SQL (1)

Create Table

```
CREATE TABLE Studios(
    name VARCHAR(20),
    website VARCHAR(255)
);
```

```
CREATE TABLE Stars (
    name VARCHAR(20),
    gender CHAR(1),
    birthyear INT,
    birthplace VARCHAR(40)
);
```

```
CREATE TABLE Movies (
    title VARCHAR(50),
    year INT,
    length INT,
    rating CHAR(2),
    studioname VARCHAR(20)
);
```

```
CREATE TABLE StarsIn (
    title VARCHAR(50),
    year INT,
    starname VARCHAR(20)
);
```

CHAR and VARCHAR

- **CHAR(n)** allocates a fixed space, and if the string that we store is shorter than **n**, then it is padded with blanks.

- Differently, **VARCHAR(n)** denotes a string of up to n characters.
  - VARCHAR(n) allows for compression to save space.

- Use CHAR(n) for frequently used fields, and use VARCHAR(n) otherwise.

Insert – Studios

```
INSERT INTO Studios
VALUES('Fox', 'foxmovies.com');

INSERT INTO Studios
VALUES('Disney', 'disney.com');

INSERT INTO Studios
VALUES('Paramount', 'www.paramount.com');
```

Insert – Movies

```
INSERT INTO Movies
VALUES('Walk the Line', 2005, 136, 'PG', 'Fox');

INSERT INTO Movies
VALUES('Pretty Woman', 1990, 119, 'R', 'Disney');

INSERT INTO Movies
VALUES('Wayne''s World', 1991, 104, 'PG', 'Paramount');

INSERT INTO Movies
VALUES('Unfaithful', 2002, 124, 'R', 'Fox');

INSERT INTO Movies
VALUES('Runaway Bride', 1999, 116, 'PG', 'Paramount');

INSERT INTO Movies
VALUES('The Princess and the Frog', 2009, 97, 'G', 'Disney');
```

Insert – Stars

```
INSERT INTO Stars
VALUES('Richard Gere', 'M', 1949, 'Philadelphia, Pennsylvania, USA');

INSERT INTO Stars
VALUES('Joaquin Phoenix', 'M', 1974, 'San Juan, Puerto Rico');

INSERT INTO Stars
VALUES('Reese Witherspoon', 'F', 1976, 'Baton Rouge, Louisiana, USA');

INSERT INTO Stars
VALUES('Julia Roberts', 'F', 1967, 'Smyrna, Georgia, USA');

INSERT INTO Stars
VALUES('Mike Myers', 'M', 1963, 'Scarborough, Ontario, Canada');

INSERT INTO Stars
VALUES('Oprah Winfrey', 'F', 1954, 'Kosciusko, Mississippi, USA');
```

Insert – StarsIn

```
INSERT INTO StarsIn VALUES('Walk the Line', 2005, 'Joaquin Phoenix');
INSERT INTO StarsIn VALUES('Walk the Line', 2005, 'Reese Witherspoon');

INSERT INTO StarsIn VALUES('Pretty Woman', 1990, 'Richard Gere');
INSERT INTO StarsIn VALUES('Pretty Woman', 1990, 'Julia Roberts');

INSERT INTO StarsIn VALUES('Wayne''s World', 1991, 'Mike Myers');

INSERT INTO StarsIn VALUES('Unfaithful', 2002, 'Richard Gere');

INSERT INTO StarsIn VALUES('Runaway Bride', 1999, 'Richard Gere');
INSERT INTO StarsIn VALUES('Runaway Bride', 1999, 'Julia Roberts');

INSERT INTO StarsIn VALUES('The Princess and the Frog', 2009, 'Oprah Winfrey');
```

Create Table with Primary Keys

```
CREATE TABLE Studios(
    name VARCHAR(20) PRIMARY KEY,
    website VARCHAR(255)
);
```

```
CREATE TABLE Stars (
    name VARCHAR(20) PRIMARY KEY,
    gender CHAR(1),
    birthyear INT,
    birthplace VARCHAR(40)
);
```

```
CREATE TABLE Movies (
    title VARCHAR(50),
    year INT,
    length INT,
    rating CHAR(2),
    studioname VARCHAR(20),
    PRIMARY KEY (title, year)
);
```

```
CREATE TABLE StarsIn (
    title VARCHAR(50),
    year INT,
    starname VARCHAR(20),
    PRIMARY KEY (title,year,starname)
);
```

Create Table with Foreign Keys

```
CREATE TABLE Studios(
    name VARCHAR(20) PRIMARY KEY,
    website VARCHAR(255)
);
```

```
CREATE TABLE Stars (
    name VARCHAR(20) PRIMARY KEY,
    gender CHAR(1),
    birthyear INT,
    birthplace VARCHAR(40)
);
```

```
CREATE TABLE Movies (
    title VARCHAR(50),
    year INT,
    length INT,
    rating CHAR(2),
    studioname VARCHAR(20),
    PRIMARY KEY (title, year),
    FOREIGN KEY (studioName) REFERENCES Studios(name) ON DELETE CASCADE
);
```

```
CREATE TABLE StarsIn (
     title VARCHAR(50),
     year INT,
     starname VARCHAR(20),
     PRIMARY KEY (title,year,starname),
     FOREIGN KEY (title,year) REFERENCES Movies(title,year) ON DELETE CASCADE,
     FOREIGN KEY (starName) REFERENCES Stars(name) ON DELETE CASCADE
);
```

Short Form for Single Att. Foreign Keys

```
CREATE TABLE Movies (
    title VARCHAR(50),
    year INT,
    length INT,
    rating CHAR(2),
    studioname VARCHAR(20) REFERENCES Studios(name) ON DELETE CASCADE,
    PRIMARY KEY (title, year)
);
```

No need to say Foreign Key

Creation and insertion order

1. **Movies** after **Studios**

2. **StarsIn** after **Movies** and **Stars**

Dropping Tables

```
DROP TABLE StarsIn;
DROP TABLE Movies;
DROP TABLE Stars;
DROP TABLE Studios;
```

Note. Order of drops is important if foreign key constraints are in place. Why?

Getting all the tuples of a table

E.g.

```
SELECT *
FROM Movies;
```

Altering Table Structure

```
ALTER TABLE Stars ADD phone CHAR(7);
```

```
ALTER TABLE Stars MODIFY phone CHAR(10);
```

```
ALTER TABLE Stars DROP COLUMN phone;
```