Airline) THEN

UPDATE Air\_Alliance

SET No\_of\_Members=No\_of\_Members+1

WHERE NEW.alliance\_code=Air\_Alliance.alliance\_code;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

--Trigger

CREATE TRIGGER airline\_code\_trig

BEFORE INSERT ON Airline

FOR EACH ROW

EXECUTE FUNCTION airline\_insert\_trigger();

--CREATE FUNCTION FOR TAIL\_NO

CREATE OR REPLACE FUNCTION airplane\_insert\_trigger()

RETURNS TRIGGER AS $$

BEGIN

new.tail\_no = 'TN' || nextval('tail\_no\_seq');

--Updating Airline Table

UPDATE Airline

SET Airline\_Fleet\_size=Airline\_Fleet\_size+1

WHERE NEW.airline\_code=Airline.airline\_code;

RETURN new;

END;

$$ LANGUAGE plpgsql;

--Trigger

CREATE TRIGGER tail\_no\_trig

BEFORE INSERT ON Airplane

FOR EACH ROW

EXECUTE FUNCTION airplane\_insert\_trigger();

--CREATING FUNCTION AND TRIGGER FOR DELETING AIRPLANE ENTRY

CREATE OR REPLACE FUNCTION airplane\_delete\_trigger()

RETURNS TRIGGER AS $$

BEGIN

--Updating Airline Table

UPDATE Airline

SET Airline\_Fleet\_size=Airline\_Fleet\_size-1

WHERE OLD.airline\_code=Airline.airline\_code;

RETURN OLD;

END;

$$ LANGUAGE plpgsql;

--Trigger

CREATE TRIGGER fleet\_trig

BEFORE DELETE ON Airplane

FOR EACH ROW

EXECUTE FUNCTION airplane\_delete\_trigger();

--CREATING FUNCTION AND TRIGGER FOR DELETING AIRLINE ENTRY

CREATE OR REPLACE FUNCTION airline\_delete\_trigger()

RETURNS TRIGGER AS $$

BEGIN

--Updating Air\_Alliance Table

UPDATE Air\_Alliance

SET No\_of\_Members=No\_of\_Members-1

WHERE OLD.alliance\_code=Air\_Alliance.Alliance\_code;

RETURN OLD;

END;

$$ LANGUAGE plpgsql;

--Trigger

CREATE TRIGGER member\_trig

BEFORE DELETE ON Airline

FOR EACH ROW

EXECUTE FUNCTION airline\_delete\_trigger();

--CREATING TRIGGER AND FUNCTION FOR AIRLINE\_AIRPORT ENTITY

CREATE OR REPLACE FUNCTION airline\_airport\_trigger()

RETURNS TRIGGER AS $$

BEGIN

NEW.Route = nextval('aa\_seq');

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

--Trigger

CREATE TRIGGER aa1\_trig

BEFORE INSERT ON airport\_airline

FOR EACH ROW

EXECUTE FUNCTION airline\_airport\_trigger();

/\*CREATING VIEWS\*/

--SHOWS THE INFO OF AIRPLANE

create or replace view airplaneInfo as

select tail\_no, airline\_code, manufacturer, model from airplane;

-- SHOWS THE INFO OF DESTINATION ROUTE

create or replace view routeinfo as

select aa.route, aa.airline\_code, ar.airline\_name, aa.airport\_code, ci.city\_name

from airline ar

left outer join airport\_airline aa on ar.airline\_code = aa.airline\_code

left outer join city ci on aa.airport\_code = ci.fk\_airport\_code;

**DML – DATA MANUPULATION LANGUAGE:**

/\*INSERTING DATA INTO TABLES\*/

INSERT INTO airport (airport\_name, no\_of\_terminals, age, capacity)

VALUES

('Kempegowda International Airport', 2, 20, 5000),

('John F. Kennedy International Airport', 6, 97, 59335),

('London Heathrow Airport', 5, 77, 80630),

('Haneda Airport', 4, 155, 87380),

('Charles de Gaulle Airport', 3, 535, 76555),

('Sydney Kingsford Smith Airport', 3, 92, 42620),

('Beijing Capital International Airport', 3, 104, 95800),

('Galeão - Antonio Carlos Jobim International Airport', 2, 61, 17178),

('Cape Town International Airport', 2, 72, 89000),

('Dubai International Airport', 3, 59, 89000),

('Sheremetyevo International Airport', 5, 64, 40350),

('Frankfurt Airport', 4, 90, 65416),

('Dallas/Fort Worth International Airport', 5, 49, 67472),

('Kuala Lumpur International Airport', 2, 25, 78000),

('Addis Ababa Bole International Airport', 3, 79, 11000);

INSERT INTO city (city\_name, fk\_airport\_code, country, area, foundation\_date, population)

VALUES

('Bengaluru', 'ARPT100', 'India', 710, '1537-07-03', 12572222),

('New York', 'ARPT101', 'USA', 469, '1624-05-24', 8537673),

('London', 'ARPT102', 'UK', 1573, '1043-12-31', 8961989),

('Tokyo', 'ARPT103', 'Japan', 2188, '1868-07-29', 9272690),

('Paris', 'ARPT104', 'France', 105, '1486-09-21', 2148271),

('Sydney', 'ARPT105', 'Australia', 469, '1788-01-26', 5312163),

('Beijing', 'ARPT106', 'China', 16411, '1919-03-04', 21542000),

('Rio de Janeiro', 'ARPT107', 'Brazil', 487, '1565-03-01', 6747815),

('Cape Town', 'ARPT108', 'South Africa', 948, '1652-04-06', 433688),

('Dubai', 'ARPT109', 'UAE', 412, '1971-12-02', 3332300),

('Moscow', 'ARPT110', 'Russia', 2511, '1147-04-11', 12615279),

('Frankfurt', 'ARPT111', 'Germany', 248, '1936-06-21', 7633803),

('Fort Worth', 'ARPT112', 'USA', 790, '1849-02-12', 927720),

('Kuala Lumpur', 'ARPT113', 'Malaysia', 244, '1857-01-01', 7898700),

('Addis Ababa', 'ARPT114', 'Ethiopia', 530, '1886-10-19', 3400000);

INSERT INTO passenger (passport\_no, fname, lname, age, gender, citizenship)

VALUES

('L77777777', 'Rahul', 'Verma', 26, 'Male', 'IN'),

('A12345678', 'John', 'Doe', 28, 'Male', 'US'),

('G98765432', 'Jane', 'Smith', 35, 'Female', 'UK'),

('C24681357', 'Robert', 'Johnson', 42, 'Male', 'CA'),

('D87654321', 'Alice', 'Brown', 22, 'Female', 'AU'),

('E55555555', 'Michael', 'White', 30, 'Male', 'DE'),

('K11111111', 'Maria', 'Garcia', 29, 'Female', 'ES'),

('H22222222', 'Andreas', 'Mueller', 32, 'Male', 'DE'),

('I33333333', 'Sophie', 'Dupont', 27, 'Female', 'FR'),

('B44444444', 'Makoto', 'Yamada', 31, 'Male', 'JP'),

('F99999999', 'Anna', 'Ivanova', 24, 'Female', 'RU');

INSERT INTO Air\_Alliance (Alliance\_Name, Founding\_Airline, Age, No\_of\_Members, City\_Name)

VALUES

('Star Alliance', 'Lufthansa', 27, 1, 'Frankfurt'),

('Oneworld', 'American Airlines', 25, 1, 'Fort Worth'),

('Skyteam', 'Delta Airlines', 23, 1, 'New York'),

('Skyline', 'LATAM Airlines', 6, 1, 'Rio de Janeiro'),

('Jetstream', 'Emirates', 13, 1, 'Dubai'),

('Skyconnect', 'Indigo', 18, 1, 'Bengaluru'),

('Airquest', 'AirAsia', 19, 1, 'Kuala Lumpur'),

('Aeroglobe', 'Ethiopian Airlines', 29, 1, 'Addis Ababa'),

('SwiftFly', 'ANA', 7, 1, 'Tokyo'),

('Wingstar', 'Qantas', 24, 1, 'Sydney');

--Inserting Values into Airline Table

INSERT INTO Airline (Airline\_Name, Country, Age, Airline\_Fleet\_Size, Alliance\_Code)

VALUES

('Air France', 'France', 90, 243, 'ALL100'),

('Lufthansa', 'Germany', 68, 274, 'ALL100'),

('British Airways', 'United Kingdom', 49, 257, 'ALL100'),

('Japan Airlines', 'Japan', 72, 145, 'ALL200'),

('TAP Air Portugal', 'Portugal', 77, 75, 'ALL200'),

('Etihad', 'United Arab Emirates', 20, 90, 'ALL200'),

('American Airlines', 'United States', 87, 944, 'ALL200'),

('Alaska Airlines', 'United States', 79, 305, 'ALL300'),

('Delta Airlines', 'United States', 94, 945, 'ALL300'),

('Spirit Airlines', 'United States', 33, 197, 'ALL300'),

('LATAM Airlines', 'Brazil', 91, 149, 'ALL400'),

('Qatar Airways', 'Qatar', 29, 253, 'ALL500'),

('Emirates', 'United Arab Emirates', 38, 260, 'ALL500'),

('Air India', 'India', 77, 123, 'ALL600'),

('Indigo', 'India', 17, 335, 'ALL600'),

('Vistara', 'India', 8, 63, 'ALL600'),

('AirAsia', 'Malaysia', 26, 255, 'ALL700'),

('Cebu Pacific', 'Phillipines', 27, 63, 'ALL700'),

('Vietnam Airlines', 'Vietnam', 67, 97, 'ALL700'),

('Ethiopian Airlines', 'Ethiopia', 77, 140, 'ALL800'),

('Kenya Airways', 'Kenya', 46, 34, 'ALL800'),

('EgyptAir', 'Egypt', 90, 78, 'ALL800'),

('ANA', 'Japan', 70, 212, 'ALL900'),

('Asiana Airlines', 'Korea', 34, 78, 'ALL900'),

('Air China', 'China', 35, 492, 'ALL900'),

('Qantas', 'Australia', 101, 125, 'ALL1000'),

('Aerolineas Argentinas', 'Argentina', 74, 82, 'ALL400'),

('Air New Zealand', 'New Zealand', 58, 106, 'ALL1000');

--Insert values into Airplane

INSERT INTO Airplane (Airline\_Code, Manufacturer, Model, Airplane\_Age, Capacity)

VALUES

('AIR101', 'Boeing', '737', 5, 150),

('AIR111', 'Airbus', 'A320', 4, 180),

('AIR131', 'Boeing', '777', 8, 300),

('AIR141', 'Airbus', 'A350', 3, 250),

('AIR151', 'Embraer', 'E190', 6, 100),

('AIR161', 'Boeing', '787', 2, 240),

('AIR171', 'Airbus', 'A330', 7, 220),

('AIR181', 'Boeing', '747', 10, 400),

('AIR191', 'Airbus', 'A321', 4, 130),

('AIR201', 'Bombardier', 'Q400', 5, 90),

('AIR211', 'Airbus', 'A380', 6, 550),

('AIR221', 'Boeing', '767', 9, 280),

('AIR231', 'Airbus', 'A321', 3, 170),

('AIR241', 'Embraer', 'E175', 4, 80),

('AIR251', 'Bombardier', 'Q400', 7, 90),

('AIR261', 'Airbus', 'A320', 5, 120),

('AIR271', 'Boeing', '737', 1, 160),

('AIR281', 'Airbus', 'A330neo', 2, 240),

('AIR291', 'Embraer', 'E195', 3, 110),

('AIR301', 'Bombardier', 'CRJ700', 6, 75),

('AIR311', 'Boeing', '757', 12, 200),

('AIR321', 'Airbus', 'A320', 15, 180),

('AIR331', 'Embraer', 'E170', 7, 80),

('AIR341', 'Bombardier', 'Global 6000', 4, 20),

('AIR351', 'Airbus', 'A340', 18, 350),

('AIR361', 'Boeing', '737', 20, 150),

('AIR121', 'Boeing', '737', 20, 166),

('AIR371', 'Embraer', 'E145', 9, 50);

INSERT INTO Airport\_Airline (Airport\_code, Airline\_Code)

VALUES

('ARPT100', 'AIR101'),

('ARPT101', 'AIR111'),

('ARPT102', 'AIR131'),

('ARPT103', 'AIR141'),

('ARPT104', 'AIR151'),

('ARPT105', 'AIR161'),

('ARPT106', 'AIR171'),

('ARPT107', 'AIR181'),

('ARPT108', 'AIR191'),

('ARPT109', 'AIR201'),

('ARPT110', 'AIR211'),

('ARPT111', 'AIR221'),

('ARPT112', 'AIR231'),

('ARPT113', 'AIR241'),

('ARPT114', 'AIR251'),

('ARPT100', 'AIR261'),

('ARPT101', 'AIR271'),

('ARPT102', 'AIR281'),

('ARPT103', 'AIR291'),

('ARPT104', 'AIR301'),

('ARPT105', 'AIR311'),

('ARPT106', 'AIR321'),

('ARPT107', 'AIR331'),

('ARPT108', 'AIR341'),

('ARPT109', 'AIR351'),

('ARPT110', 'AIR361'),

('ARPT111', 'AIR121'),

('ARPT112', 'AIR371');

**SQL QUERIES:**

/\*SQL FILE\*/

--Q1 SELECT ALL COLUMNS AND ALL ROWS FROM ONE TABLE

SELECT \* FROM Airline;

--Q2 SELECT FIVE COLUMNS AND ALL ROWS FROM ONE TABLE

SELECT Passport\_No, fname, lname, Age, Citizenship FROM Passenger;

--Q3 SELECT ALL COLUMNS FROM ALL ROWS FROM ONE VIEW

select \* from RouteInfo;

--Q4 USING A JOIN ON 2 TABLES, SELECT ALL COLUMNS AND ALL ROWS FROM THE TABLES WITHOUT THE USE OF A CARTESIAN PRODUCT

select \*

from airline ar

left outer join airplane ap on ar.airline\_code = ap.airline\_code;

--Q5 SELECT AND ORDER DATA RETRIEVED FROM ONE TABLE

select \* from airport

order by age desc;

--Q6 USING A JOIN ON 3 TABLES, SELECT 5 COLUMNS FROM THE 3 TABLES. USE SYNTAX THAT WOULD LIMIT THE OUTPUT TO 10 ROWS

select ap.tail\_no, ap.manufacturer, ar.airline\_code, ar.airline\_name, al.alliance\_name

from airplane ap

left outer join airline ar on ap.airline\_code = ar.airline\_code

left outer join air\_alliance al on ar.alliance\_code = al.alliance\_code

limit 10;

--Q7 SELECT DISTINCT ROWS USING JOINS ON 3 TABLES

select distinct ap.manufacturer

from airplane ap

left outer join airline ar on ap.airline\_code = ar.airline\_code

left outer join air\_alliance al on ar.alliance\_code = al.alliance\_code

where al.alliance\_name = 'Skyteam';

--Q8 USE GROUP BY AND HAVING IN A SELECT STATEMENT USING ONE OR MORE TABLES

select ar.airline\_name, al.alliance\_name, al.no\_of\_members

from air\_alliance al

left outer join airline ar on al.alliance\_code = ar.alliance\_code

group by ar.airline\_name, al.alliance\_name, al.no\_of\_members

having al.no\_of\_members > 3;

--Q9 USE IN CLAUSE TO SELECT DATA FROM ONE OR MORE TABLES

select tail\_no, model

from airplane

where manufacturer in ('Boeing','Airbus');

--Q10 SELECT LENGTH OF ONE COLUMN FROM ONE TABLE

select length(passport\_no)

from passenger;

--Q11 DELETE ONE RECORD FROM ONE TABLE. USE SELECT STATEMENTS TO DEMONSTRATE THE TABLE CONTENTS BEFORE AND AFTER THE DELETE STATEMENT. MAKE SURE YOU USE ROLLBACK AFTERWARDS SO THAT THE DATA WILL NOT BE PHYSICALLY REMOVED

---before deletion

select \* from passenger;

--deletion

begin transaction;

delete from passenger where fname='Jane';

--after deletion

rollback;

select \* from passenger;

--Q12 UPDATE ONE RECORD FROM ONE TABLE. USE SELECT STATEMENTS TO DEMONSTRATE THE TABLE CONTENTS BEFORE AND AFTER THE UPDATE STATEMENT. MAKE SURE YOU USE ROLLBACK AFTERWARDS SO THAT THE DATA WILL NOT BE PHYSICALLY REMOVED

---before updation

select \* from passenger;

--updation

begin transaction;

update passenger set citizenship = 'US', passport\_no='A87654321'

where fname='Rahul';

--after deletion

rollback;

select \* from passenger;

--Q13 Advanced Query 1

--THIS SHOWS THE INFO OF DESTINATION ROUTE

select aa.route, aa.airline\_code, ar.airline\_name, aa.airport\_code, ci.city\_name

from airline ar

left outer join airport\_airline aa on ar.airline\_code = aa.airline\_code

left outer join city ci on aa.airport\_code = ci.fk\_airport\_code;

--Q14 Advanced Query 2

-- DISPLAY THE FIRST RECORD WITH TAIL\_NUMBER, ALLIANCE\_NAME, AND MANUFACTURER OF ALL THE AIRPLANES WHOSE AIRLINES ARE BASED IN THE U.S.A AND ARE MANUFACTURED BY THE COMPANY ‘BOEING.’

select ap.tail\_no, al.alliance\_name, ap.manufacturer

from airplane ap

left outer join airline ar on ap.airline\_code = ar.airline\_code

left outer join air\_alliance al on ar.alliance\_code = al.alliance\_code

where ar.country in (select country from airline where country = 'United States')

group by ap.tail\_no, al.alliance\_name, ap.manufacturer

having ap.manufacturer = 'Boeing'

limit 1;