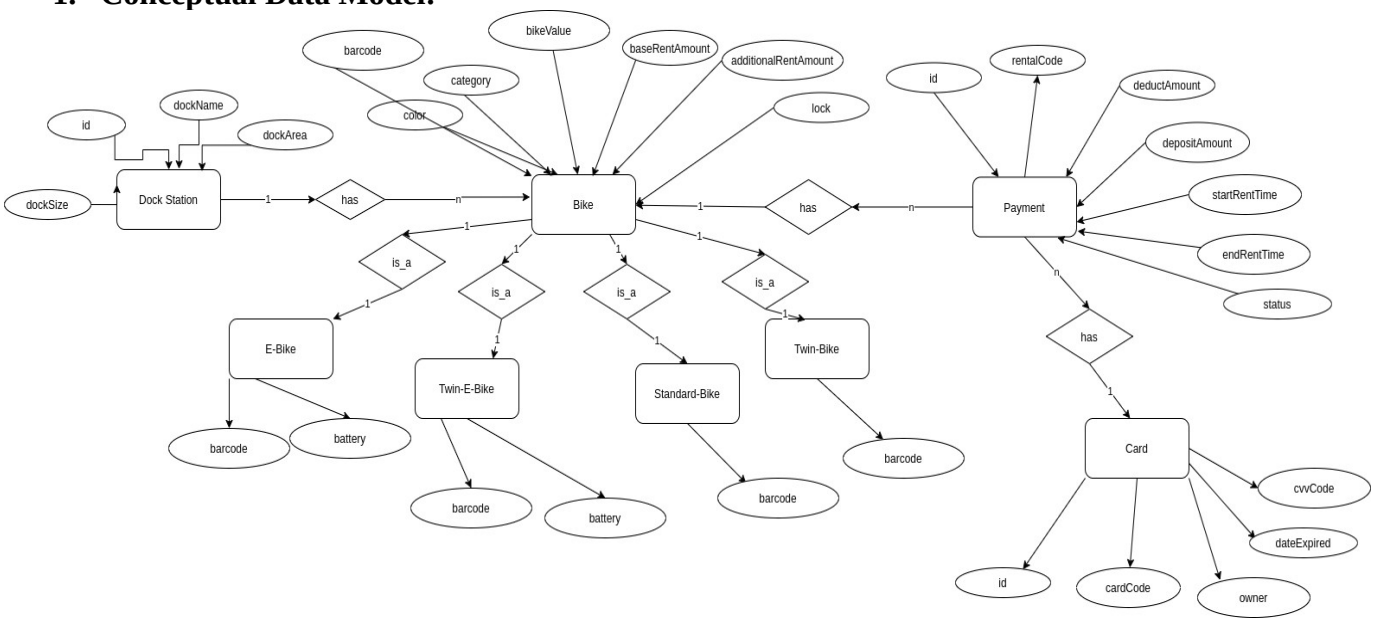
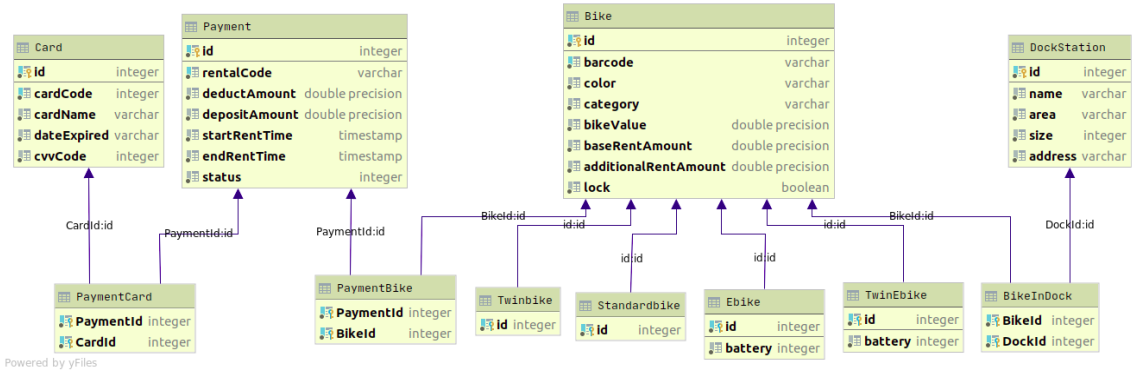
Data Modeling

1. Conceptual Data Model:



2. Logical Data Model:



3. Physical Data Model:

- **Payment**

#	PK	FK	Column Name	Data Type	Mandatory	Description
1	x		id	serial	yes	Payment id
2			rentalCode	Integer	yes	Rental code
3			deductAmount	float	Yes	Deduct amount
4			DepositAmount	Float	yes	Deposit amount
5			startRentTime	TIMESTAMP	yes	Starting rent time
6			endRentTime	TIMESTAMP	yes	Ending rent time
7			Status	integer	yes	Status of transaction

- **Payment Bike**

#	PK	FK	Column Name	Data Type	Mandatory	Description

1		x	PaymentId	Serial	Yes	Payment id
2		x	BikeId	Serial	Yes	Bike id

- DockStation

#	PK	FK	Column Name	Data Type	Mandatory	Description
1.	x		Id	Integer	Yes	ID, auto increment
2.			Name	VARCHAR	Yes	Name of dock
3.			area	VARCHAR	Yes	Area of the dock
4.			size	Integer	Yes	Max size of dock
5.			Address	VARCHAR	Yes	Address of dock

- BikeInDock

#	PK	FK	Column Name	Data Type	Mandatory	Description
1.	x	x	BikeId	Integer	Yes	Bike Id
2.	x	x	DockId	Integer	Yes	Dock Id

- Card

#	PK	FK	Column Name	Data Type	Mandatory	Description
1.	x		id	Integer	Yes	ID
2.			cardCode	VARCHAR	Yes	Card’s Code
3.			cardName	VARCHAR	Yes	Name of the card’s owner
4.			cvvCode	INT	Yes	CVV Code of the Card
5.			dateExpired	VARCHAR	Yes	Card’s Expiration Date

- PaymentCard

#	PK	FK	Column Name	Data Type	Mandatory	Description
1.	X	X	PaymentID	Integer	Yes	Payment ID
2.	X	X	CardID	Integer	Yes	Card ID

- Bike

#	PK	FK	Column Name	Data Type	Mandator	Description

					y	
1	x		Id	Integer	Yes	ID, auto increment
2			barcode	VARCHAR	Yes	Bike’s barcode
3			color	VARCHAR	Yes	Bike’s color
4			category	VARCHAR	Yes	Bike’s category
5			bikeValue	float	Yes	Bike’s value
6			baseRentAmount	float	Yes	Bike’s base rent amount
7			additionalRentAmount	float	Yes	Bike’s additional rent amount
8			lock	BOOLEAN	Yes	Bike’s lock status

- E-Bike

#	*PK*	*FK*	*Column Name*	*Data Type*	*Mandatory*	*Description*
1.	x	x	id	Integer	Yes	Bike Id
2.			battery	Integer	Yes	Bike’s battery status

- Twin E-bike

#	*PK*	*FK*	*Column Name*	*Data Type*	*Mandatory*	*Description*
1.	x	x	id	Integer	Yes	Bike Id
2.			battery	Integer	Yes	Bike’s battery status

- Standard Bike

#	*PK*	*FK*	*Column Name*	*Data Type*	*Mandatory*	*Description*
1.	x	x	id	Integer	Yes	Bike Id

- Twin Bike

#	*PK*	*FK*	*Column Name*	*Data Type*	*Mandatory*	*Description*
1.	x	x	id	Integer	Yes	Bike Id

- **Database Script:**

```
create table "ecoBikeSystem"."DockStation"
(
    id serial not null,
    name VARCHAR not null,
    area VARCHAR not null,
    size int not null,
    address VARCHAR not null
);
create unique index dockstation_id_uindex
    on "ecoBikeSystem"."DockStation" (id);
alter table "ecoBikeSystem"."DockStation"
    add constraint dockstation_pk
        primary key (id);
create table "ecoBikeSystem"."Bike"
(
    id serial not null,
    barcode VARCHAR not null,
```

```
    color VARCHAR not null,
    category VARCHAR not null,
    "bikeValue" float not null,
    "baseRentAmount" float not null,
    "additionalRentAmount" float not null,
    lock BOOLEAN default FALSE not null
);
create unique index bike_barcode_uindex
    on "ecoBikeSystem"."Bike" (barcode);
create unique index bike_id_uindex
    on "ecoBikeSystem"."Bike" (id);
alter table "ecoBikeSystem"."Bike"
    add constraint bike_pk
        primary key (id);
create table "ecoBikeSystem"."BikeInDock"
(
    "BikeId" int not null
        constraint bikeindock_bike_id_fk
            references "ecoBikeSystem"."Bike"
                on update cascade on delete cascade,
    "DockId" int not null
        constraint bikeindock_dockstation_id_fk
            references "ecoBikeSystem"."DockStation"
                on update cascade on delete cascade,
    constraint bikeindock_pk
        primary key ("BikeId", "DockId")
);
create table "ecoBikeSystem"."Payment"
(
    id serial not null,
    "rentalCode" VARCHAR not null,
    "deductAmount" float not null,
    "depositAmount" float not null,
    "startRentTime" TIMESTAMP not null,
    "endRentTime" TIMESTAMP not null,
    status int not null
);
create unique index payment_id_uindex
    on "ecoBikeSystem"."Payment" (id);
create unique index payment_rentalcode_uindex
    on "ecoBikeSystem"."Payment" ("rentalCode");
alter table "ecoBikeSystem"."Payment"
    add constraint payment_pk
        primary key (id);
create table "ecoBikeSystem"."PaymentBike"
(
    "PaymentId" int not null
        constraint paymentbike_payment_id_fk
            references "ecoBikeSystem"."Payment"
                on update cascade on delete cascade,
    "BikeId" int not null
        constraint paymentbike_bike_id_fk
            references "ecoBikeSystem"."Bike"
                on update cascade on delete cascade,
    constraint paymentbike_pk
        primary key ("PaymentId", "BikeId")
);
create table "ecoBikeSystem"."Card"
(
    id serial not null,
    "cardCode" int not null,
```

```
    "cardName" VARCHAR not null,
    "dateExpired" VARCHAR not null,
    "cvvCode" int not null
);
create unique index card_cardcode_uindex
    on "ecoBikeSystem"."Card" ("cardCode");
create unique index card_id_uindex
    on "ecoBikeSystem"."Card" (id);
alter table "ecoBikeSystem"."Card"
    add constraint card_pk
        primary key (id);
create table "ecoBikeSystem"."PaymentCard"
(
    "PaymentId" int not null
        constraint paymentcard_payment_id_fk
            references "ecoBikeSystem"."Payment"
                on update cascade on delete cascade,
    "CardId" int not null
        constraint paymentcard_card_id_fk
            references "ecoBikeSystem"."Card"
                on update cascade on delete cascade,
    constraint paymentcard_pk
        primary key ("PaymentId", "CardId")
);
create table "ecoBikeSystem"."Ebike"
(
    id int not null
        constraint ebike_pk
            primary key
        constraint ebike_bike_id_fk
            references "ecoBikeSystem"."Bike"
                on update cascade on delete cascade,
    battery int not null
);
create table "ecoBikeSystem"."TwinEbike"
(
    id int not null
        constraint twinebike_pk
            primary key
        constraint twinebike_bike_id_fk
            references "ecoBikeSystem"."Bike"
                on update cascade on delete cascade,
    battery int not null
);
create table "ecoBikeSystem"."Standardbike"
(
    id int not null
        constraint standardbike_pk
            primary key
        constraint standardbike_bike_id_fk
            references "ecoBikeSystem"."Bike"
                on update cascade on delete cascade
);
create table "ecoBikeSystem"."Twinbike"
(
    id int not null
        constraint twinbike_pk
            primary key
        constraint twinbike_bike_id_fk
            references "ecoBikeSystem"."Bike"
                on update cascade on delete cascade
```

);