create table Producer(

company\_id number primary key, /\* PK as column constraint\*/

company\_name varchar(30) not null,

country varchar(20) default('India')

);

desc Producer;

CREATE TABLE Director(

director\_id INt primary KEY,

dfname varchar(20) not NULL,

dlname varchar2(20),

birth\_year int,

death\_year int,

CONSTRAINT chk\_year CHECK(death\_year>=birth\_year) --check as table constraint

);

DESCRIBE Director;

creatE taBle Customer(

member\_id number,

cfname varchar(20) NOT NULL,

clname varchar(20),

gender char(10) CHECK(gender in ('Male','Female','Other')), --- check as column constraint

DOB date NOT NULL,

email varchar(50) NOT NULL,

active INT NOT NULL,

city varchar(30) NOT NULL,

area varchar(50),

zipcode number,

country varchar(30) NOT NULL,

no\_of\_rentals int NOT NULL,

Primary key(member\_id) --PK as table constraint---

);

describe Customer;

create table Staff(

staff\_id number,

sfname varchar(20) NOT NULL,

slname varchar(20),

email varchar(50),

DOB date,

gender char(30) CHECK(gender in ('Male','Female')),

city varchar(30),

area varchar(50),

zipcode number(10),

country varchar(30) NOT NULL,

constraint pk\_staffid primary KEY(staff\_id) ---PK as table constraint-

);

DESC Staff;

create table Customer\_phone(

phone number NOT NULL,

member\_id number,

primary key(phone,member\_id),

foreign key(member\_id) references Customer(member\_id) /\*FK as table constraint\*/

);

desC Customer\_phone;

create table Staff\_phone(

phone int NOT NULL,

staff\_id number references Staff(staff\_id), /\*FK as column constraint\*/

primary key(phone,staff\_id)

);

Describe Staff\_phone;

CREATE TABLE Rental(

rent\_id INT PRIMARY KEY,

rent\_date DATE NOT NULL,

rent\_duedate DATE NOT NULL,

staff\_id NUMBER,

FOREIGN KEY(staff\_id) REFERENCES Staff(staff\_id),

CHECK(rent\_duedate>=rent\_date)

);

desc Rental;

CREATE TABLE rent(

member\_id number,

rent\_id INT,

PRIMARY KEY (member\_id, rent\_id),

FOREIGN KEY (member\_id) REFERENCES Customer(member\_id),

FOREIGN KEY (rent\_id) REFERENCES Rental(rent\_id)

);

desc rent;

create table Payment(

transaction\_id int primary key,

type varchar(20),

amount NUMbEr(7,2) CHECK(amount>0),

datee date,

rent\_id int,

member\_id number,

staff\_id number,

foreign key(rent\_id) references Rental(rent\_id),

foreign key(member\_id) references Customer(member\_id),

foreign key(staff\_id) references Staff(staff\_id)

);

desc Payment;

CREATE TABLE Movie(

movie\_id INT PRIMARY KEY,

length INT,

cost NUMBER(7,2) NOT NULL CHECK(cost>0),

duration Int,

rating VARCHAR(20),

release\_year INT NOT NULL,

description VARCHAR(200),

movie\_title VARCHAR(100) UNIQUE, /\*UNIQUE as column constraint\*/

replacement\_cost INT DEFAULT(200),

genre VARCHAR(50) NOT NULL,

company\_id NUMBER,

director\_id INT,

rent\_id INT,

FOREIGN KEY (company\_id) REFERENCES Producer(company\_id),

FOREIGN KEY (director\_id) REFERENCES Director(director\_id),

FOREIGN KEY (rent\_id) REFERENCES Rental(rent\_id)

);

desc Movie;

CREATE TABLE Movie\_language

(

language VARCHAR(20) NOT NULL,

movie\_id INT,

PRIMARY KEY (language, movie\_id),

-- unique(language,movie\_id), /\*UNIQUE as table constraint\*/

FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id)

);

describe Movie\_language;

create TABLE Actor(

actor\_id int PRIMARY KEY,

afname varchar(20) not null,

alname varchar(20)

);

describe Actor;

CREATE TABLE Acted\_in

(

role VARCHAR(20),

actor\_id INT,

movie\_id INT,

PRIMARY KEY (actor\_id, movie\_id),

FOREIGN KEY (actor\_id) REFERENCES Actor(actor\_id),

FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id)

);

desc Acted\_in;

--/\*ALTER COMMANDS\*/

--1)ADDING a NEW COLUMN

ALTER TABLE Movie

ADD new\_released\_movie varchar2(50) not null;

--2)ADDING multiple NEW COLUMN

ALTER TABLE Movie

ADD (new\_released\_year number(4),

new\_duration int,

new\_cost int);

desc Movie;

--3)MODIFY a COLUMN & DROPPING/CHANGING DEFAULT

ALTER TABLE Movie

MODIFY new\_released\_movie DEFAULT'NULL';

--4)MODIFY multiple COLUMN & ADDING CONSTRAINTS (not null & default) & DROPPING NOT NULL

ALTER TABLE Movie

MODIFY (new\_released\_movie char(50) NULL ,

new\_duration NUMBER NOT NULL UNIQUE);

desc Movie;

--5)DROP a COLUMN

ALTER TABLE Movie

DROP COLUMN new\_released\_movie;

--6)DROP multiple COLUMN

ALTER TABLE Movie

DROP (new\_released\_year,

new\_duration);

describe Movie;

--7)RENAME a COLUMN

ALTER TABLE Movie

RENAME COLUMN genre To genre\_name;

--8)RENAME a TABLE

ALTER TABLE Movie

RENAME tO Moviee;

desc Moviee;

/\*Alter --Adding Constraints\*/

--9)ADDING UNIQUE CONSTRAINT

ALTER TABLE Moviee

/\* ADD UNIQUE(new\_cost); \*/

ADD CONSTRAINT Unique\_Cost UNIQUE(new\_cost,cost);

--10)ADDING PRIMARY KEY

/\*FIRST REMOVE AVAILABLE PK \*/

--ALTER TABLE Moviee

--ADD CONSTRAINT PK\_Title PRIMARY KEY(movie\_title,movie\_id);

--11)ADDING FOREIGN KEY

/\*FIRST REMOVE AVAILABLE FK \*/

---ALTER TABLE moviee

---ADD CONSTRAINT FK\_Companyid FOREIGN KEY(company\_id) REFERENCES Producer(company\_id);

--12)ADDING CHECK CONSTRAINT

ALTER TABLE MOVIEE

/\*ADD CONSTRAINT Chk\_cost CHECK(new\_cost>0);\*/

ADD CONSTRAINT Chk\_cost CHECK(new\_cost>0 AND rating='five');

/\*Alter --Dropping Constraint \*/

--13)DROPPING UNIQUE CONSTRAINT

ALTER TABLE MOVIEE

DROP CONSTRAINT Unique\_Cost;

--14)DROPPING PK CONSTRAINT

---ALTER TABLE MOVIEE

--DROP CONSTRAINT PK\_Title;

--15)DROPPING FK CONSTRAINT

--ALTER TABLE MOVIEE

--DROP CONSTRAINT FK\_Companyid;

--16)DROPPING CHECK CONSTRAINT

ALTER TABLE MOVIEE

DROP CONSTRAINT Chk\_cost;

--Restoring the original movie table

ALTER TABLE MOVIEE

DROP COLUMN NEW\_COST;

ALTER TABLE MOVIEE

RENAME TO Movie;

ALTER TABLE MOVIE

RENAME COLUMN GENRE\_NAME TO GENRE;

desc movie;

CREATE TABLE manager

(

manager\_id NUMBER(7) PRIMARY KEY,

name CHAR(50),

gender VARCHAR2(6) DEFAULT 'NULL'

);

DESC MANAGER;

INSERT INTO MANAGER(MANAGER\_ID,NAME) VALUES(1234,'Shubham');

INSERT INTO MANAGER VALUES(12345,'Utsav','Male');

SELECT \* FROM MANAGER;

TRUNCATE TABLE MANAGER;

SELECT \* FROM MANAGER;

DROP TABLE MANAGER;

DESCRIBE MANAGER;