NYU – TANDON SCHOOL OF ENGINEERING
CS-GY 6083 - B, SPRING 2020
Principles of Database Systems

Project Part 2

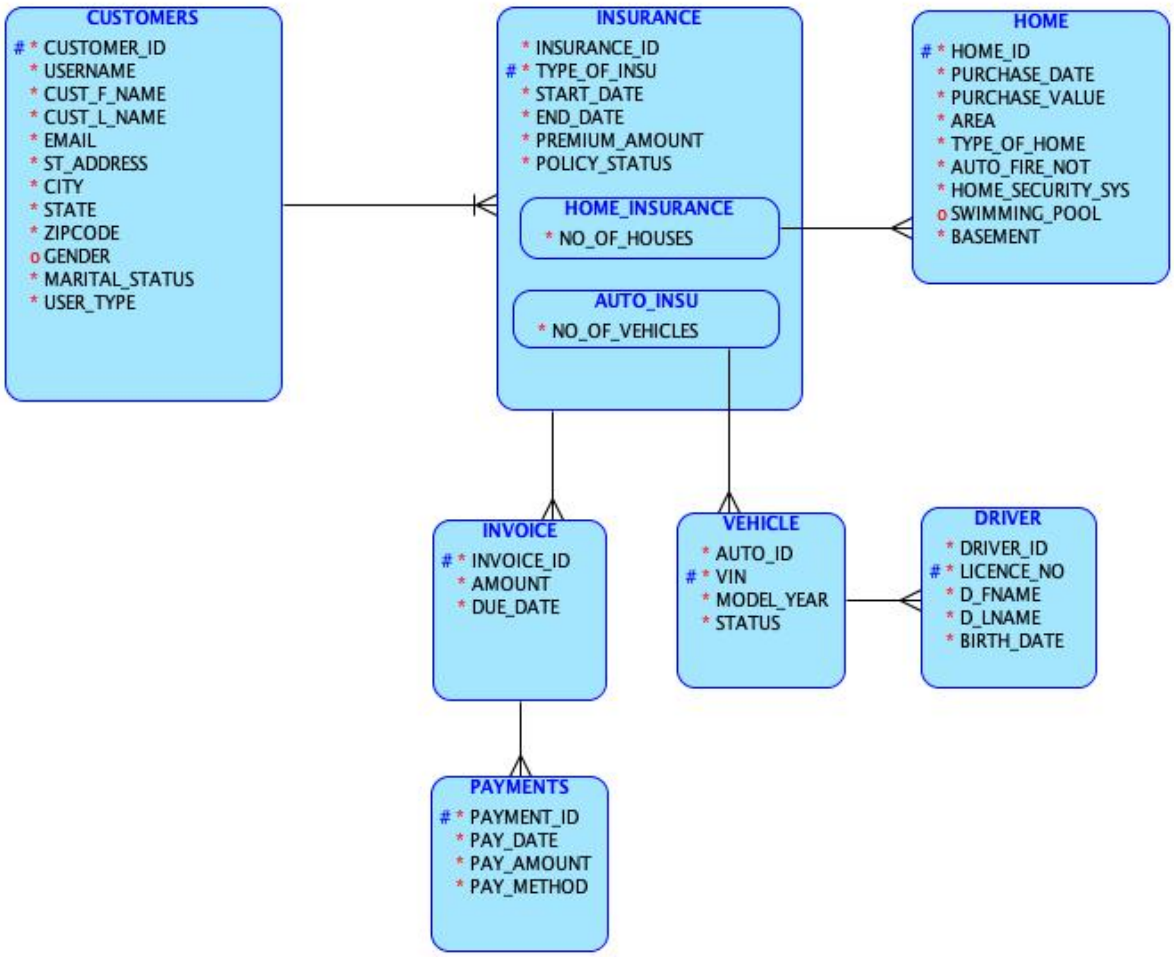
GROUP MEMBERS:

NAME : HARSHA PATTAPUSETTI
NET ID : vsp282

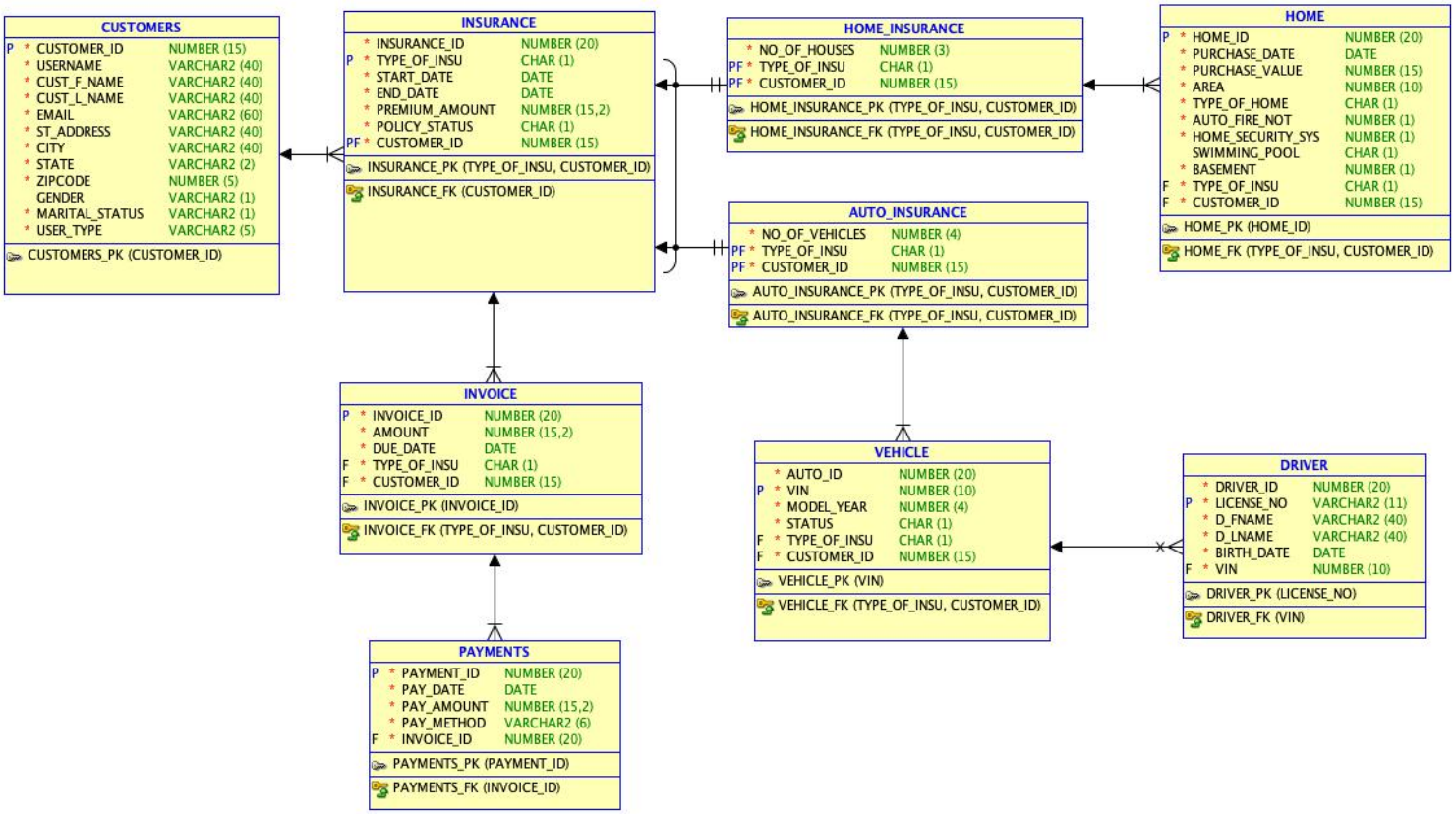
NAME : BHAGYASHRI SHETTI
NET ID : bss453

SUBMISSION DATE : 05/09/2020

LOGICAL MODEL:



RELATIONAL MODEL:



DDL CODE:

```
CREATE TABLE auto_insurance (
    no_of_vehicles  NUMBER(4) NOT NULL,
    type_of_insu    CHAR(1) NOT NULL,
    customer_id     NUMBER(15) NOT NULL
);

ALTER TABLE auto_insurance ADD CONSTRAINT auto_insurance_pk PRIMARY
KEY ( type_of_insu, customer_id );
```

```
CREATE TABLE customers (
   customer_id     NUMBER(15) NOT NULL,
   username        VARCHAR2(40) NOT NULL,
   cust_f_name     VARCHAR2(40) NOT NULL,
   cust_l_name     VARCHAR2(40) NOT NULL,
   email           VARCHAR2(60) NOT NULL,
   st_address      VARCHAR2(40) NOT NULL,
   city            VARCHAR2(40) NOT NULL,
   state           VARCHAR2(2) NOT NULL,
   zipcode         NUMBER(5) NOT NULL,
   gender          VARCHAR2(1),
   marital_status  VARCHAR2(1) NOT NULL,
   user_type       VARCHAR2(5) DEFAULT ON NULL user NOT NULL
);

ALTER TABLE customers ADD CONSTRAINT customers_pk PRIMARY KEY
( customer_id );

CREATE TABLE driver (
   driver_id   NUMBER(20) NOT NULL,
   licence_no  VARCHAR2(11) NOT NULL,
   d_fname     VARCHAR2(40) NOT NULL,
   d_lname     VARCHAR2(40) NOT NULL,
   birth_date  DATE NOT NULL,
   vin         NUMBER(10) NOT NULL
);

ALTER TABLE driver ADD CONSTRAINT driver_pk PRIMARY KEY ( licence_no );

CREATE TABLE home (
   home_id           NUMBER(20) NOT NULL,
   purchase_date     DATE NOT NULL,
   purchase_value    NUMBER(15) NOT NULL,
   area              NUMBER(10) NOT NULL,
   type_of_home      CHAR(1) NOT NULL,
   auto_fire_not     NUMBER(1) NOT NULL,
   home_security_sys NUMBER(1) NOT NULL,
   swimming_pool     CHAR(1),
   basement          NUMBER(1) NOT NULL,
   type_of_insu      CHAR(1) NOT NULL,
   customer_id       NUMBER(15) NOT NULL
);
```

```
ALTER TABLE home ADD CONSTRAINT home_pk PRIMARY KEY ( home_id );

CREATE TABLE home_insurance (
    no_of_houses  NUMBER(3) NOT NULL,
    type_of_insu  CHAR(1) NOT NULL,
    customer_id   NUMBER(15) NOT NULL
);

ALTER TABLE home_insurance ADD CONSTRAINT home_insurance_pk PRIMARY
KEY ( type_of_insu, customer_id );

CREATE TABLE insurance (
    insurance_id    NUMBER(20) NOT NULL,
    type_of_insu    CHAR(1) NOT NULL,
    start_date      DATE NOT NULL,
    end_date        DATE NOT NULL,
    premium_amount  NUMBER(15, 2) NOT NULL,
    policy_status   CHAR(1) NOT NULL,
    customer_id     NUMBER(15) NOT NULL
);

ALTER TABLE insurance
    ADD CONSTRAINT ch_inh_insurance CHECK ( type_of_insu IN (
        'A',
        'H'
    ) );

ALTER TABLE insurance ADD CONSTRAINT insurance_pk PRIMARY KEY
( type_of_insu, customer_id );

CREATE TABLE invoice (
    invoice_id    NUMBER(20) NOT NULL,
    amount        NUMBER(15, 2) NOT NULL,
    due_date      DATE NOT NULL,
    type_of_insu  CHAR(1) NOT NULL,
    customer_id   NUMBER(15) NOT NULL
);

ALTER TABLE invoice ADD CONSTRAINT invoice_pk PRIMARY KEY
( invoice_id );
```

```
CREATE TABLE payments (
   payment_id  NUMBER(20) NOT NULL,
   pay_date    DATE NOT NULL,
   pay_amount  FLOAT(15) NOT NULL,
   pay_method  VARCHAR2(6) NOT NULL,
   invoice_id  VARCHAR2(11) NOT NULL
);

ALTER TABLE payments ADD CONSTRAINT payments_pk PRIMARY KEY
( payment_id );

CREATE TABLE vehicle (
   auto_id       NUMBER(20) NOT NULL,
   vin           NUMBER(10) NOT NULL,
   model_year    NUMBER(4) NOT NULL,
   status        CHAR(1) NOT NULL,
   type_of_insu  CHAR(1) NOT NULL,
   customer_id   NUMBER(15) NOT NULL
);

ALTER TABLE vehicle ADD CONSTRAINT vehicle_pk PRIMARY KEY ( vin );

ALTER TABLE auto_insurance
   ADD CONSTRAINT auto_insurance_fk FOREIGN KEY ( type_of_insu,
                                                  customer_id )
      REFERENCES insurance ( type_of_insu, customer_id );

ALTER TABLE driver
   ADD CONSTRAINT driver_fk FOREIGN KEY ( vin )
      REFERENCES vehicle ( vin )
         ON DELETE CASCADE;

ALTER TABLE home
   ADD CONSTRAINT home_fk FOREIGN KEY ( type_of_insu, customer_id )
      REFERENCES home_insurance ( type_of_insu, customer_id );

ALTER TABLE home_insurance
   ADD CONSTRAINT home_insurance_fk FOREIGN KEY ( type_of_insu,
                                                  customer_id )
      REFERENCES insurance ( type_of_insu, customer_id );
```

```
ALTER TABLE insurance
   ADD CONSTRAINT insurance_fk FOREIGN KEY ( customer_id )
      REFERENCES customers ( customer_id );

ALTER TABLE invoice
   ADD CONSTRAINT invoice_fk FOREIGN KEY ( type_of_insu, customer_id )
      REFERENCES insurance ( type_of_insu, customer_id );

ALTER TABLE payments
   ADD CONSTRAINT payments_fk FOREIGN KEY ( invoice_id )
      REFERENCES invoice ( invoice_id );

ALTER TABLE vehicle
   ADD CONSTRAINT vehicle_fk FOREIGN KEY ( type_of_insu, customer_id )
      REFERENCES auto_insurance ( type_of_insu, customer_id );

CREATE OR REPLACE TRIGGER arc_fkarc_1_home_insurance BEFORE
   INSERT OR UPDATE OF type_of_insu, customer_id ON home_insurance
   FOR EACH ROW
DECLARE
   d CHAR(1);
BEGIN
   SELECT
      a.type_of_insu
   INTO d
   FROM
      insurance a
   WHERE
         a.type_of_insu = :new.type_of_insu
      AND a.customer_id = :new.customer_id;

   IF ( d IS NULL OR d <> 'H' ) THEN
                  raise_application_error(-20223, 'FK HOME_INSURANCE_FK in Table
HOME_INSURANCE violates Arc constraint on Table INSURANCE - discriminator
column TYPE_OF_INSU doesn"t have value "H"');
   END IF;

EXCEPTION
   WHEN no_data_found THEN
      NULL;
   WHEN OTHERS THEN
      RAISE;
```

```
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_1_auto_insurance BEFORE
   INSERT OR UPDATE OF type_of_insu, customer_id ON auto_insurance
   FOR EACH ROW
DECLARE
   d CHAR(1);
BEGIN
   SELECT
      a.type_of_insu
   INTO d
   FROM
      insurance a
   WHERE
         a.type_of_insu = :new.type_of_insu
      AND a.customer_id = :new.customer_id;

   IF ( d IS NULL OR d <> 'A' ) THEN
                  raise_application_error(-20223, 'FK AUTO_INSURANCE_FK in Table
AUTO_INSURANCE violates Arc constraint on Table INSURANCE - discriminator
column TYPE_OF_INSU doesn"t have value "A"');
   END IF;

EXCEPTION
   WHEN no_data_found THEN
      NULL;
   WHEN OTHERS THEN
      RAISE;
END;
/




--Check Constraints

ALTER TABLE customers
ADD CONSTRAINT c_cust_gen check (gender IN ('M','F'));

ALTER TABLE customers
ADD CONSTRAINT c_cust_marstat check (MARITAL_STATUS IN ('S','M','W'));
```

```
ALTER TABLE insurance
ADD CONSTRAINT c_insu_pol_stat check (policy_status in ('C','P'));

ALTER TABLE home
ADD CONSTRAINT c_home_typ_hom check (type_of_home in ('S','M','C','T'));

ALTER TABLE home
ADD CONSTRAINT c_home_auto_fir_n check (auto_fire_not between 0 and 1);

ALTER TABLE home
ADD CONSTRAINT c_home_sec_sys check (home_security_sys between 0 and 1);

ALTER TABLE home
ADD CONSTRAINT c_home_swim_pool check (swimming_pool in ('U','O','I','M'));

ALTER TABLE home
ADD CONSTRAINT c_home_basement check (basement between 0 and 1);

ALTER TABLE payments
ADD CONSTRAINT c_payments_method check (pay_method in
('PAYPAL','CREDIT','DEBIT','CHECK'));

ALTER TABLE vehicle
ADD CONSTRAINT c_vehicle_status check (status in ('L','O','F'));


ALTER TABLE customers
ADD CONSTRAINT c_cust_gen_up check (gender = upper(gender));

ALTER TABLE customers
ADD CONSTRAINT c_cust_marstat_up check (MARITAL_STATUS =
upper(MARITAL_STATUS));

ALTER TABLE insurance
ADD CONSTRAINT c_insu_typ_ins_up check (type_of_insu = upper(type_of_insu));

ALTER TABLE insurance
ADD CONSTRAINT c_insu_pol_stat_up check (policy_status = upper(policy_status));

ALTER TABLE home
ADD CONSTRAINT c_home_typ_hom_up check (type_of_home  =
upper(type_of_home));
```

```
ALTER TABLE home
ADD CONSTRAINT c_home_swim_pool_up check (swimming_pool =
upper(swimming_pool));

ALTER TABLE payments
ADD CONSTRAINT c_payments_method_up check (pay_method =
upper(pay_method));

ALTER TABLE vehicle
ADD CONSTRAINT c_vehicle_status_up check (status = upper(status));

ALTER TABLE vehicle
ADD CONSTRAINT c_vehicle_vin check (vin between 1000000000 and 9999999999);

ALTER TABLE customers
ADD CONSTRAINT c_cust_state check (length(state) = 2);

ALTER TABLE customers
ADD CONSTRAINT c_cust_fname check (cust_f_name = upper(cust_f_name));

ALTER TABLE customers
ADD CONSTRAINT c_cust_lname check (cust_l_name = upper(cust_l_name));

ALTER TABLE customers
ADD CONSTRAINT c_cust_city check (city= upper(city));

ALTER TABLE customers
ADD CONSTRAINT c_cust_zipcode check (zipcode between 10000 and 99999);

ALTER TABLE insurance
ADD CONSTRAINT c_insu_insuid check (length(insurance_id) = 7);

ALTER TABLE home
ADD CONSTRAINT c_home_home_id check (length(home_id) = 6);

ALTER TABLE invoice
ADD CONSTRAINT c_invoice_invoice_id check (length(invoice_id) = 7);

ALTER TABLE payments
ADD CONSTRAINT c_payments_pay_id check (length(payment_id) = 9);
```

```
ALTER TABLE driver
ADD CONSTRAINT c_driver_license_no check (length(license_no) = 10);

ALTER TABLE vehicle
ADD CONSTRAINT c_vehicle_model_year check (model_year between 1975 and
2020);

ALTER TABLE driver
ADD CONSTRAINT c_driver_d_fname check (d_fname = upper(d_fname));

ALTER TABLE driver
ADD CONSTRAINT c_driver_d_lname check (d_lname = upper(d_lname));




--Auto Increment

CREATE SEQUENCE customers_customer_id_seq START WITH 80000000 ORDER;

CREATE OR REPLACE TRIGGER customers_customer_id_trg BEFORE
   INSERT ON customers
   FOR EACH ROW
   WHEN ( new.customer_id IS NULL )
BEGIN
   :new.customer_id := customers_customer_id_seq.nextval;
END;
/

CREATE SEQUENCE driver_driver_id_seq START WITH 4000 ORDER;

CREATE OR REPLACE TRIGGER driver_driver_id_trg BEFORE
   INSERT ON driver
   FOR EACH ROW
   WHEN ( new.driver_id IS NULL )
BEGIN
   :new.driver_id := driver_driver_id_seq.nextval;
END;
/
```

```
CREATE SEQUENCE home_home_id_seq START WITH 2000 ORDER;

CREATE OR REPLACE TRIGGER home_home_id_trg BEFORE
   INSERT ON home
   FOR EACH ROW
   WHEN ( new.home_id IS NULL )
BEGIN
   :new.home_id := home_home_id_seq.nextval;
END;
/

CREATE SEQUENCE insurance_insurance_id_seq START WITH 900000000 ORDER;

CREATE OR REPLACE TRIGGER insurance_insurance_id_trg BEFORE
   INSERT ON insurance
   FOR EACH ROW
   WHEN ( new.insurance_id IS NULL )
BEGIN
   :new.insurance_id := insurance_insurance_id_seq.nextval;
END;
/

CREATE SEQUENCE invoice_invoice_id_seq START WITH 50000 ORDER;

CREATE OR REPLACE TRIGGER invoice_invoice_id_trg BEFORE
   INSERT ON invoice
   FOR EACH ROW
   WHEN ( new.invoice_id IS NULL )
BEGIN
   :new.invoice_id := invoice_invoice_id_seq.nextval;
END;
/

CREATE SEQUENCE payments_payment_id_seq START WITH 600000 ORDER;

CREATE OR REPLACE TRIGGER payments_payment_id_trg BEFORE
   INSERT ON payments
   FOR EACH ROW
   WHEN ( new.payment_id IS NULL )
BEGIN
   :new.payment_id := payments_payment_id_seq.nextval;
END;
/
```

```
CREATE SEQUENCE vehicle_auto_id_seq START WITH 3000 ORDER;

CREATE OR REPLACE TRIGGER vehicle_auto_id_trg BEFORE
   INSERT ON vehicle
   FOR EACH ROW
   WHEN ( new.auto_id IS NULL )
BEGIN
   :new.auto_id := vehicle_auto_id_seq.nextval;
END;
/
```

Indexes:

```
CREATE UNIQUE INDEX customers_us_unique_i ON customers(username);
CREATE UNIQUE INDEX customers_em_unique_i ON customers(email);
CREATE BITMAP INDEX customers_gen_bitmap_i ON customers(gender);
CREATE BITMAP INDEX customers_ms_bitmap_i ON customers(marital_status);
CREATE BITMAP INDEX insurance_bitmap_i ON insurance(policy_status);
CREATE BITMAP INDEX home_bitmap_i ON home(type_of_home);
CREATE BITMAP INDEX home_bitmap_i ON home(auto_fire_not);
CREATE BITMAP INDEX home_bitmap_i ON home(home_security_sys);
CREATE BITMAP INDEX home_bitmap_i ON home(swimming_pool);
CREATE BITMAP INDEX home_bitmap_i ON home(basement);
CREATE UNIQUE INDEX vehicle_unique_i ON vehicle(vin);
CREATE BITMAP INDEX vehicle_bitmap_i ON vehicle(status);
CREATE UNIQUE INDEX driver_unique_i ON driver(license_no);
CREATE BITMAP INDEX payments_bitmap_i ON payments(pay_method);
```

Note: We have implemented the website online using **MySql**, thus all the indexes (other than Bitmap Index, which also include full text index for important attributes like username etc) and auto increments (with starting values) are implemented directly through the design.

SUMMARY OF DEVELOPMENT ENVIRONMENTS USED:

- In this project, we have developed a web-based interface application for the WDS insurance management company. In this website, users can register, login, buy an insurance policy, and edit their existing insurance.

- The code for this web development is written in PHP, a server-side scripting language that is called by the XAMPP web server and is connected to a relational database. PHP is a widely-used open-source language that is specifically used for web application development and can be embedded within HTML. PHP can run on Windows, Linux, and UNIX servers.

- We use PHP because it can interact with many different database systems including Oracle and MySQL. The PHP script can access the MySQL database to retrieve all information about the web page. The procedure of setting up the MySQL database varies according to the localhost. Every database would require a user name and password in order to access the database. Database administration can be done using PHP scripts or using a program like PHPMyAdmin. We have created the WDS database and tables for storing the website information using PHPMyAdmin.

- We have written the code in NotePad++ which is a text editor for PHP on windows. We embed HTML code inside PHP tags to format them and make them user-friendly. The style sheets are generated using CSS (Cascading Style Sheets) in PHP for describing the presentation of the website.
- Both PHP and MySQL are compatible with the XAMPP server which is also free to license. XAMPP is an open-source cross-platform web server solution stack consisting mainly of the Apache HTTP server, MariaDB database, and interpreters for scripts written in PHP programming language. Due to all these languages being free it is cheap and easy to set up and create a website using PHP.

SUMMARY OF FEATURES USED:

- We have created an attribute user_type in the customers table to differentiate between user, admin and super admin. Both have different authorization on data.
- Both the user and admin login through the same login page. The admin will be redirected to the page consisting of all the user data available in the database. Depending on the position of the employee in the company the privileges are granted. The privileges are granted by the super user, who has a unique user name with all the privileges.
- We have used Unique (BTREE) and Bitmap indexes to increase the query performance.
- We use the MD5() function to encrypt the password before saving it in the database.
- We use a special PHP function htmlspecialchars() to avoid cross-site scripting.
- We use mysqli_real_escape_string() function to guard against SQL injection.
- We have used session and cookies to support multiple users logging in at the same time.
- We hosted our website online by using online file manager. The link to this is: http://wdsinsurance.byethost22.com/WDS/public/data/

LEARNING OUTCOME:

This project has introduced us to server-based web scripting and dynamic web application development.

- We can apply a structured approach to identifying the needs, interests, and functionality of a website.
- We have learned to use the PHP programming language and MySQL database administration for web development.
- We can set up and configure PHP, MySQL, and XAMPP web server development environment.
- We understand PHP language data types, built-in and user-defined functions
- We can create PHP pages dynamic based upon user interaction, interacting with HTML forms, CSS style sheets, and store and retrieve information from the database.
- We can select, insert, update, and delete data from database using SQL language.