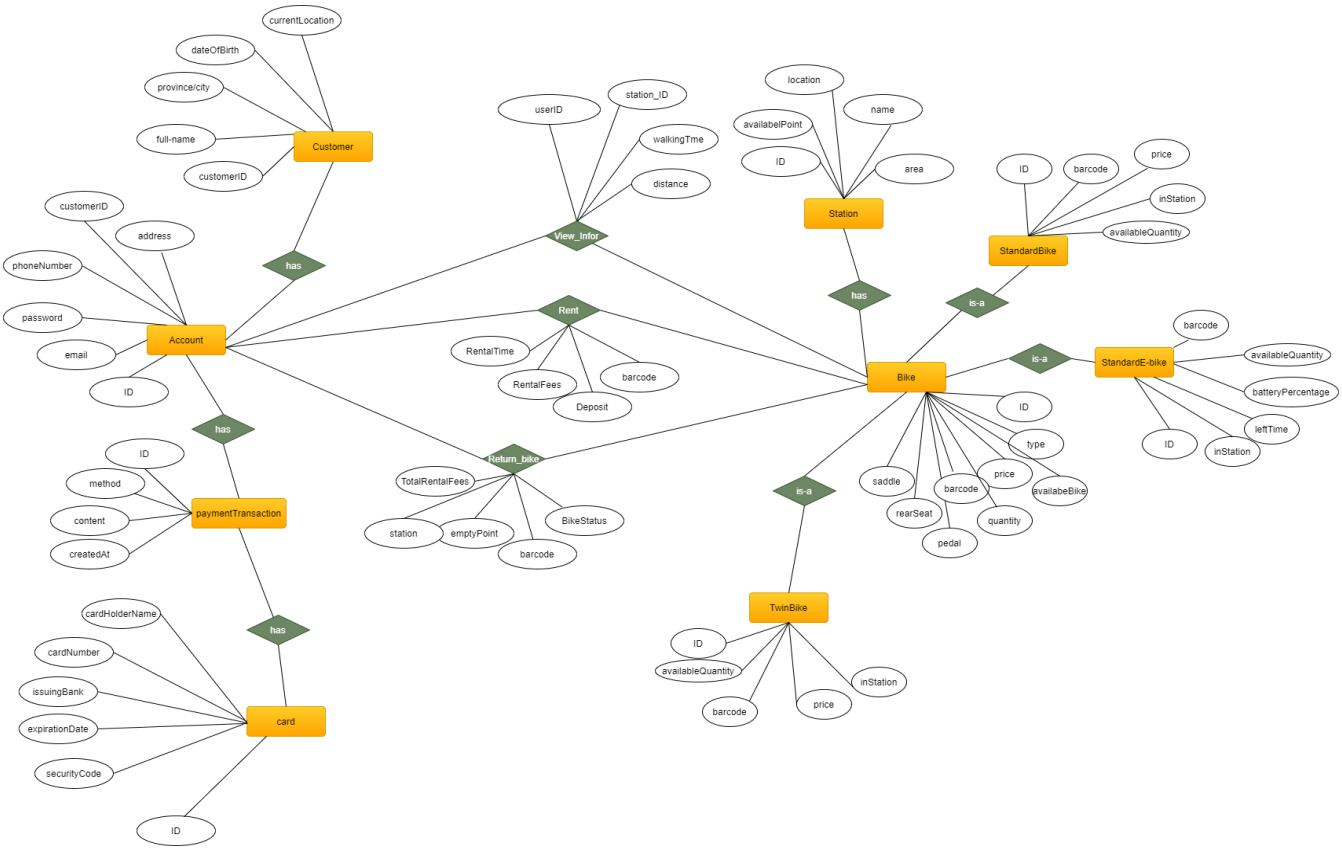
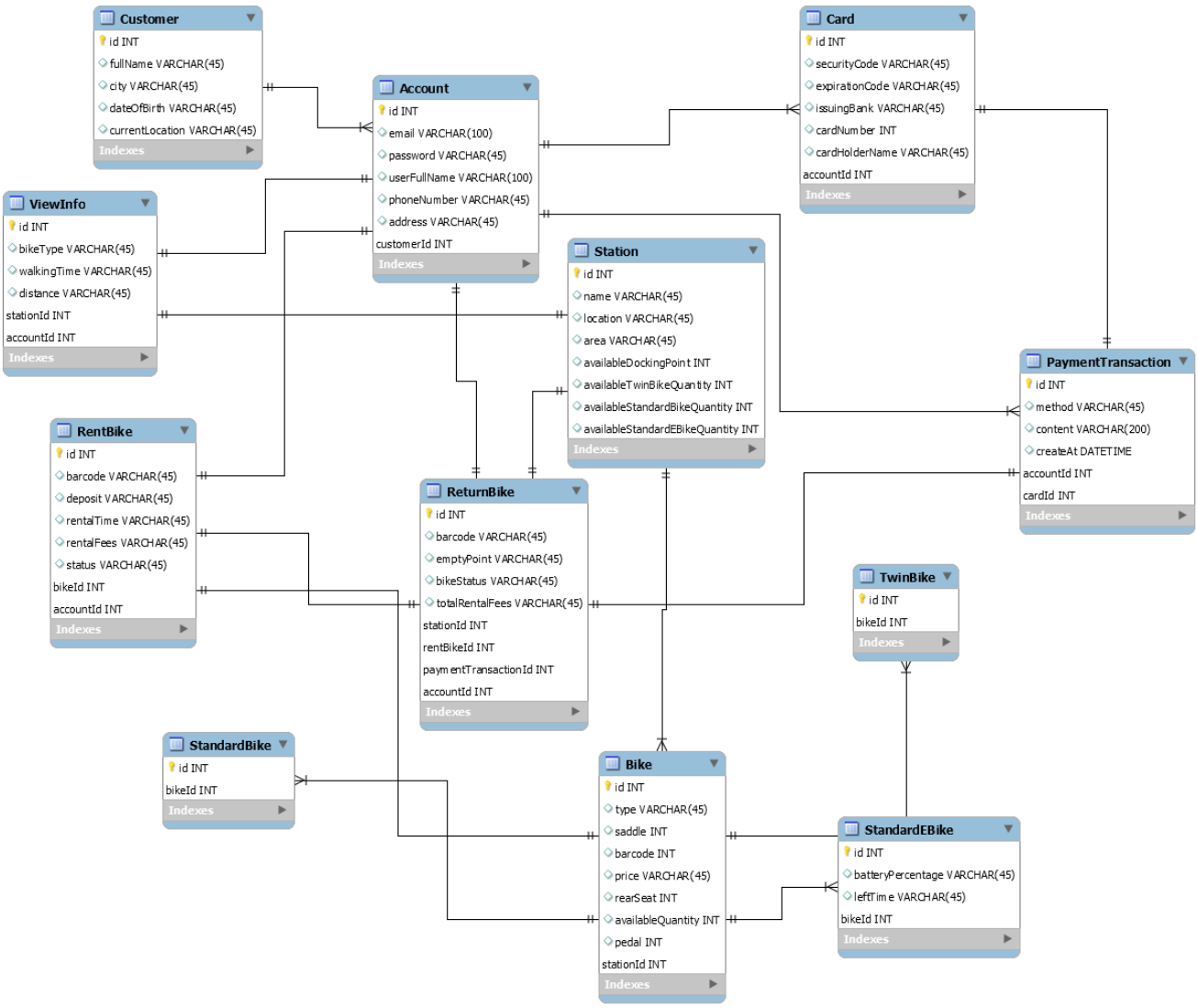
LAB 07 - DATA MODELING

1. An ER diagram :



2. Database design:

2.1 Logical Data Model:



2.2 Physical Data Model:

Customer:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			fullName	VARCHAR(45)	Yes	full name of customer
3			city	VARCHAR(45)	Yes	City where customer is in
4			dateOfBirth	DATETIME	Yes	Date when customer was born
5			currentLocation	VARCHAR(45)	Yes	Current location of customer

Account:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			email	VARCHAR(45)	Yes	full name of customer
3			password	VARCHAR(45)	Yes	City where customer is in
4			phoneNumber	VARCHAR(45)	Yes	Current location of customer
5		x	customerId	INT	Yes	Id of customer

Card:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			securityCode	VARCHAR(45)	Yes	Security code of card
3			expirationCode	VARCHAR(45)	Yes	Expiration code of card
4			issuingBank	DATETIME	Yes	Issuing bank
5			cardNumber	VARCHAR(45)	Yes	Number of card
6			cardHolderName	VARCHAR(45)	Yes	Name of holder
7		x	accountId	INT	Yes	Id of account

ViewInfo:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2		x	accountId	INT	Yes	Id of account
3		x	stationID	INT	Yes	Id of dock/station
4			walkingTime	INT	Yes	Amount time that user walking from current location to station
5			distance	INT	Yes	Distance that user walking from current location to station

RentBike:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			barCode	VARCHAR(45)	Yes	Code of bike
3			deposit	VARCHAR(45)	Yes	Deposit equal 40% of bike value
4			rentalTime	VARCHAR(45)	Yes	Date when customer was born
5			rentalFees	VARCHAR(45)	Yes	Fee for renting
6			status	VARCHAR(45)	Yes	Status of bike(percentage of battery, ..)
7		x	accountId	INT	Yes	Id of account
8		x	bikeId	INT	Yes	Id of bike

Station:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			name	VARCHAR(45)	Yes	Name of station/dock
3			location	VARCHAR(45)	Yes	Location of station
4			area	REAL	Yes	Area of station (unit: m^2)
5			availableDockingPoint	INT	Yes	Available point in station
6			availableTwinBike	INT	Yes	Number of available twin bike
7			availableStandardEBike	INT	Yes	Number of available standard e-bike
8			availableStandardBike	INT	Yes	Number of available standard bike

ReturnBike:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			barcode	VARCHAR(45)	Yes	Code of bike
3			emptyPoint	VARCHAR(45)	Yes	Number of empty points
4			bikeStatus	DATETIME	Yes	Status of bike
5			totalRentalFees	VARCHAR(45)	Yes	Total fee for renting bike
6		x	stationId	INT	Yes	Id of station
7		x	rentBikeId	INT	Yes	Id of renting bike
8		x	paymentTransactionId	INT	Yes	Id of paymentransaction
9		x	accountId	INT	Yes	Id of account

PaymentTransaction:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			method	VARCHAR(45)	Yes	Method for paying fee
3			content	VARCHAR(200)	Yes	Content of paying
4			createAt	DATETIME	Yes	Date when customer create
5		x	accountId	INT	Yes	Id of account
6		x	cardId	INT	Yes	Id of card

Bike:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2			type	VARCHAR(45)	Yes	Type of bike
3			saddle	INT	Yes	number of saddle
4			barcode	VARCHAR(45)	Yes	code of bike

5			price	INT	Yes	price or fee for renting bike per minute
6			rearSeat	INT		number of rear seat
7			availablequantity	INT		number of available bikes
8			pedal	INT		number of pedal
9		x	stationId	INT		id of station

StandardBike:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2		x	bikeId	INT	Yes	Id of bike

TwinBike

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2		x	bikeId	INT	Yes	Id of bike

StandardEBike:

#	PK	FK	Column Name	Data type	Mandatory	Description
1	x		id	INT	Yes	Id, auto increment
2		x	bikeId	INT	Yes	Id of bike
3			batteryPercentage	INT	Yes	Percentage of battery
4			leftTime	INT	Yes	Time left of the e-bike

2.2 Database Script:

// Table Customer

```
CREATE TABLE Customer (
    id    INT  AUTO_INCREMENT,
    fullName    VARCHAR(45) NOT NULL,
    city          VARCHAR(45) NOT NULL,
    dateOfBirth     DATETIME,
    currentLocation     VARCHAR(45)    NOT NULL,
    PRIMARY KEY(id)
);
```

// Table Account

```
 CREATE TABLE Account (
    id    INT     NOT NULL    AUTO_INCREMENT,
    email     VARCHAR(45) NOT NULL,
    city VARCHAR(45) NOT NULL,
    password     VARCHAR(45),
    phoneNumber     VARCHAR(45),
    customer_id INT,
    PRIMARY KEY (id),
```

```
    FOREIGN KEY    (customer_id) REFERENCES Customer (id)
);

// Table Card
CREATE TABLE Card (
    id    INT  NOT NULL    AUTO_INCREMENT,
    securityCode      VARCHAR(45)      NOT NULL,
    expirationCode     VARCHAR(45)    NOT NULL,
    issuingBank  DATETIME    NOT NULL,
    cardNumber VARCHAR(45)    NOT NULL,
    cardHolderNameVARCHAR(45)    NOT NULL,
    account_id   INT,
    PRIMARY KEY (id),
    FOREIGN KEY (account_id) REFERENCES Account (id)
);

// Table Station
CREATE TABLE Station (
    id    INT      NOT NULL       AUTO_INCREMENT,
    name     VARCHAR(45)       NOT NULL,
    location VARCHAR(45)    NOT NULL,
    area      REAL       NOT NULL,
    availableDockingPoint      INT,
    availableStandardEBike    INT,
    availableStandardBike      INT
);
// Table ViewInfo
CREATE TABLE ViewInfo (
    id    INT  NOT NULL    AUTO_INCREMENT,
    accountId     INT,
    stationID      INT,
    walkingTime INT        NOT NULL,
    distanceINT        NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (accountId) REFERENCES Account (id),
    FOREIGN KEY (stationID) REFERENCES Station(id)
);

// Table Bike
CREATE TABLE Bike (
    id    INT  NOT NULL    AUTO_INCREMENT,
    type      VARCHAR(45) NOT NULL,
    saddle   VARCHAR(45)    NOT NULL,
    barCodeVARCHAR(45)    NOT NULL,
    price     INT,
    rearSeatINT,
    availableQantity  INT,
```

```
    pedal     INT,
    stationId      INT,
    PRIMARY KEY (id),
    FOREIGN KEY (stationId) REFERENCES Station(id)
);

// Table RentBike
CREATE TABLE RentBike (
    id    INT  NOT NULL     AUTO_INCREMENT,
    barCodeVARCHAR(45)    NOT NULL,
    deposit        VARCHAR(45)    NOT NULL,
    rentalTime    VARCHAR(45)    NOT NULL,
    rentalFees    VARCHAR(45)    NOT NULL,
    status         VARCHAR(45)    NOT NULL,
    accountID     INT,
    bikeId         INT,
    PRIMARY KEY (id),
    FOREIGN KEY (accountID) REFERENCES Account(id),
    FOREIGN KEY (bikeId) REFERENCES Bike(id)
);

// Table PaymentTransaction
CREATE TABLE PaymentTransaction (
    id    INT    NOT NULL      AUTO_INCREMENT,
    method VARCHAR(45),
    content       VARCHAR(45),
    createAtDATETIME      NOT NULL,
    accountId     INT,
    cardId         INT,
    PRIMARY KEY (id),
    FOREIGN KEY (accountId) REFERENCES Account (id),
    FOREIGN KEY (cardId)    REFERENCES Card (id)
);

// Table ReturnBike
CREATE TABLE ReturnBike (
    id    INT      NOT NULL      AUTO_INCREMENT,
    barCodeVARCHAR(45) NOT NULL,
    emptyPoint  VARCHAR(45) NOT NULL,
    bikeStatus    DATETIME    NOT NULL,
    totalRentalFees   VARCHAR(45)    NOT NULL,
    stationId      INT,
    rentBikeId    INT,
    paymentTransactionId      INT,
    accountId     INT,
    PRIMARY KEY (id),
    FOREIGN KEY (stationId) REFERENCES    Account (id),
```

```
    FOREIGN KEY (rentBikeId)    REFERENCES RentBike (id),
    FOREIGN KEY (paymentTransactionId)    REFERENCES    PaymentTransaction (id),
    FOREIGN KEY (accountId)    REFERENCES      Account (id)
);

// Table StandardBike
CREATE TABLE StandardBike (
    id    INT      NOT NULL      AUTO_INCREMENT,
    bikeId          INT,
    PRIMARY KEY (id),
    FOREIGN KEY (bikeId) REFERENCES    Bike (id)
);

// Table TwinBike
CREATE TABLE TwinBike (
    id    INT  NOT NULL    AUTO_INCREMENT,
    bikeId          INT,
    PRIMARY KEY (id),
    FOREIGN KEY (bikeId)    REFERENCES Bike (id)
);

// Table StandardEBike
CREATE TABLE StandardEBike (
    id    INT  NOT NULL    AUTO_INCREMENT,
    bikeId          INT,
    batteryPercentage     INT,
    leftTime INT,
    PRIMARY KEY (id),
    FOREIGN KEY (bikeId) REFERENCES Bike (id)
);
```