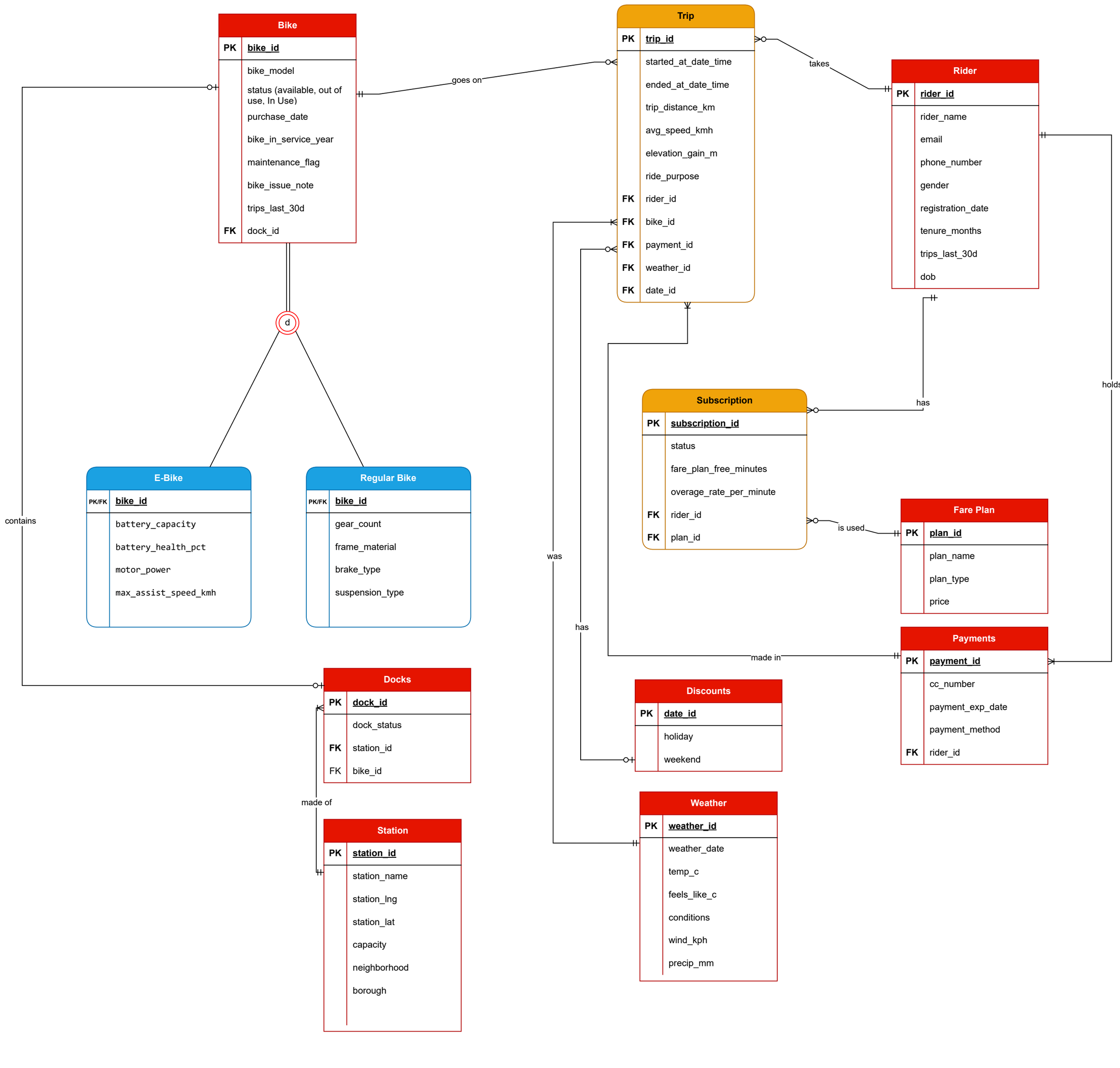
OPIM 5272 - 712 - Project Deliverable 2 (GROUP 3)



Reports:

1) Station Usage Report

What it shows: How many bikes go out and come back at each station every day.

Why it's useful: Helps the company see which stations are too busy, which ones don't have enough bikes, and where to add more docks or move bikes.

Entities used:

- **Trip** (attributes: `started_at`, `ended_at`, `start_station`, `end_station`)
- **Station** (attributes: `station_id`, `station_name`)

  Most important attributes: start station, end station, and trip time.

2) Popular Routes Report

What it shows: The most common start and end points for trips, along with how long and how far those trips usually are.

Why it's useful: Shows the main travel patterns between neighborhoods and helps plan for rush hours or improve bike lanes.

Entities used:

- **Trip** (attributes: `start_station`, `end_station`, `trip_duration_sec`, `trip_distance_km`)
- **Station** (attributes: `station_id`, `station_name`)

  Most important attributes: start station, end station, trip distance, trip duration.

3) Weather Impact Report

What it shows: How the number of trips changes when the weather is sunny, rainy, or cold, and how trip times are affected.

Why it's useful: Helps the company prepare for slow days during bad weather, plan staffing, and ensure bike safety.

Entities used:

- **Trip** (attributes: `trip_id`, `started_at`, `trip_duration_sec`)
- **Weather** (attributes: `weather_condition`, `temperature`, `precipitation`)

  Most important attributes: weather condition, trip date, trip duration.

Relationships (with Example Data):

- **Station–Dock (1:M):** One station has many docks; each dock belongs to one station.

  Example: `Union Square Station` has multiple docks like `Dock-001`, `Dock-002`.
- **Dock–Bike (1:0|1 ↔ 1:0|1):** One dock can hold one bike at a time; one bike can be in one dock at a time.

  *Example: On **2024-07-02**, Bike `Classic 2` (in service since 2022) was parked at `Union Square Dock-001`.*
- **Rider–Trip (1:M):** A rider can take many trips; each trip has one rider.

  *Example: A **member rider** took multiple trips, including Trip on `2024-07-02 08:12–08:27` from Union Square → Chelsea.*
- **Bike–Trip (1:M):** A bike can be used in many trips; each trip uses one bike.

  Example: Bike `E-Bike` (in service since 2023) was used in Trip on `2024-07-02 17:41–18:05` from Chelsea → Tribeca.
- **Trip–Payment (M:1):** Each trip is paid with one method; one method can pay many trips.

  Example: The Trip on `2024-07-03 12:30–12:45` (Bike `Classic 2`) was paid with stored card `PM11`.
- **Rider–Payment (1:M):** A rider can store many payment methods; each method belongs to one rider.

  *Example: The **casual rider** who took the `2024-07-04 19:14–19:42` trip (Bike `Classic 1`) has both a Visa and an Amex saved.*
- **Rider–Default Payment (M:1):** A rider may choose one default method.

  *Example: The **member rider** from the `2024-07-05 07:58–08:20` trip has Visa `****4242` set as default.*
- **Rider–Subscription (1:M):** A rider can have multiple subscriptions over time; each subscription belongs to one rider.

  Example: A rider on `2024-07-06 11:14–11:35` trip holds an Active Monthly 60 Plan.
- **Fare Plan–Subscription (1:M):** A fare plan can apply to many subscriptions; each subscription selects one plan.

  Example: `Monthly 60 Plan` is used by subscriptions tied to riders of trips like Chelsea → Tribeca and Union Square → Midtown East.
- **Trip–Weather (M:1):** Each trip has one weather record; one weather record applies to many trips.

  *Example: On `2024-07-02 (08:12 trip)`, weather was **Cloudy with 12m elevation gain**.*
- **Trip–Discounts/Date (M:1):** Each trip occurs on one date row; a date can have many trips.

  Example: All trips on `2024-07-02` (like Union Square → Chelsea, Chelsea → Tribeca) reference the same calendar date.
- **Bike–E-Bike (1:0|1):** A bike may appear as one E-Bike (shares same `bike_id`).

  *Example: The `2024-07-04 10:05–10:31` trip used an **E-Bike** (battery-powered, 2024 model).*
- **Bike–Regular Bike (1:0|1):** A bike may appear as one Regular Bike (shares same `bike_id`).

  Example: The `2024-07-03 09:05–09:18` trip used `Classic 2` (regular bike, 2021 model).

Example of Data - Entity: Trip

trip_id	started_at_date_time	ended_at_date_time	trip_distance_km	avg_speed_kmh	elevation_gain_m	ride_purpose	*rider_id*	*bike_id*	*payment_id*	*weather_id*	*date_id*
P501	7/2/2024 8:12:15	7/2/2024 8:27:45	**3.5**	14	12	Commute	*R123*	*B007*	*P501*	*W901*	*2024/07/02*
T1002	7/2/2024 17:41:03	7/2/2024 18:05:35	**5**	12	8	Leisure	*R145*	*B021*	*P502*	*W901*	*2024/06/03*
T1003	7/3/2024 9:05:10	7/3/2024 9:18:42	**7**	7.8	15	Fitness	*R166*	*B042*	*P503*	*W902*	*2024/05/08*
T1004	7/3/2024 12:30:22	7/3/2024 12:45:51	**2.7**	8.7	9	Commute	*R123*	*B019*	*P504*	*W903*	*2024/02/01*
T1005	7/4/2024 10:05:11	7/4/2024 10:31:40	3.8	4	1	Work	*R124*	*B007*	*P505*	*W904*	*2024/09/02*

Background of the Project:

- The main goal of this project was to design a relational database for a bike-share system that has the ability to manage trips, stations, riders, bikes, and payments. Large amounts of data being collected has generated a need to be able to capture and organize this data into a database. By modeling different relationships (i.e. Station to Dock, Rider to Trip, and Trip to Weather), we are able to have a database that connects the information that would otherwise be scattered and inconsistent. The organization and tracking that the database provides allows us to answer important business questions such as which stations continue to run out of bikes, which routes are the most popular, and how variations in the weather can have an impact on how many people ride on certain days. If the company did not have this database, they would have many inefficiencies duplicated records, and have trouble making accurate, data driven decisions.

- We wanted to have the ability to create a database that provides useful insights to improve data driven decision making for the bike share company. Reports such as Station Usage show where more docks or bikes are needed, Popular Routes indicate rider travel patterns across neighborhoods, and Weather Impact highlights how conditions affect trip volume and duration. By having the relationships built directly into the database, such as connecting trips to riders, bikes, and payments, allow for flexible reporting and allow us to analyze even more over time. It also helps reduce redundancy, improve accuracy, and gives the company a tool to better allocate resources, manage demand, and enhance the rider experience.