Team Members:
Haziel Andrade-Ayala
Tarik Klancevic
Angel Hernandez

Project Name: Rent-A-Car Database System

Section 1: Introduction of the Database

Rent-A-Car is a hire agency company that rents automobiles for short periods of time ranging from a few hours to purchasing. For our project, we are going to develop a Database for Rent-A-Car to facilitate in helping customers easily purchase & rent cars online and have the car picked up from our warehouse or delivered from the warehouse to an assigned location. This will make it easier to be able to rent any car available that week. It will also include features that enable the user to choose the make, model, and model year. Many things were working out for Rent-A-Car, however due to the pandemic, places had begun to shut everything down and Rent-A-Car now needs a way to take their current database online to create a user-friendly experience through a new database system.

Current data management problems:

- Facilitating bookings
- Schedule locations
- Manage features

- Manage dates
- Keep track of cars in the database
- Keep track of users

Motivations for your DB development: Improve user engagement to create a convenient and fast customer experience, improve efficiencies and maintain good relations with customers.

Potential benefits and users of your DB:

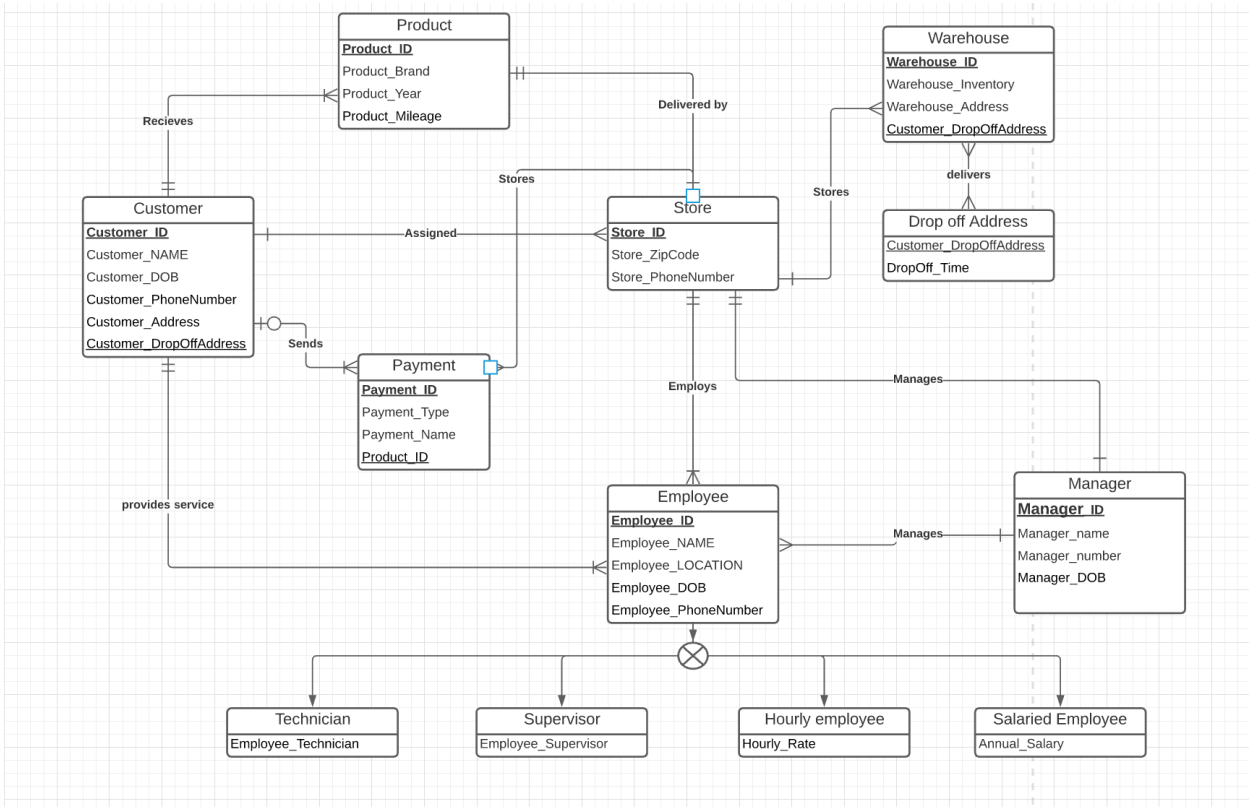
- Easy access to purchase and rent cars.
- Convenient delivery to specific locations and to warehouses.
- Customizable experience that allows you to choose the make, model, and model year.

Section 2: Business Rules and User Requirements:

1. User data must be updated as soon as new information is processed.
2. Rent-A-Car employees will have full access to the database.
3. *Customers* will be able to place orders and receive the *Product*.
4. All *Products* must be able to be delivered to the *Store*.
5. The *Store* will be able to keep all *Products* in the *Warehouse*.
6. *Employees* will be able to provide services to *Customers*.
7. *Customers* will be required to make a *Payment* before receiving the *Product*.
8. Each *Store* will save *Payment* information.
9. Each *Store* will have a mandatory *Manager*.

10. *Employees* will have a subtype *Technician*.
11. *Employees* will have a subtype *Salary Employee*.
12. *Employees* will have a subtype *Hourly Employee*.
13. *Employees* will have a subtype *Supervisor*.
14. *Warehouse* will have a subtype *Supervisor*.
15. *Employees* will have a mandatory *Manager*.
16. *Dropoff* Address will deliver to all *Warehouses*.

Section 3: EERD Diagram



Caption: 12 different entities with drop off address being a weak entity

Section 4: Relations are Normalized

1ST Normal Form:

Customer		
FName	**LName**	**PhoneNumber**
John	Lincon	(902) 727-3652
Tim	Hart	(783) 601-3029
Rose	Anderson	(595) 461-7862

Product	
Product ID	**Product Color**
#2390	Blue
#8910	Green
#1289	Yellow

Payment		
Product ID	**Price**	**Name on Card**
#2390	$15.00	John Lincon
#8910	$32.00	Tim Hart
#1289	$157.00	Rose Anderson

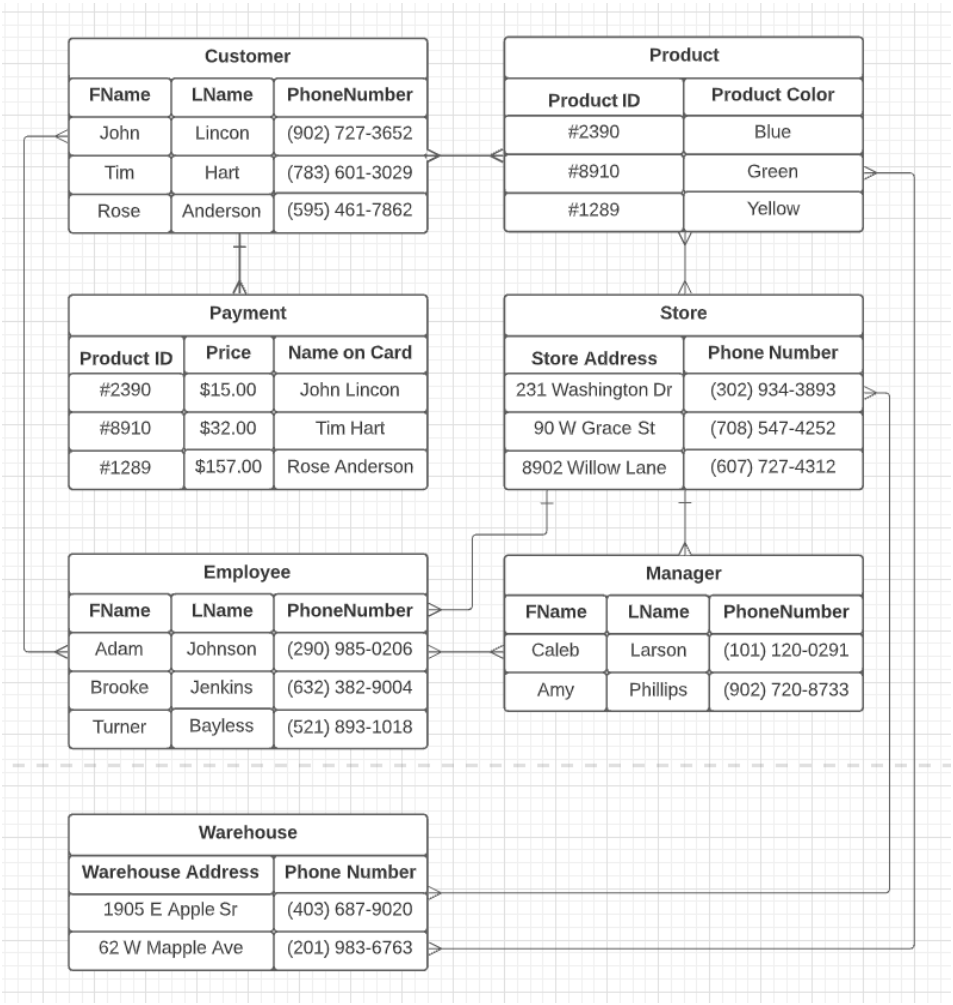
Store	
Store Address	**Phone Number**
231 Washington Dr	(302) 934-3893
90 W Grace St	(708) 547-4252
8902 Willow Lane	(607) 727-4312

Employee		
FName	**LName**	**PhoneNumber**
Adam	Johnson	(290) 985-0206
Brooke	Jenkins	(632) 382-9004
Turner	Bayless	(521) 893-1018

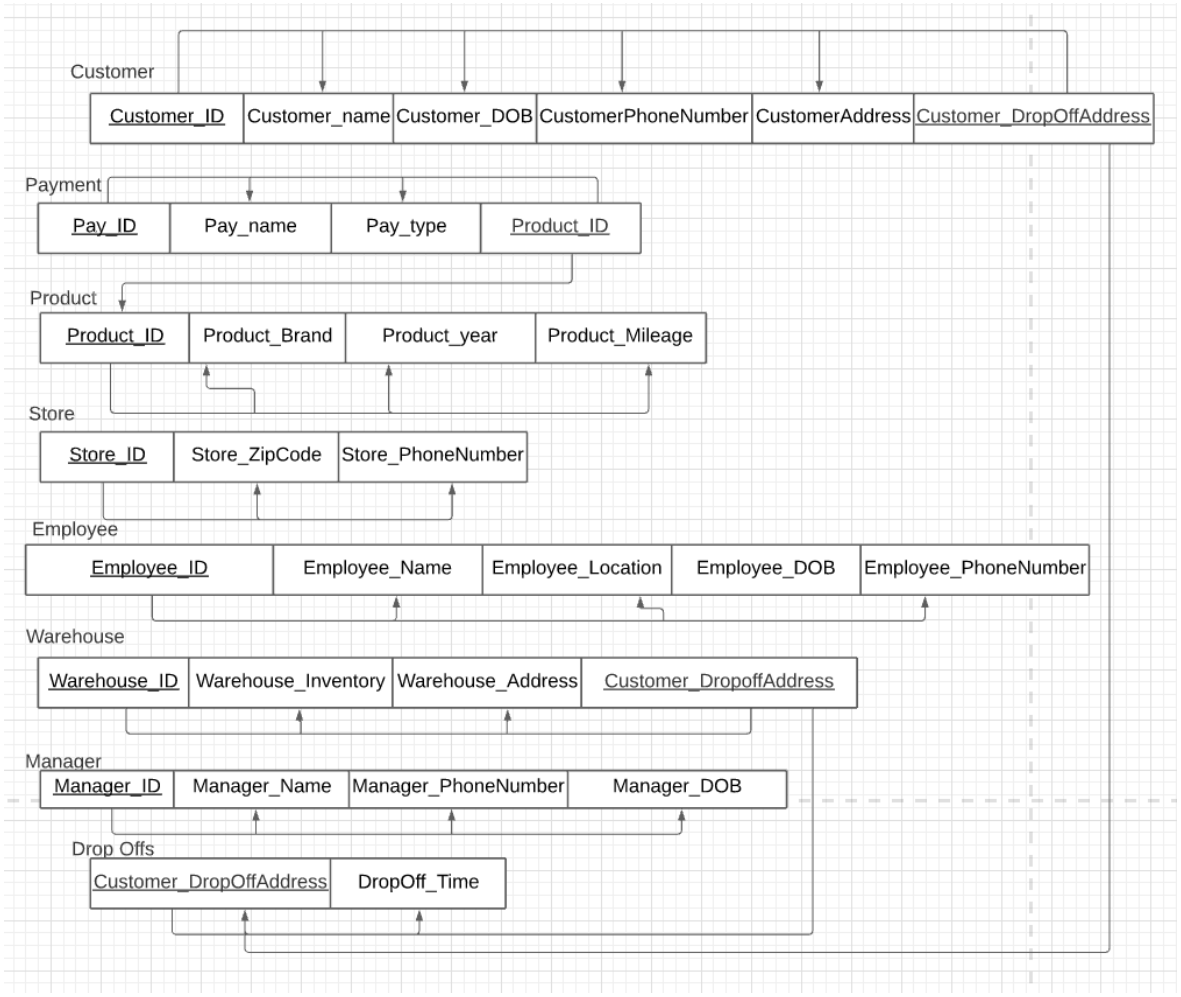
Manager		
FName	**LName**	**PhoneNumber**
Caleb	Larson	(101) 120-0291
Amy	Phillips	(902) 720-8733

Warehouse	
Warehouse Address	**Phone Number**
1905 E Apple Sr	(403) 687-9020
62 W Mapple Ave	(201) 983-6763

2ND Normal Form:


Customer
FName
LName
PhoneNumber
John
Lincon
(902) 727-3652
Tim
Hart
(783) 601-3029
Rose
Anderson
(595) 461-7862
Product
Product ID
Product Color
#2390
Blue
#8910
Green
#1289
Yellow
Payment
Product ID
Price
Name on Card
#2390
$15.00
John Lincon
#8910
$32.00
Tim Hart
#1289
$157.00
Rose Anderson
Store
Store Address
Phone Number
231 Washington Dr
(302) 934-3893
90 W Grace St
(708) 547-4252
8902 Willow Lane
(607) 727-4312
Employee
FName
LName
PhoneNumber
Adam
Johnson
(290) 985-0206
Brooke
Jenkins
(632) 382-9004
Turner
Bayless
(521) 893-1018
Manager
FName
LName
PhoneNumber
Caleb
Larson
(101) 120-0291
Amy
Phillips
(902) 720-8733
Warehouse
Warehouse Address
Phone Number
1905 E Apple Sr
(403) 687-9020
62 W Mapple Ave
(201) 983-6763

3RD Normal Form:

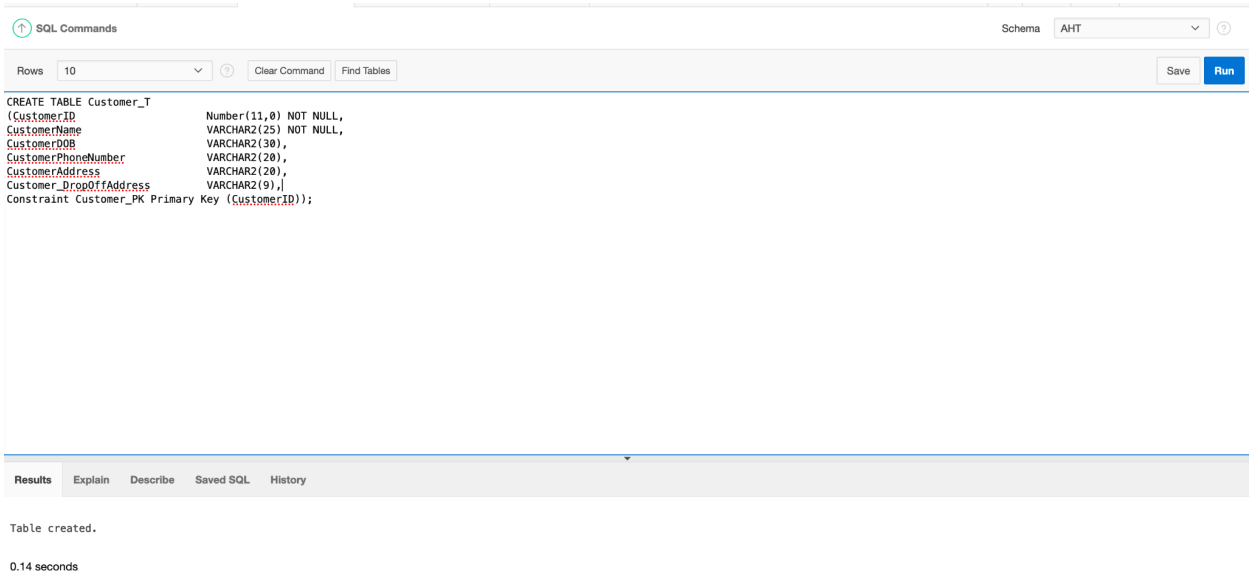

Customer
Customer_ID
Customer_name
Customer_DOB
CustomerPhoneNumber
CustomerAddress
Customer_DropOffAddress
Payment
Pay_ID
Pay_name
Pay_type
Product_ID
Product
Product_ID
Product_Brand
Product_year
Product_Mileage
Store
Store_ID
Store_ZipCode
Store_PhoneNumber
Employee
Employee_ID
Employee_Name
Employee_Location
Employee_DOB
Employee_PhoneNumber
Warehouse
Warehouse_ID
Warehouse_Inventory
Warehouse_Address
Customer_DropoffAddress
Manager
Manager_ID
Manager_Name
Manager_PhoneNumber
Manager_DOB
Drop Offs
Customer_DropOffAddress
DropOff_Time

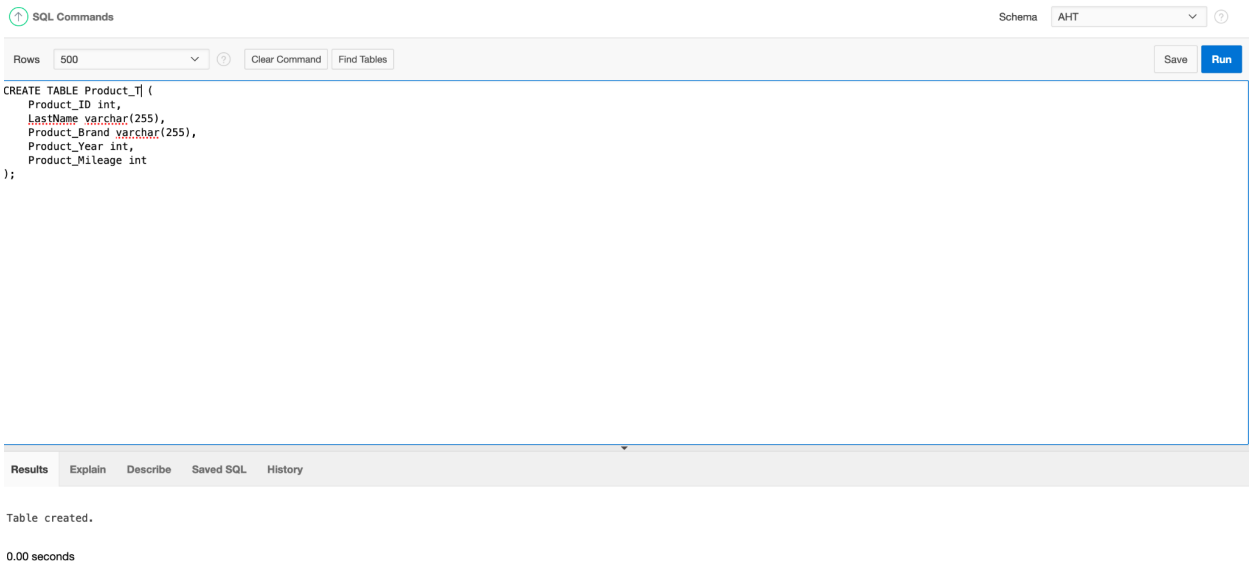
Section 5: SQL CREATE statement (Must create ALL Required Tables as many as the normalized Relations).

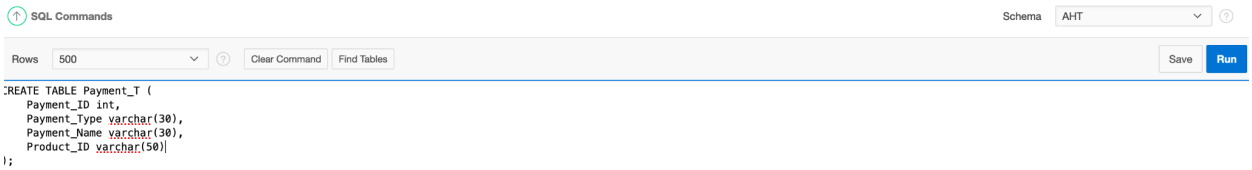
```
CREATE TABLE Customer_T
(CustomerID                 Number(11,0) NOT NULL
CustomerName                VARCHAR2(25) NOT NULL
CustomerDOB                 VARCHAR2(30)
CustomerPhoneNumber         VARCHAR2(20)
CustomerAddress             VARCHAR2(20)
Customer_DropOffAddress     VARCHAR2(9)
Constraint Customer_PK Primary Key (CustomerID));

CREATE TABLE Product (
   Product_ID int,
   LastName varchar(50),
   Product_Brand varchar(50),
   Product_Year int,
   Product_Mileage int
);

CREATE TABLE Payment_T (
    Payment_ID int,
    Payment_Type varchar(30),
```

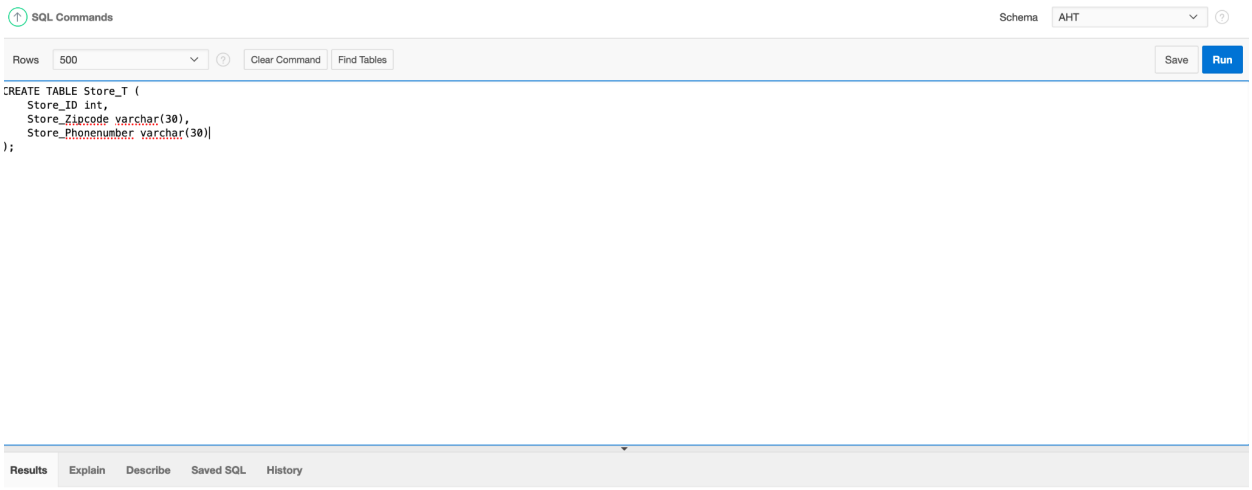
```
    Payment_Name varchar(30),
    Product_ID varchar(50)
);

CREATE TABLE Store_T (
   Store_ID int,
   Store_Zipcode varchar(30),
   Store_Phonenumber varchar(30)
);

CREATE TABLE Warehouse_T (
   Warehouse_ID int,
   Warehouse_Inventory varchar(30),
   Warehouse_Address varchar(30),
   CustomerDropOff_Address varchar(30)
);

CREATE TABLE Employee_T (
   Employee_ID int,
   Employee_Name varchar(30),
   Employee_Location varchar(30),
   Employee_DOB varchar(30),
   Employee_PhoneNumber varchar(30)
```

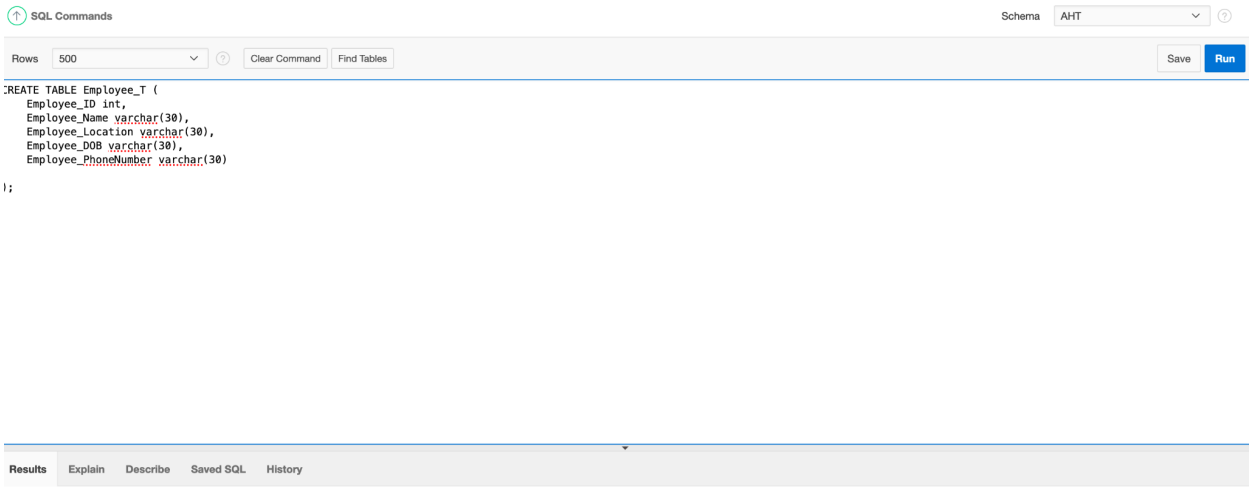
```
);

CREATE TABLE Manager_T (
    Manager_ID int,
    Manager_Name varchar(30),
    Manager_Number varchar(30),
    Manage_DOB varchar(30)

);
```

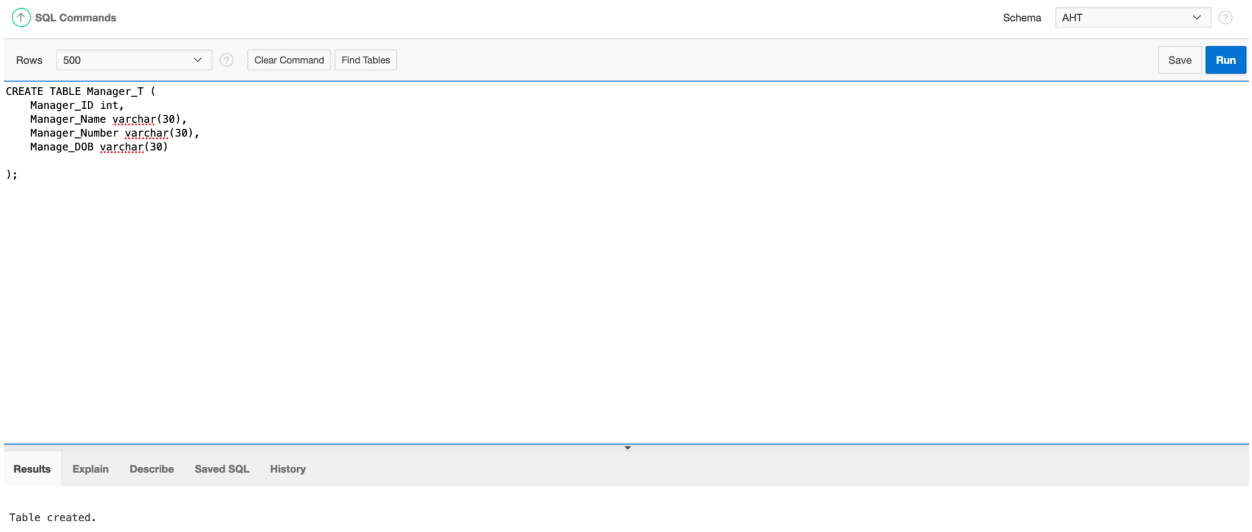

SQL Commands
Schema
AHT
Rows
10
Clear Command
Find Tables
Save
Run
CREATE TABLE Customer_T
(CustomerID Number(11,0) NOT NULL,
CustomerName VARCHAR2(25) NOT NULL,
CustomerDOB VARCHAR2(30),
CustomerPhoneNumber VARCHAR2(20),
CustomerAddress VARCHAR2(20),
Customer_DropOffAddress VARCHAR2(9),
Constraint Customer_PK Primary Key (CustomerID));
Results
Explain
Describe
Saved SQL
History
Table created.
0.14 seconds


SQL Commands
Schema
AHT
Rows
500
Clear Command
Find Tables
Save
Run
CREATE TABLE Product_T (
Product_ID int,
LastName varchar(255),
Product_Brand varchar(255),
Product_Year int,
Product_Mileage int
);
Results
Explain
Describe
Saved SQL
History
Table created.
0.00 seconds


SQL Commands
Schema
AHT
Rows
500
Clear Command
Find Tables
Save
Run
CREATE TABLE Payment_T (
Payment_ID int,
Payment_Type varchar(30),
Payment_Name varchar(30),
Product_ID varchar(50)
);


SQL Commands
Schema
AHT
Rows
500
Clear Command
Find Tables
Save
Run
CREATE TABLE Store_T (
Store_ID int,
Store_Zipcode varchar(30),
Store_Phonenumber varchar(30)
);
Results
Explain
Describe
Saved SQL
History
Table created.


SQL Commands
Schema
AHT
Rows
500
Clear Command
Find Tables
Save
Run
CREATE TABLE Employee_T (
Employee_ID int,
Employee_Name varchar(30),
Employee_Location varchar(30),
Employee_DOB varchar(30),
Employee_PhoneNumber varchar(30)
);
Results
Explain
Describe
Saved SQL
History
Table created.




```
INSERT INTO Customer_T (CustomerID)
VALUES (58943030, 549202134, 329840328, 459204569, 34854029,
38560234, 34953256,38990122 );

SELECT Warehouse_ID, CUSTOMER_DROPOFFADDRESS
FROM Warehouse_T
```

```
INNER JOIN Customer_T
ON Warehouse_T.Warehouse_ID = Customer_T.CUSTOMER_DROPOFFADDRESS;
```

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

```
SELECT Customer_Address.Customer_DropOffAddress,
Customer_T.Customer_Address
FROM Warehouse_T
INNER JOIN Customer_T
ON Warehouse_T.Customer_DropOffAddress =
Customer_T.Customer_Address;
```

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

```sql
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

```sql
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

1)

1. INSERT INTO statement (Must enter 5-8 records for each of the Tables created).

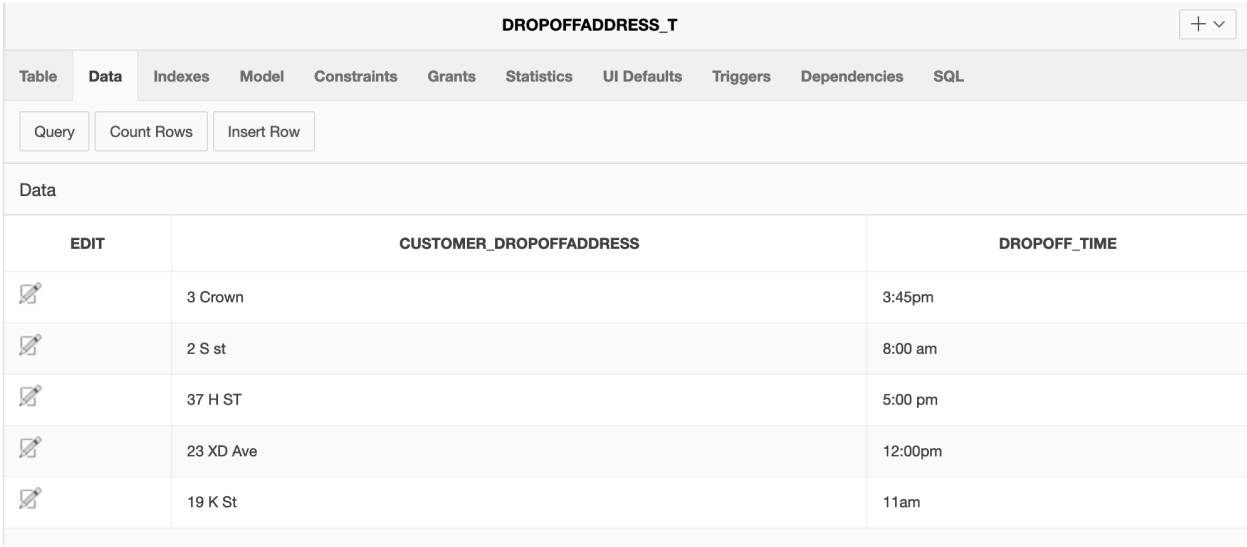
CUSTOMER_T

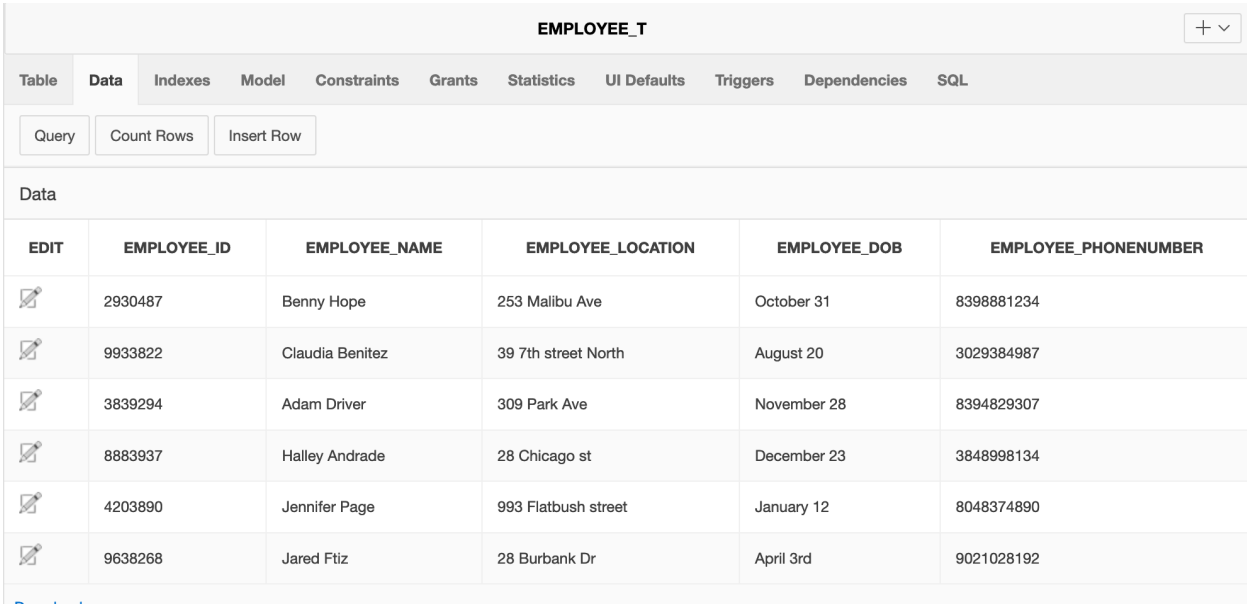
Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

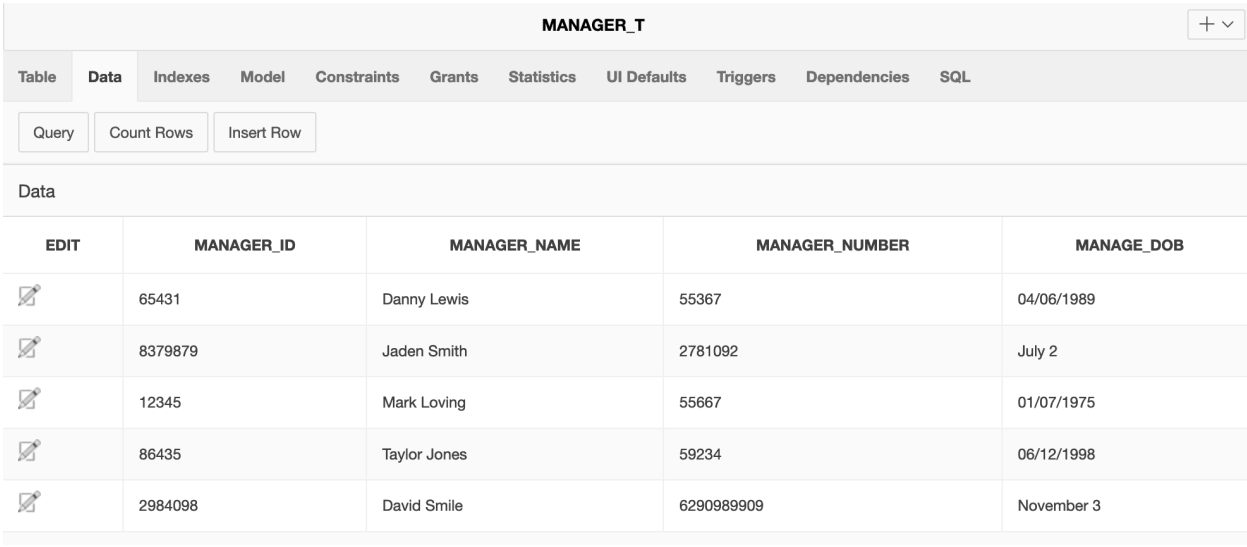
Query | Count Rows | Insert Row

Data

EDIT	CUSTOMERID	CUSTOMERNAME	CUSTOMERDOB	CUSTOMERPHONENUMBER	CUSTOMERADDRESS	CUSTOMER_DROPOFFADDRESS
	102030	Tom Erichsen	March 21	5402903430	8212 Floyd Ave	23 XD Ave
	2304932	Edward Jones	May 22	5402934590	563 Marshall Ave	3 Crown
	3908773	John Doe	June 3	2873809988	12 California Street	19 K St
	209873	Micheal Angelo	December 16	2890987789	23 Hellen street	37 H ST
	3833333	Saki Moon	September 21	3768990089	505 Drive st	2 S st


DROPOFFADDRESS_T
Table
Data
Indexes
Model
Constraints
Grants
Statistics
UI Defaults
Triggers
Dependencies
SQL
Query
Count Rows
Insert Row
Data
EDIT
CUSTOMER_DROPOFFADDRESS
DROPOFF_TIME
3 Crown
3:45pm
2 S st
8:00 am
37 H ST
5:00 pm
23 XD Ave
12:00pm
19 K St
11am


EMPLOYEE_T
Table
Data
Indexes
Model
Constraints
Grants
Statistics
UI Defaults
Triggers
Dependencies
SQL
Query
Count Rows
Insert Row
Data
EDIT
EMPLOYEE_ID
EMPLOYEE_NAME
EMPLOYEE_LOCATION
EMPLOYEE_DOB
EMPLOYEE_PHONENUMBER
2930487
Benny Hope
253 Malibu Ave
October 31
8398881234
9933822
Claudia Benitez
39 7th street North
August 20
3029384987
3839294
Adam Driver
309 Park Ave
November 28
8394829307
8883937
Halley Andrade
28 Chicago st
December 23
3848998134
4203890
Jennifer Page
993 Flatbush street
January 12
8048374890
9638268
Jared Ftiz
28 Burbank Dr
April 3rd
9021028192



EDIT	MANAGER_ID	MANAGER_NAME	MANAGER_NUMBER	MANAGE_DOB
	65431	Danny Lewis	55367	04/06/1989
	8379879	Jaden Smith	2781092	July 2
	12345	Mark Loving	55667	01/07/1975
	86435	Taylor Jones	59234	06/12/1998
	2984098	David Smile	6290989909	November 3

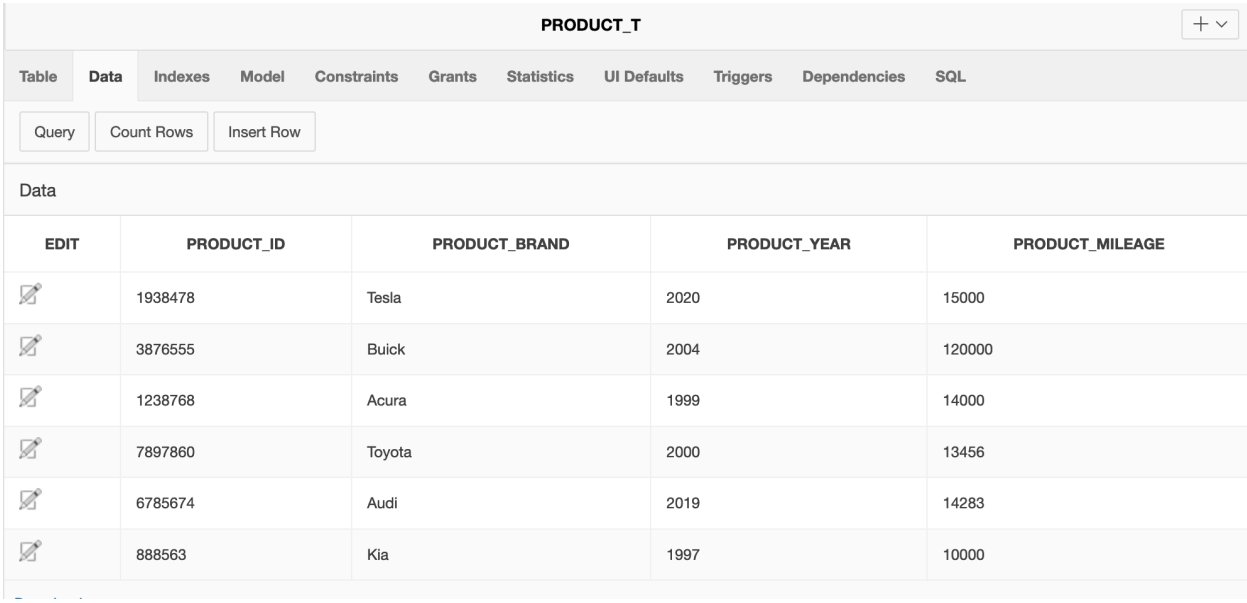
PAYMENT_T

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

Data

EDIT	PAYMENT_ID	PAYMENT_TYPE	PAYMENT_NAME	PRODUCT_ID
	2034921	VISA	CARD	34094321
	2034365	MASTERCARD	CARD	39894321
	1034365	AMEX	CARD	81894321
	8334365	VISA	CARD	39894421
	3514365	CASH	CASH	39928341


PRODUCT_T
Table
Data
Indexes
Model
Constraints
Grants
Statistics
UI Defaults
Triggers
Dependencies
SQL
Query
Count Rows
Insert Row
Data
EDIT
PRODUCT_ID
PRODUCT_BRAND
PRODUCT_YEAR
PRODUCT_MILEAGE
1938478
Tesla
2020
15000
3876555
Buick
2004
120000
1238768
Acura
1999
14000
7897860
Toyota
2000
13456
6785674
Audi
2019
14283
888563
Kia
1997
10000

STORE_T

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

Data

EDIT	STORE_ID	STORE_ZIPCODE	STORE_PHONENUMBER
	2039849	28498	2098394758
	2349874	23409	8908475849
	9987869	23450	2890756946
	5983498	56745	3556787654
	5876874	78646	4587620900

WAREHOUSE_T

Table | Data | Indexes | Model | Constraints | Grants | Statistics | UI Defaults | Triggers | Dependencies | SQL

Query | Count Rows | Insert Row

Data

EDIT	WAREHOUSE_ID	WAREHOUSE_INVENTORY	WAREHOUSE_ADDRESS	CUSTOMERDROPOFF_ADDRESS
	7203	76	78 Ken street	23 XD Ave
	1256	150	800 W Broad	3 Crown
	1647	550	657 Brooke Rd	19 K St
	4453	525	477 Parham Rd	37 H ST
	4521	400	410 Pillow St	2 S st

EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_LOCATION	EMPLOYEE_DOB	EMPLOYEE_PHONENUMBER
2930487	Benny Hope	253 Malibu Ave	October 31	8398881234
9933822	Claudia Benitez	39 7th street North	August 20	3029384987
3839294	Adam Driver	309 Park Ave	November 28	8394829307
8883937	Halley Andrade	28 Chicago st	December 23	3848998134
4203890	Jennifer Page	993 Flatbush street	January 12	8048374890
9638268	Jared Ftiz	28 Burbank Dr	April 3rd	9021028192

MANAGER_ID	MANAGER_NAME	MANAGER_NUMBER	MANAGE_DOB
65431	Danny Lewis	55367	04/06/1989
8379879	Jaden Smith	2781092	July 2
12345	Mark Loving	55667	01/07/1975
86435	Taylor Jones	59234	06/12/1998
2984098	David Smile	6290989909	November 3

Owner	Table Name	Index Name	Used in Plan	Columns	Uniqueness	Status	Index Type	Join Index
AHT	CUSTOMER_T	CUSTOMER_PK		CUSTOMERID	UNIQUE	VALID	NORMAL	NO

Query Plan

Operation	Options	Object	Rows	Time	Cost	Bytes	Filter Predicates *	Access Predicates
SELECT STATEMENT			5	1	6	65		
HASH JOIN			5	1	6	65		"WAREHOUSE_T"."WAREHOUSE_ID" = TO_NUMBER("CUSTOMER_T"."CUSTOMER_DROPOFFADDRESS")
TABLE ACCESS	FULL	WAREHOUSE_T	5	1	3	20		
TABLE ACCESS	FULL	CUSTOMER_T	5	1	3	45		

* Unindexed columns are shown in red

Peer Evaluation:

Angel: During the creation of our project, I realized the importance of working as a team. At one point, we were stuck as a group but we were able to walk through each situation. We worked as a team when creating the relational diagrams and delegated the responsibilities of creating the codes for SQL. We constantly communicated with each other and helped each other out.

Haziel: In my experience, I think our project involved deeper analysis into the core subjects of what SQL really does and how it can be extremely useful when applying it to real world scenarios. I worked on part of the SQL along with Angel and I would say that the hardest part in my opinion was utilizing joins like inner because it became a bit technical at times to analyze relevant relationships. What I found most rewarding was applying concepts to practice with a group because I felt as though there was something new that we had created.

Tarik: In my experience with the project I realized that working with a group of 3 really helped with answering questions. I worked with Haziel and Angel on the EERD diagram and found out that it was helpful for all of us to be there together and use each other's ideas. I also liked that as a team we did well to communicate on discord and update each other on our availability. I found working with Angel and Haziel was really smooth and would work with them again.

New Skill I learned:

Tarik: Before this semester I never worked with SQL or other Database programming. I found that the book and Powerpoints helped me better understand how to read the Database and figure out what goes where.

Angel: Prior to the start of this class, I had little to no experience with SQL. As the semester progressed, I developed a key understanding of SQL along with database programming.

Haziel: A new skill that I learned was practicing how to navigate in SQL more effectively then from when I had first started at the start of the semester.