CREATE OR REPLACE VIEW Producers\_Movies AS

SELECT P.COMPANY\_ID,COMPANY\_NAME,MOVIE\_ID,MOVIE\_TITLE

FROM PRODUCER P JOIN MOVIE M

ON P.COMPANY\_ID=M.COMPANY\_ID;

DESCRIBE PRODUCERS\_MOVIES;

SELECT \* FROM PRODUCERS\_MOVIES;

/\*INSTEAD OF TRIGGER\*/

--INSERT

CREATE OR REPLACE TRIGGER Tri\_Insert

INSTEAD OF INSERT

ON Producers\_Movies

FOR EACH ROW

BEGIN

INSERT INTO PRODUCER(COMPANY\_ID,COMPANY\_NAME) VALUES(:NEW.COMPANY\_ID,:NEW.COMPANY\_NAME);

INSERT INTO MOVIE(COMPANY\_ID,MOVIE\_ID,MOVIE\_TITLE,COST,RELEASE\_YEAR,GENRE) VALUES(:NEW.COMPANY\_ID,:NEW.MOVIE\_ID,:NEW.MOVIE\_TITLE,1100,2017,'Comedy');

END PRODUCERS\_MOVIES;

INSERT INTO Producers\_Movies VALUES(2345691,'Vapour Film Company',991140,'Coming 2 America');

SELECT MOVIE\_ID,MOVIE\_TITLE FROM MOVIE;

SELECT \* FROM PRODUCER;

SELECT \* FROM Producers\_Movies;

--UPDATE

CREATE OR REPLACE TRIGGER Tri\_Update

INSTEAD OF UPDATE

ON PRODUCERs\_MOVIES

FOR EACH ROW

BEGIN

UPDATE PRODUCER

SET COMPANY\_NAME=:NEW.COMPANY\_NAME

WHERE COMPANY\_ID=:OLD.COMPANY\_ID;

END;

UPDATE PRODUCERS\_MOVIES SET COMPANY\_NAME='Vapour Company' WHERE COMPANY\_ID=2345691;

SELECT \* FROM PRODUCER;

SELECT \* FROM Producers\_Movies;

--DELETE

CREATE OR REPLACE TRIGGER Tri\_Delete

INSTEAD OF DELETE

ON PRODUCERS\_MOVIES

FOR EACH ROW

BEGIN

DELETE

FROM MOVIE WHERE MOVIE\_TITLE=:OLD.MOVIE\_TITLE;

END;

/

DELETE FROM Producers\_Movies WHERE MOVIE\_TITLE='Coming 2 America';

SELECT MOVIE\_ID,MOVIE\_TITLE FROM MOVIE;

SELECT \* FROM Producers\_Movies;

DROP TRIGGER TRI\_DELETE;

ALTER TRIGGER TRI\_INSERT DISABLE;

ALTER TRIGGER TRI\_INSERT ENABLE;

ALTER TABLE MOVIE DISABLE ALL TRIGGERS;

ALTER TABLE PRODUCER ENABLE ALL TRIGGERS;

DROP TRIGGER TRI\_INSERT;

DROP TRIGGER TRI\_UPDATE;

/\*BEFORE TRIGGER\*/

--INSERT & UPDATE

CREATE OR REPLACE TRIGGER Tri\_Insert

BEFORE INSERT OR UPDATE ON Movie

FOR EACH ROW

DECLARE

Invalid\_cost EXCEPTION;

Invalid\_rating EXCEPTION;

PRAGMA EXCEPTION\_INIT (INVALID\_COST,-00001);

PRAGMA EXCEPTION\_INIT (INVALID\_RATING,-00002);

BEGIN

IF (:NEW.COST >2000 OR :NEW.COST<1000) THEN

RAISE INVALID\_COST;

ELSIF (:NEW.RATING NOT IN('One','Two','Three','Four','Five')) THEN

RAISE INVALID\_RATING;

END IF;

EXCEPTION

WHEN INVALID\_COST THEN

RAISE\_APPLICATION\_ERROR(-20001,'COST IS TOO HIGH');

WHEN INVALID\_RATING THEN

RAISE\_APPLICATION\_ERROR(

num=>-20002,

msg=>'RATING MUST BE FROM [ONE,TWO,THREE,FOUR,FIVE]');

END PRODUCERS\_MOVIES;

/

INSERT INTO MOVIE(MOVIE\_ID,COST,RATING,RELEASE\_YEAR,GENRE,MOVIE\_TITLE) VALUES(991140,1500,'Four',2000,'Thriller','Coming 2 America');

SELECT \* FROM MOVIE;

UPDATE MOVIE SET MOVIE\_TITLE='Coming America',COST=3000 WHERE MOVIE\_ID=991140;

SELECT \* FROM MOVIE;

DROP TRIGGER TRI\_INSERT;

--DELETE

CREATE OR REPLACE TRIGGER Tri\_delete

BEFORE DELETE ON MOVIE

REFERENCING OLD AS o

NEW AS n

FOR EACH ROW

BEGIN

IF :O.MOVIE\_ID=991134 THEN

RAISE\_APPLICATION\_ERROR(-20001,'CANNOT DELETE');

END IF;

END;

/

DELETE FROM MOVIE WHERE MOVIE\_ID=991134;

select \* from movie;

DROP TRIGGER TRI\_DELETE;

/\*AFTER TRIGGER\*/

CREATE OR REPLACE TRIGGER After\_Demo

AFTER INSERT OR UPDATE OR DELETE

ON MOVIE

FOR EACH ROW

BEGIN

IF INSERTING THEN

DBMS\_OUTPUT.PUT\_LINE('INSERTED SUCCESSFULLY');

END IF;

IF UPDATING THEN

DBMS\_OUTPUT.PUT\_LINE('UPDATED SUCCESSFULLY');

END IF;

IF DELETING THEN

DBMS\_OUTPUT.PUT\_LINE('DELETED SUCCESSFULLY');

END IF;

END;

/

INSERT INTO MOVIE(MOVIE\_ID,COST,RATING,RELEASE\_YEAR,GENRE,MOVIE\_TITLE) VALUES(991141,1500,'Four',2000,'Thriller','Coming 3 America');

SELECT \* FROM MOVIE;

UPDATE MOVIE SET MOVIE\_TITLE='Coming America',COST=3000 WHERE MOVIE\_ID=991140;

SELECT \* FROM MOVIE;

DELETE FROM MOVIE WHERE MOVIE\_ID=991140;

select \* from movie;