**Final Project Part 2**

**(Group Name: SoonYeong)**

We create DDLs for both MySQL and SQLite3, but we have documented DDL and DML only for MySQL.

1. **Create Conceptual Diagram/Schema for database**
   1. **Add a screenshot (ERD Diagram)**

|  |
| --- |
| **A computer screen shot of a computer  AI-generated content may be incorrect.** |

* 1. **Explain the Diagram structure (entities, relations types, objects)**

This is a relational database for Flight Information Display System (FIDS).  
The database consists of 7 entities (tables), properly normalized, with clean foreign key relationships.

* + 1. Entities (Tables) and Their Roles

|  |  |
| --- | --- |
| **Entity (Table)** | **Description** |
| **Airports** | Stores airport details like code, name, city, country, and status (active/inactive). |
| **Airlines** | Stores airline details with codes and active status. |
| **Remarks** | Stores flight remarks or statuses like "Progressing", "Delayed", etc. |
| **ActiveFlightSchedules** | Holds active flight details, schedules, airports, airline, status remarks. |
| **Users** | Stores users who manage or view the system, linked optionally to an airport or airline. |
| **Roles** | Defines roles for users (e.g., Admin, Operations Manager). |
| **UserRoles** | Joins users and roles (many-to-many relationship). |

* + 1. Relationship Types

|  |  |  |
| --- | --- | --- |
| **Relationship** | **Type** | **Description** |
| Airports → ActiveFlightSchedules | One-to-Many | One airport can have many flights assigned as either departure or arrival points. |
| Airlines → ActiveFlightSchedules | One-to-Many | One airline operates many flights. |
| Remarks → ActiveFlightSchedules | One-to-Many | One remark status (e.g., Delayed) applies to many flights. |
| Airports → Users | One-to-Many (optional) | A user may be associated with an airport (optional). |
| Airlines → Users | One-to-Many (optional) | A user may be associated with an airline (optional). |
| Users → UserRoles → Roles | Many-to-Many | Users can have multiple roles, roles can belong to multiple users. |
| UserRoles → Users | Many-to-One | UserRoles table references Users. |
| UserRoles → Roles | Many-to-One | UserRoles table references Roles. |

* + 1. Objects

|  |  |  |
| --- | --- | --- |
| **Table** | **Primary Key** | **Foreign Keys** |
| Airports | AirportCode |  |
| Airlines | AirlineCode |  |
| Remarks | RemarkCode |  |
| ActiveFlightSchedules | FlightId (Auto-increment) | AirportCode, AirlineCode, OriginDestAirport, Remarks |
| Users | UserID | AirportCode, AirlineCode |
| Roles | RoleID | - |
| UserRoles | UserRoleID | UserID, RoleID |

1. **Database**
   1. **Database Constraints**

Various constraints such as primary keys (PK), foreign keys (FK), unique constraints, and auto-increment are applied to ensure data consistency and integrity.

|  |  |  |
| --- | --- | --- |
| **Constraint Type** | **Purpose** | **Where Used** |
| **Primary Key** | Uniquely identifies a row | All tables |
| **Unique** | Prevents duplicate entries in one or more columns | Airports, Airlines, Remarks, Roles, UserRoles |
| **Foreign Key** | Enforces relationships between tables | Users, UserRoles, ActiveFlightSchedules |
| **Check** | Restricts values in a column to specified options | UseYn |
| **Default** | Provides default values if none are supplied | UseYn |
| **Auto Increment** | Automatically generates unique sequential numbers | ActiveFlightSchedules(FlightId) |

1. **Write code to create a database and build queries**
   1. **DDL (MySQL)**

|  |
| --- |
| SET FOREIGN\_KEY\_CHECKS = 1;  -- =============================  -- Table: Airports  -- Stores airport information including location and status.  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE Airports (  AirportCode CHAR(3) NOT NULL COMMENT 'Primary key representing the airport code.',  AirportName VARCHAR(50) NOT NULL COMMENT 'Name of the airport along with its IATA code.',  City VARCHAR(50) COMMENT 'Name of the city where the airport is located.',  Country VARCHAR(50) COMMENT 'Name of the country where the airport is located.',  UseYn CHAR(1) NOT NULL DEFAULT 'Y' CHECK (UseYn IN ('Y', 'N')) COMMENT 'Indicates whether the record is active. (Y for Yes, N for No)',  PRIMARY KEY (AirportCode),  UNIQUE (AirportName, City, Country) -- Avoid duplicate airport definitions  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Stores airport information including location and status.';  -- =============================  -- Table: Airlines  -- Stores airline information including status.  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE Airlines (  AirlineCode CHAR(2) NOT NULL COMMENT 'Primary key representing the airline code.',  AirlineName VARCHAR(50) NOT NULL COMMENT 'Name of the airline along with its IATA code.',  UseYn CHAR(1) NOT NULL DEFAULT 'Y' CHECK (UseYn IN ('Y', 'N')) COMMENT 'Indicates whether the record is active. (Y for Yes, N for No)',  PRIMARY KEY (AirlineCode),  UNIQUE (AirlineName) -- Ensure no duplicate airline names  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Stores airline information including status.';  -- =============================  -- Table: Remarks  -- Stores remark codes and descriptions for flight status.  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE Remarks (  RemarkCode CHAR(3) NOT NULL COMMENT 'Primary key representing the remark code.',  RemarkName VARCHAR(50) NOT NULL COMMENT 'Description of the current flight status (e.g., Progressing, Delayed).',  UseYn CHAR(1) NOT NULL DEFAULT 'Y' CHECK (UseYn IN ('Y', 'N')) COMMENT 'Indicates whether the record is active. (Y for Yes, N for No)',  PRIMARY KEY (RemarkCode),  UNIQUE (RemarkName) -- Avoid duplicate remark names  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Stores remark codes and descriptions for flight status.';  -- =============================  -- Table: Roles  -- Stores system user roles.  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE Roles (  RoleID VARCHAR(20) NOT NULL COMMENT 'Primary key representing the role ID.',  RoleName VARCHAR(50) NOT NULL COMMENT 'Name of the role used to identify the role within the system.',  PRIMARY KEY (RoleID),  UNIQUE (RoleName) -- Ensure unique role names  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Stores system user roles.';  -- =============================  -- Table: Users  -- Stores user information including linked airline and airport.  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE Users (  UserID VARCHAR(20) NOT NULL COMMENT 'Primary key representing the user ID.',  UserName VARCHAR(50) NOT NULL COMMENT 'Name used to identify the user within the system.',  Password VARCHAR(64) NOT NULL COMMENT 'User password stored as a hash (e.g., SHA-256).',  AirportCode CHAR(3) COMMENT 'Foreign key referencing the Airports table.',  AirlineCode CHAR(2) COMMENT 'Foreign key referencing the Airlines table.',  PRIMARY KEY (UserID),  FOREIGN KEY (AirportCode) REFERENCES Airports(AirportCode),  FOREIGN KEY (AirlineCode) REFERENCES Airlines(AirlineCode)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Stores user information including linked airport and airline.';  -- =============================  -- Table: UserRoles  -- Assigns roles to users (many-to-many relationship).  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE UserRoles (  UserRoleID VARCHAR(20) NOT NULL COMMENT 'Primary key representing the UserRole ID.',  UserID VARCHAR(20) NOT NULL COMMENT 'Foreign key referencing the Users table.',  RoleID VARCHAR(20) NOT NULL COMMENT 'Foreign key referencing the Roles table.',  PRIMARY KEY (UserRoleID),  UNIQUE (UserID, RoleID), -- Ensure a user cannot have the same role assigned multiple times  FOREIGN KEY (UserID) REFERENCES Users(UserID),  FOREIGN KEY (RoleID) REFERENCES Roles(RoleID)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Assigns roles to users (many-to-many relationship).';  -- =============================  -- Table: ActiveFlightSchedules  -- Stores active flight schedules, including timing, locations, and remarks.  -- Author: Soon-Hyuck Lee  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  CREATE TABLE ActiveFlightSchedules (  FlightId INT NOT NULL AUTO\_INCREMENT COMMENT 'Primary key uniquely identifying each active flight schedule. Auto-incrementing sequential numeric value.',  FlightNumber VARCHAR(12) NOT NULL COMMENT 'Numeric part of the flight number, e.g., "1234". Used with AirlineCode to form full flight number (e.g., "AA1234").',  AirportCode CHAR(3) NOT NULL COMMENT 'Foreign key referencing Airports table, representing the departure airport code.',  AirlineCode CHAR(2) NOT NULL COMMENT 'Foreign key referencing Airlines table, representing the airline code.',  ScheduledDate VARCHAR(8) NOT NULL COMMENT 'Scheduled date of the flight in YYYYMMDD format (e.g., 20250411).',  ScheduledTime VARCHAR(4) NOT NULL COMMENT 'Scheduled time of the flight in 24-hour HHmm format (e.g., 1530).',  EstimatedDate VARCHAR(8) COMMENT 'Estimated date of the flight in YYYYMMDD format.',  EstimatedTime VARCHAR(4) COMMENT 'Estimated time of the flight in 24-hour HHmm format.',  OriginDestAirport CHAR(3) NOT NULL COMMENT 'Foreign key referencing Airports table, representing the arrival airport code.',  Remarks CHAR(3) COMMENT 'Foreign key referencing Remarks table, representing the remark code.',  PRIMARY KEY (FlightId),  FOREIGN KEY (AirportCode) REFERENCES Airports(AirportCode),  FOREIGN KEY (AirlineCode) REFERENCES Airlines(AirlineCode),  FOREIGN KEY (OriginDestAirport) REFERENCES Airports(AirportCode),  FOREIGN KEY (Remarks) REFERENCES Remarks(RemarkCode)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='Stores active flight schedules, including timing, locations, and remarks.'; |

* 1. **DML (Insert Sample Data)**

|  |
| --- |
| -- =============================  -- Sample Data: Airports  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO Airports (AirportCode, AirportName, City, Country, UseYn) VALUES  ('JFK', 'John F. Kennedy International Airport', 'New York', 'USA', 'Y'),  ('LAX', 'Los Angeles International Airport', 'Los Angeles', 'USA', 'Y'),  ('ICN', 'Incheon International Airport', 'Seoul', 'South Korea', 'Y');  -- =============================  -- Sample Data: Airlines  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO Airlines (AirlineCode, AirlineName, UseYn) VALUES  ('AA', 'American Airlines', 'Y'),  ('DL', 'Delta Air Lines', 'Y'),  ('KE', 'Korean Air', 'Y');  -- =============================  -- Sample Data: Remarks  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO Remarks (RemarkCode, RemarkName, UseYn) VALUES  ('PRG', 'Progressing', 'Y'),  ('BRD', 'Boarding', 'Y'),  ('DPT', 'Departed', 'Y'),  ('ARR', 'Arrived', 'Y'),  ('DLT', 'Delayed', 'Y'),  ('CNL', 'Cancelled', 'Y');  -- =============================  -- Sample Data: Roles  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO Roles (RoleID, RoleName) VALUES  ('ADMIN', 'Administrator'),  ('OPS', 'Operations Manager'),  ('VIEWER', 'Viewer');  -- =============================  -- Sample Data: Users  -- Passwords are hashed (use SHA-256 or better in production!)  -- Here simple text for demonstration  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO Users (UserID, UserName, Password, AirportCode, AirlineCode) VALUES  ('user01', 'Alice Johnson', '5e884898da28047151d0e56f8dc62927', 'JFK', 'AA'),  ('user02', 'Bob Smith', '6cb75f652a9b52798eb6cf2201057c73', 'LAX', 'DL'),  ('user03', 'Charlie Kim', '2b3a8d0a34e3e237c54b4cf2d80a3c41', 'ICN', 'KE');  -- =============================  -- Sample Data: UserRoles  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO UserRoles (UserRoleID, UserID, RoleID) VALUES  ('UR01', 'user01', 'ADMIN'),  ('UR02', 'user02', 'OPS'),  ('UR03', 'user03', 'VIEWER');  -- =============================  -- Sample Data: ActiveFlightSchedules  -- Author: Jiyeong Oh  -- Date Created: 2025-04-11  -- Last Modified: 2025-04-11  -- =============================  INSERT INTO ActiveFlightSchedules (FlightNumber, AirportCode, AirlineCode, ScheduledDate, ScheduledTime, EstimatedDate, EstimatedTime, OriginDestAirport, Remarks) VALUES  ('1001', 'JFK', 'AA', '20250415', '0800', '20250415', '0815', 'LAX', 'PRG'),  ('2020', 'LAX', 'DL', '20250416', '0930', '20250416', '0945', 'ICN', 'BRD'),  ('3050', 'ICN', 'KE', '20250417', '2200', '20250417', '2230', 'JFK', 'DPT'); |

* 1. **DML (Collect & Insert Data)**

|  |
| --- |
| -- =============================  -- Author: Jiyeong Oh  -- Date Created: 2025-04-15  -- Last Modified: 2025-04-15  -- =============================  from selenium import webdriver  from bs4 import BeautifulSoup  import time  driver = webdriver.Chrome()  driver.get("https://www.ind.com/flights")  time.sleep(5) # Waiting unitl the data is loaded  soup = BeautifulSoup(driver.page\_source, "html.parser")  value = []  table = soup.select\_one("table.flight-destinations\_\_table")  for row in table.select("tbody"):  cols = row.find\_all("td")  row\_values = []  for col in cols:  text = col.text.strip()  combined = text  row\_values.append(combined)  value.append(row\_values)  value = value[0]  driver.quit()  import pandas as pd  df = pd.DataFrame(value, columns=['AirportName'])  df['AirportCode'] = df['AirportName'].apply(lambda x: x.split('(')[-1].split(')')[0])  df['AirportName'] = df['AirportName'].apply(lambda x: x.split('\*')[0].split('(')[0])  df.loc[len(df)] = ['Indianapolis', 'IND']  df\_airports = df  df\_airports  import pymysql  conn = pymysql.connect(  host='localhost',  user='root',  password='root',  db='flight\_data',  port=8889,  charset='utf8mb4'  )  cursor = conn.cursor()  sql = """  INSERT INTO Airports (AirportCode, AirportName)  VALUES (%s, %s)  """    values = df\_airports[['AirportCode', 'AirportName']].values.tolist()  cursor.executemany(sql, values)  conn.commit()  conn.close()  from selenium import webdriver  from bs4 import BeautifulSoup  import time  driver = webdriver.Chrome()  driver.get("https://www.ind.com/flights")  time.sleep(5) # Waiting unitl the data is loaded  soup = BeautifulSoup(driver.page\_source, "html.parser")  value = []  table = soup.select\_one("table.flight-airlines\_\_table")  for row in table.select("tbody"):  cols = row.find\_all("td")  row\_values = []  for col in cols:  text = col.text.strip()  combined = text  row\_values.append(combined)  value.append(row\_values)  value = value[0]  value = value[0::3]  driver.quit()  df = pd.DataFrame(value, columns=['AirlineName'])  AirlineCode = pd.Series(['AC', 'AS', 'G4', 'AA', 'DL', 'F9', 'WN', 'NK','SY', 'UA'])  df['AirlineCode'] = AirlineCode  df\_airlines = df  df\_airlines  import pymysql  conn = pymysql.connect(  host='localhost',  user='root',  password='root',  db='flight\_data',  port=8889,  charset='utf8mb4'  )  cursor = conn.cursor()  sql = """  INSERT INTO Airlines (AirlineCode, AirlineName)  VALUES (%s, %s)  """    values = df\_airlines[['AirlineCode', 'AirlineName']].values.tolist()  cursor.executemany(sql, values)  conn.commit()  conn.close()  import pymysql  conn = pymysql.connect(  host='localhost',  user='root',  password='root',  db='flight\_data',  port=8889,  charset='utf8mb4'  )  cursor = conn.cursor()  sql = "INSERT INTO Remarks (RemarkCode, RemarkName, UseYn) VALUES (%s, %s, %s)"  remark\_values = [  ('PRG', 'Progressing', 'Y'),  ('BRD', 'Boarding', 'Y'),  ('DPT', 'Departed', 'Y'),  ('ARR', 'Arrived', 'Y'),  ('DLT', 'Delayed', 'Y'),  ('CNL', 'Cancelled', 'Y'),  ('ONT', 'On Time', 'Y'),  ('ERL', 'Early', 'Y')  ]  cursor.executemany(sql, remark\_values)  conn.commit()  conn.close()  conn = pymysql.connect(  host='localhost',  user='root',  password='root',  db='flight\_data',  port=8889,  charset='utf8mb4'  )  cursor = conn.cursor()  sql\_roles = """  INSERT INTO Roles (RoleID, RoleName) VALUES  ('ADMIN', 'Administrator'),  ('OPS', 'Operations Manager'),  ('VIEWER', 'Viewer');  """  sql\_users = """  INSERT INTO Users (UserID, UserName, Password, AirportCode, AirlineCode) VALUES  ('user01', 'Alice Johnson', '5e884898da28047151d0e56f8dc62927', 'ATL', 'AA'),  ('user02', 'Bob Smith', '6cb75f652a9b52798eb6cf2201057c73', 'IND', 'DL'),  ('user03', 'Charlie Kim', '2b3a8d0a34e3e237c54b4cf2d80a3c41', 'IND', 'AA');  """  sql\_user\_roles = """  INSERT INTO UserRoles (UserRoleID, UserID, RoleID) VALUES  ('UR01', 'user01', 'ADMIN'),  ('UR02', 'user02', 'OPS'),  ('UR03', 'user03', 'VIEWER');  """  cursor.execute(sql\_roles)  cursor.execute(sql\_users)  cursor.execute(sql\_user\_roles)  conn.commit()  conn.close()  from selenium import webdriver  from bs4 import BeautifulSoup  import time  driver = webdriver.Chrome()  driver.get("https://www.ind.com/flights/flight-status/arrivals?flightNumber=&airline=all&arrdep=A")  time.sleep(5) # Waiting unitl the data is loaded  soup = BeautifulSoup(driver.page\_source, "html.parser")  theader = soup.select\_one("thead.flight-table\_\_header")  head = []  for row in theader.select("tr"):  cols = row.find\_all("th")  head = [col.text.strip() for col in cols]  value = []  tbody = soup.select\_one("tbody.flight-table\_\_body")  for row in tbody.select("tr.flight-table\_\_row"):  cols = row.find\_all("td")  row\_values = []  for col in cols:  text = col.text.strip()  img = col.find("img")  if img:  img\_src = 'https://www.ind.com'+img.get("src")  combined = f"{text} ({img\_src})" if text else img\_src  else:  combined = text  row\_values.append(combined)  value.append(row\_values)  driver.quit()  import pandas as pd  df = pd.DataFrame(value, columns=head)  df  # Preprocessing  from datetime import datetime  import numpy as np  status\_mapping = {  'Progressing': 'PRG',  'Boarding': 'BRD',  'Departed': 'DPT',  'Arrived': 'ARR',  'Delayed': 'DLT',  'Cancelled': 'CNL',  'On Time': 'ONT',  'Early': 'ERL'  }  df['FlightNumber'] = df['Flight #'].apply(lambda x: x.split(' ')[1])  df['AirlineCode'] = df['Flight #'].apply(lambda x: x.split(' ')[0])  df['AirportCode'] = 'IND' # Indianapolis  df['ScheduledDate'] = datetime.today().strftime('%Y%m%d')  df['ScheduledTime'] = df['ETA'].apply(  lambda x: datetime.strptime(x.strip().lower(), "%I:%M %p").strftime("%H%M"))  df['EstimatedDate'] = datetime.today().strftime('%Y%m%d')  df['EstimatedTime'] = df['ETA'].apply(  lambda x: datetime.strptime(x.strip().lower(), "%I:%M %p").strftime("%H%M"))  df['OriginDestAirport'] = np.random.choice(df\_airports['AirportCode'].dropna().unique(), size=len(df)) # Arbitrary data as airports data is not perfectly matching with those of df  #df['CheckinCounter'] =  df['BoardingGate'] = df['Gate #']  #df['BaggageClaimBelt'] =  df['Remarks'] = df['Status'].map(status\_mapping)  df  import pymysql  conn = pymysql.connect(  host='localhost',  user='root',  password='root',  db='flight\_data',  port=8889,  charset='utf8mb4'  )  cursor = conn.cursor()  # df = pd.read\_csv(f"flights\_arrival\_{df['AirportCode'][0]}.csv")  for \_, row in df.iterrows():  row = row.where(pd.notnull(row), None)  sql = """  INSERT INTO ActiveFlightSchedules (  FlightNumber, AirportCode, AirlineCode,  ScheduledDate, ScheduledTime,  EstimatedDate, EstimatedTime,  OriginDestAirport, Remarks  ) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)  """    values = (  row['FlightNumber'],  row['AirportCode'],  row['AirlineCode'],  row['ScheduledDate'],  row['ScheduledTime'],  row['EstimatedDate'],  row['EstimatedTime'],  row['OriginDestAirport'],  row['Remarks']  )    cursor.execute(sql, values)  conn.commit()  conn.close() |

1. **Overall Contribution Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Task** | **Contribution** | **AVR Time spent (hrs)** |
| **Soon-Hyuck Lee** | Conceptional Schema  Database  Code | Contributed to the database design by proposing schema ideas, drawing the ERD diagram, defining constraints, adding descriptive comments to each column, and preparing sample datasets for the schema. | 10 hours |
| **JiYeong Oh** | Database  Code | Enhanced database integrity by refining constraints and normalizing the tables, and built the database using datasets scrapped from the Indianapolis International Airport website. | 10 hours |