411Gangsters (Ganotisi, Sethi, Gentela, Mangel)

Abdussalam Alawini

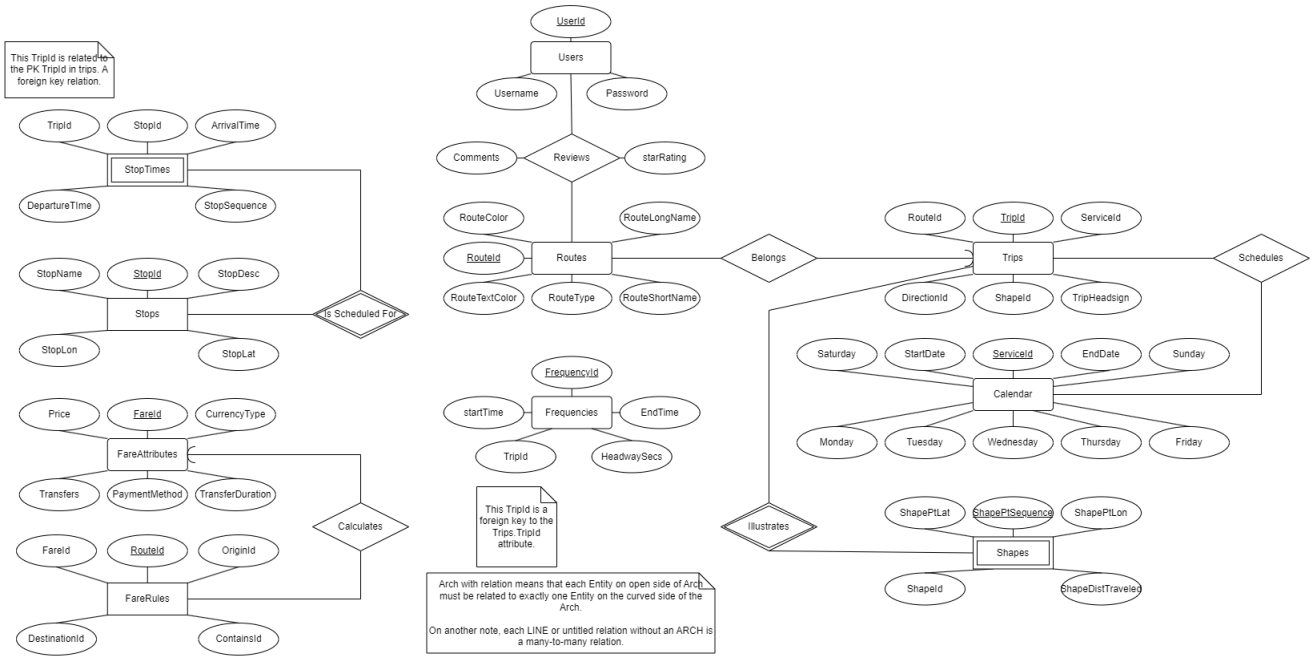
CS 411: Database Systems

October 02, 2023

Conceptual and Logical Design

For this project, our group decided to create an ER Diagram instead of a UML diagram because, with a database as huge as ours, we want to have an overview and an easy outline to remember specific correlations and connections between entities.

ER DIAGRAM



This database schema is already normalized in Third Normal Form, also known as 3NF, which follows the following rules: (1) There is no primary attribute dependency: Every non-prime attribute must be non-transitively dependent on every candidate key in the table; (2) There are no transitive dependencies, where a non-prime attribute depends on another non-prime attribute. The following shows how every table in our schema is 3NF:

1. **Users** have the following functional dependencies: UserId -> UserName, UserId -> Password, we have UserId as a super key, which covers (1) and (2).
2. **Routes** have the following functional dependencies: RouteId -> RouteShortName, RouteId -> RouteLongName, RouteId -> RouteType, RouteId -> RouteColor, RouteId -> RouteTextColor, which then implies RouteId is a super key for all of these attributes, which covers (1) and (2).
3. **Trips** have the following functional dependencies: TripId, ShapeId -> RouteId, TripId, ShapeId -> ServiceId, TripId, ShapeId -> TripHeadsign, TripId, ShapeId -> DirectionId. This implies that TripId, ShapeId is a super key for all of these attributes which covers (1) and (2).
4. **Calendar** has the following functional dependencies: ServiceId -> Monday, ServiceId -> Tuesday, ServiceId -> Wednesday, ServiceId -> Thursday, ServiceId -> Friday, ServiceId -> Saturday, ServiceId -> Sunday, ServiceId -> StartDate, ServiceId -> EndDate. This implies that ServiceId is a super key for all of these attributes, which covers (1) and (2).
5. **Shapes** have the following functional dependencies: ShapeId, ShapePtSequence -> ShapePtLat, ShapeId, ShapePtSequence -> ShapePtLon, ShapeId, ShapePtSequence -> ShapeDistTraveled. This implies that the combination of (ShapeId, ShapePtSequence) is a super key for all of these attributes, which covers (1) and (2).
6. **Frequencies** have the following functional dependencies: FrequencyId -> TripId, FrequencyId -> StartTime, FrequencyId -> EndTime, and FrequencyId -> HeadwaySecs. This implies that (FrequencyId) is a super key for all of these attributes, which covers (1) and (2).

7. **Stops** have the following functional dependencies: (StopId) -> (StopName), (StopId) -> (StopDesc), (StopId) -> (StopLon) and (StopId) -> (StopLat). This implies that (StopId) is a super key for all of these attributes, which covers (1) and (2).
8. **StopTimes** has the following functional dependencies: (StopId, TripId) -> (ArrivalTime), (StopId, TripId) -> (DepartureTime), and (StopId, TripId) -> (StopSequence). This implies that (StopId, TripId) is a super key for all of these attributes, which covers (1) and (2).
9. **FareRules** has the following functional dependencies: (RouteId) -> (FareId),(RouteId) -> (OriginId),(RouteId) -> (DestinationId) and(RouteId) -> (ContainsId). This implies that (RouteId) is a super key for all of these attributes, which covers (1) and (2).
10. **FareAttributes** has the following functional dependencies:(FareId) -> (Price),(FareId) -> (CurrencyType),(FareId) -> (PaymentMethod),(FareId) -> (Transfers) and(FareId) -> (TransferDuration). This implies that (FareAttributes) is a super key for all of these attributes, which covers (1) and (2).
11. **Reviews** has the following functional dependencies:(UserId, ReviewId) -> (Comments),(UserId, ReviewId) -> (starRating). This implies that (Reviews) is a super key for all of these attributes, which covers (1) and (2).

Relational Schema. The following is our relational schema for this database and it follows the following rule: (1) It is not a DDL, meaning that we won't be creating tables of some sort;

Users(UserId: INT [PK],
UserName: VARCHAR(100),
Password: VARCHAR(255)
)

Routes(RouteId: VARCHAR(7) [PK],
RouteShortName: VARCHAR(7).

RouteLongName: VARCHAR(255),

RouteType: INT,

RouteColor: VARCHAR(6),

RouteTextColor: VARCHAR(6)

)

Trips(TripId: VARCHAR(10) [PK],

RouteId: VARCHAR(7) [FK to Route.RouteId],

ServiceId: VARCHAR(3),

TripHeadsign: VARCHAR(255),

DirectionId: INT,

ShapeId: INT [PK]

)

Calendar(ServiceId: VARCHAR(3) [PK], Monday: INT, Tuesday: INT, Wednesday: INT, Thursday: INT, Friday: INT, Saturday: INT, Sunday: INT, StartDate: VARCHAR(10), EndDate: VARCHAR(10))

Shapes(ShapeId: INT [PK],

ShapePtLat: REAL,

ShapePtLon: REAL,

ShapePtSequence: INT [PK],

ShapeDistTraveled: REAL)

Frequencies (

FrequencyId: INT [PK]

TripId: VARCHAR(10) [FK to Trips.tripId],

StartTime: VARCHAR(8),

EndTime: VARCHAR(8),

HeadwaySecs: INT

)

Stops (StopId: INT [PK],

StopName: VARCHAR(255),

StopDesc:VARCHAR(255),

StopLon: REAL,

StopLat: REAL

)

StopTimes (StopId: INT [FK to Stops.StopId],

TripId: INT VARCHAR(10) [FK to Trips.TripId],

ArrivalTime: VARCHAR(8),

DepartureTime: VARCHAR(8),

StopSequence: INT

)

FareRules(RouteId: VARCHAR(255) [PK],

FareId: VARCHAR(255) [FK to FareAttributes.FareId],

OriginId: VARCHAR(255),

DestinationId: VARCHAR(255),

ContainsId: VARCHAR(255)

)

FareAttributes (FareId: VARCHAR(255) [PK],

Price: REAL,

CurrencyType: VARCHAR(3),

PaymentMethod: INT,

Transfers: VARCHAR(255),

TransferDuration: INT

)

Reviews (ReviewId: INT [PK],

UserId: INT [FK to Users.UserId],

Comments: VARCHAR(1000),

starRating: INT

)

Description and Assumptions of each Entity and Relation.

Users:

Description: This entity represents the users of the system, such as employees or administrators who might need access to the transportation data.

Assumptions:

- Each user has a unique UserId as the primary key.
- Users have a UserName (username) and a Password for authentication.

Routes:

Description: This entity stores information about different transportation routes.

Assumptions:

- A unique RouteId identifies each route.
- It includes details like RouteShortName, RouteLongName, RouteType, RouteColor, and RouteTextColor.

Trips:

Description: This entity represents individual trips made on specific routes.

Assumptions:

- Each trip has a unique TripId as the primary key.
- The RouteId is a foreign key referring to the Routes table.
- It contains information like ServiceId, TripHeadsign, DirectionId, and ShapeId.

Calendar:

Description: This entity stores information about service schedules on specific days.

Assumptions:

- The ServiceId uniquely identifies a service schedule.
- The fields like Monday, Tuesday, etc., represent binary indicators (0 or 1) for the service availability on respective days.
- StartDate and EndDate indicate the period during which the service operates.

Shapes:

Description: This entity represents the shape of routes.

Assumptions:

- Each shape has a unique ShapeId as the primary key.
- ShapePtLat and ShapePtLon store latitude and longitude coordinates for points along the shape.
- ShapePtSequence indicates the sequence of points, and ShapeDistTraveled represents the distance traveled along the shape.

Frequencies:

Description: This entity represents frequency information for trips.

Assumptions:

- The combination of TripId and StartTime uniquely identifies each frequency.
- It includes fields such as EndTime and HeadwaySecs.

Stops:

Description: This entity stores information about transportation stops.

Assumptions:

- A unique StopId identifies each stop as the primary key.
- Details include StopName, StopDesc, StopLon, and StopLat.

StopTimes:

Description: This entity represents the times when trips arrive and depart from stops.

Assumptions:

- The combination of StopId, TripId, and ArrivalTime uniquely identifies each stop time.
- It includes fields such as DepartureTime and StopSequence.

FareRules:

Description: This entity defines fare rules and conditions.

Assumptions:

- A unique FareId identifies each fare rule as the primary key.
- It includes fields like RouteId, OriginId, DestinationId, and ContainsId for specifying fare conditions.

FareAttributes:

Description: This entity stores fare-related information.

Assumptions:

- A unique FareId identifies each fare attribute as the primary key.
- It contains details like Price, CurrencyType, PaymentMethod, Transfers, and TransferDuration for fares.

Certainly, here is the description and assumption for the new table "Reviews":

Reviews:

Description: This entity represents user reviews for the transportation system or related services. It allows users to provide comments and star ratings to share their feedback and experiences.

Assumptions:

- Each review can have a text comment, which is stored in the `Comments` column with a maximum length of 1000 characters.
- Users can rate their experience using the `starRating` column, where higher values indicate better ratings (e.g., 5 stars for excellent and 1 star for poor).
- The ER diagram does not include a specific ID for the relationship table Reviews, but it in a real-world scenario assumes that each review is associated with a UserId and a specific ReviewId as shown in the relationship schema.