

Assignment 2 Bruce Norman

1)

```
DROP PROCEDURE IF EXISTS tennis.avgPenalties;
```

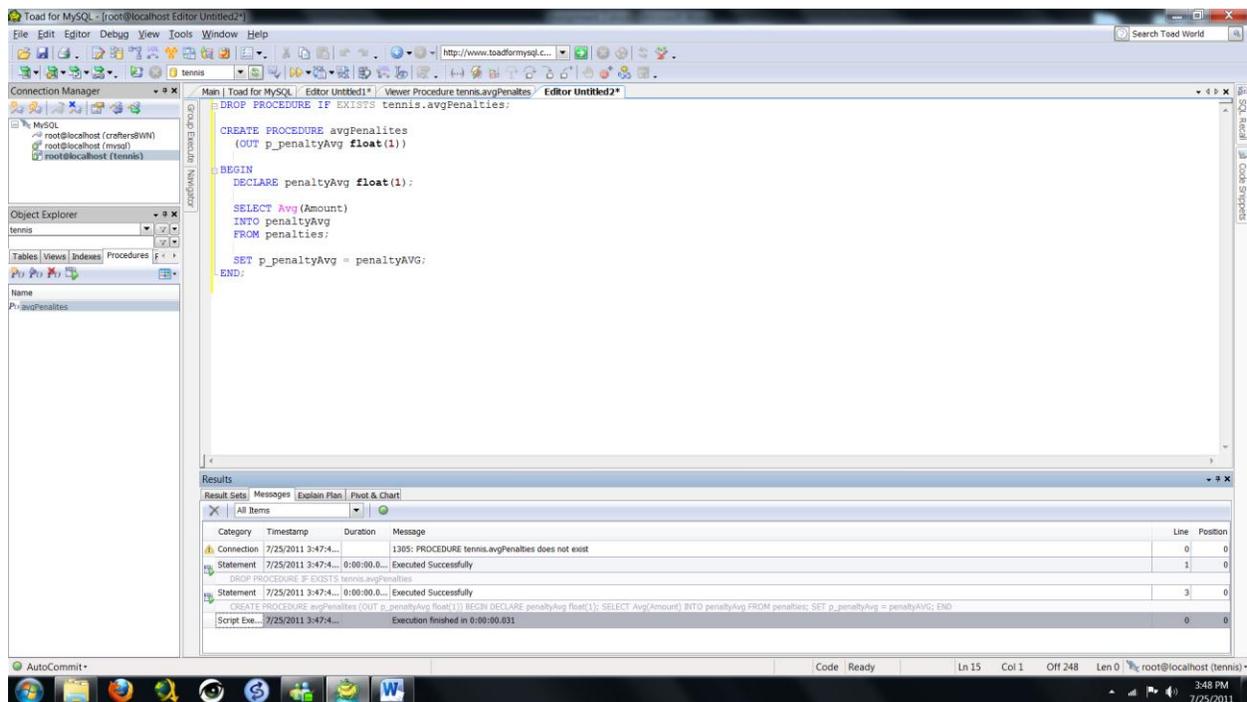
```
CREATE PROCEDURE avgPenalites  
  (OUT p_penaltyAvg float(1))
```

```
BEGIN
```

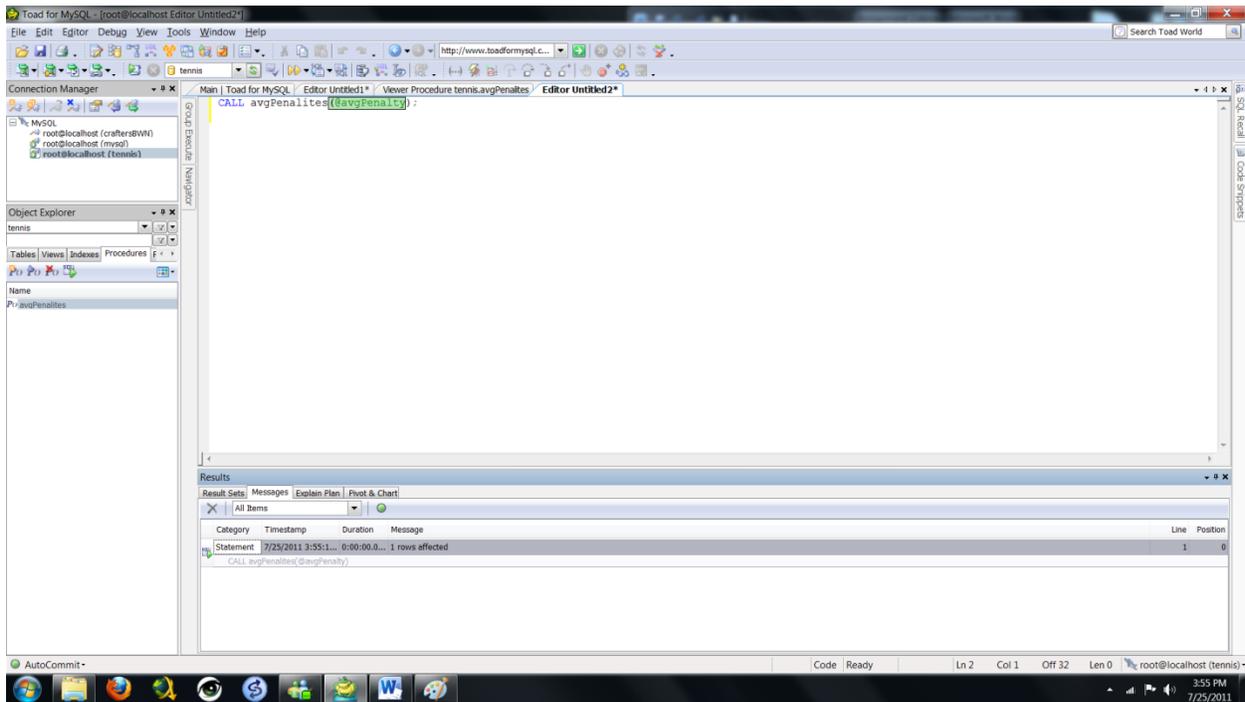
```
  DECLARE penaltyAvg float(1);
```

```
  SELECT Avg(Amount)  
  INTO penaltyAvg  
  FROM penalties;
```

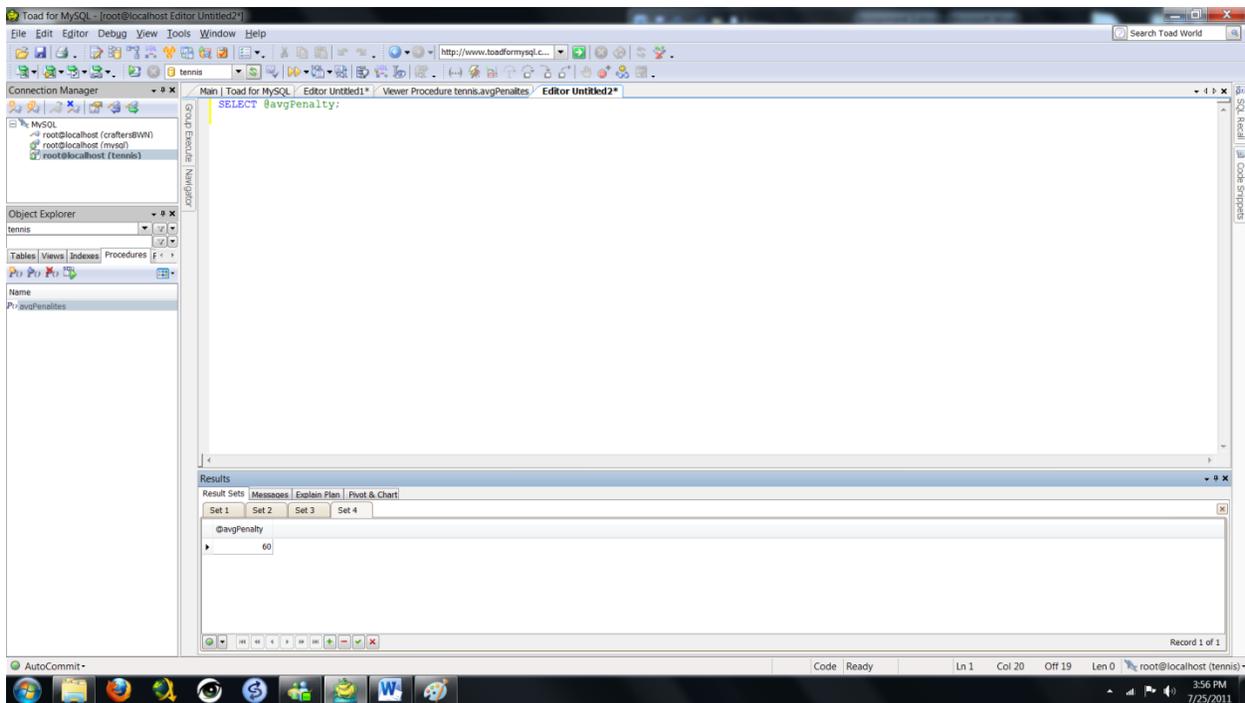
```
  SET p_penaltyAvg = penaltyAVG;  
END;
```



```
CALL avgPenalties(@avgPenalty);
```



```
SELECT @avgPenalty;
```



2)

```
DROP PROCEDURE IF EXISTS tennis.penalties;
```

```
CREATE PROCEDURE penalties
  (OUT p_penaltySm float(1),
  OUT p_penaltyLg float(1),
  OUT p_penaltyAvg float(1))

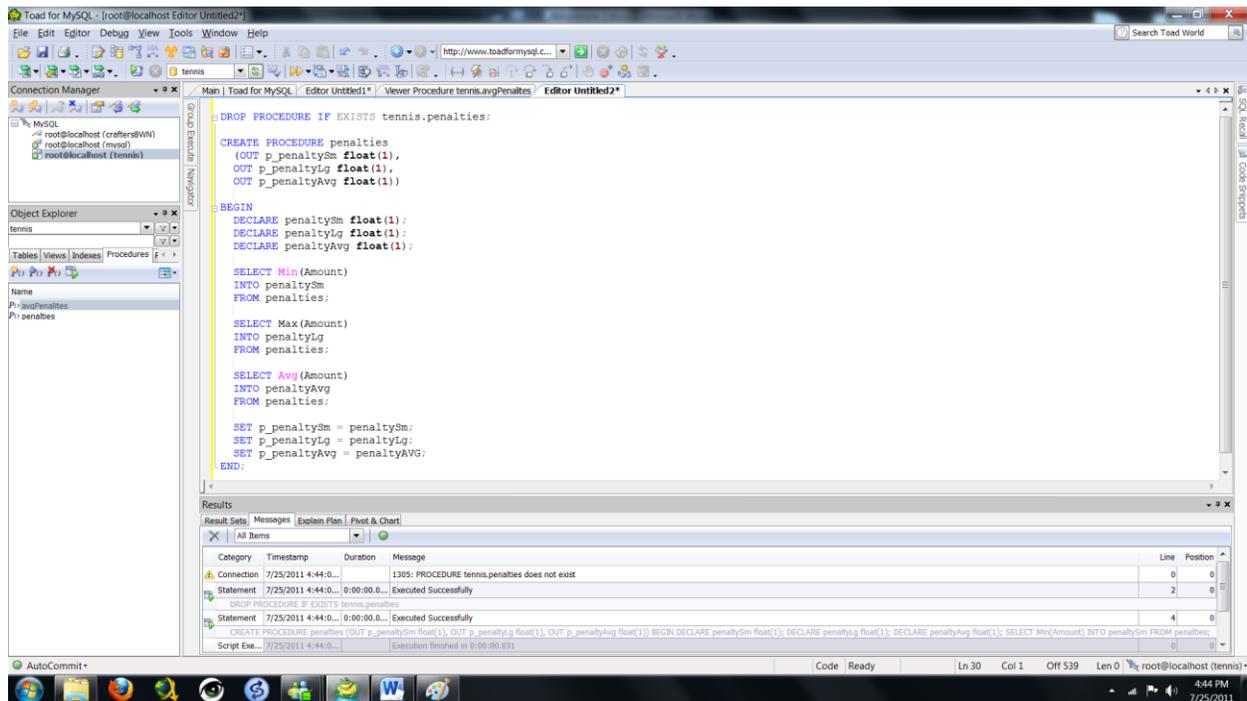
BEGIN
  DECLARE penaltySm float(1);
  DECLARE penaltyLg float(1);
  DECLARE penaltyAvg float(1);

  SELECT Min(Amount)
  INTO penaltySm
  FROM penalties;

  SELECT Max(Amount)
  INTO penaltyLg
  FROM penalties;

  SELECT Avg(Amount)
  INTO penaltyAvg
  FROM penalties;

  SET p_penaltySm = penaltySm;
  SET p_penaltyLg = penaltyLg;
  SET p_penaltyAvg = penaltyAvg;
END;
```



The screenshot displays the Toad for MySQL interface. The main editor window shows the following SQL script:

```
DROP PROCEDURE IF EXISTS tennis.penalties;

CREATE PROCEDURE penalties
  (OUT p_penaltySm float(1),
  OUT p_penaltyLg float(1),
  OUT p_penaltyAvg float(1))

BEGIN
  DECLARE penaltySm float(1);
  DECLARE penaltyLg float(1);
  DECLARE penaltyAvg float(1);

  SELECT Min(Amount)
  INTO penaltySm
  FROM penalties;

  SELECT Max(Amount)
  INTO penaltyLg
  FROM penalties;

  SELECT Avg(Amount)
  INTO penaltyAvg
  FROM penalties;

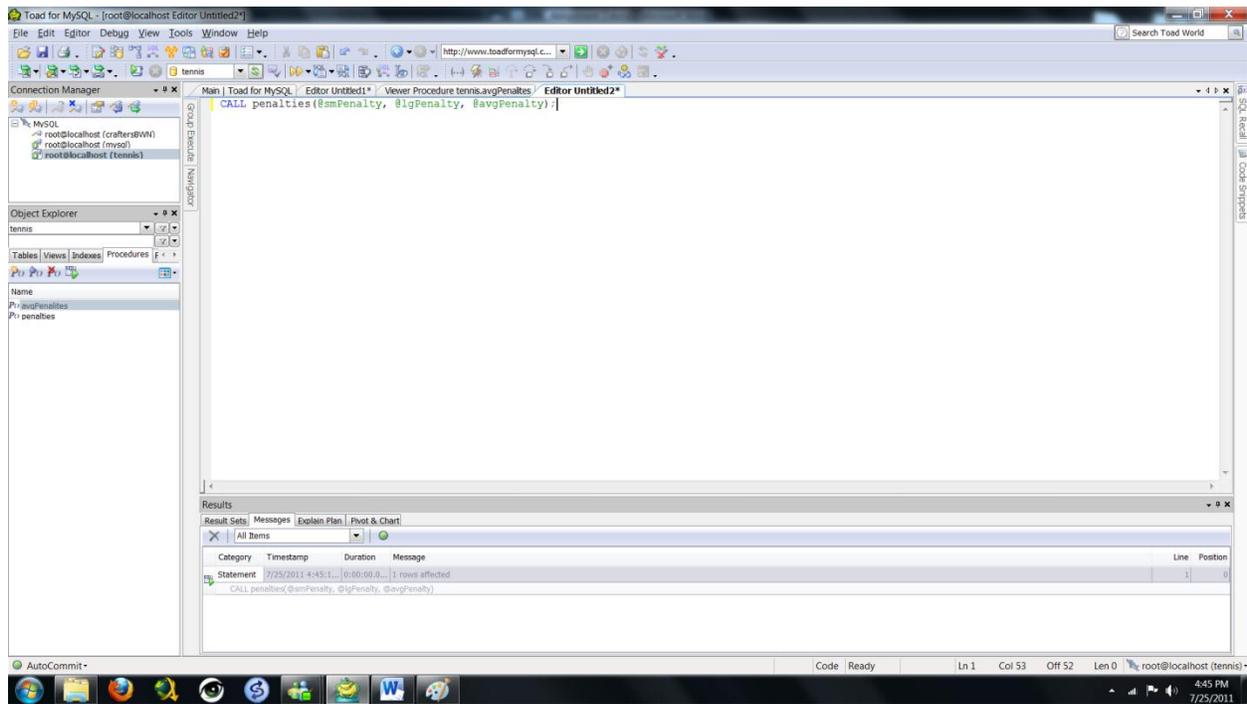
  SET p_penaltySm = penaltySm;
  SET p_penaltyLg = penaltyLg;
  SET p_penaltyAvg = penaltyAvg;
END;
```

The Results pane at the bottom shows the execution log:

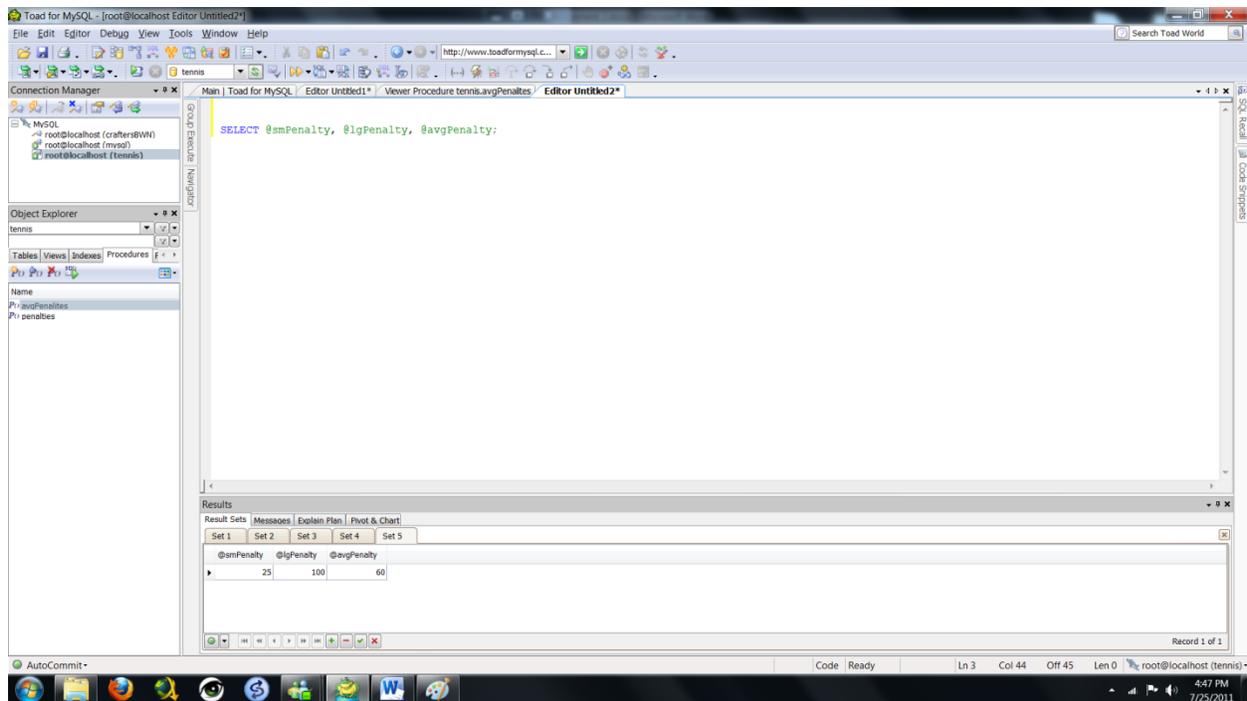
Category	Timestamp	Duration	Message	Line	Position
Connection	7/25/2011 4:44:0...		1305: PROCEDURE tennis.penalties does not exist	0	0
Statement	7/25/2011 4:44:0...	0:00:00.0...	Executed Successfully	2	0
Statement	7/25/2011 4:44:0...	0:00:00.0...	Executed Successfully	4	0
Script Exe...	7/25/2011 4:44:0...		Execution finished in 0:00:00.031	0	0

The status bar at the bottom indicates the current position is Ln 30, Col 1, Off 539, and the user is root@localhost (tennis).

```
CALL penalties(@smPenalty, @lgPenalty, @avgPenalty);
```



```
SELECT @smPenalty, @lgPenalty, @avgPenalty;
```



3)

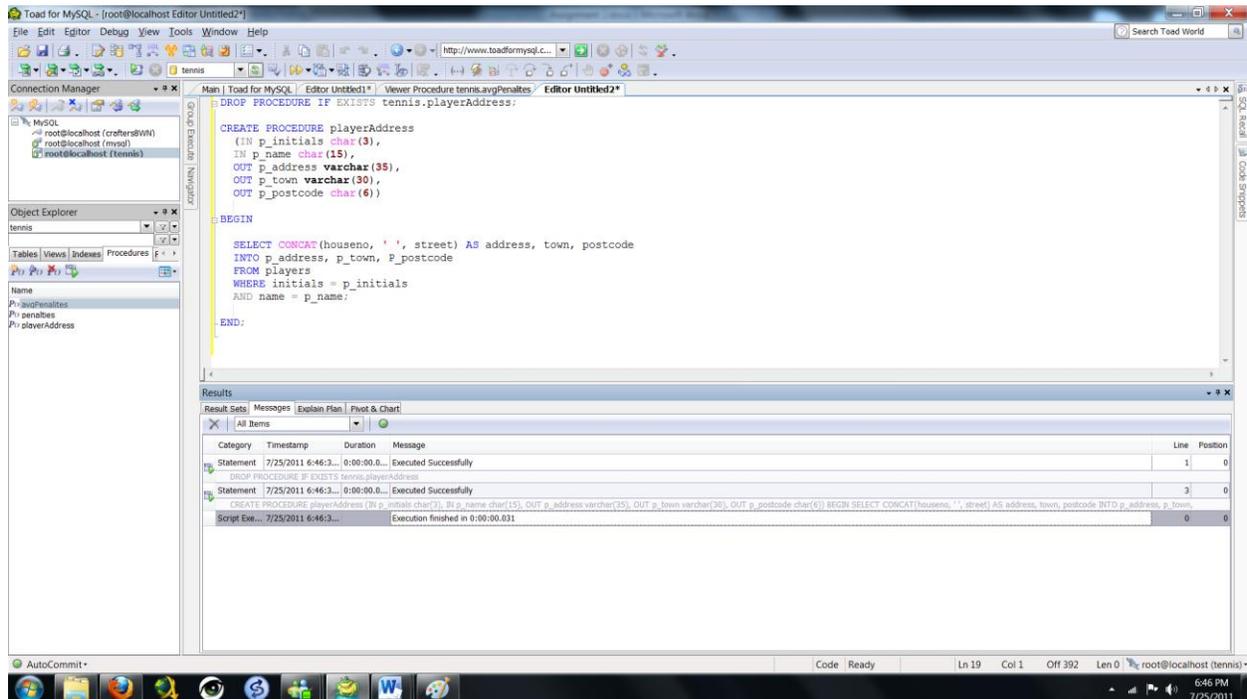
```
DROP PROCEDURE IF EXISTS tennis.playerAddress;
```

```
CREATE PROCEDURE playerAddress  
  (IN p_initials char(3),  
   IN p_name char(15),  
   OUT p_address varchar(35),  
   OUT p_town varchar(30),  
   OUT p_postcode char(6))
```

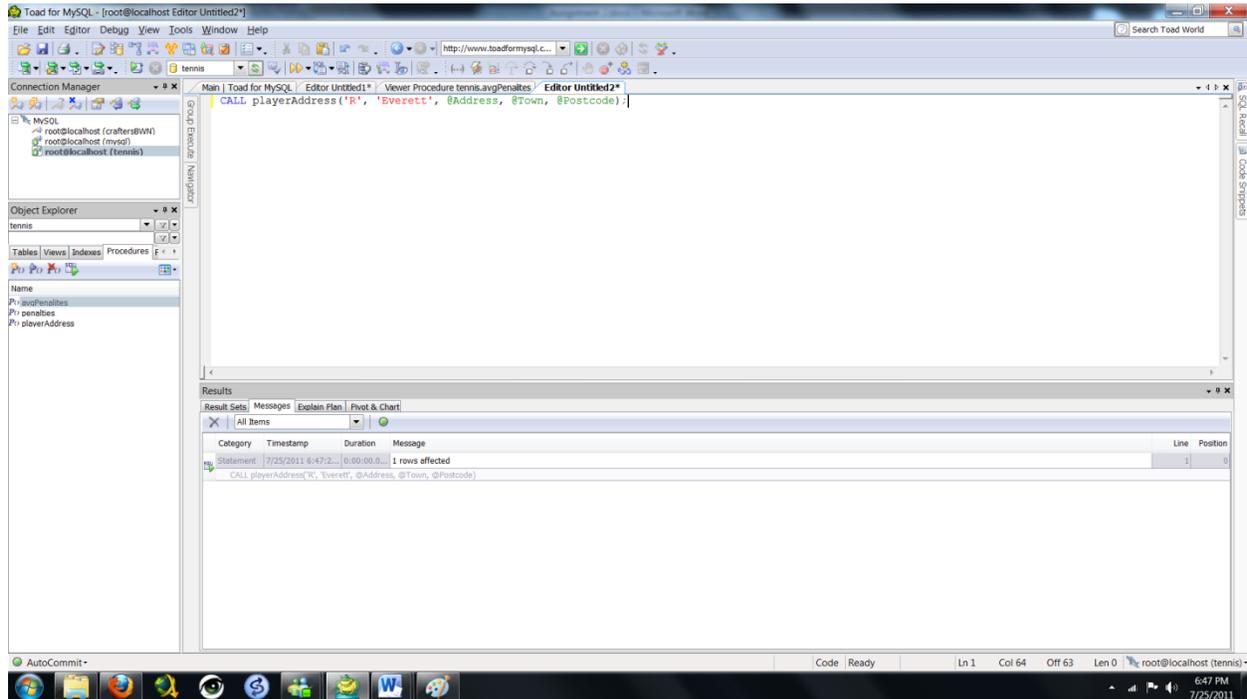
```
BEGIN
```

```
  SELECT CONCAT(houseno, ' ', street) AS address, town, postcode  
  INTO p_address, p_town, P_postcode  
  FROM players  
  WHERE initials = p_initials  
  AND name = p_name;
```

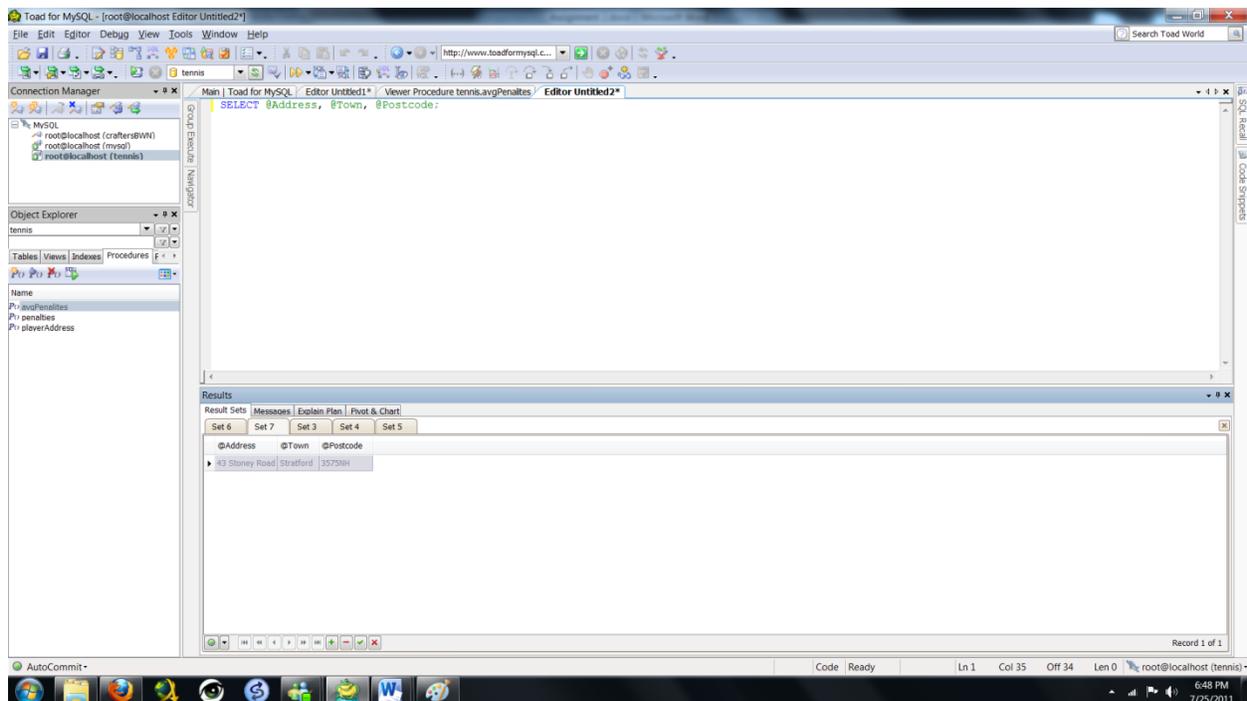
```
END;
```



```
CALL playerAddress('R', 'Everett', @Address, @Town, @Postcode);
```



```
SELECT @Address, @Town, @Postcode;
```



4)

```
DROP PROCEDURE IF EXISTS tennis.newPlayer;
```

```
CREATE PROCEDURE newPlayer
```

```
(IN P_PLAYERNO INTEGER,  
IN P_NAME CHAR(15),  
IN P_INITIALS CHAR(3),  
IN P_BIRTH_DATE DATE,  
IN P_SEX CHAR(1),  
IN P_JOINED SMALLINT,  
IN P_STREET VARCHAR(30),  
IN P_HOUSENO CHAR(4),  
IN P_POSTCODE CHAR(6),  
IN P_TOWN VARCHAR(30),  
IN P_PHONENO CHAR(13),  
IN P_LEAGUENO CHAR(4),  
OUT P_STATUS VARCHAR(255))
```

```
BEGIN
```

```
DECLARE STATUS VARCHAR(255);  
DECLARE DUP_KEY INTEGER DEFAULT 0;  
DECLARE CONTINUE HANDLER FOR 1062
```

```
SET DUP_KEY = 1;
```

```
INSERT INTO PLAYERS(P_PLAYERNO, NAME, INITIALS, BIRTH_DATE, SEX, JOINED,  
STREET, HOUSENO, POSTCODE, TOWN, PHONENO, LEAGUENO)
```

```
VALUES(P_PLAYERNO, P_NAME, P_INITIALS, P_BIRTH_DATE, P_SEX, P_JOINED,  
P_STREET, P_HOUSENO, P_POSTCODE, P_TOWN, P_PHONENO, P_LEAGUENO);
```

```
IF(DUP_KEY = 1) THEN
```

```
SET P_STATUS = "Duplicate primary key/player number";
```

```
ELSE
```

```
SET P_STATUS = "Record inserted";
```

```
END IF;
```

```
END;
```

The screenshot shows the Toad for MySQL interface. The main editor window contains the following SQL code:

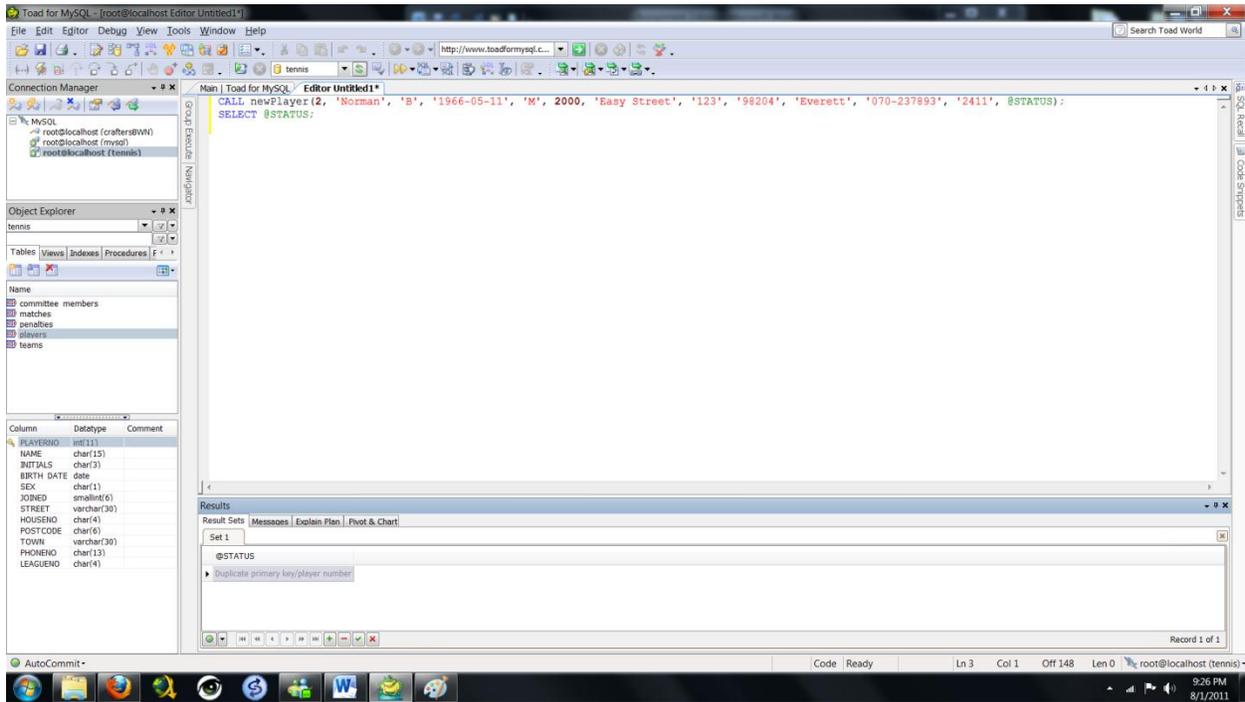
```
DROP PROCEDURE IF EXISTS tennis.newPlayer;  
  
CREATE PROCEDURE newPlayer  
(IN P_PLAYERNO INTEGER,  
IN P_NAME CHAR(15),  
IN P_INITIALS CHAR(3),  
IN P_BIRTH_DATE DATE,  
IN P_SEX CHAR(1),  
IN P_JOINED SMALLINT,  
IN P_STREET VARCHAR(30),  
IN P_HOUSENO CHAR(4),  
IN P_POSTCODE CHAR(6),  
IN P_TOWN VARCHAR(30),  
IN P_PHONENO CHAR(13),  
IN P_LEAGUENO CHAR(4),  
OUT P_STATUS VARCHAR(255))  
  
BEGIN  
  DECLARE STATUS VARCHAR(255);  
  DECLARE DUP_KEY INTEGER DEFAULT 0;  
  DECLARE CONTINUE HANDLER FOR 1062  
  SET DUP_KEY = 1;  
  INSERT INTO PLAYERS(P_PLAYERNO, NAME, INITIALS, BIRTH_DATE, SEX, JOINED, STREET, HOUSENO, POSTCODE, TOWN, PHONENO, LEAGUENO)  
  VALUES(P_PLAYERNO, P_NAME, P_INITIALS, P_BIRTH_DATE, P_SEX, P_JOINED, P_STREET, P_HOUSENO, P_POSTCODE, P_TOWN, P_PHONENO, P_LEAGUENO);  
  IF(DUP_KEY = 1) THEN  
    SET P_STATUS = "Duplicate primary key/player number";  
  ELSE  
    SET P_STATUS = "Record inserted";  
  END IF;  
END;
```

The Results pane at the bottom shows the following execution messages:

Category	Timestamp	Duration	Message	Line	Position
Statement	8/1/2011 9:19:20...	0:00:00.0...	Executed Successfully	1	0
Statement	8/1/2011 9:19:20...	0:00:00.0...	Executed Successfully	3	0

The Object Explorer on the left shows the 'tennis' database with tables like 'committee_members', 'matches', 'players', and 'teams'.

```
CALL newPlayer(2, 'Norman', 'B', '1966-05-11', 'M', 2000, 'Easy Street',
'123', '98204', 'Everett', '070-237893', '2411', @STATUS);
SELECT @STATUS;
```



```
CALL newPlayer(3, 'Norman', 'B', '1966-05-11', 'M', 2000, 'Easy Street',
'123', '98204', 'Everett', '070-237893', '2411', @STATUS);
SELECT @STATUS;
```

