

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,

SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_F
OR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema progetto

DROP SCHEMA IF EXISTS `progetto` ;

-- Schema progetto

CREATE SCHEMA IF NOT EXISTS `progetto` DEFAULT CHARACTER SET utf8 ;

USE `progetto` ;

-- Table `progetto`.`general_user`

DROP TABLE IF EXISTS `progetto`.`general_user` ;

CREATE TABLE IF NOT EXISTS `progetto`.`general_user` (

`Username` VARCHAR(45) NOT NULL,

`Dati_anagrafici` DATE NOT NULL,

`ind_fatturazione` VARCHAR(45) NOT NULL,

`ind_domicilio` VARCHAR(45) NOT NULL,

`privilegio` ENUM('admin', 'utente') NOT NULL DEFAULT 'utente',

`password` CHAR(32) NOT NULL,

`telefono` VARCHAR(10) NULL,

```
`cellulare` VARCHAR(10) NULL,  
`Email` VARCHAR(45) NULL,  
PRIMARY KEY (`Username`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb3;
```

```
-- -----  
-- Table `progetto`.`Categoria`  
-- -----
```

```
DROP TABLE IF EXISTS `progetto`.`Categoria` ;
```

```
CREATE TABLE IF NOT EXISTS `progetto`.`Categoria` (  
  `IDCategoria` INT NOT NULL AUTO_INCREMENT,  
  `Nome` VARCHAR(45) NOT NULL,  
  `UsernameC` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`IDCategoria`),  
  INDEX `fk_Categoria_General user1_idx` (`UsernameC` ASC) VISIBLE,  
  CONSTRAINT `fk_Categoria_General user1`  
    FOREIGN KEY (`UsernameC`)  
    REFERENCES `progetto`.`general_user` (`Username`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `progetto`.`Annuncio`  
-- -----
```

```
DROP TABLE IF EXISTS `progetto`.`Annuncio` ;
```

```
CREATE TABLE IF NOT EXISTS `progetto`.`Annuncio` (  
  `IDAnnuncio` INT NOT NULL AUTO_INCREMENT,  
  `Titolo` VARCHAR(45) NOT NULL,  
  `Contenuto` VARCHAR(255) NOT NULL,  
  `Data` DATE NOT NULL,  
  `UsernameC` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`IDAnnuncio`),  
  INDEX `fk_Annuncio_General user1_idx` (`UsernameC` ASC) VISIBLE,  
  CONSTRAINT `fk_Annuncio_General user1`  
    FOREIGN KEY (`UsernameC`)  
    REFERENCES `progetto`.`general_user` (`Username`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```

`IDAnnuncio` INT NOT NULL AUTO_INCREMENT,
`titolo` VARCHAR(45) NOT NULL,
`prezzo` INT NOT NULL,
`oggetto` VARCHAR(45) NOT NULL,
`descrizione` VARCHAR(100) NOT NULL,
`stato` ENUM('venduto', 'online') NOT NULL,
`UsernameA` VARCHAR(45) NOT NULL,
`IDA` INT NOT NULL,
PRIMARY KEY (`IDAnnuncio`, `UsernameA`),
INDEX `fk_Annuncio_General user1_idx` (`UsernameA` ASC) VISIBLE,
INDEX `fk_Annuncio_Categoria1_idx` (`IDA` ASC) VISIBLE,
CONSTRAINT `fk_Annuncio_General user1`
    FOREIGN KEY (`UsernameA`)
    REFERENCES `progetto`.`general_user` (`Username`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Annuncio_Categoria1`
    FOREIGN KEY (`IDA`)
    REFERENCES `progetto`.`Categoria` (`IDCategoria`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `progetto`.`Messaggio_privato`
-----

```

```

DROP TABLE IF EXISTS `progetto`.`Messaggio_privato` ;

```

```

CREATE TABLE IF NOT EXISTS `progetto`.`Messaggio_privato` (
  `IDMessaggio` INT NOT NULL AUTO_INCREMENT,
  `timestamp` DATETIME NOT NULL,

```

```

`msgtxt` VARCHAR(200) NOT NULL,
`IDAnnuncioM` INT NOT NULL,
`UsernameM` VARCHAR(45) NOT NULL,
`UsernameCM` VARCHAR(45) NOT NULL,
`UsernameRicevente` VARCHAR(45) NOT NULL,
PRIMARY KEY (`IDMessaggio`, `IDAnnuncioM`, `UsernameM`, `UsernameCM`, `UsernameRicevente`),
INDEX `fk_Messaggio privato_Annuncio1_idx` (`IDAnnuncioM` ASC, `UsernameM` ASC) VISIBLE,
INDEX `fk_Messaggio privato_General user1_idx` (`UsernameCM` ASC) VISIBLE,
CONSTRAINT `fk_Messaggio privato_Annuncio1`
  FOREIGN KEY (`IDAnnuncioM`, `UsernameM`)
  REFERENCES `progetto`.`Annuncio` (`IDAnnuncio`, `UsernameA`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Messaggio privato_General user1`
  FOREIGN KEY (`UsernameCM`)
  REFERENCES `progetto`.`general_user` (`Username`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `progetto`.`Commento`
-----

DROP TABLE IF EXISTS `progetto`.`Commento` ;

CREATE TABLE IF NOT EXISTS `progetto`.`Commento` (
  `IDCommento` INT NOT NULL AUTO_INCREMENT,
  `timestamp_commento` DATETIME NOT NULL,
  `msgtxt_commento` VARCHAR(350) NOT NULL,
  `IDAnnuncioC` INT NOT NULL,
  `Username_Commento` VARCHAR(45) NOT NULL,

```

```

`UsernameCommento` VARCHAR(45) NOT NULL,
PRIMARY KEY (`IDCommento`, `IDAnnuncioC`, `Username_Commento`, `UsernameCommento`),
INDEX `fk_Commento_Annuncio1_idx` (`IDAnnuncioC` ASC, `Username_Commento` ASC) VISIBLE,
INDEX `fk_Commento_General user1_idx` (`UsernameCommento` ASC) VISIBLE,
CONSTRAINT `fk_Commento_Annuncio1`
    FOREIGN KEY (`IDAnnuncioC`, `Username_Commento`)
    REFERENCES `progetto`.`Annuncio` (`IDAnnuncio`, `UsernameA`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Commento_General user1`
    FOREIGN KEY (`UsernameCommento`)
    REFERENCES `progetto`.`general_user` (`Username`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `progetto`.`seguì`
-----

```

```

DROP TABLE IF EXISTS `progetto`.`seguì` ;

```

```

CREATE TABLE IF NOT EXISTS `progetto`.`seguì` (
    `UsernameS` VARCHAR(45) NOT NULL,
    `IDAnnuncioS` INT NOT NULL,
    PRIMARY KEY (`UsernameS`, `IDAnnuncioS`),
    INDEX `fk_General user_has_Annuncio_Annuncio1_idx` (`IDAnnuncioS` ASC) VISIBLE,
    INDEX `fk_General user_has_Annuncio_General user_idx` (`UsernameS` ASC) VISIBLE,
    CONSTRAINT `fk_General user_has_Annuncio_General user`
        FOREIGN KEY (`UsernameS`)
        REFERENCES `progetto`.`general_user` (`Username`)
        ON DELETE NO ACTION

```

```
ON UPDATE NO ACTION,  
CONSTRAINT `fk_General user_has_Annuncio_Annuncio1`  
FOREIGN KEY (`IDAnnuncioS`)  
REFERENCES `progetto`.`Annuncio` (`IDAnnuncio`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
USE `progetto` ;
```

```
-- -----  
-- procedure inserireUtente  
-- -----
```

```
USE `progetto`;  
DROP procedure IF EXISTS `progetto`.`inserireUtente`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `inserireUtente` (in var_username varchar(45),in var_data DATE ,in  
var_ind_fatturazione varchar(45), in var_ind_domicilio varchar(45),  
in var_password char(32), in var_privilegio enum('admin','utente'), in var_telefono varchar(10), in  
var_cellulare varchar(10), in var_email varchar(45))
```

```
BEGIN
```

```
insert into `General_user`(Username, Dati_anagrafici, ind_fatturazione, Ind_Domicilio, password, Privilegio,  
telefono, Cellulare, Email)
```

```
values (var_username, var_data, var_ind_fatturazione, var_ind_domicilio, var_password, var_privilegio,  
var_telefono, var_cellulare, var_email);
```

```
END$$
```

```
DELIMITER ;
```

```

-----

-- procedure login

-----

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`login`;

DELIMITER $$

USE `progetto`$$

CREATE PROCEDURE `login` (in var_username char(32), in var_pass varchar(45), out var_role
INT)
BEGIN
    declare var_user_role ENUM('admin','utente');
    select `privilegio` from `General_user`
        where `username` = var_username
    and `password` = var_pass
    into var_user_role;
    if var_user_role='admin' then

        set var_role=1;
        elseif var_user_role= 'utente' then

            set var_role=2;
        end if;

END$$

DELIMITER ;

-----

-- procedure creare_annuncio

```

```
USE `progetto`;
```

```
DROP procedure IF EXISTS `progetto`.`creare_annuncio`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `creare_annuncio` (in var_titolo varchar(45), in var_prezzo int, in var_oggetto  
varchar(45), in var_descrizione varchar(100), in var_stato enum('venduto','online') , in var_username  
varchar(45), in var_categoria varchar(45), out var_ritorno int)
```

```
BEGIN
```

```
    declare var INT;
```

```
    declare var2 INT;
```

```
    declare exit handler for sqlexception
```

```
begin
```

```
    rollback;
```

```
    resignal;
```

```
end;
```

```
    set transaction isolation level repeatable read;
```

```
start transaction;
```

```
    SELECT COUNT(*)
```

```
from `General_user`
```

```
where Username=var_username into var;
```

```
IF var > 0
```

```
then
```

```
    SELECT COUNT(*)
```

```
from Categoria
```

```
where Nome =var_categoria into var2;
```

```
if var2>0
```

```
then
```

```
    INSERT INTO Annuncio (titolo, prezzo, oggetto, descrizione, stato, UsernameA, IDA)
```



```
        values (var_titolo, var_prezzo, var_oggetto, var_descrizione , var_stato, var_username
,(select IDCategoria from Categoria where (Nome=Var_categoria)));
```

```
        Set var_ritorno = last_insert_id();
```

```
    end if;
```

```
    end if;
```

```
commit;
```

```
END$$
```

```
DELIMITER ;
```

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

```
-- procedure seguire_annuncio
```

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

```
USE `progetto`;
```

```
DROP procedure IF EXISTS `progetto`.`seguire_annuncio`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `seguire_annuncio` (in var_idannuncio int, in var_username varchar(45))
```

```
BEGIN
```

```
    declare var int;
```

```
    declare var2 int;
```

```
    declare exit handler for sqlexception
```

```
begin
```

```
    rollback;
```

```
    resignal;
```

```
end;
```

```

        set transaction isolation level read committed;

start transaction;

select count(*)
from General_user
where Username=var_username into var;

if var>0
then

        select count(*)

        from Annuncio

        where IDAnnuncio=var_idannuncio into var2;

if var2>0
then

        INSERT INTO segui ( UsernameS, IDAnnuncioS)

        VALUES (var_username, var_idannuncio);

end if;

end if;

commit;

END$$

DELIMITER ;

-----

-- procedure creareCategoria
-----

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`creareCategoria`;

DELIMITER $$

USE `progetto`$$

CREATE DEFINER=`Manuel`@`localhost` PROCEDURE `creareCategoria`(in var_nome varchar(45), in
var_username varchar(45))

```

```
BEGIN

    declare var INT;

    declare exit handler for sqlexception

begin

    rollback;

    resignal;

end;


    set transaction isolation level repeatable read;

start transaction;
```

```
        SELECT count(*)
from `General_user`
where Username=var_username into var ;

IF var > 0
then

        INSERT INTO Categoria (Nome, UsernameC)
        VALUES (var_nome, var_username);
```

```
end if;

commit;

END$$
```

```
DELIMITER ;
```

```
-- -----
-- procedure scrivi_commento
-- -----
```

```
USE `progetto`;
```

```
DROP procedure IF EXISTS `progetto`.`scrivi_commento`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `scrivi_commento` (in var_msgtctC varchar(350), in var_idannuncio int, in  
var_username varchar(45), in var_usernameec varchar(45))
```

```
BEGIN
```

```
    declare var INT;
```

```
    declare var2 INT;
```

```
    declare exit handler for sqlexception
```

```
begin
```

```
    rollback;
```

```
    resignal;
```

```
end;
```

```
    set transaction isolation level repeatable read;
```

```
start transaction;
```

```
    SELECT count(*)
```

```
from `Annuncio`
```

```
where `IDAnnuncio` = var_idannuncio into var ;
```

```
IF var > 0
```

```
then
```

```
        SELECT count(*)
```

```
        from `General_user`
```

```
        where `Username`=var_username into var2 ;
```

```
        IF var2> 0
```

```
        then
```

```
                INSERT INTO Commento (timestamp_commento,  
msgtxt_commento, IDAnnuncioC, Username_Commento, UsernameCommento)
```

```
                VALUES ((now()), var_msgtctC, var_idannuncio, var_username,  
var_usernameec);
```

```
        end if;
```

```

        end if;

    commit;

END$$

DELIMITER ;

-----

-- procedure scrivi_messaggio
-----

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`scrivi_messaggio`;

DELIMITER $$

USE `progetto`$$

CREATE PROCEDURE `scrivi_messaggio` (in var_msgtct varchar(350), in var_idannuncio int, in var_username
varchar(45), in var_scrittore varchar(45), in var_ricevente varchar(45))

BEGIN

    declare var INT;

    declare var2 INT;

    declare exit handler for sqlexception

begin

    rollback;

    resignal;

end;


    set transaction isolation level repeatable read;

start transaction;

    SELECT count(*)

from `Annuncio`

where `IDAnnuncio` = var_idannuncio into var ;

IF var > 0

```

then

SELECT count(*)

from `General_user`

where `Username`=var_username into var2 ;

IF var2> 0

then

INSERT INTO Messaggio_privato(timestamp, msgtxt, IDAnnuncioM,
UsernameM, UsernameCM,UsernameRicevente)

VALUES ((now()), var_msgtct, var_idannuncio, var_username,
var_scrittore, var_ricevente);

end if;

end if;

commit;

END\$\$

DELIMITER ;

-- procedure annuncioVenduto

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`annuncioVenduto`;

DELIMITER \$\$

USE `progetto`\$\$

CREATE PROCEDURE `annuncioVenduto` (in var_IDannuncio int, in var_stato ENUM('venduto','online'))

begin

declare var int;

declare exit handler for sqlexception

begin

```

rollback;

resignal;

end;

set transaction isolation level Read committed;

start transaction;

    select count(*)
    from Annuncio
    where IDAnnuncio=var_IDannuncio into var;
    if var>0
    then
        UPDATE `Annuncio`
        SET
            `stato` = var_stato
        WHERE
            `IDAnnuncio` = var_IDannuncio;
    end if;

commit;

END$$

```

```

DELIMITER ;

```

```

-- -----
-- procedure modifica_annuncio
-- -----

```

```

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`modifica_annuncio`;

```

```

DELIMITER $$

USE `progetto`$$

```

```
CREATE PROCEDURE `modifica_annuncio` (in var_IDannuncio int, in var_titolo varchar(45), in var_prezzo int,
in var_descrizione varchar(100))
```

```
BEGIN
```

```
    declare var int;
```

```
    declare exit handler for sqlexception
```

```
begin
```

```
    rollback;
```

```
    resignal;
```

```
end;
```

```
    set transaction isolation level repeatable read;
```

```
start transaction;
```

```
select count(*)
```

```
from Annuncio
```

```
where IDannuncio=var_IDannuncio into var;
```

```
if var>0
```

```
then
```

```
    UPDATE `Annuncio`
```

```
    SET
```

```
        `titolo`= var_titolo,
```

```
        `prezzo`= var_prezzo,
```

```
        `descrizione`=var_descrizione
```

```
    WHERE
```

```
        `IDannuncio`= var_IDannuncio;
```

```
    end if;
```

```
commit;
```

```
END$$
```

```
DELIMITER ;
```

```
-- -----  
-- procedure prendi_annunciDiunVenditore
```

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`prendi_annunciDiunVenditore`;

DELIMITER \$\$

USE `progetto`\$\$

CREATE PROCEDURE `prendi_annunciDiunVenditore` (in var_username varchar(45))

BEGIN

 declare var int;

 declare exit handler for sqlexception

begin

 rollback;

 resignal;

end;

set transaction isolation level read committed;

set transaction read only;

start transaction;

select count(*)

from General_user

where Username=var_username into var;

if var>0

then

 select IDAnnuncio , titolo, prezzo, oggetto, descrizione, (select nome from Categoria where
(IDA=IDCategoria)) as IDA

 from Annuncio

 where UsernameA= var_username;

end if;

END\$\$

DELIMITER ;

-- -----

-- procedure listaAnnunci

-- -----

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`listaAnnunci`;

DELIMITER \$\$

USE `progetto`\$\$

CREATE PROCEDURE `listaAnnunci` ()

BEGIN

declare var int;

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

set transaction read only;

start transaction;

select count(*)

from Annuncio

where stato != 'venduto' into var;

if var>0

then

select IDAnnuncio, titolo, prezzo, oggetto, descrizione, UsernameA, IDA

from Annuncio;

```
end if;

commit;

end$$
```

```
DELIMITER ;
```

```
-- -----
-- procedure prendiUno_Annuncio
-- -----
```

```
USE `progetto`;

DROP procedure IF EXISTS `progetto`.`prendiUno_Annuncio`;
```

```
DELIMITER $$

USE `progetto`$$

CREATE PROCEDURE `prendiUno_Annuncio` (in var_idannuncio int)
BEGIN
    declare var int;

    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;

    set transaction read only;

    start transaction;

    select count(*)
    from Annuncio
    where IDAnnuncio=var_idannuncio and stato != 'venduto' into var;

    if var>0
    then
```

```

        select IDAnnuncio, titolo, prezzo, oggetto, descrizione, stato, UsernameA, IDA
        from `Annuncio`
        where IDAnnuncio=var_idannuncio;

    else

        signal sqlstate '45000'

        set MESSAGE_TEXT ='Lannuncio non è più in commercio ';

    end if;

    commit;

END$$

```

```

DELIMITER ;

```

```

-----
-- procedure VisualizzaCommenti
-----

```

```

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`VisualizzaCommenti`;

```

```

DELIMITER $$

USE `progetto`$$

CREATE PROCEDURE `VisualizzaCommenti` (in var_id int)
BEGIN
    declare var int;

    declare exit handler for sqlexception

    begin
        rollback;

        resignal;

    end;

    set transaction isolation level read committed;

```

```

set transaction read only;

start transaction;

select count(*)
from Annuncio
where IDAnnuncio= var_id into var;

if var>0
then

        select IDCommento, timestamp_commento, msgtxt_commento, UsernameCommento,
UsernameCommento

        from Commento

        where IDAnnuncioC= var_id;

END IF;
END$$

```

```

DELIMITER ;

```

```

-- -----
-- procedure VisualizzaMessaggi
-- -----

```

```

USE `progetto`;

DROP procedure IF EXISTS `progetto`.`VisualizzaMessaggi`;

DELIMITER $$

USE `progetto`$$

CREATE PROCEDURE `VisualizzaMessaggi` (in var_id int,in var_user varchar(45))

BEGIN

        declare var int;

        declare var2 int;

        declare exit handler for sqlexception

begin

        rollback;

```

```

        resignal;
end;

set transaction isolation level read committed;
set transaction read only;

start transaction;

select count(*)
from Annuncio
where IDAnnuncio= var_id into var;
if var>0
then
        select count(*)
        from General_user
        where var_user=Username into var2;
        if var2>0
        then
                select IDMessaggio, timestamp, msgtxt, usernameRicevente, UsernameCM,
UsernameM, IDAnnuncioM
                from Messaggio_privato
                where (UsernameM= var_user or usernameRicevente = var_user ) and
var_id=IDAnnuncioM;
        end if;
        end if;
END$$

DELIMITER ;

-----
-- procedure VisualizzaCategoria
-----

USE `progetto`;

```

```
DROP procedure IF EXISTS `progetto`.`VisualizzaCategoria`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `VisualizzaCategoria` ()
```

```
BEGIN
```

```
declare exit handler for sqlexception
```

```
begin
```

```
rollback;
```

```
resignal;
```

```
end;
```

```
set transaction isolation level read uncommitted;
```

```
set transaction read only;
```

```
start transaction;
```

```
select IDCategoria, nome
```

```
from Categoria;
```

```
END$$
```

```
DELIMITER ;
```

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

```
-- procedure Report
```

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

```
USE `progetto`;
```

```
DROP procedure IF EXISTS `progetto`.`Report`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `Report` ()
```

```
BEGIN
```

```
declare exit handler for sqlexception
```

```
begin
```

```
    rollback;
```

```
    resignal;
```

```
end;
```

```
set transaction isolation level read committed;
```

```
set transaction read only;
```

```
start transaction;
```

```
    select UsernameA,
```

```
           COUNT(CASE WHEN stato = 'venduto' THEN 1 END) AS annunci_venduti,
```

```
           COUNT(*) AS annunci_totali,
```

```
           ROUND((COUNT(CASE WHEN stato = 'venduto' THEN 1 END) * 100.0) / COUNT(*), 2) AS  
percentuale_venduti
```

```
    from Annuncio
```

```
    group by UsernameA;
```

```
END$$
```

```
DELIMITER ;
```

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

```
-- procedure VisualizzaLeChat
```

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

```
USE `progetto`;
```

```
DROP procedure IF EXISTS `progetto`.`VisualizzaLeChat`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
CREATE PROCEDURE `VisualizzaLeChat` (in var_User varchar(45))
```

```
BEGIN
```

```
    declare exit handler for sqlexception
```



```
begin
```

```
    rollback;
```

```
    resignal;
```

```
end;
```

```
set transaction isolation level read committed;
```

```
set transaction read only;
```

```
start transaction;
```

```
    select msgtxt,UsernameM, UsernameCM, UsernameRicevente, IDAnnuncioM, IDMessaggio
```

```
from Messaggio_privato
```

```
WHERE UsernameM IN (
```

```
    SELECT DISTINCT UsernameM
```

```
    FROM Messaggio_privato
```

```
    WHERE UsernameCM = var_User OR UsernameRicevente = var_User
```

```
);
```

```
END$$
```

```
DELIMITER ;
```

```
USE `progetto`;
```

```
DELIMITER $$
```

```
USE `progetto`$$
```

```
DROP TRIGGER IF EXISTS `progetto`.`MexNotifica` $$
```

```
USE `progetto`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `progetto`.`MexNotifica` BEFORE UPDATE ON `Annuncio` FOR  
EACH ROW
```

```
BEGIN
```

```
declare var int;
```

```
    declare var3 int;
```

```
declare varscri varchar(45);  
    select count(*)  
from segui  
where IDAnnuncioS = new.IDAnnuncio  
into var;  
set var3 = 0;
```

```
while var3 != var do
```

```
    select UsernameS into varscri  
from segui  
where IDAnnuncioS = new.IDAnnuncio  
LIMIT 1 OFFSET var3;
```

```
    Insert into Messaggio_privato(timestamp, msgtxt, IDAnnuncioM, UsernameM,  
UsernameCM,UsernameRicevente)  
    value((NOW()), 'ho modificato il post! vallo a guardare', new.IDAnnuncio,  
new.UsernameA,new.UsernameA, varscri);
```

```
    set var3 = var3 + 1;  
    end while;
```

```
END$$
```

```
USE `progetto`$$
```

```
DROP TRIGGER IF EXISTS `progetto`.`Pulizia` $$
```

```
USE `progetto`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `progetto`.`Pulizia` BEFORE UPDATE ON `Annuncio` FOR EACH  
ROW
```

```
BEGIN
```

```

        if new.stato='venduto' and old.stato != 'venduto'
then
        delete from Messaggio_privato where IDAnnuncioM= new.IDAnnuncio;
delete from Commento where IDAnnuncioC=new.IDAnnuncio;
delete from segui where IDAnnuncioS=new.IDAnnuncio;
        end if;
END$$

```

```

USE `progetto`$$
DROP TRIGGER IF EXISTS `progetto`.`Messaggio_privato_BEFORE_INSERT` $$
USE `progetto`$$
CREATE DEFINER = CURRENT_USER TRIGGER `progetto`.`Messaggio_privato_BEFORE_INSERT` BEFORE
INSERT ON `Messaggio_privato` FOR EACH ROW
BEGIN
        if length(new.msgtxt)>199
then
        signal sqlstate '45000'
set MESSAGE_TEXT ='il messaggio supera la lunghezza consentita';
        end if;
END$$

```

```

USE `progetto`$$
DROP TRIGGER IF EXISTS `progetto`.`Commento_BEFORE_INSERT` $$
USE `progetto`$$
CREATE DEFINER = CURRENT_USER TRIGGER `progetto`.`Commento_BEFORE_INSERT` BEFORE INSERT ON
`Commento` FOR EACH ROW
BEGIN
        if length(new.msgtxt_commento)>349
then
        signal sqlstate '45000'

```

```
set MESSAGE_TEXT ='il Commento supera la lunghezza consentita';  
end if;
```

```
END$$
```

```
DELIMITER ;
```

```
SET SQL_MODE = '';
```

```
DROP USER IF EXISTS Utente;
```

```
SET
```

```
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_F  
OR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
CREATE USER 'Utente' IDENTIFIED BY 'Utente02!';
```

```
GRANT EXECUTE ON procedure `progetto`.`login` TO 'Utente';
```

```
GRANT UPDATE, TRIGGER, SELECT, INSERT, DELETE ON TABLE `progetto`.`general_user` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`inserireUtente` TO 'Utente';
```

```
GRANT DELETE, INSERT, SELECT, UPDATE, TRIGGER ON TABLE `progetto`.`Annuncio` TO 'Utente';
```

```
GRANT DELETE, INSERT, SELECT, UPDATE, TRIGGER ON TABLE `progetto`.`Commento` TO 'Utente';
```

```
GRANT DELETE, INSERT, SELECT, UPDATE, TRIGGER ON TABLE `progetto`.`Messaggio_privato` TO 'Utente';
```

```
GRANT DELETE, INSERT, SELECT, UPDATE, TRIGGER ON TABLE `progetto`.`seguì` TO 'Utente';
```

```
GRANT DELETE, INSERT, SELECT, UPDATE, TRIGGER ON TABLE `progetto`.`general_user` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`creare_annuncio` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`annuncioVenduto` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`listaAnnunci` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`modifica_annuncio` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`prendi_annunciDiunVenditore` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`prendiUno_Annuncio` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`scrivi_commento` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`scrivi_messaggio` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`seguire_annuncio` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`VisualizzaCategoria` TO 'Utente';
```

```
GRANT EXECUTE ON procedure `progetto`.`VisualizzaMessaggi` TO 'Utente';
GRANT EXECUTE ON procedure `progetto`.`VisualizzaCommenti` TO 'Utente';
GRANT EXECUTE ON procedure `progetto`.`VisualizzaLeChat` TO 'Utente';
SET SQL_MODE = '';
DROP USER IF EXISTS Login;
SET
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_F
OR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
CREATE USER 'Login' IDENTIFIED BY 'login';

GRANT EXECUTE ON procedure `progetto`.`login` TO 'Login';
GRANT UPDATE, TRIGGER, SELECT, INSERT, DELETE ON TABLE `progetto`.`general_user` TO 'Login';
GRANT EXECUTE ON procedure `progetto`.`inserireUtente` TO 'Login';
SET SQL_MODE = '';
DROP USER IF EXISTS admin;
SET
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_F
OR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
CREATE USER 'admin' IDENTIFIED BY 'admin';

GRANT EXECUTE ON procedure `progetto`.`login` TO 'admin';
GRANT UPDATE, TRIGGER, SELECT, INSERT, DELETE ON TABLE `progetto`.`general_user` TO 'admin';
GRANT EXECUTE ON procedure `progetto`.`inserireUtente` TO 'admin';
GRANT TRIGGER, UPDATE, SELECT, INSERT, DELETE ON TABLE `progetto`.`Categoria` TO 'admin';
GRANT EXECUTE ON procedure `progetto`.`creareCategoria` TO 'admin';
GRANT EXECUTE ON procedure `progetto`.`VisualizzaCategoria` TO 'admin';
GRANT EXECUTE ON procedure `progetto`.`Report` TO 'admin';

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

-- Data for table `progetto`.`general_user`

START TRANSACTION;

USE `progetto`;

INSERT INTO `progetto`.`general_user` (`Username`, `Dati_anagrafici`, `ind_fatturazione`, `ind_domicilio`,
`privilegio`, `password`, `telefono`, `cellulare`, `Email`) VALUES ('test1', '2002-08-08', '11111', '11111',
'admin', 'test', NULL, NULL, NULL);

INSERT INTO `progetto`.`general_user` (`Username`, `Dati_anagrafici`, `ind_fatturazione`, `ind_domicilio`,
`privilegio`, `password`, `telefono`, `cellulare`, `Email`) VALUES ('test2', '2002-08-07', '11112', '11112',
'utente', 'test1', NULL, NULL, NULL);

INSERT INTO `progetto`.`general_user` (`Username`, `Dati_anagrafici`, `ind_fatturazione`, `ind_domicilio`,
`privilegio`, `password`, `telefono`, `cellulare`, `Email`) VALUES ('test3', '2002-08-06', '11113', '11113',
'utente', 'test2', NULL, NULL, NULL);

COMMIT;

-- Data for table `progetto`.`Categoria`

START TRANSACTION;

USE `progetto`;

INSERT INTO `progetto`.`Categoria` (`IDCategoria`, `Nome`, `UsernameC`) VALUES (1, 'vestiti', 'test1');

INSERT INTO `progetto`.`Categoria` (`IDCategoria`, `Nome`, `UsernameC`) VALUES (2, 'auto', 'test1');

INSERT INTO `progetto`.`Categoria` (`IDCategoria`, `Nome`, `UsernameC`) VALUES (3, 'moto', 'test1');

COMMIT;

-- Data for table `progetto`.`Annuncio`

START TRANSACTION;

USE `progetto`;

```
INSERT INTO `progetto`.`Annuncio` (`IDAnnuncio`, `titolo`, `prezzo`, `oggetto`, `descrizione`, `stato`,  
`UsernameA`, `IDA`) VALUES (1, 'value', 24, 'vestiti', 'belli', 'online', 'test2', 1);
```

```
COMMIT;
```