

Tony Nguyen

Dr. Shawn Bowers

CPSC 321 01

28 November 2023

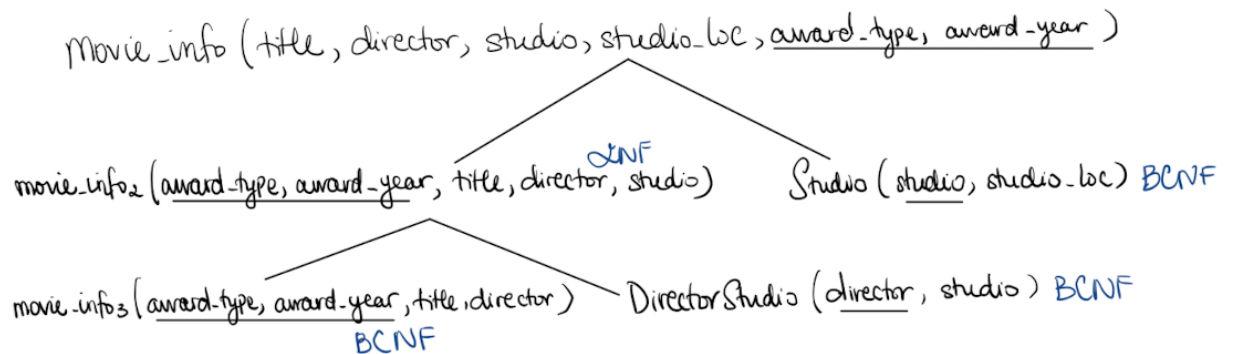
Homework 7

Part 1

1. Given the original schema and the functional dependencies (FDs), the primary key for the original schema is (award_type, award_year). From there, we can have the following information about the decomposed tables:

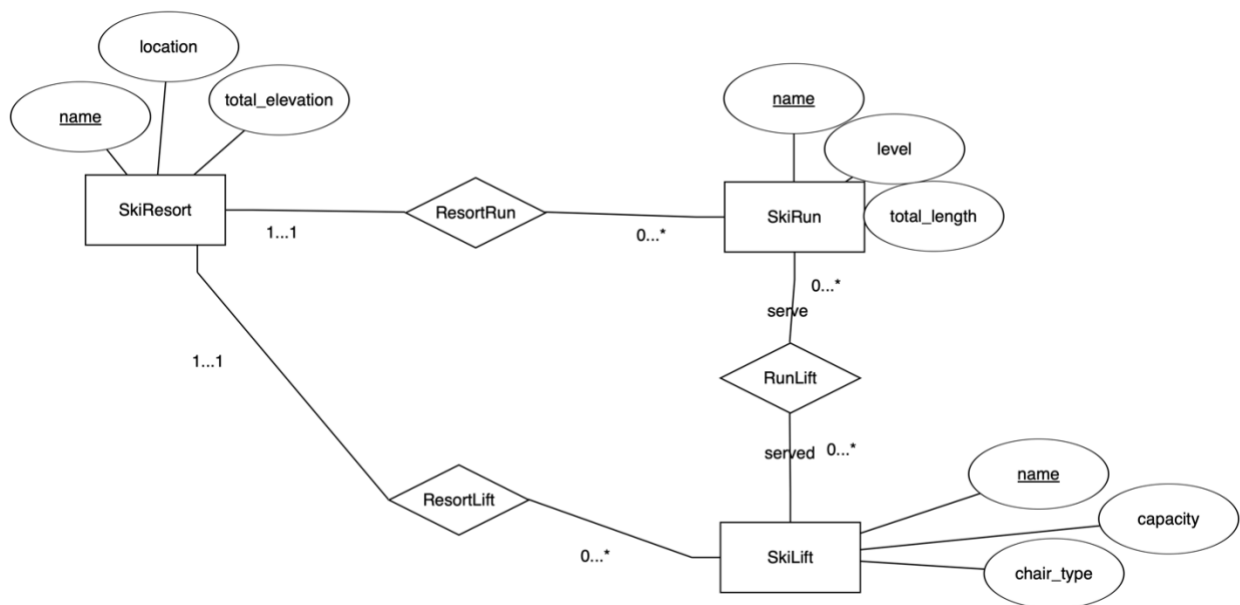
- movie_info₁(director, studio, studio_loc)
 - Candidate key: director.
 - Highest normal form: 2NF since we have a transitive dependency of studio → studio_loc.
- movie_info₂(title, director, studio)
 - Candidate key: (title, director).
 - Highest normal form: 1NF since we have a partial key dependency of director → studio.
- movie_info₃(award_type, award_year, title)
 - Candidate key: (award_type, award_year).
 - Highest normal form: BCNF since there is no redundancy caused by FDs.
- movie_info₄(award_type, award_year, studio_loc)
 - Candidate key: (award_type, award_year)
 - Highest normal form: BCNF, as there is no bad FD, thanks to Transitivity.

2. Schema decomposition using the BCNF algorithm.



Part 2

3. Draw an ERD for the ski resort.

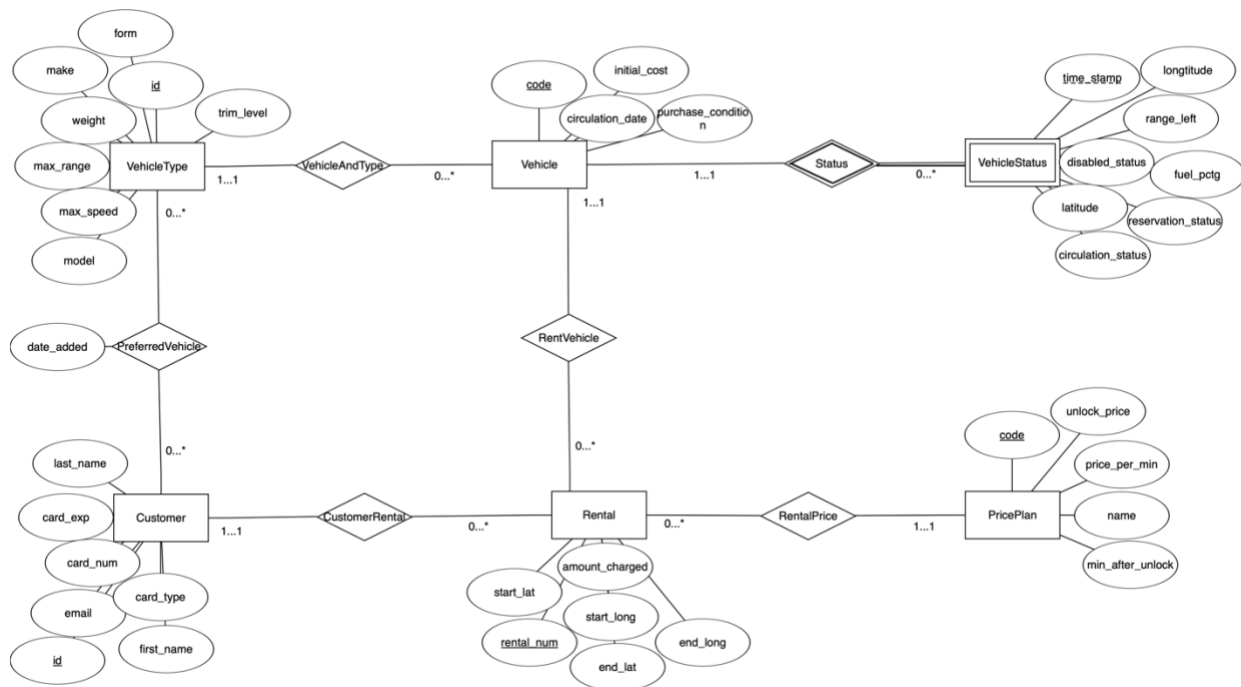


4. Database schema translation for Question 3.

- SkiResort(name, location, total_elevation).
- SkiRun(name, level, total_length).
- SkiLift(name, capacity, chair_type).
- ResortRun(resort_name, run_name).

- ResortRun.run_name → SkiRun.name.
- ResortRun.resort_name → SkiResort.name.
- ResortLift(resort_name, lift_name).
 - ResortLift.resort_name → SkiResort.name.
 - ResortLift.lift_name → SkiLift.name.
- RunLift(run_name, lift_name).
 - Note: Only contains SkiRun-SkiLift pairs that are connected.
 - RunLift.lift_name → SkiLift.name.
 - RunLift.run_name → SkiRun.name.

5. Create an ERD for the micromobility vehicles.



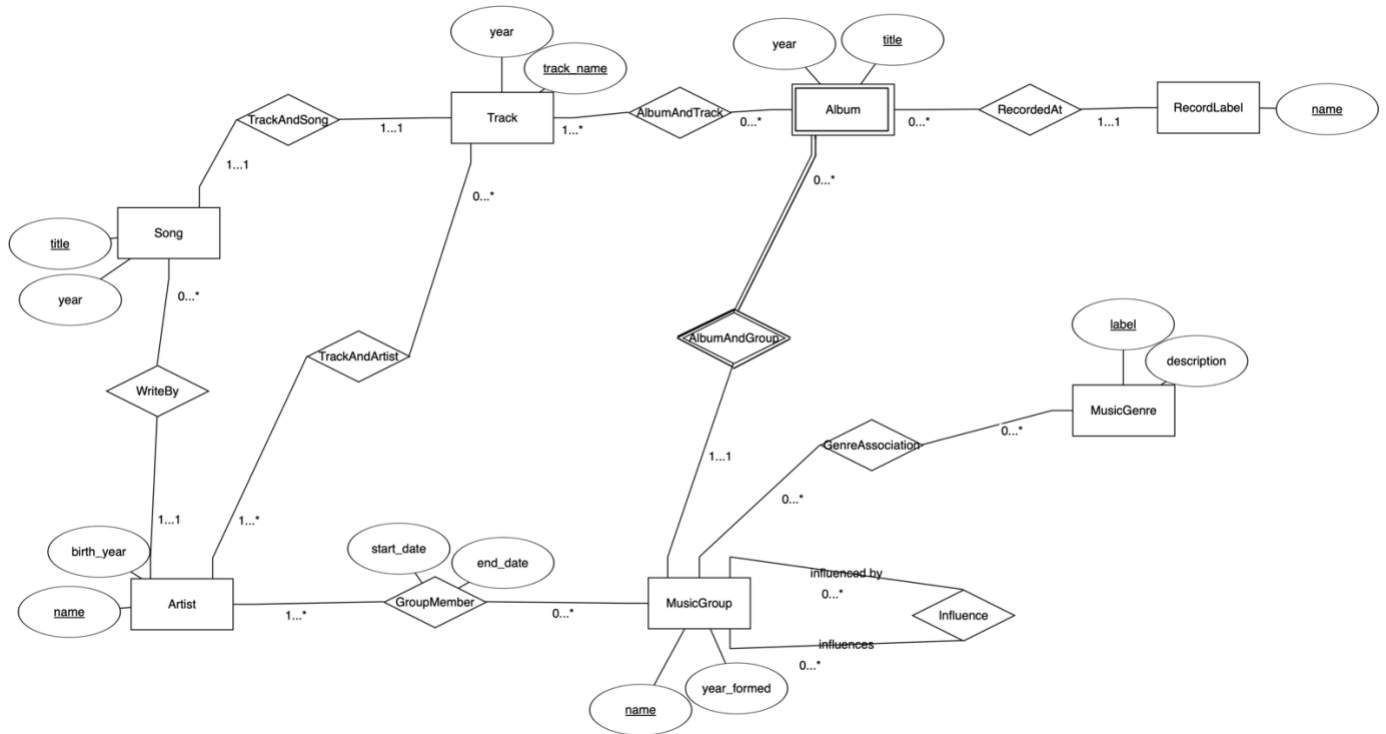
6. Database schema translation for Question 5.

- VehicleType(id, form, make, trim_level, weight, max_range, max_speed, model).
- Vehicle(code, circulation_date, initial_cost, purchase_condition).

- VehicleStatus(vehicle_code, time_stamp, circulation_status, reservation_status, fuel_pctg, disabled_status, latitude, longitude, range_left).
 - Note: Only Vehicle that has a VehicleStatus will be on this table.
 - VehicleStatus.vehicle_code → Vehicle.code.
- Customer(id, first_name, last_name, email, card_type, card_num, card_exp).
- Rental(rental_num, start_lat, start_long, end_lat, end_long, amount_charged).
- PricePlan(code, name, unlock_price, price_per_min, min_after_unlock).
- VehicleAndVehicleType(vehicle_code, vehicle_type_id).
 - VehicleAndVehicleType.vehicle_code → Vehicle.code.
 - VehicleAndVehicleType.vehicle_type_id → VehicleType.id.
- PreferredVehicle(c_id, vt_id, date_added).
 - PreferredVehicle.c_id → Customer.id.
 - PreferredVehicle.vt_id → VehicleType.id.
- CustomerRental(c_id, rental_num).
 - Note: Only Customer who rents a vehicle will be on this table.
 - CustomerRental.c_id → Customer.id.
 - CustomerRental.rental_num → Rental.rental_num.
- RentVehicle(rental_num, v_code).
 - Note: Only rented vehicles will be on this table.
 - RentVehicle.rental_num → Rental.rental_num.
 - RentVehicle.v_code → Vehicle.code.
- RentalPrice(rent_num, p_code).
 - Note: Only price plans that have been used by a rental will be on the table.

- RentalPrice.rent_num → Rental.rent_num.
- RentalPrice.p_code → PricePlan.code.

7. Create an ERD for HW2 Question 1.



8. Database relation translation for Question 7.

- Song(title, track_id, year).
 - Song.track_id → Track.track_name.
- Artist(name, birth_year).
- Track(track_name, year).
- MusicGroup(name, year_formed).
- Album(title, year, group, record_label).
 - Album.group → MusicGroup.name.
 - Album.record_label → RecordLabel.name.
- RecordLabel(name).

- MusicGenre(label, description).
- SongAndArtist(song_title, artist_name).
 - SongAndArtist.song_title → Song.title.
 - SongAndArtist.artist_name → Artist.name.
- TrackAndArtist(track_name, artist_name).
 - TrackAndArtist.track_name → Track.track_name.
 - TrackAndArtist.artist_name → Artist.name.
- AlbumAndTrack(album_title, track_name, group).
 - AlbumAndTrack.album_title → Album.title.
 - AlbumAndTrack.track_name → Track.track_name.
 - AlbumAndTrack.group → Music.group.
- GroupAndArtist(group_name, artist_name, start_date, end_date).
 - GroupAndArtist.group_name → MusicGroup.name
 - GroupAndArtist.artist_name → Artist.name.
- GroupAndGenre(group_name, genre).
 - GroupAndGenre.group_name → MusicGroup.name.
 - GroupAndGenre.genre → MusicGenre.label.
- Influence(group_name, influenced_by).
 - Influence.group_name → MusicGroup.name.
 - Influence.influenced_by → MusicGroup.name.
- Compared with HW2, the only change I made was to implement the TrackAndArtist schema. I did so as there is a possibility that the artist(s) who perform a given track is not the one(s) who wrote the associated song.