King Abdulaziz University
Faculty of Computing and Information Technology
Computer Science Department

CPCS 241

Databases

Group Project – Phase II

Student Names: ……………………………………………………………………

Student Numbers: ……………………………………………………………………

Group Number: ……………………………………………………………………

Semester: Spring 2022

Submission Date: Monday 10/09/1443 H, **11/04/2022** G

Submission Time: DAR, GAR & IAR @ 12:00 PM

Submission Format: Paper copy + Soft copy via BB

Criteria	Max Mark	[redacted]	Obtained Mark
[redacted]	[redacted]	[redacted]	[redacted]
Analysis	**1**	[redacted]	
ER Design **(HVCLO6)**	**2**	[redacted]	
ER-to-Relational Mapping (HVCLO10)	**2**	[redacted]	
Normalization **(HVCLO11)**	**2**	[redacted]	
[redacted]	[redacted]	[redacted]	[redacted]
Total	**7**	[redacted]	

Instructors: Dr. Lama Al Khuzayem and Dr. Rsha Mirza

lalkhuzayem@kau.edu.sa , rmirza@kau.edu.sa

CPCS241 – Database I – Spring 2022 – Group Project

Smart Public Transportation System

DB Design

Group No: [6]

Student Name	Student Number
Ghada Essa Alsulami (Leader)	1905190
Sarah Abdullhadi Aljohani	1906525
Lujain Abdulaziz Alsulami	2006847
Deemah Abdullah Alghamdi	2006352
Dania Mohammed Kenani	2005302

Contents

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

Traffic jam is one the major problems in big cities. The vehicles must wait long time to move out of the jam. This happens due to the increasing number of vehicles and overuse of roads which is reflected on vehicles slow, lost a lot of time without production, and negatively effects work. To reduce this problem, we should use public transport as much as possible. Hence, our idea came in frame of building up a public transportation system to help our city to own convenient transportations which limit from the endless congestion.

How Does System Work?

We will explain briefly here. First, there are central stations will be distributed in every area of the city (west, east, north, south), and stations found at the end of the roads that carries passengers from the point into destination on time to leave without waiting and stopping. These stations all are related to an application that passenger can get our services easily and comfortably. This application saves passengers time and give them the authority to choose any resources they want. Each central station has some features we will explain them later, also a manager who check that no conflict on timetable between journeys. Every bus will have driver, responsible for its journey. There are many things to explain, but we will go in details in the next parts to make an idea more understandable.

The user\ passenger to be able to use our service, firstly, he should be sign in to our application, then the subscription packages will be appear after successfully log in into his account, the packages available to it are annual, monthly, and weekly. Once, he chooses on of them we can say welcome to our system, and the timetable will be shown the details of each journey. Anytime and anywhere, we are ready to help them.

Figure imaginary that shows, that our system will be like this picture everything is in a network that organized and do the best to reduce the crowd in the streets.

1.2 Data Requirements

PASSENGERS

- ID number.
- Name (First. Middle, Last).
- Phone Number.
- Occupation.
- Birth of data.
- Gender.
- Health status.
- Email.

EMPLOYEE

- SSN (social Security Number).
- Name (First. Middle, Last).
- License Number (Driver).
- Gender.
- Birth of data.
- Salary.
- Start of work date.
- Address.
- Phone Number.
- Organizer (Manager).
- Permissions (Administrator).

[Employee superclass has three subclasses: Administrator, Manager, Driver]

BUS

- Plate number.
- Bus ID.
- Bus capacity(seats).

STATION

- Station number.
- Zip code(neighbourhood).
  [Station superclass has two subclasses: Central, Terminal]

SUBSCRIPTION

- Payment cost.
- Date of payment.
- Subscription Duration (Start, End).
- Bill Number.
  [Subscription has two type attributes: Type of package, and payment method]

TIMETABLE

- Journey ID.
- Destination.
- Date.
- Time (Starting-Ending).

SERVICE

- Playback Number.
- Service Name.

LINES

- Line Name
- Final stop
- Length.

LEAVES

- Date.
- Name.

ACCOUNT

- Email.
- Password.
- Phone number.
- Account ID.

1.3 Business Rules

MANAGER

- He should take care of the timetable. So that it provides buses at different times at each station.

LINE

- There are multiple roads a bus can chose from them depending on peak times of region.

PASSENGERS

- To be able to serve this passenger and reserve a seat for him, must they have a unique ID number / the residence number.
- Should be an adult or, a child with an adult.
- The subscription should be valid.

DRIVERS

- Must they have a bus driver's license.
- Each driver should drive one bus a day no switching.
- The driver should follow the timetable.

TIMETABLE

- Each journey has unique ID.
- Each type of destination has different price.
- Each journey has a limited number of passengers.
- The time of starting /ending the journey must be specified.

SUBSCRIPTIONS

- There are two payment methods, each passenger must choose one of them.
- The cost depends on destination.

BUSES

- Each bus has a unique ID number that allow passengers to enter the bus by barcode-scanning.
- Not allow non-passengers- accompanying person- to enter.
- Once, the departure time come which timetable imposes for each journey, the bus will move to its destination without stop at any other station.

- Passengers are not allowed to eat, and drink inside the bus.
- Bus provides an open network for subscriber within annual packages.
- Passenger must be adhered to assigned seat for him in the information of subscription.

STATION

- There is a central station that responsible in the operation of bus services, and smaller one at the outer ends of the routes.
- The private cars are not allowed to wait at the small stations to avoid crowded, but they can park on the central station in allocated parking.
- No more one bus can be at the same time in the station.

SERVICES

- Each station at least has the basics service such as machine for water and snakes, bathrooms, waiting area.
- Any station in main areas have more features such as place for entertainment, coffee shops, and small restaurants.

LEAVES

- Each holiday schedule like a notional event for everyone, so that why in these times the stations are closed such as (first day in Eid Al-Adha and Eid Alfater), and in the national day.
- All the stations closed except the main area satiation.

1.4 Intended Output of the system

OUTPUT & QUERIES:

- Show the number of days left of subscription.
- Show, the number of subscribers in each package.
- Show, the timetable of journeys.
- Update passenger subscription.
- Update driver information.

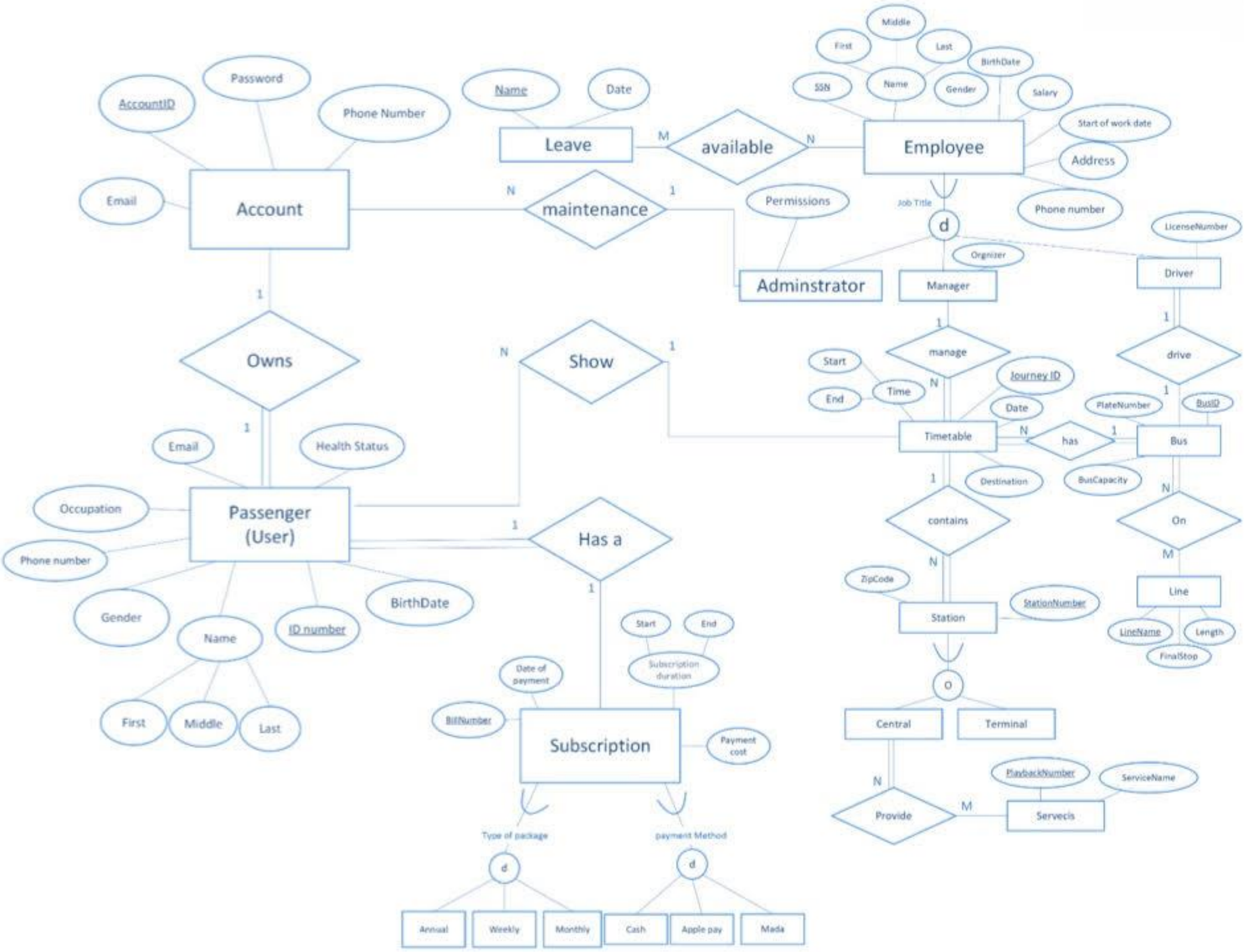
TRANSACTION

- Change time of journey.
- Update the data of passengers.
- Update passenger subscription.
- Update driver information.

PART II: DB DEISGN

2 ER Diagram Design

2.1 ER Diagram



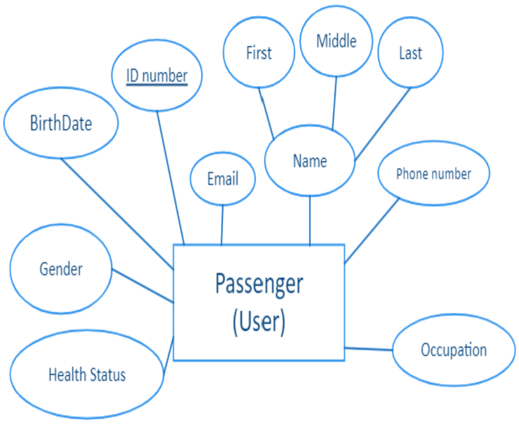
2.2 Design of Business Rules

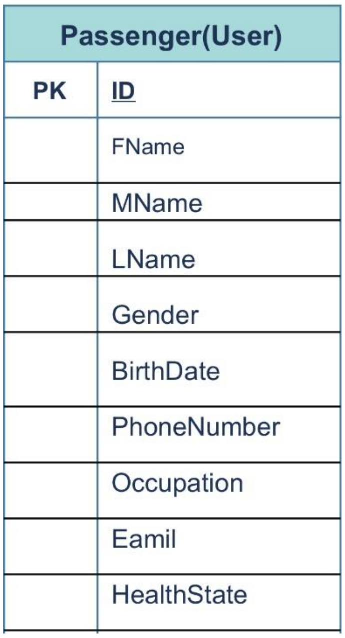
Business Rule	Design Decisions	Justification (if any)
Each user/passenger owns an account	1:1 binary relationship between user/passenger and Account	Each user/passenger should be own an account while the account isn't.
Each user/passenger has a subscription.	1:1 binary relationship between user/passenger and subscription.	Each user/passenger must has a subscription. A subscription has a user.
A Timetable will be shown to many users.	1:N binary relationship between Timetable and Passenger.	Many passenger/users will have the same (only one) Timetable.
Each timetable should be managed by a manger	1:N binary relationship between manager and timetable.	Only one manager manages many timetables. All timetables must be managed by one manager.
Each timetable will contain the stations	1:N binary relationship between timetable and station.	A timetable must contain many stations, and each station must contain one timetables.
The central stations will provide many services	N:M binary relationship between central Station and Services.	For every central station provides many services. While each service provided in many central stations.
Each driver must only drive one bus	1:1 binary relationship between Driver and Bus.	Only one driver must drive only one bus. While only one bus driven by one driver.
Any bus can be on any line	N:M binary relationship between Driver and Bus.	For every bus can be on many lines, while each line can have many busses on it.
Employees have holidays	N:M binary relationship between holidays and Employees.	Multiple holidays available to many employees.
Each bus has timetable	N:M binary relationship between bus and Timetable.	Each bus has many timetables. While many timetable must exist in every one bus.

An administrator maintenance accounts	1:N binary relationship between Administrator and Accounts.	Only one Administrator maintenance many Accounts. while each account maintenance by one Administrator.

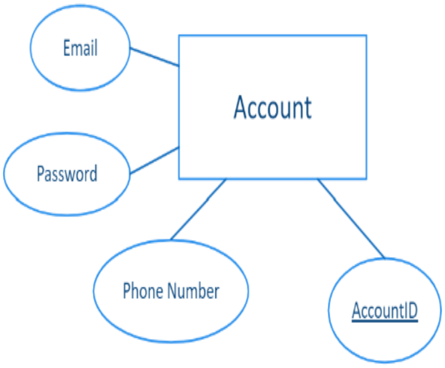
3 ER-to-logical Schema Mapping

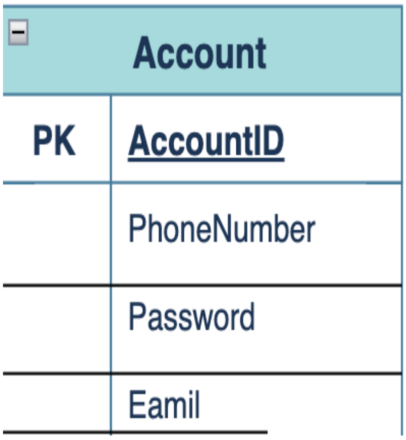
3.1 Mapping of Regular Entity Types



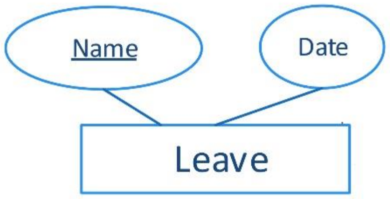


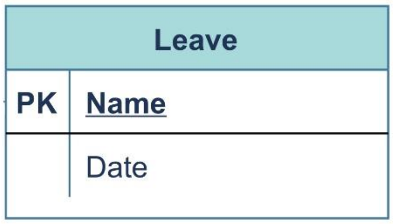
Passenger(User)	
PK	ID
	FName
	MName
	LName
	Gender
	BirthDate
	PhoneNumber
	Occupation
	Eamil
	HealthState



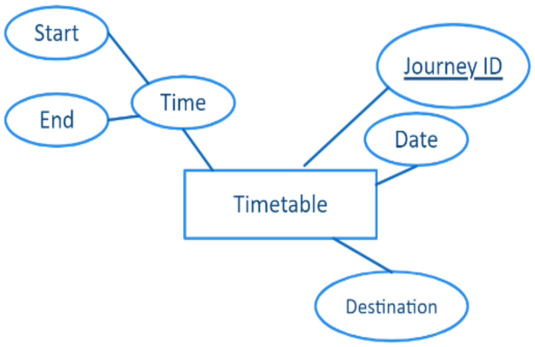


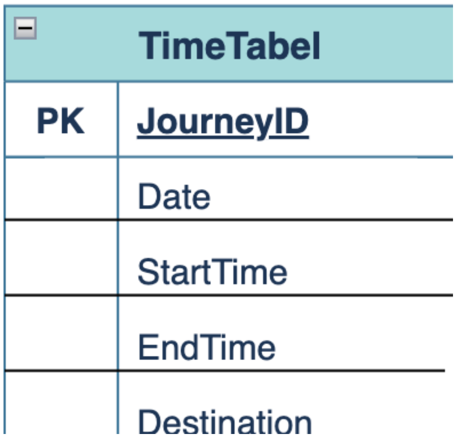
Account	
PK	AccountID
	PhoneNumber
	Password
	Eamil



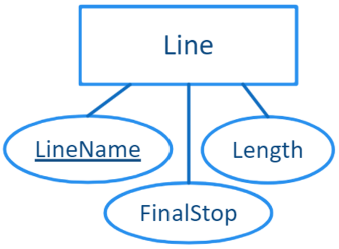


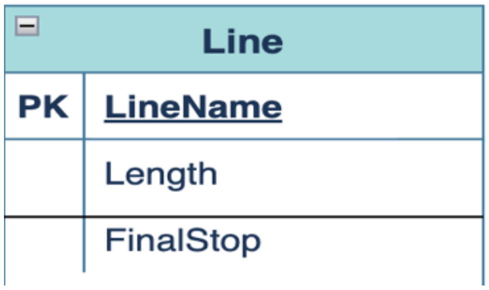
Leave	
PK	Name
	Date



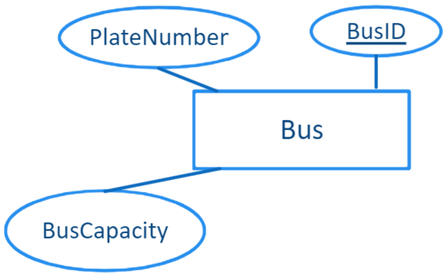


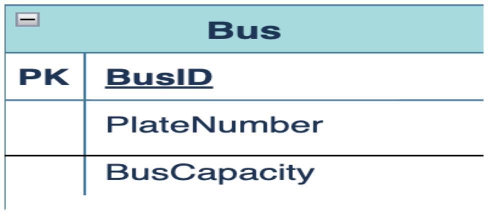
TimeTabel	
PK	JourneyID
	Date
	StartTime
	EndTime
	Destination



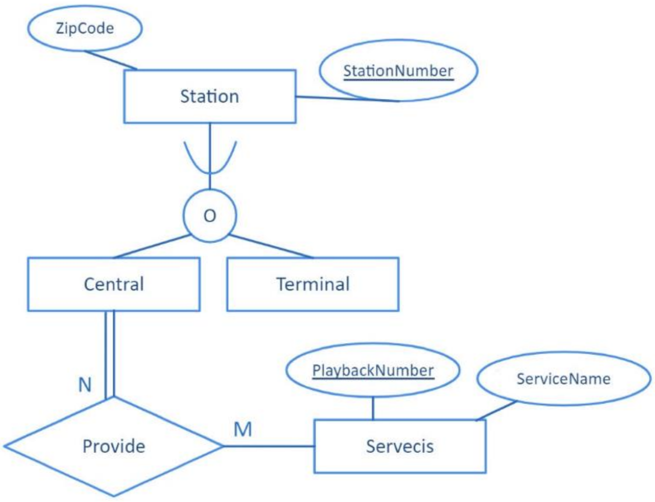


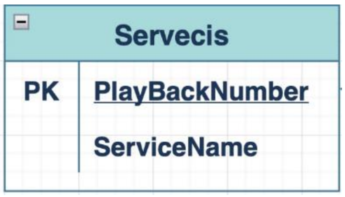
Line	
PK	LineName
	Length
	FinalStop



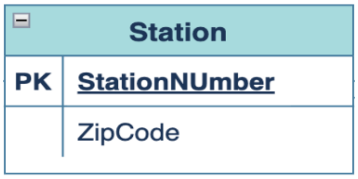


Bus	
PK	BusID
	PlateNumber
	BusCapacity

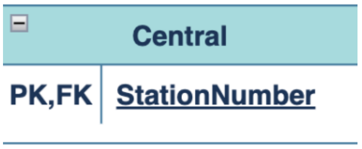




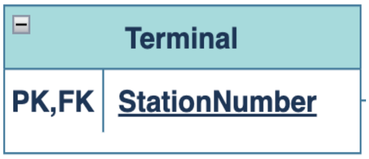
Servecis	
PK	PlayBackNumber
	ServiceName



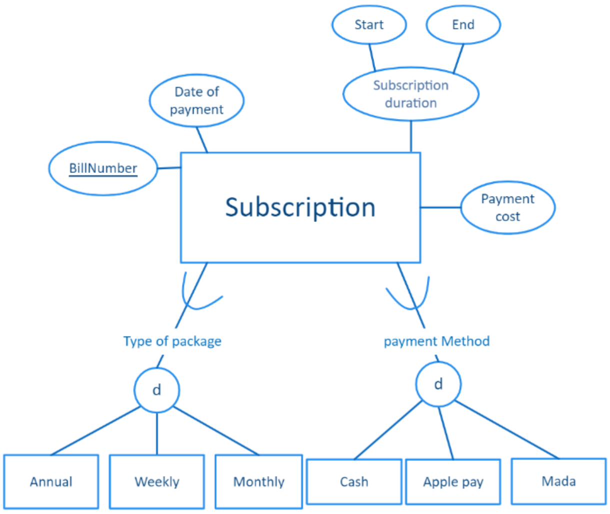
Station	
PK	StationNUmber
	ZipCode

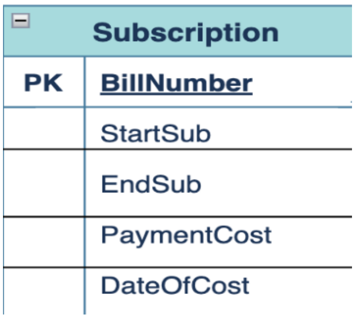


Central	
PK,FK	StationNumber

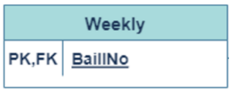


Terminal	
PK,FK	StationNumber

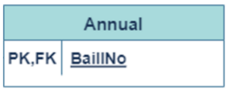




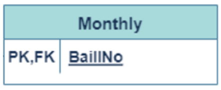
Subscription	
PK	BillNumber
	StartSub
	EndSub
	PaymentCost
	DateOfCost



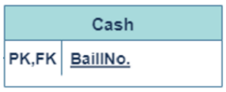
Weekly	
PK,FK	BaillNo



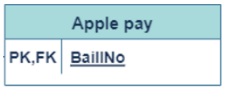
Annual	
PK,FK	BaillNo



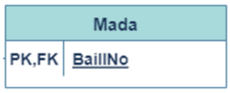
Monthly	
PK,FK	BaillNo



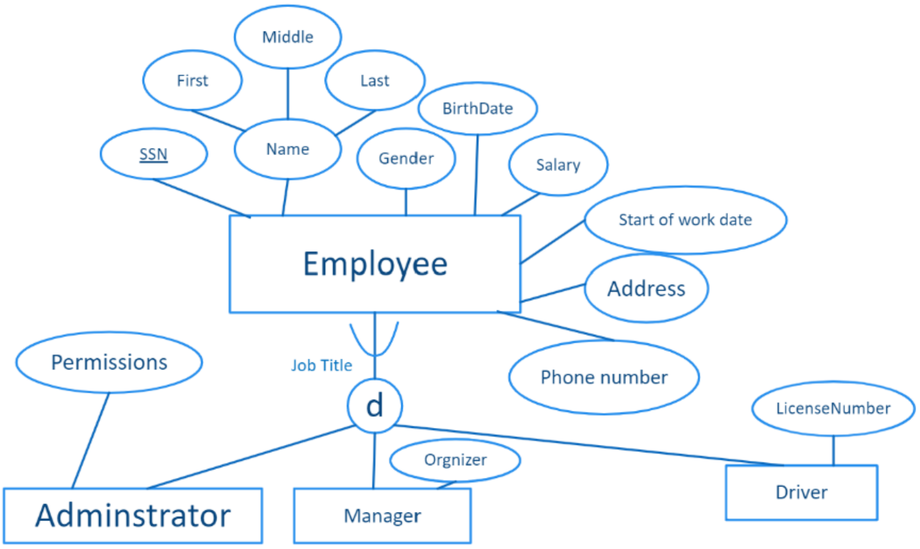
Cash	
PK,FK	BaillNo.

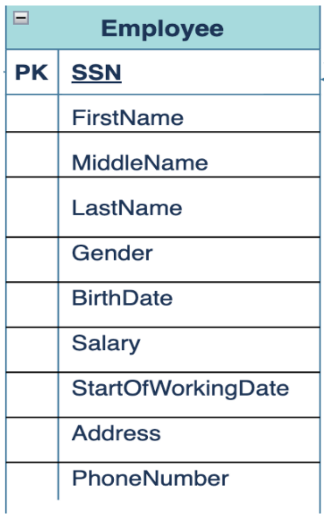


Apple pay	
PK,FK	BaillNo

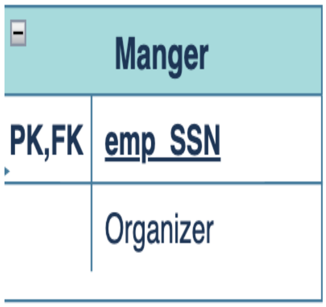


Mada	
PK,FK	BaillNo

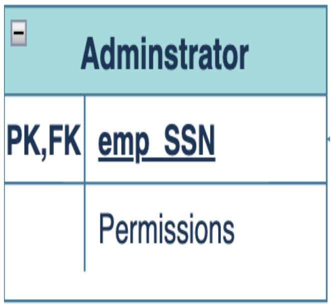




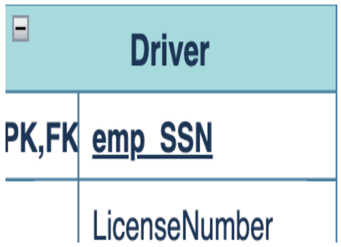
	Employee
PK	**SSN**
	FirstName
	MiddleName
	LastName
	Gender
	BirthDate
	Salary
	StartOfWorkingDate
	Address
	PhoneNumber



	Manger
PK,FK	emp_SSN
	Organizer



	Adminstrator
PK,FK	emp_SSN
	Permissions

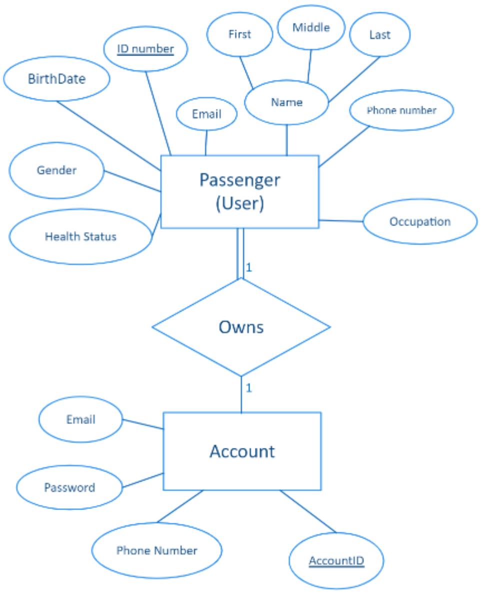


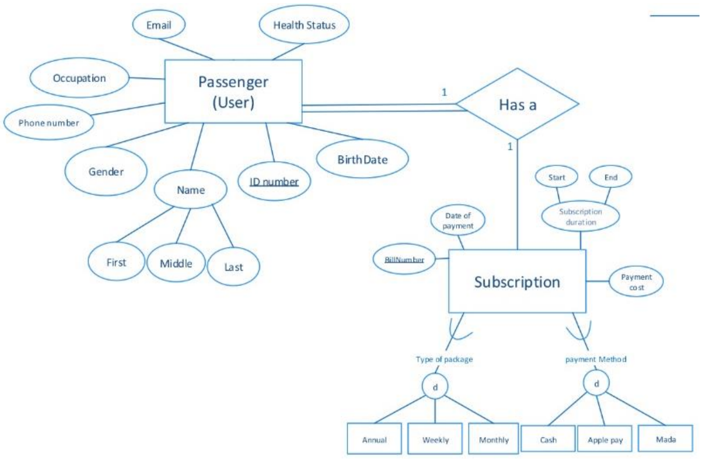
	Driver
PK,FK	emp_SSN
	LicenseNumber

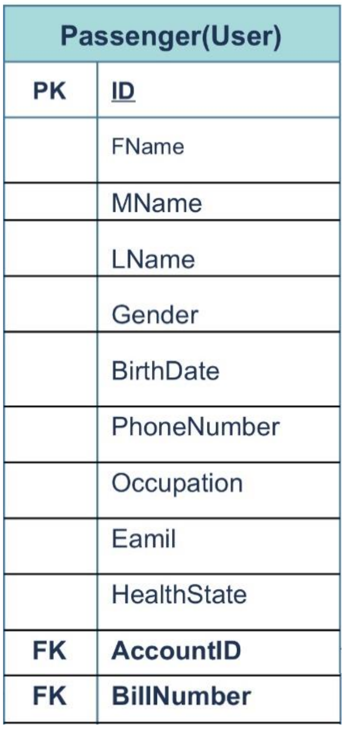
3.2 Mapping of Weak Entity Types

- We do not have a weak entity in our system.

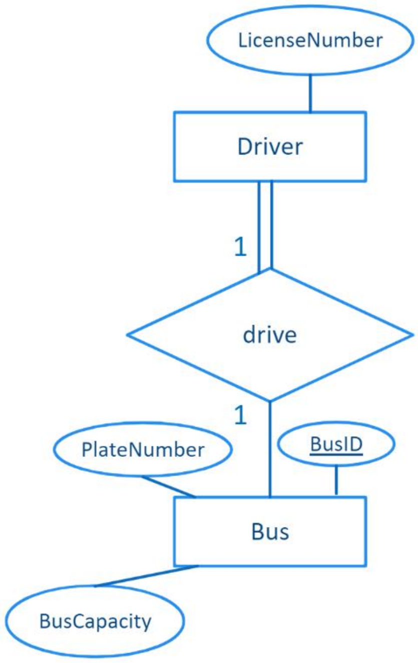
3.3 Mapping of Binary 1-1 Relationship Types

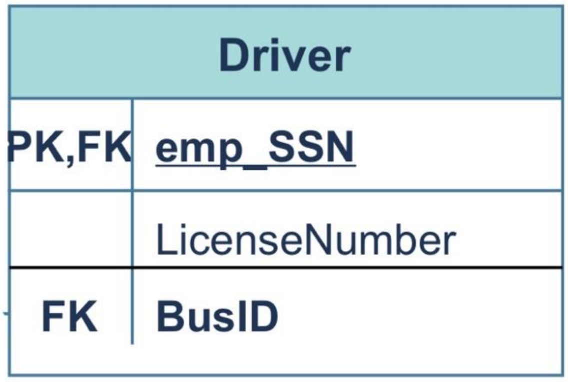






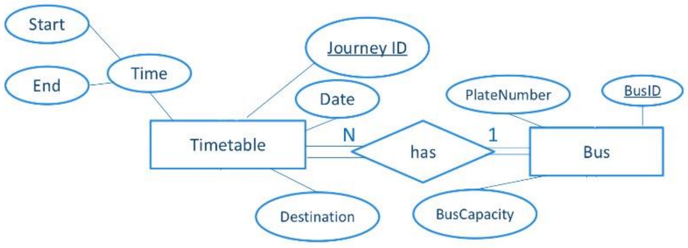
Passenger(User)	
PK	ID
	FName
	MName
	LName
	Gender
	BirthDate
	PhoneNumber
	Occupation
	Eamil
	HealthState
FK	**AccountID**
FK	**BillNumber**

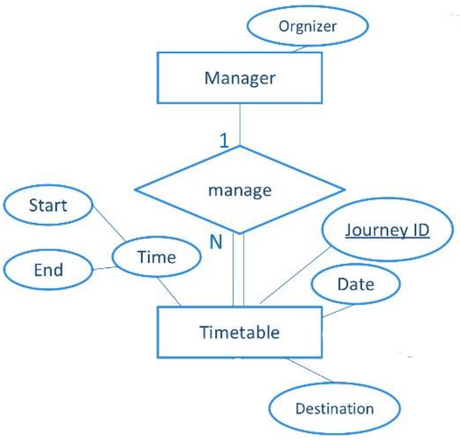


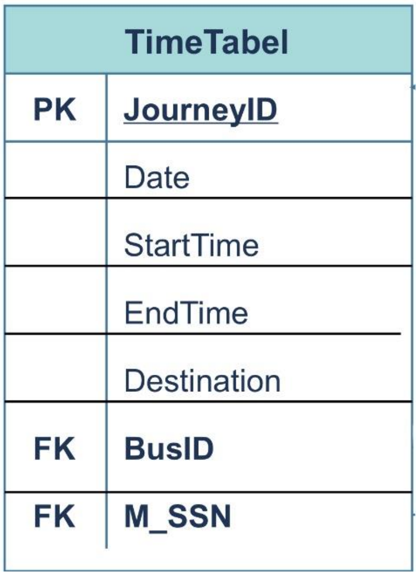


	Driver
PK,FK	emp_SSN
	LicenseNumber
FK	BusID

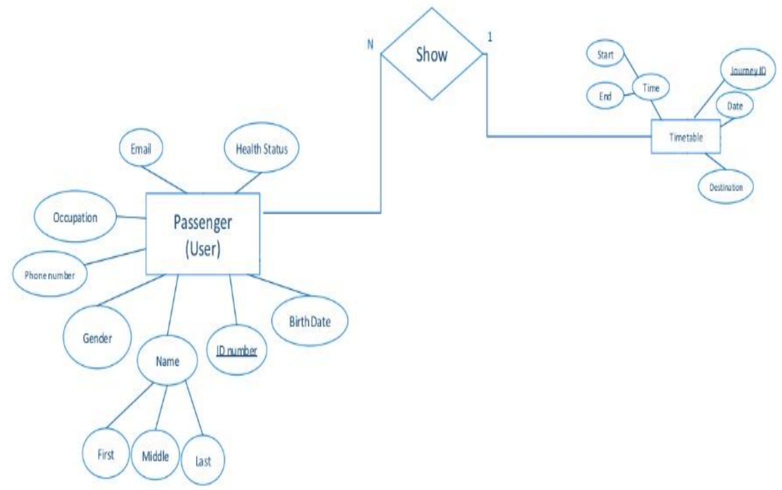
3.4 Mapping of Binary 1-N Relationship Types



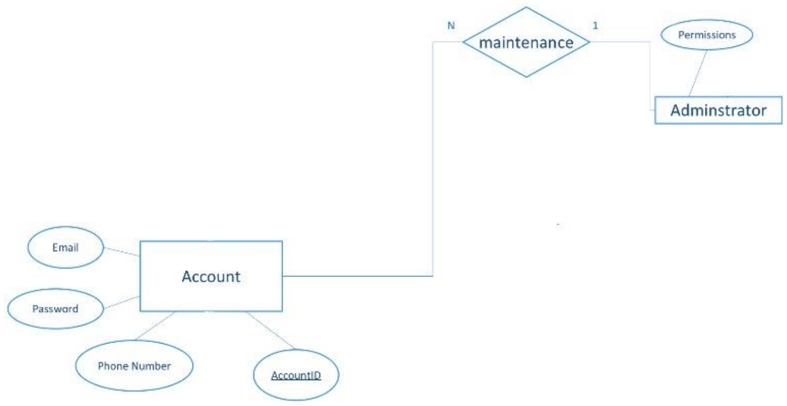


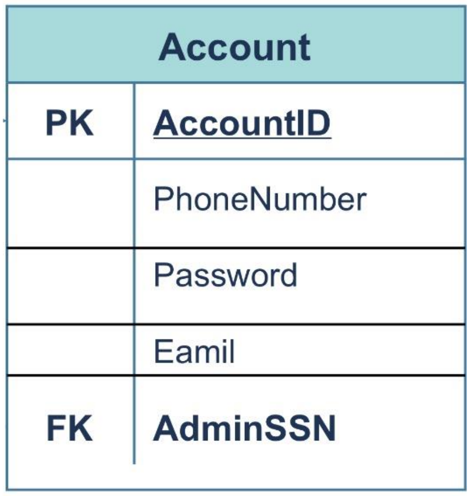


	TimeTabel
PK	JourneyID
	Date
	StartTime
	EndTime
	Destination
FK	BusID
FK	M_SSN

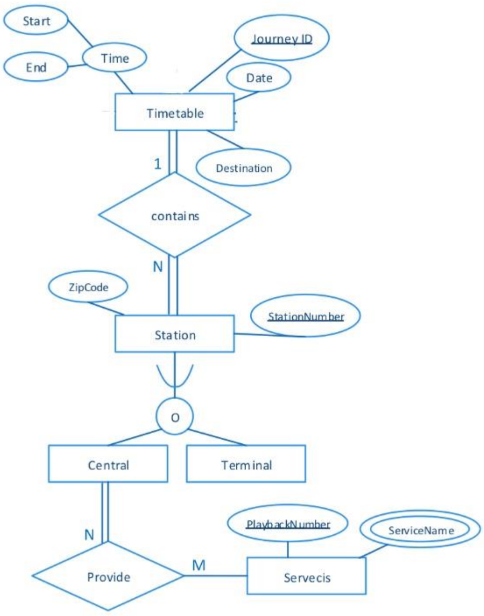


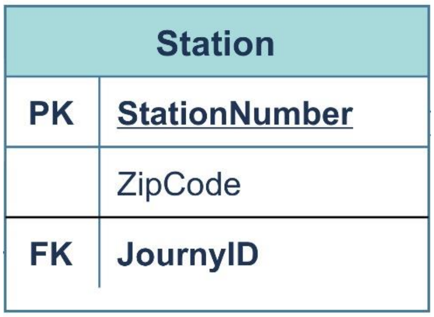
Passenger(User)	
PK	**ID**
	FName
	MName
	LName
	Gender
	BirthDate
	PhoneNumber
	Occupation
	Eamil
	HealthState
FK	**AccountID**
FK	**BillNumber**
FK	**JournyID**





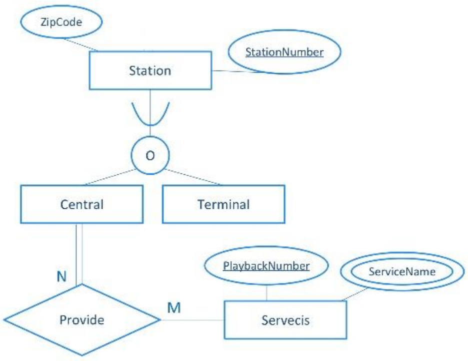
Account	
PK	AccountID
	PhoneNumber
	Password
	Eamil
FK	AdminSSN

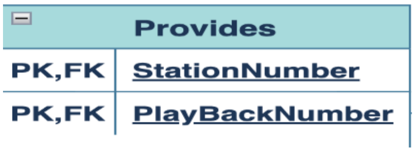




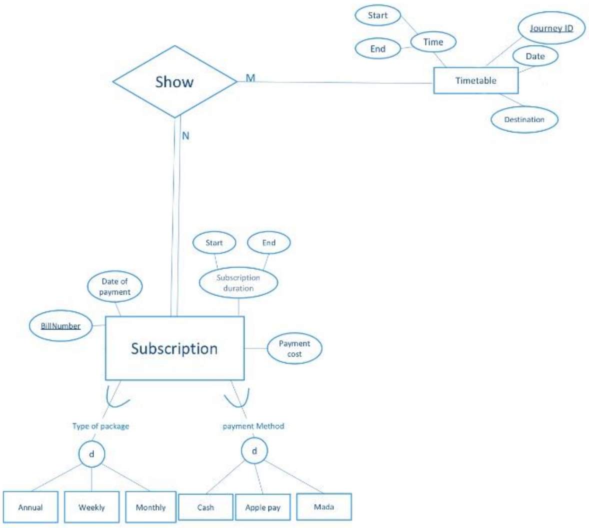
Station	
PK	StationNumber
	ZipCode
FK	JournyID

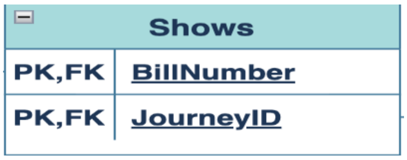
3.5 Mapping of Binary M-N Relationship Types



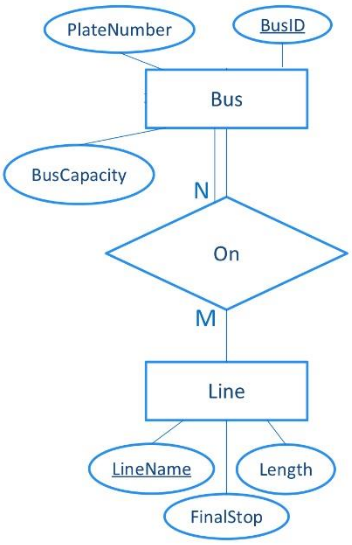


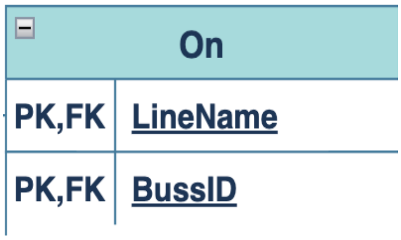
Provides	
PK,FK	StationNumber
PK,FK	PlayBackNumber



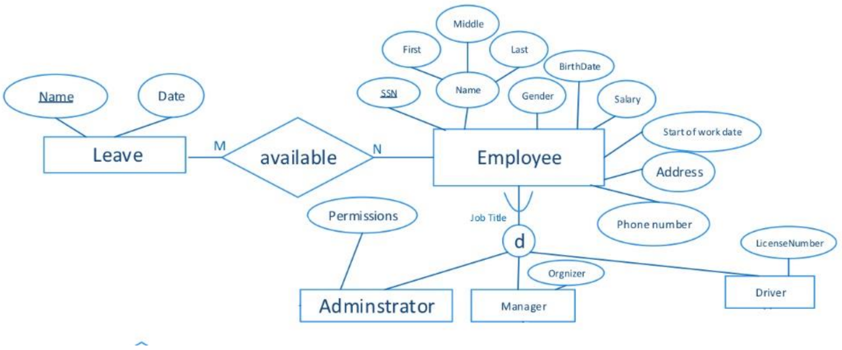


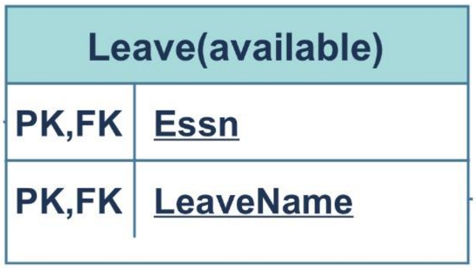
Shows	
PK,FK	BillNumber
PK,FK	JourneyID





On	
PK,FK	LineName
PK,FK	BussID





Leave(available)	
PK,FK	Essn
PK,FK	LeaveName

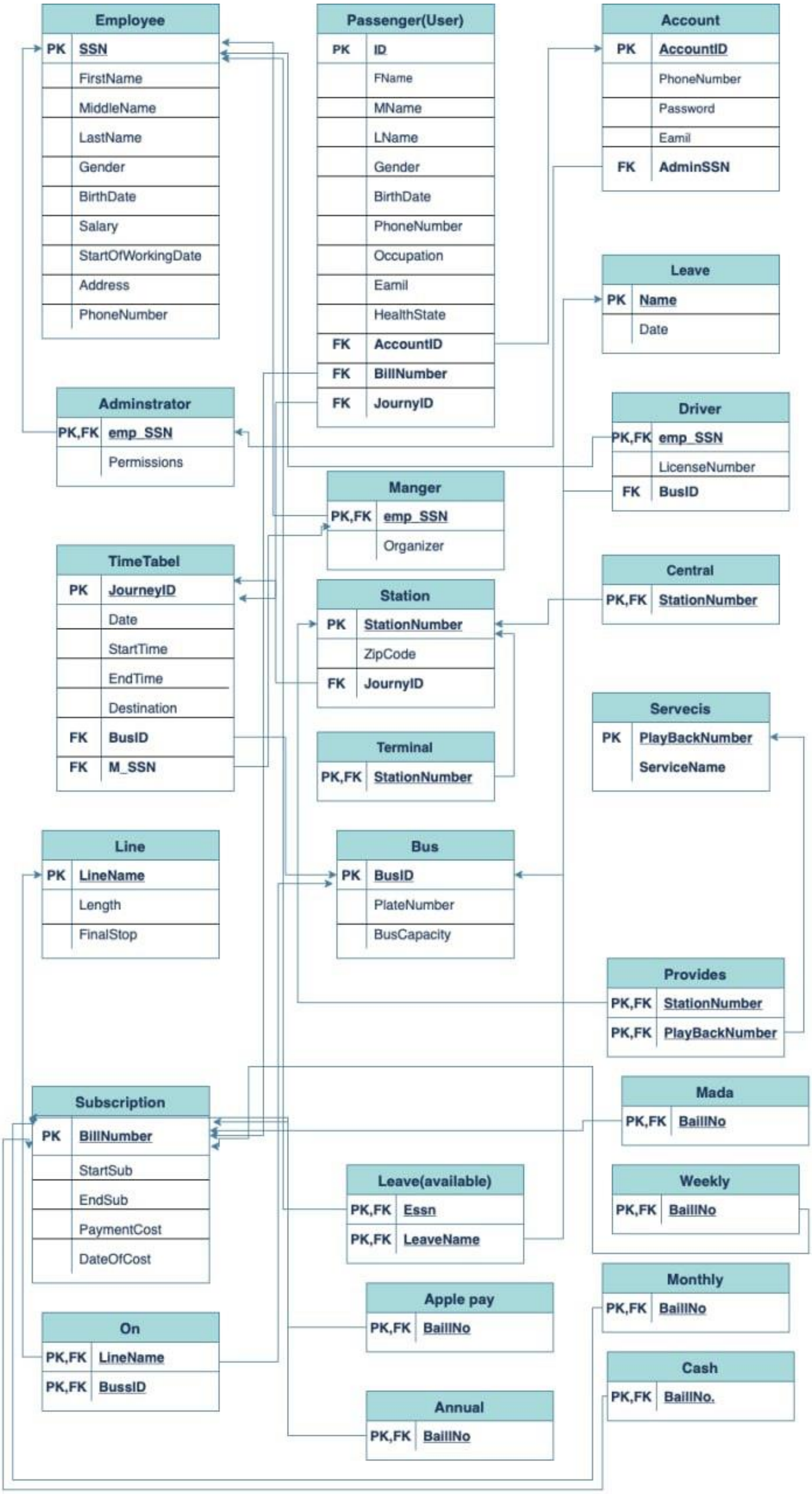
3.6 Mapping of Multivalued Attributes

- We do not have a multivalued attributes in our system.

3.7 Mapping of N-ary Relationship Types

- We do not have a weak entity in our system.

3.8 Schema Diagram


Employee
PK SSN
FirstName
MiddleName
LastName
Gender
BirthDate
Salary
StartOfWorkingDate
Address
PhoneNumber
Passenger(User)
PK ID
FName
MName
LName
Gender
BirthDate
PhoneNumber
Occupation
Eamil
HealthState
FK AccountID
FK BillNumber
FK JournyID
Account
PK AccountID
PhoneNumber
Password
Eamil
FK AdminSSN
Leave
PK Name
Date
Adminstrator
PK,FK emp_SSN
Permissions
Driver
PK,FK emp_SSN
LicenseNumber
FK BusID
Manger
PK,FK emp_SSN
Organizer
TimeTabel
PK JourneyID
Date
StartTime
EndTime
Destination
FK BusID
FK M_SSN
Central
PK,FK StationNumber
Station
PK StationNumber
ZipCode
FK JournyID
Servecis
PK PlayBackNumber
ServiceName
Terminal
PK,FK StationNumber
Line
PK LineName
Length
FinalStop
Bus
PK BusID
PlateNumber
BusCapacity
Provides
PK,FK StationNumber
PK,FK PlayBackNumber
Mada
PK,FK BaillNo
Subscription
PK BillNumber
StartSub
EndSub
PaymentCost
DateOfCost
Leave(available)
PK,FK Essn
PK,FK LeaveName
Weekly
PK,FK BaillNo
Apple pay
PK,FK BaillNo
Monthly
PK,FK BaillNo
On
PK,FK LineName
PK,FK BussID
Cash
PK,FK BaillNo.
Annual
PK,FK BaillNo

4 Normalization

4.1 First Normal Form

In the first normal form, we don’t have composite, multivalued attributes, and nested relations. Therefore, the attributes should be with there atomic values in their domains. As we can see, our relational schema has a simple attribute so, we conclude that our relations do not violate any guidelines for the first normal form.

Account

AccountID	Email	PhoneNumber	Password	ASSN

Passenger/User

ID Number	BirthDate	Gender	First	Middle	Last	Occupation	Email	health State	PhoneNum ber	Account ID	Bill no.	JournyID

Subscription

Bill no.	Date of payment	Start subscription	End subscription	Payment cost

Leave

Name	Date

Employee

SSN	First	Middle	Last	Gender	BirthDate	StartOfWorkDate	adress	salary	PhoneNumber

Timetable

JournyID	Date	StartTime	EndTime	Destination	MSSN	BusID

Bus

BusID	PlateNumber	BusCapacity

Station

StationNumber	ZipCode	JournyID

Line

LineName	FinalStop	Length

Service

PlaybackNumber	ServiceName

Available

ESSN	LeaveName

On

BusID	LineName

Provide

StationNumber	PlaybackNumber

Adminstrator

ESSN	Permissions

Manager

ESSN	Orgnizer

Driver

ESSN	LicenseNumber	BusID

Annual

BillNumber

Monthly

BillNumber

Weekly

BillNumber

Cash

BillNumber

Apple Pay

BillNumber

Mada

BillNumber

Central Station

StationNumber

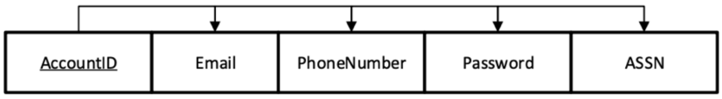
Terminal Station

StationNumber

4.2 Second Normal Form

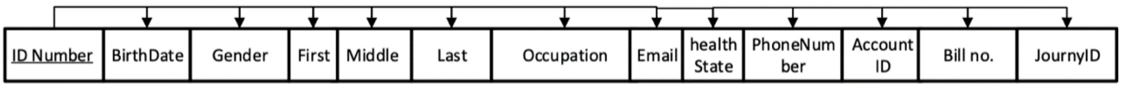
To make our relational schema in Second normal form [2NF], every non-prime attribute should be fully functionally dependent on the primary key. Full FD: means that the removal of the primary key attribute (or an attribute that is part of a PK) results in losing the functional dependency so after applied this stage , we conclude that our relations do not violate any guidelines for the second normal form.

Account



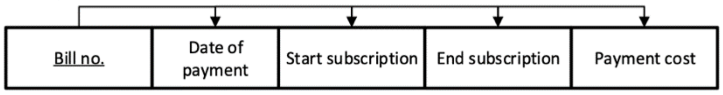
{AccountID}->Email,PhoneNunmber,Password,ASSN →are full FD.

Passenger/User



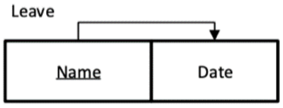
{ID Number}-> BirthDate ,Gender, First, Middle, Last, Occupation, Email, health State, PhoneNumber, AccountID, Bill no. , JournyID →are full FD.

Subscription



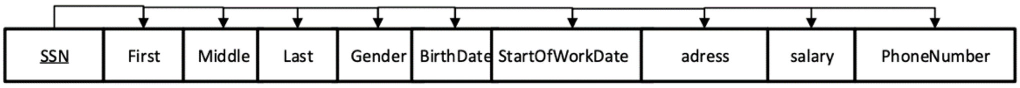
{Bill no.} -> Date of payment, Start subscription , End subscription, Payment cost → are full FD.

Leave

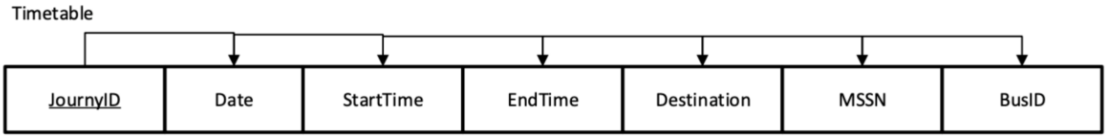


{Name} -> Date → is a full FD.

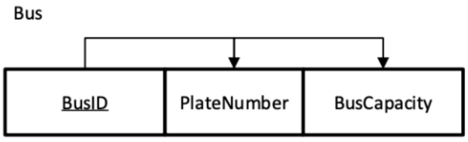
Employee



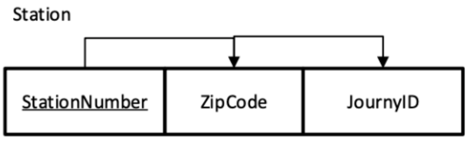
{SSN}-> First, Middle, Last, Gender, BirthDate , StartOfWorkDate, address, salary, PhoneNumber → are full FD.



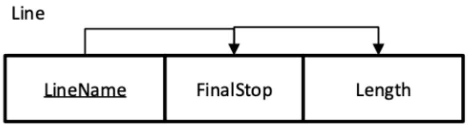
{JournyID}-> Date, StartTime, EndTime, Destination, MSSN, BusID → are full FD.



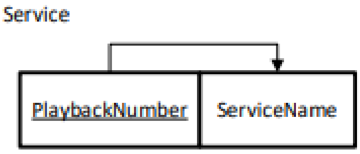
{BusID}-> PlateNumber, BusCapacity → are full FD.



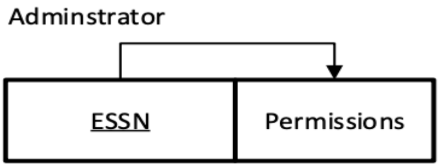
{StationNumber}-> ZipCode, JournyID → are full FD.



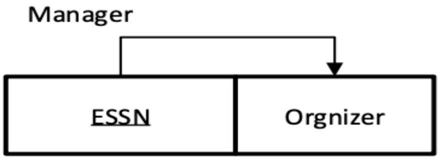
{LineName}-> FinalStop, Length → are full FD.



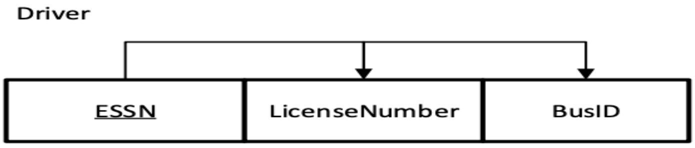
{PlaybackNumber}-> ServiceName → is a full FD.



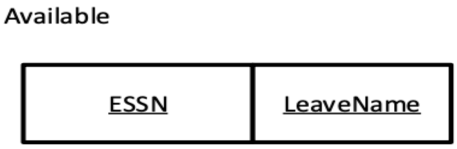
{ESSN}-> Permissions → is a full FD

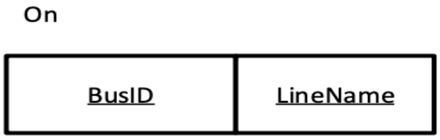


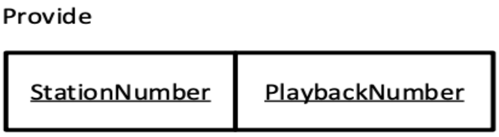
{ESSN}-> Organizer → is a full FD.

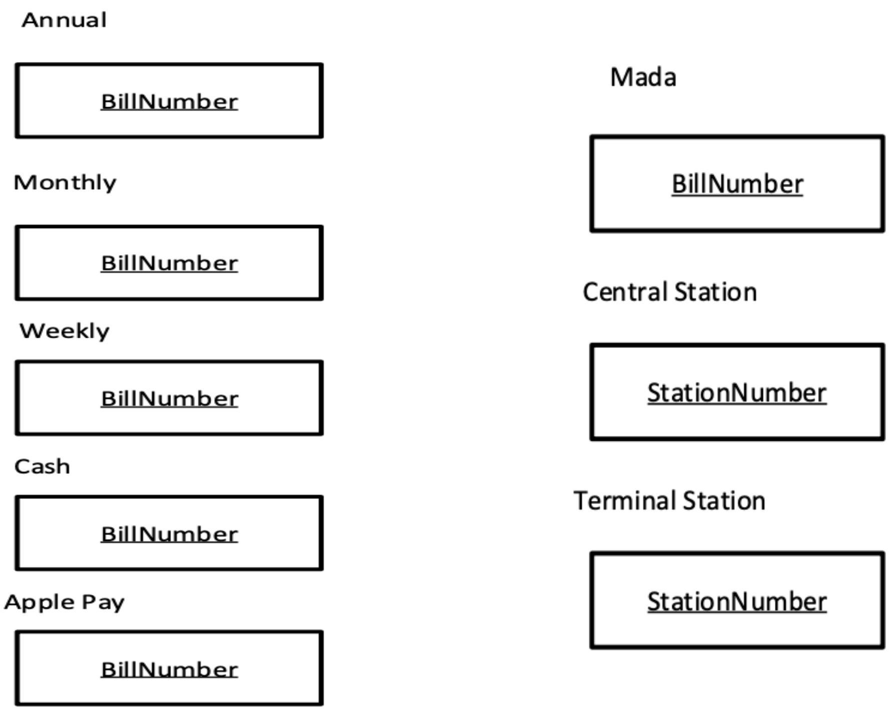


{ESSN}-> LicenseNumber, BusID → are full FD.





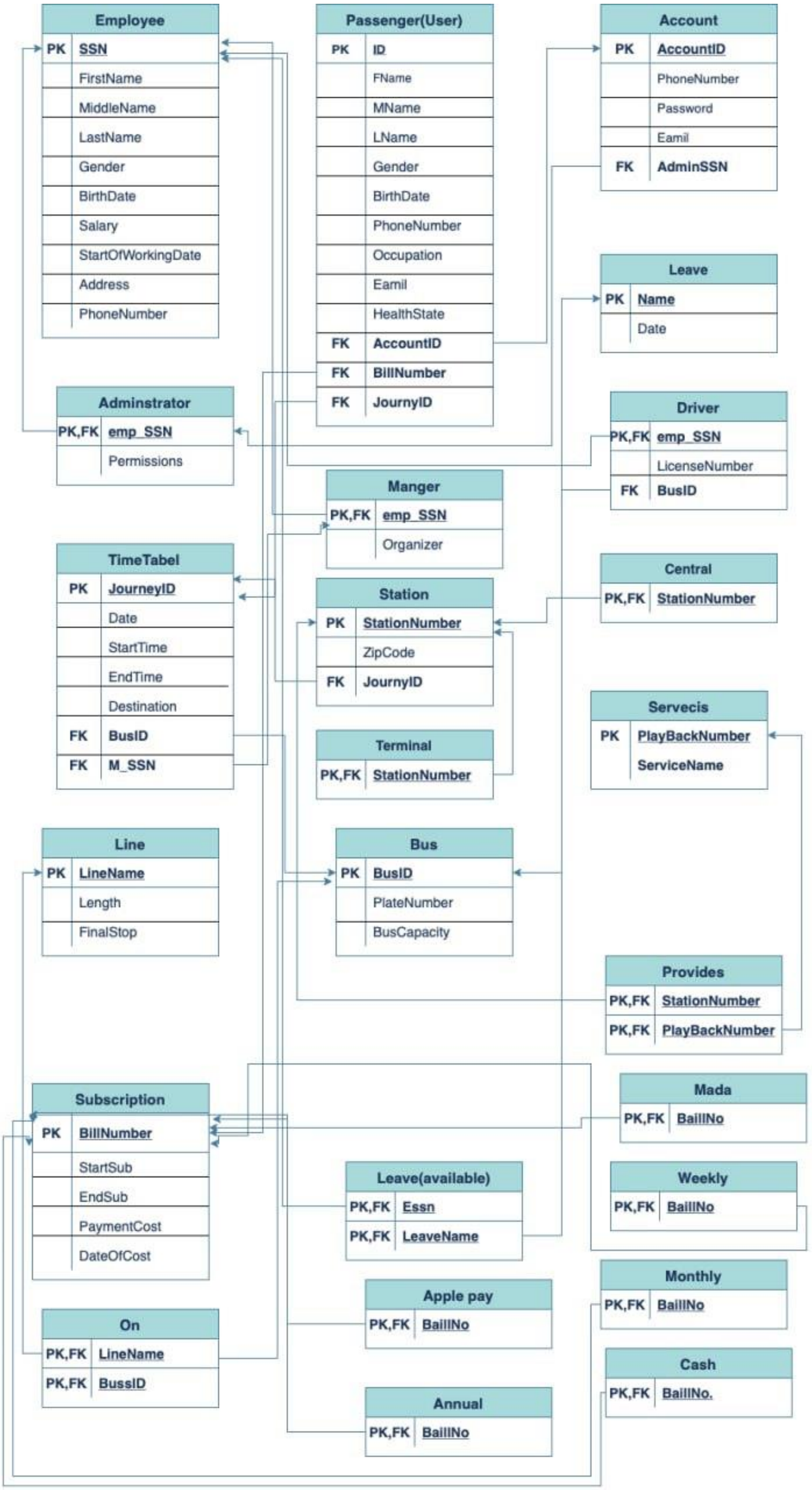




4.3 Third Normal Form

Already all relation on 3rd normal form, because there is no non-prime attribute in any relation that its transitively dependent on another non-prime attribute so, that means its functionally dependent on primary key only.

5 Final DB Schema Diagram

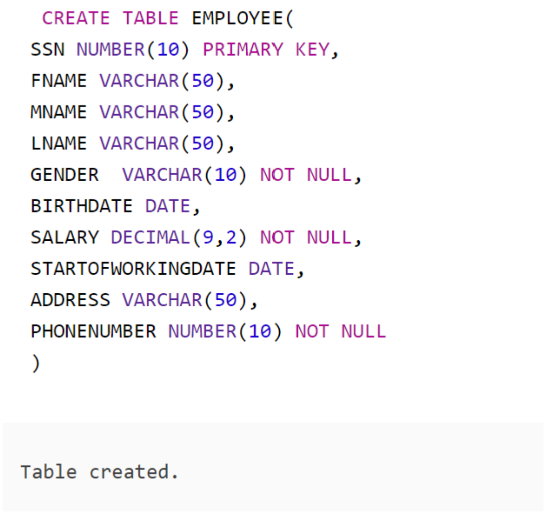


PART III: IMPLEMENTATION

6 Table Creation Script

6.1 *<Employee>* TABLE

```
CREATE TABLE EMPLOYEE(

SSN NUMBER(10) PRIMARY KEY,
FNAME VARCHAR(50),
MNAME VARCHAR(50),
LNAME VARCHAR(50),
GENDER  VARCHAR(10) NOT NULL,
BIRTHDATE DATE,
SALARY DECIMAL(9,2) NOT NULL,
STARTOFWORKINGDATE DATE,
ADDRESS VARCHAR(50),
PHONENUMBER NUMBER(10) NOT NULL
);
```

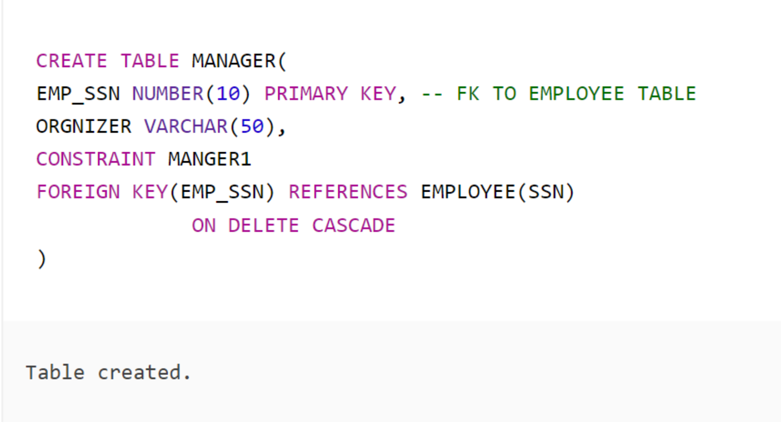


```
 CREATE TABLE EMPLOYEE(
SSN NUMBER(10) PRIMARY KEY,
FNAME VARCHAR(50),
MNAME VARCHAR(50),
LNAME VARCHAR(50),
GENDER  VARCHAR(10) NOT NULL,
BIRTHDATE DATE,
SALARY DECIMAL(9,2) NOT NULL,
STARTOFWORKINGDATE DATE,
ADDRESS VARCHAR(50),
PHONENUMBER NUMBER(10) NOT NULL
)
```

```
Table created.
```

6.2 *<Manager>* TABLE

```
CREATE TABLE MANAGER(
EMP_SSN NUMBER(10) PRIMARY KEY, -- FK TO EMPLOYEE TABLE
ORGNIZER VARCHAR(50),
CONSTRAINT MANGER1
FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE(SSN)
             ON DELETE CASCADE
);
```

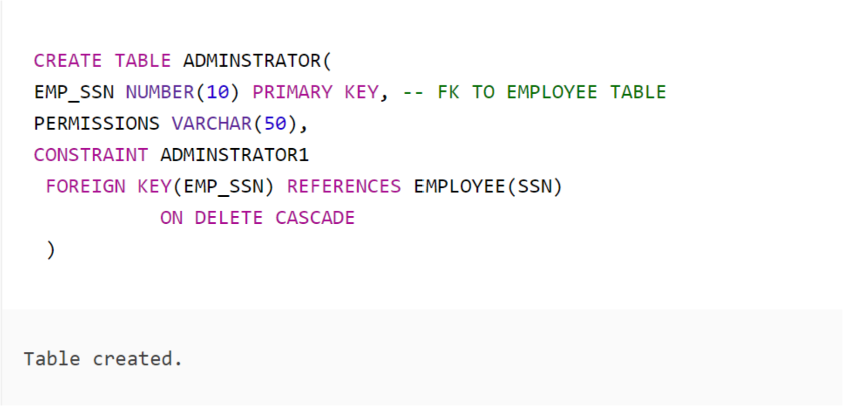


```
CREATE TABLE MANAGER(
EMP_SSN NUMBER(10) PRIMARY KEY, -- FK TO EMPLOYEE TABLE
ORGNIZER VARCHAR(50),
CONSTRAINT MANGER1
FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE(SSN)
             ON DELETE CASCADE
)
```

```
Table created.
```

6.3<*ADMINSTRATOR*> TABLE

```
CREATE TABLE ADMINSTRATOR(
EMP_SSN NUMBER(10) PRIMARY KEY, -- FK TO EMPLOYEE TABLE
PERMISSIONS VARCHAR(50),
CONSTRAINT ADMINSTRATOR1
 FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE(SSN)
           ON DELETE CASCADE

 );
```

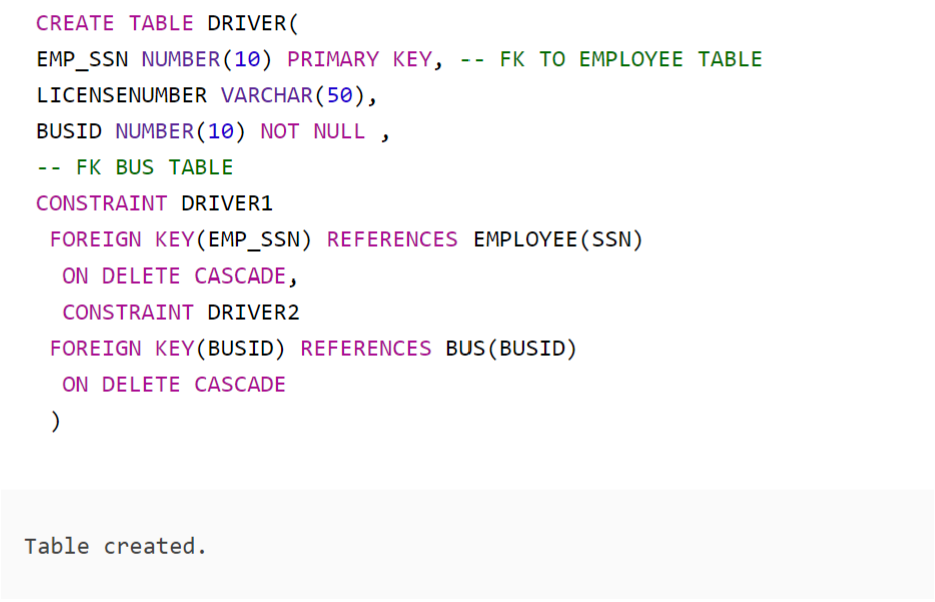


```
CREATE TABLE ADMINSTRATOR(
EMP_SSN NUMBER(10) PRIMARY KEY, -- FK TO EMPLOYEE TABLE
PERMISSIONS VARCHAR(50),
CONSTRAINT ADMINSTRATOR1
 FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE(SSN)
           ON DELETE CASCADE
 )
```

```
Table created.
```

6.4 <*DRIVER*> TABLE

```
CREATE TABLE DRIVER(
EMP_SSN NUMBER(10) PRIMARY KEY, -- FK TO EMPLOYEE TABLE
LICENSENUMBER VARCHAR(50),
BUSID NUMBER(10) NOT NULL ,
-- FK BUS TABLE
CONSTRAINT DRIVER1
 FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE(SSN)
  ON DELETE CASCADE,
  CONSTRAINT DRIVER2
 FOREIGN KEY(BUSID) REFERENCES BUS(BUSID)  ON DELETE CASCADE );
```

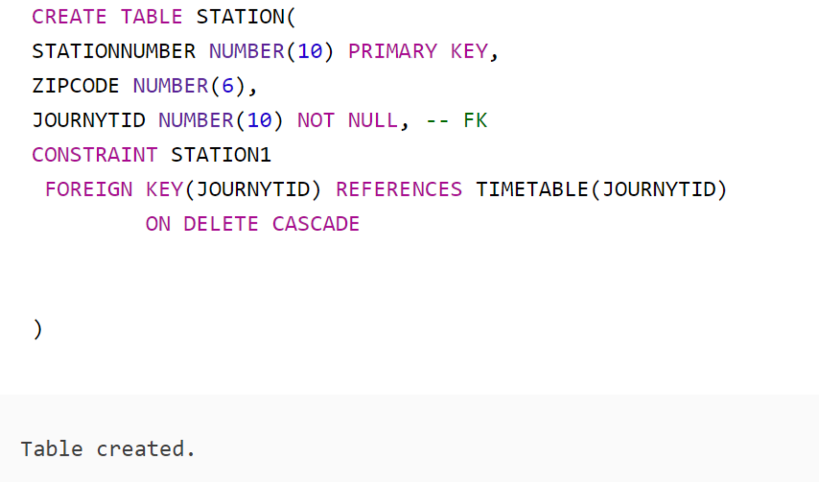


```
CREATE TABLE DRIVER(
EMP_SSN NUMBER(10) PRIMARY KEY, -- FK TO EMPLOYEE TABLE
LICENSENUMBER VARCHAR(50),
BUSID NUMBER(10) NOT NULL ,
-- FK BUS TABLE
CONSTRAINT DRIVER1
 FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE(SSN)
  ON DELETE CASCADE,
  CONSTRAINT DRIVER2
 FOREIGN KEY(BUSID) REFERENCES BUS(BUSID)
  ON DELETE CASCADE
 )
```

```
Table created.
```

6.5 <*STATION*> TABLE

```
CREATE TABLE STATION(
STATIONNUMBER NUMBER(10) PRIMARY KEY,
ZIPCODE NUMBER(6),
JOURNYTID NUMBER(10) NOT NULL, -- FK
CONSTRAINT STATION1
 FOREIGN KEY(JOURNYTID) REFERENCES TIMETABLE(JOURNYTID)
         ON DELETE CASCADE


);
```

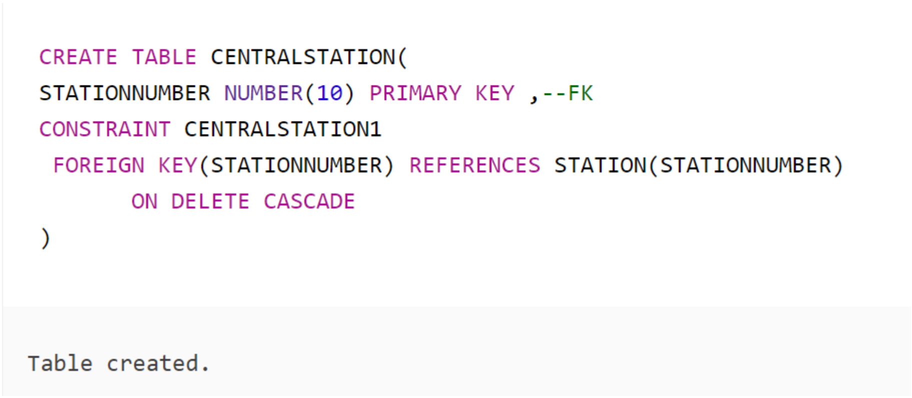


```
CREATE TABLE STATION(
STATIONNUMBER NUMBER(10) PRIMARY KEY,
ZIPCODE NUMBER(6),
JOURNYTID NUMBER(10) NOT NULL, -- FK
CONSTRAINT STATION1
 FOREIGN KEY(JOURNYTID) REFERENCES TIMETABLE(JOURNYTID)
         ON DELETE CASCADE



)
```

```
Table created.
```

6.6 <*CENTRALSTATION*> TABLE

```
CREATE TABLE CENTRALSTATION(
STATIONNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT CENTRALSTATION1
 FOREIGN KEY(STATIONNUMBER) REFERENCES STATION(STATIONNUMBER)  ON DELETE CASCADE );
```

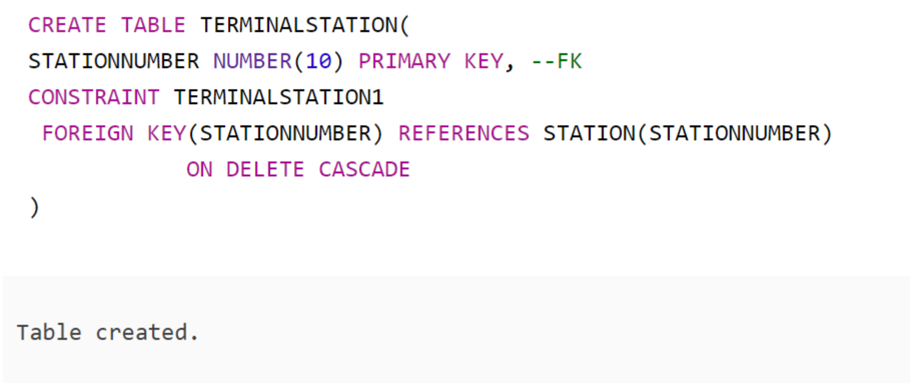


```
CREATE TABLE CENTRALSTATION(
STATIONNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT CENTRALSTATION1
 FOREIGN KEY(STATIONNUMBER) REFERENCES STATION(STATIONNUMBER)
       ON DELETE CASCADE
)
```

```
Table created.
```

6.7<*TERMINALSTATION*> TABLE

```
CREATE TABLE TERMINALSTATION(
STATIONNUMBER NUMBER(10) PRIMARY KEY, --FK
CONSTRAINT TERMINALSTATION1
 FOREIGN KEY(STATIONNUMBER) REFERENCES STATION(STATIONNUMBER)
           ON DELETE CASCADE
);
```

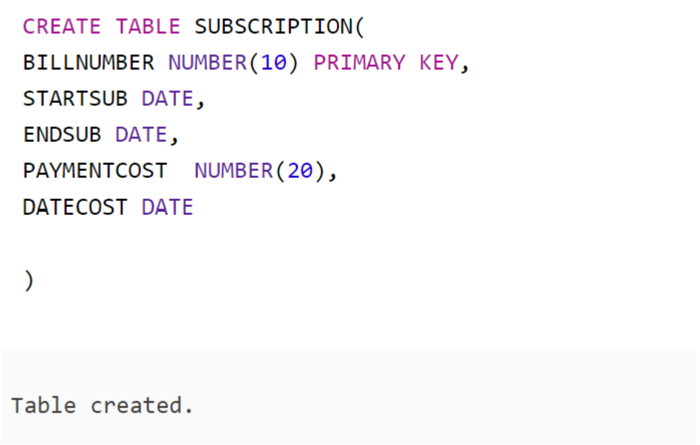


```
CREATE TABLE TERMINALSTATION(
STATIONNUMBER NUMBER(10) PRIMARY KEY, --FK
CONSTRAINT TERMINALSTATION1
 FOREIGN KEY(STATIONNUMBER) REFERENCES STATION(STATIONNUMBER)
           ON DELETE CASCADE
)
```

```
Table created.
```

6.8<*SUBSCRIPTION*> TABLE

```
CREATE TABLE SUBSCRIPTION(
BILLNUMBER NUMBER(10) PRIMARY KEY,
STARTSUB DATE,
ENDSUB DATE,
PAYMENTCOST  NUMBER(20),
DATECOST DATE

);
```

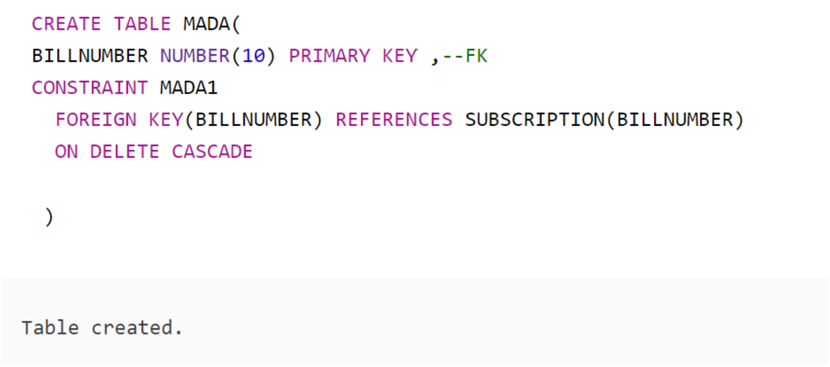


```
CREATE TABLE SUBSCRIPTION(
BILLNUMBER NUMBER(10) PRIMARY KEY,
STARTSUB DATE,
ENDSUB DATE,
PAYMENTCOST  NUMBER(20),
DATECOST DATE

)
```

```
Table created.
```

6.9 <*MADA*> TABLE

```
CREATE TABLE MADA(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT MADA1
  FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
  ON DELETE CASCADE

 );
```

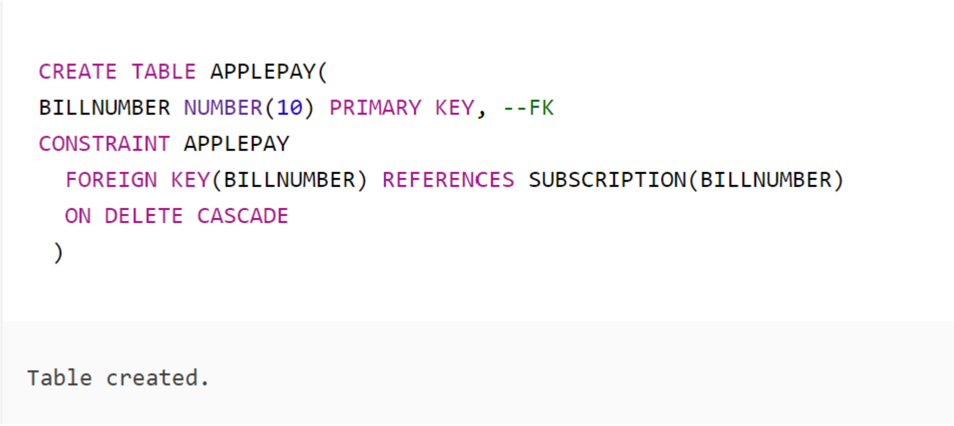


```
CREATE TABLE MADA(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT MADA1
  FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
  ON DELETE CASCADE

 )
```

```
Table created.
```

6.10<*APPLEPAY*> TABLE

```
CREATE TABLE APPLEPAY(

BILLNUMBER NUMBER(10) PRIMARY KEY, --FK
CONSTRAINT APPLEPAY
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
ON DELETE CASCADE
);
```

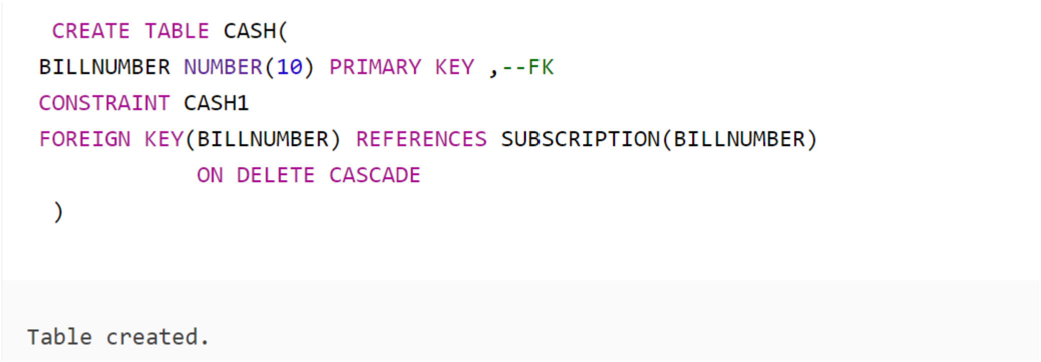


```
CREATE TABLE APPLEPAY(
BILLNUMBER NUMBER(10) PRIMARY KEY, --FK
CONSTRAINT APPLEPAY
  FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
  ON DELETE CASCADE
 )
```

```
Table created.
```

6.11<*CASH*> TABLE

```
CREATE TABLE CASH(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT CASH1
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
            ON DELETE CASCADE
 );
```

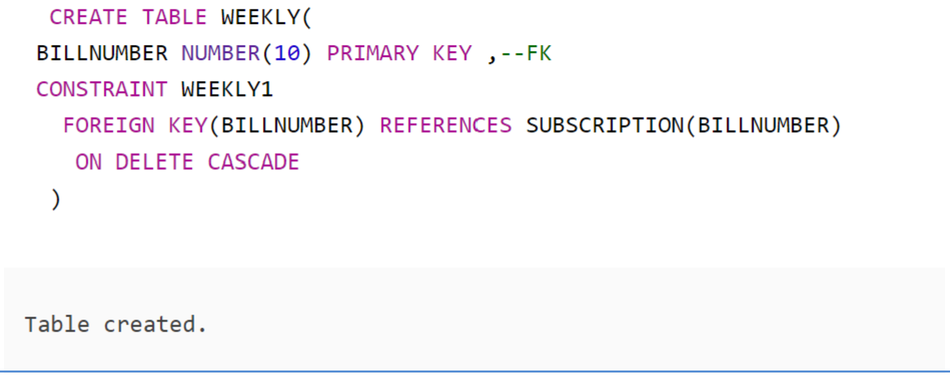


```
 CREATE TABLE CASH(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT CASH1
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
            ON DELETE CASCADE
 )
```

```
Table created.
```

6.12<*WEEKLY*> TABLE

```
CREATE TABLE WEEKLY(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT WEEKLY1
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
ON DELETE CASCADE
);
```

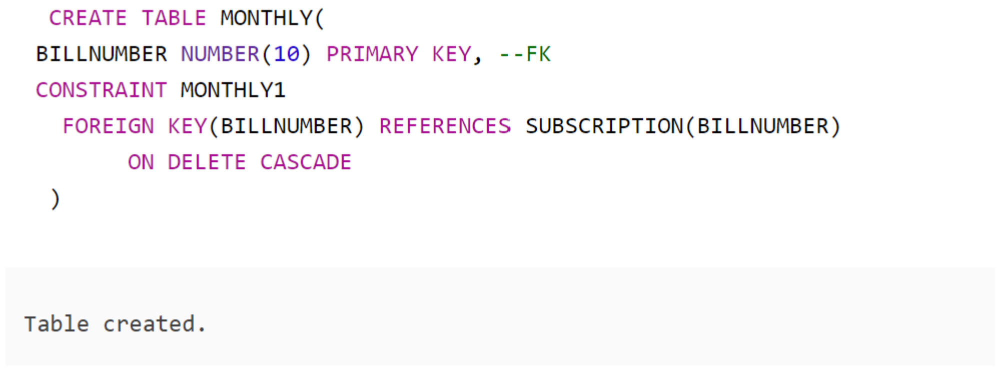


```
 CREATE TABLE WEEKLY(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--FK
CONSTRAINT WEEKLY1
  FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
   ON DELETE CASCADE
 )
```

```
Table created.
```

6.13 <*MONTHLY*> TABLE

```
CREATE TABLE MONTHLY(
BILLNUMBER NUMBER(10) PRIMARY KEY, --FK
CONSTRAINT MONTHLY1
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
ON DELETE CASCADE
);
```

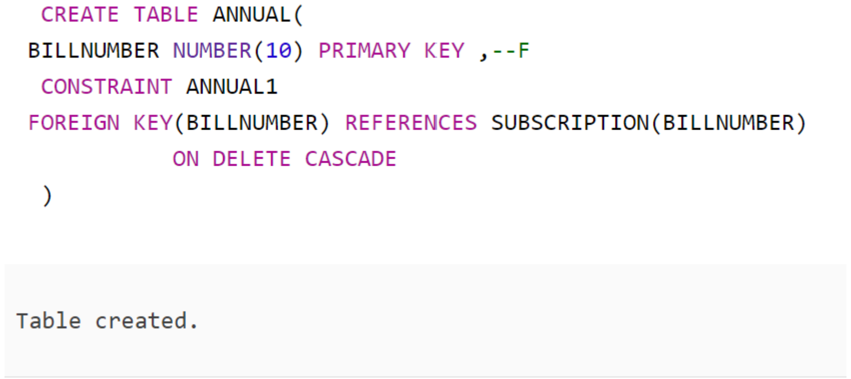


```
 CREATE TABLE MONTHLY(
BILLNUMBER NUMBER(10) PRIMARY KEY, --FK
CONSTRAINT MONTHLY1
  FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
       ON DELETE CASCADE
 )
```

```
Table created.
```

6.14<*ANNUAL*> TABLE

```
CREATE TABLE ANNUAL(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--F
 CONSTRAINT ANNUAL1
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
           ON DELETE CASCADE
 );
```

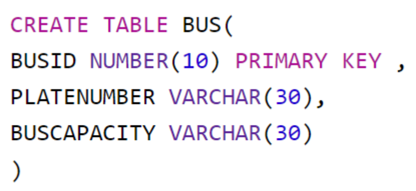


```
 CREATE TABLE ANNUAL(
BILLNUMBER NUMBER(10) PRIMARY KEY ,--F
 CONSTRAINT ANNUAL1
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
           ON DELETE CASCADE
 )
```

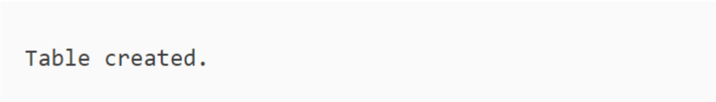
```
Table created.
```

6.15 <*BUS*> TABLE

```
CREATE TABLE BUS(
BUSID NUMBER(10) PRIMARY KEY ,
PLATENUMBER VARCHAR(30),
BUSCAPACITY VARCHAR(30)
);
```



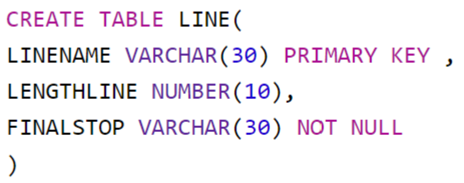
```
CREATE TABLE BUS(
BUSID NUMBER(10) PRIMARY KEY ,
PLATENUMBER VARCHAR(30),
BUSCAPACITY VARCHAR(30)
)
```



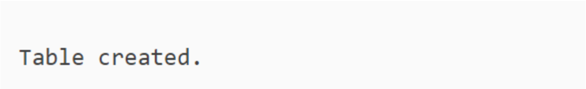
```
Table created.
```

6.16<*LINE*> TABLE

```
CREATE TABLE LINE(
LINENAME VARCHAR(30) PRIMARY KEY ,
LENGTHLINE NUMBER(10),
FINALSTOP VARCHAR(30) NOT NULL
);
```



```
CREATE TABLE LINE(
LINENAME VARCHAR(30) PRIMARY KEY ,
LENGTHLINE NUMBER(10),
FINALSTOP VARCHAR(30) NOT NULL
)
```

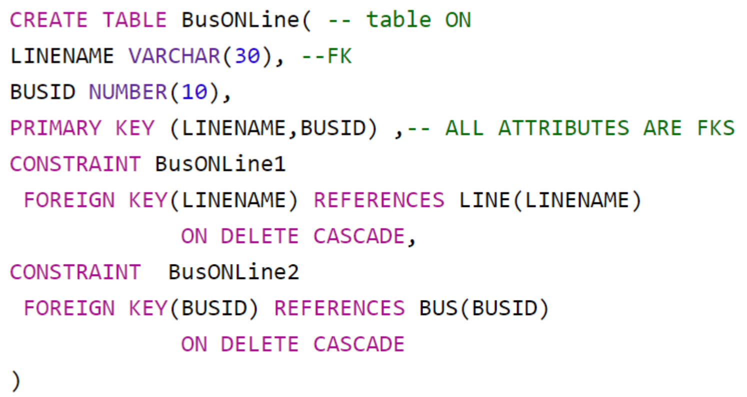


```
Table created.
```

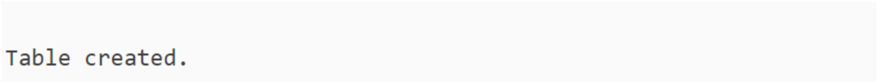
6.17 <*ON*> TABLE

```
CREATE TABLE BusONLine( -- table ON
LINENAME VARCHAR(30), --FK
BUSID NUMBER(10),
PRIMARY KEY (LINENAME,BUSID) ,-- ALL ATTRIBUTES ARE FKS
CONSTRAINT BusONLine1
 FOREIGN KEY(LINENAME) REFERENCES LINE(LINENAME)
             ON DELETE CASCADE,
CONSTRAINT  BusONLine2
 FOREIGN KEY(BUSID) REFERENCES BUS(BUSID)
             ON DELETE CASCADE
);
```

(NOTE: We changed the name only in SQL LIVE because "ON " is a reserved word in the environment)



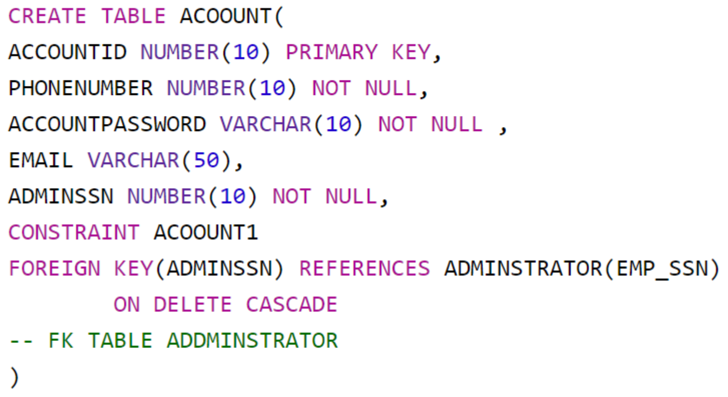
```
CREATE TABLE BusONLine( -- table ON
LINENAME VARCHAR(30), --FK
BUSID NUMBER(10),
PRIMARY KEY (LINENAME,BUSID) ,-- ALL ATTRIBUTES ARE FKS
CONSTRAINT BusONLine1
 FOREIGN KEY(LINENAME) REFERENCES LINE(LINENAME)
             ON DELETE CASCADE,
CONSTRAINT  BusONLine2
 FOREIGN KEY(BUSID) REFERENCES BUS(BUSID)
             ON DELETE CASCADE
)
```



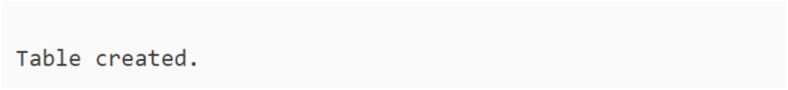
```
Table created.
```

6.18 <*ACCOUNT*> TABLE

```
CREATE TABLE ACOOUNT(
ACCOUNTID NUMBER(10) PRIMARY KEY,
PHONENUMBER NUMBER(10) NOT NULL,
ACCOUNTPASSWORD VARCHAR(10) NOT NULL ,
EMAIL VARCHAR(50),
ADMINSSN NUMBER(10) NOT NULL,
CONSTRAINT ACOOUNT1
FOREIGN KEY(ADMINSSN) REFERENCES ADMINSTRATOR(EMP_SSN)
        ON DELETE CASCADE
-- FK TABLE ADDMINSTRATOR );
```



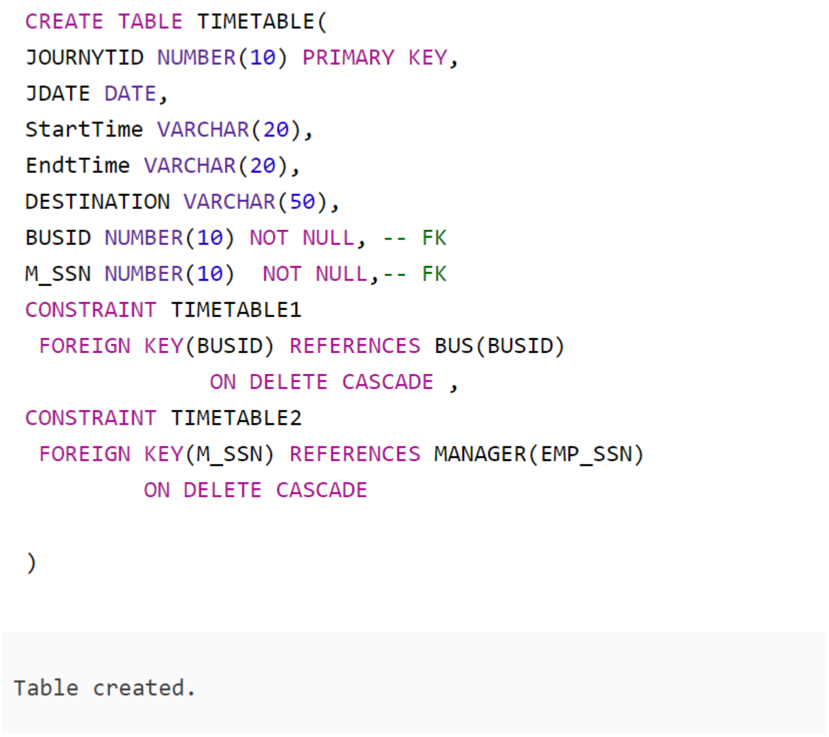
```
CREATE TABLE ACOOUNT(
ACCOUNTID NUMBER(10) PRIMARY KEY,
PHONENUMBER NUMBER(10) NOT NULL,
ACCOUNTPASSWORD VARCHAR(10) NOT NULL ,
EMAIL VARCHAR(50),
ADMINSSN NUMBER(10) NOT NULL,
CONSTRAINT ACOOUNT1
FOREIGN KEY(ADMINSSN) REFERENCES ADMINSTRATOR(EMP_SSN)
        ON DELETE CASCADE
-- FK TABLE ADDMINSTRATOR
)
```



```
Table created.
```

6.19 <*TIMETABLE*> TABLE

```
CREATE TABLE TIMETABLE(
JOURNYTID NUMBER(10) PRIMARY KEY,
JDATE DATE,
StartTime VARCHAR(20),
EndtTime VARCHAR(20),
DESTINATION VARCHAR(50),
BUSID NUMBER(10) NOT NULL, -- FK
M_SSN NUMBER(10)  NOT NULL,-- FK
CONSTRAINT TIMETABLE1
 FOREIGN KEY(BUSID) REFERENCES BUS(BUSID)
              ON DELETE CASCADE ,
CONSTRAINT TIMETABLE2
 FOREIGN KEY(M_SSN) REFERENCES MANAGER(EMP_SSN)
         ON DELETE CASCADE


);
```

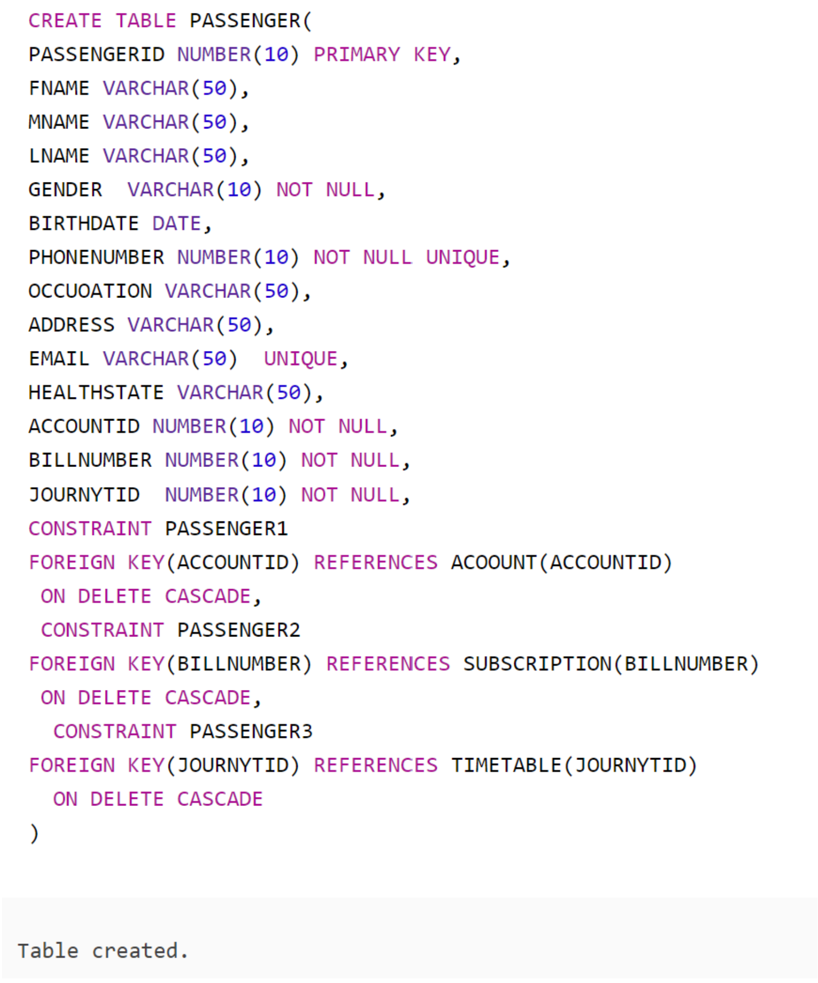


```
CREATE TABLE TIMETABLE(
JOURNYTID NUMBER(10) PRIMARY KEY,
JDATE DATE,
StartTime VARCHAR(20),
EndtTime VARCHAR(20),
DESTINATION VARCHAR(50),
BUSID NUMBER(10) NOT NULL, -- FK
M_SSN NUMBER(10)  NOT NULL,-- FK
CONSTRAINT TIMETABLE1
 FOREIGN KEY(BUSID) REFERENCES BUS(BUSID)
              ON DELETE CASCADE ,
CONSTRAINT TIMETABLE2
 FOREIGN KEY(M_SSN) REFERENCES MANAGER(EMP_SSN)
         ON DELETE CASCADE

)
```

```
Table created.
```

6.20 <*PASSENGER*> TABLE

```
CREATE TABLE PASSENGER(
PASSENGERID NUMBER(10) PRIMARY KEY,
FNAME VARCHAR(50),
MNAME VARCHAR(50),
LNAME VARCHAR(50),
GENDER  VARCHAR(10) NOT NULL,
BIRTHDATE DATE,
PHONENUMBER NUMBER(10) NOT NULL UNIQUE,
OCCUOATION VARCHAR(50),
ADDRESS VARCHAR(50),
EMAIL VARCHAR(50)  UNIQUE,
HEALTHSTATE VARCHAR(50),
ACCOUNTID NUMBER(10) NOT NULL,
BILLNUMBER NUMBER(10) NOT NULL,
JOURNYTID  NUMBER(10) NOT NULL,
CONSTRAINT PASSENGER1
FOREIGN KEY(ACCOUNTID) REFERENCES ACOOUNT(ACCOUNTID)
 ON DELETE CASCADE,
 CONSTRAINT PASSENGER2
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
 ON DELETE CASCADE,
  CONSTRAINT PASSENGER3
FOREIGN KEY(JOURNYTID) REFERENCES TIMETABLE(JOURNYTID)
ON DELETE CASCADE );
```

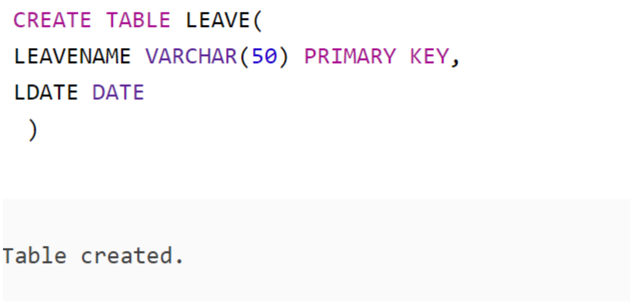


```
CREATE TABLE PASSENGER(
PASSENGERID NUMBER(10) PRIMARY KEY,
FNAME VARCHAR(50),
MNAME VARCHAR(50),
LNAME VARCHAR(50),
GENDER  VARCHAR(10) NOT NULL,
BIRTHDATE DATE,
PHONENUMBER NUMBER(10) NOT NULL UNIQUE,
OCCUOATION VARCHAR(50),
ADDRESS VARCHAR(50),
EMAIL VARCHAR(50)  UNIQUE,
HEALTHSTATE VARCHAR(50),
ACCOUNTID NUMBER(10) NOT NULL,
BILLNUMBER NUMBER(10) NOT NULL,
JOURNYTID  NUMBER(10) NOT NULL,
CONSTRAINT PASSENGER1
FOREIGN KEY(ACCOUNTID) REFERENCES ACOOUNT(ACCOUNTID)
 ON DELETE CASCADE,
 CONSTRAINT PASSENGER2
FOREIGN KEY(BILLNUMBER) REFERENCES SUBSCRIPTION(BILLNUMBER)
 ON DELETE CASCADE,
  CONSTRAINT PASSENGER3
FOREIGN KEY(JOURNYTID) REFERENCES TIMETABLE(JOURNYTID)
  ON DELETE CASCADE
)
```

```
Table created.
```

6.21 <*LEAVE*> TABLE

```
CREATE TABLE LEAVE(
LEAVENAME VARCHAR(50) PRIMARY KEY,
LDATE DATE
 );
```

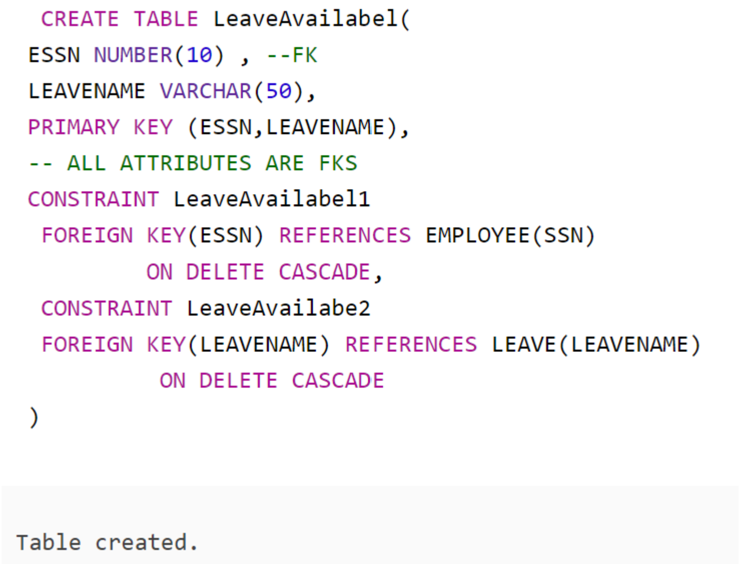


```
CREATE TABLE LEAVE(
LEAVENAME VARCHAR(50) PRIMARY KEY,
LDATE DATE
 )
```

```
Table created.
```

6.22 <*LEAVEAVAILABLE*> TABLE

```
CREATE TABLE LeaveAvailabel(
ESSN NUMBER(10) , --FK
LEAVENAME VARCHAR(50),
PRIMARY KEY (ESSN,LEAVENAME),
-- ALL ATTRIBUTES ARE FKS
CONSTRAINT LeaveAvailabel1
 FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN)
         ON DELETE CASCADE,
 CONSTRAINT LeaveAvailabe2
 FOREIGN KEY(LEAVENAME) REFERENCES LEAVE(LEAVENAME)
          ON DELETE CASCADE
);
```

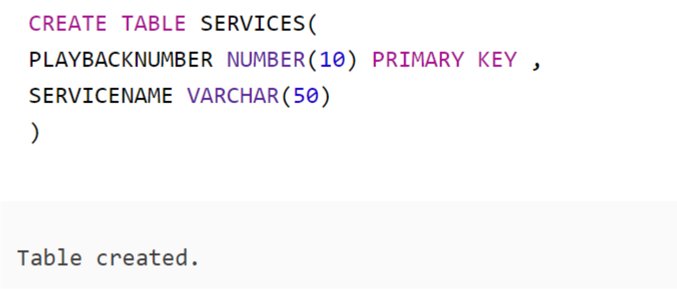


```
 CREATE TABLE LeaveAvailabel(
ESSN NUMBER(10) , --FK
LEAVENAME VARCHAR(50),
PRIMARY KEY (ESSN,LEAVENAME),
-- ALL ATTRIBUTES ARE FKS
CONSTRAINT LeaveAvailabel1
 FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN)
         ON DELETE CASCADE,
 CONSTRAINT LeaveAvailabe2
 FOREIGN KEY(LEAVENAME) REFERENCES LEAVE(LEAVENAME)
          ON DELETE CASCADE
)
```

```
Table created.
```

6.23 <*SERVICES*> TABLE

```
CREATE TABLE SERVICES(
PLAYBACKNUMBER NUMBER(10) PRIMARY KEY ,
SERVICENAME VARCHAR(50) );
```

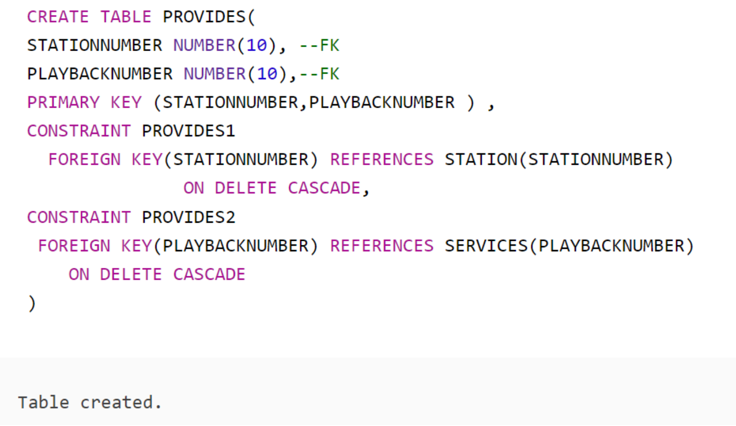


```
CREATE TABLE SERVICES(
PLAYBACKNUMBER NUMBER(10) PRIMARY KEY ,
SERVICENAME VARCHAR(50)
)
```

```
Table created.
```

6.24 <*PROVIDES*> TABLE

```
CREATE TABLE PROVIDES(
STATIONNUMBER NUMBER(10), --FK
PLAYBACKNUMBER NUMBER(10),--FK
PRIMARY KEY (STATIONNUMBER,PLAYBACKNUMBER ) ,
CONSTRAINT PROVIDES1
  FOREIGN KEY(STATIONNUMBER) REFERENCES STATION(STATIONNUMBER)
               ON DELETE CASCADE,
CONSTRAINT PROVIDES2
 FOREIGN KEY(PLAYBACKNUMBER) REFERENCES SERVICES(PLAYBACKNUMBER)
    ON DELETE CASCADE
);
```



```
CREATE TABLE PROVIDES(
STATIONNUMBER NUMBER(10), --FK
PLAYBACKNUMBER NUMBER(10),--FK
PRIMARY KEY (STATIONNUMBER,PLAYBACKNUMBER ) ,
CONSTRAINT PROVIDES1
  FOREIGN KEY(STATIONNUMBER) REFERENCES STATION(STATIONNUMBER)
               ON DELETE CASCADE,
CONSTRAINT PROVIDES2
 FOREIGN KEY(PLAYBACKNUMBER) REFERENCES SERVICES(PLAYBACKNUMBER)
    ON DELETE CASCADE
)
```

```
Table created.
```

7 Constraints Script

In this section we well focus in some constraints we have involved in our database. Since these are not all constraints that exist in our database. We are only focus to the main constraints.

Business Rule	SQL Script	Table
Each employee has a unique SSN to distinguish between other employee (Tuple). (Entity integrity)	`SSN NUMBER(10) PRIMARY KEY`	Employee

Manager SSN in the Manager table (Referencing Relation) should be reference to exist SSN (PK) in the Employee table (Referenced Relation).
If the constraint violated, in delete, if the SSN of Employee has deleted from the table it should be also delete from the referencing relation which is Manager table. The appropriate option to prevent the occurrence of the referential integrity is CASCADE.
	`CREATE TABLE MANAGER(`
	`EMP_SSN NUMBER(10) PRIMARY`
	`KEY, -- FK TO EMPLOYEE TABLE`
	`ORGNIZER VARCHAR(50),`
	`CONSTRAINT MANGER1`
	`FOREIGN KEY(EMP_SSN)`
	`REFERENCES EMPLOYEE(SSN)`
	`            ON DELETE`
	`CASCADE`
	`);`
		Manager

Each Passenger has a different email and phone number. If two passengers have the same data the system will reject the command.	• `PHONENUMBER NUMBER(10) NOT NULL UNIQUE.` • `EMAIL VARCHAR(50) UNIQUE.`	Passenger
Each Station has a ZipCode with a numeric data type. If	`ZIPCODE NUMBER(6)`	Station

the entered attribute was string the system automatically rejects the insertion.		

8 Queries and Transactions

In this section we will show the most important queries which are select, update, and delete also includes some aggregate functions and nested query.

8.1 *<CHANGE MANAGER: Update Example>*

<u>Query in Natural Language (English)</u>

If the manager has serviced for 5 years, another manager will be hired. The inner query will executed first and return the SSN of employee who starts in '01-jan-2017', then the outer query will set the data of new manager which exists in the employee.

<u>SQL Script</u>

update MANAGER

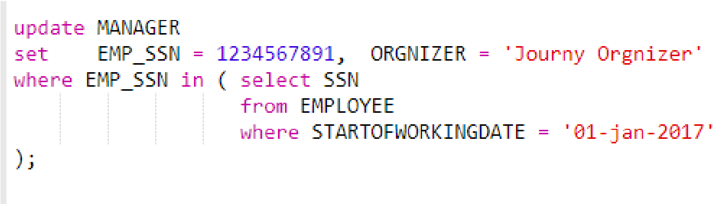
set EMP_SSN = 1234567891, ORGNIZER = 'Journy Orgnizer'

where EMP_SSN in (select SSN

from EMPLOYEE

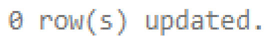
where STARTOFWORKINGDATE = '01-jan-2017'

);



```
update MANAGER
set    EMP_SSN = 1234567891,  ORGNIZER = 'Journy Orgnizer'
where EMP_SSN in ( select SSN
                   from EMPLOYEE
                   where STARTOFWORKINGDATE = '01-jan-2017'
);
```

<u>Caption of the First Five Rows of the Output</u>



```
0 row(s) updated.
```

8.2 < Number of employees below average salary: Select Example >

Query in Natural Language (English)

The inner query will return the average salary, and then the outer query counts the number of employees whose salary below the average salary.

SQL Script

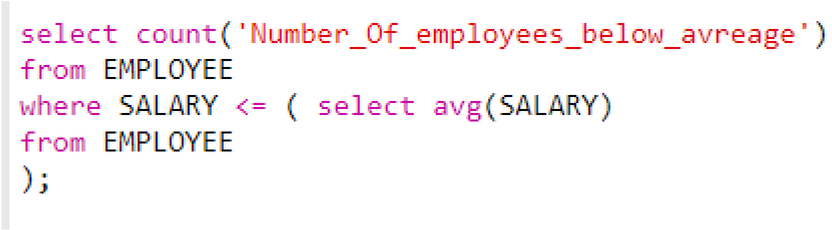
select count('Number_Of_employees_below_avreage')

from EMPLOYEE

where SALARY <= (select avg(SALARY)

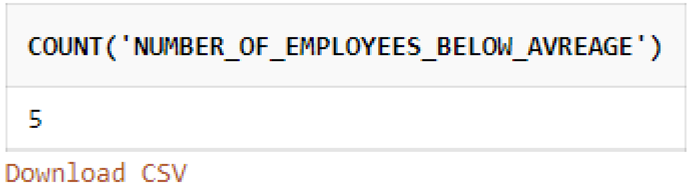
from EMPLOYEE

);



```
select count('Number_Of_employees_below_avreage')
from EMPLOYEE
where SALARY <= ( select avg(SALARY)
from EMPLOYEE
);
```

Caption of the First Five Rows of the Output



COUNT('NUMBER_OF_EMPLOYEES_BELOW_AVREAGE')
5

8.3 <Discount for special needs : Update Example >

Query in Natural Language (English)

A user who has a special need gets discount by 70%. First the inner query will return the bill numbers for the passenger who has a special needs state, then the outer query will set discount payment cost of the given list.

SQL Script

UPDATE SUBSCRIPTION

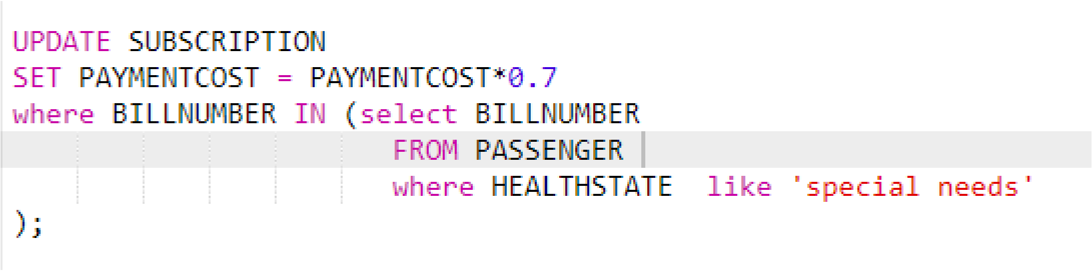
SET PAYMENTCOST = PAYMENTCOST*0.7

where BILLNUMBER IN (select BILLNUMBER

FROM PASSENGER

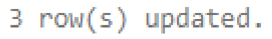
where HEALTHSTATE like 'special needs'

);



```
UPDATE SUBSCRIPTION
SET PAYMENTCOST = PAYMENTCOST*0.7
where BILLNUMBER IN (select BILLNUMBER
                        FROM PASSENGER
                        where HEALTHSTATE  like 'special needs'
);
```

Caption of the First Five Rows of the Output



```
3 row(s) updated.
```

8.4 < Discount for students : Update Example >

Query in Natural Language (English)

A passenger who is a student gets 85% discount. Query will return the list of passengers who are students, then set the discount payment cost of the given list.

SQL Script

UPDATE SUBSCRIPTION

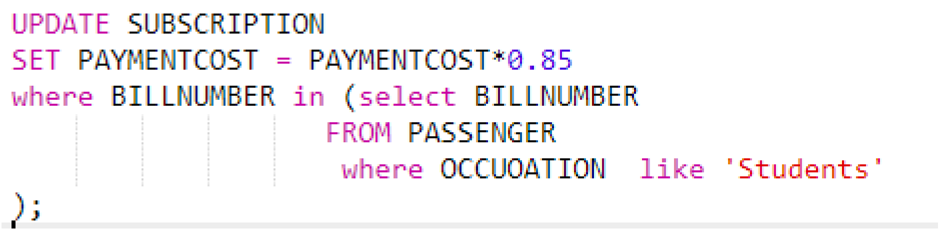
SET PAYMENTCOST = PAYMENTCOST*0.85

where BILLNUMBER in (select BILLNUMBER

FROM PASSENGER

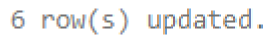
where OCCUOATION like 'Students'

);



```
UPDATE SUBSCRIPTION
SET PAYMENTCOST = PAYMENTCOST*0.85
where BILLNUMBER in (select BILLNUMBER
                    FROM PASSENGER
                     where OCCUOATION  like 'Students'
);
```

<u>Caption of the First Five Rows of the Output</u>



```
6 row(s) updated.
```

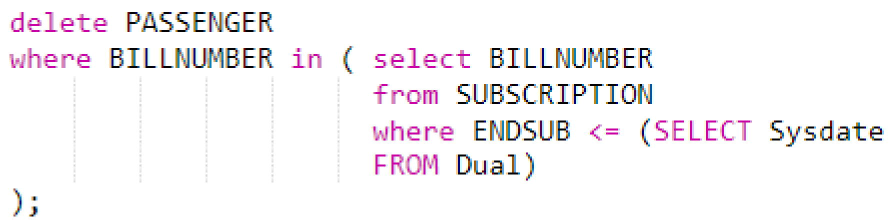
8.5 *<Subscription Expired : DELTE EXAMPLE >*

<u>Query in Natural Language (English)</u>

All expired subscriptions will be deleted. To delete passenger whose subscription expired we need to know the bill number from the subscription table where the end subscription date has expired by comparing the current date using SYSDATE function.

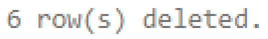
<u>SQL Script</u>

```
delete PASSENGER

where BILLNUMBER in ( select BILLNUMBER

         from SUBSCRIPTION

         where ENDSUB <= (SELECT Sysdate

         FROM Dual)

);
```



```
delete PASSENGER
where BILLNUMBER in ( select BILLNUMBER
                      from SUBSCRIPTION
                      where ENDSUB <= (SELECT Sysdate
                      FROM Dual)
);
```

<u>Caption of the First Five Rows of the Output</u>



```
6 row(s) deleted.
```

8.6 Update Example (We have already used)

Update in Natural Language (English)

SQL Script

Caption of the Output

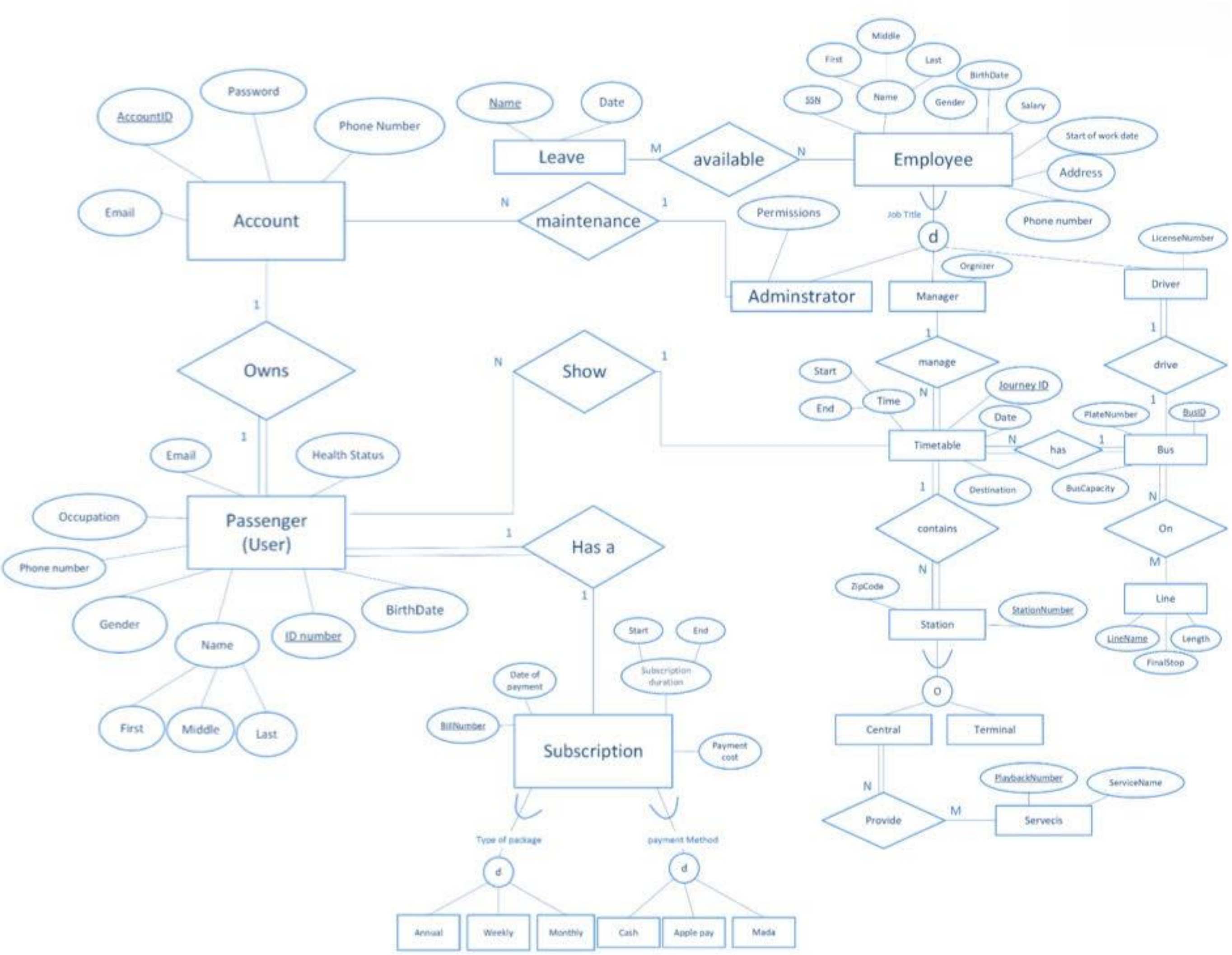
8.7 Delete Example (We have already used)

Delete in Natural Language (English)

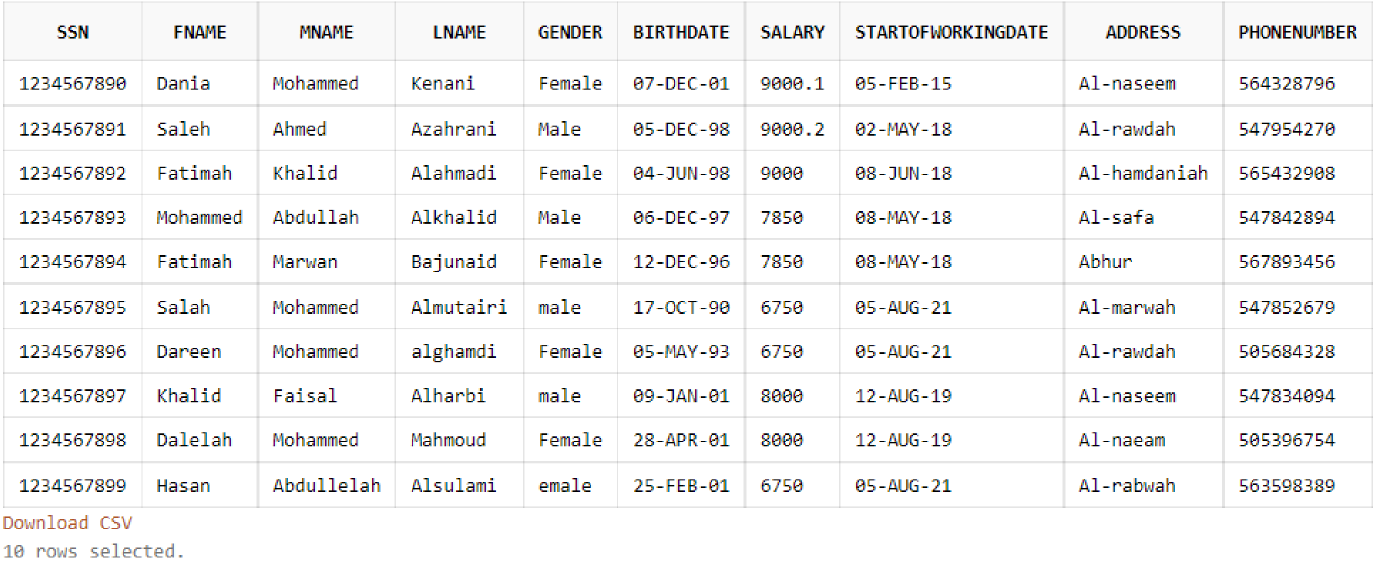
SQL Script

Caption of the Output

APPENDIX


Password
AccountID
Phone Number
Email
Account
Name
Date
Leave
M
available
N
Middle
First
Last
BirthDate
SSN
Name
Gender
Salary
Employee
Start of work date
Address
Phone number
N
maintenance
1
Permissions
Job Title
d
LicenseNumber
Orgnizer
Adminstrator
Manager
Driver
1
Owns
1
N
Show
1
1
manage
1
drive
Start
End
Time
N
Journey ID
Date
PlateNumber
BusID
1
Email
Health Status
Timetable
N
has
1
Bus
Occupation
Passenger
(User)
1
Destination
BusCapacity
N
Phone number
1
Has a
contains
On
M
1
N
Gender
Name
ID number
BirthDate
ZipCode
Line
Station
StationNumber
LineName
Length
FinalStop
Start
End
Date of
payment
Subscription
duration
o
First
Middle
Last
BillNumber
Subscription
Payment
cost
Central
Terminal
PlaybackNumber
ServiceName
N
M
Provide
Servecis
Type of package
payment Method
d
d
Annual
Weekly
Monthly
Cash
Apple pay
Mada

EMPLOYEE TABLE

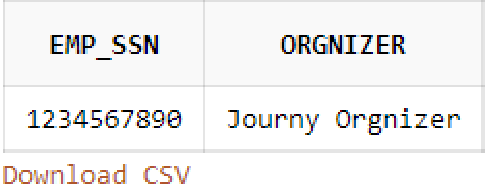


SSN	FNAME	MNAME	LNAME	GENDER	BIRTHDATE	SALARY	STARTOFWORKINGDATE	ADDRESS	PHONENUMBER
1234567890	Dania	Mohammed	Kenani	Female	07-DEC-01	9000.1	05-FEB-15	Al-naseem	564328796
1234567891	Saleh	Ahmed	Azahrani	Male	05-DEC-98	9000.2	02-MAY-18	Al-rawdah	547954270
1234567892	Fatimah	Khalid	Alahmadi	Female	04-JUN-98	9000	08-JUN-18	Al-hamdaniah	565432908
1234567893	Mohammed	Abdullah	Alkhalid	Male	06-DEC-97	7850	08-MAY-18	Al-safa	547842894
1234567894	Fatimah	Marwan	Bajunaid	Female	12-DEC-96	7850	08-MAY-18	Abhur	567893456
1234567895	Salah	Mohammed	Almutairi	male	17-OCT-90	6750	05-AUG-21	Al-marwah	547852679
1234567896	Dareen	Mohammed	alghamdi	Female	05-MAY-93	6750	05-AUG-21	Al-rawdah	505684328
1234567897	Khalid	Faisal	Alharbi	male	09-JAN-01	8000	12-AUG-19	Al-naseem	547834094
1234567898	Dalelah	Mohammed	Mahmoud	Female	28-APR-01	8000	12-AUG-19	Al-naeam	505396754
1234567899	Hasan	Abdullelah	Alsulami	emale	25-FEB-01	6750	05-AUG-21	Al-rabwah	563598389

Download CSV

10 rows selected.

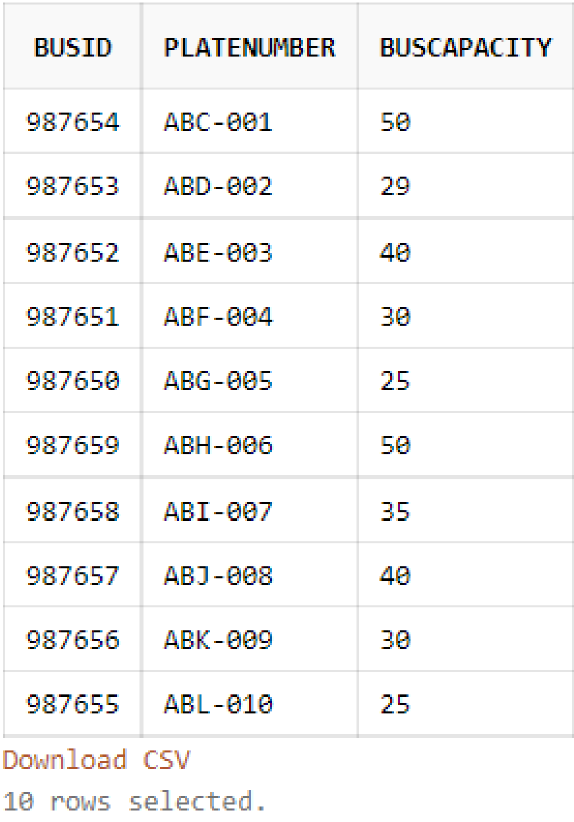
MANAGER TABLE



EMP_SSN	ORGNIZER
1234567890	Journy Orgnizer

Download CSV

BUS TABLE

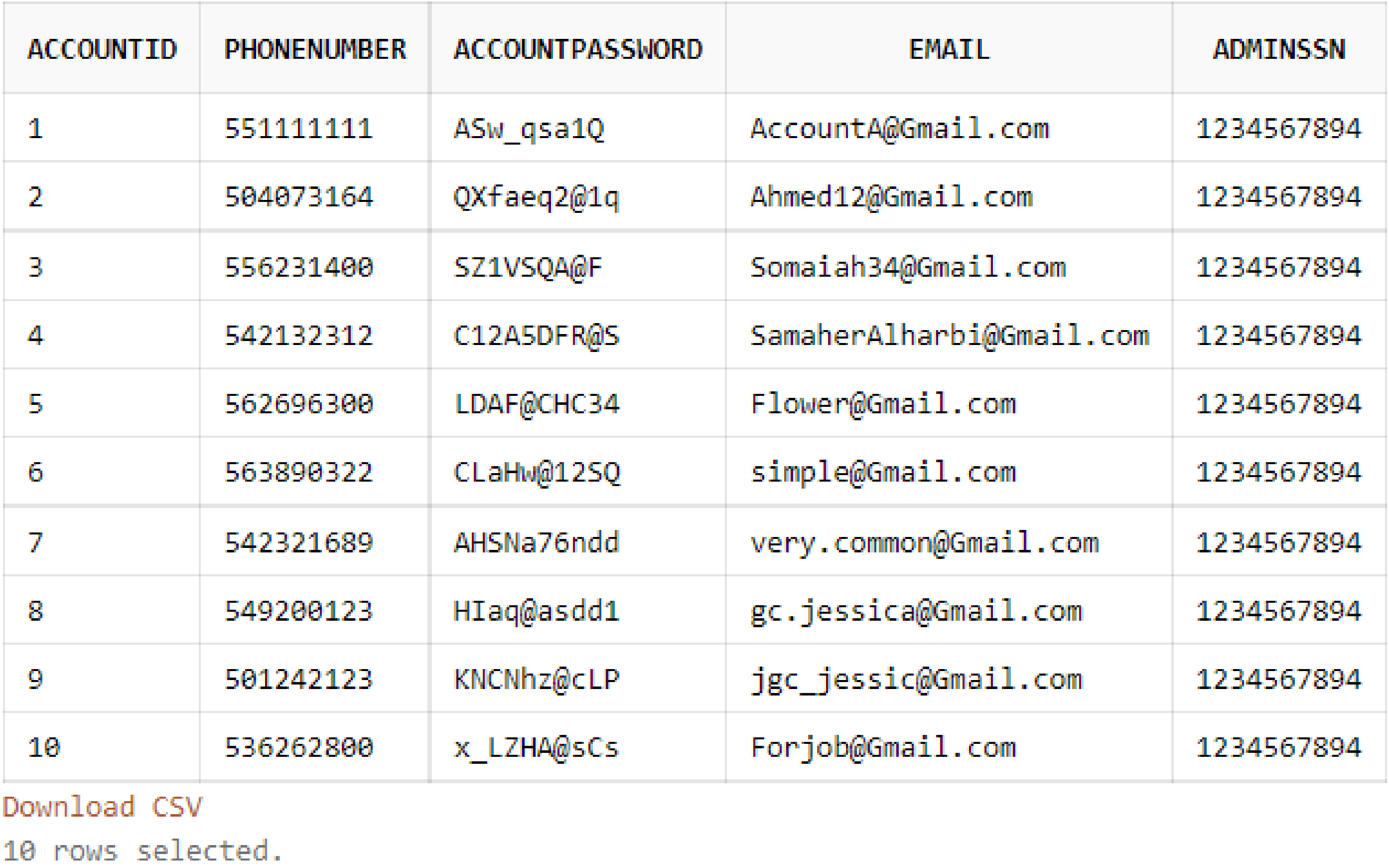


BUSID	PLATENUMBER	BUSCAPACITY
987654	ABC-001	50
987653	ABD-002	29
987652	ABE-003	40
987651	ABF-004	30
987650	ABG-005	25
987659	ABH-006	50
987658	ABI-007	35
987657	ABJ-008	40
987656	ABK-009	30
987655	ABL-010	25

Download CSV

10 rows selected.

ACCONT TABLE

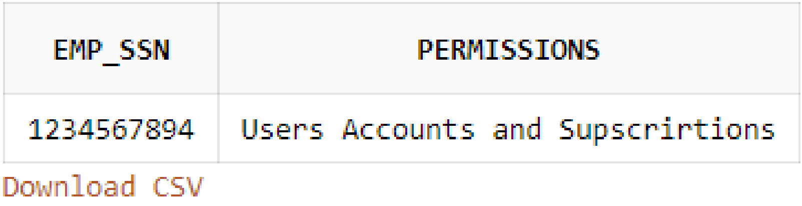


ACCOUNTID	PHONENUMBER	ACCOUNTPASSWORD	EMAIL	ADMINSSN
1	551111111	ASw_qsa1Q	AccountA@Gmail.com	1234567894
2	504073164	QXfaeq2@1q	Ahmed12@Gmail.com	1234567894
3	556231400	SZ1VSQA@F	Somaiah34@Gmail.com	1234567894
4	542132312	C12A5DFR@S	SamaherAlharbi@Gmail.com	1234567894
5	562696300	LDAF@CHC34	Flower@Gmail.com	1234567894
6	563890322	CLaHw@12SQ	simple@Gmail.com	1234567894
7	542321689	AHSNa76ndd	very.common@Gmail.com	1234567894
8	549200123	HIaq@asdd1	gc.jessica@Gmail.com	1234567894
9	501242123	KNCNhz@cLP	jgc_jessic@Gmail.com	1234567894
10	536262800	x_LZHA@sCs	Forjob@Gmail.com	1234567894

Download CSV

10 rows selected.

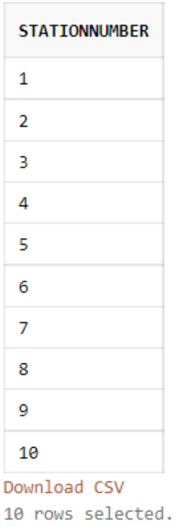
ADDMINSTRATOR TABLE



EMP_SSN	PERMISSIONS
1234567894	Users Accounts and Supscrirtions

Download CSV

CETRAL STATION TABLE

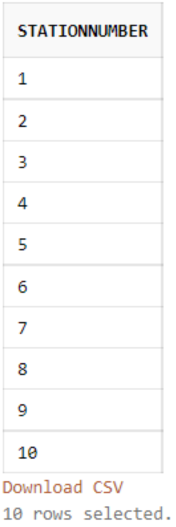


STATIONNUMBER
1
2
3
4
5
6
7
8
9
10

Download CSV

10 rows selected.

TERMINAL STATION TABLE

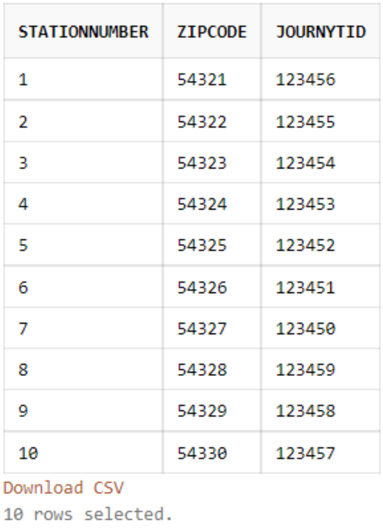


STATIONNUMBER
1
2
3
4
5
6
7
8
9
10

Download CSV

10 rows selected.

STATION TABLE

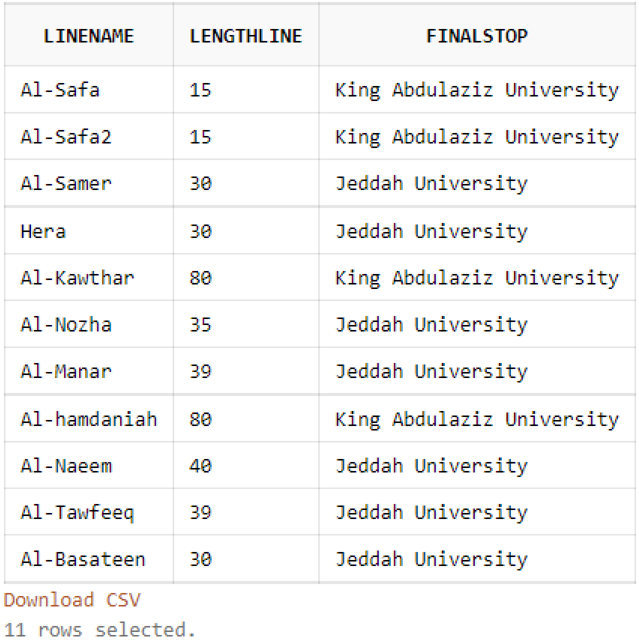


STATIONNUMBER	ZIPCODE	JOURNYTID
1	54321	123456
2	54322	123455
3	54323	123454
4	54324	123453
5	54325	123452
6	54326	123451
7	54327	123450
8	54328	123459
9	54329	123458
10	54330	123457

Download CSV

10 rows selected.

LINE TABLE

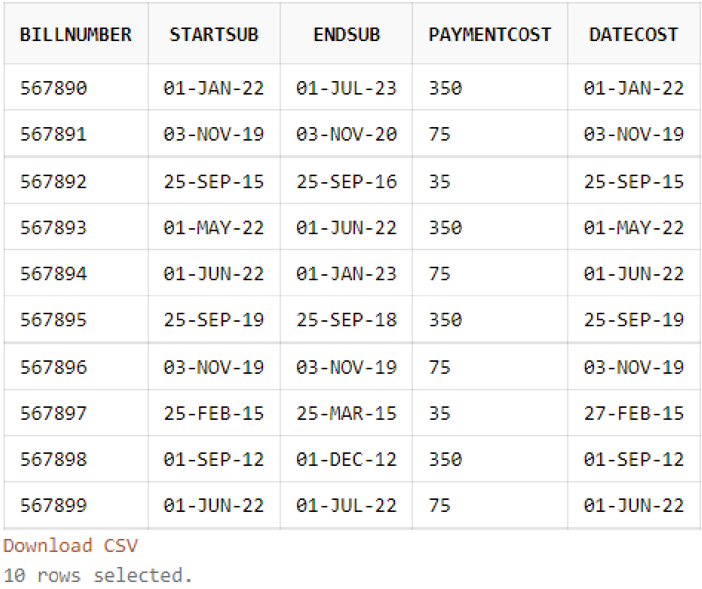


LINENAME	LENGTHLINE	FINALSTOP
Al-Safa	15	King Abdulaziz University
Al-Safa2	15	King Abdulaziz University
Al-Samer	30	Jeddah University
Hera	30	Jeddah University
Al-Kawthar	80	King Abdulaziz University
Al-Nozha	35	Jeddah University
Al-Manar	39	Jeddah University
Al-hamdaniah	80	King Abdulaziz University
Al-Naeem	40	Jeddah University
Al-Tawfeeq	39	Jeddah University
Al-Basateen	30	Jeddah University

Download CSV

11 rows selected.

SUBSCRIPTION TABLE

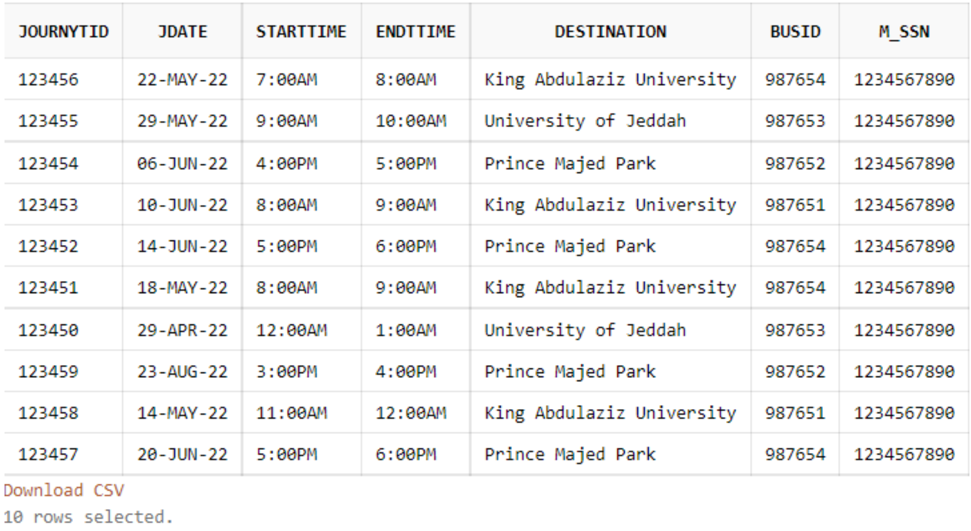


BILLNUMBER	STARTSUB	ENDSUB	PAYMENTCOST	DATECOST
567890	01-JAN-22	01-JUL-23	350	01-JAN-22
567891	03-NOV-19	03-NOV-20	75	03-NOV-19
567892	25-SEP-15	25-SEP-16	35	25-SEP-15
567893	01-MAY-22	01-JUN-22	350	01-MAY-22
567894	01-JUN-22	01-JAN-23	75	01-JUN-22
567895	25-SEP-19	25-SEP-18	350	25-SEP-19
567896	03-NOV-19	03-NOV-19	75	03-NOV-19
567897	25-FEB-15	25-MAR-15	35	27-FEB-15
567898	01-SEP-12	01-DEC-12	350	01-SEP-12
567899	01-JUN-22	01-JUL-22	75	01-JUN-22

Download CSV

10 rows selected.

TIMETABLE TABLE

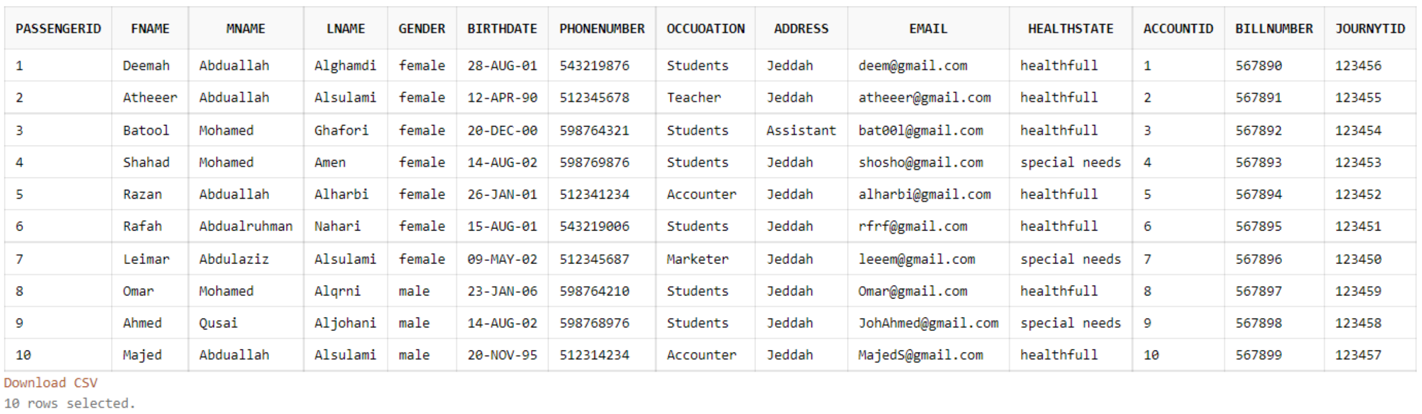


JOURNYTID	JDATE	STARTTIME	ENDTTIME	DESTINATION	BUSID	M_SSN
123456	22-MAY-22	7:00AM	8:00AM	King Abdulaziz University	987654	1234567890
123455	29-MAY-22	9:00AM	10:00AM	University of Jeddah	987653	1234567890
123454	06-JUN-22	4:00PM	5:00PM	Prince Majed Park	987652	1234567890
123453	10-JUN-22	8:00AM	9:00AM	King Abdulaziz University	987651	1234567890
123452	14-JUN-22	5:00PM	6:00PM	Prince Majed Park	987654	1234567890
123451	18-MAY-22	8:00AM	9:00AM	King Abdulaziz University	987654	1234567890
123450	29-APR-22	12:00AM	1:00AM	University of Jeddah	987653	1234567890
123459	23-AUG-22	3:00PM	4:00PM	Prince Majed Park	987652	1234567890
123458	14-MAY-22	11:00AM	12:00AM	King Abdulaziz University	987651	1234567890
123457	20-JUN-22	5:00PM	6:00PM	Prince Majed Park	987654	1234567890

Download CSV

10 rows selected.

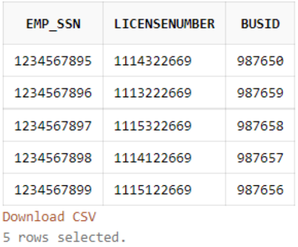
PASSENGER TABLE



PASSENGERID	FNAME	MNAME	LNAME	GENDER	BIRTHDATE	PHONENUMBER	OCCUOATION	ADDRESS	EMAIL	HEALTHSTATE	ACCOUNTID	BILLNUMBER	JOURNYTID
1	Deemah	Abduallah	Alghamdi	female	28-AUG-01	543219876	Students	Jeddah	deem@gmail.com	healthfull	1	567890	123456
2	Atheeer	Abduallah	Alsulami	female	12-APR-90	512345678	Teacher	Jeddah	atheeer@gmail.com	healthfull	2	567891	123455
3	Batool	Mohamed	Ghafori	female	20-DEC-00	598764321	Students	Assistant	bat00l@gmail.com	healthfull	3	567892	123454
4	Shahad	Mohamed	Amen	female	14-AUG-02	598769876	Students	Jeddah	shosho@gmail.com	special needs	4	567893	123453
5	Razan	Abduallah	Alharbi	female	26-JAN-01	512341234	Accounter	Jeddah	alharbi@gmail.com	healthfull	5	567894	123452
6	Rafah	Abdualruhman	Nahari	female	15-AUG-01	543219006	Students	Jeddah	rfrf@gmail.com	healthfull	6	567895	123451
7	Leimar	Abdulaziz	Alsulami	female	09-MAY-02	512345687	Marketer	Jeddah	leeem@gmail.com	special needs	7	567896	123450
8	Omar	Mohamed	Alqrni	male	23-JAN-06	598764210	Students	Jeddah	Omar@gmail.com	healthfull	8	567897	123459
9	Ahmed	Qusai	Aljohani	male	14-AUG-02	598768976	Students	Jeddah	JohAhmed@gmail.com	special needs	9	567898	123458
10	Majed	Abduallah	Alsulami	male	20-NOV-95	512314234	Accounter	Jeddah	MajedS@gmail.com	healthfull	10	567899	123457

Download CSV

10 rows selected.

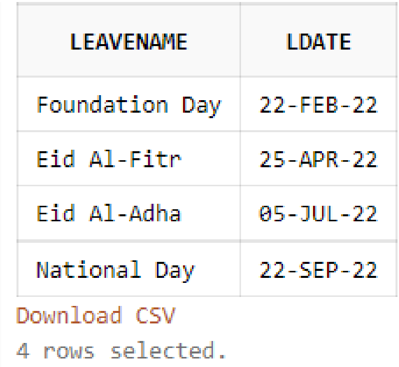


EMP_SSN	LICENSENUMBER	BUSID
1234567895	1114322669	987650
1234567896	1113222669	987659
1234567897	1115322669	987658
1234567898	1114122669	987657
1234567899	1115122669	987656

Download CSV

5 rows selected.

DRIVER TABLE

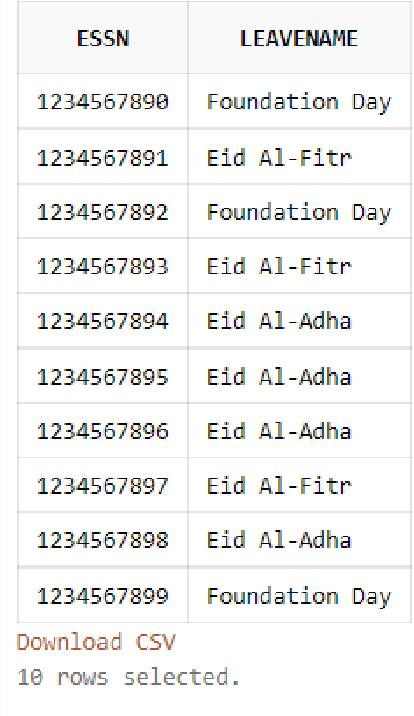


LEAVENAME	LDATE
Foundation Day	22-FEB-22
Eid Al-Fitr	25-APR-22
Eid Al-Adha	05-JUL-22
National Day	22-SEP-22

Download CSV

4 rows selected.

LEAVE TABLE



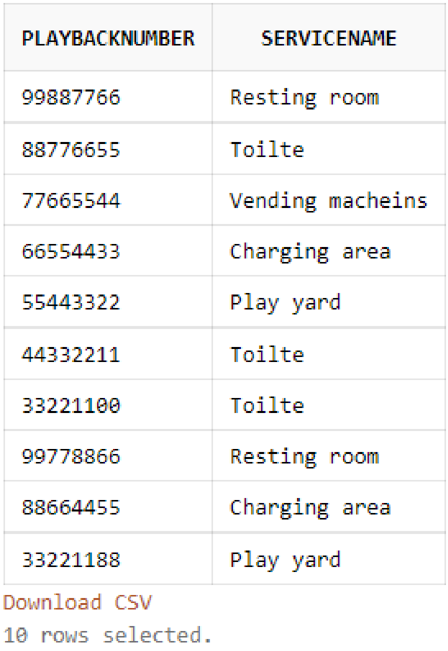
ESSN	LEAVENAME
1234567890	Foundation Day
1234567891	Eid Al-Fitr
1234567892	Foundation Day
1234567893	Eid Al-Fitr
1234567894	Eid Al-Adha
1234567895	Eid Al-Adha
1234567896	Eid Al-Adha
1234567897	Eid Al-Fitr
1234567898	Eid Al-Adha
1234567899	Foundation Day

Download CSV

10 rows selected.

LEAVE AVAILABLE TABLE

SERVICE TABLE

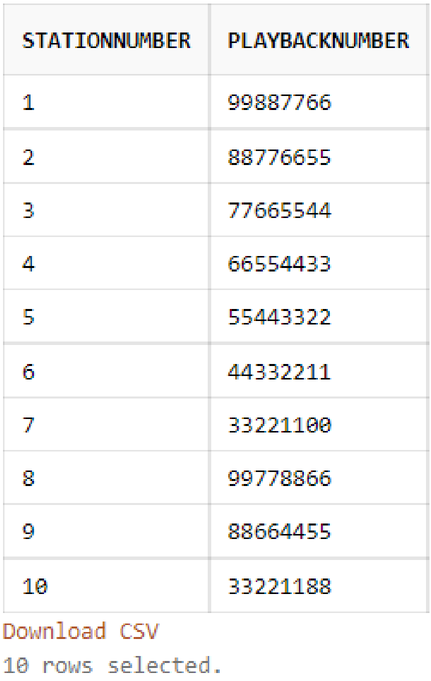


PLAYBACKNUMBER	SERVICENAME
99887766	Resting room
88776655	Toilte
77665544	Vending macheins
66554433	Charging area
55443322	Play yard
44332211	Toilte
33221100	Toilte
99778866	Resting room
88664455	Charging area
33221188	Play yard

Download CSV

10 rows selected.

PROVIDES TABLE

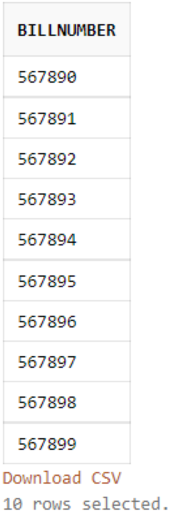


STATIONNUMBER	PLAYBACKNUMBER
1	99887766
2	88776655
3	77665544
4	66554433
5	55443322
6	44332211
7	33221100
8	99778866
9	88664455
10	33221188

Download CSV

10 rows selected.

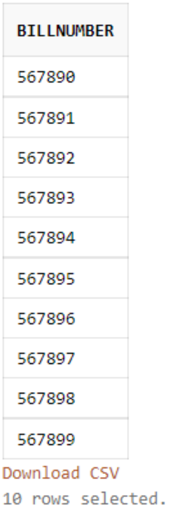
THESE TABLES FOR, MADA, APPLEPAY, CASH, WEEKLY, MONTHLY, AND ANNUAL .



BILLNUMBER
567890
567891
567892
567893
567894
567895
567896
567897
567898
567899

Download CSV

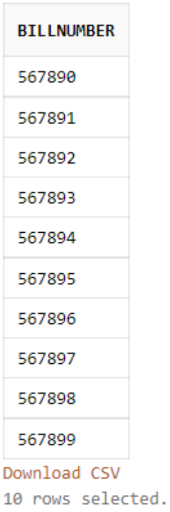
10 rows selected.



BILLNUMBER
567890
567891
567892
567893
567894
567895
567896
567897
567898
567899

Download CSV

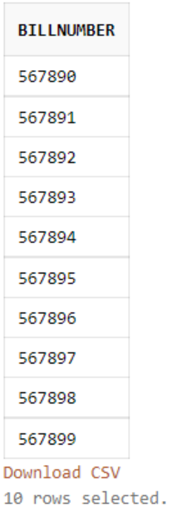
10 rows selected.



BILLNUMBER
567890
567891
567892
567893
567894
567895
567896
567897
567898
567899

Download CSV

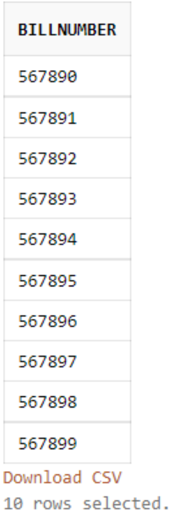
10 rows selected.



BILLNUMBER
567890
567891
567892
567893
567894
567895
567896
567897
567898
567899

Download CSV

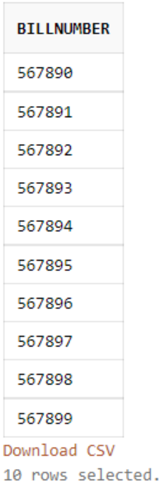
10 rows selected.



BILLNUMBER
567890
567891
567892
567893
567894
567895
567896
567897
567898
567899

Download CSV

10 rows selected.



BILLNUMBER
567890
567891
567892
567893
567894
567895
567896
567897
567898
567899

Download CSV

10 rows selected.