****

**Athlone Institute of Technology**



**COURSE:** Higher Diploma In Computing

**COURSE ID:** Al\_Kcomp\_G

**SUBJECT:** Database Applications

**YEAR:** 5

**LECTURER**: Mr. Noel Tierney

**SUBJECT:** PL/SQL

**PROJECT TITLE:** Farm Rental Agreements

|  |  |  |
| --- | --- | --- |
| Student Number | Student Name | Signature (for hardcopy) |
| A00037189 | Brendan Kenny |  |

***Contents***

[***Introduction*** 3](#_Toc419494428)

[***Entity Relationship Diagrams*** 4](#_Toc419494429)

[***Entity Relationship Diagram (ERD) Overview*** 4](#_Toc419494430)

[***Extended ERD with Table Column Names:*** 4](#_Toc419494431)

[***Q.1 Before Insert Trigger*** 5](#_Toc419494432)

[***Q.2 Before Update Trigger*** 9](#_Toc419494433)

[***Q.3 After Insert Trigger*** 14](#_Toc419494434)

[***Q.4 After Update Trigger*** 23](#_Toc419494435)

[***Q.5 Before Insert, Update, Delete Trigger*** 27](#_Toc419494436)

[***Q.6 After Insert Update Delete On Rental Contract*** 31](#_Toc419494437)

[***Q.7 Before Insert/Update/Delete Table Level Trigger*** 39](#_Toc419494438)

[***Q.8 Procedure without Cursor*** 45](#_Toc419494439)

[***Q.9 Procedure With Cursor*** 48](#_Toc419494440)

[***Q.10 Procedure with Cursor to Retrieve, Display and Delete*** 52](#_Toc419494441)

[***Q.11 Package with Three Procedures*** 59](#_Toc419494442)

[***Appendix*** 69](#_Toc419494443)

[***Q.17 Package with Insert, Update And Delete Procedures*** 69](#_Toc419494444)

[***Q.19 Trigger That Uses a Cursor*** 72](#_Toc419494445)

[***Q.18 Trigger that Calls a Procedure*** 75](#_Toc419494446)

[***Q.14 Table Level Trigger Before Delete*** 79](#_Toc419494447)

## ***Introduction***

The following database was created to facilitate farm rental agreements between the relevant parties that would be party to such agreements, namely Farmers, Landlords, Auctioneers and Solicitors. The resulting agreements are recorded in the Rental Contract table and relate to the relevant Farms as recorded in the Farm table. The relationships between the relevant tables are displayed in the ERD diagrams and were an aid to me in the creation of the database, when creating the user constraints and in the subsequent queries, procedures, triggers and packages in later stages of my project.

The following pages will illustrate the various Triggers, Procedures and Packages that are used in PL/SQL.

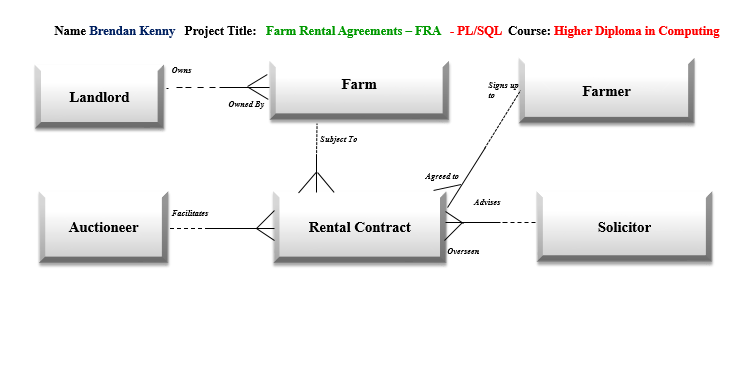
Brendan Kenny

29 March 2015

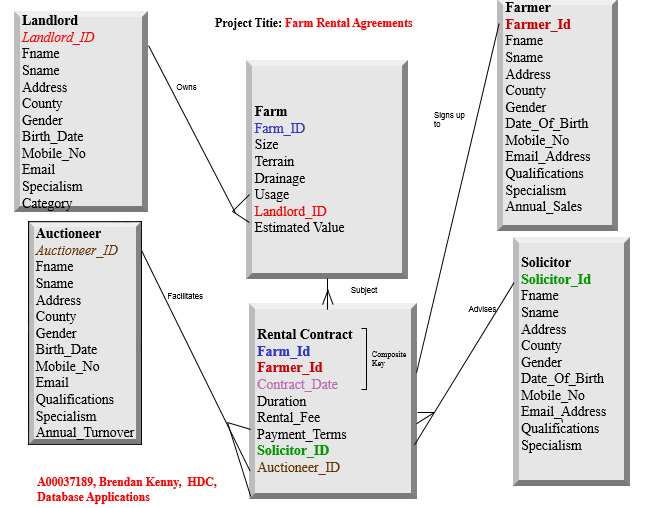


## ***Entity Relationship Diagrams***

### ***Entity Relationship Diagram (ERD) Overview***



### ***Extended ERD with Table Column Names:***



## ***Q.1 Before Insert Trigger***

***A Rental Contract can be added to the Database provided certain conditions are met. The details are :***

***(a)*** ***New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments.***

***(b)*** ***New Contracts can only be added to this table Solicitors who are from counties other than Longford or Westmeath and are party to contracts of not less than six months duration.***

***(c)*** ***New Contracts can only be added to this table for Auctioneers Solicitors whose specialism is either Housing or Retail Units and are party to Contracts of not less than €5000.***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Drop Trigger** **Q1BeforeInsert;**

**Set** **Serveroutput** **On** **Size** 1000000;

**Create** **or** **Replace** **Trigger Q1BeforeInsert**

**Before** **Insert** **On** **Rental\_Contract**

**For** **Each** **Row**

**Declare**

**V\_Qualifications** **Farmer.Qualifications%Type;**

**V\_County** **Solicitor.County%Type;**

**V\_Specialism** **Auctioneer**.**Specialism%Type;**

**Begin**

**SELECT** **Qualifications** into **V\_Qualifications**

**FROM** **Farmer**

**WHERE** **Farmer.Farmer\_Id** **=** **:new.Farmer\_Id**;

**SELECT** **County** into **V\_County**

**FROM** **Solicitor**

**WHERE** **Solicitor.Solicitor\_Id** **=** **:new.Solicitor\_Id**;

**SELECT** **Specialism** into **V\_Specialism**

**FROM** **Auctioneer**

**WHERE** **Auctioneer.Auctioneer\_Id** **=** **:new.Auctioneer\_Id;**

**If:new.Payment\_Terms** **Not In** **(‘Monthly Instalments’**)**And** **V\_Qualifications** =‘**Farming** **Cert’ Then**

**Raise\_Application\_Error(**-20101**,** *'****New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments’*);**

**End If;**

**If** **:new.Duration** < 6 **And** **V\_County** **In** **(‘Longford’, ‘Westmeath’**) **Then**

**Raise\_Application\_Error(**-20101**,** ***'New Contracts can only be added to this table Solicitors who are from counties other than Longford or Westmeath and are party to contracts of not less than six months duration’*);**

**End If;**

**If** **:new.Rental\_Fee** < 5000 **And** **V\_Specialism** **In** **(‘Housing’, ‘Retail Units’) Then**

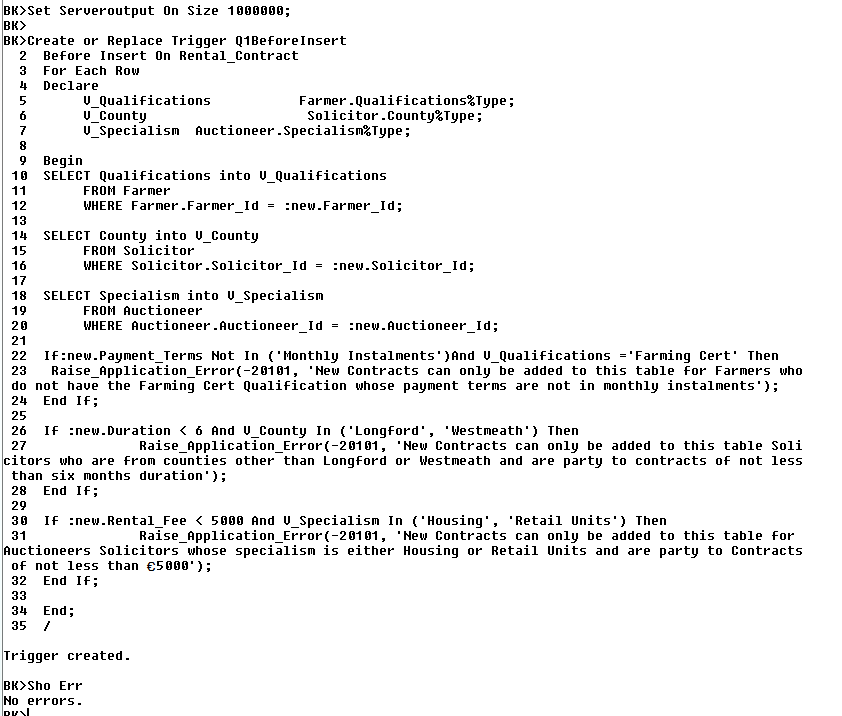
**Raise\_Application\_Error(-20101,** ***'New Contracts can only be added to this table for Auctioneers Solicitors whose specialism is either Housing or Retail Units and are party to Contracts of not less than €5000'*);**

**End If;**

**End;**

**/**

**Sho Err**



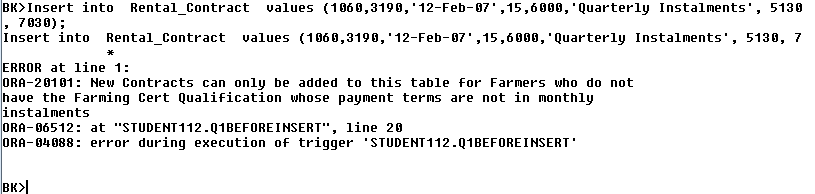
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

***(a)*** ***New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments.***

**Invalid Insert:**

Insert into Rental\_Contract values (1060,**3190**,'12-Feb-07',15,6000,'Quarterly Instalments', 5130, 7030);

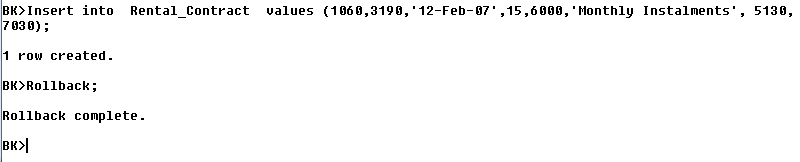
Rollback;



**Valid Insert:**

Insert into Rental\_Contract values (1060,**3190**,'12-Feb-07',15,6000,**'Monthly Instalments**', 5130, 7030);

Rollback;



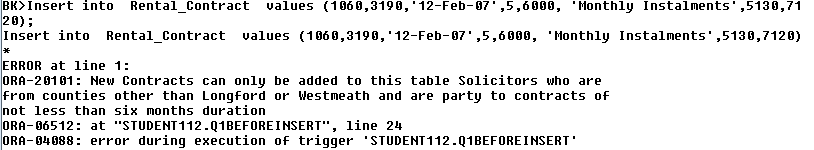
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

***(b)*** ***New Contracts can only be added to this table Solicitors who are from counties other than Longford or Westmeath and are party to contracts of not less than six months duration.***

**Invalid Insert:**

**Insert into Rental\_Contract values (1060,3190,'12-Feb-07',5,6000, 'Monthly Instalments',5130,7120);**

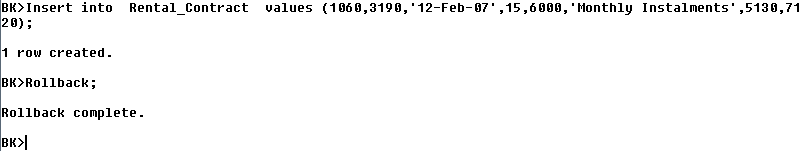
**Rollback;**



**Valid Insert:**

**Insert into Rental\_Contract values (1060,3190,'12-Feb-07',15,6000,'Monthly Instalments',5130,7120);**

**Rollback;**

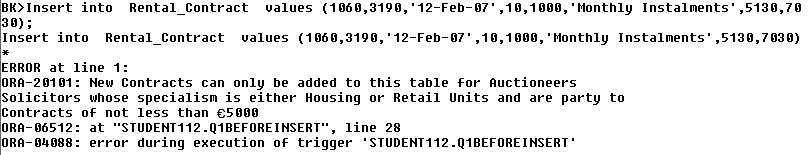


**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**(c)*** ***New Contracts can only be added to this table for Auctioneers Solicitors whose specialism is either Housing or Retail Units and are party to Contracts of not less than €5000.***

**Invalid Insert:**

**Insert into Rental\_Contract values (1060,3190,'12-Feb-07',10,1000,'Monthly Instalments',5130,7030);**

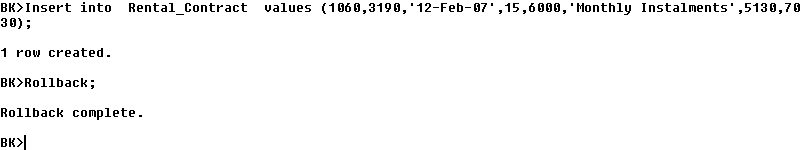
**Rollback;**



**Valid Insert:**

**Insert into Rental\_Contract values (1060,3190,'12-Feb-07',15,6000,'Monthly Instalments',5130,7030);**

**Rollback;**



**Drop Trigger Q1BeforeInsert;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Q.2 Before Update Trigger***

***Before an Update can be made to the Farm table, the following conditions must be met :***

***(a)* *Farms belonging to Longford farmers may not be increased by less than €10000 in value.***

***(b)* *Only Farms that have been rented out more than once may be updated by amounts in excess of €10000***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Drop Trigger** **Farm\_Est\_Value;**

**Set** **Serveroutput On Size** 1000000**;**

**Create** **or** **Replace** **Trigger** **Farm\_Est\_Value**

**Before** **Update** **On** **Farm**

**For** **Each** **Row**

**Declare**

**V\_Count\_Farm** **Number(2);**

**V\_County\_Landlord** **Landlord.County%Type**;

**Begin**

**Select** **NVL(Count(\*),0)** into **V\_Count\_Farm**

**From** **Rental\_Contract**

**Where** **Rental\_Contract.Farm\_Id** = :**new**.**Farm\_Id;**

**Select** **Landlord.County** into **V\_County\_Landlord**

**From** **Landlord**

**Where** **Landlord.Landlord\_Id**=:**new**.**Landlord\_Id;**

**If**(:**New**.**Estimated\_Value**-:**Old**.**Estimated\_Value**)<10000 **And V\_County\_Landlord** = ‘**Longford**’**Then**

**Raise\_Application\_Error**(**-20101**, **‘Farms belonging to Longford Landlords may not be increased by less than €10000 in value’**);

**End If;**

**If(:New**.**Estimated\_Value** -:**Old**.**Estimated\_Value**)>10000 **And** (**V\_Count\_Farm**<2) **Then**

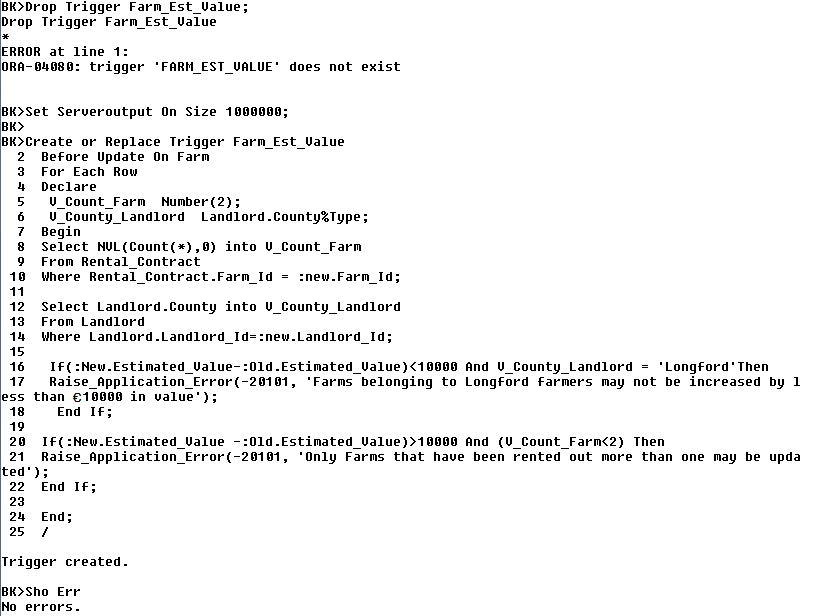
**Raise\_Application\_Error**(**-20101**, **‘Only Farms that have been rented out more than once may be updated’);**

**End If;**

**End;**

**/**

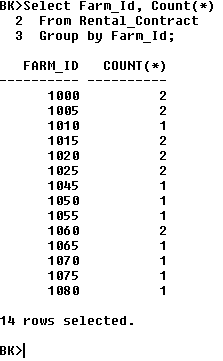
**Sho Err**



**Select Farm\_Id, Count(\*)**

**From Rental\_Contract**

**Group by Farm\_Id;**



***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

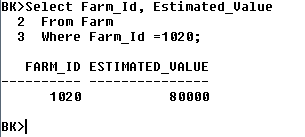
1. ***Farms belonging to Longford farmers may not be increased by less than €10000 in value.***

***This Update will fail because the new update is less than €10000***

**Select Farm\_Id, Estimated\_Value**

**From Farm**

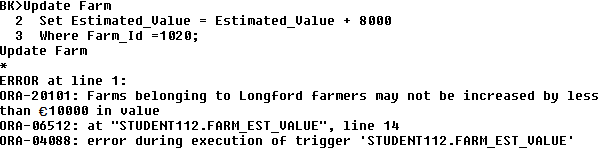
**Where Farm\_Id =1020;**



**Update Farm**

**Set Estimated\_Value = Estimated\_Value + 8000**

**Where Farm\_Id =1020;**

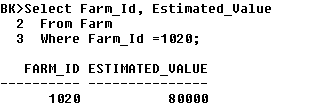


***This Update will suceed because the new update is in excess of €10000.***

**Select Farm\_Id, Estimated\_Value**

**From Farm**

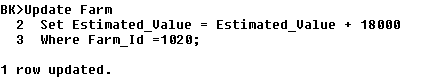
**Where Farm\_Id =1020;**



**Update Farm**

**Set Estimated\_Value = Estimated\_Value + 18000**

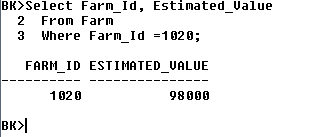
**Where Farm\_Id =1020;**



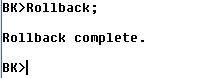
**Select Farm\_Id, Estimated\_Value**

**From Farm**

**Where Farm\_Id =1020;**



**Rollback;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

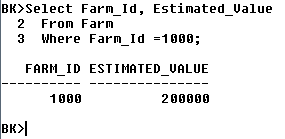
1. ***Only Farms that have been rented out more than once may be updated by more than €10000.***

***This Update will fail because the update is less than €10000.***

**Select Farm\_Id, Estimated\_Value**

**From Farm**

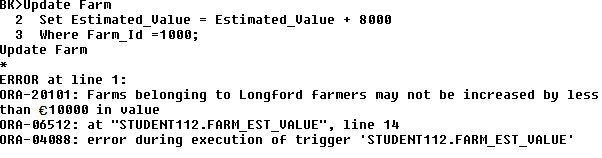
**Where Farm\_Id =1000;**



**Update Farm**

**Set Estimated\_Value = Estimated\_Value + 8000**

**Where Farm\_Id =1000;**

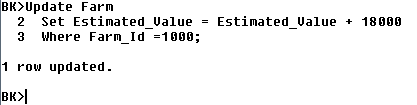


***This Update will Succeed because the update is greater than €10000.***

**Update Farm**

**Set Estimated\_Value = Estimated\_Value + 18000**

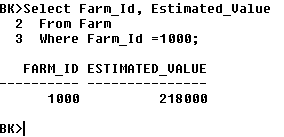
**Where Farm\_Id =1000;**

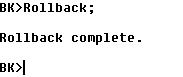


**Select Farm\_Id, Estimated\_Value**

**From Farm**

**Where Farm\_Id =1000;**





***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

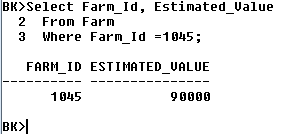
1. ***Only Farms that have been rented out more than once may be updated by more than €10000.***

***This update will succeed because the farm in question has been rented out less than twice and the amount of the update is less than €10000***

**Select Farm\_Id, Estimated\_Value**

**From Farm**

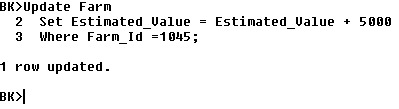
**Where Farm\_Id =1045;**



**Update Farm**

**Set Estimated\_Value = Estimated\_Value + 5000**

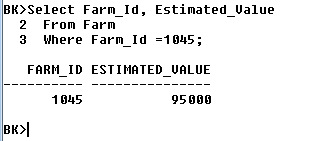
**Where Farm\_Id =1045;**



**Select Farm\_Id, Estimated\_Value**

**From Farm**

**Where Farm\_Id =1045;**

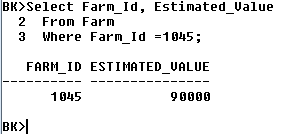


***This update will fail because the farm in question has been rented out less than twice and the amount of the update is greater than €10000***

**Select Farm\_Id, Estimated\_Value**

**From Farm**

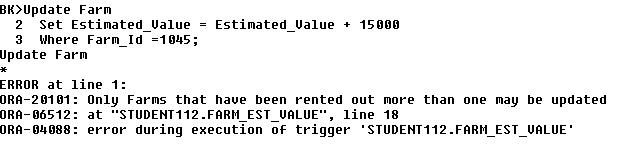
**Where Farm\_Id =1045;**



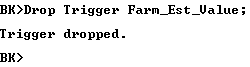
**Update Farm**

**Set Estimated\_Value = Estimated\_Value + 15000**

**Where Farm\_Id =1045;**



**Drop Trigger Farm\_Est\_Value;**



## ***Q.3 After Insert Trigger***

***After a successful insert has been made to the Rental Contract table, the following changes to the Database should be made :***

***1.***

***(a)* *A new Rental Contract of a duration of between 6 and 12 months should result in an increase of €2000 in the associated farms estimated value if that farm is used for tillage.***

***(b)* *A new Rental Contract of a duration of more than 12 months should result in an increase of €1500 in the associated farms estimated value if that farm is used for grazing.***

***2.***

***(a)* *A new Rental Contract with a rental fee in excess of €5000 will result in an increase in annual turnover of €800 for the associated Auctioneer provided that Auctioneer specialises in Land.***

***(b)* *A new Rental Contract with a Rental Fee less than €5000 will result in an increase in annual turnover of €500, if the associated Auctioneer specialises in anything other than land.***

***3.***

***(a)* *A new Rental Contract with Payment Terms in ‘Monthly Instalments’ will result in an increase in the associated Farmer’s Annual Sales of €800, provided the farmer in question is from Dublin.***

***(b)* *A new Rental Contract with Payment Terms in ‘Quarterly Instalments’ will result in an increase in the associated Farmer’s Annual Sales of €200 if the Farmer in question is from Louth.***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Set** **Serveroutput** **On** **Size** 1000000**;**

**Drop Table New\_Contracts\_Log;**

**Create Table New\_Contracts\_Log** (User\_Involved Varchar2 (10), Date\_Added Date, Farm\_No Number (4), Duration Number (7));

**Create or Replace** Trigger **New\_Contracts**

**After Insert** on **Rental\_Contract**

**For Each Row**

Declare

**V\_Usage** **Farm**.**Usage%Type;**

**V\_Specialism Auctioneer.Specialism%Type;**

**V\_County Farmer.County%Type;**

**Begin**

**Insert** into **New\_Contracts\_Log** Values (User, Sysdate,:New.Farm\_Id,:New.Duration);

**Select** **Usage** Into **V\_Usage**

**From** **Farm**

**Where** **Farm.Farm\_Id** = **:New.Farm\_Id;**

**Select** **Specialism** Into **V\_Specialism**

**From** **Auctioneer**

**Where** **Auctioneer.Auctioneer\_Id** = **:New.Auctioneer\_Id;**

**Select** **County** Into **V\_County**

**From** **Farmer**

**Where** **Farmer.Farmer\_Id** = **:New.Farmer\_Id;**

**If :New**.**Duration Between 6 And 12** **And** (**V\_Usage** = ‘**Tillage**’) **Then**

**Update** **Farm**

**Set Estimated\_Value** = **Estimated\_Value**+**2000**

**Where Farm\_Id** = **:New.Farm\_Id**;

**End If;**

**If** **:New.Duration** **> 12** **And** (**V\_Usage** = ‘**Grazing**’) **Then**

**Update** **Farm**

**Set Estimated\_Value** = **Estimated\_Value**+**1500**

**Where Farm\_Id** = **:New.Farm\_Id;**

**End If;**

**If** **:New.Rental\_Fee> 5000** **And** (**V\_Specialism** = ‘**Land**’) **Then**

**Update** **Auctioneer**

**Set Annual\_Turnover** = **Annual\_Turnover**+**800**

**Where Auctioneer\_Id** = **:New.Auctioneer\_Id;**

**End If;**

**If** **:New.Rental\_Fee** **< 5000** **And** (**V\_Usage** <>‘**Land**’) **Then**

**Update** **Auctioneer**

**Set Annual\_Turnover = Annual\_Turnover**+**500**

**Where Auctioneer\_Id** = **:New.Auctioneer\_Id;**

**End If;**

**If** **:New.Payment\_Terms**=**’Monthly Instalments’** **And** (**V\_County** = ‘**Dublin**’) **Then**

**Update** **Farmer**

**Set Annual\_Sales** = **Annual\_Sales**+**800**

**Where Farmer\_Id** = **:New.Farmer\_Id;**

**End If;**

**If** **:New.Payment\_Terms**= ‘**Quarterly Instalments’** **And** (**V\_County** = ‘**Louth**’) **Then**

**Update** **Farmer**

**Set Annual\_Sales** = **Annual\_Sales**+**200**

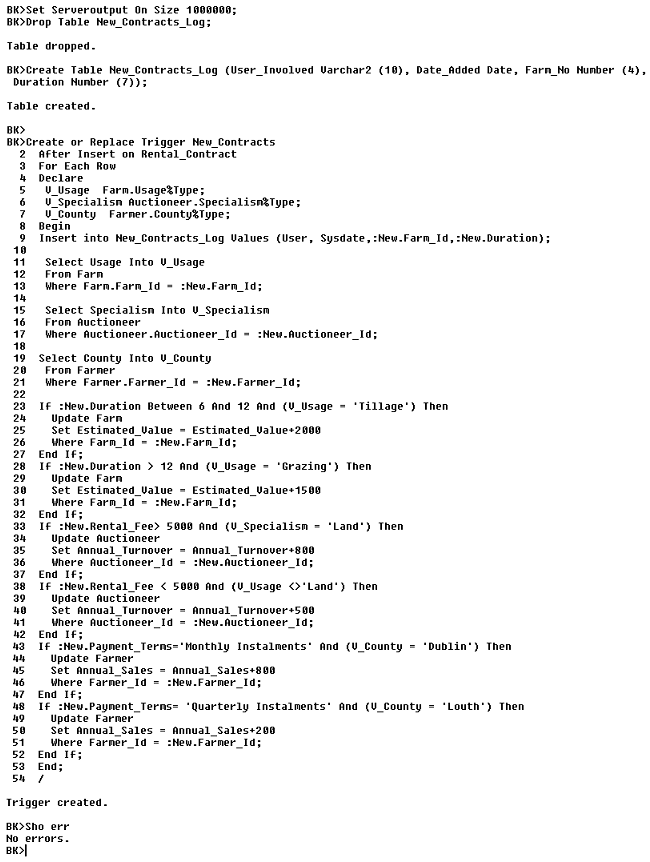
**Where Farmer\_Id** = **:New.Farmer\_Id;**

**End If;**

**End;**

/

**Sho err**

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

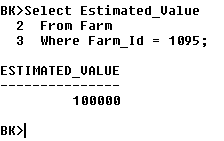
1. ***(a)* *A new Rental Contract of a duration of between 6 and 12 months should result in an increase of €2000 in the associated farms estimated value if that farm is used for tillage.***

***Before Update:***

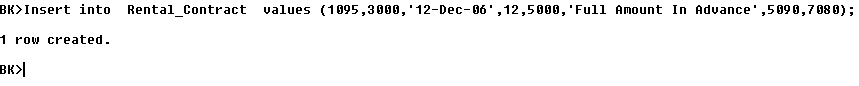
**Select Estimated\_Value**

**From Farm**

**Where Farm\_Id = 1095;**



**Insert into Rental\_Contract values (1095,3000,'12-Dec-06',12,5000,'Full Amount In Advance',5090,7080);**

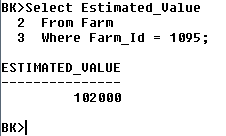


***After Update:***

**Select Estimated\_Value**

**From Farm**

**Where Farm\_Id = 1095;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

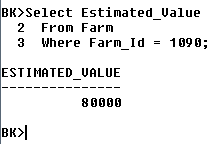
***1.* *(b)* *A new Rental Contract of a duration of more than 12 months should result in an increase of €1500 in the associated farms estimated value if that farm is used for grazing.***

***Before Update***

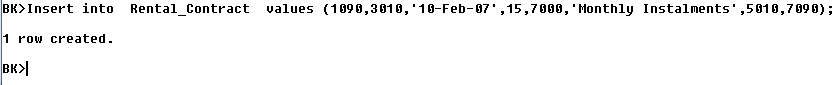
**Select Estimated\_Value**

**From Farm**

**Where Farm\_Id = 1090;**



**Insert into Rental\_Contract values (1090,3010,'10-Feb-07',15,7000,'Monthly Instalments',5010,7090);**

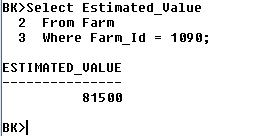


***After Update***

**Select Estimated\_Value**

**From Farm**

**Where Farm\_Id = 1090;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

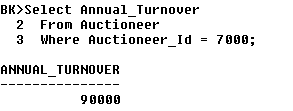
***2. (a)* *A new Rental Contract with a rental fee in excess of €5000 will result in an increase in annual turnover of €800 for the associated Auctioneer provided that Auctioneer specialises in Land.***

***Before Update:***

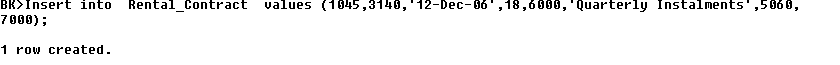
**Select Annual\_Turnover**

**From Auctioneer**

**Where Auctioneer\_Id = 7000;**



**Insert into Rental\_Contract values** (1045,3140,'12-Dec-06',18,**6000**,'Quarterly Instalments',5060,**7000**);

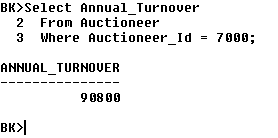


***After Update:***

**Select Annual\_Turnover**

**From Auctioneer**

**Where Auctioneer\_Id = 7000;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

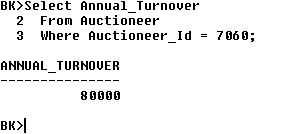
1. ***(b)* *A new Rental Contract with a Rental Fee less than €5000 will result in an increase in annual turnover of €500, if the associated Auctioneer specialises in anything other than land.***

***Before Update:***

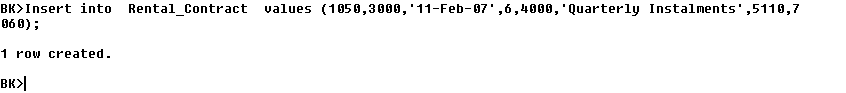
**Select Annual\_Turnover**

**From Auctioneer**

**Where Auctioneer\_Id = 7060;**



**Insert into Rental\_Contract values** (1050,3000,'11-Feb-07',6,**4000**,'Quarterly Instalments',5110,**7060**);

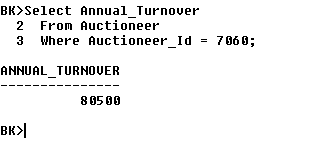


***After Update:***

**Select Annual\_Turnover**

**From Auctioneer**

**Where Auctioneer\_Id = 7060;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

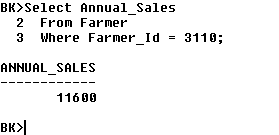
1. ***(a)* *A new Rental Contract with Payment Terms in ‘Monthly Instalments’ will result in an increase in the associated Farmer’s Annual Sales of €800, provided the farmer in question is from Dublin.***

***Before Update:***

**Select Annual\_Sales**

**From Farmer**

**Where Farmer\_Id = 3110;**



**Insert into Rental\_Contract values** (1045,**3110**,'12-Dec-14',18,6000,**'Monthly Instalments'**,5060,7120);

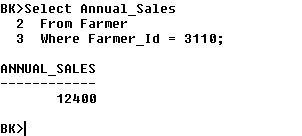


***After Update:***

**Select Annual\_Sales**

**From Farmer**

**Where Farmer\_Id = 3110;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

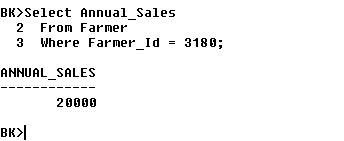
1. ***(b)* *A new Rental Contract with Payment Terms in ‘Quarterly Instalments’ will result in an increase in the associated Farmer’s Annual Sales of €200 if the Farmer in question is from Louth.***

***Before Update:***

**Select Annual\_Sales**

**From Farmer**

**Where Farmer\_Id = 3180;**



**Insert into Rental\_Contract values** (1045,**3180**,'27-Feb-12',18,6000, **‘Quarterly Instalments'**,5060,7120);

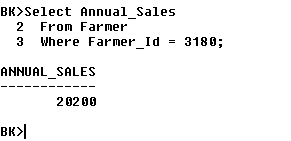


***After Update:***

**Select Annual\_Sales**

**From Farmer**

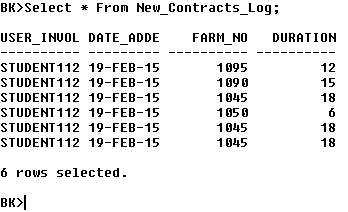
**Where Farmer\_Id = 3180;**



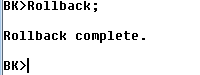
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Select \* From** **New\_Contracts\_Log;**

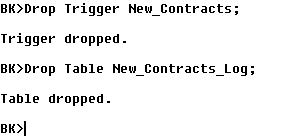


**Rollback;**



**Drop Trigger** **New\_Contracts;**

**Drop Table New\_Contracts\_Log;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Q.4 After Update Trigger***

**After a successful update on the Farm table, the following changes to the database should be made:**

**1. If the Estimated Value of a Farm has been updated and the Rental Fee for the associated Farm is under €3000:**

**Payment Terms for that Farm should be updated to 'Full Amount in Advance'.**

**2. If the Size of a Farm has been updated and the Rental Fee for the associated Farm is under €3000:**

**The Contract Duration for that Farm should be increased by the minimum duration length (six months).**

***Updates to the Farm table should be reflected in a table called Farm\_Adjustments.***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Set** **Serveroutput** **On** **Size** 1000000**;**

**Drop Table** **Farm\_Adjustments;**

**Create Table** **Farm\_Adjustments** **(Oracle\_User Varchar2(10), Date\_Changed Date, Farm\_No Number(4), New\_Esimated\_Value Number(6));**

**Create or Replace Trigger** **New\_Farm\_Updates**

**After Update** **on** **Farm**

**For Each Row**

**Declare**

**V\_Rental\_Fee Rental\_Contract.Rental\_Fee%Type;**

**V\_Min\_Duration Rental\_Contract.Duration%Type;**

**Begin**

**Select Rental\_Fee into V\_Rental\_Fee**

**From Rental\_Contract**

**Where Rental\_Contract.Farm\_Id = :Old.Farm\_Id;**

**If:Old.Estimated\_Value<>:New.Estimated\_Value And (V\_Rental\_Fee >3000) Then**

**Update** **Rental\_Contract**

**Set Payment\_Terms = ‘Full amount in Advance’**

**Where Rental\_Contract**.**Farm\_Id**=**:Old.Farm\_Id**;

**End If;**

**If :Old.Size\_**<>**:New**.Size\_ **Then**

**Select Min(Duration) into V\_Min\_Duration**

**From Rental\_Contract**

**Where Rental\_Contract.Farm\_Id = :Old.Farm\_Id;**

**Update** **Rental\_Contract**

**Set Duration = Duration + V\_Min\_Duration**

**Where Rental\_Contract.Farm\_Id** = **:Old.Farm\_Id**;

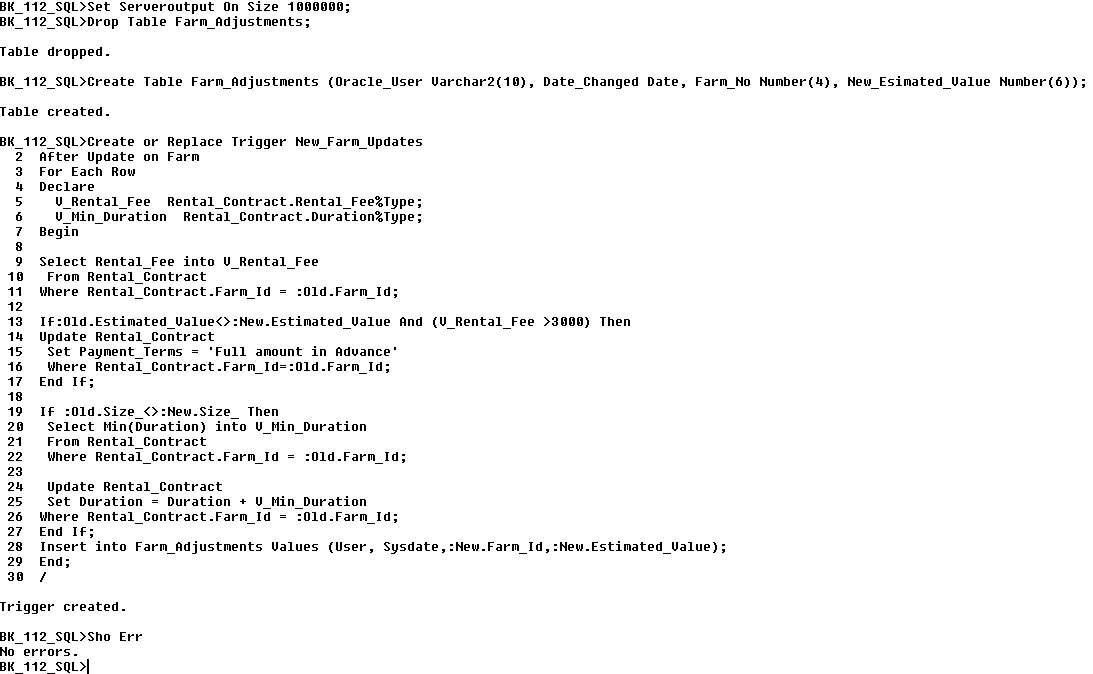
**End If;**

**Insert into** **Farm\_Adjustments** **Values** **(User, Sysdate,:New.Farm\_Id,:New.Estimated\_Value);**

**End;**

**/**

**Sho Err**



**1. If the Estimated Value of a Farm has been updated and the Rental Fee for the associated Farm is under €3000:**

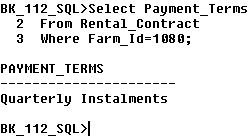
**Payment Terms for that Farm should be updated to 'Full Amount in Advance'.**

***Before Update:***

**Select Payment\_Terms**

**From Rental\_Contract**

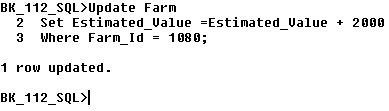
**Where Farm\_Id=1080;**



**Update Farm**

**Set Estimated\_Value =Estimated\_Value + 2000**

**Where Farm\_Id = 1080;**

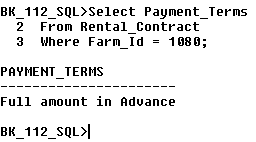


***After Update:***

**Select Payment\_Terms**

**From Rental\_Contract**

**Where Farm\_Id = 1080;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**2. If the Size of a Farm has been updated and the Rental Fee for the associated Farm is under €3000:**

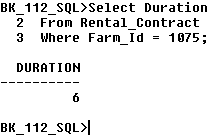
**The Contract Duration for that Farm should be increased by the minimum duration length (six months).**

***Before Update:***

**Select Duration**

**From Rental\_Contract**

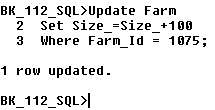
**Where Farm\_Id = 1075;**



**Update Farm**

**Set Size\_=Size\_+100**

**Where Farm\_Id = 1075;**

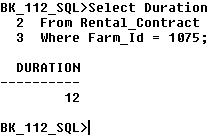


***After Update:***

**Select Duration**

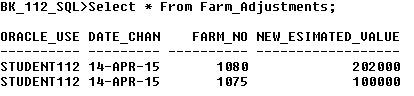
**From Rental\_Contract**

**Where Farm\_Id = 1075;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Select \* From Farm\_Adjustments;**

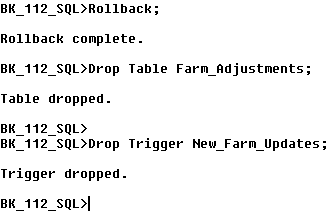


**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Rollback;**

**Drop Table** **Farm\_Adjustments;**

**Drop Trigger New\_Farm\_Updates;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Q.5 Before Insert, Update, Delete Trigger***

***In advance of a row being added, modified or removed from the Rental Contract table, certain conditions must be met with respect to those modifications. They are:***

***(a) New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments.***

***(b) Contract Duration may not be increased in value if the auctioneer does not specialise in Land*.**

***(c) Contract cannot be deleted as the rental fee is less than €8000, the solicitor is not from* *Galway*.**

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Set Serveroutput on Size 1000000;**

**Drop Trigger RC\_IUD\_Approval;**

**Create or Replace Trigger RC\_IUD\_Approval**

**Before Insert Or Update Or Delete On Rental\_Contract**

**For Each Row**

**Declare**

**V\_Qualifications** **Farmer.Qualifications%Type;**

**V\_Specialism Auctioneer.Specialism%Type;**

**V\_County Solicitor.County%Type;**

**Begin**

**/\*(a) *New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments.\*/***

**If Inserting Then**

**SELECT** **Qualifications** into **V\_Qualifications**

**FROM** **Farmer**

**WHERE** **Farmer.Farmer\_Id** **=** **:new.Farmer\_Id**;

**If:new.Payment\_Terms** **Not In** **(‘Monthly Instalments’**)**And** **V\_Qualifications** =‘**Farming** **Cert’ Then**

**Raise\_Application\_Error(**-20101**,** *'****New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments’*);**

**End If;**

**/\*(b) *Contract Duration may not be increased in value if the auctioneer does not specialise in Land*\*/**

**Elsif Updating Then**

**Select Specialism into V\_Specialism**

**From Auctioneer**

**Where Auctioneer.Auctioneer\_Id = :New.Auctioneer\_Id;**

**If:New.Duration > :Old.Duration And V\_Specialism<> ('LAND') Then**

**Raise\_Application\_Error** (**-20101**, **‘Contracts may not be increased in duration, auctioneers must specialise in land'**);

**End If;**

**/\*(c*) Contract cannot be deleted as the rental fee is less than €8000, the solicitor is not from* *Galway* \*/**

**Elsif Deleting Then**

**Select County into V\_County**

**From Solicitor**

**Where Solicitor.Solicitor\_Id =:Old.Solicitor\_Id;**

**If:Old.Rental\_Fee < 8000 And V\_County<> ('GALWAY') Then**

**Raise\_Application\_Error** (**-20101**, **‘This Contract cannot be deleted as the rental fee is less than €8000, the solicitor is not from Galway'**);

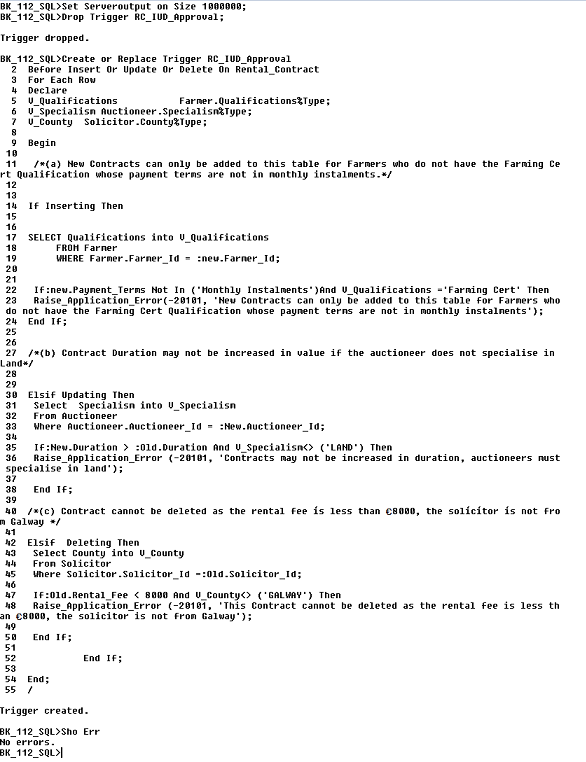
**End If;**

**End If;**

**End;**

**/**

**Sho Err**

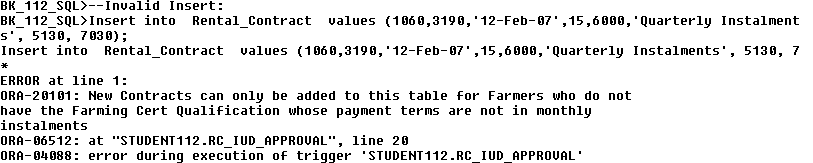


**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

***(a)*** ***New Contracts can only be added to this table for Farmers who do not have the Farming Cert Qualification whose payment terms are not in monthly instalments.***

**--Invalid Insert:**

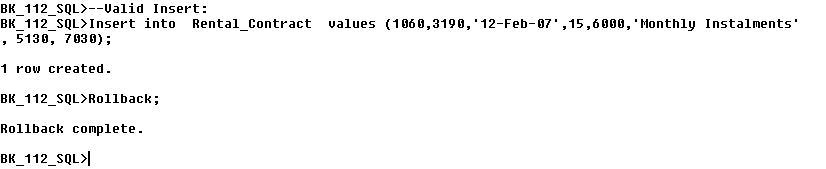
Insert into Rental\_Contract values (1060,**3190**,'12-Feb-07',15,6000,'Quarterly Instalments', 5130, 7030);



**--Valid Insert:**

Insert into Rental\_Contract values (1060,**3190**,'12-Feb-07',15,6000,**'Monthly Instalments**', 5130, 7030);

Rollback;



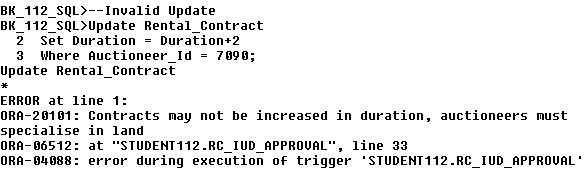
***(b) Contract Duration may not be increased in value if the auctioneer does not specialise in Land*.**

**--Invalid Update**

Update Rental\_Contract

Set Duration = Duration+2

Where Auctioneer\_Id = 7090;

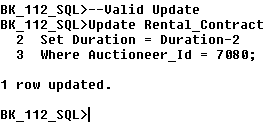


**--Valid Update**

Update Rental\_Contract

Set Duration = Duration-2

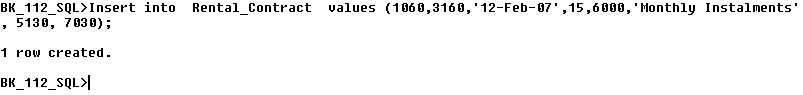
Where Auctioneer\_Id = 7080;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

***(c) Contracts cannot be deleted if the rental fee is less than €8000 and the solicitor is not from* *Galway*.**

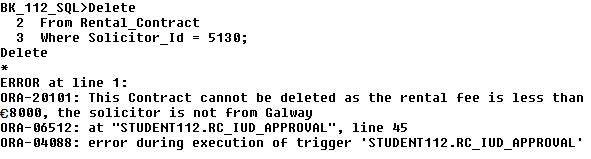
Insert into Rental\_Contract values (1060,3160,'12-Feb-07',15,**6000**,'Monthly Instalments', **5130**, 7030);



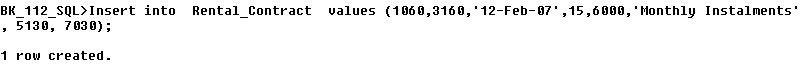
**Delete**

**From Rental\_Contract**

**Where Solicitor\_Id = 5130;**



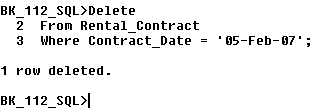
**Insert into Rental\_Contract values** (**1075,3160,'05-Feb-07',15**,**9000**,**'Monthly Instalments',** **5000**, **7130);**



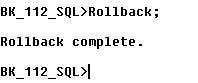
**Delete**

**From Rental\_Contract**

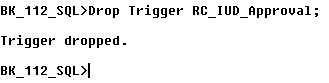
**Where Contract\_Date = '05-Feb-07';**



**Rollback;**



**Drop Trigger RC\_IUD\_Approval;**



## ***Q.6 After Insert Update Delete On Rental Contract***

1. **After a Successful insert has been made to the Rental Contract table, the following changes to the Database should be made:**

**(i) The relevant farm should have it’s size increased by 100 acres. This change should be reflected in the Farm table.**

1. **After a Successful Deletion has been made to the Rental Contract table, the following changes to the Database should be made:**

**(i) The relevant farm should have in size decreased by 100 acres. This change should be reflected in the Farm table.**

1. **After a Successful update has been made to the Rental Contract table, the following changes to the Database should be made:**
2. **If there is an increase in the duration of the updated Rental Contract then:**
3. **The associated farm should have it’s size increased by 100 acres.**
4. **If the updated Rental Contracts duration is reduced:**
5. **The associated farm should have it’s size decreased by 100 acres.**
6. **If there is no change in the updated Rental Contract’s duration:**
7. **There should be no change in the associated farm’s size.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Drop Trigger Add\_Remove\_Inc\_Dec\_Contract;**

**Create Or Replace Trigger Add\_Remove\_Inc\_Dec\_Contract**

After **Insert** Or **Update** Or **Delete** On **Rental\_Contract**

For Each Row

**Begin**

**If Inserting** Then

**Update** Farm

**Set** Size\_=Size\_+100

**Where** Farm.Farm\_Id = **:New**.Farm\_Id;

**Elsif** **Deleting** Then

**Update** Farm

**Set** Size\_ = Size\_-100

**Where** Farm.Farm\_Id=**:Old**.Farm\_Id;

**Elsif** **Updating** Then

**If:New.**Duration**>:Old.**Duration **Then**

**Update** Farm

**Set** Size\_= Size\_+100

**Where** Farm.Farm\_Id = **:New**.Farm\_Id;

**Elsif:New.**Duration<:**Old**.Duration **Then**

**Update** Farm

**Set** Size\_ = Size\_-100

**Where** Farm.Farm\_Id=**:New**.Farm\_Id;

**Elsif:New**.Duration = **:Old**.Duration **Then**

**Update** Farm

**Set** Size\_ = Size\_+0

**Where** Farm.Farm\_Id = **:New**.Farm\_Id;

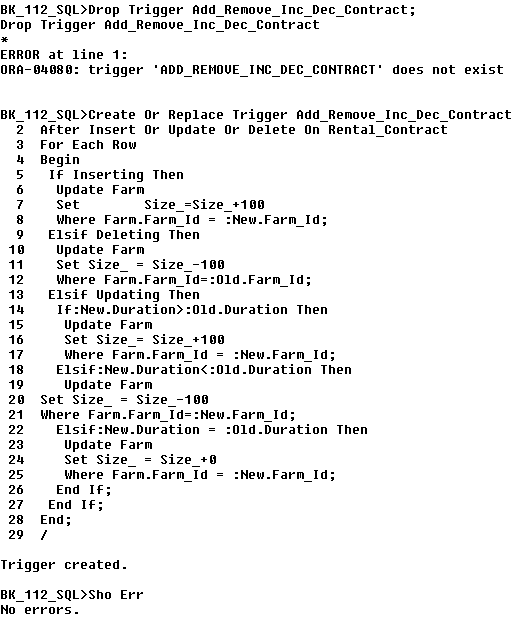
**End If;**

**End If;**

**End;**

**/**

**Sho Err**



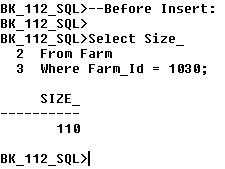
1. **After a Successful insert has been made to the Rental Contract table, the following changes to the Database should be made:**
2. **The relevant farm should have it’s size increased by 100 acres. This change should be reflected in the Farm table.**

***--Before Insert:***

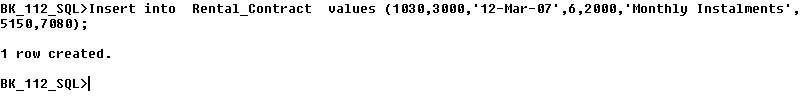
**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1030**;



**Insert** into **Rental\_Contract** **values** (**1030**,3000,**'12-Mar-07'**,6,2000,'Monthly Instalments’,5150,7080);

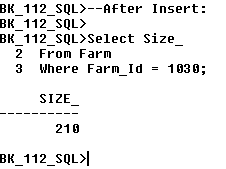


***--After Insert:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1030**;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

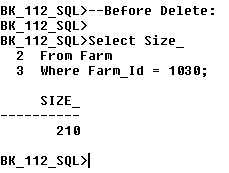
1. **After a Successful Deletion has been made to the Rental Contract table, the following changes to the Database should be made:**
2. **The relevant farm should have in size decreased by 100 acres. This change should be reflected in the Farm table.**

***--Before Delete:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1030**;

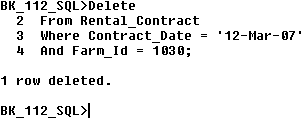


**Delete**

**From** **Rental\_Contract**

**Where** **Contract\_Date** **= '12-Mar-07'**

**And** **Farm\_Id** **= 1030;**

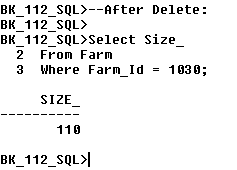


***--After Delete:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1030**;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

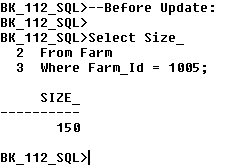
1. **After a Successful update has been made to the Rental Contract table, the following changes to the Database should be made:**
2. **If there is an increase in the duration of the updated Rental Contract then:**
3. **The associated farm should have it’s size increased by 100 acres.**

***--Before Update:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1005**;

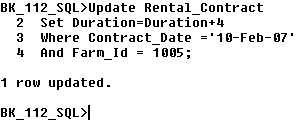


**Update Rental\_Contract**

**Set Duration=Duration+4**

**Where Contract\_Date** =**'10-Feb-07'**

**And** **Farm\_Id** **= 1005;**

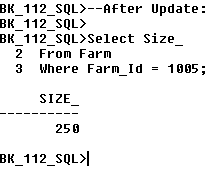


***--After Update:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1005**;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**3. After a Successful update has been made to the Rental Contract table, the following changes to the Database should be made:**

1. **If there is an increase in the duration of the updated Rental Contract then:**

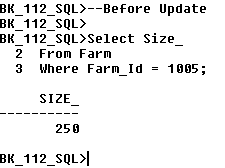
**i. The associated farm should have it’s size increased by 100 acres.**

***--Before Update***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1005**;

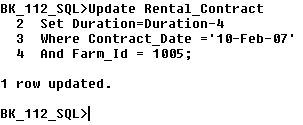


**Update Rental\_Contract**

**Set Duration=Duration-4**

**Where Contract\_Date** =**'10-Feb-07'**

**And** **Farm\_Id** **= 1005;**

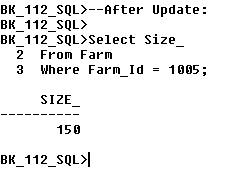


***--After Update:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1005**;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**3. After a Successful update has been made to the Rental Contract table, the following changes to the Database should be made:**

**(c) If there is no change in the updated Rental Contract’s duration:**

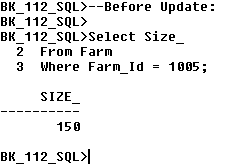
**i. There should be no change in the associated farm’s size.**

***--Before Update:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1005**;

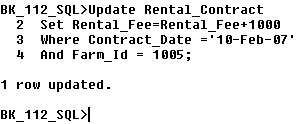


**Update Rental\_Contract**

**Set Rental\_Fee=Rental\_Fee+1000**

**Where Contract\_Date** =**'10-Feb-07'**

**And** **Farm\_Id** **= 1005;**

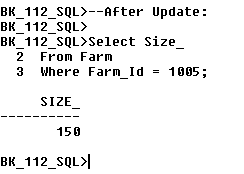


***--After Update:***

**Select** **Size\_**

**From** **Farm**

**Where Farm\_Id** = **1005**;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Rollback;**



**Drop Trigger Add\_Remove\_Inc\_Dec\_Contract;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Q.7 Before Insert/Update/Delete Table Level Trigger***

**Any changes to the database, whether update, insert or delete, are only permitted during normal working hours - Monday to Friday, 8am to 4pm.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Set Serveroutput on Size 1000000;**

**Create** **Or** **Replace** **Trigger** **Secure\_Farm**

**Before** **Insert** **Or Update Or Delete On** **Farm**

**Begin**

**/\*Changes to the database are not permitted outside normal working hours\*/**

**If** (**To\_Char**(**Sysdate,’DY’**) **In (‘Sat’,’Sun’))** **Or (To\_Char**(**sysdate,’HH24’**) **Not Between** ‘**08**’ **AND** ‘**18**’) **Then**

**If Deleting Then**

**Raise\_Application\_Error**(**-20500**, ‘**You may only delete from Farm during normal hours – Mon to Fri 8am to 6pm.’);**

**Elsif Inserting Then**

**Raise\_Application\_Error**(**-20500**, ‘**You may only insert into Farm during normal hours – Mon to Fri 8am to 6pm.’);**

**Elsif Updating Then**

**Raise\_Application\_Error**(**-20500**, ‘**You may only update Farm during normal hours – Mon to Fri 8am to 6pm.’);**

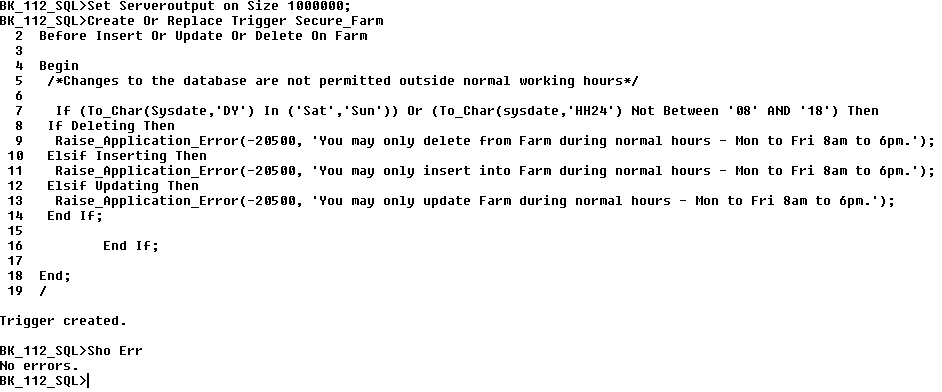
**End If;**

**End If;**

**End;**

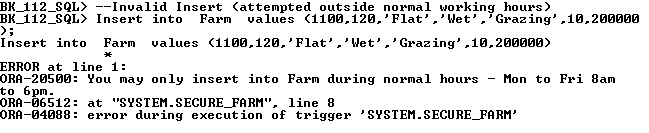
**/**

**Sho Err**



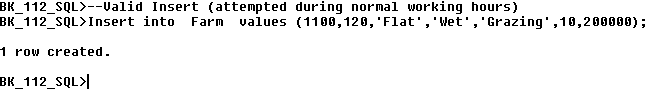
**--Invalid Insert** **(attempted outside normal working hours)**

**Insert** **into** **Farm** **values (1100,120,'Flat','Wet','Grazing',10,200000);**



**--Valid Insert** **(attempted during normal working hours)**

**Insert** **into** **Farm** **values (1100,120,'Flat','Wet','Grazing',10,200000);**

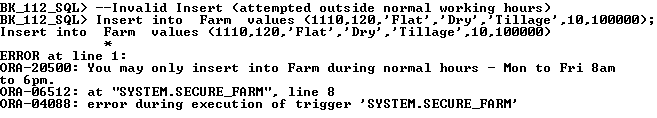


**Rollback;**



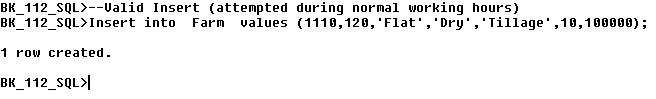
**--Invalid Insert** **(attempted outside normal working hours)**

**Insert** **into** **Farm** **values (1110,120,'Flat','Dry','Tillage',10,100000);**

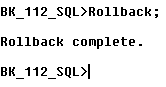


**--Valid Insert** **(attempted during normal working hours)**

**Insert** **into** **Farm** **values (1110,120,'Flat','Dry','Tillage',10,100000);**

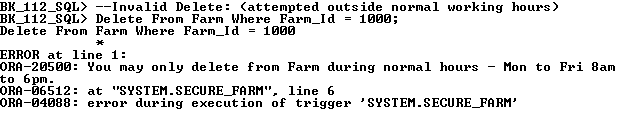


**Rollback;**



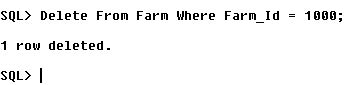
**--Invalid Delete: (attempted outside normal working hours)**

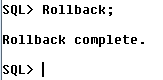
**Delete** **From Farm** **Where** **Farm**\_Id **= 1000;**



**--Valid Delete: (attempted during normal working hours)**

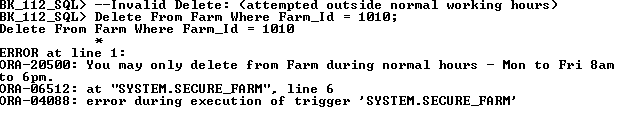
**Delete** **From Farm** **Where** **Farm**\_Id **= 1000;**





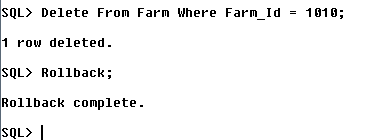
**--Invalid Delete: (attempted outside normal working hours)**

**Delete** **From Farm** **Where** **Farm**\_Id **= 1010;**



**--Valid Delete: (attempted during normal working hours)**

**Delete** **From Farm** **Where** **Farm**\_Id **= 1010;**

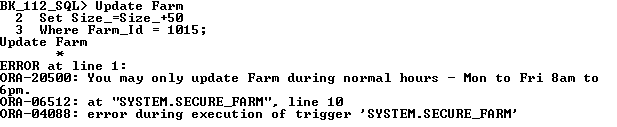


**--Invalid Update: (attempted outside normal working hours)**

**Update Farm**

**Set** **Size\_=Size\_+50**

**Where** **Farm\_Id** **=** **1015;**

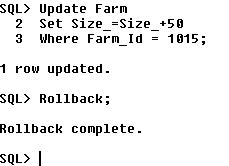


**--Valid Update: (attempted during normal working hours)**

**Update Farm**

**Set** **Size\_=Size\_+50**

**Where** **Farm\_Id** **=** **1015;**

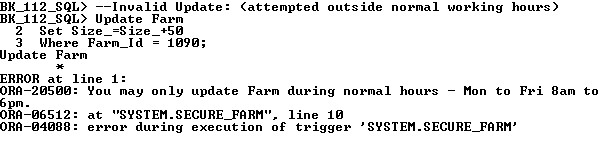


**--Invalid Update: (attempted outside normal working hours)**

**Update Farm**

**Set** **Size\_=Size\_+50**

**Where** **Farm\_Id =** **1090;**

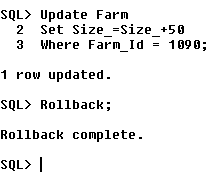


**--Valid Update: (attempted during normal working hours)**

**Update Farm**

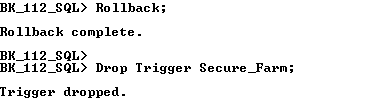
**Set** **Size\_=Size\_+50**

**Where** **Farm\_Id =** **1090;**



**Rollback;**

**Drop Trigger Secure\_Farm;**



## ***Q.8 Procedure without Cursor***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***The below Procedure allows the user to specify a Farm ID and an amount by which to increase the Farm Size of the Specified Farm. Use of an Out Variable is demonstrated here to count the number of Rental Contracts associated with the Farm in question.***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Set serveroutput on size 1000000;**

**Create Table Farm\_Size\_Failures**

**(Farm\_No number(4),**

**New\_Size number(5),**

**Error\_Msg Varchar2(25),**

**User\_Name Varchar2(10),**

**System\_Date Date);**

**Variable NumOfContracts NUMBER**

**PRINT NumOfContracts**

**Create or Replace Procedure Find\_Single\_Farm**(**Farmno IN** **integer,New\_Size IN integer, No\_Contracts OUT Number) is**

**V\_Farm\_Id Farm.Farm\_Id%Type;**

**V\_Size\_ Farm.Size\_%Type;**

**V\_Terrain Farm.Terrain%Type;**

**V\_Count\_Contracts Number(**30**):=**0**;**

**Min\_Size\_Inc Exception;**

**Max\_Size\_Inc Exception;**

**Begin**

**If New\_Size<50 Then**

**Raise Min\_Size\_Inc;**

**End if;**

**If New\_Size>150 Then**

**Raise Max\_Size\_Inc;**

**End if;**

**Select Farm\_Id,Size\_,Terrain Into V\_Farm\_Id,V\_Size\_,V\_Terrain**

**From Farm**

**Where Farm.Farm\_Id =** **Farmno;**

**DBMS\_Output.Put\_Line(‘Here is the farm number: ’|| V\_Farm\_Id);**

**DBMS\_Output.Put\_Line(‘Here is the farm size: ’|| V\_Size\_);**

**DBMS\_Output.Put\_Line(‘Here is the farm terrain: ’|| V\_Terrain);**

**DBMS\_Output.Put\_Line(‘Total number of contracts: ’|| V\_Count\_Contracts);**

**Select Count(\*) into V\_Count\_Contracts**

**From Rental\_Contract**

**Where Farm\_Id = Farmno;**

**Update Farm**

**Set Size\_= Size\_+ New\_Size**

**Where Farm.Farm\_Id=Farmno;**

**No\_Contracts :=V\_Count\_Contracts;**

**Select Farm\_Id,Size\_,Terrain Into V\_Farm\_Id,V\_Size\_,V\_Terrain**

**From Farm**

**Where Farm.Farm\_Id =** **Farmno;**

**DBMS\_Output.Put\_Line(‘Here is the farm number: ’|| V\_Farm\_Id);**

**DBMS\_Output.Put\_Line(‘Here is the farm size: ’|| V\_Size\_);**

**DBMS\_Output.Put\_Line(‘Here is the farm terrain: ’|| V\_Terrain);**

**DBMS\_Output.Put\_Line(‘Total number of contracts: ’|| V\_Count\_Contracts);**

**Exception**

**When Min\_Size\_Inc Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘Min Increase is 50’,User,SysDate);**

**When Max\_Size\_Inc Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘Max Increase is 150’,User,SysDate);**

**When No\_Data\_Found Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘No Such Farm’,User,SysDate);**

**When Others Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘Unforseen Problem’,User,SysDate);**

**End;**

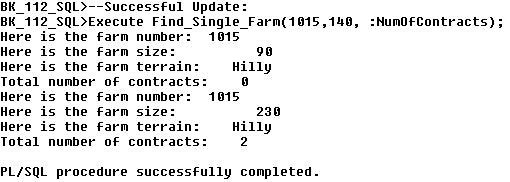
**/**

**Sho err**



**--Successful Update:**

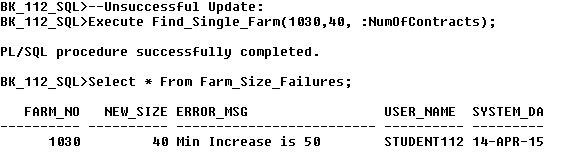
**Execute Find\_Single\_Farm(1015,140, :NumOfContracts);**



**--Unsuccessful Update:**

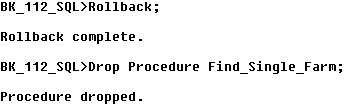
**Execute Find\_Single\_Farm(1030,40, :NumOfContracts);**

**Select \* From Farm\_Size\_Failures;**



**Rollback;**

**Drop Procedure Find\_Single\_Farm;**



**Drop Table Farm\_Size\_Failures;**



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## ***Q.9 Procedure With Cursor***

***This Procedure will display the Rental Contracts with the lowest rental fees. It will allow the User to specify a fee increase and then proceed to display relevant information of the Rental Contracts in question. An IN parameter* along with an *If-Else-Then structure* and System/User defined *Exceptions* are included in this Procedure. After execution, the audit log will be updated to reflect the number of changes made.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Set serveroutput on size 1000000;**

**Drop Table Fee\_Change\_Log;**

**Create Table Fee\_Change\_Log(**Ora\_User **Varchar2(**20**),** Change\_Date **Date,**  Rows\_Changed **Number(**3**),** Details **Varchar(**40**));**

**Drop** **Procedure Increase\_Lowest\_Fees;**

**Create or Replace Procedure Increase\_Lowest\_Fees(Increase** **IN** **Integer) As**

**Cursor** **Rentalcur** Is **Select Rental\_Fee, Payment\_Terms, Contract\_Date**

**From Rental\_Contract**

**Order by Rental\_Fee Asc**

**For Update;**

**V\_Rental\_Fee** **Rental\_Contract**.**Rental\_Fee%Type;**

**V\_Payment\_Terms** **Rental\_Contract**.**Payment\_Terms%Type;**

**V\_Contract\_Date** **Rental\_Contract**.**Contract\_Date%Type;**

**V\_Rows\_Changed** **Number(3) :=0;**

**Increase\_Too\_High Exception;**

**Increase\_Too\_Low Exception;**

**Increase\_Denied Exception;**

**Begin**

**Open** **Rentalcur;**

**DBMS\_Output.Put\_Line(‘Here are the five contracts with the lowest fees in sequence’);**

**For V\_Count in 1..5**

**Loop**

**Fetch Rentalcur Into** **V\_Rental\_Fee, V\_Payment\_Terms,V\_Contract\_Date;**

**Exit** **When** **Rentalcur** **%Notfound**;

**Dbms\_output.Put\_line(chr(**10**));**

**DBMS\_Output.Put\_Line('The User Specified Increase for Fee increase is : '** **||** **Increase);**

**DBMS\_Output.Put\_Line(‘The Contract Fee is: ’ || V\_Rental\_Fee**);

**DBMS\_Output.Put\_Line(‘The Payment Terms are: ’ || V\_Payment\_Terms**);

**DBMS\_Output.Put\_Line(‘The Contract date is: ’** || **V\_Contract\_Date**);

**DBMS\_Output.Put\_Line(‘\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);**

**If Increase>1500** **Then**

**Raise** **Increase\_Too\_High;**

**ElsIf Increase<1000** **Then**

**Raise** **Increase\_Too\_Low;**

**Else**

**If V\_Payment\_Terms = ‘Monthly Instalments’ Then**

**Raise** **Increase\_Denied;**

**Else**

**Update Rental\_Contract**

**Set Rental\_Fee = Rental\_Fee+Increase**

**Where Current Of Rentalcur;**

**End If;**

**End If;**

**V\_Rows\_Changed :=V\_Rows\_Changed+1;**

**End Loop;**

**Insert** **Into** **Fee\_Change\_Log** **Values(User,** **Sysdate,** **V\_Rows\_Changed, ‘Fee Increased’);**

**Close** **Rentalcur;**

**Exception**

**When** **Increase\_Too\_High** **Then**

**Insert** **into** **Fee\_Change\_Log Values** **(User,** **Sysdate,** **V\_Rows\_Changed,** **'Increase Too High!');**

**When** **Increase\_Too\_Low Then**

**Insert** **into** **Fee\_Change\_Log Values** **(User,** **Sysdate,** **V\_Rows\_Changed,** **'Increase Too Low!');**

**When** **Increase\_Denied Then**

**Insert** **into** **Fee\_Change\_Log Values** **(User,** **Sysdate,** **V\_Rows\_Changed,** **‘Increase\_Denied!');**

**When** **Dup\_Val\_On\_Index** **Then**

**Insert** **into** **Fee\_Change\_Log Values(User,** **Sysdate,** **V\_Rows\_Changed,** **'Duplicate Entry Found!**'**);**

**When** **Others** **Then**

**Insert** **into** **Fee\_Change\_Log Values(User,** **Sysdate,** **V\_Rows\_Changed,** **'Unforseen Error!');**

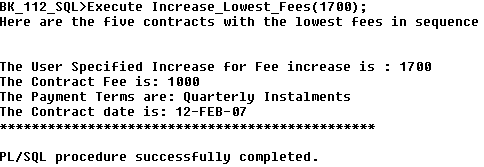
**End Increase\_Lowest\_Fees;**

**/**

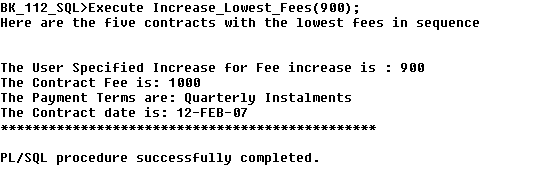
**Sho err**



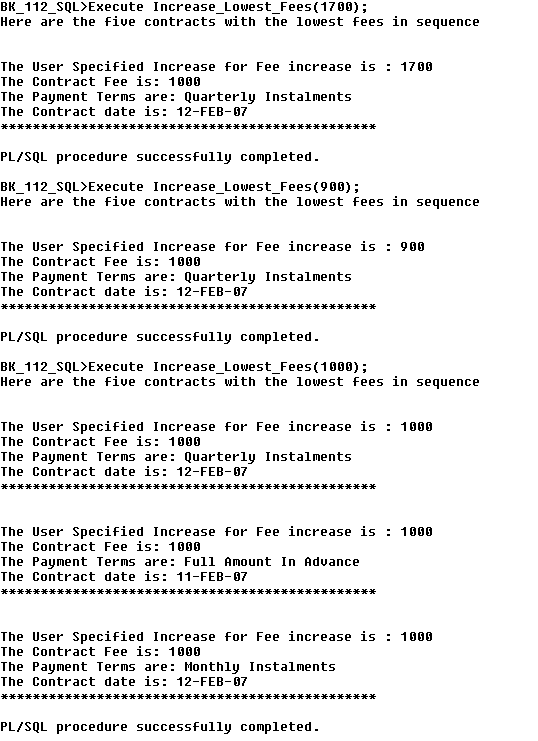
**Execute** **Increase\_Lowest\_Fees(1700);**



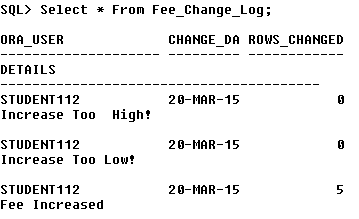
**Execute** **Increase\_Lowest\_Fees(900);**



**Execute** **Increase\_Lowest\_Fees(1000);**



**Select \* From Fee\_Change\_Log;**



**Rollback;**

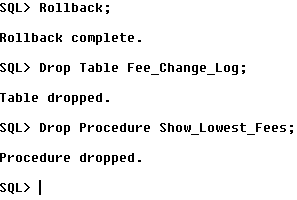


**Drop Table Fee\_Change\_Log;**



**Drop Procedure Increase\_Lowest\_Fees;**





## ***Q.10 Procedure with Cursor to Retrieve, Display and Delete***

***This Procedure will allow the User to specify a number of Farms, that have the Lowest Estimated Value, that are to be deleted from the Farm Table. The number Specified must not exceed 5 and must not be less than 3. Exception handling was implemented into the code to ensure that these conditions are met.***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Set** ServerOutput **On** **Size** **1000000;**

**Set** Linesize **30000;**

**Drop** **Table** **Farm\_DeleteLog;**

**Create** **Table** **Farm\_DeleteLog(**Oracle\_User **Varchar2(30),** Deletion\_Date **Date,** Farm\_ID **Number(30),** Usage **Varchar2(30),** Terrain **Varchar2(30),** Farm\_EstValue **Number(30),** Sum\_Est\_Value **Number(30));**

**Drop** **Table** **Farm\_AuditLog;**

**Create** **Table** **Farm\_AuditLog(**Oracle\_User **Varchar2(30),** Log\_Date **Date,** Exception\_Info **Varchar2(50));**

**Variable TotalValue** **NUMBER**

**Variable** **NumOfDeletes** **NUMBER**

**PRINT TotalValue**

**PRINT** **NumOfDeletes**

**Drop** **Procedure** **FarmDelete;**

**Create** **or** **Replace** **Procedure** **FarmDelete(**Num\_To\_Delete **In** **Integer,** Sum\_Value **Out** **Integer,** No\_Of\_Deletes **Out** **Integer)** **As**

**Cursor** **FarmCursor** **Is** **SELECT** **Farm\_ID, Usage, Terrain, Estimated\_Value**

**FROM** **Farm**

**ORDER** **BY** **Estimated\_Value**

**FOR** **UPDATE;**

**V\_Total\_Estimated\_Value** **Number(10,2):=0;**

**V\_Total\_Est\_Val\_Bottom5** **Number(10,2):=0;**

**V\_DeleteCounter** **Number(10,2):=0;**

**V\_Farm\_Id Farm.Farm\_Id%Type;**

**V\_Usage Farm.Usage%Type;**

**V\_Terrain** **Farm.Terrain%Type;**

**V\_Estimated\_Value** **Farm.Estimated\_Value%Type;**

**NumToDeleteTooSmall** **Exception;**

**NumToDeleteTooBig** **Exception;**

**ResaleValueMissing** **Exception;**

**BEGIN**

**IF** **Num\_To\_Delete** **<=** **2** **Then**

**Raise** **NumToDeleteTooSmall;**

**ElsIf** **Num\_To\_Delete** **>=** **6** **Then**

**Raise** **NumToDeleteTooBig;**

**Else**

**OPEN** **FarmCursor;**

**SELECT** **SUM(Estimated\_Value)** **into** **V\_Total\_Estimated\_Value**

**FROM** **Farm;**

**DBMS\_Output.Put\_line(chr(10));**

**DBMS\_Output.Put\_Line('Total Resale Value of all Farms BEFORE Deletions** : ' **||**

**V\_Total\_Estimated\_Value);**

**DBMS\_Output.Put\_line(chr(10));**

**FOR** V\_Count **in** **1..Num\_To\_Delete**

**LOOP**

**FETCH** **FarmCursor** **Into** V\_Farm\_ID**,** V\_Usage**,** V\_Terrain**,**V\_Estimated\_Value**;**

**EXIT** **When** **FarmCursor%**NotFound**;**

**DBMS\_Output.Put\_Line('=================================');**

**DBMS\_Output.Put\_Line('Farm ID : ' || V\_Farm\_ID);**

**DBMS\_Output.Put\_Line('Farm Usage: ' || V\_Usage);**

**DBMS\_Output.Put\_Line('Farm Terrain: ' || V\_Terrain);**

**DBMS\_Output.Put\_Line('Estimated Value : ' || V\_Estimated\_Value);**

**DBMS\_Output.Put\_Line('=================================');**

**DELETE**

**FROM** **Farm**

**WHERE** **CURRENT** **OF** **FarmCursor;**

V\_Total\_Estimated\_Value **:=** V\_Total\_Estimated\_Value **-** V\_Estimated\_Value**;**

**DBMS\_Output.Put\_Line('--Record Deleted Successfully!--');**

**DBMS\_Output.Put\_Line('=================================');**

**DBMS\_Output.Put\_line(chr(10));**

V\_DeleteCounter **:=** V\_DeleteCounter **+** 1**;**

**INSERT** **INTO** **Farm\_DeleteLog** **Values(User,** **Sysdate,** V\_Farm\_ID**,** V\_Usage**,** V\_Terrain**,** V\_Estimated\_Value**,**

V\_Total\_Estimated\_Value**);**

**END** **LOOP;**

**CLOSE** **FarmCursor;**

Sum\_Value **:=** V\_Total\_Estimated\_Value**;**

No\_Of\_Deletes **:=** V\_DeleteCounter**;**

**DBMS\_Output.Put\_Line('Total Estimated Value of all Farms AFTER Deletions : ' || V\_Total\_Estimated\_Value);**

**End** **If;**

**EXCEPTION**

**When** **NumToDeleteTooBig** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Number specified for deletion is TOO**

**EXCESSIVE!**'**);**

**When** **NumToDeleteTooSmall** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Number specified for deletion is TOO**

**SMALL!');**

**When** **No\_Data\_Found** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Farm NOT FOUND');**

**When** **Others** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Unforseen Error Occurred!');**

**END** **FarmDelete;**

**/**

**Sho Err**



**-- Test NumToDeleteTooBig Exception**

**SELECT** **Count(\*)** **FROM** **Farm;**

**Execute** **FarmDelete(7,** **:TotalValue,** **:NumOfDeletes);**

**SELECT** **Count(\*)** **FROM** **Farm;**

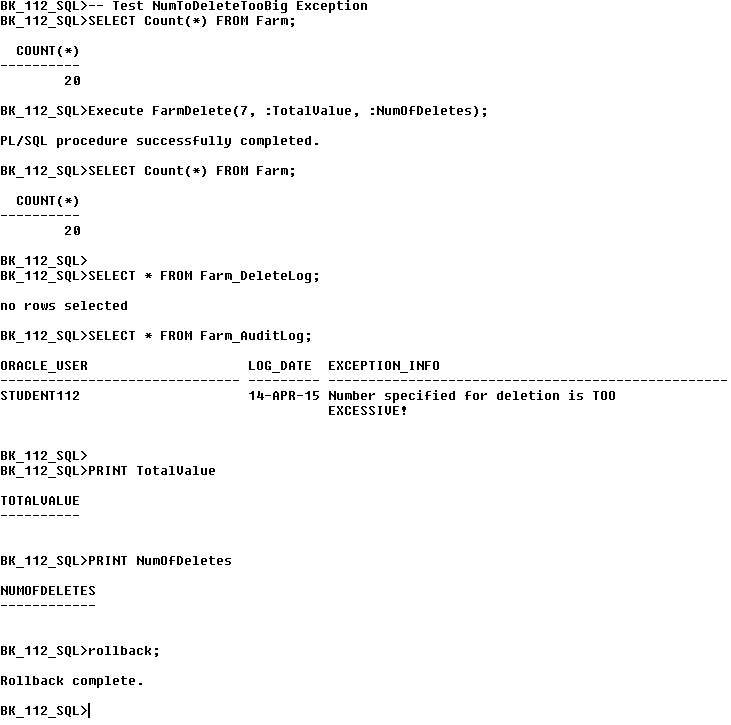
**SELECT** **\*** **FROM** **Farm\_DeleteLog;**

**SELECT** **\*** **FROM** **Farm\_AuditLog;**

**PRINT TotalValue**

**PRINT** **NumOfDeletes**

**rollback;**



**-- Test NumToDeleteTooSmall Exception**

**SELECT** **Count(\*)** **FROM** **Farm;**

**Execute** **FarmDelete(1,** **:TotalValue, :NumOfDeletes);**

**SELECT** **Count(\*)** **FROM** **Farm;**

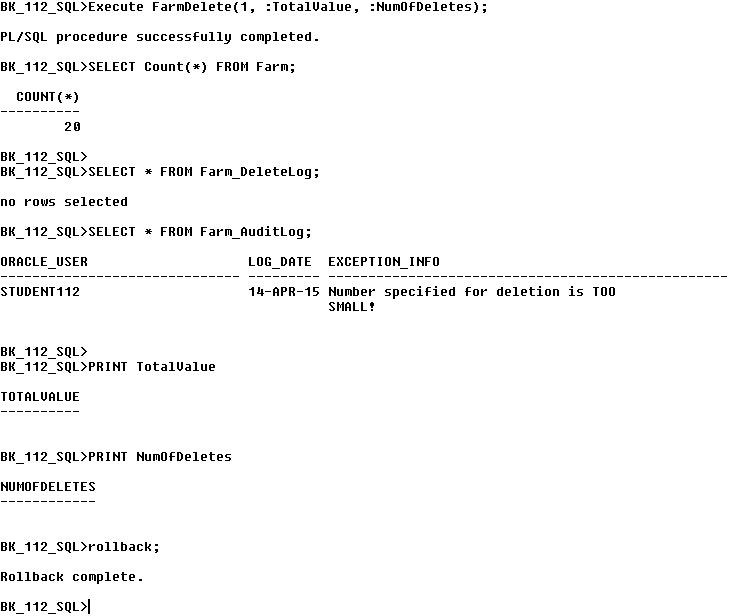
**SELECT** **\*** **FROM** **Farm\_DeleteLog;**

**SELECT** **\*** **FROM** **Farm\_AuditLog;**

**PRINT** **TotalValue**

**PRINT** **NumOfDeletes**

**rollback;**



**-- Successful Execution**

**SELECT** **Count(\*)** **FROM** **Farm;**

**Execute** **FarmDelete(3,** **:TotalValue, :NumOfDeletes);**

**SELECT** **Count(\*)** **FROM** **Farm;**

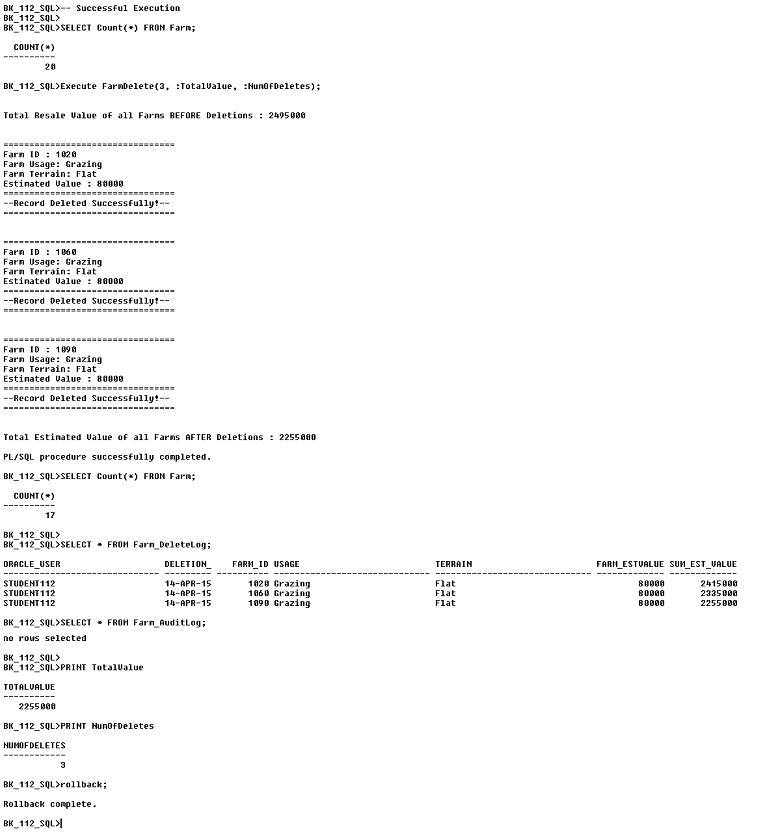
**SELECT** **\*** **FROM** **Farm\_DeleteLog;**

**SELECT** **\*** **FROM** **Farm\_AuditLog;**

**PRINT** **TotalValue**

**PRINT** **NumOfDeletes**

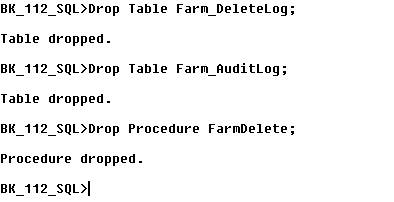
**rollback;**



**Drop Table Farm\_DeleteLog;**

**Drop Table Farm\_AuditLog;**

**Drop Procedure FarmDelete;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Q.11 Package with Three Procedures***

***The following SQL Package incorporates three different Procedures, based on Questions 8, 9 and 10.***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Set** Linesize 30000

**Set** Serveroutput **On** **Size** 1000000

-- Audit Table for Find\_Single\_Farm Procedure

**Drop** **Table** Farm\_Size\_Failures**;**

**Create Table Farm\_Size\_Failures (Farm\_No number(4),New\_Size number(5),Error\_Msg Varchar2(25),User\_Name Varchar2(10),System\_Date Date);**

-- Variable for Find\_Single\_Farm Procedure

**Variable NumOfContracts NUMBER**

**PRINT NumOfContracts**

-- Audit Table for Increase\_Lowest\_Fees Procedure

**Drop Table Fee\_Change\_Log;**

**Create Table Fee\_Change\_Log(**Ora\_User **Varchar2(**20**),** Change\_Date **Date,**  Rows\_Changed **Number(**3**),** Details **Varchar(**40**));**

-- Audit Table for FarmDelete Procedure

**Drop** **Table** **Farm\_DeleteLog;**

**Create** **Table** **Farm\_DeleteLog(**Oracle\_User **Varchar2(30),** Deletion\_Date **Date,** Farm\_ID **Number(30),** Usage **Varchar2(30),** Terrain **Varchar2(30),** Farm\_EstValue **Number(30),** Sum\_Est\_Value **Number(30));**

-- Audit Table for FarmDelete Procedure

**Drop** **Table** **Farm\_AuditLog;**

**Create** **Table** **Farm\_AuditLog(**Oracle\_User **Varchar2(30),** Log\_Date **Date,** Exception\_Info **Varchar2(50));**

-- Variable for FarmDelete Procedure

**Variable TotalValue** **NUMBER**

**Variable** **NumOfDeletes** **NUMBER**

**PRINT TotalValue**

**PRINT** **NumOfDeletes**

**Create** **or** **Replace** **Package** Procedure\_Package **Is**

**Procedure** Find\_Single\_Farm

(**Farmno IN** **integer,New\_Size IN integer, No\_Contracts OUT Number);**

**Procedure** Increase\_Lowest\_Fees

**(Increase** **IN** **Integer);**

**Procedure** FarmDelete

**(**Num\_To\_Delete **In** **Integer,** Sum\_Value **Out** **Integer,** No\_Of\_Deletes **Out** **Integer);**

**End** Procedure\_Package**;**

**/**

Sho Err

**Create** **Or** **Replace** **Package** **BODY** Procedure\_Package **Is**

-- ==================================================

-- First procedure included is : Find\_Single\_Farm

-- ==================================================

**Procedure Find\_Single\_Farm**(**Farmno IN** **integer,New\_Size IN integer, No\_Contracts OUT Number) is**

**V\_Farm\_Id Farm.Farm\_Id%Type;**

**V\_Size\_ Farm.Size\_%Type;**

**V\_Terrain Farm.Terrain%Type;**

**V\_Count\_Contracts Number(**30**):=**0**;**

**Min\_Size\_Inc Exception;**

**Max\_Size\_Inc Exception;**

**Begin**

**If New\_Size<50 Then**

**Raise Min\_Size\_Inc;**

**End if;**

**If New\_Size>150 Then**

**Raise Max\_Size\_Inc;**

**End if;**

**Select Farm\_Id,Size\_,Terrain Into V\_Farm\_Id,V\_Size\_,V\_Terrain**

**From Farm**

**Where Farm.Farm\_Id =** **Farmno;**

**DBMS\_Output.Put\_Line(‘Here is the farm number: ’|| V\_Farm\_Id);**

**DBMS\_Output.Put\_Line(‘Here is the farm size: ’|| V\_Size\_);**

**DBMS\_Output.Put\_Line(‘Here is the farm terrain: ’|| V\_Terrain);**

**DBMS\_Output.Put\_Line(‘Total number of contracts: ’|| V\_Count\_Contracts);**

**Select Count(\*) into V\_Count\_Contracts**

**From Rental\_Contract**

**Where Farm\_Id = Farmno;**

**Update Farm**

**Set Size\_= Size\_+ New\_Size**

**Where Farm.Farm\_Id=Farmno;**

**No\_Contracts :=V\_Count\_Contracts;**

**Select Farm\_Id,Size\_,Terrain Into V\_Farm\_Id,V\_Size\_,V\_Terrain**

**From Farm**

**Where Farm.Farm\_Id =** **Farmno;**

**DBMS\_Output.Put\_Line(‘Here is the farm number: ’|| V\_Farm\_Id);**

**DBMS\_Output.Put\_Line(‘Here is the farm size: ’|| V\_Size\_);**

**DBMS\_Output.Put\_Line(‘Here is the farm terrain: ’|| V\_Terrain);**

**DBMS\_Output.Put\_Line(‘Total number of contracts: ’|| V\_Count\_Contracts);**

**Exception**

**When Min\_Size\_Inc Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘Min Increase is 50’,User,SysDate);**

**When Max\_Size\_Inc Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘Max Increase is 150’,User,SysDate);**

**When No\_Data\_Found Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘No Such Farm’,User,SysDate);**

**When Others Then**

**Insert into Farm\_Size\_Failures values (Farmno,New\_Size, ‘Unforseen Problem’,User,SysDate);**

**End Find\_Single\_Farm;**

-- ==================================================

-- Second procedure included is : Increase\_Lowest\_Fees

-- ==================================================

**Procedure Increase\_Lowest\_Fees(Increase** **IN** **Integer) As**

**Cursor** **Rentalcur** Is **Select Rental\_Fee, Payment\_Terms, Contract\_Date**

**From Rental\_Contract**

**Order by Rental\_Fee Asc**

**For Update;**

**V\_Rental\_Fee** **Rental\_Contract**.**Rental\_Fee%Type;**

**V\_Payment\_Terms** **Rental\_Contract**.**Payment\_Terms%Type;**

**V\_Contract\_Date** **Rental\_Contract**.**Contract\_Date%Type;**

**V\_Rows\_Changed** **Number(3) :=0;**

**Increase\_Too\_High Exception;**

**Increase\_Too\_Low Exception;**

**Increase\_Denied Exception;**

**Begin**

**Open** **Rentalcur;**

**DBMS\_Output.Put\_Line(‘Here are the five contracts with the lowest fees in sequence’);**

**For V\_Count in 1..5**

**Loop**

**Fetch Rentalcur Into** **V\_Rental\_Fee, V\_Payment\_Terms,V\_Contract\_Date;**

**Exit** **When** **Rentalcur** **%Notfound**;

**Dbms\_output.Put\_line(chr(**10**));**

**DBMS\_Output.Put\_Line('The User Specified Increase for Fee increase is : '** **||** **Increase);**

**DBMS\_Output.Put\_Line(‘The Contract Fee is: ’ || V\_Rental\_Fee**);

**DBMS\_Output.Put\_Line(‘The Payment Terms are: ’ || V\_Payment\_Terms**);

**DBMS\_Output.Put\_Line(‘The Contract date is: ’** || **V\_Contract\_Date**);

**DBMS\_Output.Put\_Line(‘\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);**

**If Increase>1500** **Then**

**Raise** **Increase\_Too\_High;**

**ElsIf Increase<1000** **Then**

**Raise** **Increase\_Too\_Low;**

**Else**

**If V\_Payment\_Terms = ‘Monthly Instalments’ Then**

**Raise** **Increase\_Denied;**

**Else**

**Update Rental\_Contract**

**Set Rental\_Fee = Rental\_Fee+Increase**

**Where Current Of Rentalcur;**

**End If;**

**End If;**

**V\_Rows\_Changed :=V\_Rows\_Changed+1;**

**End Loop;**

**Insert** **Into** **Fee\_Change\_Log** **Values(User,** **Sysdate,** **V\_Rows\_Changed, ‘Fee Increased’);**

**Close** **Rentalcur;**

**Exception**

**When** **Increase\_Too\_High** **Then**

**Insert** **into** **Fee\_Change\_Log Values** **(User,** **Sysdate,** **V\_Rows\_Changed,** **'Increase Too High!');**

**When** **Increase\_Too\_Low Then**

**Insert** **into** **Fee\_Change\_Log Values** **(User,** **Sysdate,** **V\_Rows\_Changed,** **'Increase Too Low!');**

**When** **Increase\_Denied Then**

**Insert** **into** **Fee\_Change\_Log Values** **(User,** **Sysdate,** **V\_Rows\_Changed,** **‘Increase\_Denied!');**

**When** **Dup\_Val\_On\_Index** **Then**

**Insert** **into** **Fee\_Change\_Log Values(User,** **Sysdate,** **V\_Rows\_Changed,** **'Duplicate Entry Found!**'**);**

**When** **Others** **Then**

**Insert** **into** **Fee\_Change\_Log Values(User,** **Sysdate,** **V\_Rows\_Changed,** **'Unforseen Error!');**

**End Increase\_Lowest\_Fees;**

-- ==================================================

-- Third procedure included is : FarmDelete

-- ==================================================

**Procedure** **FarmDelete(**Num\_To\_Delete **In** **Integer,** Sum\_Value **Out** **Integer,** No\_Of\_Deletes **Out** **Integer)** **As**

**Cursor** **FarmCursor** **Is** **SELECT** **Farm\_ID, Usage, Terrain, Estimated\_Value**

**FROM** **Farm**

**ORDER** **BY** **Estimated\_Value**

**FOR** **UPDATE;**

**V\_Total\_Estimated\_Value** **Number(10,2):=0;**

**V\_Total\_Est\_Val\_Bottom5** **Number(10,2):=0;**

**V\_DeleteCounter** **Number(10,2):=0;**

**V\_Farm\_Id Farm.Farm\_Id%Type;**

**V\_Usage Farm.Usage%Type;**

**V\_Terrain** **Farm.Terrain%Type;**

**V\_Estimated\_Value** **Farm.Estimated\_Value%Type;**

**NumToDeleteTooSmall** **Exception;**

**NumToDeleteTooBig** **Exception;**

**ResaleValueMissing** **Exception;**

**BEGIN**

**IF** **Num\_To\_Delete** **<=** **2** **Then**

**Raise** **NumToDeleteTooSmall;**

**ElsIf** **Num\_To\_Delete** **>=** **6** **Then**

**Raise** **NumToDeleteTooBig;**

**Else**

**OPEN** **FarmCursor;**

**SELECT** **SUM(Estimated\_Value)** **into** **V\_Total\_Estimated\_Value**

**FROM** **Farm;**

**DBMS\_Output.Put\_line(chr(10));**

**DBMS\_Output.Put\_Line('Total Resale Value of all Farms BEFORE Deletions** : ' **||**

**V\_Total\_Estimated\_Value);**

**DBMS\_Output.Put\_line(chr(10));**

**FOR** V\_Count **in** **1..Num\_To\_Delete**

**LOOP**

**FETCH** **FarmCursor** **Into** V\_Farm\_ID**,** V\_Usage**,** V\_Terrain**,**V\_Estimated\_Value**;**

**EXIT** **When** **FarmCursor%**NotFound**;**

**DBMS\_Output.Put\_Line('=================================');**

**DBMS\_Output.Put\_Line('Farm ID : ' || V\_Farm\_ID);**

**DBMS\_Output.Put\_Line('Farm Usage: ' || V\_Usage);**

**DBMS\_Output.Put\_Line('Farm Terrain: ' || V\_Terrain);**

**DBMS\_Output.Put\_Line('Estimated Value : ' || V\_Estimated\_Value);**

**DBMS\_Output.Put\_Line('=================================');**

**DELETE**

**FROM** **Farm**

**WHERE** **CURRENT** **OF** **FarmCursor;**

V\_Total\_Estimated\_Value **:=** V\_Total\_Estimated\_Value **-** V\_Estimated\_Value**;**

**DBMS\_Output.Put\_Line('--Record Deleted Successfully!--');**

**DBMS\_Output.Put\_Line('=================================');**

**DBMS\_Output.Put\_line(chr(10));**

V\_DeleteCounter **:=** V\_DeleteCounter **+** 1**;**

**INSERT** **INTO** **Farm\_DeleteLog** **Values(User,** **Sysdate,** V\_Farm\_ID**,** V\_Usage**,** V\_Terrain**,** V\_Estimated\_Value**,**

V\_Total\_Estimated\_Value**);**

**END** **LOOP;**

**CLOSE** **FarmCursor;**

Sum\_Value **:=** V\_Total\_Estimated\_Value**;**

No\_Of\_Deletes **:=** V\_DeleteCounter**;**

**DBMS\_Output.Put\_Line('Total Estimated Value of all Farms AFTER Deletions : ' || V\_Total\_Estimated\_Value);**

**End** **If;**

**EXCEPTION**

**When** **NumToDeleteTooBig** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Number specified for deletion is TOO**

**EXCESSIVE!**'**);**

**When** **NumToDeleteTooSmall** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Number specified for deletion is TOO**

**SMALL!');**

**When** **No\_Data\_Found** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Farm NOT FOUND');**

**When** **Others** **Then**

**INSERT** **Into** **Farm\_AuditLog** **Values(User,** **Sysdate,** **'Unforseen Error Occurred!');**

**END** **FarmDelete;**

**END** Procedure\_Package**;**

**/**

**Sho Err**

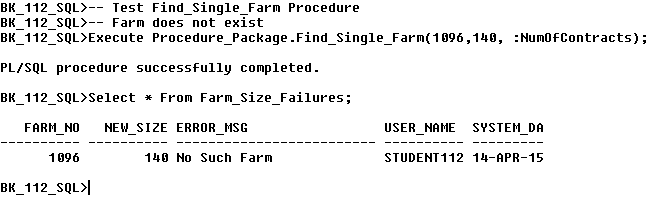


**-- Test Find\_Single\_Farm Procedure**

**-- Farm does not exist**

**Execute Procedure\_Package.Find\_Single\_Farm(1096,140, :NumOfContracts);**

**Select \* From Farm\_Size\_Failures;**

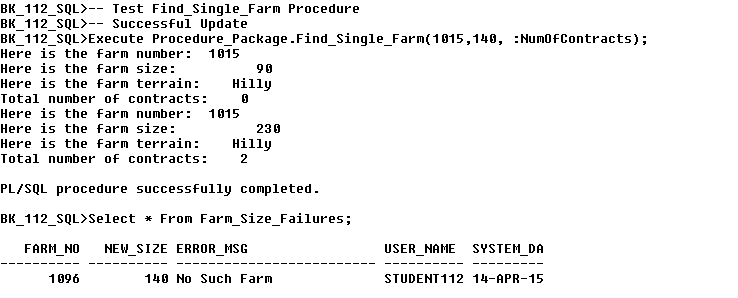


**-- Test Find\_Single\_Farm Procedure**

**-- Successful Update**

**Execute Procedure\_Package.Find\_Single\_Farm(1015,140, :NumOfContracts);**

**Select \* From Farm\_Size\_Failures;**

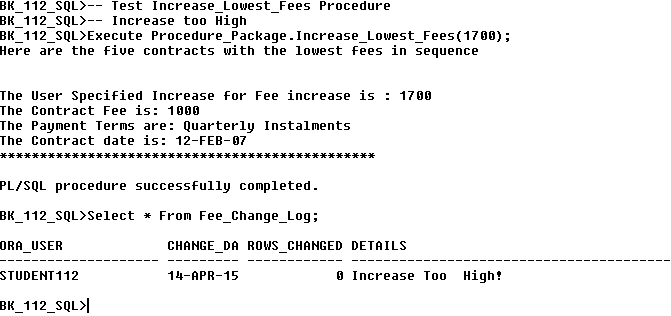


**-- Test Increase\_Lowest\_Fees Procedure**

**-- Increase too High**

**Execute** **Procedure\_Package.Increase\_Lowest\_Fees(1700);**

**Select \* From Fee\_Change\_Log;**

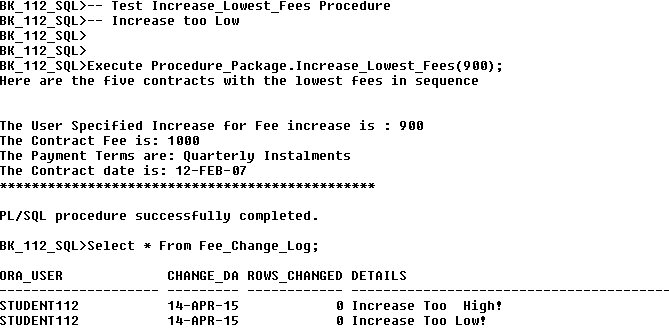


**-- Test Increase\_Lowest\_Fees Procedure**

**-- Increase too Low**

**Execute** **Procedure\_Package.Increase\_Lowest\_Fees(900);**

**Select \* From Fee\_Change\_Log;**

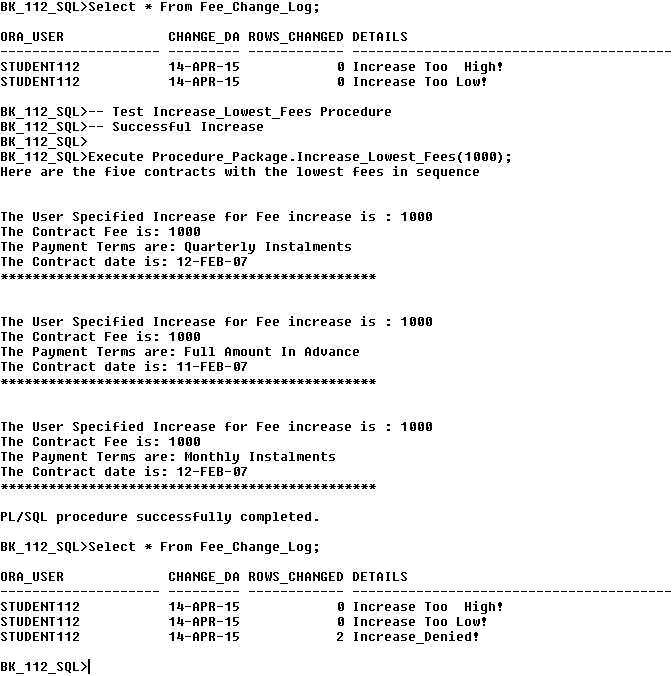


**-- Test Increase\_Lowest\_Fees Procedure**

**-- Successful Increase**

**Execute** **Procedure\_Package.Increase\_Lowest\_Fees(1000);**

**Select \* From Fee\_Change\_Log;**



**-- Test NumToDeleteTooBig Exception**

**-- Test FarmDelete Procedure**

**-- Deletes Too High Exception**

**SELECT** **Count(\*)** **FROM** **Farm;**

**Execute** **Procedure\_Package.FarmDelete(7,** **:TotalValue,** **:NumOfDeletes);**

**SELECT** **Count(\*)** **FROM** **Farm;**

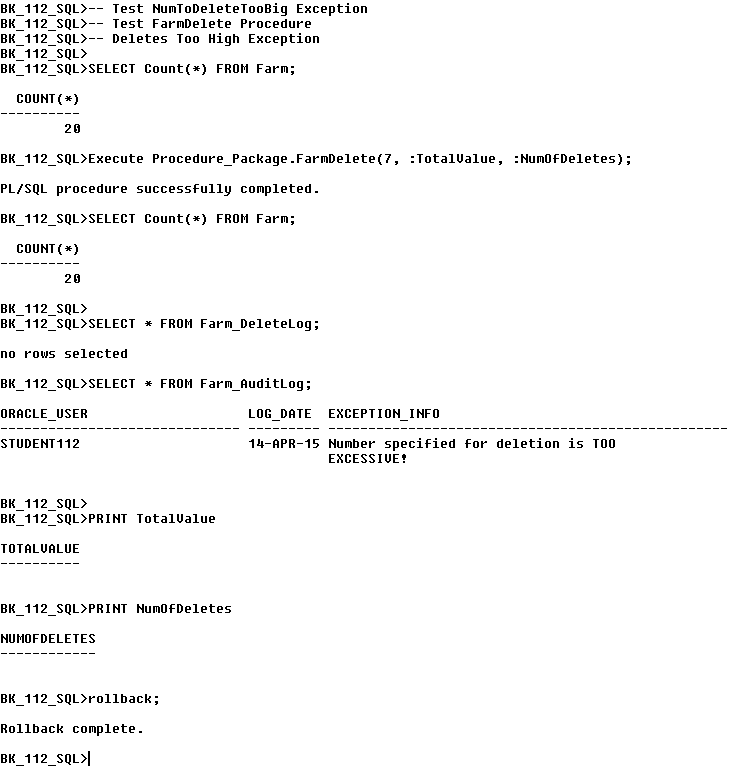
**SELECT** **\*** **FROM** **Farm\_DeleteLog;**

**SELECT** **\*** **FROM** **Farm\_AuditLog;**

**PRINT TotalValue**

**PRINT** **NumOfDeletes**

**rollback;**



**-- Test FarmDelete Procedure**

**-- NumToDeleteTooSmall Exception**

**SELECT** **Count(\*)** **FROM** **Farm;**

**Execute** **Procedure\_Package.FarmDelete(1,** **:TotalValue, :NumOfDeletes);**

**SELECT** **Count(\*)** **FROM** **Farm;**

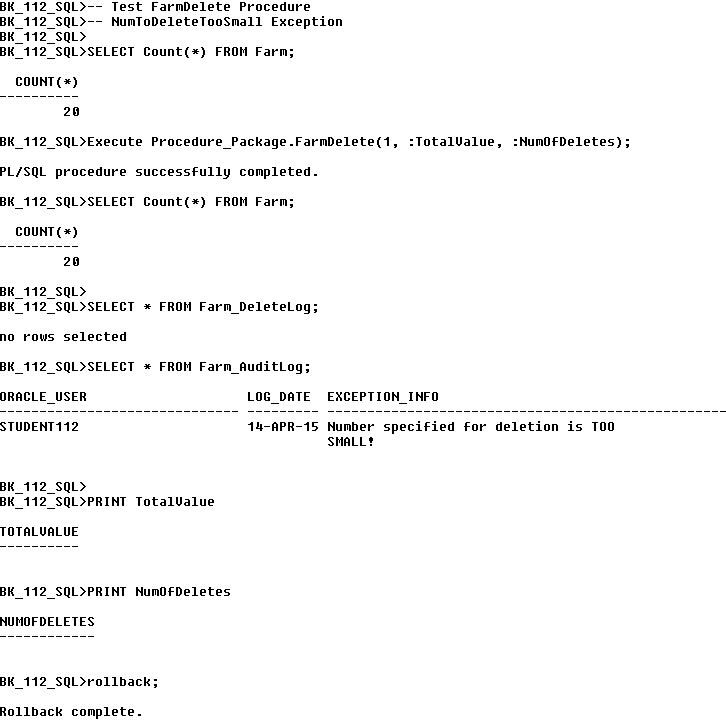
**SELECT** **\*** **FROM** **Farm\_DeleteLog;**

**SELECT** **\*** **FROM** **Farm\_AuditLog;**

**PRINT** **TotalValue**

**PRINT** **NumOfDeletes**

**rollback;**



**-- Test FarmDelete Procedure**

**-- Successful Execution**

**SELECT** **Count(\*)** **FROM** **Farm;**

**Execute** **Procedure\_Package.FarmDelete(5,** **:TotalValue, :NumOfDeletes);**

**SELECT** **Count(\*)** **FROM** **Farm;**

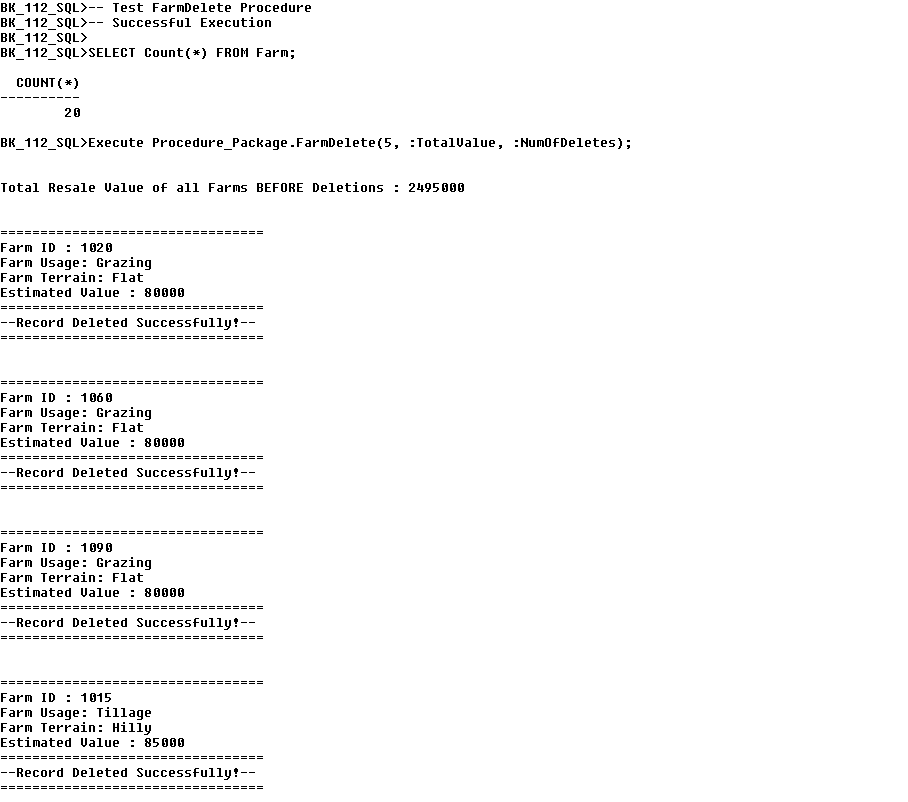
**SELECT** **\*** **FROM** **Farm\_DeleteLog;**

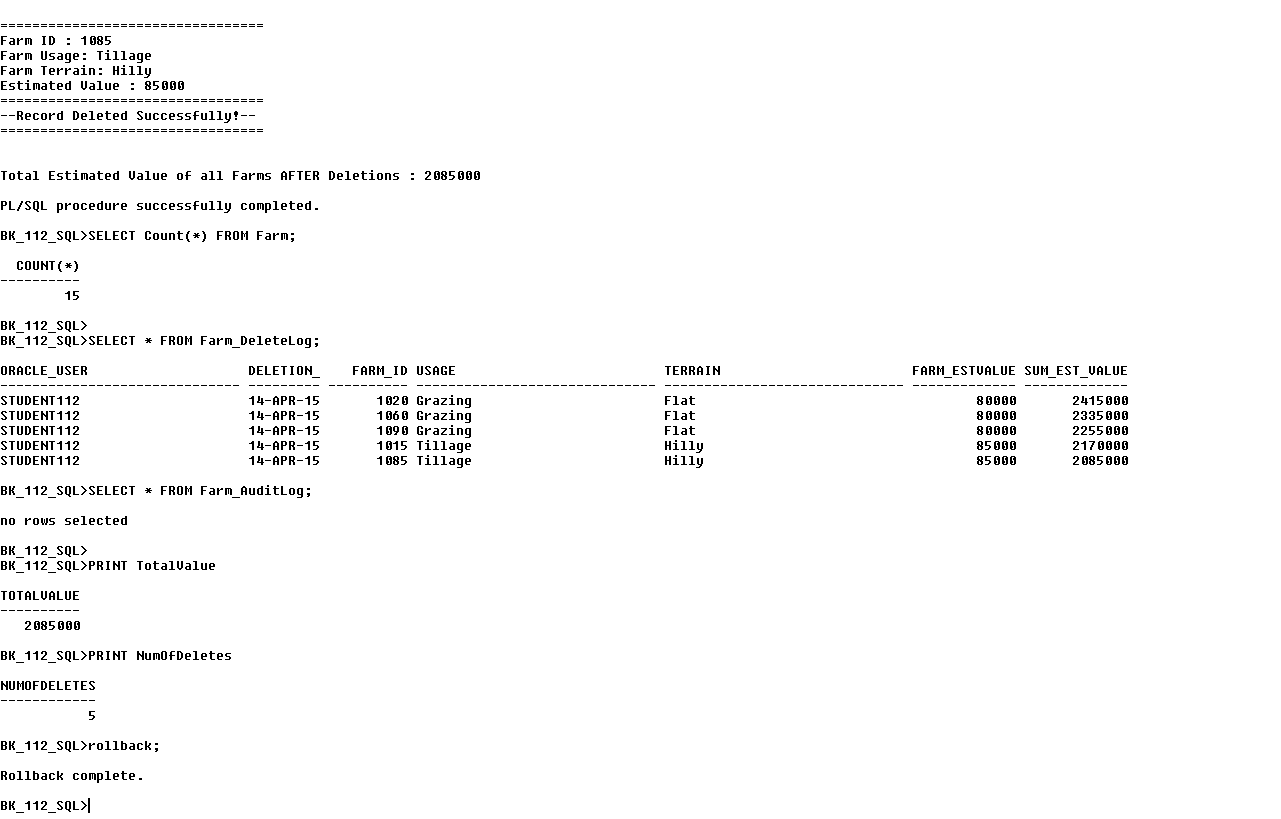
**SELECT** **\*** **FROM** **Farm\_AuditLog;**

**PRINT** **TotalValue**

**PRINT** **NumOfDeletes**

**rollback;**



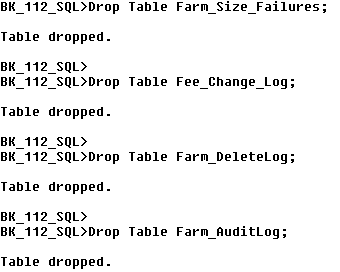


**Drop** **Table** Farm\_Size\_Failures**;**

**Drop Table Fee\_Change\_Log;**

**Drop** **Table** **Farm\_DeleteLog;**

**Drop** **Table** **Farm\_AuditLog;**



**Drop Package** Procedure\_Package**;**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Appendix***

## ***Q.17 Package with Insert, Update And Delete Procedures***

***The below PL/SQL Package incorporates three individual Procedures, all of which are based on the same table. The Procedures included in this package allow the User to Add a new record to the Farmer Table, Update the Specialism for a specified Farmer and Delete a specified record from the Farmer Table.***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**Set** ServerOutput **On** **Size** **1000000;**

**Set** Linesize **30000;**

**Create** **or** **Replace** **Package** Farmer\_Pack **Is**

**Procedure** **Add\_Farmer**

**(**A\_Id **In** Farmer**.**Farmer\_Id**%Type,** A\_Fname **In** Farmer**.F**name**%Type,** A\_Sname **In** Farmer**.S**name**%Type,** A\_Address **In**

Farmer**.**Address**%Type,** A\_County **In** Farmer**.**County**%Type,** A\_Gender **In** Farmer**.**Gender**%Type,** A\_Date\_Of\_Birth **In** Farmer**.**Date\_Of\_Birth**%Type,** A\_Phone **In** Farmer**.**Mobile\_No**%Type,** A\_Email **In** Farmer**.**Email\_Address**%Type,** A\_Qualifications **in** Farmer**.**Qualifications**%Type,** A\_Specialism **in** Farmer**.**Specialism**%Type,** A\_Sales **in** Farmer**.**Annual\_Sales**%Type);**

**Procedure** **Upd\_Farmer\_Specialism**

**(**A\_Id **In** Farmer**.Farmer**\_Id**%Type,** A\_Specialism **In** Farmer**.**Specialism**%Type);**

**Procedure** **Del\_Farmer**

**(**A\_Id **In** Farmer**.**Farmer\_Id**%Type);**

**End** Farmer\_Pack**;**

**/**

Sho Err

**Create** **Or** **Replace** **Package** **BODY** Farmer\_Pack **Is**

**-- ===============================================**

**-- First procedure included is : Add\_Farmer**

**-- ===============================================**

**Procedure** Add\_Farmer

**(**A\_Id **In** Farmer**.**Farmer\_Id**%Type,** A\_Fname **In** Farmer**.F**name**%Type,** A\_Sname **In** Farmer**.S**name**%Type,** A\_Address **In** Farmer**.**Address**%Type,** A\_County **In** Farmer**.**County**%Type,** A\_Gender **In** Farmer**.**Gender**%Type,** A\_Date\_Of\_Birth **In** Farmer**.**Date\_Of\_Birth**%Type,** A\_Phone **In** Farmer**.**Mobile\_No**%Type,** A\_Email **In** Farmer**.**Email\_Address**%Type,** A\_Qualifications **in** Farmer**.**Qualifications**%Type,** A\_Specialism **in** Farmer**.**Specialism**%Type,** A\_Sales **in** Farmer**.**Annual\_Sales**%Type)** **Is**

**BEGIN**

**INSERT** **into** Farmer**(**Farmer\_ID**,** Fname**,** Sname, Address**,** County**,** Gender,Date\_Of\_Birth**,** Mobile\_No**,** Email\_Address**,**

Qualifications**,** Specialism, Annual\_Sales**)** **Values** **(**A\_Id**,** A\_Fname**,**A\_Sname**,** A\_Address**,** A\_County**,** A\_Gender**,** A\_Date\_Of\_Birth**,** A\_Phone**,** A\_Email**,** A\_Qualifications, A\_Specialism, A\_Sales**);**

**END** Add\_Farmer**;**

**-- ===============================================**

**-- Second procedure included is : Upd\_Farmer\_Specialism**

**-- ===============================================**

**Procedure** **Upd\_Farmer\_Specialism**

**(**A\_Id **In** Farmer**.**Farmer\_Id**%Type,** A\_Specialism **In** Farmer**.**Specialism**%Type)** **Is**

**BEGIN**

**UPDATE** Farmer

**SET** Specialism **=** A\_Specialism

**WHERE** Farmer\_ID **=** A\_Id**;**

**IF** **SQL%**NotFound **Then**

Raise\_Application\_Error**(-**20203**,** 'Farmer ID does NOT Exist, UPDATE Failed!'**);**

**End** **If;**

**END** Upd\_Farmer\_Specialism**;**

**-- ===============================================**

**-- Third procedure included is : Del\_Farmer**

**-- ===============================================**

**Procedure** **Del\_Farmer**

**(**A\_Id **In** Farmer**.**Farmer\_Id**%Type)** **Is**

**BEGIN**

**DELETE**

**FROM** Farmer

**WHERE** Farmer\_Id **=** A\_Id**;**

**IF** **SQL%**NotFound **Then**

Raise\_Application\_Error**(-**20203**,** 'Farmer ID does NOT Exist, DELETE Failed!'**);**

**End** **If;**

**END** Del\_Farmer**;**

**END** Farmer\_Pack**;**

**/**

Sho Err



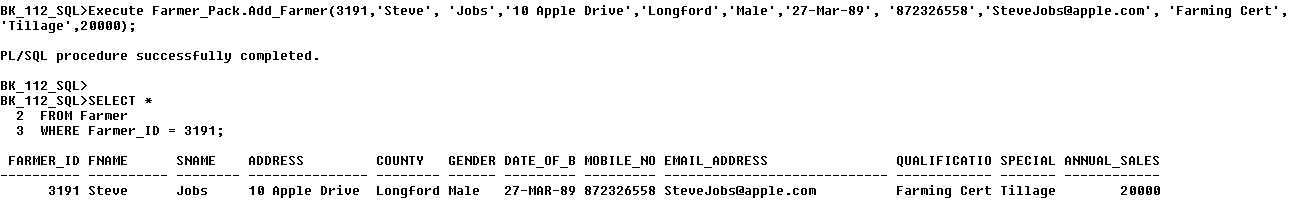


**Execute** Farmer\_Pack**.**Add\_Farmer**(**3191**,**'Steve', 'Jobs'**,**'10 Apple Drive'**,**'Longford'**,'**Male'**,**'27-Mar-89'**,** '872326558'**,**'SteveJobs@apple.com'**,** 'Farming Cert','Tillage',20000**);**

**SELECT** **\***

**FROM** Farmer

**WHERE** Farmer\_ID **=** 3191**;**



-- Test Upd\_Farmer\_Specialism Procedure

**SELECT** Specialism

**FROM** Farmer

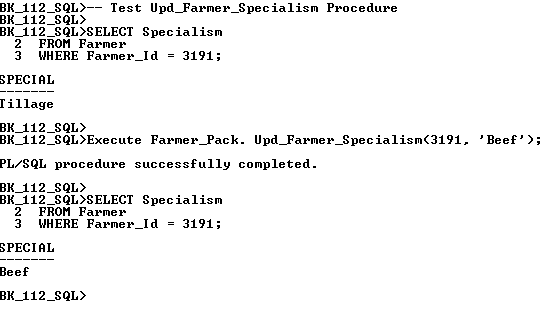
**WHERE** Farmer\_Id **=** 3191**;**

**Execute** Farmer\_Pack**.** Upd\_Farmer\_Specialism**(**3191**,** 'Beef'**);**

**SELECT** Specialism

**FROM** Farmer

**WHERE** Farmer\_Id **=** 3191**;**



-- Test Del\_Farmer Procedure

**SELECT** **Count(\*)**

**FROM** Farmer

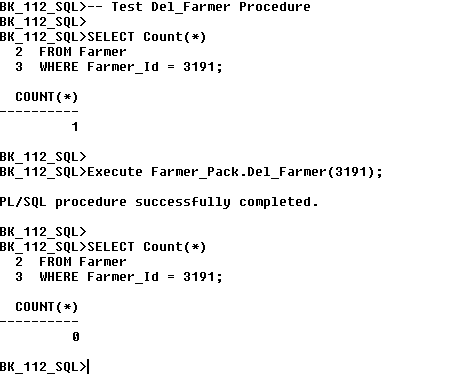
**WHERE** Farmer\_Id **=** 3191**;**

**Execute** Farmer\_Pack**.**Del\_Farmer**(**3191**);**

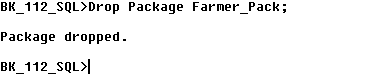
**SELECT** **Count(\*)**

**FROM** Farmer

**WHERE** Farmer\_Id **=** 3191**;**



Drop **Package** Farmer\_Pack;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## ***Q.19 Trigger That Uses a Cursor***

***Below is a demonstration of an After Update Trigger, based upon the Auctioneer Table, which incorporates the use of a Cursor to make specified changes to the Rental Contract Table as follows: When an Auctioneer's Annual Turnover figure is updated, two Rental Contracts with the lowest Fee, associated with the Auctioneer in question will have €1000 added to their Rental Fee.***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Set serveroutput on size 1000000;**

**Drop** **Table** **Auct\_Audit\_Log;**

**Create** **Table** Auct\_Audit\_Log**(**Name **Varchar2(**30**),** Pre\_Tot\_Rental\_Fee **Number(**6**),** Post\_Tot\_Rental\_Fee **Number(**6**),** Tot\_Rental\_Fee\_Increase **Number(**6**),** Oracle\_User **Varchar2(**30**),** Date\_Changed **Date);**

**Drop** **Table** **Auct\_Exceptions\_Log;**

**Create** **Table** Auct\_Exceptions\_Log**(**Oracle\_User **VarChar2(**20**),** Date\_Changed **Date,** Auct\_ID **Number(**4**),** Details **Varchar(**40**));**

**Create** **Or** **Replace** **Trigger** **Auct\_Adjust\_Inc\_Rental\_Fee**

**After** **Update** **Of** Annual\_Turnover **On** Auctioneer

**For** **Each** **row**

**Declare**

**Cursor** Rental\_Fee\_Cur **Is** **SELECT** Contract\_Date**,** Duration**,** Rental\_Fee

**FROM** Rental\_Contract

**WHERE** Auctioneer\_ID **=** **:new.**Auctioneer\_ID

**Order** **By** Rental\_Fee

**For** **Update** **Of** Rental\_Fee**;**

V\_Contract\_Date Rental\_Contract**.**Contract\_Date**%Type;**

V\_Duration Rental\_Contract**.**Duration**%Type;**

V\_Rental\_Fee Rental\_Contract**.**Rental\_Fee**%Type;**

V\_Sum\_Fee\_Before **Number(**30**):=**0**;**

V\_Sum\_Fee\_After **Number(**30**):=**0**;**

**BEGIN**

**Open** Rental\_Fee\_Cur**;**

**SELECT** **Sum(**Rental\_Fee**)** **into** V\_Sum\_Fee\_Before

**FROM** Rental\_Contract

**WHERE** Auctioneer\_ID **=** **:new.**Auctioneer\_ID**;**

DBMS\_Output**.**Put\_Line**(**'Sum Total of Rental Fees Before Changes are made is : ' **||**

V\_Sum\_Fee\_Before**);**

**For** V\_Count **in** 1..2

**Loop**

**Fetch** Rental\_Fee\_Cur **into** V\_Contract\_Date**,** V\_Duration**,** V\_Rental\_Fee**;**

**Exit** **When** Rental\_Fee\_Cur**%**NotFound**;**

DBMS\_Output**.**Put\_Line**(**'=================================='**);**

DBMS\_Output**.**Put\_Line**(**'Contract Date: ' **||** V\_Contract\_Date**);**

DBMS\_Output**.**Put\_Line**(**'Duration: ' **||** V\_Duration**);**

DBMS\_Output**.**Put\_Line**(**'Rental Fee : ' **||** V\_Rental\_Fee**);**

DBMS\_Output**.**Put\_Line**(**'=================================='**);**

**UPDATE** Rental\_Contract

**SET** Rental\_Fee **=** Rental\_Fee **+** 1000

**WHERE** **Current** **Of** Rental\_Fee\_Cur**;**

**End** **Loop;**

**Close** Rental\_Fee\_Cur**;**

**SELECT** **Sum(**Rental\_Fee**)** **into** V\_Sum\_Fee\_After

**FROM** Rental\_Contract

**WHERE** Auctioneer\_ID **=** **:new.**Auctioneer\_ID**;**

DBMS\_Output**.**Put\_Line**(**'Sum Total of Rental Fees After Changes are made is : ' **||**

V\_Sum\_Fee\_After**);**

**INSERT** **into** **Auct\_Audit\_Log Values(:new.**Auctioneer\_ID**,** V\_Sum\_Fee\_Before**,** V\_Sum\_Fee\_After**,** V\_Sum\_Fee\_After **–**

V\_Sum\_Fee\_Before**,** **User,** **Sysdate);**

-- Exceptions Handling

**Exception**

**When** No\_Data\_Found **Then**

**Insert** **into** **Auct\_Exceptions\_Log** **Values(User,** **Sysdate,** **:new.**Auctioneer\_ID**,** 'Data Not Found!'**);**

**When** DUP\_VAL\_ON\_INDEX **Then**

**Insert** **into** **Auct\_Exceptions\_Log** **Values(User,** **Sysdate,** **:new.**Auctioneer\_ID**,** 'Duplicate Entry Found!'**);**

**When** **Others** **Then**

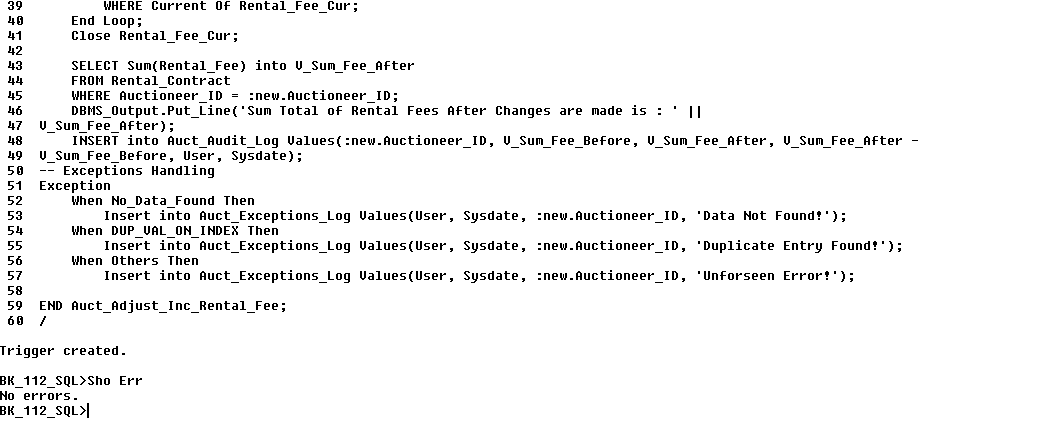
**Insert** **into** **Auct\_Exceptions\_Log** **Values(User,** **Sysdate,** **:new.**Auctioneer\_ID**,** 'Unforseen Error!'**);**

**END** **Auct\_Adjust\_Inc\_Rental\_Fee;**

**/**

Sho Err



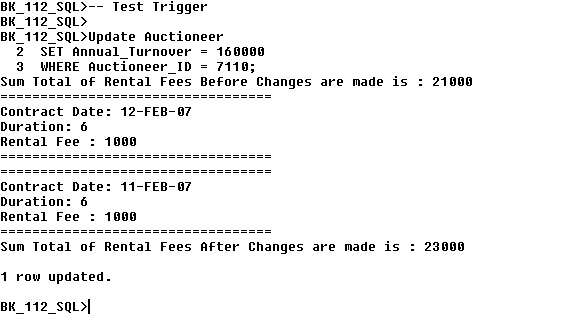


-- Test Trigger

**Update** Auctioneer

**SET** Annual\_Turnover **=** 160000

**WHERE** Auctioneer\_ID **=** 7110**;**



**Drop Trigger Auct\_Adjust\_Inc\_Rental\_Fee;**



## ***Q.18 Trigger that Calls a Procedure***

**The following Trigger calls a Procedure to update the Auctioneer's Annual Turnover depending on what action the User has performed on the Rental Contract table.**

**(a) If a New Contract is added to the Rental Contract Table :**

**The Auctioneer associated with this new contract will have their Annual Turnover increased by the Rental Fee amount of the recently added Contract.**

**(b) If an Existing Contract has had its Fee Updated :**

**The Auctioneer of the Contract in question will have their Annual Turnover increased by the difference between the previous Rental Fee of the recently updated Rental Fee.**

**(c) If a New Rental Contract is Removed to the Rental Contract :**

**The Auctioneer of the Contract in question will have their Annual Turnover decreased by the Rental Fee of the recently deleted Contract.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

-- Create the Inc\_Auct\_Turnover Procedure

**Drop** **Procedure** Inc\_Auct\_Turnover**;**

**Create** **or** **Replace** **Procedure** Inc\_Auct\_Turnover**(**V\_Auct\_ID **In** Auctioneer**.**Auctioneer\_ID**%Type,** V\_Amount **In** Auctioneer**.**Annual\_Turnover**%Type)** **Is**

**BEGIN**

**UPDATE** Auctioneer

**SET** Annual\_Turnover **=** **Nvl(**Annual\_Turnover**,** 0**)** **+** V\_Amount

**WHERE** Auctioneer\_ID **=** V\_Auct\_ID**;**

**END** Inc\_Auct\_Turnover**;**

**/**

Sho Err

-- Create the Compute\_Turnover Trigger

**Drop** **Trigger** Compute\_Turnover**;**

**Create** **Or** **Replace** **Trigger** Compute\_Turnover

**After** **INSERT** **OR** **UPDATE** **Of** Rental\_Fee **OR** **DELETE** **ON** Rental\_Contract

**For** **Each** **Row**

**BEGIN**

**If** Deleting **Then**

Inc\_Auct\_Turnover**(:Old.**Auctioneer\_ID**,** **-**1 **\*** **:Old.**Rental\_Fee**);**

**ElsIf** Updating **Then**

Inc\_Auct\_Turnover**(:New.**Auctioneer\_ID**,** **:New.**Rental\_Fee **-** **:Old.**Rental\_Fee**);**

**Else**

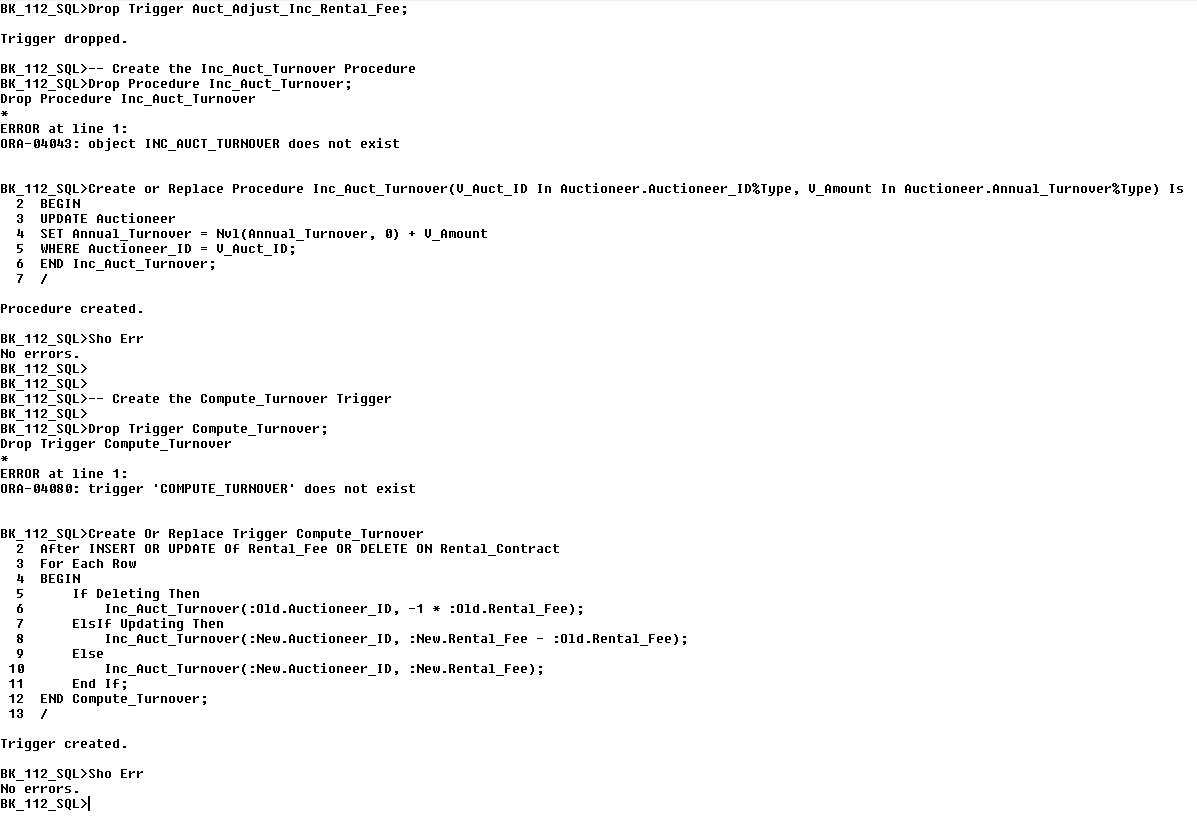
Inc\_Auct\_Turnover**(:New.**Auctioneer\_ID**,** **:New.**Rental\_Fee**);**

**End** **If;**

**END** Compute\_Turnover**;**

**/**

Sho Err

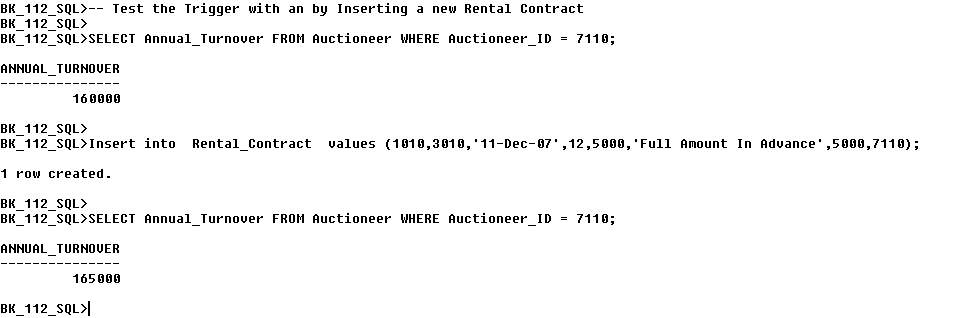


-- Test the Trigger with an by Inserting a new Rental Contract

**SELECT** Annual\_Turnover **FROM** Auctioneer **WHERE** Auctioneer\_ID **=** 7110**;**

**Insert into** Rental\_Contract **values** (1010,3010,'11-Dec-07',12,5000,'Full Amount In Advance',5000,7110);

**SELECT** Annual\_Turnover **FROM** Auctioneer **WHERE** Auctioneer\_ID **=** 7110**;**



-- Test the Trigger by Updating a Rental Contract

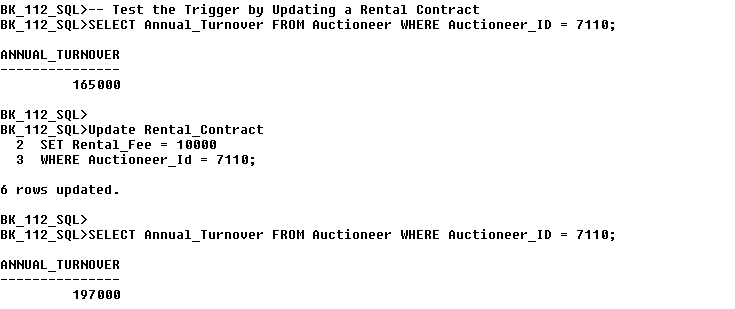
**SELECT** Annual\_Turnover **FROM** Auctioneer **WHERE** Auctioneer\_ID **=** 7110**;**

**Update** Rental\_Contract

**SET** Rental\_Fee **=** 10000

**WHERE** Auctioneer\_Id **=** 7110**;**

**SELECT** Annual\_Turnover **FROM** Auctioneer **WHERE** Auctioneer\_ID **=** 7110**;**



-- Test the Trigger by Deleting a Rental Contract

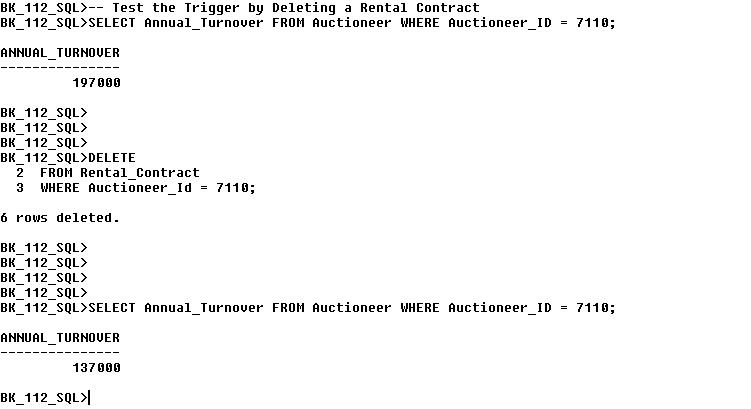
**SELECT** Annual\_Turnover **FROM** Auctioneer **WHERE** Auctioneer\_ID **=** 7110**;**

**DELETE**

**FROM** Rental\_Contract

**WHERE** Auctioneer\_Id **=** 7110**;**

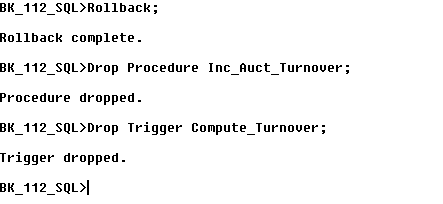
**SELECT** Annual\_Turnover **FROM** Auctioneer **WHERE** Auctioneer\_ID **=** 7110**;**



**Rollback;**

**Drop** **Procedure** Inc\_Auct\_Turnover**;**

**Drop** **Trigger** Compute\_Turnover**;**



## ***Q.14 Table Level Trigger Before Delete***

***This Table Level Trigger dictates the following conditions before a Deletion can occur on the Rental Contract Table:***

***a) A User may not delete an entry from the Rental Contract table if there are less than 15 Records present on the Table.***

***b) In the case that a Deletion will be allowed, the Rental Contract Audit table will be updated to reflect the User details, date of deletion and the Number of entries residing on the Rental Contract Table before the Deletion takes place.***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Set Serveroutput on Size 1000000;**

**DROP** **TABLE** Audit\_Contracts**;**

**CREATE** **TABLE** Audit\_Contracts**(**Oracle\_User **VarChar2(**20**),** Delete\_Date **Date,** No\_Contracts **Number(**30**,**2**),** Sum\_Fees **Number(**30**,**2**));**

**Create** **Or** **Replace** **Trigger** Q14\_Contract\_Audit

**Before** **Delete** **On** Rental\_Contract

**Declare**

V\_No\_Contracts **Number(**30**):=**0**;**

V\_Sum\_Rental\_Fee **Number(**30**):=**0**;**

**Begin**

**SELECT** **COUNT(\*)** **into** V\_No\_Contracts

**FROM** Rental\_Contract**;**

**SELECT** **SUM(**Rental\_Fee**)** **into** V\_Sum\_Rental\_Fee

**FROM** Rental\_Contract**;**

**If** V\_No\_Contracts **<** 15 **Then**

Raise\_Application\_Error**(-**20101**,** 'You may NOT DELETE from this Table if the number of entries is less than 15'**);**

**Elsif** V\_No\_Contracts **>=** 15 **Then**

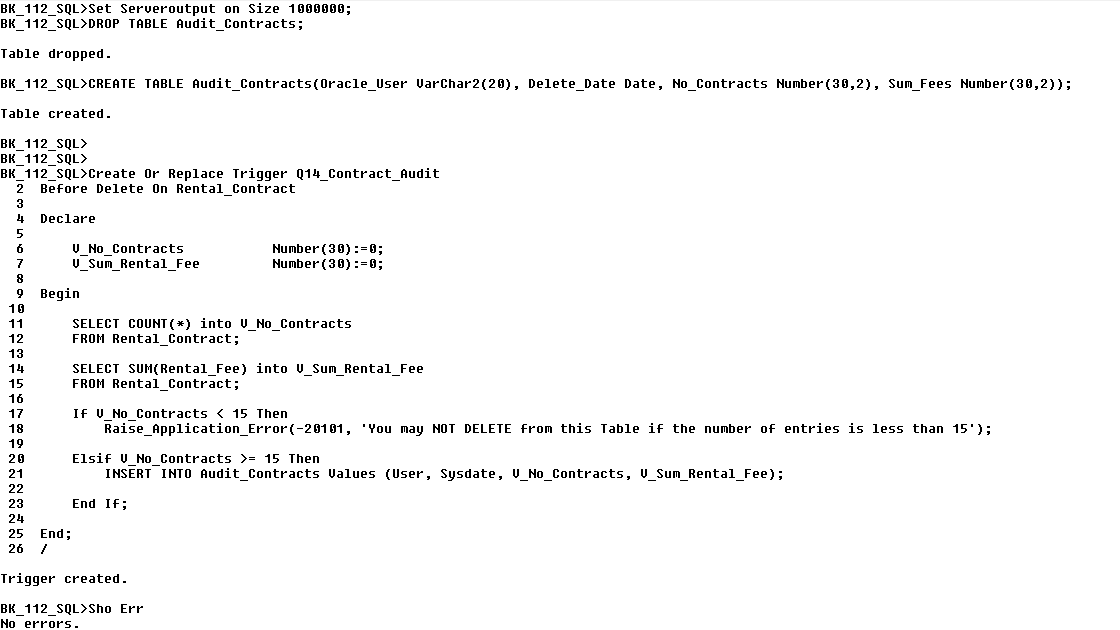
**INSERT** **INTO** Audit\_Contracts **Values** **(User,** **Sysdate,** V\_No\_Contracts**,** V\_Sum\_Rental\_Fee**);**

**End** **If;**

**End;**

**/**

Sho Err



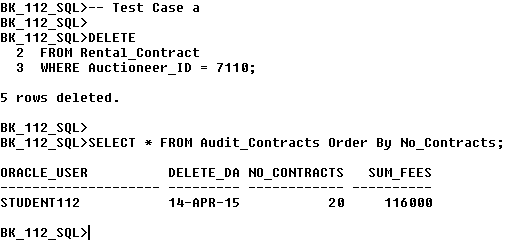
**-- Test Case a**

**DELETE**

**FROM** Rental\_Contract

**WHERE** Auctioneer\_ID **=** 7110**;**

**SELECT** **\*** **FROM** Audit\_Contracts **Order** **By** No\_Contracts**;**



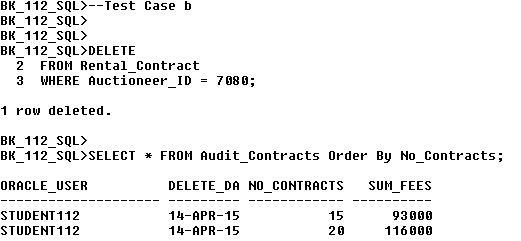
**--Test Case b**

**DELETE**

**FROM** Rental\_Contract

**WHERE** Auctioneer\_ID **=** 7080**;**

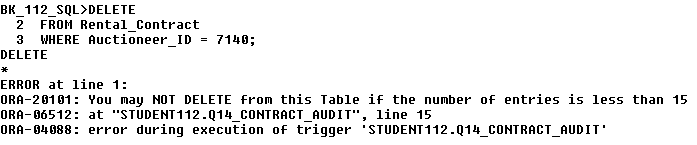
**SELECT** **\*** **FROM** Audit\_Contracts **Order** **By** No\_Contracts**;**



**DELETE**

**FROM** Rental\_Contract

**WHERE** Auctioneer\_ID **=** 7140**;**



**DROP** **TABLE** Audit\_Contracts**;**

**Drop Trigger** Q14\_Contract\_Audit;

