[](http://www.egitimevi.net/)

MIDDLE EAST TECHNICAL UNIVERSITY

IS 503

DATABASE CONCEPTS AND APPLICATIONS

PROJECT TOPIC: RIDESHARING APPLICATION DATABASE DESIGN APPLICATION

**NAME, SURNAME: RUSTEM OZAN OZDEMIR**

**STUDENT NUMBER: 1938935**

**NAME, SURNAME: MAHMUT UĞUR BÖCÜ**

**STUDENT NUMBER: 2459055**

**1.PROJECT DESIGN**

We have designed database for a Ridesharing Application like Uber where:

1. Customer can make request
2. Customer pays Money to driver for service
3. Customer can review his/her trips
4. Drivers have trip and car

The Project is developed by using MySQL Workbench.

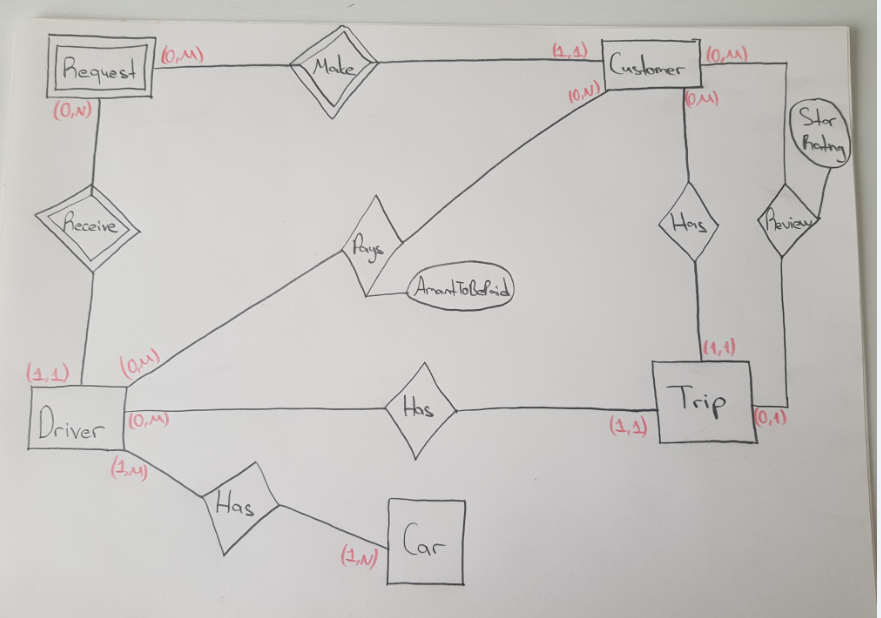
**2.ASSUMPTIONS**

In this Project, our assumptions can be seen as follows:

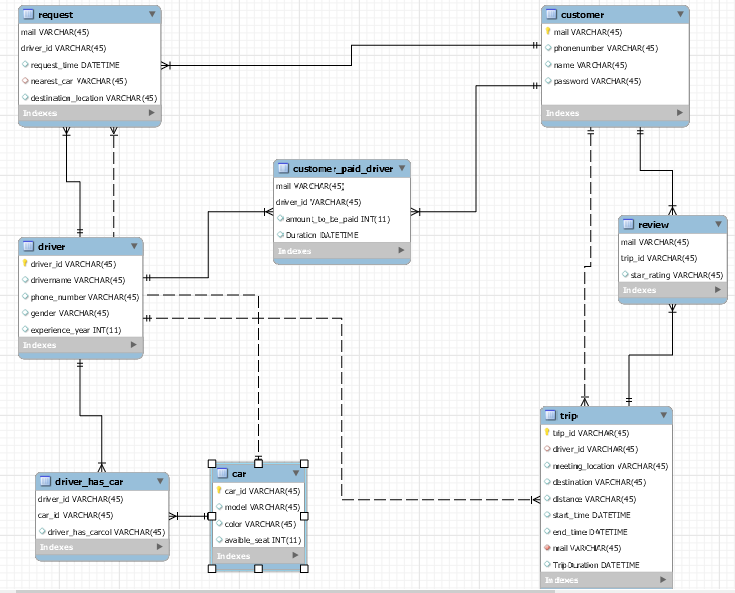
1. Each driver must own car
2. A car can be shared by different drivers
3. A trip cannot be shared by drivers
4. A request assigns to a driver by the system
5. All customers can review their trips
6. Star rating range is between 1 and 5.

**3.SCHEMA DESIGN**

ER Diagram is used for schema design. It can be seen as follows:



*ER Diagram*



*ER Diagram OF Ridesharing Application (Crow’s Foot)*

**Entities:**

1. Customer
2. Driver
3. Car
4. Trip

**Relations:**

1. Driver\_has\_car
2. Customer\_paid\_driver
3. Review
4. Request

**Relational Mapping**

**Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| mail | phonenumber | name | password |

Primary key: mail

**Driver**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| driver\_id | drivername | phone\_number | gender | experience\_year |

Primary key: driver\_id

**Car**

|  |  |  |  |
| --- | --- | --- | --- |
| car\_id | model | color | avaible\_seat |

Primary key: car\_id

**Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| trip\_id | driver\_id | meeting\_location | destination | distance | start\_time | end\_time | mail | TripDuration |

Primary key: trip\_id

Foreign keys: “driver\_id references driver\_id in driver Table”. “mail references mail in customer Table”.

**Customer\_paid\_driver**

|  |  |  |
| --- | --- | --- |
| mail | driver\_id | amount\_to\_be\_paid |

Primary keys: Mail and driver\_id

Foreign keys: “driver\_id references driver\_id in driver Table”. “mail references mail in customer Table”.

**Driver\_has\_car**

|  |  |
| --- | --- |
| driver\_id | car\_id |

Primary keys: driver\_id and car\_id

Foreign keys: “driver\_id references driver\_id in driver Table”. “car\_id references car\_id in car Table”.

**Request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mail | driver\_id | request\_time | nearest\_car | destination\_location |

Primary keys: mail and driver\_id

Foreign keys: “driver\_id references driver\_id in driver Table”. “mail references mail in customer Table”. “nearest\_car references car\_id in car Table”.

**Review**

|  |  |  |
| --- | --- | --- |
| mail | trip\_id | star\_rating |

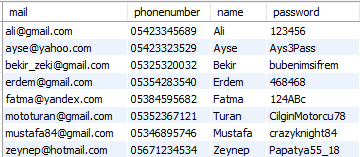
Primary keys: mail and trip\_id

Foreign keys: “mail references mail in customer Table”. “trip\_id references trip\_id in trip Table”.

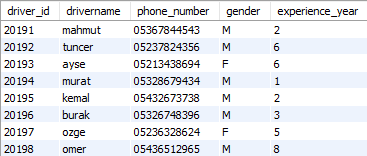
**4. TABLES AND QUERIES**

**4.1 TABLES AND TUPLES**

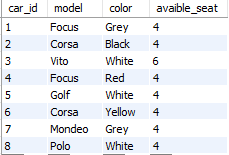
Customer Table



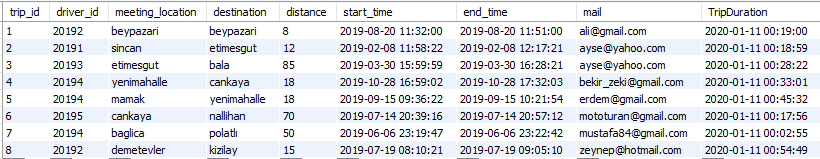
Driver Table



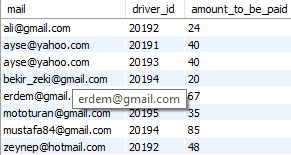
Car Table



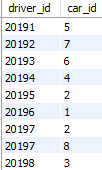
Trip Table



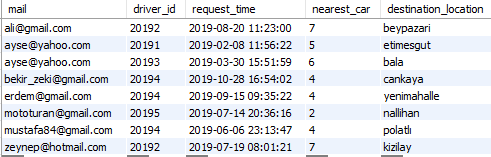
Customer\_paid\_driver Table



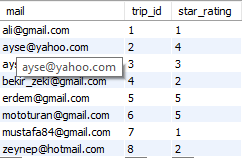
Driver\_has\_car Table



Request Table



Review Table



**SQL Of Creation Of Tables:**

**Customer Table**

CREATE TABLE `customer` (

`mail` varchar(45) NOT NULL,

`phonenumber` varchar(45) DEFAULT NULL,

`name` varchar(45) DEFAULT NULL,

`password` varchar(45) DEFAULT NULL,

PRIMARY KEY (`mail`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Driver Table**

CREATE TABLE `customer\_paid\_driver` (

`mail` varchar(45) NOT NULL,

`driver\_id` varchar(45) NOT NULL,

`amount\_to\_be\_paid` int(11) DEFAULT NULL,

`Duration` datetime DEFAULT NULL,

PRIMARY KEY (`mail`,`driver\_id`),

KEY `driver\_id\_paid\_idx` (`driver\_id`),

CONSTRAINT `driver\_id\_paid` FOREIGN KEY (`driver\_id`) REFERENCES `driver` (`driver\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `mail\_paid` FOREIGN KEY (`mail`) REFERENCES `customer` (`mail`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Car Table**

CREATE TABLE `car` (

`car\_id` varchar(45) NOT NULL,

`model` varchar(45) DEFAULT NULL,

`color` varchar(45) DEFAULT NULL,

`avaible\_seat` int(11) DEFAULT NULL,

PRIMARY KEY (`car\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Trip Table**

CREATE TABLE `trip` (

`trip\_id` varchar(45) NOT NULL,

`driver\_id` varchar(45) DEFAULT NULL,

`meeting\_location` varchar(45) DEFAULT NULL,

`destination` varchar(45) DEFAULT NULL,

`distance` varchar(45) DEFAULT NULL,

`start\_time` datetime DEFAULT NULL,

`end\_time` datetime DEFAULT NULL,

`mail` varchar(45) NOT NULL,

`TripDuration` datetime GENERATED ALWAYS AS (timediff(`end\_time`,`start\_time`)) STORED,

PRIMARY KEY (`trip\_id`),

KEY `mail\_idx` (`mail`),

KEY `driver\_id\_idx` (`driver\_id`),

CONSTRAINT `driver\_id\_trip` FOREIGN KEY (`driver\_id`) REFERENCES `driver` (`driver\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `mail\_trip` FOREIGN KEY (`mail`) REFERENCES `customer` (`mail`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Customer\_paid\_driver Table**

CREATE TABLE `customer\_paid\_driver` (

`mail` varchar(45) NOT NULL,

`driver\_id` varchar(45) NOT NULL,

`amount\_to\_be\_paid` int(11) DEFAULT NULL,

PRIMARY KEY (`mail`,`driver\_id`),

KEY `driver\_id\_paid\_idx` (`driver\_id`),

CONSTRAINT `driver\_id\_paid` FOREIGN KEY (`driver\_id`) REFERENCES `driver` (`driver\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `mail\_paid` FOREIGN KEY (`mail`) REFERENCES `customer` (`mail`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Driver\_has\_car Table**

CREATE TABLE `driver\_has\_car` (

`driver\_id` varchar(45) NOT NULL,

`car\_id` varchar(45) NOT NULL,

PRIMARY KEY (`driver\_id`,`car\_id`),

KEY `car\_id\_driverhascar\_idx` (`car\_id`),

CONSTRAINT `car\_id\_driverhascar` FOREIGN KEY (`car\_id`) REFERENCES `car` (`car\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `driver\_id\_driverhascar` FOREIGN KEY (`driver\_id`) REFERENCES `driver` (`driver\_id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Request Table**

CREATE TABLE `request` (

`mail` varchar(45) NOT NULL,

`driver\_id` varchar(45) NOT NULL,

`request\_time` datetime DEFAULT NULL,

`nearest\_car` varchar(45) DEFAULT NULL,

`destination\_location` varchar(45) DEFAULT NULL,

PRIMARY KEY (`mail`,`driver\_id`),

KEY `driver\_id\_request\_idx` (`driver\_id`),

KEY `nearest\_car\_idx` (`nearest\_car`),

CONSTRAINT `driver\_id\_request` FOREIGN KEY (`driver\_id`) REFERENCES `driver` (`driver\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `mail\_request` FOREIGN KEY (`mail`) REFERENCES `customer` (`mail`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `nearest\_car` FOREIGN KEY (`nearest\_car`) REFERENCES `car` (`car\_id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**Review Table**

CREATE TABLE `review` (

`mail` varchar(45) NOT NULL,

`trip\_id` varchar(45) NOT NULL,

`star\_rating` varchar(45) DEFAULT NULL,

PRIMARY KEY (`mail`,`trip\_id`),

KEY `trip\_id\_review\_idx` (`trip\_id`),

CONSTRAINT `mail\_review` FOREIGN KEY (`mail`) REFERENCES `customer` (`mail`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `trip\_id\_review` FOREIGN KEY (`trip\_id`) REFERENCES `trip` (`trip\_id`) ON DELETE CASCADE ON UPDATE CASCADE

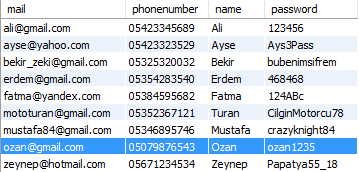
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

**4.2 QUERIES**

Making insert, update and deletion automatically.

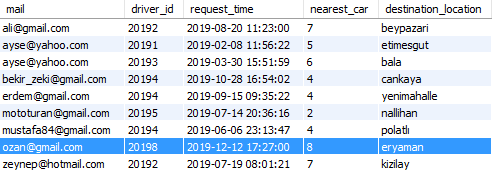
**4.2.1.1 INSERT a new customer to customer Table**

INSERT INTO `project`.`customer` (`mail`, `phonenumber`, `name`, `password`) VALUES ('[ozan@gmail.com](mailto:ozan@gmail.com)', '05079876543', 'Ozan', 'ozan1235');



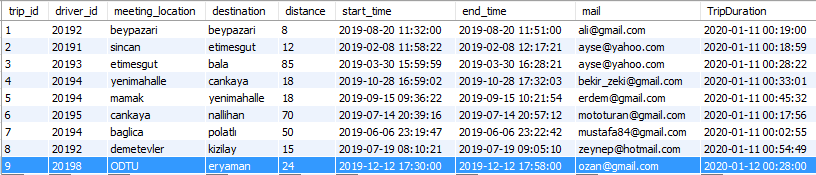
**4.2.1.2 INSERT a request of that user to request Table**

INSERT INTO `project`.`request` (`mail`, `driver\_id`, `request\_time`, `nearest\_car`, `destination\_location`) VALUES ('[ozan@gmail.com](mailto:ozan@gmail.com)', '20198', '2019-12-12 17:27:00', '8', 'eryaman');



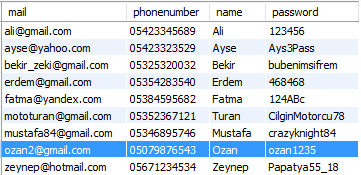
**4.2.1.3 INSERT a trip of that user to trip Table**

INSERT INTO `project`.`trip` (`trip\_id`, `driver\_id`, `meeting\_location`, `destination`, `distance`, `start\_time`, `end\_time`, `mail`) VALUES ('9', '20198', 'ODTU', 'eryaman', '24', '2019-12-12 17:30:00', '2019-12-12 17:58:00', '[ozan@gmail.com](mailto:ozan@gmail.com)');



**4.2.2.1 UPDATE existing customer mail in customer Table (cascading effects on other tables)**

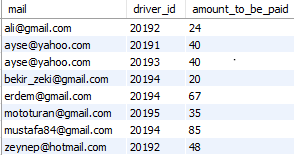
UPDATE `project`.`customer` SET `mail` = '[ozan2@gmail.com](mailto:ozan2@gmail.com)' WHERE (`mail` = '[ozan@gmail.com](mailto:ozan@gmail.com)');

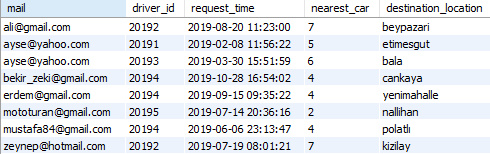


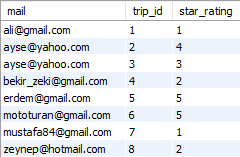
**4.2.3.1 DELETE existing customer in customer Table (cascading effects on other tables)**

DELETE FROM `project`.`customer` WHERE (`mail` = '[ozan2@gmail.com](mailto:ozan2@gmail.com)');

The related tables (Customer\_paid\_driver, request and review tables) back to original form after deleting operations regarding ozan2@gmail.com.

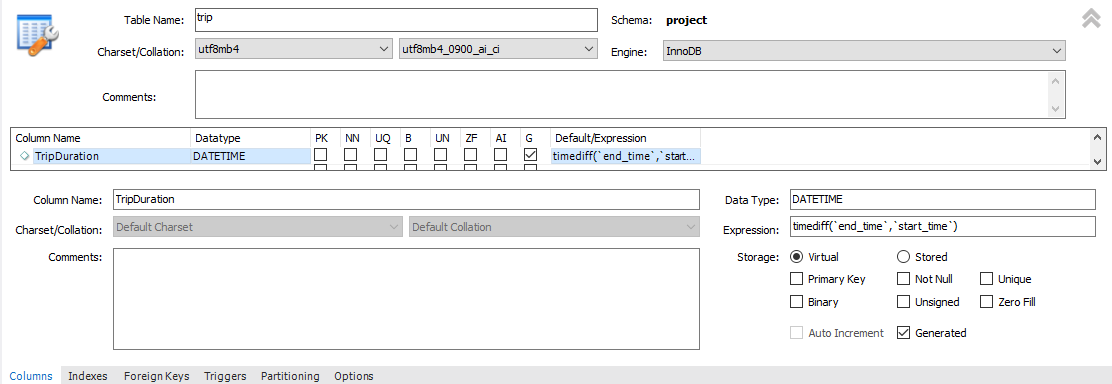






**4.2.4 DERIVED ATTRIBUTE**

Trip duration can be calculated from start\_time of the trip and the end\_time of the same trip.



**4.2.5.1 AGGREGATE QUERY**

1.      In order to find average distance of trips. The trip table can be calculated as follows:

SELECT AVG(project.trip.distance) AS AVERAGEOFDISTANCE

FROM project.trip



2.      In order to find the maximum number of trips made by same customer, the following query can be used:

SELECT  MAX(TOTAL)

From (SELECT COUNT(\*)  AS TOTAL FROM project.trip

Group By project.trip.mail) Results



**4.2.5.2 COMPLEX JOIN QUERY**

To find the certain driver attributes who carries ‘[ayse@yahoo.com](mailto:ayse@yahoo.com)’ can be found as follows:

SELECT project.trip.trip\_id, project.trip.mail, project.driver.driver\_id, project.driver.drivername, project.driver.phone\_number

FROM project.driver, project.trip

WHERE project.driver.driver\_id = project.trip.driver\_id AND project.trip.mail = '[ayse@yahoo.com](mailto:ayse@yahoo.com)'



To find sum of total star ratings of drivers with their names can be found as follows:

Select T.drivername, SUM(T.star\_rating)

From (

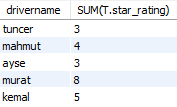
Select project.driver.drivername, project.driver.driver\_id, project.review.star\_rating, project.review.trip\_id

From project.driver, project.review, project.trip

Where project.driver.driver\_id = project.trip.driver\_id AND project.review.trip\_id = project.trip.trip\_id

) AS T

Group By T.drivername



**4.2.6 VIEWS**

1. Create a view to determine total number of stars for each driver.

USE `project`;

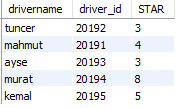
CREATE OR REPLACE VIEW `vw1` AS

Select project.driver.drivername, project.driver.driver\_id, SUM(project.review.star\_rating) AS STAR

From project.driver, project.review, project.trip

Where project.driver.driver\_id = project.trip.driver\_id AND project.review.trip\_id = project.trip.trip\_id

Group By project.driver.drivername;;



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `project`.`vw1` AS

SELECT

`project`.`driver`.`drivername` AS `drivername`,

`project`.`driver`.`driver\_id` AS `driver\_id`,

SUM(`project`.`review`.`star\_rating`) AS `STAR`

FROM

((`project`.`driver`

JOIN `project`.`review`)

JOIN `project`.`trip`)

WHERE

((`project`.`driver`.`driver\_id` = `project`.`trip`.`driver\_id`)

AND (`project`.`review`.`trip\_id` = `project`.`trip`.`trip\_id`))

GROUP BY `project`.`driver`.`drivername`

2. Create a view to determine destination location of drivers

USE `project`;

CREATE OR REPLACE VIEW `vw2` AS

select D.drivername, R.destination\_location

from project.request R, project.driver D

where R.driver\_id = D.driver\_id;



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `project`.`vw2` AS

SELECT

`d`.`drivername` AS `drivername`,

`r`.`destination\_location` AS `destination\_location`

FROM

(`project`.`request` `r`

JOIN `project`.`driver` `d`)

WHERE

(`r`.`driver\_id` = `d`.`driver\_id`)

**4.2.7 ERROR HANDLING AND TRIGGER**

Foreign key constraints are done by using insert, update and delete operations in section 4.2.1, 4.2.2 and 4.2.3.

Trigger which can be seen below control updated number of seats in car. If the updated number of seats is greater than 6, it sets previous number of seat. For example, if the updated seat number is 7 in car\_id=1, trigger control seat number and final seat number will be previous seat number which is 4.

delimiter |

CREATE TRIGGER t1 BEFORE UPDATE ON project.car

FOR EACH ROW

BEGIN

IF NEW.avaible\_seat > 6 THEN

SET NEW.avaible\_seat = OLD.avaible\_seat;

END IF;

END;

|

delimiter ;

Trigger which can be seen below control inserted number of seats in car. If the inserted number of seats is greater than 6, it sets number of seat “NULL”. For example, if the inserted seat number is 7 in new tuple , trigger control seat number and final seat number will be “NULL”.

delimiter |

CREATE TRIGGER t2 BEFORE INSERT ON project.car

FOR EACH ROW

BEGIN

IF NEW.avaible\_seat > 6 THEN

SET NEW.avaible\_seat = NULL;

END IF;

END;

|

delimiter ;