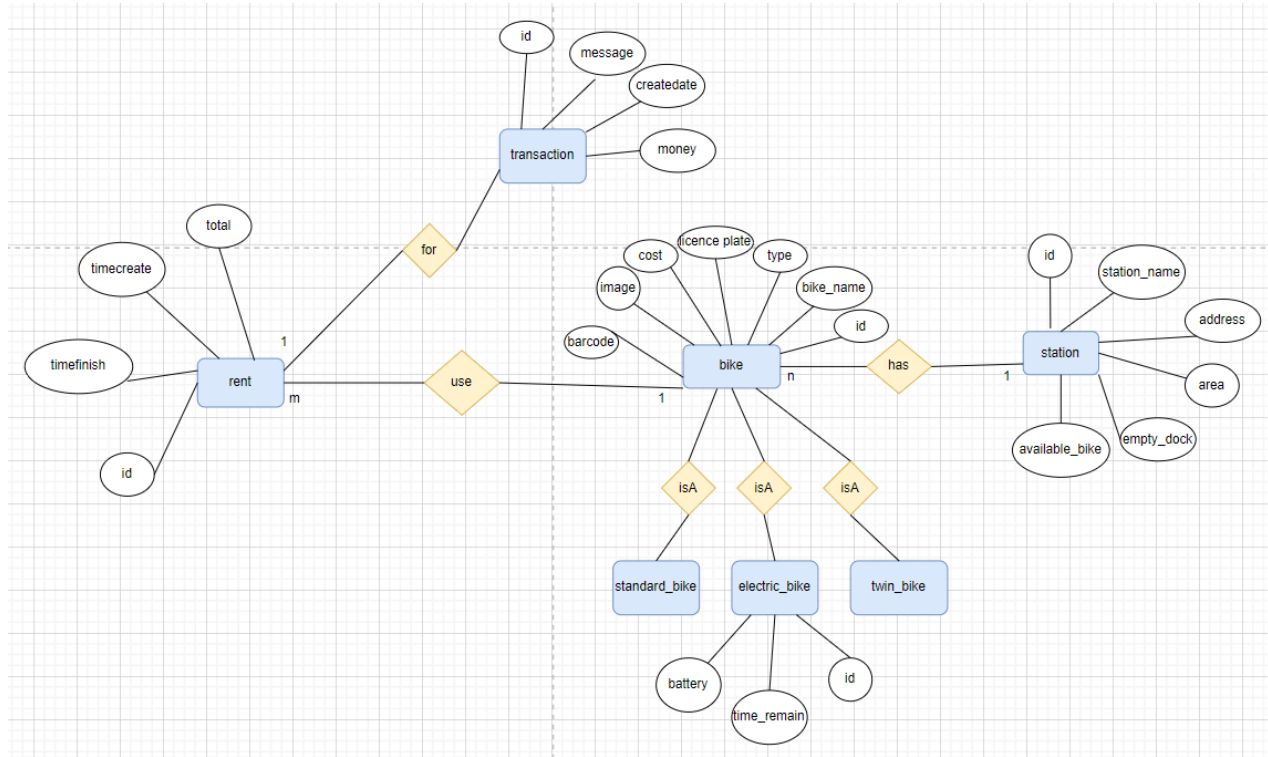


Database specifications for EcoBike Application

I. ER-Diagram



II. Table Specification

- Transaction

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID of transaction, auto increment
2			message	Varchar(50)	Yes	Content of transaction
5			createdate	datetime	Yes	Creation date of the transaction
6			money	double	Yes	Total amount of money is used for the transaction

- Station

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID of station, auto increment
2			station_name	Varchar(50)	Yes	Name of the station
3			empty_dock	Integer	Yes	Number of empty dock points in the station
4			available_bike	Integer	Yes	Number of available bike for renting in the station
5			area	double	Yes	Area of the station
6			address	varchar(50)	Yes	Address of the station

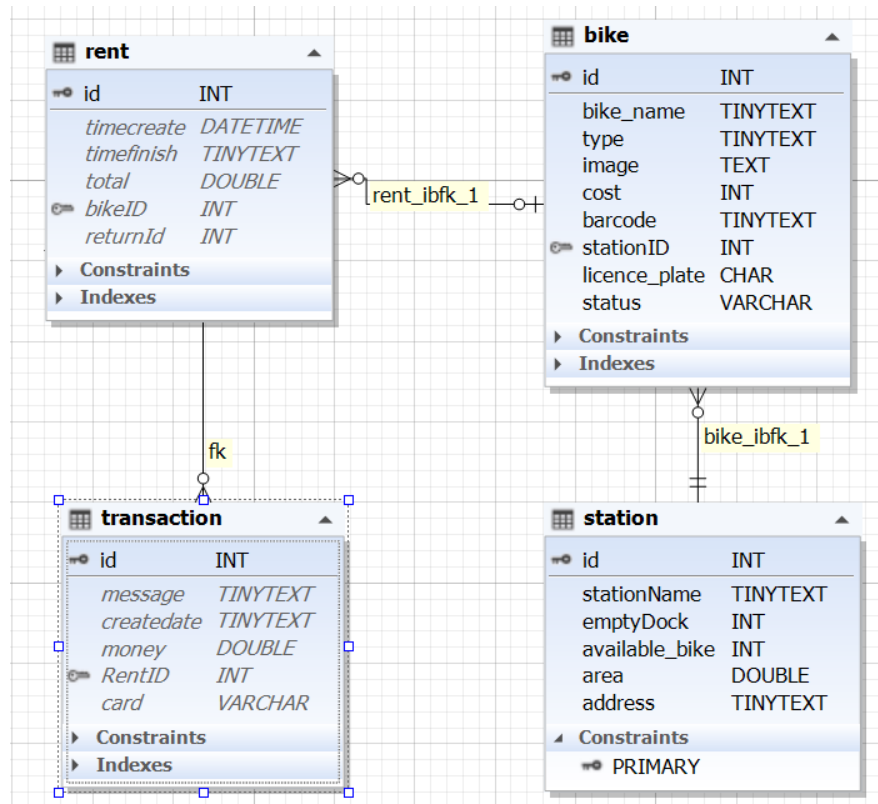
- Bike

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	id of the bike , auto increment
2			bike_name	Varchar(50)	Yes	Name of the bike
3			type	Varchar(50)	Yes	type of bike
4			image	Varchar(256)	Yes	Image of bike
5			cost	Integer	Yes	value of the bike
6			status	Varchar(10)	Yes	Status of the bike
7			barcode	Varchar(6)	Yes	Barcode of the bike
8			stationID	integer	Yes	Id of station that is contains bike
9			licence_plate	Varchar(6)	Yes	License plate of bike

- **Rent**

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID of order, auto increment
2			returnid	double	Yes	Id of station user wants to return
3			timecreate	datetime	Yes	When user rents bike
4			timefinish	datetime	No	When the user returns bike
5			total	double	No	The amount of renting money (not include deposit)
6		x	bikeID	integer	Yes	The id of the bike that user is/was renting

III. Database design



IV. SQL scripts

```
PRAGMA foreign_keys = off;
```

```
BEGIN TRANSACTION;
```

```
USE ecobikeproject;
```

```
CREATE TABLE IF NOT EXISTS Station (  
    id INTEGER PRIMARY KEY AUTO_INCREMENT,  
    station_name text(50) NOT NULL,  
    empty_dock INTEGER NOT NULL,  
    available_bike INTEGER NOT NULL,  
    area real NOT NULL,  
    address TEXT (50) NOT NULL);
```

```
CREATE TABLE IF NOT EXISTS Bike (  
    id INTEGER PRIMARY KEY AUTO_INCREMENT,  
    bike_name TEXT (50) NOT NULL,  
    type TEXT (50) NOT NULL,  
    image TEXT (256) NOT NULL,  
    cost INTEGER NOT NULL,  
    barcode TEXT (6) NOT NULL,  
    stationID integer NOT NULL,  
    licence_plate char(6) NOT NULL,  
    status varchar(10) NOT NULL,  
    FOREIGN KEY (stationID) REFERENCES Station(id));
```

```
CREATE TABLE IF NOT EXISTS Rent (  
    id INTEGER PRIMARY KEY AUTO_INCREMENT,  
    timecreate TEXT (10),  
    timefinish TEXT (10),  
    total REAL,  
    bikeID integer NOT NULL,  
    returnid integer,  
    FOREIGN KEY (bikeID) REFERENCES Bike(id));
```

```
CREATE TABLE IF NOT EXISTS Transaction (  
    id INTEGER PRIMARY KEY AUTO_INCREMENT,  
    message text (50) NOT NULL,  
    RentID integer NOT NULL,  
    createdate TEXT (10) NOT NULL,  
    money REAL NOT NULL,  
    FOREIGN KEY (RentID) REFERENCES rent(id));
```

```
COMMIT TRANSACTION;
```

```
PRAGMA foreign_keys = on;
```