DDL Output

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Debashis Jena

UMGC

SQL> SET ECHO ON

SQL> SET SERVEROUTPUT ON;

SQL> DROP VIEW Get\_BranchInfo\_View;

View GET\_BRANCHINFO\_VIEW dropped.

SQL> DROP VIEW Get\_Report\_View;

View GET\_REPORT\_VIEW dropped.

SQL> DROP TRIGGER CustomerLSequence\_Trigger;

Error starting at line : 5 in command -

DROP TRIGGER CustomerLSequence\_Trigger

Error report -

ORA-04080: trigger 'CUSTOMERLSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL> DROP TRIGGER LibrarySequence\_Trigger;

Error starting at line : 6 in command -

DROP TRIGGER LibrarySequence\_Trigger

Error report -

ORA-04080: trigger 'LIBRARYSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL> DROP TRIGGER LibraryCustomerSequence\_Trigger;

Error starting at line : 7 in command -

DROP TRIGGER LibraryCustomerSequence\_Trigger

Error report -

ORA-04080: trigger 'LIBRARYCUSTOMERSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL> DROP TRIGGER BranchSequence\_Trigger;

Error starting at line : 8 in command -

DROP TRIGGER BranchSequence\_Trigger

Error report -

ORA-04080: trigger 'BRANCHSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL> DROP TRIGGER CatalogSequence\_Trigger;

Error starting at line : 9 in command -

DROP TRIGGER CatalogSequence\_Trigger

Error report -

ORA-04080: trigger 'CATALOGSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL> DROP TRIGGER BranchItemSequence\_Trigger;

Error starting at line : 10 in command -

DROP TRIGGER BranchItemSequence\_Trigger

Error report -

ORA-04080: trigger 'BRANCHITEMSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL> DROP TRIGGER TransactionSequence\_Trigger;

Error starting at line : 11 in command -

DROP TRIGGER TransactionSequence\_Trigger

Error report -

ORA-04080: trigger 'TRANSACTIONSEQUENCE\_TRIGGER' does not exist

04080. 00000 - "trigger '%s' does not exist"

\*Cause: The TRIGGER name is invalid.

\*Action: Check the trigger name.

SQL>

SQL> DROP TABLE Library CASCADE CONSTRAINTS;

Table LIBRARY dropped.

SQL> DROP TABLE Customer CASCADE CONSTRAINTS;

Table CUSTOMER dropped.

SQL> DROP TABLE LibraryCustomer CASCADE CONSTRAINTS;

Table LIBRARYCUSTOMER dropped.

SQL> DROP TABLE Branch CASCADE CONSTRAINTS;

Table BRANCH dropped.

SQL> DROP TABLE Catalog CASCADE CONSTRAINTS;

Table CATALOG dropped.

SQL> DROP TABLE BranchItem CASCADE CONSTRAINTS;

Table BRANCHITEM dropped.

SQL> DROP TABLE Transaction CASCADE CONSTRAINTS;

Table TRANSACTION dropped.

SQL>

SQL>

SQL> PROMPT === Drop Sequences ===

=== Drop Sequences ===

SQL> DROP SEQUENCE LibraryId\_Seq;

Sequence LIBRARYID\_SEQ dropped.

SQL> DROP SEQUENCE LibraryCustomerId\_Seq;

Sequence LIBRARYCUSTOMERID\_SEQ dropped.

SQL> DROP SEQUENCE BranchId\_Seq;

Sequence BRANCHID\_SEQ dropped.

SQL> DROP SEQUENCE CustomerLId\_Seq;

Sequence CUSTOMERLID\_SEQ dropped.

SQL> DROP SEQUENCE BranchItemId\_Seq;

Sequence BRANCHITEMID\_SEQ dropped.

SQL> DROP SEQUENCE TransactionId\_Seq;

Sequence TRANSACTIONID\_SEQ dropped.

SQL> DROP SEQUENCE CatalogId\_Seq;

Sequence CATALOGID\_SEQ dropped.

SQL>

SQL> /\*\* Creating Tables \*\*/

SQL> PROMPT Creating Table 'Library'

Creating Table 'Library'

SQL> CREATE TABLE LIBRARY

2 (

3 LIBRARYID NUMBER(8,0) NOT NULL

4 , LIBRARYNAME VARCHAR2(50) NOT NULL

5 , PHONE VARCHAR2(15)

6 , ADDRESS1 VARCHAR2(50)

7 , ADDRESS2 VARCHAR2(50)

8 , CITY VARCHAR2(50)

9 , STATE VARCHAR2(50)

10 , ZIPCODE VARCHAR2(10)

11 , CREATEDBY VARCHAR2(50) NOT NULL

12 , CREATEDDATE DATE NOT NULL

13 , MODIFIEDBY DATE NOT NULL

14 , MODIFIEDDATE DATE

15 );

Table LIBRARY created.

SQL> PROMPT Creating Table 'Customer'

Creating Table 'Customer'

SQL> CREATE TABLE CUSTOMER

2 (

3 CUSTOMERID NUMBER(8,0) NOT NULL

4 ,FIRSTNAME VARCHAR2(32) NOT NULL

5 ,LASTNAME VARCHAR2(32) NOT NULL

6 ,ADDRESS1 VARCHAR2(50)

7 ,ADDRESS2 VARCHAR2(50)

8 ,CITY VARCHAR2(50)

9 ,STATE VARCHAR2(50)

10 ,ZIPCODE VARCHAR2(10)

11 ,CREATEDBY VARCHAR2(30) NOT NULL

12 ,CREATEDDATE DATE NOT NULL

13 ,MODIFIEDBY VARCHAR2(30) NOT NULL

14 ,MODIFIEDDATE DATE NOT NULL

15 );

Table CUSTOMER created.

SQL>

SQL> PROMPT CREATING TABLE 'LIBRARYCUSTOMER'

CREATING TABLE 'LIBRARYCUSTOMER'

SQL> CREATE TABLE LIBRARYCUSTOMER

2 (

3 LIBRARYCUSTOMERID NUMBER(8,0) NOT NULL

4 ,LIBRARYID NUMBER(8,0) NOT NULL

5 ,CUSTOMERID NUMBER(8,0) NOT NULL

6 ,LIBRARYCARDNUMBER INTEGER

7 ,LIBRARYCARDPIN INTEGER

8 ,LIBRARYCARDEXPIRY DATE

9 ,CREATEDBY VARCHAR2(30) NOT NULL

10 ,CREATEDDATE DATE NOT NULL

11 ,MODIFIEDBY VARCHAR2(30) NOT NULL

12 ,MODIFIEDDATE DATE NOT NULL

13 );

Table LIBRARYCUSTOMER created.

SQL>

SQL> PROMPT CREATING TABLE 'BRANCH'

CREATING TABLE 'BRANCH'

SQL> CREATE TABLE BRANCH

2 (

3 BRANCHID NUMBER(8,0) NOT NULL

4 ,LIBRARYID NUMBER(8,0) NOT NULL

5 ,BRANCHNAME VARCHAR2(32)

6 ,BRANCHPHONE VARCHAR2(32)

7 ,ADDRESS1 VARCHAR2(1000)

8 ,ADDRESS2 VARCHAR2(50)

9 ,CITY VARCHAR2(50)

10 ,STATE VARCHAR2(50)

11 ,ZIPCODE VARCHAR2(10)

12 ,CREATEDBY VARCHAR2(30) NOT NULL

13 ,CREATEDDATE DATE NOT NULL

14 ,MODIFIEDBY VARCHAR2(30) NOT NULL

15 ,MODIFIEDDATE DATE NOT NULL

16 );

Table BRANCH created.

SQL>

SQL> PROMPT CREATING TABLE 'CATALOG'

CREATING TABLE 'CATALOG'

SQL> CREATE TABLE CATALOG

2 (

3 CATALOGID NUMBER(8,0) NOT NULL

4 ,TITLE VARCHAR2(100) NOT NULL

5 ,DESCRIPTION VARCHAR2(500)

6 ,PUBLISHER VARCHAR2(100)

7 ,RELEASEDATE DATE

8 ,TYPE VARCHAR2(32)

9 ,BOOKISBN VARCHAR2(32)

10 ,BOOKPAGES VARCHAR2(32)

11 ,DVDLENGTH VARCHAR2(10)

12 ,CREATEDBY VARCHAR2(30) NOT NULL

13 ,CREATEDDATE DATE NOT NULL

14 ,MODIFIEDBY VARCHAR2(30) NOT NULL

15 ,MODIFIEDDATE DATE NOT NULL

16 );

Table CATALOG created.

SQL>

SQL> PROMPT CREATING TABLE 'BRANCHITEM'

CREATING TABLE 'BRANCHITEM'

SQL> CREATE TABLE BRANCHITEM

2 (

3 BRANCHITEMID NUMBER(8,0) NOT NULL

4 ,BRANCHID NUMBER(8,0) NOT NULL

5 ,CATALOGID NUMBER(8,0) NOT NULL

6 ,COPYNUMBER INTEGER

7 ,PURCHASEDATE DATE

8 ,CREATEDBY VARCHAR2(30) NOT NULL

9 ,CREATEDDATE DATE NOT NULL

10 ,MODIFIEDBY VARCHAR2(30) NOT NULL

11 ,MODIFIEDDATE DATE NOT NULL

12 );

Table BRANCHITEM created.

SQL>

SQL> PROMPT CREATING TABLE 'TRANSACTION'

CREATING TABLE 'TRANSACTION'

SQL> CREATE TABLE TRANSACTION

2 (

3 TRANSACTIONID NUMBER(8,0) NOT NULL

4 ,BRANCHITEMID NUMBER(8,0) NOT NULL

5 ,CUSTOMERID NUMBER(8,0) NOT NULL

6 ,CHECKOUTDATE DATE

7 ,DUEDATE DATE

8 ,RETURNDATE DATE

9 ,CREATEDBY VARCHAR2(30) NOT NULL

10 ,CREATEDDATE DATE NOT NULL

11 ,MODIFIEDBY VARCHAR2(30) NOT NULL

12 ,MODIFIEDDATE DATE NOT NULL

13 );

Table TRANSACTION created.

SQL>

SQL> PROMPT Creating Index 'Lib\_Branch\_FK\_I'

Creating Index 'Lib\_Branch\_FK\_I'

SQL> CREATE INDEX Lib\_Branch\_FK\_I ON Branch

2 (LibraryId);

Index LIB\_BRANCH\_FK\_I created.

SQL>

SQL> PROMPT Creating Index 'Sub\_Inv\_FK\_I'

Creating Index 'Sub\_Inv\_FK\_I'

SQL> CREATE INDEX Bran\_BranItem\_FK\_I ON BranchItem

2 (BranchId);

Index BRAN\_BRANITEM\_FK\_I created.

SQL>

SQL> PROMPT Creating Index 'BranItem\_Tran\_FK\_I'

Creating Index 'BranItem\_Tran\_FK\_I'

SQL> CREATE INDEX BranItem\_Tran\_FK\_I ON Transaction

2 (BranchItemId);

Index BRANITEM\_TRAN\_FK\_I created.

SQL> PROMPT Creating Index 'Cust\_Tran\_FK\_I'

Creating Index 'Cust\_Tran\_FK\_I'

SQL> CREATE INDEX Cust\_Tran\_FK\_I ON Transaction

2 (CustomerId);

Index CUST\_TRAN\_FK\_I created.

SQL> PROMPT Creating Index 'Lib\_LibCust\_FK\_I'

Creating Index 'Lib\_LibCust\_FK\_I'

SQL> CREATE INDEX Lib\_LibCust\_FK\_I ON LibraryCustomer

2 (LibraryId);

Index LIB\_LIBCUST\_FK\_I created.

SQL> PROMPT Creating Index 'Cust\_Tran\_FK\_I'

Creating Index 'Cust\_Tran\_FK\_I'

SQL> CREATE INDEX LibCustom\_Cust\_FK\_I ON LibraryCustomer

2 (CustomerId);

Index LIBCUSTOM\_CUST\_FK\_I created.

SQL> PROMPT Creating Index 'BranItem\_Cat\_FK\_I'

Creating Index 'BranItem\_Cat\_FK\_I'

SQL> CREATE INDEX BranItem\_Cat\_FK\_I ON BranchItem

2 (CatalogId);

Index BRANITEM\_CAT\_FK\_I created.

SQL> PROMPT Creating Primary Key on 'Library'

Creating Primary Key on 'Library'

SQL> ALTER TABLE Library

2 ADD CONSTRAINT Lib\_PK PRIMARY KEY

3 (LibraryId);

Table LIBRARY altered.

SQL>

SQL> PROMPT Creating Primary Key on 'LibraryCustomer'

Creating Primary Key on 'LibraryCustomer'

SQL> ALTER TABLE LibraryCustomer

2 ADD CONSTRAINT LibCust\_PK PRIMARY KEY

3 (LibraryCustomerId);

Table LIBRARYCUSTOMER altered.

SQL>

SQL> PROMPT Creating Primary Key on 'Branch'

Creating Primary Key on 'Branch'

SQL> ALTER TABLE Branch

2 ADD CONSTRAINT Branch\_PK PRIMARY KEY

3 (BranchId);

Table BRANCH altered.

SQL>

SQL> PROMPT Creating Primary Key on 'Customer'

Creating Primary Key on 'Customer'

SQL> ALTER TABLE Customer

2 ADD CONSTRAINT Customer\_PK PRIMARY KEY

3 (CustomerId);

Table CUSTOMER altered.

SQL>

SQL> PROMPT Creating Primary Key on 'BranchItem'

Creating Primary Key on 'BranchItem'

SQL> ALTER TABLE BranchItem

2 ADD CONSTRAINT BranItem\_PK PRIMARY KEY

3 (BranchItemId);

Table BRANCHITEM altered.

SQL>

SQL> PROMPT Creating Primary Key on 'Transaction'

Creating Primary Key on 'Transaction'

SQL> ALTER TABLE Transaction

2 ADD CONSTRAINT Tran\_PK PRIMARY KEY

3 (TransactionId);

Table TRANSACTION altered.

SQL>

SQL> PROMPT Creating Primary Key on 'Catalog'

Creating Primary Key on 'Catalog'

SQL> ALTER TABLE Catalog

2 ADD CONSTRAINT Cat\_PK PRIMARY KEY

3 (CatalogId);

Table CATALOG altered.

SQL>

SQL>

SQL> PROMPT Creating Foreign Keys on 'Branch'

Creating Foreign Keys on 'Branch'

SQL> ALTER TABLE Branch ADD CONSTRAINT

2 Lib\_Bran\_FK FOREIGN KEY

3 (LibraryId) REFERENCES Library

4 (LibraryId);

Table BRANCH altered.

SQL>

SQL> PROMPT Creating Foreign Keys on 'LibraryCustomer'

Creating Foreign Keys on 'LibraryCustomer'

SQL> ALTER TABLE LibraryCustomer ADD CONSTRAINT

2 Lib\_LibCust\_FK FOREIGN KEY

3 (LibraryId) REFERENCES Library

4 (LibraryId);

Table LIBRARYCUSTOMER altered.

SQL>

SQL> ALTER TABLE LibraryCustomer ADD CONSTRAINT

2 Cust\_LibCust\_FK FOREIGN KEY

3 (Customerid) REFERENCES Customer

4 (Customerid);

Table LIBRARYCUSTOMER altered.

SQL>

SQL> PROMPT Creating Foreign Keys on 'BranchItem'

Creating Foreign Keys on 'BranchItem'

SQL> ALTER TABLE BranchItem ADD CONSTRAINT

2 Bran\_BranItem\_FK FOREIGN KEY

3 (BranchId) REFERENCES Branch

4 (BranchId);

Table BRANCHITEM altered.

SQL>

SQL> ALTER TABLE BranchItem ADD CONSTRAINT

2 BranItem\_Cat\_FK FOREIGN KEY

3 (CatalogId) REFERENCES Catalog

4 (CatalogId);

Table BRANCHITEM altered.

SQL>

SQL> ALTER TABLE Transaction ADD CONSTRAINT

2 BranItem\_Tran\_FK FOREIGN KEY

3 (BranchItemId) REFERENCES BranchItem

4 (BranchItemId);

Table TRANSACTION altered.

SQL>

SQL> ALTER TABLE Transaction ADD CONSTRAINT

2 Cust\_Tran\_FK FOREIGN KEY

3 (CustomerId) REFERENCES Customer

4 (CustomerId);

Table TRANSACTION altered.

SQL>

SQL> --PROMPT Library seqs

SQL> PROMPT Creating Sequence 'LibraryId\_Seq'

Creating Sequence 'LibraryId\_Seq'

SQL> CREATE SEQUENCE LibraryId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence LIBRARYID\_SEQ created.

SQL> PROMPT Creating Sequence 'LibraryCustomerId\_Seq'

Creating Sequence 'LibraryCustomerId\_Seq'

SQL> CREATE SEQUENCE LibraryCustomerId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence LIBRARYCUSTOMERID\_SEQ created.

SQL> PROMPT Creating Sequence 'BranchId\_Seq'

Creating Sequence 'BranchId\_Seq'

SQL> CREATE SEQUENCE BranchId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence BRANCHID\_SEQ created.

SQL> PROMPT Creating Sequence 'CustomerLId\_Seq'

Creating Sequence 'CustomerLId\_Seq'

SQL> CREATE SEQUENCE CustomerLId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence CUSTOMERLID\_SEQ created.

SQL> PROMPT Creating Sequence 'BranchItemId\_Seq'

Creating Sequence 'BranchItemId\_Seq'

SQL> CREATE SEQUENCE BranchItemId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence BRANCHITEMID\_SEQ created.

SQL> PROMPT Creating Sequence 'TransactionId\_Seq'

Creating Sequence 'TransactionId\_Seq'

SQL> CREATE SEQUENCE TransactionId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence TRANSACTIONID\_SEQ created.

SQL> PROMPT Creating Sequence 'CatalogId\_Seq'

Creating Sequence 'CatalogId\_Seq'

SQL> CREATE SEQUENCE CatalogId\_Seq

2 INCREMENT BY 1

3 START WITH 1

4 NOMAXVALUE

5 MINVALUE 1

6 NOCYCLE

7 NOCACHE

8 /

Sequence CATALOGID\_SEQ created.

SQL>

SQL> CREATE OR REPLACE TRIGGER LibrarySequence\_Trg

2 BEFORE INSERT ON Library

3 FOR EACH ROW

4 BEGIN

5 :new.LibraryId := LibraryId\_Seq.NEXTVAL;

6 END;

7 /

Trigger LIBRARYSEQUENCE\_TRG compiled

SQL>

SQL> CREATE OR REPLACE TRIGGER LibraryCustomerSequence\_Trg

2 BEFORE INSERT ON LibraryCustomer

3 FOR EACH ROW

4 BEGIN

5 :new.LibraryCustomerId := LibraryCustomerId\_Seq.NEXTVAL;

6 END;

7 /

Trigger LIBRARYCUSTOMERSEQUENCE\_TRG compiled

SQL>

SQL> CREATE OR REPLACE TRIGGER BranchSequence\_Trg

2 BEFORE INSERT ON Branch

3 FOR EACH ROW

4 BEGIN

5 :new.BranchId := BranchId\_Seq.NEXTVAL;

6 END;

7 /

Trigger BRANCHSEQUENCE\_TRG compiled

SQL>

SQL> CREATE OR REPLACE TRIGGER CustomerLSequence\_Trg

2 BEFORE INSERT ON Customer

3 FOR EACH ROW

4 BEGIN

5 :new.CustomerId := CustomerLId\_Seq.NEXTVAL;

6 END;

7 /

Trigger CUSTOMERLSEQUENCE\_TRG compiled

SQL> CREATE OR REPLACE TRIGGER BranchItemSequence\_Trg

2 BEFORE INSERT ON BranchItem

3 FOR EACH ROW

4 BEGIN

5 :new.BranchItemId := BranchItemId\_Seq.NEXTVAL;

6 END;

7 /

Trigger BRANCHITEMSEQUENCE\_TRG compiled

SQL> CREATE OR REPLACE TRIGGER TransactionSequence\_Trg

2 BEFORE INSERT ON Transaction

3 FOR EACH ROW

4 BEGIN

5 :new.TransactionId := TransactionId\_Seq.NEXTVAL;

6 END;

7 /

Trigger TRANSACTIONSEQUENCE\_TRG compiled

SQL>

SQL> CREATE OR REPLACE TRIGGER CatalogSequence\_Trg

2 BEFORE INSERT ON Catalog

3 FOR EACH ROW

4 BEGIN

5 :new.CatalogId := CatalogId\_Seq.NEXTVAL;

6 END;

7 /

Trigger CATALOGSEQUENCE\_TRG compiled

SQL> /\*\* Creating Views \*\*/

SQL>

SQL> PROMPT Creating View 'Get\_BranchInfo\_View'

Creating View 'Get\_BranchInfo\_View'

SQL> CREATE VIEW Get\_BranchInfo\_View AS

2 Select b.BranchId, b.LibraryId, b.BranchName,b.BranchPhone,

3 b.Address1, b.Address2, b.City, b.State, b.ZipCode,

4 bi.BranchItemId, bi.CatalogId, bi.CopyNumber, bi.PurchaseDate

5 from Branch b

6 inner join BranchItem bi

7 on b.BranchId = bi.BranchId;

View GET\_BRANCHINFO\_VIEW created.

SQL>

SQL> PROMPT Creating View 'Get\_Report\_View'

Creating View 'Get\_Report\_View'

SQL> CREATE VIEW Get\_Report\_View AS

2 Select t.TransactionId, t.CheckOutDate, t.DueDate, c.CustomerId, c.FirstName, c.LastName,

3 c.Address1, c.City, c.State, c.ZipCode, lc.LibraryCustomerId,

4 lc.LibraryCardNumber, lc.LibraryCardPin,

5 c.CatalogId, bInfo.CopyNumber, bInfo.PurChaseDate, bInfo.BranchName, bInfo.BranchId, lib.LibraryName from Library lib

6 join LibraryCustomer lc on lc.LibraryId = lib.LibraryId

7 join Customer c on lc.CustomerId = c.CustomerId

8 join Get\_BranchInfo\_View bInfo on lib.LibraryId = bInfo.LibraryId

9 join Catalog c on bInfo.CatalogId = c.CatalogId

10 left outer join Transaction t on t.CustomerId = c.CustomerId;

View GET\_REPORT\_VIEW created.