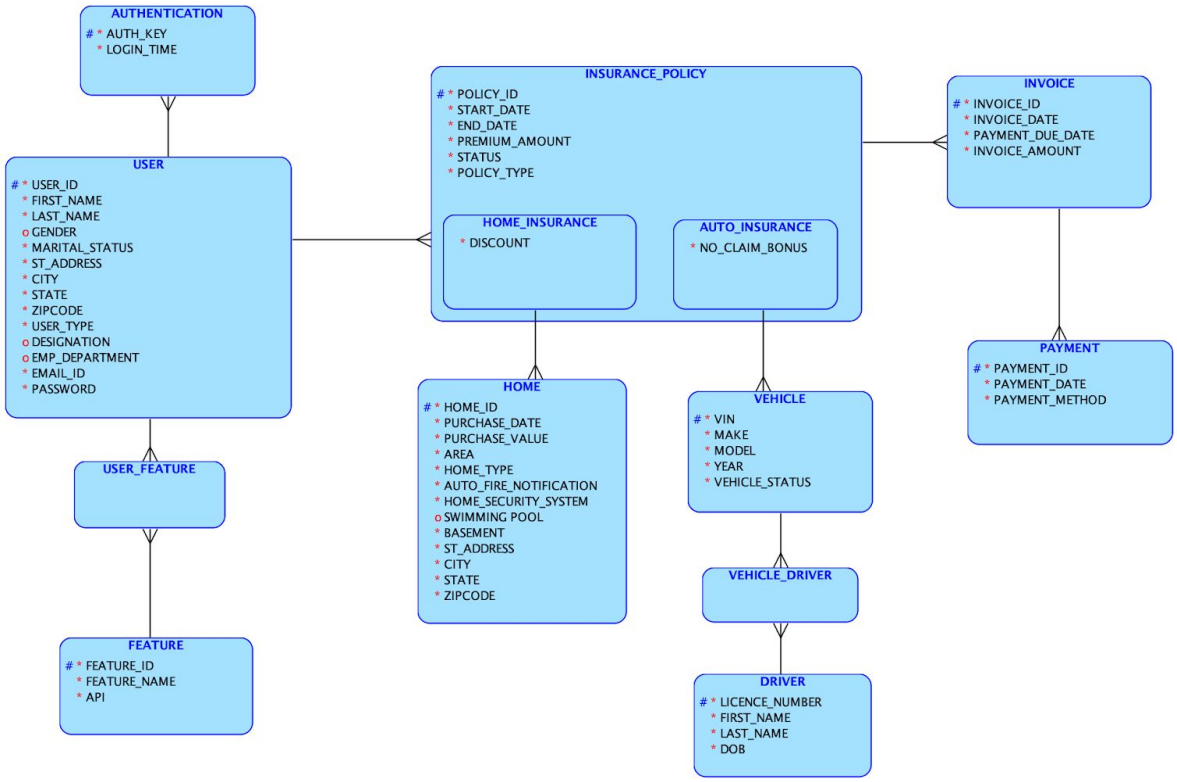
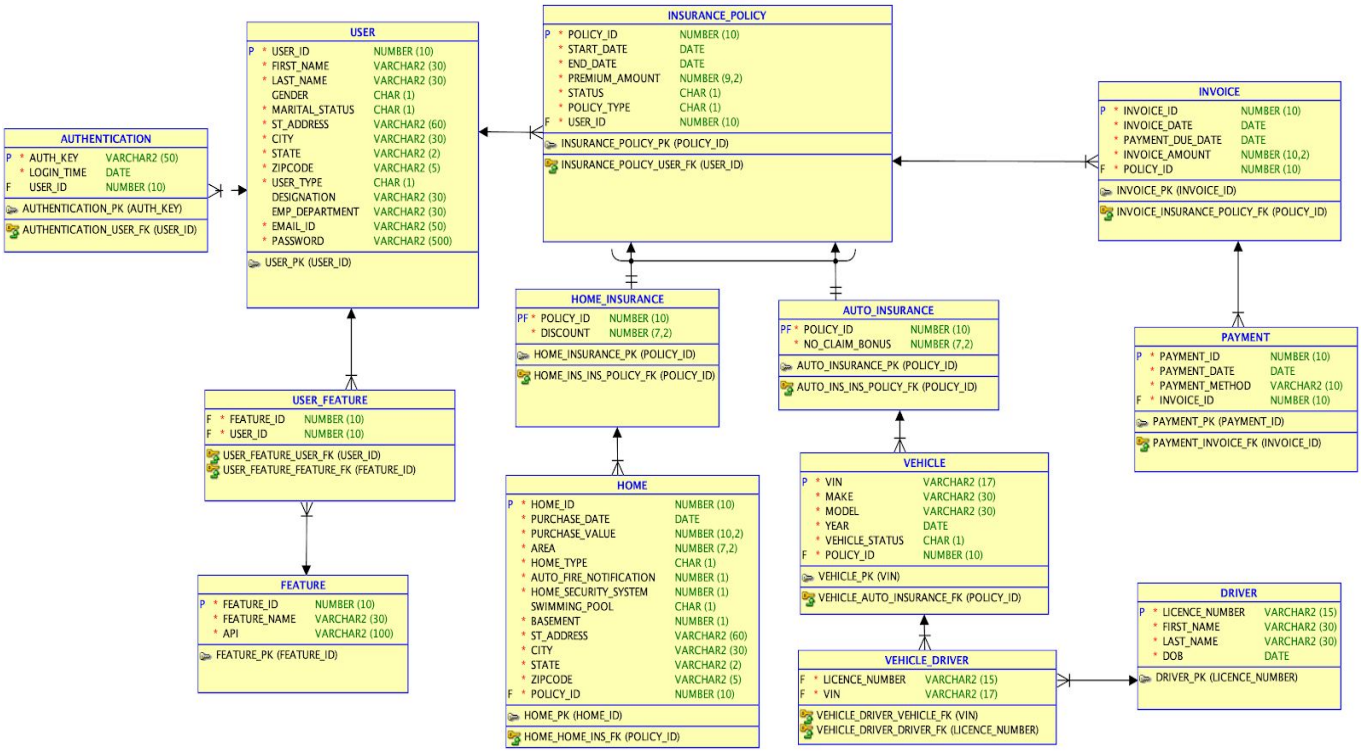
Table of Content

Logical Model



Relational Model



DDL Code In Oracle

```
-- Generated by Oracle SQL Developer Data Modeler 19.2.0.182.1216
--   at:        2020-05-08 18:21:28 EDT
--   site:      Oracle Database 12cR2
--   type:      Oracle Database 12cR2



CREATE TABLE authentication (
    auth_key     VARCHAR2(50) NOT NULL,
    login_time   DATE NOT NULL,
    user_id      NUMBER(10)
);

COMMENT ON COLUMN authentication.auth_key IS
    'UNIQUE AUTHENTICATION KEY FOR EVERY USER';

COMMENT ON COLUMN authentication.login_time IS
    'LOGIN TIME OF USER';

COMMENT ON COLUMN authentication.user_id IS
    'USER ID';

ALTER TABLE authentication ADD CONSTRAINT authentication_pk PRIMARY KEY ( auth_key
);

CREATE TABLE auto_insurance (
    policy_id        NUMBER(10) NOT NULL,
    no_claim_bonus   NUMBER(7, 2) NOT NULL
);

COMMENT ON COLUMN auto_insurance.policy_id IS
    'UNIQUE ID OF INSURANCE POLICY';

COMMENT ON COLUMN auto_insurance.no_claim_bonus IS
    'NO CLAIM BONUS ON AUTO INSURANCE POLICY';

ALTER TABLE auto_insurance ADD CONSTRAINT auto_insurance_pk PRIMARY KEY (
policy_id );

CREATE TABLE driver (
```

```
    licence_number   VARCHAR2(15) NOT NULL,
    first_name       VARCHAR2(30) NOT NULL,
    last_name        VARCHAR2(30) NOT NULL,
    dob              DATE NOT NULL
);

COMMENT ON COLUMN driver.licence_number IS
    'LICENCE NUMBER OF DRIVER';

COMMENT ON COLUMN driver.first_name IS
    'FIRST NAME OF DRIVER';

COMMENT ON COLUMN driver.last_name IS
    'LAST NAME OF DRIVER';

COMMENT ON COLUMN driver.dob IS
    'DATE BIRTH OF DRIVER';

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

CREATE TABLE feature (
    feature_id     NUMBER(10) NOT NULL,
    feature_name   VARCHAR2(30) NOT NULL,
    api            VARCHAR2(100) NOT NULL
);

COMMENT ON COLUMN feature.feature_id IS
    'UINIQUE ID OF FEATURE';

COMMENT ON COLUMN feature.feature_name IS
    'NAME OF FEATURE';

COMMENT ON COLUMN feature.api IS
    'API FOR FEATURE';

ALTER TABLE feature ADD CONSTRAINT feature_pk PRIMARY KEY ( feature_id );

CREATE TABLE home (
    home_id                NUMBER(10) NOT NULL,
    purchase_date            DATE NOT NULL,
    purchase_value           NUMBER(10, 2) NOT NULL,
    area                   NUMBER(7, 2) NOT NULL,
    home_type                CHAR(1) NOT NULL,
```

```
    auto_fire_notification   NUMBER(1) NOT NULL,
    home_security_system     NUMBER(1) NOT NULL,
    swimming_pool            CHAR(1),
    basement                 NUMBER(1) NOT NULL,
    st_address               VARCHAR2(60) NOT NULL,
    city                    VARCHAR2(30) NOT NULL,
    state                    VARCHAR2(2) NOT NULL,
    zipcode                  VARCHAR2(5) NOT NULL,
    policy_id                NUMBER(10) NOT NULL
);

COMMENT ON COLUMN home.home_id IS
    'UINIQUE ID OF HOME';

COMMENT ON COLUMN home.purchase_date IS
    'HOME PURCHASE DATE';

COMMENT ON COLUMN home.purchase_value IS
    'PURCHASE VALUE OF HOME';

COMMENT ON COLUMN home.area IS
    'AREA OF HOME';

COMMENT ON COLUMN home.home_type IS
    'TYPE OF HOME';

COMMENT ON COLUMN home.auto_fire_notification IS
    'AUTO FIRE NOTIFICATION OF HOME';

COMMENT ON COLUMN home.home_security_system IS
    'SECURITY SYSTEM OF HOME';

COMMENT ON COLUMN home.swimming_pool IS
    'SWIMMING POOL WITH HOME';

COMMENT ON COLUMN home.basement IS
    'BASEMENT IN HOME';

COMMENT ON COLUMN home.st_address IS
    'STREET ADDRESS OF HOME';

COMMENT ON COLUMN home.city IS
    'CITY';
```

```
COMMENT ON COLUMN home.state IS
   'STATE';

COMMENT ON COLUMN home.zipcode IS
   'ZIPCODE';

COMMENT ON COLUMN home.policy_id IS
   'UNIQUE ID OF INSURANCE POLICY';

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

CREATE TABLE home_insurance (
   policy_id   NUMBER(10) NOT NULL,
   discount    NUMBER(7, 2) NOT NULL
);

COMMENT ON COLUMN home_insurance.policy_id IS
   'UNIQUE ID OF INSURANCE POLICY';

COMMENT ON COLUMN home_insurance.discount IS
   'DISCOUNT ON HOME INSURANCE POLICY';

ALTER TABLE home_insurance ADD CONSTRAINT home_insurance_pk PRIMARY KEY (
policy_id );

CREATE TABLE insurance_policy (
   policy_id        NUMBER(10) NOT NULL,
   start_date       DATE NOT NULL,
   end_date         DATE NOT NULL,
   premium_amount   NUMBER(9, 2) NOT NULL,
   status           CHAR(1) NOT NULL,
   policy_type      CHAR(1) NOT NULL,
   user_id          NUMBER(10) NOT NULL
);

ALTER TABLE insurance_policy
   ADD CONSTRAINT ch_inh_insurance_policy CHECK ( policy_type IN (
      'A',
      'H'
   ) );

COMMENT ON COLUMN insurance_policy.policy_id IS
```

```
   'UNIQUE ID OF INSURANCE POLICY';

COMMENT ON COLUMN insurance_policy.start_date IS
   'START DATE OF INSURANCE POLICY';

COMMENT ON COLUMN insurance_policy.end_date IS
   'END DATE OF INSURANCE POLICY';

COMMENT ON COLUMN insurance_policy.premium_amount IS
   'PREMIUM AMOUNT OF INSURANCE POLICY';

COMMENT ON COLUMN insurance_policy.status IS
   'STATUS OF POLICY';

COMMENT ON COLUMN insurance_policy.policy_type IS
   'TYPE OF POLICY';

COMMENT ON COLUMN insurance_policy.user_id IS
   'USER ID';

ALTER TABLE insurance_policy ADD CONSTRAINT insurance_policy_pk PRIMARY KEY (
policy_id );

CREATE TABLE invoice (
   invoice_id         NUMBER(10) NOT NULL,
   invoice_date       DATE NOT NULL,
   payment_due_date   DATE NOT NULL,
   invoice_amount     NUMBER(10, 2) NOT NULL,
   policy_id          NUMBER(10) NOT NULL
);

COMMENT ON COLUMN invoice.invoice_id IS
   'UNIQUE ID OF INVOICE';

COMMENT ON COLUMN invoice.invoice_date IS
   'INVOICE DATE';

COMMENT ON COLUMN invoice.payment_due_date IS
   'PAYMENT DUE DATE';

COMMENT ON COLUMN invoice.invoice_amount IS
   'INVOICE AMOUNT';
```

```
COMMENT ON COLUMN invoice.policy_id IS
   'UNIQUE ID OF INSURANCE POLICY';

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

CREATE TABLE payment (
   payment_id       NUMBER(10) NOT NULL,
   payment_date     DATE NOT NULL,
   payment_method   VARCHAR2(10) NOT NULL,
   invoice_id       NUMBER(10) NOT NULL
);

COMMENT ON COLUMN payment.payment_id IS
   'UNIQUE PAYMENT ID';

COMMENT ON COLUMN payment.payment_date IS
   'PAYMENT DATE';

COMMENT ON COLUMN payment.payment_method IS
   'PAYMENT METHOD';

COMMENT ON COLUMN payment.invoice_id IS
   'UNIQUE INCOIVE ID';

ALTER TABLE payment ADD CONSTRAINT payment_pk PRIMARY KEY ( payment_id );

CREATE TABLE "USER" (
   user_id          NUMBER(10) NOT NULL,
   first_name       VARCHAR2(30) NOT NULL,
   last_name        VARCHAR2(30) NOT NULL,
   gender           CHAR(1),
   marital_status   CHAR(1) NOT NULL,
   st_address       VARCHAR2(60) NOT NULL,
   city             VARCHAR2(30) NOT NULL,
   state            VARCHAR2(2) NOT NULL,
   zipcode          VARCHAR2(5) NOT NULL,
   user_type        CHAR(1) NOT NULL,
   designation      VARCHAR2(30),
   emp_department   VARCHAR2(30),
   email_id         VARCHAR2(50) NOT NULL,
   password         VARCHAR2(500) NOT NULL
);
```

```
COMMENT ON COLUMN "USER".user_id IS
   'UNIQUE ID OF USER';

COMMENT ON COLUMN "USER".first_name IS
   'FIRST NAME OF USER';

COMMENT ON COLUMN "USER".last_name IS
   'LAST NAME OF USER';

COMMENT ON COLUMN "USER".gender IS
   'GENDER OF USER';

COMMENT ON COLUMN "USER".marital_status IS
   'MARITAL STATUS OF USER';

COMMENT ON COLUMN "USER".st_address IS
   'STREET ADDRESS OF USER';

COMMENT ON COLUMN "USER".city IS
   'CITY OF USER';

COMMENT ON COLUMN "USER".state IS
   'STATE OF USER';

COMMENT ON COLUMN "USER".zipcode IS
   'ZIPCODE OF USER';

COMMENT ON COLUMN "USER".user_type IS
   'TYPE OF USER';

COMMENT ON COLUMN "USER".designation IS
   'DESIGNATION OF EMPLOYE';

COMMENT ON COLUMN "USER".emp_department IS
   'DEPARTMENT OF EMPLOYEE';

COMMENT ON COLUMN "USER".email_id IS
   'EMAIL ID OF USER';

COMMENT ON COLUMN "USER".password IS
   'USER PASWORD';

ALTER TABLE "USER" ADD CONSTRAINT user_pk PRIMARY KEY ( user_id );
```

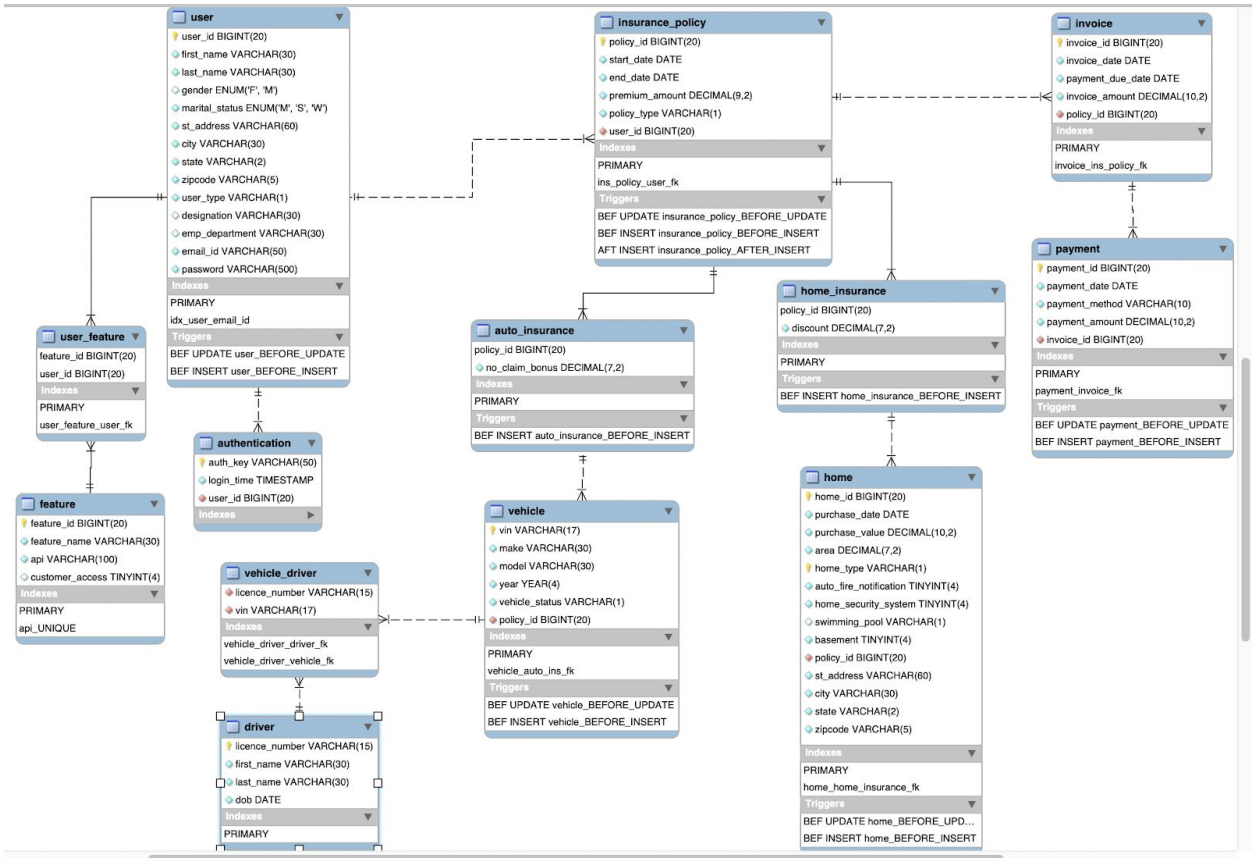
```
CREATE TABLE user_feature (
   feature_id   NUMBER(10) NOT NULL,
   user_id      NUMBER(10) NOT NULL
);

COMMENT ON COLUMN user_feature.feature_id IS
   'FEATURE ID';

COMMENT ON COLUMN user_feature.user_id IS
   'USER ID';

CREATE TABLE vehicle (
   vin              VARCHAR2(17) NOT NULL,
   make             VARCHAR2(30) NOT NULL,
   model            VARCHAR2(30) NOT NULL,
   year             DATE NOT NULL,
   vehicle_status   CHAR(1) NOT NULL,
   policy_id        NUMBER(10) NOT NULL
);

COMMENT ON COLUMN vehicle.vin IS
   'VIN OF VEHICLE';

COMMENT ON COLUMN vehicle.make IS
   'MAKE OF VEHICLE';

COMMENT ON COLUMN vehicle.model IS
   'MODEL OF VEHICLE';

COMMENT ON COLUMN vehicle.year IS
   'YEAR OF VEHICLE';

COMMENT ON COLUMN vehicle.vehicle_status IS
   'STATUS OF VEHICLE';

COMMENT ON COLUMN vehicle.policy_id IS
   'UNIQUE ID OF INSURANCE POLICY';

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

CREATE TABLE vehicle_driver (
   licence_number   VARCHAR2(15) NOT NULL,
```

```
    vin             VARCHAR2(17) NOT NULL
);

COMMENT ON COLUMN vehicle_driver.licence_number IS
    'UNIQUE LICENCE NUMBER';

COMMENT ON COLUMN vehicle_driver.vin IS
    'VIN';

ALTER TABLE authentication
    ADD CONSTRAINT authentication_user_fk FOREIGN KEY ( user_id )
        REFERENCES "USER" ( user_id );

ALTER TABLE auto_insurance
    ADD CONSTRAINT auto_ins_ins_policy_fk FOREIGN KEY ( policy_id )
        REFERENCES insurance_policy ( policy_id );

ALTER TABLE home
    ADD CONSTRAINT home_home_ins_fk FOREIGN KEY ( policy_id )
        REFERENCES home_insurance ( policy_id );

ALTER TABLE home_insurance
    ADD CONSTRAINT home_ins_ins_policy_fk FOREIGN KEY ( policy_id )
        REFERENCES insurance_policy ( policy_id );

ALTER TABLE insurance_policy
    ADD CONSTRAINT insurance_policy_user_fk FOREIGN KEY ( user_id )
        REFERENCES "USER" ( user_id );

ALTER TABLE invoice
    ADD CONSTRAINT invoice_insurance_policy_fk FOREIGN KEY ( policy_id )
        REFERENCES insurance_policy ( policy_id );

ALTER TABLE payment
    ADD CONSTRAINT payment_invoice_fk FOREIGN KEY ( invoice_id )
        REFERENCES invoice ( invoice_id );

ALTER TABLE user_feature
    ADD CONSTRAINT user_feature_feature_fk FOREIGN KEY ( feature_id )
        REFERENCES feature ( feature_id );

ALTER TABLE user_feature
    ADD CONSTRAINT user_feature_user_fk FOREIGN KEY ( user_id )
```

```
      REFERENCES "USER" ( user_id );

ALTER TABLE vehicle
   ADD CONSTRAINT vehicle_auto_insurance_fk FOREIGN KEY ( policy_id )
      REFERENCES auto_insurance ( policy_id );

ALTER TABLE vehicle_driver
   ADD CONSTRAINT vehicle_driver_driver_fk FOREIGN KEY ( licence_number )
      REFERENCES driver ( licence_number );

ALTER TABLE vehicle_driver
   ADD CONSTRAINT vehicle_driver_vehicle_fk FOREIGN KEY ( vin )
      REFERENCES vehicle ( vin );

CREATE OR REPLACE TRIGGER arc_fkarc_2_home_insurance BEFORE
   INSERT OR UPDATE OF policy_id ON home_insurance
   FOR EACH ROW
DECLARE
   d CHAR(1);
BEGIN
   SELECT
      a.policy_type
   INTO d
   FROM
      insurance_policy a
   WHERE
      a.policy_id = :new.policy_id;

   IF ( d IS NULL OR d <> 'H' ) THEN
      raise_application_error(-20223, 'FK HOME_INS_INS_POLICY_FK in Table
HOME_INSURANCE violates Arc constraint on Table INSURANCE_POLICY - discriminator
column POLICY_TYPE 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_2_auto_insurance BEFORE
   INSERT OR UPDATE OF policy_id ON auto_insurance
   FOR EACH ROW
DECLARE
   d CHAR(1);
BEGIN
   SELECT
      a.policy_type
   INTO d
   FROM
      insurance_policy a
   WHERE
      a.policy_id = :new.policy_id;

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

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

Relational Model in MySQL Workbench



DDL Code In MySQL

Create Table

```
-- -----------------------------------------------------
-- Table `wds`.`user`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`user` (
  `user_id` BIGINT(20) NOT NULL COMMENT 'UNIQUE ID OF USER',
  `first_name` VARCHAR(30) NOT NULL COMMENT 'FIRST NAME OF USER',
  `last_name` VARCHAR(30) NOT NULL COMMENT 'LAST NAME OF USER',
  `gender` ENUM('F', 'M') NULL DEFAULT NULL COMMENT 'GENDER OF USER',
  `marital_status` ENUM('M', 'S', 'W') NOT NULL COMMENT 'MARITAL STATUS OF USER',
  `st_address` VARCHAR(60) NOT NULL COMMENT 'STREET ADDRESS OF USER',
  `city` VARCHAR(30) NOT NULL COMMENT 'CITY OF USER',
  `state` VARCHAR(2) NOT NULL COMMENT 'STATE OF USER',
  `zipcode` VARCHAR(5) NOT NULL COMMENT 'ZIPCODE OF USER',
  `user_type` VARCHAR(1) NOT NULL COMMENT 'TYPE OF USER',
```

```
  `designation` VARCHAR(30) NULL DEFAULT NULL COMMENT 'DESIGNATION OF
EMPLOYE',
  `emp_department` VARCHAR(30) NULL DEFAULT NULL COMMENT 'DEPARTMENT OF
EMPLOYEE',
  `email_id` VARCHAR(50) NOT NULL,
  `password` VARCHAR(500) NOT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE INDEX `idx_user_email_id` (`email_id` ASC))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`authentication`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`authentication` (
  `auth_key` VARCHAR(50) NOT NULL,
  `login_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `user_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`auth_key`),
  UNIQUE INDEX `Auth_key_UNIQUE` (`auth_key` ASC),
  INDEX `user_id_idx` (`user_id` ASC)  ,
  CONSTRAINT `user_id`
    FOREIGN KEY (`user_id`)
    REFERENCES `wds`.`user` (`user_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`insurance_policy`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`insurance_policy` (
  `policy_id` BIGINT(20) NOT NULL COMMENT 'UNIQUE ID OF INSURANCE POLICY',
  `start_date` DATE NOT NULL COMMENT 'START DATE OF INSURANCE POLICY',
  `end_date` DATE NOT NULL COMMENT 'END DATE OF INSURANCE POLICY',
  `premium_amount` DECIMAL(9,2) NOT NULL COMMENT 'PREMIUM AMOUNT OF
INSURANCE POLICY',
  `policy_type` VARCHAR(1) NOT NULL COMMENT 'TYPE OF POLICY',
```

```
  `user_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`policy_id`),
  INDEX `ins_policy_user_fk` (`user_id` ASC) ,
  CONSTRAINT `ins_policy_user_fk`
    FOREIGN KEY (`user_id`)
    REFERENCES `wds`.`user` (`user_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`auto_insurance`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`auto_insurance` (
  `policy_id` BIGINT(20) NOT NULL COMMENT 'UNIQUE ID OF INSURANCE POLICY',
  `no_claim_bonus` DECIMAL(7,2) NOT NULL COMMENT 'NO CLAIM BONUS ON AUTO
INSURANCE POLICY',
  PRIMARY KEY (`policy_id`),
  CONSTRAINT `auto_ins_ins_policy_fk`
    FOREIGN KEY (`policy_id`)
    REFERENCES `wds`.`insurance_policy` (`policy_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`driver`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`driver` (
  `licence_number` VARCHAR(15) NOT NULL COMMENT 'LICENCE NUMBER OF DRIVER',
  `first_name` VARCHAR(30) NOT NULL COMMENT 'FIRST NAME OF DRIVER',
  `last_name` VARCHAR(30) NOT NULL COMMENT 'LAST NAME OF DRIVER',
  `dob` DATE NOT NULL COMMENT 'DATE BIRTH OF DRIVER',
  PRIMARY KEY (`licence_number`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----------------------------------------------------
-- Table `wds`.`feature`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`feature` (
  `feature_id` BIGINT(20) NOT NULL COMMENT 'UINIQUE ID OF FEATURE',
  `feature_name` VARCHAR(30) NOT NULL COMMENT 'NAME OF FEATURE',
  `api` VARCHAR(100) NOT NULL COMMENT 'API FOR FEATURE',
  `customer_access` TINYINT(4) NULL DEFAULT NULL,
  PRIMARY KEY (`feature_id`),
  UNIQUE INDEX `api_UNIQUE` (`api` ASC)  )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`home_insurance`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`home_insurance` (
  `policy_id` BIGINT(20) NOT NULL COMMENT 'UNIQUE ID OF INSURANCE POLICY',
  `discount` DECIMAL(7,2) NOT NULL COMMENT 'DISCOUNT ON HOME INSURANCE
POLICY',
  PRIMARY KEY (`policy_id`),
  CONSTRAINT `home_ins_ins_policy_fk`
    FOREIGN KEY (`policy_id`)
    REFERENCES `wds`.`insurance_policy` (`policy_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`home`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`home` (
  `home_id` BIGINT(20) NOT NULL COMMENT 'UINIQUE ID OF HOME',
  `purchase_date` DATE NOT NULL COMMENT 'HOME PURCHASE DATE',
  `purchase_value` DECIMAL(10,2) NOT NULL COMMENT 'PURCHASE VALUE OF HOME',
  `area` DECIMAL(7,2) NOT NULL COMMENT 'AREA OF HOME',
```

```
  `home_type` VARCHAR(1) NOT NULL COMMENT 'TYPE OF HOME',
  `auto_fire_notification` TINYINT(4) NOT NULL COMMENT 'AUTO FIRE NOTIFICATION OF
HOME',
  `home_security_system` TINYINT(4) NOT NULL COMMENT 'SECURITY SYSTEM OF
HOME',
  `swimming_pool` VARCHAR(1) NULL DEFAULT NULL COMMENT 'SWIMMING POOL WITH
HOME',
  `basement` TINYINT(4) NOT NULL COMMENT 'BASEMENT IN HOME',
  `policy_id` BIGINT(20) NOT NULL,
  `st_address` VARCHAR(60) NOT NULL,
  `city` VARCHAR(30) NOT NULL,
  `state` VARCHAR(2) NOT NULL,
  `zipcode` VARCHAR(5) NOT NULL,
  PRIMARY KEY (`home_id`, `home_type`),
  INDEX `home_home_insurance_fk` (`policy_id` ASC)  ,
  CONSTRAINT `home_home_insurance_fk`
    FOREIGN KEY (`policy_id`)
    REFERENCES `wds`.`home_insurance` (`policy_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`invoice`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`invoice` (
  `invoice_id` BIGINT(20) NOT NULL COMMENT 'UNIQUE ID OF INVOICE',
  `invoice_date` DATE NOT NULL COMMENT 'INVOICE DATE',
  `payment_due_date` DATE NOT NULL COMMENT 'PAYMENT DUE DATE',
  `invoice_amount` DECIMAL(10,2) NOT NULL COMMENT 'INVOICE AMOUNT',
  `policy_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`invoice_id`),
  INDEX `invoice_ins_policy_fk` (`policy_id` ASC)  ,
  CONSTRAINT `invoice_ins_policy_fk`
    FOREIGN KEY (`policy_id`)
    REFERENCES `wds`.`insurance_policy` (`policy_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
```

```
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`payment` (
  `payment_id` BIGINT(20) NOT NULL COMMENT 'UNIQUE PAYMENT ID',
  `payment_date` DATE NOT NULL COMMENT 'PAYMENT DATE',
  `payment_method` VARCHAR(10) NOT NULL COMMENT 'PAYMENT METHOD',
  `payment_amount` DECIMAL(10,2) NOT NULL,
  `invoice_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`payment_id`),
  INDEX `payment_invoice_fk` (`invoice_id` ASC)  ,
  CONSTRAINT `payment_invoice_fk`
    FOREIGN KEY (`invoice_id`)
    REFERENCES `wds`.`invoice` (`invoice_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`user_feature`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`user_feature` (
  `feature_id` BIGINT(20) NOT NULL,
  `user_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`feature_id`, `user_id`),
  INDEX `user_feature_user_fk` (`user_id` ASC)  ,
  CONSTRAINT `user_feature_feature_fk`
    FOREIGN KEY (`feature_id`)
    REFERENCES `wds`.`feature` (`feature_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `user_feature_user_fk`
    FOREIGN KEY (`user_id`)
    REFERENCES `wds`.`user` (`user_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`vehicle`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`vehicle` (
  `vin` VARCHAR(17) NOT NULL COMMENT 'VIN OF VEHICLE',
  `make` VARCHAR(30) NOT NULL COMMENT 'MAKE MODEL YEAR OF VEHICLE',
  `model` VARCHAR(30) NOT NULL,
  `year` YEAR(4) NOT NULL,
  `vehicle_status` VARCHAR(1) NOT NULL COMMENT 'STATUS OF VEHICLE',
  `policy_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`vin`),
  INDEX `vehicle_auto_ins_fk` (`policy_id` ASC)  ,
  CONSTRAINT `vehicle_auto_ins_fk`
    FOREIGN KEY (`policy_id`)
    REFERENCES `wds`.`auto_insurance` (`policy_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `wds`.`vehicle_driver`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `wds`.`vehicle_driver` (
  `licence_number` VARCHAR(15) NOT NULL,
  `vin` VARCHAR(17) NOT NULL,
  INDEX `vehicle_driver_driver_fk` (`licence_number` ASC)  ,
  INDEX `vehicle_driver_vehicle_fk` (`vin` ASC)  ,
  CONSTRAINT `vehicle_driver_driver_fk`
    FOREIGN KEY (`licence_number`)
    REFERENCES `wds`.`driver` (`licence_number`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `vehicle_driver_vehicle_fk`
    FOREIGN KEY (`vin`)
    REFERENCES `wds`.`vehicle` (`vin`)
    ON DELETE CASCADE
```

```
   ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

USE `wds` ;
```

Procedure

```
-- -----------------------------------------------------
-- procedure add_authentication_details
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `add_authentication_details`(IN authKey
VARCHAR(40), IN userId int)
BEGIN
	insert into authentication(auth_key,user_id) value(authKey,userId);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure add_driver
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `add_driver`(IN licence_number varchar(15),IN
first_name varchar(30), IN last_name varchar(30),
IN dob date,IN vin varchar(17))
BEGIN
	case
   when (select count(*) from driver d where d.licence_number = licence_number) = 0 then
		insert into driver value(licence_number,first_name,last_name,dob);
	end case;
   insert into vehicle_driver values(licence_number,vin);
END$$

DELIMITER ;

-- -----------------------------------------------------
```

```
-- procedure add_home
-- -----------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `add_home`(IN homeId int,IN purchaseDate
date,IN purchaseValue decimal,IN area decimal,
IN home_type varchar(1),IN auto_fire_notification int,IN home_security_system int,IN
swimming_pool varchar(1),
IN basement int, IN policy_id int, IN st_address varchar(45),IN city varchar(30), IN state
varchar(2),IN zipcode varchar(5))
BEGIN
	insert into home
values(homeId,purchaseDate,purchaseValue,area,home_type,auto_fire_notification,home_secu
rity_system,
   swimming_pool,basement,policy_id,st_address,city,state,zipcode);
END$$

DELIMITER ;

-- -----------------------------------------------
-- procedure add_insurance
-- -----------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `add_insurance`(IN policyId int,IN startDate
date,IN endDate date,IN premiumAmount decimal, IN policyType varchar(1),
IN userId int, IN noClaimBonus decimal ,IN discountIn decimal)
BEGIN
	insert into insurance_policy
values(policyId,startDate,endDate,premiumAmount,policyType,userId);
   case
		when policyType = 'A' Then
			insert into auto_insurance values(policyId,noClaimBonus);
		when policyType = 'H' Then
			insert into home_insurance values(policyId,discountIn);
	end case;
END$$

DELIMITER ;

-- -----------------------------------------------
```

```
-- procedure add_new_user
-- -----------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `add_new_user`(IN user_id_in int, IN
first_name_in varchar(30),
IN last_name_in varchar(50),
IN gender_in varchar(1),
IN marital_status_in varchar(1),
IN st_address_in varchar(60),
IN city_in varchar(30),
IN state_in varchar(2),
IN zipcode_in varchar(5),
IN user_type_in varchar(1),
IN designation_in varchar(30),
IN emp_department_in varchar(30),
IN emailid_in varchar(50),
IN password_in varchar(500), IN authKey_in varchar(40))
BEGIN
        INSERT INTO
user(user_id,first_name,last_name,gender,marital_status,st_address,city,state,zipcode,user_typ
e
   ,designation,emp_department,email_id,password)

VALUES(user_id_in,first_name_in,last_name_in,gender_in,marital_status_in,st_address_in,city
_in,state_in,zipcode_in,user_type_in
   ,designation_in,emp_department_in,emailid_in,password_in);
        INSERT INTO authentication(auth_key,user_id) VALUES(authKey_in,user_id_in);
END$$

DELIMITER ;

-- -----------------------------------------------
-- procedure add_vehicle
-- -----------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `add_vehicle`(IN vin varchar(17),IN make
varchar(30),IN model varchar(30),IN yearIn year(4),IN vehicle_status varchar(1),IN policy_id int)
BEGIN
        insert into vehicle values(vin,make,model,yearIn,vehicle_status,policy_id);
```

```
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure delete_record
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `delete_record`(IN tablename varchar(20), IN
rowidint int,IN rowidint1 int, IN rowidchar varchar(50),IN rowidchar1 varchar(50) )
BEGIN
        case
   when tablename = 'driver' then delete from driver where licence_number = rowidchar;
   when tablename = 'home' then delete from home where home_id = rowidint;
   when tablename = 'insurance_policy' then delete from insurance_policy where policy_id =
rowidint;
   when tablename = 'invoice' then delete from invoice where invoice_id = rowidint;
   when tablename = 'payment' then delete from payment where payment_id = rowidint;
   when tablename = 'user' then delete from user where user_id = rowidint;
   when tablename = 'user_feature' then delete from user_feature where user_id = rowidint and
feature_id = rowidint1;
   when tablename = 'vehicle' then delete from vehicle where vin = rowidchar;
   when tablename = 'vehicle_driver' then delete from vehicle_driver where licence_number =
rowidchar and vin = rowidchar1;
   end case;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure get_access_details
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_access_details`(IN userId int, IN apiIn
varchar(100))
BEGIN
        select * from user_feature
   where user_id = userId and feature_id = (select feature_id from feature where api = apiIn);
END$$
```

```
DELIMITER ;

-- ----------------------------------------------------
-- procedure get_authentication_details
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_authentication_details`(IN authkey
varchar(40))
BEGIN
	select * from authentication where auth_key = authkey;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure get_customer_list
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_customer_list`()
BEGIN
	select * from user where user_type = 'C';
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure get_insurance_details
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_insurance_details`(IN userId int)
BEGIN
	select i.policy_id, i.start_date, i.end_date, i.premium_amount, i.policy_type,
ai.no_claim_bonus, hi.discount,

h.home_id,h.purchase_date,h.purchase_value,h.area,h.home_type,h.auto_fire_notification,h.ho
```

```
me_security_system,h.swimming_pool,h.basement,h.st_address,h.city,h.state,h.zipcode,v.vin,v.
make,v.model,v.year,v.vehicle_status,d.licence_number,d.first_name,d.last_name
        from insurance_policy i
        left join auto_insurance ai on i.policy_id = ai.policy_id
        left join home_insurance hi on i.policy_id = hi.policy_id
   left join home h on hi.policy_id = h.policy_id
   left join vehicle v on ai.policy_id = v.policy_id
   left join vehicle_driver vd on v.vin = vd.vin
   left join driver d on vd.licence_number = d.licence_number
   where i.user_id = userId;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure get_invoice_detail
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_invoice_detail`(IN policyId int)
BEGIN
        select i.invoice_id, i.invoice_date,i.payment_due_date, i.invoice_amount,
i.policy_id,p.payment_id,p.payment_date,
   p.payment_method,p.payment_amount
   from invoice i
   left join payment p on i.invoice_id = p.invoice_id
   where i.policy_id = policyId;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure get_restricted_feature
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_restricted_feature`(IN apiIn varchar(100))
BEGIN
        select * from feature where api = apiIn and customer_access <> 1;
END$$
```

```
DELIMITER ;

-- ----------------------------------------------------
-- procedure get_user_by_emailid
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_user_by_emailid`(IN emailid varchar(50))
BEGIN
	SELECT * FROM user where email_id = emailid;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure get_user_details
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `get_user_details`(IN userid int)
BEGIN
	SELECT * FROM user where user_id = userid;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure provide_access
-- ----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `provide_access`(IN featureId int, IN userId int)
BEGIN
	Insert into user_feature values(featureId,userId);
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure update_password
```

```
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `update_password`(IN emailId varchar(50),IN
passwordIn varchar(500))
BEGIN
	UPDATE user
	SET password = passwordIn
	WHERE email_id = emailId;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure update_payment
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `update_payment`(IN paymentId int, IN
paymentDate date, IN paymentMethod varchar(10),IN paymentAmount double, IN invoiceId int)
BEGIN
	insert into payment values(paymentId, paymentDate,paymentMethod,
paymentAmount,invoiceId);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure update_user_details
-- -----------------------------------------------------

DELIMITER $$
USE `wds`$$
CREATE DEFINER=`asif`@`%` PROCEDURE `update_user_details`(IN userId int, IN firstName
varchar(30), IN lastName varchar(30), IN genderIn Enum('M','F'),
IN maritalStatus ENUM('M', 'S', 'W'), IN stAddress VARCHAR(60), IN city VARCHAR(30), IN
state VARCHAR(2), IN zipcode VARCHAR(5))
BEGIN
	update user
   set first_name = firstName,
   last_name = lastName,
```

```
    gender = genderIn,
    marital_status = maritalStatus,
    st_address = stAddress,
    city = city,
    state = state,
    zipcode = zipcode
    where user_id = userId;
END$$
```

Trigger

```
DELIMITER ;
USE `wds`;

DELIMITER $$
USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`user_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`user`
FOR EACH ROW
BEGIN
        if (length(new.zipcode) <> 5 OR length(new.state) <> 2 OR length(new.user_type) <> 1
        OR length(new.marital_status) <> 1 OR length(new.gender) <> 1 OR (new.user_type <>
'E' AND new.user_type <>'C')
        OR (new.gender <> 'F' AND new.gender <> 'M' AND new.gender IS NOT NULL)
    OR (new.marital_status <> 'M' AND new.marital_status <> 'S' AND new.marital_status <>
'W')) then
        signal sqlstate '45000'
            set message_text = 'Invalid input';
    end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`user_BEFORE_UPDATE`
BEFORE UPDATE ON `wds`.`user`
FOR EACH ROW
BEGIN
        if (length(new.zipcode) <> 5 OR length(new.state) <> 2 OR length(new.user_type) <> 1
        OR length(new.marital_status) <> 1 OR length(new.gender) <> 1 OR (new.user_type <>
'E' AND new.user_type <>'C')
```

```
      OR (new.gender <> 'F' AND new.gender <> 'M' AND new.gender IS NOT NULL)
   OR (new.marital_status <> 'M' AND new.marital_status <> 'S' AND new.marital_status <>
'W')) then
      signal sqlstate '45000'
         set message_text = 'Invalid input';
   end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`insurance_policy_AFTER_INSERT`
AFTER INSERT ON `wds`.`insurance_policy`
FOR EACH ROW
BEGIN
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),now(),DATE_ADD(NOW(), INTERVAL 1
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 2
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 3
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 4
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 5
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 6
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 7
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 8
MONTH),(new.premium_amount/12),new.policy_id);
         insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL 9
MONTH),(new.premium_amount/12),new.policy_id);
```

```
        insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL
10 MONTH),(new.premium_amount/12),new.policy_id);
        insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL
11 MONTH),(new.premium_amount/12),new.policy_id);
        insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL
12 MONTH),(new.premium_amount/12),new.policy_id);
        insert into invoice
values(FLOOR(RAND()*(999999-100000)+1000000),NOW(),DATE_ADD(NOW(), INTERVAL
13 MONTH),(new.premium_amount/12),new.policy_id);
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`insurance_policy_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`insurance_policy`
FOR EACH ROW
BEGIN
 if (length(new.policy_type) <> 1 OR
   (new.policy_type <> 'A' AND new.policy_type <>'H')) then
      signal sqlstate '45000'
         set message_text = 'Invalid input';
   end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`insurance_policy_BEFORE_UPDATE`
BEFORE UPDATE ON `wds`.`insurance_policy`
FOR EACH ROW
BEGIN
 if (length(new.policy_type) <> 1 OR
   (new.policy_type <> 'A' AND new.policy_type <>'H')) then
      signal sqlstate '45000'
         set message_text = 'Invalid input';
   end if;
END$$

USE `wds`$$
```

```
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`auto_insurance_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`auto_insurance`
FOR EACH ROW
BEGIN
        DECLARE d VARCHAR(1);
   SELECT a.policy_type
   INTO d
   FROM insurance_policy a
   WHERE a.policy_id = new.policy_id;

   IF ( d IS NULL OR d <> 'A' ) THEN
                        signal sqlstate '45000'
        set message_text = 'Invalid input';
   END IF;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`home_insurance_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`home_insurance`
FOR EACH ROW
BEGIN
        DECLARE d VARCHAR(1);
   SELECT a.policy_type
   INTO d
   FROM insurance_policy a
   WHERE a.policy_id = new.policy_id;

   IF ( d IS NULL OR d <> 'H' ) THEN
                        signal sqlstate '45000'
        set message_text = 'Invalid input';
   END IF;

END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`home_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`home`
```

```
FOR EACH ROW
BEGIN
	if((new.home_type <> 'S' AND new.home_type <> 'M' AND new.home_type <> 'C' AND
   new.home_type <> 'T') OR (new.auto_fire_notification <> 0 AND new.auto_fire_notification
<> 1)
   OR (new.home_security_system <> 0 AND new. home_security_system <> 1)
   OR (new.basement <>0 AND new.basement <> 1)
   OR (new.swimming_pool IS NOT NULL AND new.swimming_pool <> 'U' AND
new.swimming_pool <> 'O'
   AND new.swimming_pool <> 'I' AND new.swimming_pool <> 'M'))
   then
	signal sqlstate '45000'
	set message_text = 'Invalid input';
   end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`home_BEFORE_UPDATE`
BEFORE UPDATE ON `wds`.`home`
FOR EACH ROW
BEGIN
	if((new.home_type <> 'S' AND new.home_type <> 'M' AND new.home_type <> 'C' AND
   new.home_type <> 'T') OR (new.auto_fire_notification <> 0 AND new.auto_fire_notification
<> 1)
   OR (new.home_security_system <> 0 AND new. home_security_system <> 1)
   OR (new.basement <>0 AND new.basement <> 1)
   OR (new.swimming_pool IS NOT NULL AND new.swimming_pool <> 'U' AND
new.swimming_pool <> 'O'
   AND new.swimming_pool <> 'I' AND new.swimming_pool <> 'M'))
   then
	signal sqlstate '45000'
	set message_text = 'Invalid input';
   end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`payment_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`payment`
FOR EACH ROW
```

```
BEGIN
	if (new.payment_method <> 'Credit' AND new.payment_method <> 'Debit' AND
    new.payment_method <> 'PayPal' AND new.payment_method <> 'Check') then
		signal sqlstate '45000'
		set message_text = 'Invalid input';
    end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`payment_BEFORE_UPDATE`
BEFORE UPDATE ON `wds`.`payment`
FOR EACH ROW
BEGIN
	if (new.payment_method <> 'Credit' AND new.payment_method <> 'Debit' AND
    new.payment_method <> 'PayPal' AND new.payment_method <> 'Check') then
		signal sqlstate '45000'
		set message_text = 'Invalid input';
    end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`vehicle_BEFORE_INSERT`
BEFORE INSERT ON `wds`.`vehicle`
FOR EACH ROW
BEGIN
	if(new.vehicle_status <> 'L' AND new.vehicle_status <> 'F' AND
    new.vehicle_status <> 'O')
	then signal sqlstate '45000'
	set message_text = 'Invalid input';
    end if;
END$$

USE `wds`$$
CREATE
DEFINER=`asif`@`%`
TRIGGER `wds`.`vehicle_BEFORE_UPDATE`
BEFORE UPDATE ON `wds`.`vehicle`
FOR EACH ROW
BEGIN
```

```
        if(new.vehicle_status <> 'L' AND new.vehicle_status <> 'F' AND
    new.vehicle_status <> 'O')
        then signal sqlstate '45000'
        set message_text = 'Invalid input';
    end if;
END$$


DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

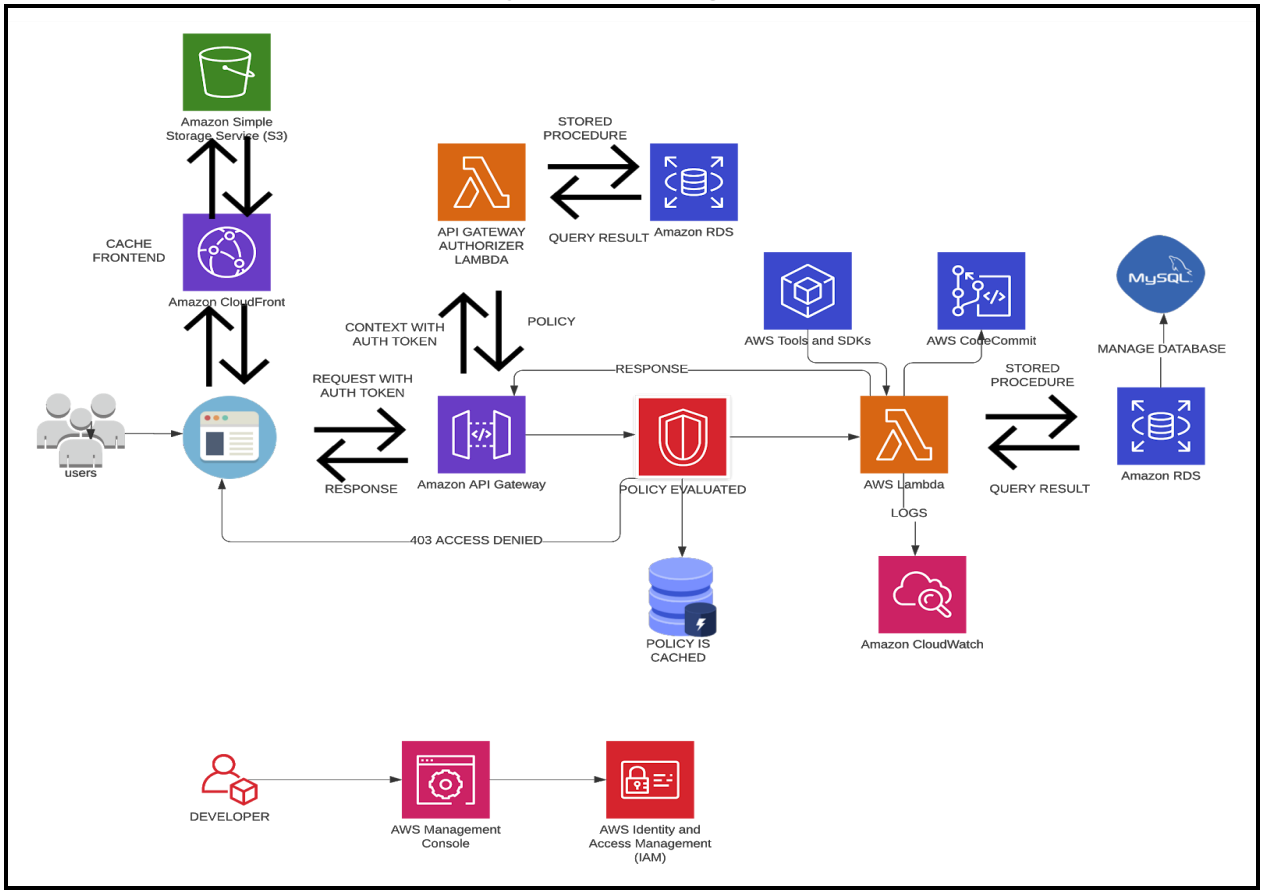
Event

```
CREATE EVENT IF NOT EXISTS `Clean_Older_Than_24_Hours`
ON SCHEDULE
  EVERY 1 HOUR
  COMMENT 'Clean up expired sessions'
  DO
    DELETE FROM authentication
    WHERE login_time < TIMESTAMP(DATE_SUB(NOW(), INTERVAL 24 hour));
```

Summary of development environment used

Programming/scripting languages : Java, HTML5, Javascript, Ajax, css
AWS : RDS, Lambda, CloudFront, Cloud Watch, S3, API Gateway, CloudWatch, IAM, CodeCommit
Database : MySQL
Designing tool: MySQL Workbench, Lucid Charts
Code editor : Eclipse , VS Code

System Design



Summary of Features Used

In our website for WDS Insurance company we have two views, one of customers and another for WDS employees. Below are the listed features for both the view of customer and employee.

For Customer

- Create new account.
- Login/Logout.
- Change password.
- View profile details.
- Edit profile details.
- Apply for home and auto insurance policy.
- View existing insurance details - Home, Vehicle and Driver details.
- View invoice for an insurance.
- View payment history for an invoice.
- View payment due for an invoice.
- Pay full or partial for an invoice.
- Session maintained for maximum 24 hours.

For Employee

- Login/Logout.
- View profile details of a customer.
- Edit profile details of a customer.
- View Insurance details of a customer.
- Add or remove a new employee.
- Provide access to employee.
- Remove access from employee.
- Delete user, insurance, invoice, payment, home, vehicle, driver with cascade on delete feature.
- Remove a driver from insurance of a vehicle.
- All the features are access driven.