**RUSHABH KADAM 180001045**

**JEMIN VAGADIA 180001023**

Indian Institute of Technology, Indore

CS 257

DBIS Project

**DBMS PROJECT:**

**RTO DATABASE MANAGEMENT SYSTEM:**

**INTRODUCTION:**

* **RTO needs a database to manage information regarding vehicles, vehicle owner details also the people who break traffic rules. RTO will have to store its department and employee details too.**
* **RTO management will be having a lot of work regarding registration of vehicles and issue of driver’s license. RTO related problems will be solved by storing all the information related to vehicle and driver at database by RTO administrator.**
* **RTO DBMS will be keeping a view to make the existing registration and issues of information and about license easier and faster.**
* **The system provides information regarding the RTO application.**
* **RTO DATABASE will contain information about license applicant, license holder, vehicle registration, rules offended.**

**PROBLEM FACED:**

1. **DIFFICULTY IN REGISTRATION:** In many villages, there is only one day camp of RTO and the people who want driving license they should remain present on that day if they missed that day then they have to go to the district RTO office. So it is a disadvantage because that may not be able to go or he having worked on that day. so that here we are developing one web application which provides easiest and efficient way for RTO works like making driving license, the registration number of the vehicle.
2. **REMOVING THIRD PARTY INTERVENTION:**In many cases, we found that RTO office work gets completed through the third party called agent. When a person goes to the RTO office for a driving license, vehicle passing, and registration number of the vehicle then a person go through the agent and agent will complete person work by taking a lot of money and that person is unaware of all these systems. According to the TOI new on dated 3 September 2015 RTO office is the more corrupted area. So using our web application we are somewhere to help to reduce corruption

* **PROPOSED SOLUTION:**Here, we are developing a web application for RTO so here we give a brief description of our project overview. First, we provide familiar environment means the needy user can access this site for their work purpose related to RTO. First user needs to fill the registration form so that we provide authentication to him and then user can choose option he wants means if he select to making a driving license then we provide driving license requirement details and give available date to him so that he come on that date direct give the test so that he can save his time as well as money. If user wants to pass his vehicle number then also it takes time in old system but here we provide facility that user he buy new vehicle he should have to first register on our site and fill all the required and importance details of vehicle and we gives this details to RTO office directly so that this work will get complete within less time and the user get his number template easily. The administrator is providing for authentication purpose as well as it handles all the database of RTO and manages all the process. He has authority to approved learning license number, permanent license number; pass the vehicle registration number, etc. Facilities are provided by the administrator.
* **ER Analysis**: Identifying Entity Sets and Relationship Sets

1. APPLICANT (ENTITY)
2. DEPARTMENT (ENTITY)
3. E\_D (RELATION B/W EMPLOYEE AND DEPARTMENT)
4. EMAIL (ENTITY)
5. EMPLOYEE(ENTITY)
6. EMPLOYEE\_DETAILS (RELATION B/W EMPLOYEE AND PERSONAL)
7. LICENSE (ENTITY)
8. LICENSE HOLDER (RELATION B/W LICENSE AND PERSONAL)
9. LICENSE\_TEST (ENTITY)
10. PERSONAL (ENTITY)
11. REGISTRATION (RELATION BETWEEN VEHICLES AND PERSONAL)
12. RULES (ENTITY)
13. RULES\_CONTRAVENED (RELATION B/W VEHICLE, LICENSE, PERSONAL AND RULES)
14. VEHICLES (ENTITY)

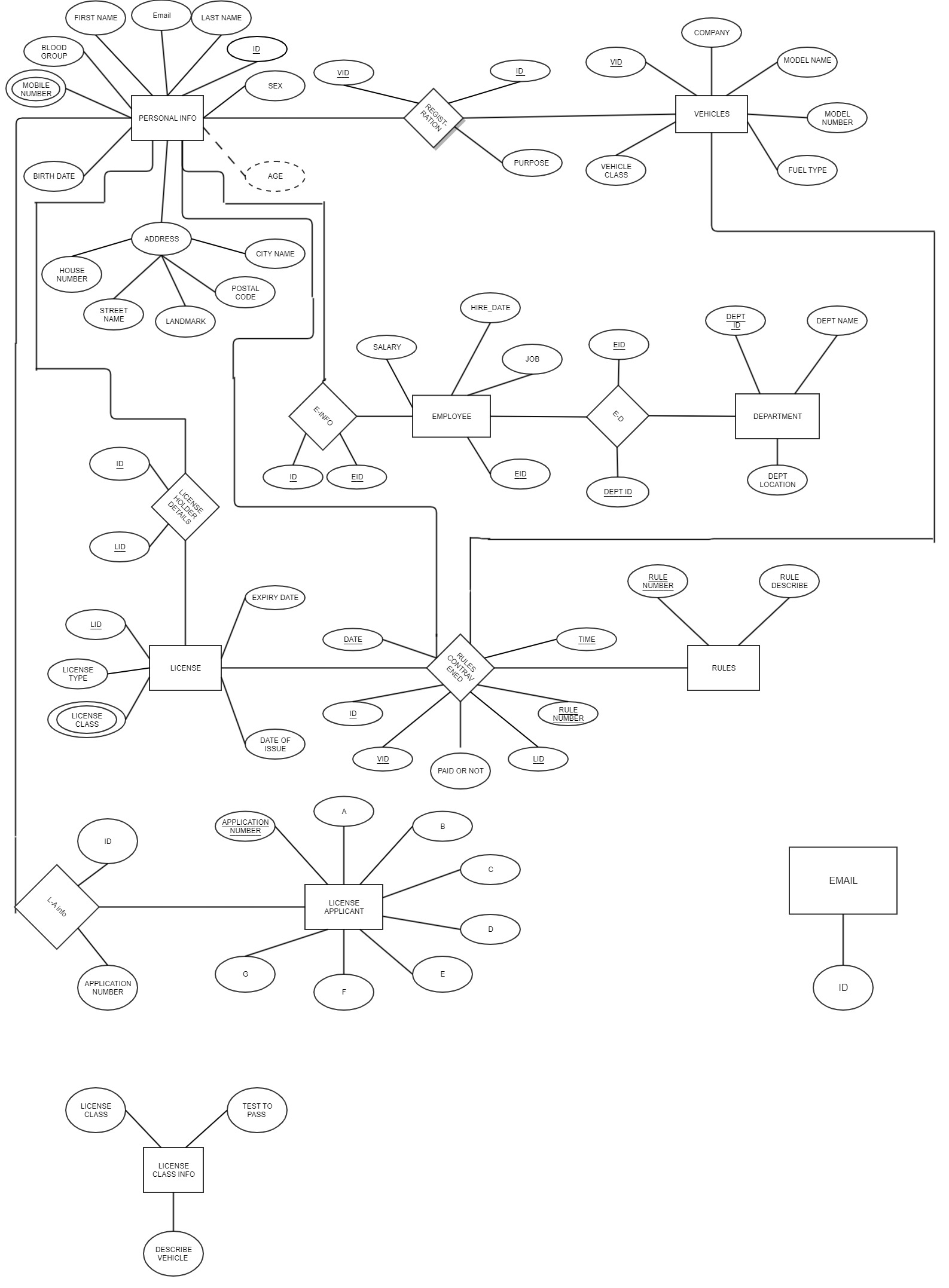
* **ENTITY DEFINITIONS**:

1. APPLICANT: This entity set will contain information about license applicant. it will contain ID, A, B, C, D, E attributes where a, b, c, d, e are types of license class these attributes will store value APPLIED, N.A., PASS, FAIL values only.
2. DEPARTMENT: This entity set will contain department number, department name, department location.
3. E\_D: This is a relationship between employee and department. This will contain EID, DEPARTMENT\_ID AS attributes.
4. EMAIL: This table will store the ID of the person whom the mail is sent by RTO for any reason
5. EMPLOYEE:This entity set contains information regarding employees like ID, employee ID, salary, job and hire date.
6. EMPLOYEE\_DETAILS : This table will relate EID AND ID of the employee so that personal information of the employee need not be filled again somewhere else.
7. LICENSE: This entity set contains personal details about license holder such as license holder name, birth date, age, blood group, address, mobile number, sex.
8. LICENSE DETAILS: This entity will contain details of License details such as LID, expiry date, license class (A, B, C, D, E which denotes

MC 50 CC / MCWOG OR FOR / LMV-NT /MGV /…), DATE OF ISSUE.

1. LICENSE\_TEST: This will define license class, test to be passed for that license and type of vehicle for license\_class.
2. PERSONAL: This is a table which will contain all personal details about any person. In real case this could be aadhar card.
3. Registration: This shows relationship between vehicle and it’s owner it will have ID, VID, and purpose to buy vehicle.
4. RULES: This entity will show rule number and it’s description.
5. RULES\_CONTRAVENED: This entity will tell about traffic rules offended for e.g. ID of the person who broke the law, VID of the vehicle, LID (if any) of the driver, date, time of that, rule number offended, fine paid or not.
6. VEHICLES : This will contain information of vehicles.

.



TABLES:

CREATE TABLE PERSONAL(

ID INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

FIRST\_NAME VARCHAR(255) NOT NULL,

LAST\_NAME VARCHAR(255) NOT NULL,

EMAIL VARCHAR(255) NOT NULL,

SEX CHAR(1) NOT NULL,

MOBILE\_NUMBER INT(20) NOT NULL,

BLOOD\_GROUP VARCHAR(255) NOT NULL,

BIRTH\_DATE DATE NOT NULL,

AGE INT AS (YEAR(CURRENT\_TIMESTAMP) - YEAR(BIRTH\_DATE)),

HOUSE\_NUMBER INT,

STREET\_NAME VARCHAR(255),

LANDMARK VARCHAR(255),

POSTAL\_CODE INT,

CITY\_NAME VARCHAR(255)

);

CREATE TABLE VEHICLES(

VID INT PRIMARY KEY NOT NULL,

COMPANY VARCHAR(255),

MODEL\_NAME VARCHAR(255),

MODEL\_NUMBER INT,

VEHICLE\_CLASS VARCHAR(255),

FUEL\_TYPE VARCHAR(255)

);

CREATE TABLE REGISTRATION(

VID INT NOT NULL,

ID INT NOT NULL,

PURPOSE VARCHAR(255)

);

CREATE TABLE EMAIL(

ID INT NOT NULL

);

CREATE TABLE EMPLOYEE(

EID INT PRIMARY KEY NOT NULL,

JOB VARCHAR(255),

SALARY VARCHAR(255),

HIREDATE DATE

);

CREATE TABLE DEPARTMENT(

DEPT\_ID INT PRIMARY KEY NOT NULL,

DEPT\_NAME VARCHAR(255),

DEPT\_LOCATION VARCHAR(255)

);

CREATE TABLE LICENSE(

LID INT PRIMARY KEY NOT NULL,

DATE\_OF\_ISSUE DATE NOT NULL,

A VARCHAR(255),

B VARCHAR(255),

C VARCHAR(255),

D VARCHAR(255),

E VARCHAR(255)

);

CREATE TABLE LICENSE\_HOLDER(

ID INT PRIMARY KEY NOT NULL,

LID INT NOT NULL

);

CREATE TABLE EMPLOYEE\_DETAILS(

ID INT PRIMARY KEY NOT NULL,

EID INT NOT NULL

);

CREATE TABLE RULES(

RULE\_NUMBER INT PRIMARY KEY NOT NULL,

DECRIPTION VARCHAR(255) NOT NULL,

FINE INT

);

CREATE TABLE RULES\_CONTRAVENED(

ID INT NOT NULL,

VID INT NOT NULL,

LID INT,

RULE\_NUMBER INT NOT NULL,

RDATE DATETIME,

PAID\_OR\_NOT VARCHAR(20) DEFAULT 'NOT PAID'

);

CREATE TABLE LICENSE\_TEST(

LICENSE\_CLASS VARCHAR(255) PRIMARY KEY NOT NULL,

TEST VARCHAR(255) ,

VEHICLE VARCHAR(255)

);

CREATE TABLE L\_A(

ID INT NOT NULL,

APPLICATION\_NUMBER INT PRIMARY KEY NOT NULL AUTO\_INCREMENT

);

CREATE TABLE APPLICANT(

APPLICATION\_NUMBER INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

A VARCHAR(255),

B VARCHAR(255),

C VARCHAR(255),

D VARCHAR(255),

E VARCHAR(255)

);

CREATE TABLE DEPARTMENT(

DEPT\_ID INT PRIMARY KEY,

DEPT\_NAME VARCHAR(255),

DEPT\_LOCATION VARCHAR(255)

);

CREATE TABLE E\_D(

EID INT PRIMARY KEY,

DEPT\_ID INT

);

TRIGGERS:-

//for expired license

CREATE OR REPLACE TRIGGER expired\_license

BEFORE INSERT

ON license

FOR EACH ROW

DELETE FROM license WHERE LID = NEW.LID AND DATEDIFF(CURDATE(), DATE\_OF\_ISSUE) > 3650;

//If already registered vehicle is to be registered email will be sent to previous owner

CREATE OR REPLACE TRIGGER registration\_verify

BEFORE INSERT

ON registration

FOR EACH ROW

INSERT INTO EMAIL(ID) select registration.ID FROM registration WHERE registration.ID = NEW.ID;

//email will be sent to rule\_offender

CREATE OR REPLACE TRIGGER rule\_offender

AFTER INSERT

ON rules\_contravened

FOR EACH ROW

INSERT INTO EMAIL(ID) SELECT rules\_contravened.ID FROM rules\_contravened WHERE rules\_contravened = NEW.ID;

//After applying for license email will be sent to applicant for further information

CREATE OR REPLACE TRIGGER license\_notification

AFTER INSERT

ON applicant

FOR EACH ROW

INSERT INTO EMAIL(ID) SELECT applicant.ID FROM applicant WHERE applicant.ID = NEW.ID;

//Those who have given test of license are either passed or failed data of those //will be removed

DELIMITER //

CREATE OR REPLACE TRIGGER license\_pass\_fail

AFTER UPDATE

ON applicant

FOR EACH ROW

DELETE FROM applicant WHERE (ID = NEW.ID);

END//

DELIMITER ;

EVENTS:

//To remove details of those whose license is expired

DELIMITER $$

CREATE

EVENT expired\_license

ON SCHEDULE AT CURRENT\_TIMESTAMP

DO BEGIN

DELETE FROM license where DATEDIFF(CURDATE(), DATE\_OF\_ISSUE) > 3650;

END $$

DELIMITER ;

//To remove details of employees who are retired

DELIMITER $$

CREATE

EVENT employee\_age

ON SCHEDULE AT CURRENT\_TIMESTAMP

DO BEGIN

DELETE FROM employee where EID = (SELECT EID FROM employee\_details INNER JOIN personal where employee\_details.ID = personal.ID AND personal.AGE > 60);

END $$

DELIMITER ;

//delete data of those who have given license’s test

DELIMITER $$

CREATE

EVENT license\_pass

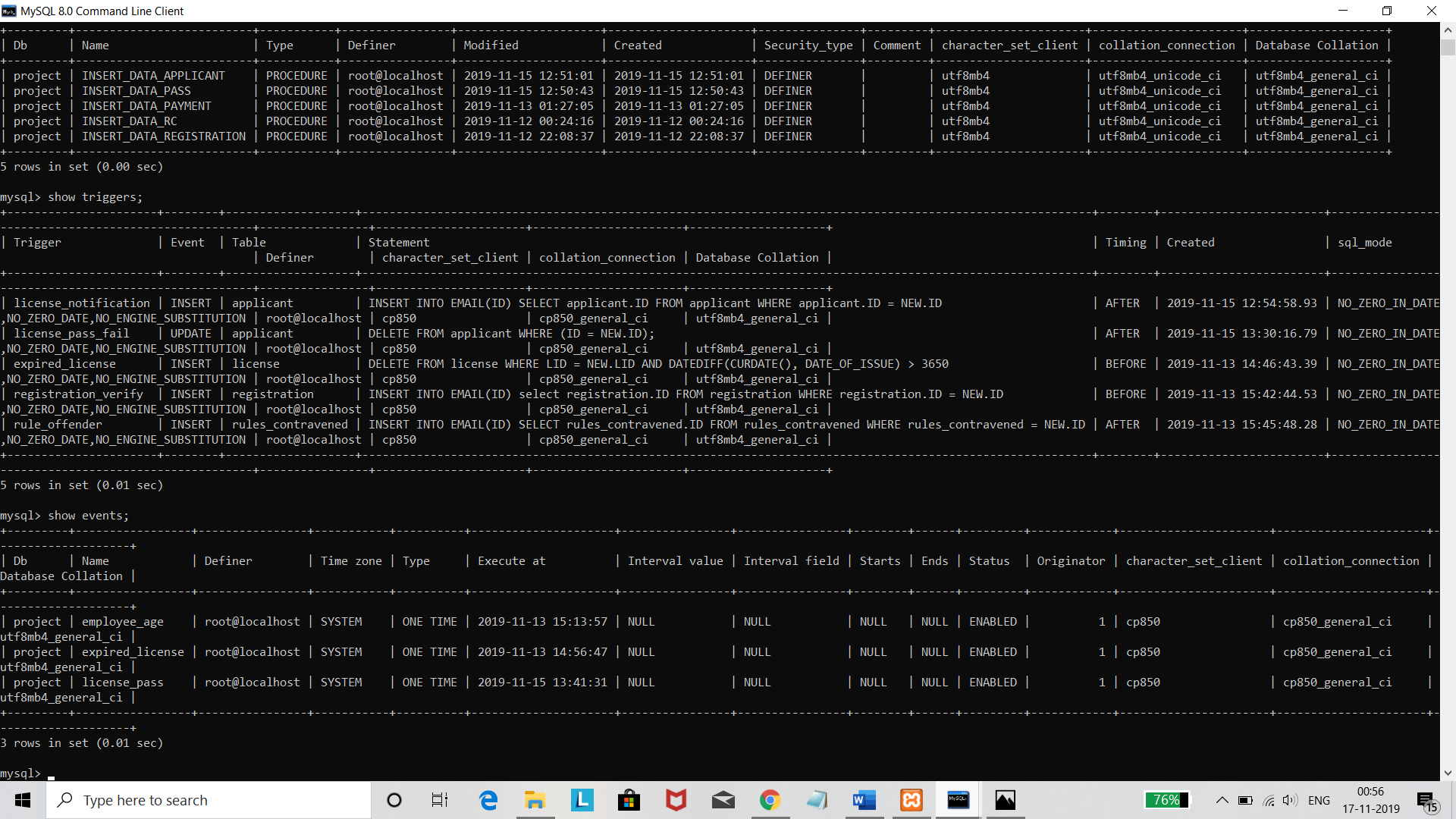
ON SCHEDULE AT CURRENT\_TIMESTAMP

DO BEGIN

DELETE FROM applicant where A = "PASS" or A = "FAIL";

END $$

DELIMITER ;



PROCEDURES:

+---------+--------------------------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+

| Db | Name | Type | Definer | Modified | Created | Security\_type | Comment | character\_set\_client | collation\_connection | Database Collation |

+---------+--------------------------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+

| project | INSERT\_DATA\_APPLICANT | PROCEDURE | root@localhost | 2019-11-15 12:51:01 | 2019-11-15 12:51:01 | DEFINER | | utf8mb4 | utf8mb4\_unicode\_ci | utf8mb4\_general\_ci |

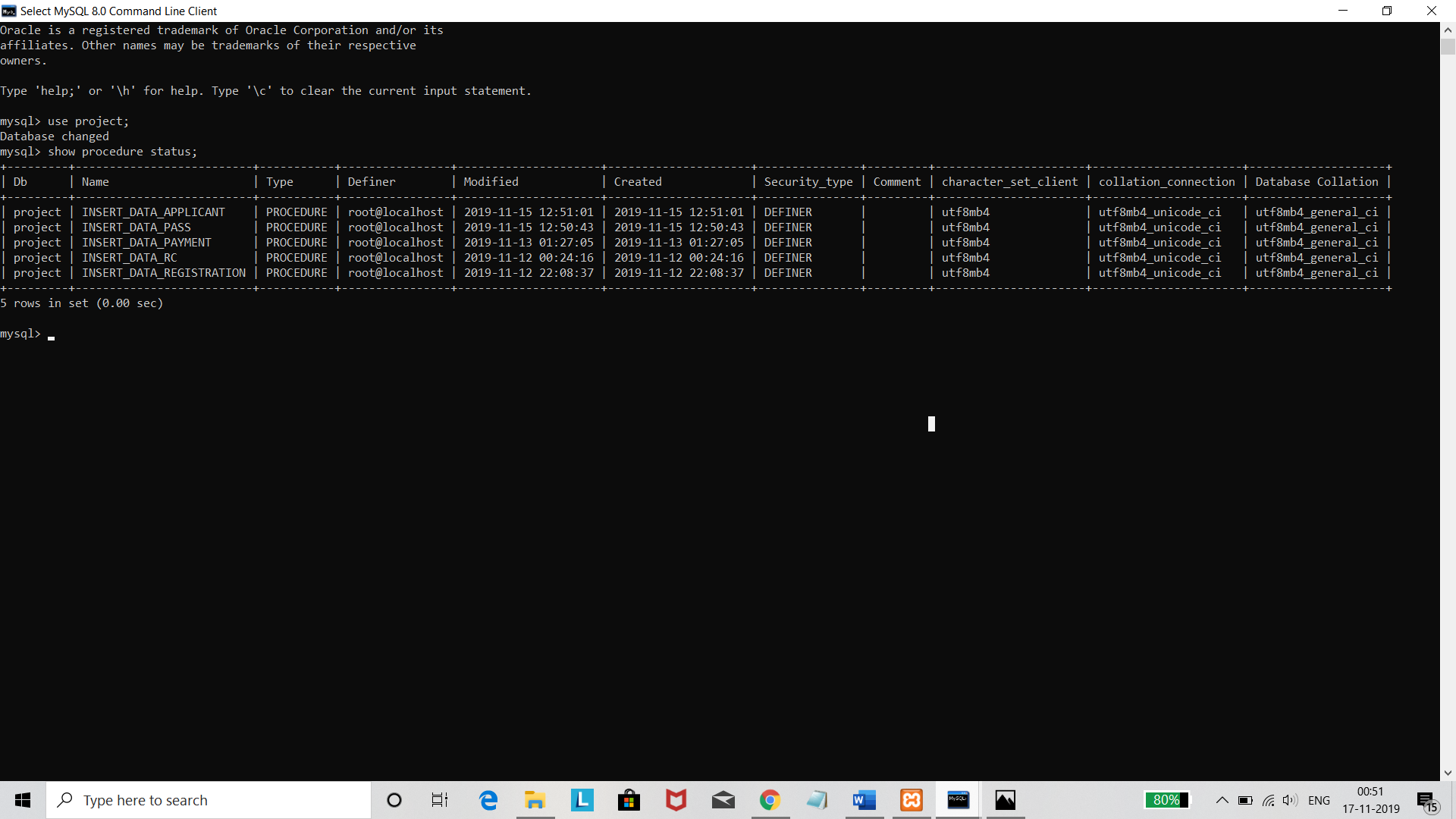
| project | INSERT\_DATA\_PASS | PROCEDURE | root@localhost | 2019-11-15 12:50:43 | 2019-11-15 12:50:43 | DEFINER | | utf8mb4 | utf8mb4\_unicode\_ci | utf8mb4\_general\_ci |

| project | INSERT\_DATA\_PAYMENT | PROCEDURE | root@localhost | 2019-11-13 01:27:05 | 2019-11-13 01:27:05 | DEFINER | | utf8mb4 | utf8mb4\_unicode\_ci | utf8mb4\_general\_ci |

| project | INSERT\_DATA\_RC | PROCEDURE | root@localhost | 2019-11-12 00:24:16 | 2019-11-12 00:24:16 | DEFINER | | utf8mb4 | utf8mb4\_unicode\_ci | utf8mb4\_general\_ci |

| project | INSERT\_DATA\_REGISTRATION | PROCEDURE | root@localhost | 2019-11-12 22:08:37 | 2019-11-12 22:08:37 | DEFINER | | utf8mb4 | utf8mb4\_unicode\_ci | utf8mb4\_general\_ci |

+---------+--------------------------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+



1. INSERT\_DATA\_APPLICANT: INSERT INTO applicant(ID, A, B, C, D, E) VALUES(id, a, b, c, d, e)
2. INSERT\_DATA\_PASS: REPLACE INTO applicant (ID, A, B, C, D, E) VALUES (id, a, b, c, d, e)
3. INSERT\_DATA\_PAYMENT: UPDATE rules\_contravened SET PAID\_OR\_NOT = paid\_or\_not where ID = id AND VID = vid
4. INSERT\_DATA\_RC: INSERT INTO rules\_contravened(ID, VID, LID, RULE\_NUMBER, RDATE) VALUES(id, vid, lid, rule\_number, rdate)
5. INSERT\_DATA\_REGISTRATION: INSERT INTO registration(VID, ID, PURPOSE) VALUES (id, vid, purpose)

TECHNOLOGY USED:

1. Front End: Web pages, Simple user friendly COMPLETELY RESPONSIVE UI created using HTML, CSS, JAVASCRIPT, BOOTSTRAP, AJAX.
2. Back End: Using mysql database, phpmyadmin, xampp localhost server.

