**Group 18**

Phase 3

**Group Members**

**Alana Kaplan**

Section: B

Email Address: [akaplan8@gatech.edu](mailto:akaplan8@gatech.edu)

T-Square Username: akaplan8

**Ann Tribone**

Section: B

Email Address: [atribone@gatech.edu](mailto:atribone@gatech.edu)

T-Square Username: atribone3

**Erin McNamara**

Section: B

Email Address: [emcnamara3@gatech.edu](mailto:emcnamara3@gatech.edu)

T-Square Username: emcnamara3

**Henry Smith**

Section: B

Email Address: [henry.smith@gatech.edu](mailto:henry.smith@gatech.edu)

T-Square Username: hsmith41

**Create Table Statements**

CREATE TABLE IF NOT EXISTS `driving\_plan` (

`type` char(20) NOT NULL default '',

`discount` int(10) NOT NULL,

`monthly\_payment` int(10) NOT NULL,

`annual\_fee` int(10) NOT NULL,

PRIMARY KEY (`type`) );

CREATE TABLE IF NOT EXISTS `payment\_information` (

`username` char(20) NOT NULL,

`billing\_address` char(100) default NULL,

`expiration\_date` datetime default NULL,

`name\_on\_card` varchar(50) default NULL,

`ccv` int(4) default NULL,

`card\_number` bigint(16) default NULL,

KEY `username` (`username`) );

CREATE TABLE IF NOT EXISTS `users` (

`password` char(20) NOT NULL,

`username` char(20) NOT NULL default '',

`usertype` char(20) default NULL,

PRIMARY KEY (`username`) );

CREATE TABLE IF NOT EXISTS `gt\_student` (

`username` char(20) NOT NULL,

`email` char(50) default NULL,

`address` char(70) default NULL,

`plan\_type` char(20) default NULL,

`first\_name` char(20) default NULL,

`middle\_initial` char(1) default NULL,

`last\_name` char(20) default NULL,

`phone` int(10) default NULL,

UNIQUE KEY `username` (`username`),

KEY `plan\_type` (`plan\_type`) );

CREATE TABLE IF NOT EXISTS `car\_information` (

`vehicle\_sno` bigint(16) NOT NULL,

`auxilary\_cable` char(1) default NULL,

`transmission\_type` char(9) default NULL,

`seating\_capacity` int(2) default NULL,

`blue\_tooth\_connectivity` char(1) default NULL,

`daily\_rate` int(5) NOT NULL,

`hourly\_rate` int(5) NOT NULL,

`color` char(7) default NULL,

`type` char(12) default NULL,

`model\_name` char(10) NOT NULL,

`under\_maitenence\_flag` char(1) default NULL,

`location` char(12) NOT NULL,

PRIMARY KEY (`vehicle\_sno`),

KEY `location` (`location`) );

CREATE TABLE IF NOT EXISTS `location` (

`location\_name` char(20) NOT NULL default '',

`capacity` int(10) default NULL,

PRIMARY KEY (`location\_name`) );

CREATE TABLE IF NOT EXISTS `maintenance\_request` (

`request\_date` datetime NOT NULL default '0000-00-00 00:00:00',

`vehicle\_sno` bigint(16) NOT NULL default '0',

`description` char(200) default NULL,

`employee\_username` char(20) default NULL,

PRIMARY KEY (`vehicle\_sno`,`request\_date`),

KEY `employee\_username` (`employee\_username`) );

CREATE TABLE IF NOT EXISTS `reservation` (

`pickup\_date` datetime NOT NULL default '0000-00-00 00:00:00',

`return\_date` datetime NOT NULL default '0000-00-00 00:00:00',

`late\_by` datetime default NULL,

`estimated\_cost` int(7) default NULL,

`return\_status` char(1) default NULL,

`extended\_time` int(5) default NULL,

`late\_fees` int(7) default NULL,

`location\_name` char(20) default NULL,

`vehicle\_sno` bigint(16) default NULL,

`username` char(20) NOT NULL default '',

PRIMARY KEY (`pickup\_date`,`return\_date`,`username`),

KEY `vehicle\_sno` (`vehicle\_sno`),

KEY `username` (`username`),

KEY `location\_name` (`location\_name`) );

**SQL Statements**

1. **Availability**

SELECT plan\_type FROM gt\_student

WHERE gt\_student.username = username

SELECT \* FROM car\_information

LEFT JOIN reservation

ON car\_information.vehicle\_sno= reservation.vehicle\_sno

WHERE reservation.pickup\_date NOT BETWEEN pickupDateString AND returnDateString

AND reservation.return\_date NOT BETWEEN pickupDateString AND returnDateString

AND car\_information.type = carType

GROUP BY car\_information.location

ORDER BY CASE WHEN car\_information.location=location then 0 else 1 end

1. **Login**

SELECT usertype FROM users WHERE username= username AND password = password

1. **Manage Cars**

SELECT location\_name FROM location

INSERT INTO car\_information (vehicle\_sno, auxilary\_cable, transmission\_type, seating\_capacity, blue\_tooth\_connectivity, daily\_rate, hourly\_rate, color, type, model\_name, location) VALUES (sno, aux, transType, scap, blue, drate, hrate, color, cType, model, loc)

SELECT model\_name FROM car\_information WHERE location = loc

SELECT type FROM car\_information WHERE model\_name = model

SELECT color FROM car\_information WHERE model\_name = model

SELECT seating\_capacity FROM car\_information WHERE model\_name = model

SELECT transmission\_type FROM car\_information WHERE model\_name = model

UPDATE car\_information SET location = loc WHERE model\_name = model

1. **New Rental**

SELECT DISTINCT location\_name FROM location

SELECT DISTINCT type FROM car\_information

1. **Personal Info**

SELECT \*

FROM gt\_student, payment\_information

WHERE gt\_student.username = username

AND payment\_information.username = username

UPDATE gt\_student SET

gt\_student.first\_name=IFNULL(firstname1, gt\_student.first\_name),

gt\_student.middle\_initial=IFNULL(mi1, gt\_student.middle\_initial),

gt\_student.last\_name=IFNULL(lastname1, gt\_student.last\_name),

gt\_student.email=IFNULL(email1, gt\_student.email),

gt\_student.phone=IFNULL(phone1, gt\_student.phone),

gt\_student.plan\_type=IFNULL(planType, gt\_student.plan\_type)

WHERE gt\_student.username = username

UPDATE payment\_information SET

payment\_information.name\_on\_card=IFNULL(name1, payment\_information.name\_on\_card),

payment\_information.billing\_address=IFNULL(bill1, payment\_information.billing\_address),

payment\_information.expiration\_date=IFNULL(exdate1, payment\_information.expiration\_date),

payment\_information.card\_number=IFNULL(cc1, payment\_information.card\_number),

payment\_information.ccv=IFNULL(ccv1, payment\_information.ccv)

WHERE payment\_information.username = username

1. **Rental Change Request**

SELECT vehicle\_sno FROM reservation WHERE username = username

SELECT type FROM car\_information WHERE vehicle\_sno = sno

SELECT location\_name FROM reservation WHERE vehicle\_sno = sno

SELECT return\_date FROM reservation WHERE vehicle\_sno = sno

SELECT vehicle\_sno FROM reservation WHERE DATEDIFF(minute, self.date, return\_date) <=0

SELECT username FROM reservation WHERE vehicle\_sno = sno

SELECT pickup\_date FROM reservation WHERE vehicle\_sno = sno

SELECT return\_date FROM reservation WHERE vehicle\_sno = sno

SELECT email FROM gt\_student WHERE username = username

SELECT phone FROM gt\_student WHERE username = username

1. **Rental Info**

SELECT car\_information.type, car\_information.location, reservation.pickup\_date, reservation.return\_date, car\_information.model\_name, reservation.estimated\_cost, car\_information.vehicle\_sno

FROM car\_information

LEFT JOIN reservation ON car\_information.vehicle\_sno= reservation.vehicle\_sno

WHERE reservation.return\_date>NOW()

AND reservation.username = username

SELECT car\_information.location, reservation.pickup\_date, car\_information.model\_name, reservation.estimated\_cost, reservation.return\_status

FROM car\_information

LEFT JOIN reservation ON car\_information.vehicle\_sno= reservation.vehicle\_sno

WHERE reservation.return\_date>NOW()

AND reservation.username = username

SELECT reservation.return\_date FROM reservation

WHERE reservation.vehicle\_sno = affectedCar

AND reservation.return\_date>NOW()

AND reservation.return\_date< returnTimeString

SELECT reservation.estimated\_cost, reservation.return\_date, car\_information.hourly\_rate

FROM reservation LEFT JOIN car\_information ON reservation.vehicle\_sno = car\_information.vehicle\_sno

WHERE reservation.vehicle\_sno = affectedCar AND reservation.username = username

UPDATE reservation SET estimated\_cost = cost

return\_date= returnTimeString

WHERE vehicle\_sno = affectedCar AND return\_date = oldDate

SELECT username FROM reservation WHERE vehicle\_sno = affectedCar

AND pickup\_date BETWEEN oldDate AND returnTimeString

UPDATE reservation SET late\_fees = hoursLate

WHERE return\_date = returnTimeString

AND username = username

SELECT phone FROM gt\_student WHERE username = username

1. **Admin Report**

SELECT reservation.vehicle\_sno,

TYPE , model\_name, SUM( estimated\_cost ) AS total\_cost, SUM( late\_fees ) AS total\_late

FROM reservation

LEFT JOIN car\_information ON reservation.vehicle\_sno

GROUP BY vehicle\_sno

LIMIT 0 , 30

1. **Driving Plans**

SELECT \* FROM driving\_plan

1. **Frequent User Report**

SELECT reservation.username, plan\_type, COUNT( \* ) FROM reservation

LEFT JOIN gt\_student ON reservation.username = gt\_student.username

WHERE reservation.pickup\_date BETWEEN DATE\_ADD( NOW( ) , INTERVAL -3 MONTH) AND NOW( )

GROUP BY reservation.username

ORDER BY COUNT( \* ) DESC LIMIT 0 , 5

1. **Location Preference Report**

(SELECT '3 months ago' as month, reservation.location\_name, COUNT(\*) as count, SUM(HOUR(TIMEDIFF(reservation.return\_date , reservation.pickup\_date))) as hours FROM reservation

WHERE reservation.pickup\_date BETWEEN DATE\_ADD(NOW(), INTERVAL -3 MONTH) AND DATE\_ADD(NOW(), INTERVAL -2 MONTH)

GROUP BY reservation.location\_name

ORDER BY count DESC)

UNION

(SELECT '2 months ago' as month, reservation.location\_name, COUNT(\*) as count, SUM(HOUR(TIMEDIFF(reservation.return\_date , reservation.pickup\_date))) as hours FROM reservation

WHERE reservation.pickup\_date BETWEEN DATE\_ADD(NOW(), INTERVAL -2 MONTH) AND DATE\_ADD(NOW(), INTERVAL -1 MONTH)

GROUP BY reservation.location\_name

ORDER BY count DESC)

UNION

(SELECT '1 month ago' as month, reservation.location\_name, COUNT(\*) as count, SUM(HOUR(TIMEDIFF(reservation.return\_date , reservation.pickup\_date))) as hours FROM reservation

WHERE reservation.pickup\_date BETWEEN DATE\_ADD(NOW(), INTERVAL -1 MONTH) AND NOW()

GROUP BY reservation.location\_name

ORDER BY count DESC)

1. **Maintenance History Report**

SELECT car\_information.vehicle\_sno, request\_date, employee\_username, description, model\_name FROM maintenance\_request JOIN car\_information ON car\_information.vehicle\_sno = maintenance\_request.vehicle\_sno

GROUP BY car\_information.vehicle\_sno

ORDER BY COUNT(car\_information.vehicle\_sno)

1. Login Check

Select \* FROM users WHERE username= username AND password = password

2. New user

INSERT INTO users(username, password, usertype) VALUES (username, password, usertype)

3. Update Personal Information

UPDATE gt\_student SET

gt\_student.first\_name=IFNULL(firstname1, gt\_student.first\_name),

gt\_student.middle\_initial=IFNULL(mi1, gt\_student.middle\_initial),

gt\_student.last\_name=IFNULL(lastname1, gt\_student.last\_name),

gt\_student.email=IFNULL(email1, gt\_student.email),

gt\_student.phone=IFNULL(phone1, gt\_student.phone),

gt\_student.plan\_type=IFNULL(planType, gt\_student.plan\_type)

WHERE gt\_student.username = username

UPDATE car\_information SET

car\_information.location =

WHERE payment\_information.username = $username

4. View Personal Information

SELECT \*

FROM gt\_student, payment\_information

WHERE gt\_student.username = username

AND payment\_information.username = payinfo

5. View Driving Plans

SELECT \* FROM driving\_plan

6. Rent a Car

SELECT car\_information.daily\_rate AS daily, car\_information.hourly\_rate AS hourly

FROM car\_information

WHERE car\_information.vehicle\_sno = $vehicle\_sno

INSERT INTO reservation (pickup\_date, return\_date, late\_by, estimated\_cost, return\_status, extended\_time, late\_fees, location\_name, vehicle\_sno, username)

VALUES ($pickup\_date, $return\_date, 0,

(IF $return\_date - $pickup\_date > 1DAY,

SELECT daily \* (($return\_date - $pickup\_date)/1DAY)

ELSE SELECT hourly \* (($return\_date - $pickup\_date)/1HOUR), 0, 0,0,

$location\_name, $vehicle\_sno, $username)

7.Car Availability

SELECT \* FROM car\_information

LEFT JOIN reservation

ON car\_information.vehicle\_sno= reservation.vehicle\_sno

WHERE reservation.pickup\_date NOT BETWEEN pickupDateString AND returnDateString

AND reservation.return\_date NOT BETWEEN pickupDateString AND returnDateString AND

car\_information.type = carType

GROUP BY car\_information.location

ORDER BY CASE WHEN car\_information.location=location then 0 else 1 end

8. View Current Rental

SELECT pickup\_date, return\_date, model\_name, estimated\_cost, location

FROM car\_information LEFT JOIN reservation

ON car\_infromation.veichle\_sno= reservation.veichle\_sno

WHERE extended\_time=0

8.1 Extending Rental

SELECT COUNT(\*) AS interfering\_reservations FROM reservation

ON $veichle\_sno= reservation.veichle\_sno

WHERE $returndate NOT BETWEEN reservation.pickup\_date AND reservation.return\_date

SELECT car\_information.daily\_rate AS daily, car\_information.hourly\_rate AS hourly

FROM car\_information

WHERE car\_information.vehicle\_sno = $vehicle\_sno

UPDATE reservation SET return\_date = $returndate,

estimated\_cost = (IF $return\_date - reservation.pickup\_date > 1DAY,

SELECT daily \* (($return\_date - reservation.pickup\_date)/1DAY)

ELSE SELECT hourly \* (($return\_date - reservation.pickup\_date)/1HOUR)

WHERE reservation.vehicle\_sno = $vehicle\_sno

AND reservation.username = $username

AND reservation.pickup\_date = $pickup\_date

8.2. Previous Reservations

Select pickup\_date, return\_date, model\_name, estimated\_cost, location\_name, return\_status FROM car\_information LEFT JOIN reservation ON car\_infromation.veichle\_sno= reservation.veichle\_sno WHERE reservation.username = $username

9. Add Car

INSERT INTO car\_information VALUES ($vehicle\_sno, $auxiliary\_cable, $transmission\_type,

$seating\_capacity, $blue\_tooth\_connectivity, $daily\_rate, $hourly\_rate, $color, $type, $model\_name, $under\_maitenence\_flag, $location)

10. Change Car’s Location

UPDATE car\_information SET

car\_information.location=$newLocation

WHERE car\_information.location = $oldLocation

AND car\_information.type = $type

AND car\_information.model\_name = $model\_name

AND car\_information.color = $color

AND car\_information.seating\_capacity = $seating\_capacity

AND car\_information.transmission\_type = $transmission\_type

11. Request Maintenance

INSERT INTO maintenance\_request VALUES ($request\_date, $vehicle\_sno, $description, $employee\_username)

12.1 Pull up Rental Information

SELECT car\_information.type, car\_information.location, reservation.pickup\_date, reservation.return\_date, FROM car\_information LEFT JOIN reservation ON car\_infromation.veichle\_sno= reservation.veichle\_sno

WHERE veichle\_sno = (SELECT vehicle\_sno FROM car\_information LEFT JOIN reservation ON car\_infromation.veichle\_sno= reservation.veichle\_sno WHERE return\_date>GETDATE())

12.2 Request Rental Change

SELECT vehicle\_sno AS affectedCar FROM reservation

WHERE veichle\_sno = (SELECT vehicle\_sno FROM car\_information LEFT JOIN reservation ON car\_infromation.veichle\_sno= reservation.veichle\_sno WHERE return\_date>GETDATE())

SELECT return\_date AS oldDate FROM reservation

WHERE vehicle\_sno = affectedCar

AND return\_date>GETDATE()

UPDATE reservation SET estimated\_cost = (estimated\_cost+(($return\_time-return\_time)\*(SELECT car\_information.hourly\_rate WHERE car\_information.vehicle\_sno = affectedCar)),

return\_date=$return\_date

WHERE veichle\_sno = affectedCar

SELECT username AS affectedUser FROM reservation WHERE vehicle\_sno = affectedCar

AND return\_date BETWEEN oldDate AND $return\_date

IF affectedUser IS NOT NULL {

UPDATE reservation SET late\_fees = ($return\_time-oldTime)\*50

}

12.3 Assign Late Fees //deprecated, do not use

SELECT username From SELECT vehicle\_sno FROM car\_information LEFT JOIN reservation ON car\_infromation.veichle\_sno= reservation.veichle\_sno WHERE veichlereturn\_date, pickup\_date from WHERE return\_date

13. Maintenance History Report

SELECT request\_date, car\_information.vehicle\_sno, employee\_username, description, model\_name FROM maintenance\_request JOIN car\_information ON car\_information.vehicle\_sno

GROUP BY car\_information.vehicle\_sno

ORDER BY COUNT(car\_information.vehicle\_sno)

14. Administrative Report

SELECT reservation.vehicle\_sno,

TYPE , model\_name, SUM( estimated\_cost ) AS total\_cost, SUM( late\_fees ) AS total\_late

FROM reservation

LEFT JOIN car\_information ON reservation.vehicle\_sno

GROUP BY vehicle\_sno

LIMIT 0 , 30

15. Location Preference Report

SELECT '3 months ago' as month, reservation.location\_name, COUNT(\*) as count, SUM(HOUR(TIMEDIFF(reservation.return\_date , reservation.pickup\_date))) as hours FROM reservation

WHERE reservation.pickup\_date BETWEEN DATE\_ADD(NOW(), INTERVAL -3 MONTH) AND DATE\_ADD(NOW(), INTERVAL -2 MONTH)

GROUP BY reservation.location\_name

ORDER BY count DESC)

UNION

SELECT '2 months ago' as month, reservation.location\_name, COUNT(\*) as count, SUM(HOUR(TIMEDIFF(reservation.return\_date , reservation.pickup\_date))) as hours

WHERE reservation.pickup\_date BETWEEN DATE\_ADD(NOW(), INTERVAL -2 MONTH) AND DATE\_ADD(NOW(), INTERVAL -1 MONTH)

GROUP BY reservation.location\_name

ORDER BY count DESC)

UNION

(SELECT '1 month ago' as month, reservation.location\_name, COUNT(\*) as count, SUM(HOUR(TIMEDIFF(reservation.return\_date , reservation.pickup\_date))) as hours FROM reservation

WHERE reservation.pickup\_date BETWEEN DATE\_ADD(NOW(), INTERVAL -1 MONTH) AND NOW()

GROUP BY reservation.location\_name

ORDER BY count DESC)

16. Frequent Users Report

SELECT reservation.username, plan\_type, COUNT( \* ) FROM reservation

LEFT JOIN gt\_student ON reservation.username = gt\_student.username

WHERE reservation.pickup\_date BETWEEN DATE\_ADD( NOW( ) , INTERVAL -3 MONTH) AND NOW( )

GROUP BY reservation.username

ORDER BY COUNT( \* ) DESC LIMIT 0 , 5

**ASSUMPTIONS MADE:**

We assume that GETDATE() returns a current timestamp, and that we can add and subtract amounts of time from that timestamp.

We assume that MONTH() will round down to the first day of the month that is passed in as a parameter.