

|  |
| --- |
|  |
| Related image  PUB LEASE CONTRACTS  Advanced Databases  PL/SQL Triggers  Assignment  By Yash Sinojia [A00268852] |
| MASTER OF SCIENCE IN DATA ANALYTICS  2019 - 2020  FACULTY OF BUSINESS AND HOSPITALITY  SEMESTER 2 |

CONTENTS

[1. INTRODUCTION 2](#_Toc40256801)

[2. PROJECT OUTLINE 4](#_Toc40256802)

[3. ENTITY RELATIONSHIP DIAGRAM 5](#_Toc40256803)

[4. TABLES 6](#_Toc40256804)

[5. TRIGGERS 9](#_Toc40256805)

[5.1 Before Insert On Lease\_Agreement 9](#_Toc40256806)

[5.2 Before Delete On Pub 11](#_Toc40256807)

[5.3 After Insert On Letting\_Agent 13](#_Toc40256808)

[5.4 After Update On Lease\_Agreement 15](#_Toc40256809)

[5.5 After Delete On Pub 19](#_Toc40256810)

# INTRODUCTION

This assignment is an ambitious attempt to generate 5 distinct PL/SQL triggers on the set of Pub Lease Contracts (PLC) tables. This assignment also takes the challenge of understanding PL/SQL triggers on a limited time span, assimilating the knowledge and apply it on forming and testing triggers, making sure that all the possible case scenarios are covered within testing scope.

The triggers are categorized into Before and After Triggers and within that sub-categorized into Insert, Update and Delete triggers.

The 5 queries formed in this report are without any exception based on Intermediate Level Trigger Examples provided by quite informative screencasts of Noel Tierney. The practice and confidence gained from the resource materials provided has been appreciated and has been significantly helpful in completing this assignment within the short period of approaching deadline.

Note that all the sceanarios described here complies within the scopes of guidelines and limits as described in the project overview. All the queries have been executed on Oracle SQL+ Command Line.

# PROJECT OUTLINE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21 | Yash | Sinojia | Pub Lease Contracts | PLC |

Historically pubs were owner operated. However, leasing pubs has become very common in recent years. Some years ago, the Vintners association commissioned a project (that remains unfinished) to design a database application that would capture details of pub leasing and the level of activity in the sector. Key details captured by systems analysts include:

* A landlord may own many pubs and each pub is owned by a single landlord
* A publican may participate in many lease agreements and a lease agreement involves a single publican
* A letting agent (auctioneer) may arrange many lease agreements and a lease agreement involves a single letting agent
* A solicitor may draw up many lease agreements and a lease agreement is drawn up by a single solicitor

Lease\_Agreement

Lease\_Agreement\_Id

Pub\_Id

Publican\_Id

Start\_Date

End\_Date

Deposit

Instalment\_Frequency

Payment\_Method

Installment\_Amount

Letting\_Agent\_Id

Solicitor\_Id

Solicitor

Solicitor\_Id

Fname

Sname

Address

County

Gender

Date\_Of\_Birth

Mobile\_No

Email\_Address

Qualifications

Letting\_Agent

Letting\_Agent\_Id

Name

City

County

Specialism

Web\_Site

Email

Year\_Founded

Annual\_Turnover

Reputation

Publican

Publican\_Id

Fname

Sname

Address

County

Gender

Date\_Of\_Birth

Mobile\_No

Email\_Address

Qualifications

Specialism

Landlord

Landlord\_Id

FName

Sname

Address

Town

County

Tel\_No

Email

Specialism

Pub

Pub\_Id

Name

Address

City

Country

Tel\_No

Email

Capacity

Year\_Built

Sq\_Metres

Customer\_Profile

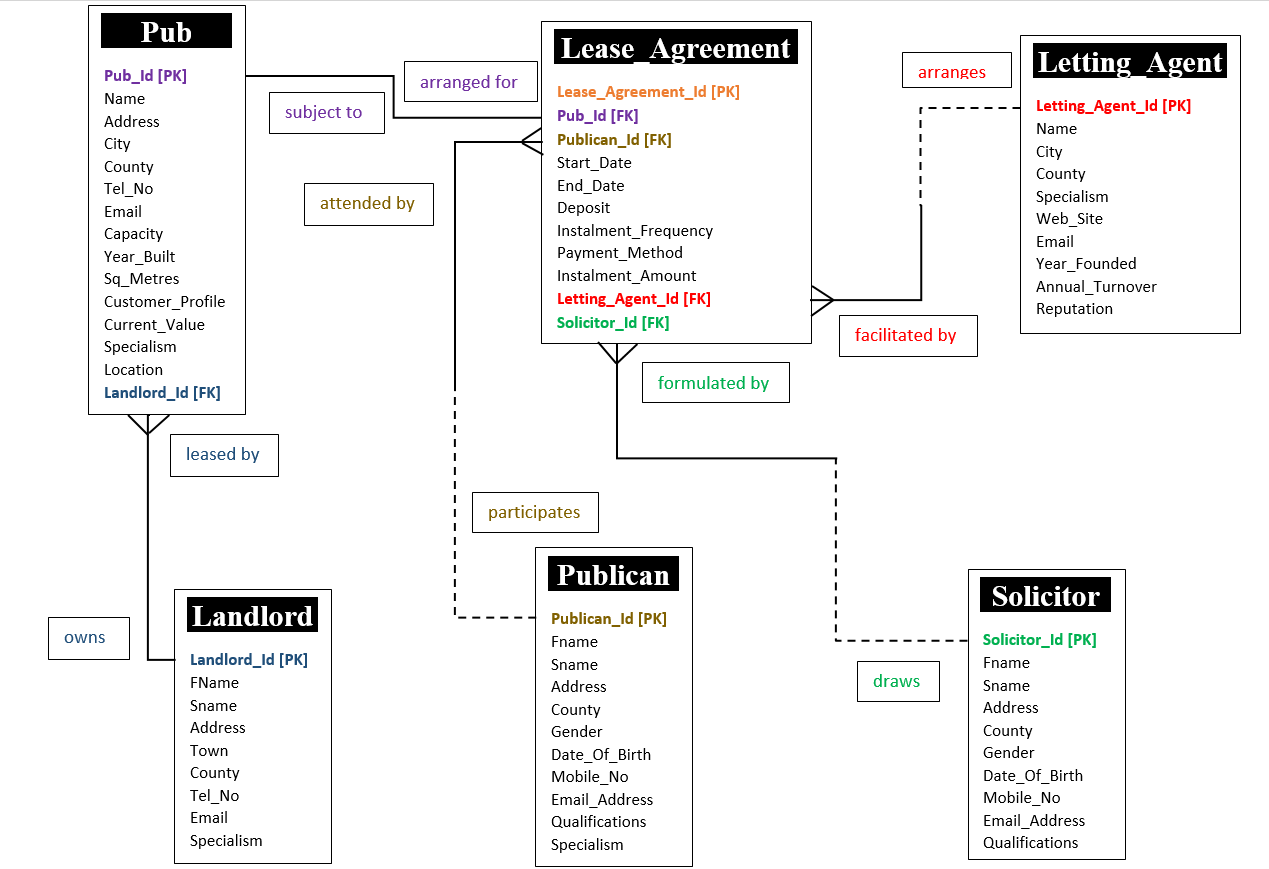
Current\_Value

Specialism

Location

Landlord\_Id

# ENTITY RELATIONSHIP DIAGRAM



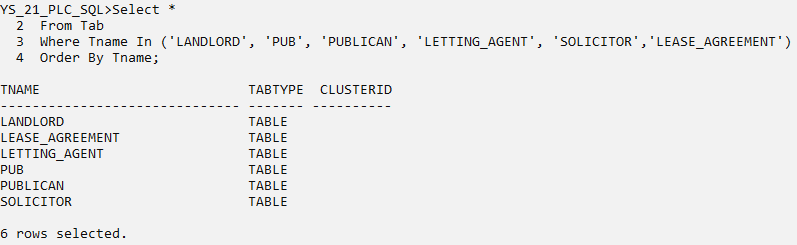
# TABLES

Select \*

From Tab

Where Tname In ('LANDLORD', 'PUB', 'PUBLICAN', 'LETTING\_AGENT', 'SOLICITOR','LEASE\_AGREEMENT')

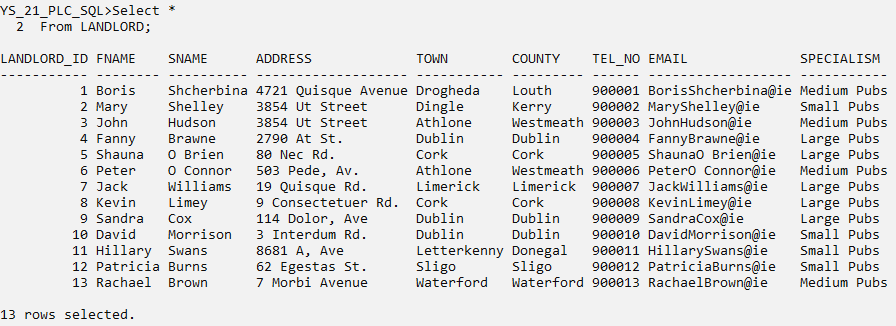
Order By Tname;



* Landlord

Select \*

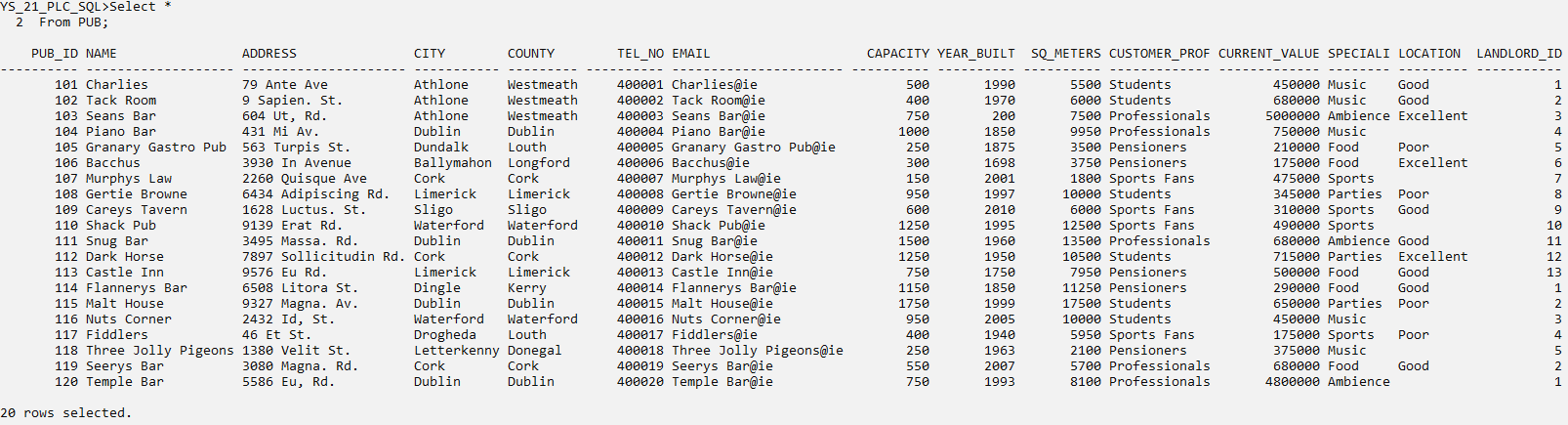
From LANDLORD;



* Pub

Select \*

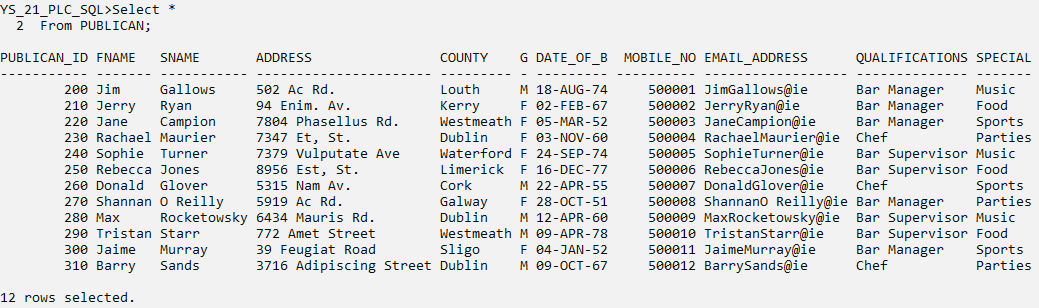
From PUB;



* Publican

Select \*

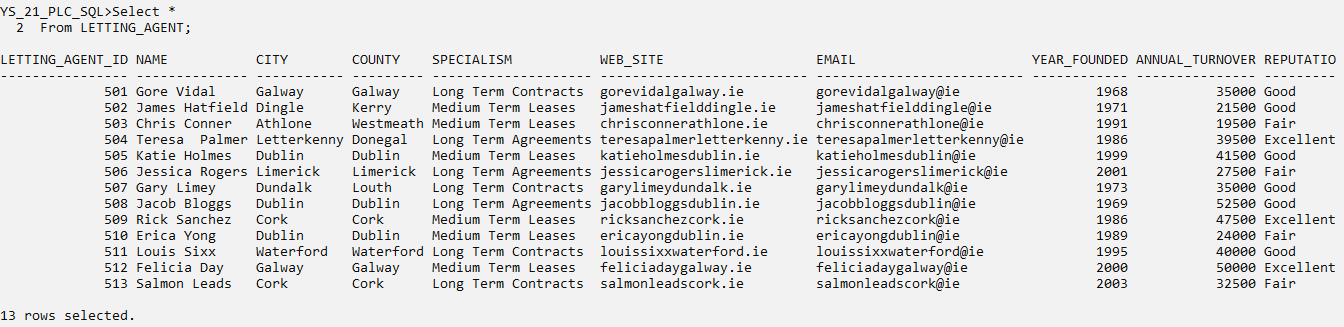
From PUBLICAN;



* Letting Agent

Select \*

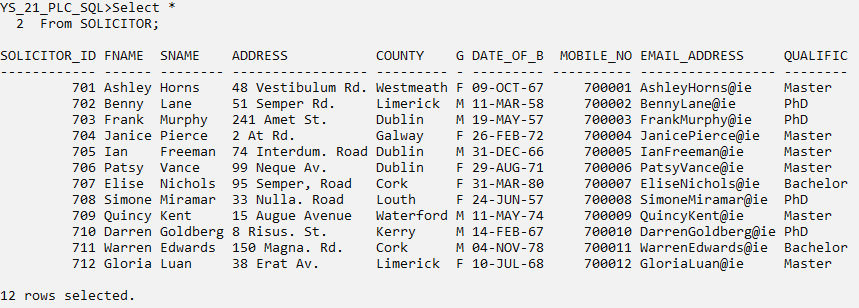
From LETTING\_AGENT;



* Solicitor

Select \*

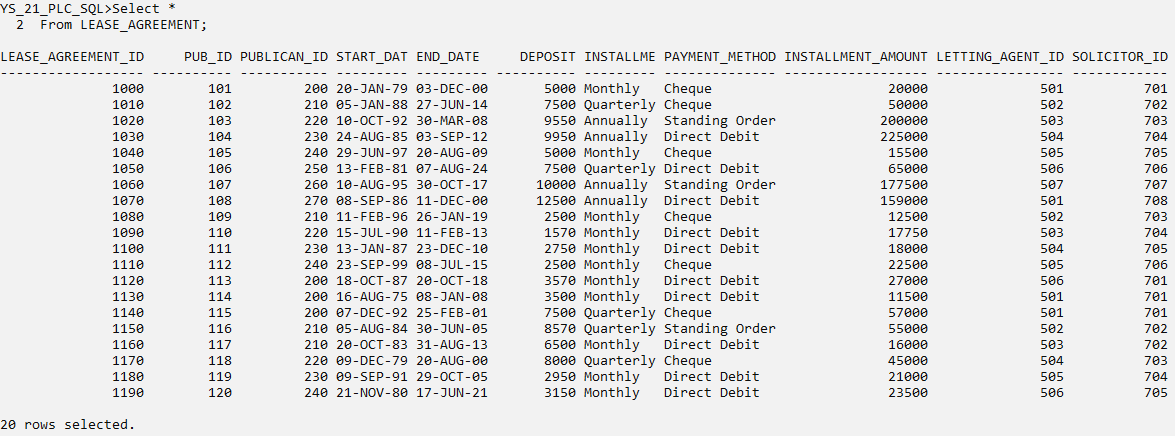
From SOLICITOR;



* Lease Agreement

Select \*

From LEASE\_AGREEMENT;



# TRIGGERS

Set Linesize 160

Set SQLPrompt YS\_21SQL>

## **5.1 Before Insert On Lease\_Agreement**

Question.1

Build an application called ‘Lease Pay Method Tracker’ that will help process to add new lease agreements and adding their details to the Lease Agreement table in the database. The implementation requirements are as follows:

1. Lease Agreement payed by Direct Debit may be added as an agreement provided the Installment Frequency is Monthly.
2. Lease Agreement payed by Cheque may be added as an agreement provided the corresponding Pub’s Customer Profile is Students.

Solution

Cl Scr

Set Serveroutput On Size 1000000

Create Or Replace Trigger Lease\_Pay\_Method\_Tracker

Before Insert On Lease\_Agreement

For Each Row

Declare

V\_Cust\_Profile Pub.Customer\_Profile%Type;

Begin

If :New.Payment\_Method = 'Direct Debit' And :New.Installment\_frequency <> 'Monthly' Then

Raise\_Application\_Error(-20101, 'A - New Payment Must Be Paid Monthly');

End If;

If :New.Payment\_Method = 'Cheque' Then

Select Customer\_Profile Into V\_Cust\_Profile

From Pub

Where Pub.Pub\_Id = :New.Pub\_Id;

If V\_Cust\_Profile <> 'Students' Then

Raise\_Application\_Error (-20101, 'B - New Lease Agreement Customer Profile For Pub Must be Students');

End If;

End If;

End;

/

Show Err

Insert Into Lease\_Agreement (Lease\_Agreement\_Id, Pub\_Id, Publican\_Id, Start\_Date, End\_Date, Deposit, Installment\_frequency, Payment\_Method, Installment\_Amount, Letting\_Agent\_Id, Solicitor\_Id) Values (1200, 120, 240, '15-Oct-19', '14-Oct-20', 7500, 'Quarterly', 'Direct Debit', 55000, 505, 705);

Insert Into Lease\_Agreement (Lease\_Agreement\_Id, Pub\_Id, Publican\_Id, Start\_Date, End\_Date, Deposit, Installment\_frequency, Payment\_Method, Installment\_Amount, Letting\_Agent\_Id, Solicitor\_Id) Values (1200, 120, 240, '15-Oct-19', '14-Oct-20', 7500, 'Monthly', 'Direct Debit', 55000, 505, 705);

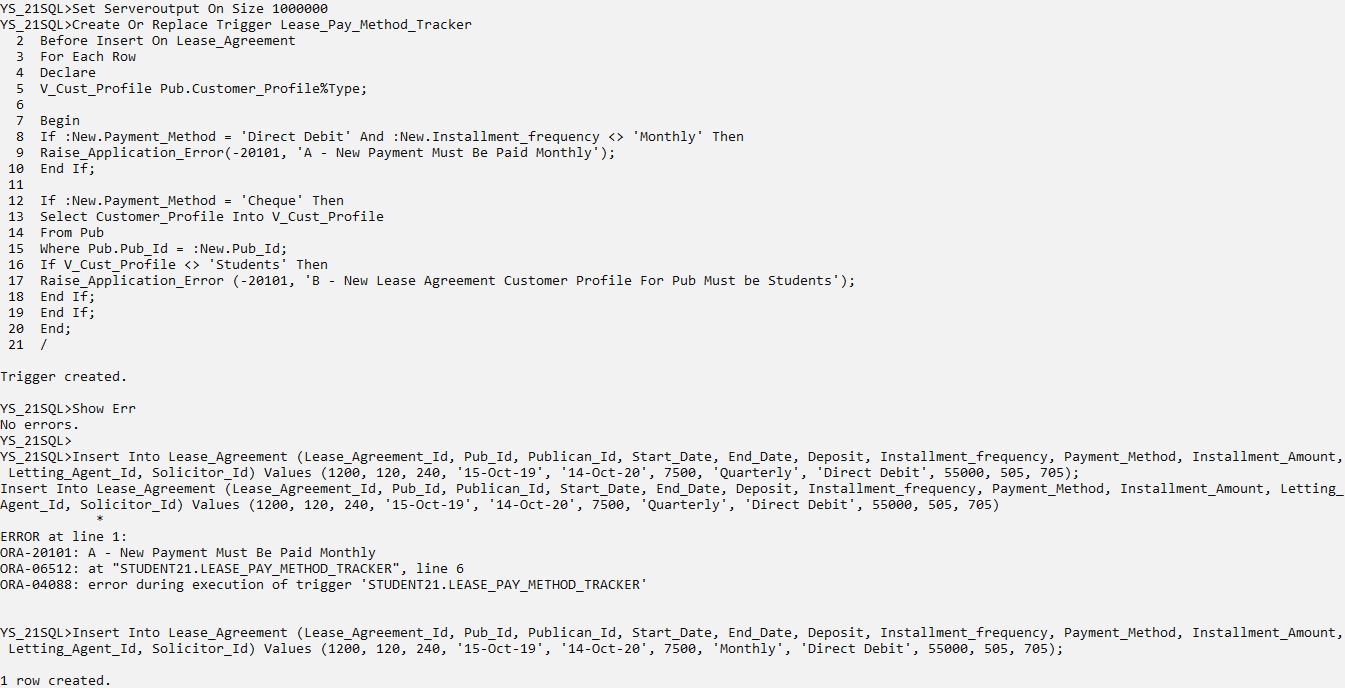
Insert Into Lease\_Agreement (Lease\_Agreement\_Id, Pub\_Id, Publican\_Id, Start\_Date, End\_Date, Deposit, Installment\_frequency, Payment\_Method, Installment\_Amount, Letting\_Agent\_Id, Solicitor\_Id) Values (1201, 103, 210, '15-May-19', '14-May-20', 9500, 'Annually', 'Cheque', 25000, 505, 705);

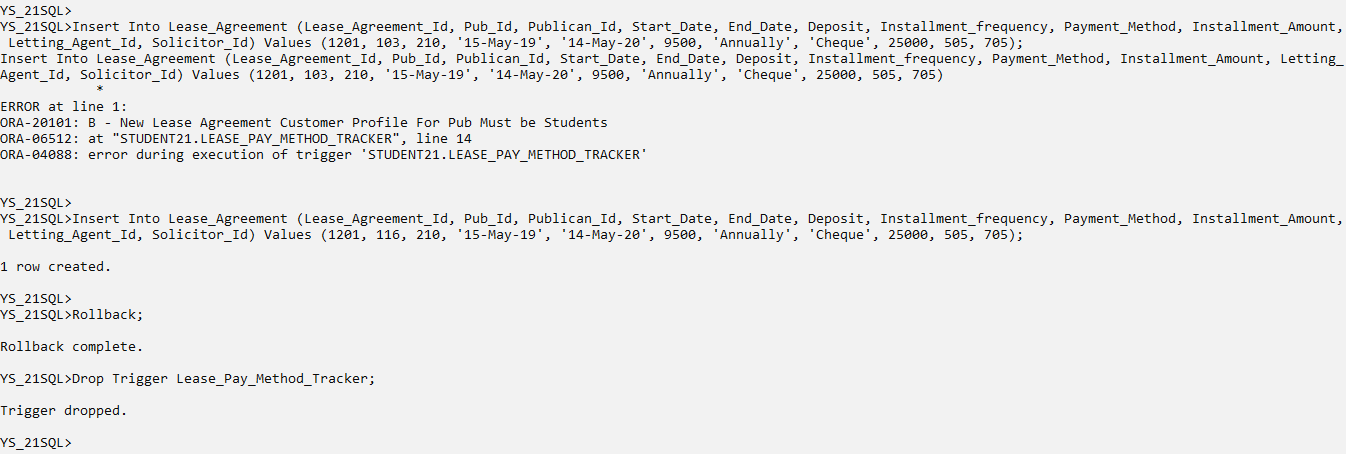
Insert Into Lease\_Agreement (Lease\_Agreement\_Id, Pub\_Id, Publican\_Id, Start\_Date, End\_Date, Deposit, Installment\_frequency, Payment\_Method, Installment\_Amount, Letting\_Agent\_Id, Solicitor\_Id) Values (1201, 116, 210, '15-May-19', '14-May-20', 9500, 'Annually', 'Cheque', 25000, 505, 705);

Rollback;

Drop Trigger Lease\_Pay\_Method\_Tracker;

Snip





## **5.2 Before Delete On Pub**

Question.2

Build an application called Pub\_Del\_Checker that will implement the requirements outlined below regarding the removal of Pubs from the database:

1. Pub whose County is Westmeath may not be removed if it’s Landlord is in County Westmeath.
2. Pub whose Town is Athlone may not be removed if their Customers are Students.

Solution

Cl Scr

Set Serveroutput On Size 1000000

Create Or Replace Trigger Pub\_Del\_Checker

Before Delete On Pub

For Each Row

Declare

V\_Landlord\_Loc Landlord.County%Type;

Begin

If :Old.County = ’Westmeath’ Then

Select County Into V\_Landlord\_Loc

From Landlord

Where Landlord.Landlord\_Id = :Old.Landlord\_Id;

If V\_Landlord\_Loc = ’Westmeath’ Then

Raise\_Application\_Error (-20101, ’A – Westmeath Pub whose Landlord is in Westmeath may not deleted’);

End If;

End If;

If :Old.City = ’Athlone’ And :Old.Customer\_Profile = ’Students’ Then

Raise\_Application\_Error (-20101, ’B – Athlonian Pub may not be removed if their customers are Students’);

End If;

End;

/

Show Err

Delete

From Pub

Where Pub\_Id = 103;

Alter Table Lease\_Agreement Disable Constraint Lease\_Agreement\_Pub\_Id\_Fk;

Delete

From Pub

Where Pub\_Id = 106;

Delete

From Pub

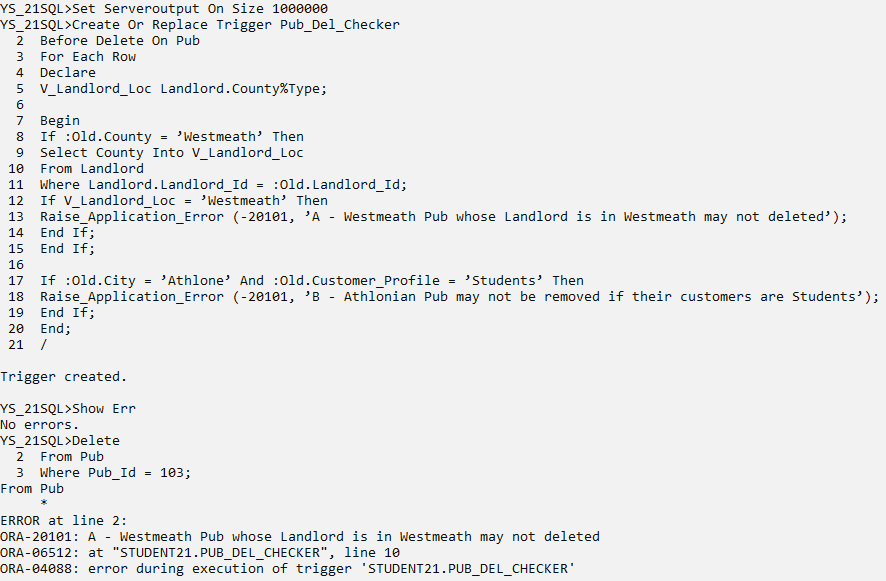
Where Pub\_Id = 101;

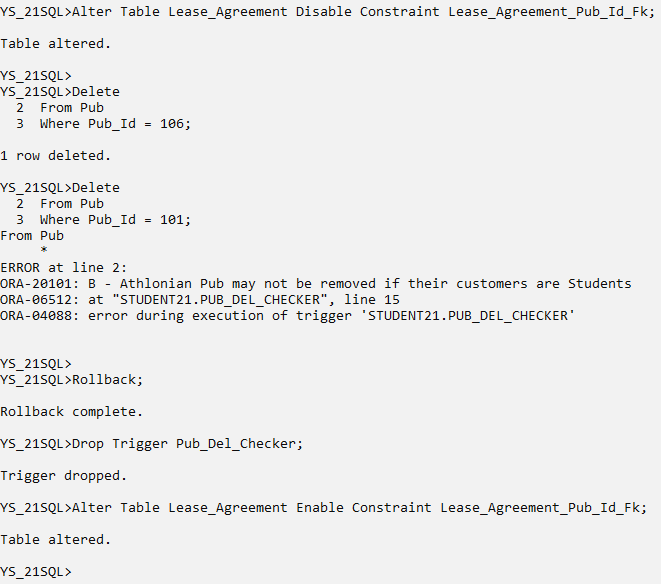
Rollback;

Drop Trigger Pub\_Del\_Checker;

Alter Table Lease\_Agreement Enable Constraint Lease\_Agreement\_Pub\_Id\_Fk;

Snip





## **5.3 After Insert On Letting\_Agent**

Question.3

Build an application called Lease\_Completion that will help finalise the leasing process when Pubs are built and their details are added to the Pub table. The requirements that need to be implemented are as follows:

1. If the Annual\_Turnover of the Letting Agent of the new Lease Agreement will be €30k or lower then the turnover will be increased by 20%.
2. If the Annual\_Turnover of the Letting Agent of the new Lease Agreement is above €30k then the turnover will be increased by 10%.
3. Display the Annual\_Turnover of the Letting Agent of the new Lease Agreement before and after the turnover is adjusted.

Solution

Cl Scr

Set Serveroutput On Size 1000000

Create Or Replace Trigger Lease\_Completion

After Insert On Lease\_Agreement

For Each Row

Declare

V\_Turnover Letting\_Agent.Annual\_Turnover%Type;

Begin

Select Annual\_Turnover Into V\_Turnover

From Letting\_Agent

Where Letting\_Agent.Letting\_Agent\_Id = :New.Letting\_Agent\_Id;

Dbms\_Output.Put\_Line (’Turnover of Agent ’ || :New.Letting\_Agent\_Id || ’ Before Processing Of ’ || :New.Lease\_Agreement\_Id || ’ Was Found To Be €’ || V\_Turnover);

If V\_Turnover <= 30000 Then

Update Letting\_Agent

Set Annual\_Turnover = Annual\_Turnover \* 1.2

Where Letting\_Agent.Letting\_Agent\_Id = :New.Letting\_Agent\_Id;

Elsif V\_Turnover > 30000 Then

Update Letting\_Agent

Set Annual\_Turnover = Annual\_Turnover \* 1.1

Where Letting\_Agent.Letting\_Agent\_Id = :New.Letting\_Agent\_Id;

End If;

Select Annual\_Turnover Into V\_Turnover

From Letting\_Agent

Where Letting\_Agent.Letting\_Agent\_Id = :New.Letting\_Agent\_Id;

Dbms\_Output.Put\_Line (’Turnover of Agent ’ || :New.Letting\_Agent\_Id || ’ After Processing Of ’ || :New.Lease\_Agreement\_Id || ’ Was Adjusted To €’ || V\_Turnover);

End;

/

Sho Err

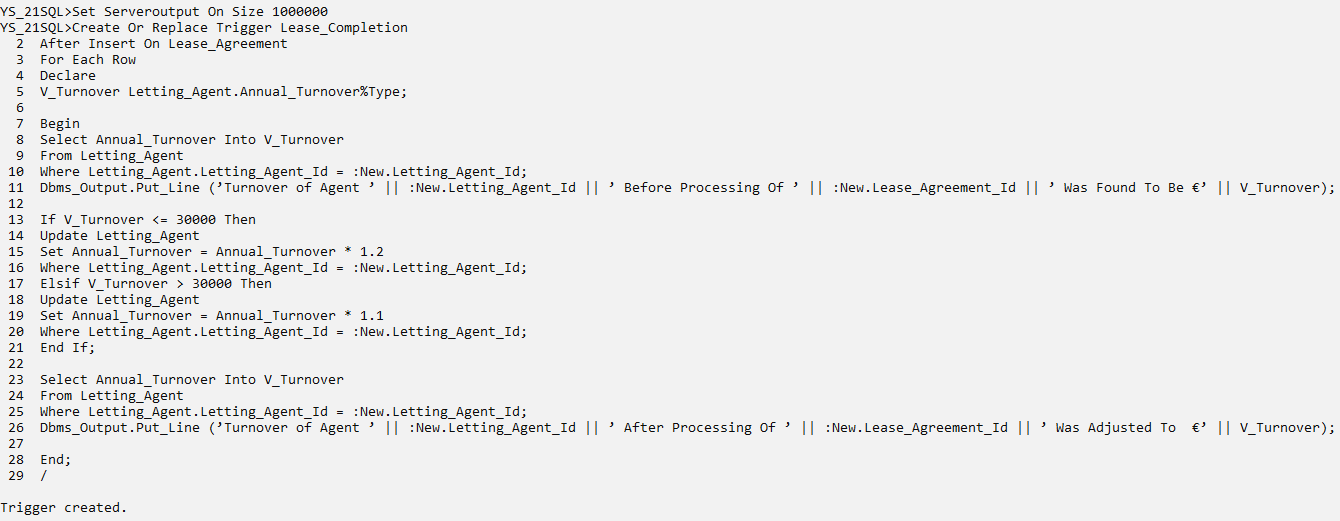
Insert Into Lease\_Agreement (Lease\_Agreement\_Id, Start\_Date, End\_Date, Installment\_frequency, Payment\_Method, Installment\_Amount, Letting\_Agent\_Id) Values (1200, ’11-Feb-19’, ’10-Feb-20’, ’Quarterly’, ’Standing Order’, 100000, 502);

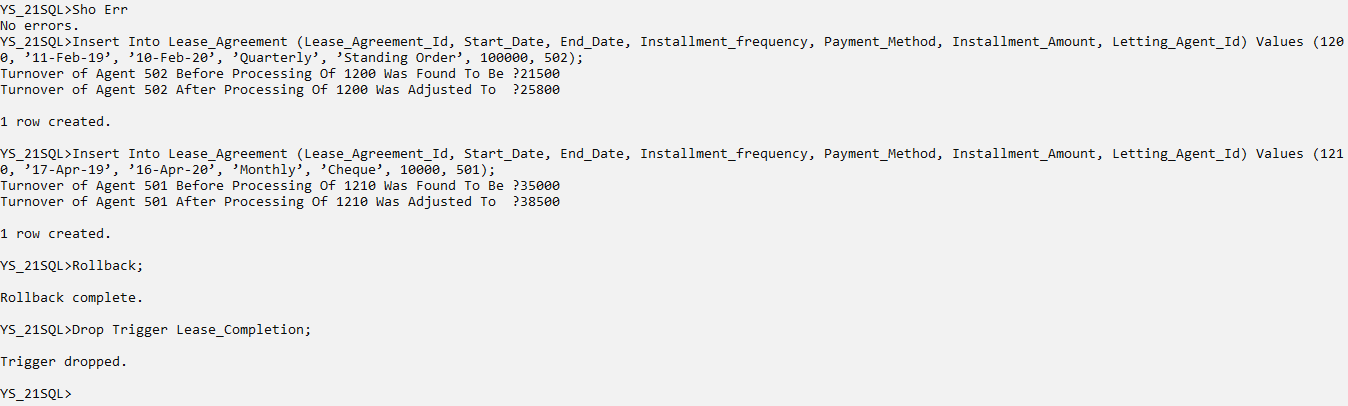
Insert Into Lease\_Agreement (Lease\_Agreement\_Id, Start\_Date, End\_Date, Installment\_frequency, Payment\_Method, Installment\_Amount, Letting\_Agent\_Id) Values (1210, ’17-Apr-19’, ’16-Apr-20’, ’Monthly’, ’Cheque’, 10000, 501);

Rollback;

Drop Trigger Lease\_Completion;

Snip





## **5.4 After Update On Lease\_Agreement**

Question.4

When a user (successfully) makes changes to the details of a Lease\_Agreement, some changes need be carried out on data in other tables. The details are:

1. If the Deposit for a Lease\_Agreement has increased, then the Capacity of corresponding Pub will increase by 10% if the Pub is located in County Dublin.
2. If a Lease\_Agreement changes Installment\_frequency (but doesn’t change deposit), then the Capacity of Pub will increase by 20% if the Pub is located in County Cork.
3. Display the Lease\_Agreement Id and the Location of the corresponding Pub.
4. In a table called Lease\_Update\_Log a record should be kept of:
5. Name of the user who added the row.
6. Date of update and
7. Lease\_Agreement\_Id of the agreement that will automatically implement the above requirements each time a row is added to the table.

Solution

Cl Scr

Set Serveroutput On Size 1000000

Drop Table Lease\_Update\_Log;

Create Table Lease\_Update\_Log (Oracle\_User Varchar2(10), System\_Date Date, Lease\_Num Number(4), Pub\_Loc Varchar2(10));

Create Or Replace Trigger Updated\_Lease\_Completion

After Update On Lease\_Agreement

For Each Row

Declare

V\_Location Pub.City%Type;

Begin

If :New.Deposit > :Old.Deposit Then

Select City Into V\_Location

From Pub

Where Pub.Pub\_Id = :New.Pub\_Id;

Dbms\_Output.Put\_Line ('Pub In ' || V\_Location || ' Is Where ' || :New.Lease\_Agreement\_Id || ' Is Based');

If V\_Location = 'Dublin' Then

Update Pub

Set Capacity = Capacity \* 1.1

Where Pub.Pub\_Id = :New.Pub\_Id;

End If;

End If;

If :New.Installment\_frequency <> :Old.Installment\_frequency And :New.Deposit = :Old.Deposit Then

Select City Into V\_Location

From Pub

Where Pub.Pub\_Id = :New.Pub\_Id;

Dbms\_Output.Put\_Line ('Pub In ' || V\_Location || ' Is Where ' || :New.Lease\_Agreement\_Id || ' Is Based');

If V\_Location = 'Cork' Then

Update Pub

Set Capacity = Capacity \* 1.2

Where Pub.Pub\_Id = :New.Pub\_Id;

End If;

End If;

Insert Into Lease\_Update\_Log Values (User, Sysdate, :New.Lease\_Agreement\_Id, V\_Location);

End;

/

Sho Err

/\* Increase Pub capacity by 10% if it’s deposit in the agreement has been increased and is in County Dublin\*/

Select Pub\_Id, Name, Capacity

From Pub

Where Pub\_Id = 104;

Update Lease\_Agreement

Set Deposit = Deposit + 100

Where Lease\_Agreement\_Id = 1030;

Select Pub\_Id, Name, Capacity

From Pub

Where Pub\_Id = 104;

Select \* From Lease\_Update\_Log;

/\* Increase Pub capacity by 20% if it’s Installment frequency has been changed to quarterly in the agreement has been increased and is in County Cork\*/

Select Pub\_Id, Name, Capacity

From Pub

Where Pub\_Id = 107;

Update Lease\_Agreement

Set Installment\_frequency = 'Quarterly'

Where Lease\_Agreement\_Id = 1060;

Select Pub\_Id, Name, Capacity

From Pub

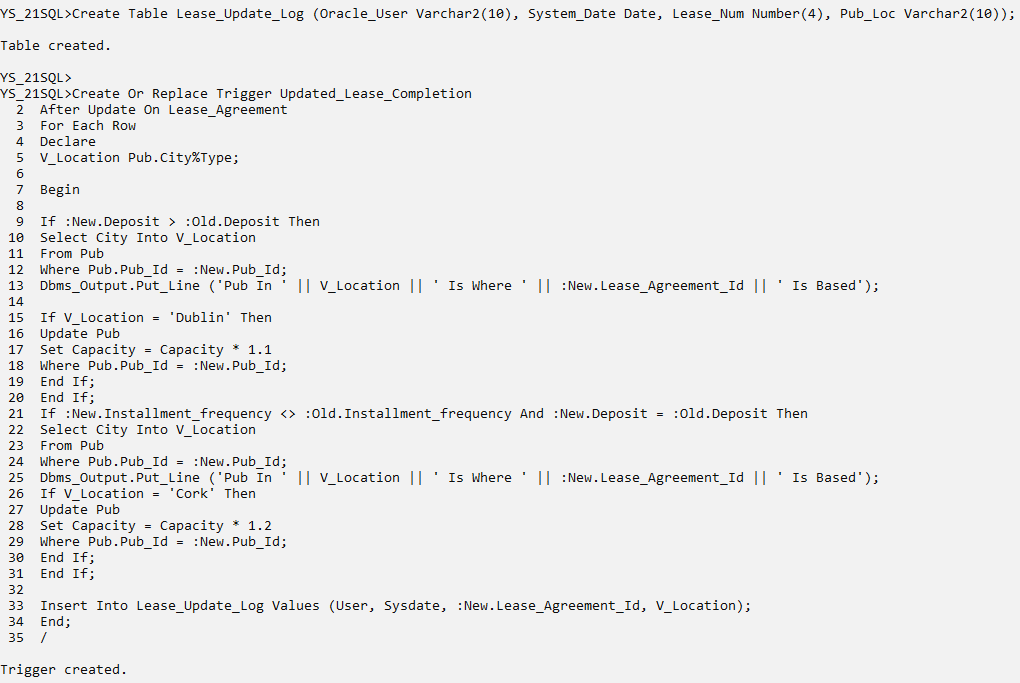
Where Pub\_Id = 107;

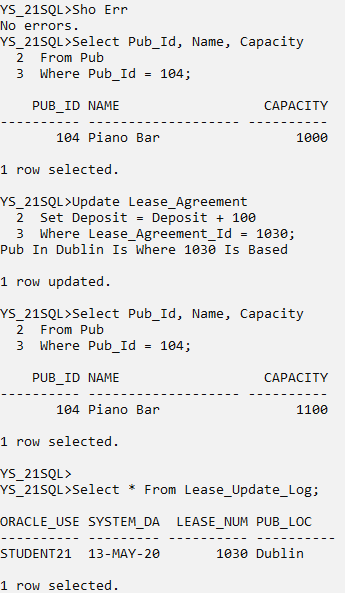
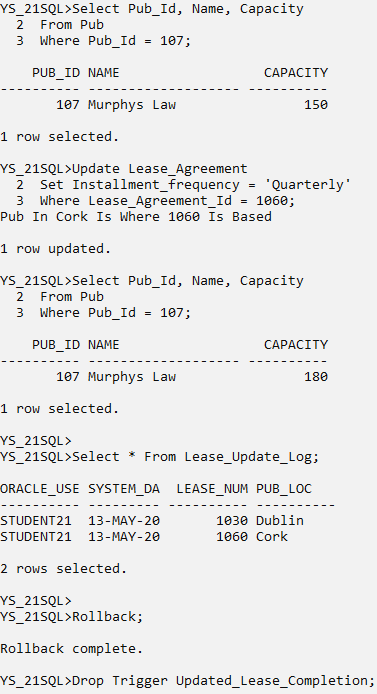
Select \* From Lease\_Update\_Log;

Rollback;

Drop Trigger Updated\_Lease\_Completion;

Snip





## **5.5 After Delete On Pub**

Question.5

When a user (successfully) removes a Lease Agreement, some changes need be carried out on data in other tables. The details are:

1. If the Installment frequency of a Lease Agreement is Monthly, then the Current Value of the corresponding Pub will decrease by 20% if it’s for Sports Fans.
2. If the Installment frequency of a Lease Agreement is Quarterly or Annually then the Current Value of the corresponding Pub will decrease by 10% if it’s for Students.
3. Display the Lease\_Agreement\_Id which has been deleted and the City of Pub on which the agreement is associated.
4. In a table called Lease\_Delete\_Log a record should be kept of:
5. Date of the removal,
6. Lease\_Agreement\_Id,
7. Start Date of the Agreement that’s just been deleted, and
8. Installment\_frequency of the corresponding Pub

Solution

Cl Scr

Set Serveroutput On Size 1000000

Drop Table Lease\_Delete\_Log;

Create Table Lease\_Delete\_Log (Exe\_Date Date, Lease\_Num Number(4), Starting\_From Date, Inst\_Freq Varchar2(15));

Create Or Replace Trigger Del\_Lease\_Completion

After Delete On Lease\_Agreement

For Each Row

Declare

V\_Pub\_Special Pub.Specialism%Type;

Begin

If :Old.Installment\_frequency In ('Monthly') Then

Select Specialism Into V\_Pub\_Special

From Pub

Where Pub.Pub\_Id = :Old.Pub\_Id;

If V\_Pub\_Special = 'Sports Fans' Then

Update Pub

Set Current\_Value = Current\_Value\*0.9

Where Pub.Pub\_Id = :Old.Pub\_Id;

End If;

End If;

If :Old.Installment\_frequency In ('Quarterly', 'Annually') Then

Select Specialism Into V\_Pub\_Special

From Pub

Where Pub.Pub\_Id = :Old.Pub\_Id;

If V\_Pub\_Special = 'Students' Then

Update Pub

Set Current\_Value = Current\_Value\*0.8

Where Pub.Pub\_Id = :Old.Pub\_Id;

End If;

End If;

Dbms\_Output.Put\_Line ('Pub Specialism Is ' || V\_Pub\_Special || ' Is On What Agreement ' || :Old.Lease\_Agreement\_Id || ' Was Made. ');

Insert Into Lease\_Delete\_Log Values (Sysdate, :Old.Lease\_Agreement\_Id, :Old.Start\_Date, V\_Pub\_Special);

End;

/

Sho Err

Select Pub\_Id, Specialism, Current\_Value

From Pub

Where Pub\_Id = 109;

Delete

From Lease\_Agreement

Where Lease\_Agreement\_Id = 1080;

Select Pub\_Id, Specialism, Current\_Value

From Pub

Where Pub\_Id = 109;

Select \*

From Lease\_Delete\_Log;

Select Pub\_Id, Specialism, Current\_Value

From Pub

Where Pub\_Id = 103;

Delete

From Lease\_Agreement

Where Lease\_Agreement\_Id = 1020;

Select Pub\_Id, Specialism, Current\_Value

From Pub

Where Pub\_Id = 103;

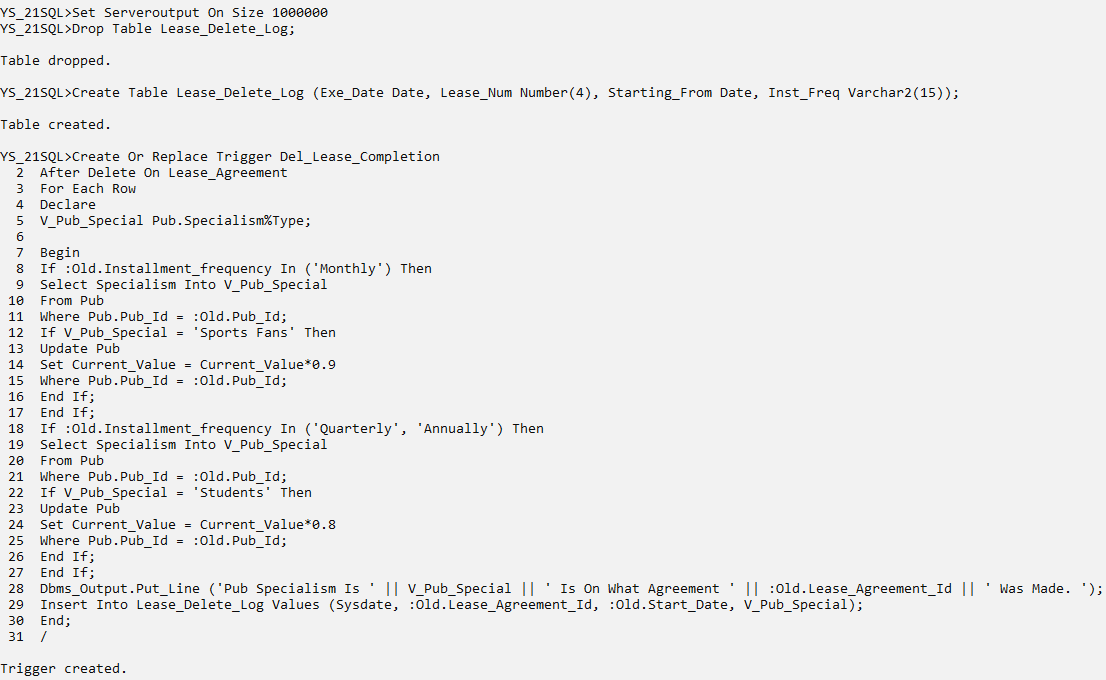
Select \*

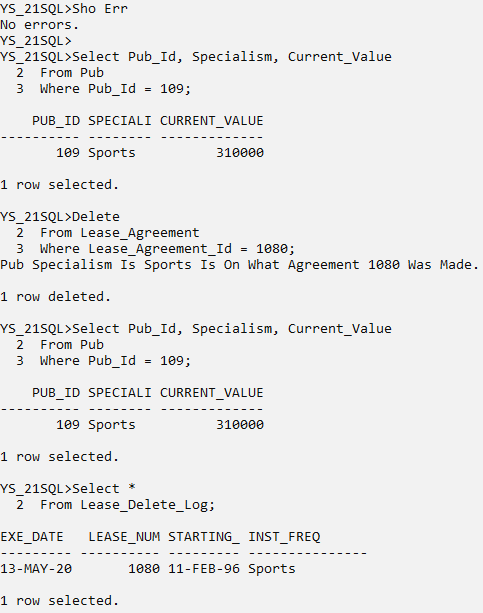
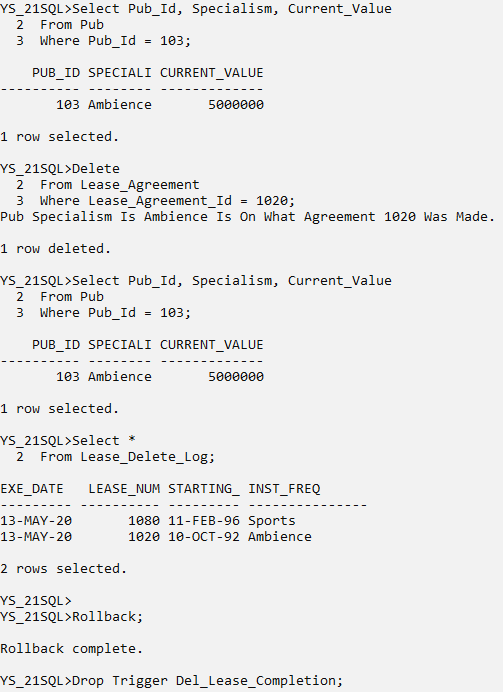
From Lease\_Delete\_Log;

Rollback;

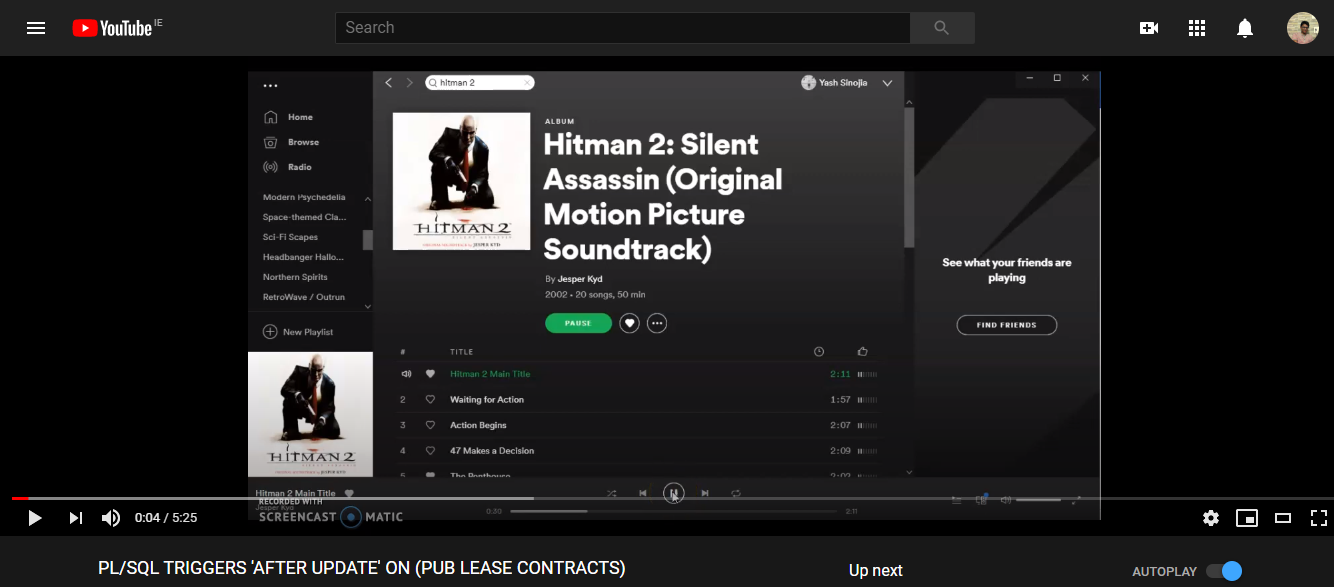
Drop Trigger Del\_Lease\_Completion;

Snip





# YouTube Description



[Link to Trigger Screencast](https://youtu.be/fCOGQZgRfsE)