ADT Group-25
Final Project Part 2

AIRLINES CUSTOMER SATISFACTION DASHBOARD

Team Members

- Ayantika Nandi (*Team Lead*)
- Megha Nagabhushana Reddy
- Shahrukh Quraishi

1. Create Conceptual Diagram/Schema for database

The four primary entities in the database structure are Class, Customer_type, Travel_type, and Ratings. Different travel categories and flight classes are defined by the Travel_type and Class tables, respectively. Customer_type sorts customers into groups such as disloyal and loyal. The Ratings table is the primary location for customer feedback on a range of topics related to the flight experience, including satisfaction levels, demographic information, and particular flight characteristics like seat comfort and cleanliness. To ensure data integrity, foreign key restrictions create links between the Ratings table and the other entities. The construction of each table and inclusion of each constraint is credited to the author of that table, demonstrating collaborative database development.

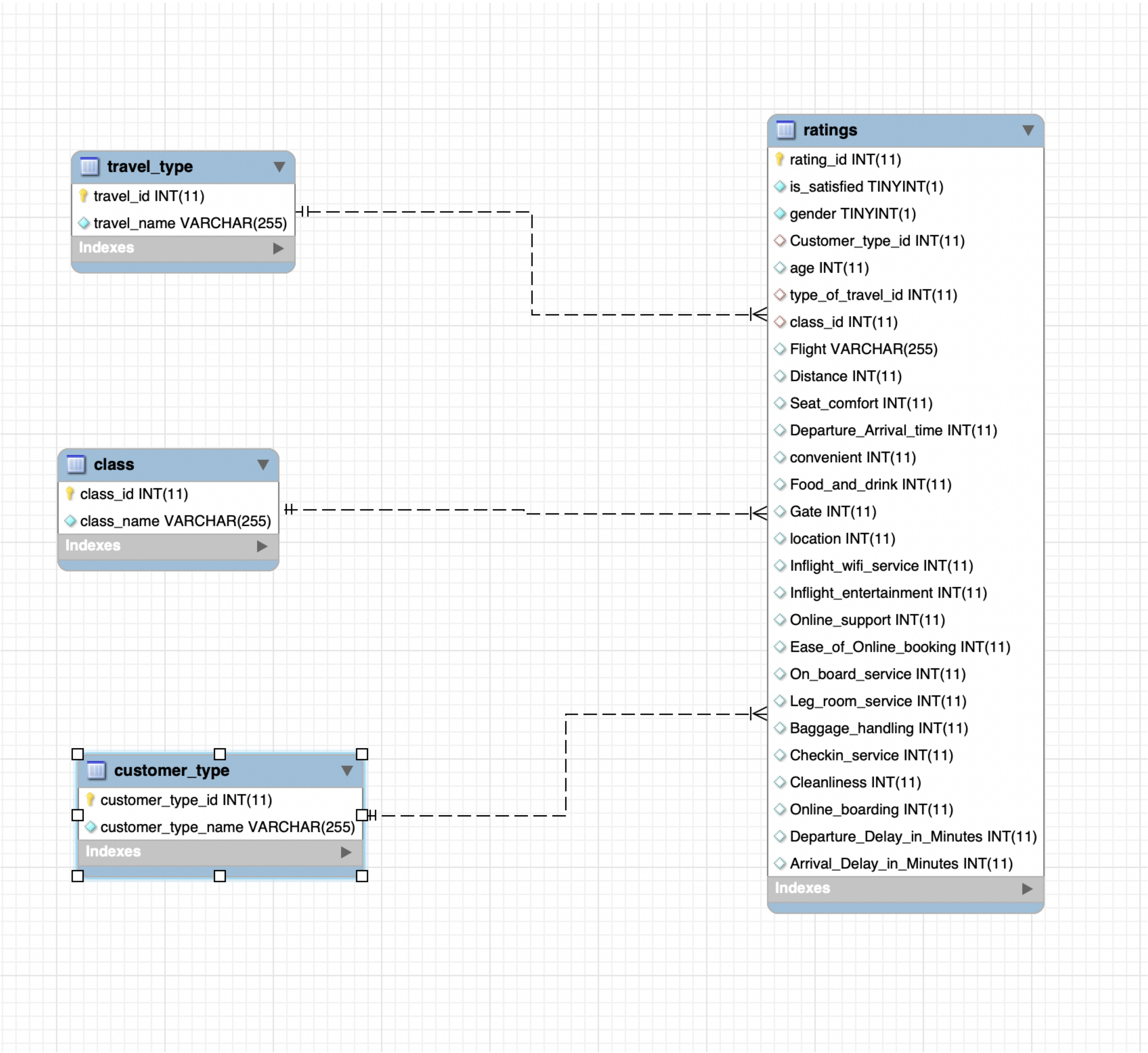

Fig 1: Schema for the database

2. Database Constraints

a. Travel_type table:

Explanation of Database Constraints:

1. ***Primary Key constraint:*** The *travel_id* column is designated as the primary key, ensuring every value is unique and not null. This is crucial for uniquely identifying each record, making data retrieval and relationships more efficient and reliable.

2. ***Auto-increment***: By setting the *travel_id* field to auto-increment, the database automatically assigns a unique, sequentially increasing number to each new record.
   This feature simplifies data insertion by eliminating the need to manually specify a unique identifier for each new row like the primary key, making it easier to handle and organize data.

3. ***Not Null constraint:*** Applying the NOT NULL constraint to the *travel_name* ensures that this important information is always provided when records are inserted or updated. This helps prevent data inconsistency and ensures that every travel type has an identifiable name, facilitating easier management and lookup of travel records.

4. ***Variable Character Field*** : The data type choice for *travel_name* accommodates strings of varying lengths, up to 255 characters, providing sufficient space for descriptive names while optimizing storage efficiency. VARCHAR is preferred for text data that can vary significantly in length, offering both flexibility and space efficiency.

Additional Constraints:

- *Enforcement of Formatting Standards:* Depending on the requirements, it might be necessary to enforce specific formats for the data entered into the database. For instance, ensuring all *travel_name* entries are in uppercase could involve application-level validation, database triggers, or CHECK constraints, providing consistency across the dataset.
- *Integration with Other Data Tables*: Should the *travel_type* table relate to other tables within the database, incorporating FOREIGN KEY constraints will be essential. These constraints enforce referential integrity, ensuring that all references between tables are valid and maintain the logical relationships between different data elements.

b. Class table:

Explanation of Database Constraints:

1. ***Primary Key:*** The *class_id* field serves as the table's primary key, making every value within it unique and not null.

2. ***Auto-increment:*** The auto-increment feature automatically assigns a unique, sequentially higher number to each new entry, eliminating the need for manual specification. This system is essential for distinctively identifying each class entry, streamlining database operations.
3. ***Not Null*** : By enforcing the *class_name* field to be non-nullable, the database ensures that every class entry is named, preventing the existence of unnamed or anonymous classes. This clarity aids in the straightforward identification and administration of class entries.

4. ***Unique*** : Imposing a uniqueness constraint on *class_name* guarantees that each name is one-of-a-kind within the table. This distinction prevents confusion by ensuring that each class can be identified solely by its name, thus preserving the database's clarity and data integrity.

5. ***Data Type for class_name***: Selecting VARCHAR(255) for the class_name allows for a broad range of names up to 255 characters, providing ample flexibility for naming while optimizing storage. This choice is optimal for fields where the length of the text can vary, offering a balance between versatility and space efficiency.

Additional Constraints:

- *Advanced Formatting and Constraints**:*** Tailoring the database to specific operational needs might involve applying additional constraints, such as enforcing particular formatting rules (e.g., capitalization of class names) or implementing CHECK constraints for complex validation rules (e.g., inclusion of certain keywords or adherence to a naming pattern) if needed.

- *Establishing Foreign Key constraints*: For databases where the class table interacts with others (e.g., tables for travel, customer, rating), incorporating FOREIGN KEY constraints is advisable. These links maintain the integrity and consistency of cross-table relationships, ensuring a cohesive and reliable database structure.

c. Customer table:

Primary Key Constraint(PK):

Customer_type_id is the primary key in this table, according to the Primary Key Constraint. It saves integer values, as indicated by the INT data type. AUTO_INCREMENT indicates that for every new record added to the table, MySQL creates a distinct value for this field automatically. For every new record, it increases by 1 from the beginning. We guarantee that every customer type item in the table has a distinct identity by designating customer_type_id as a primary key. This facilitates quick reference and identification of particular clientele.

Not Null Constraint:

The customer_type_name column must always have a value, according to the NOT NULL requirement. It makes sure that every customer type item has a name attached to it by preventing NULL values from being inserted into this column. By guaranteeing the constant provision of necessary information—in this case, the name of the customer type—this constraint contributes to the preservation of data integrity. It assists in preventing errors and inconsistent data while querying or modifying the data by prohibiting NULL values.

d. Rating table:

a. ***Primary key****:* In this table primary key is mainly used to represent an unique identifier for each rating row.

b. ***Foreign keys***: There are broadly three foreign keys in this table:

   i. *Customer_type_id:* This identifier signifies the customer type and acts as the primary key, directly linked to the customer type table, providing crucial insights into customer segmentation and preferences.

   ii. *Type_of_travel_id*: This foreign key represents the type of travel undertaken by each individual customer while submitting their rating. It references the travel type table, allowing for a detailed analysis of how different travel experiences impact customer satisfaction.

   iii. *Class_id*: Serving as the final foreign key in this table, the class_id indicates the specific class of travel chosen by passengers when providing their ratings. This key is associated with the class table, enabling a comprehensive understanding of customer preferences across different travel classes.

c. ***Additional rating columns***: Moreover, the rating table encompasses various other user-provided ratings, each with a one-to-one mapping that cannot be further normalized due to their unique nature.

d. ***On-delete cascade behavior***: Concerning the on-delete cascade functionality applied to customer type, class, and travel type entries in the rating table, the associated entries will be automatically deleted according to the specified constraints.

3. <u>Write code to create a database and build queries.</u>

a) <u>Table Creation:</u>

Travel_type table:

```
--Author: Ayantika Nandi

CREATE TABLE travel_type (
   travel_id INT AUTO_INCREMENT PRIMARY KEY,
   travel_name VARCHAR(255) NOT NULL
);
```

Class table:

```
-- Author: Ayantika Nandi

CREATE TABLE class (
   class_id INT AUTO_INCREMENT PRIMARY KEY,
   class_name VARCHAR(255) NOT NULL UNIQUE
);
```

Customer table:

```
-- Author: Megha Nagabhushana Reddy

CREATE TABLE customer_type (
   customer_type_id INT AUTO_INCREMENT PRIMARY KEY,
   customer_type_name VARCHAR(255) NOT NULL
);
```

Rating table:

```
-- Author: Shahrukh Quraishi

CREATE TABLE ratings (
   rating_id INT PRIMARY KEY AUTO_INCREMENT,
   is_satisfied TINYINT(1) NOT NULL,
   gender TINYINT(1) NOT NULL,
   Customer_type_id INT,
   age INT,
   type_of_travel_id INT,
   class_id INT,
   Flight VARCHAR(255),
   Distance INT,
   Seat_comfort INT,
   Departure_Arrival_time INT,
   convenient INT,
   Food_and_drink INT,
   Gate INT,
   location INT,
   Inflight_wifi_service INT,
   Inflight_entertainment INT,
   Online_support INT,
   Ease_of_Online_booking INT,
   On_board_service INT,
   Leg_room_service INT,
   Baggage_handling INT,
   Checkin_service INT,
   Cleanliness INT,
   Online_boarding INT,
   Departure_Delay_in_Minutes INT,
   Arrival_Delay_in_Minutes INT);
```

b) Data Insertion:

Travel_type table:

```
-- Author: Ayantika Nandi

-- Adding 'Personal Travel' as a travel type
INSERT INTO travel_type (travel_name) VALUES ('Personal Travel');

-- Adding 'Business travel' as a travel type
INSERT INTO travel_type (travel_name) VALUES ('Business travel');
```

Class table:

```
-- Author: Ayantika Nandi

-- Inserting 'Eco' as a class type
```

```
INSERT INTO class (class_name) VALUES ('Eco');

-- Inserting 'Business' as a class type
INSERT INTO class (class_name) VALUES ('Business');

-- Inserting 'Eco Plus' as a class type
INSERT INTO class (class_name) VALUES ('Eco Plus');
```

Customer_type table:

```
-- Author: Megha Nagabhushana Reddy

INSERT INTO customer_type(customer_type_name) VALUES ('Loyal Customer');

INSERT INTO customer_type(customer_type_name) VALUES ('disloyal Customer');
```

Ratings table:

```
-- Author: Shahrukh Quraishi

– insert into ratings one row example

INSERT INTO ratings (
    is_satisfied,
    gender,
    Customer_type_id,
    age,
    type_of_travel_id,
    class_id,
    Flight,
    Distance,
    Seat_comfort,
    Departure_Arrival_time,
    convenient,
    Food_and_drink,
    Gate,
    location,
    Inflight_wifi_service,
    Inflight_entertainment,
    Online_support,
    Ease_of_Online_booking,
    On_board_service,
    Leg_room_service,
    Baggage_handling,
    Checkin_service,
    Cleanliness,
    Online_boarding,
    Departure_Delay_in_Minutes,
    Arrival_Delay_in_Minutes
) VALUES (
    1, 1, 1, 30, 1, 1, 'ABC123', 1000, 4, 3, 5, 4, 2, 1, 3, 4, 5, 4, 5, 3, 4, 5, 15, 10, 5, 1
);
```

c) Foreign key constraints:

```
-- Author: Megha Nagabhushana Reddy

#Foreign key constraint for Customer_type_id column:
ALTER TABLE ratings
ADD CONSTRAINT fk_customer_type_id
FOREIGN KEY (Customer_type_id)
REFERENCES customer_type(customer_type_id)
ON DELETE CASCADE;
```

```
#Foreign key constraint for type_of_travel_id column:
ALTER TABLE ratings
ADD CONSTRAINT fk_type_of_travel_id
FOREIGN KEY (type_of_travel_id)
REFERENCES travel_type(travel_id)
ON DELETE CASCADE;


#Foreign key constraint for class_id column:
ALTER TABLE ratings
ADD CONSTRAINT fk_class_id
FOREIGN KEY (class_id)
REFERENCES class(class_id)
ON DELETE CASCADE;
```

4. Overall Contribution Summary:

Name	**Task**	**Contribution**	**Average Time Spent (hrs)**
Ayantika Nandi	Conceptual Schema, Database, Table Creation, Data Insertion	Developed the database schema, designed the Travel_type and Class tables, defined database constraints, and managed data insertion for these tables.	**~10**
Megha Nagabhushana Reddy	Table creation, Data Insertion, Foreign key constraints	Authored the customer_type table, performed data insertion, and established foreign	**~10**

		key constraints and developed the database schema,	
Shahrukh Quraishi	Table creation, Data Insertion	Created the ratings table, managed data insertion and developed the database schema,	**~10**