Load the database in your sql by the following command and then start practicing the following codes…

1. make a new database say abc

2. Mysql -u [username] -p abc < dsc.sql (this file is given by us)

All tables in database dsc.sql will be transferred to your created new database.

USE DSCworkshop;

#selecting DSCworkshop to use it for different tables

SHOW TABLES;

#it will show the all tables which belong to DSCworkshop database

+-----------------------+

| Tables\_in\_dscworkshop |

+-----------------------+

| actors |

| movies |

| movies\_genres |

+-----------------------+

DESCRIBE movies;

#it will give structure of the table named ‘MOVIES’

+-----------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+--------------+------+-----+---------+-------+

| id | int(11) | NO | | 0 | |

| name | varchar(100) | YES | | NULL | |

| year | int(11) | YES | | NULL | |

| rankscore | float | YES | | NULL | |

+-----------+--------------+------+-----+---------+-------+\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \* FROM movies;

# will show all rows of table MOVIES

+-----+-----------------------------------------------------------------------------+------+-----------+

| id | name | year | rankscore |

+-----+-----------------------------------------------------------------------------+------+-----------+

| 0 | #28 | 2002 | NULL |

| 1 | #7 Train: An Immigrant Journey, The | 2000 | NULL |

| 2 | $ | 1971 | 6.4 |

| 3 | $1,000 Reward | 1913 | NULL |

| 4 | $1,000 Reward | 1915 | NULL |

| 5 | $1,000 Reward | 1923 | NULL |

| 6 | $1,000,000 Duck | 1971 | 5 |

| 7 | $1,000,000 Reward, The | 1920 | NULL |

| 8 | $10,000 Under a Pillow | 1921 | NULL |

| 9 | $100,000 | 1915 | NULL |

| 10 | $100,000 Pyramid, The | 2001 | NULL |

| 11 | $1000 a Touchdown | 1939 | 6.7 |

| 12 | $20,000 Carat, The | 1913 | NULL |

| 13 | $21 a Day Once a Month | 1941 | NULL |

| 14 | $2500 Bride, The | 1912 | NULL |

| 15 | $30 | 1999 | 7.5 |

| 16 | $30,000 | 1920 | NULL |

| 17 | $300 y tickets | 2002 | NULL |

| 18 | $40,000 | 1996 | 9.6 |

| 19 | $5,000 Reward | 1918 | NULL |

| 20 | $5,000,000 Counterfeiting Plot, The | 1914 | NULL |

| 21 | $5.15/Hr. | 2004 | NULL |

| 22 | $5.20 an Hour Dream, The | 1980 | NULL |

| 23 | $50,000 Challenge, The | 1989 | NULL |

| 24 | $50,000 Climax Show, The | 1975 | 2.6 |

| 25 | $50,000 Jewel Theft, The | 1915 | NULL |

| 26 | $50,000 Reward | 1924 | NULL |

| 28 | $500 Reward, The | 1911 | NULL |

| 29 | $500,000 Reward | 1911 | NULL |

| 31 | $pent | 2000 | 4.3 |

| 32 | $ucces Part One | 1986 | NULL |

| 33 | $windle | 2002 | 5.4 |

| 34 | & frres | 2000 | NULL |

| 35 | ''Bear'' Facts, The | 1914 | NULL |

| 36 | '15' | 2002 | 6.8 |

| 37 | '24-25' ne vozvrashchayetsya | 1968 | NULL |

| 38 | '38 | 1987 | 6.7 |

| 39 | '42 | 1951 | NULL |

| 40 | '49, un souffle de colre | 1996 | NULL |

| 41 | '49-'17 | 1917 | 5.8 |

| 42 | '60s Pop Rock Reunion | 2004 | NULL |

| 43 | '60s, The | 1999 | NULL |

| 44 | '68 | 1988 | 5.5 |

| 45 | '70s Celebration: The Beat Is Back, A | 1993 | NULL |

| 46 | '70s: The Decade That Changed Television, The | 2000 | NULL |

| 47 | '88 Dodge Aries | 2002 | NULL |

| 48 | '93 jie tou ba wang | 1993 | NULL |

| 49 | '94 du bi dao zhi qing | 1994 | 5.7 |

| 50 | '?' Motorist, The | 1906 | 6.8 |

| 51 | 'A' | 1965 | 7.1 |

| 52 | 'A' gai waak | 1983 | 7.2 |

| 53 | 'A' gai waak juk jaap | 1987 | 7.2 |

| 54 | 'Abbot' and 'Cresceus' Race, The | 1901 | NULL |

| 55 | 'Akasaka no shimai' yori: yoru no wada | 1960 | NULL |

| 56 | 'Amphitrite', The | 1897 | NULL |

| 57 | 'Ang Galing galing mo, Babes' | 2002 | NULL |

| 58 | 'Arriet's Baby | 1913 | NULL |

| 59 | 'Arry and 'Arriet's Evening Out | 1909 | NULL |

| 60 | 'Arry on the Steamboat | 1901 | NULL |

| 61 | 'As se hizo' - Torremolinos 73 | 2003 | NULL |

| 62 | 'Babicky dobjejte presne!' | 1983 | 5.6 |

| 63 | 'Bad Buck' of Santa Ynez | 1915 | NULL |

| 64 | 'Bats' Abound | 1999 | NULL |

| 65 | 'Belle de Varsovie, La' | 1993 | NULL |

| 66 | 'Betty Bee' (sopravvivere d'arte) | 1999 | NULL |

| 67 | 'Billy Elliot' Boy, The | 2001 | NULL |

| 68 | 'Borovo' Institut | 1984 | NULL |

| 69 | 'Breaker' Morant | 1980 | 7.9 |

| 70 | 'Brennus', Le | 1897 | NULL |

| 71 | 'Broadway Melody of 1940' | 1940 | 7.1 |

| 72 | 'Bullitt': Steve McQueen's Commitment to Reality | 1968 | 6.6 |

| 73 | 'burbs, The | 1989 | 5.9 |

| 74 | 'C' Word, The | 1993 | NULL |

| 76 | 'Cat and Mouse' at the Home | 1983 | NULL |

| 77 | 'Catch Me If You Can': Behind the Camera | 2003 | NULL |

| 78 | 'Columbia' | 1899 | NULL |

| 79 | 'Columbia' and 'Defender' Rounding Stake-Boat | 1899 | NULL |

| 80 | 'Columbia' and 'Shamrock II' Finishing Second Race | 1901 | NULL |

| 81 | 'Columbia' and 'Shamrock II': Jockeying and Starting | 1901 | NULL |

| 82 | 'Columbia' and 'Shamrock II': Start of Second Race | 1901 | NULL |

| 83 | 'Columbia' and 'Shamrock II': Starting in the Third Race | 1901 | NULL |

| 84 | 'Columbia' and 'Shamrock II': Turning the Outer Stake Boat | 1901 | NULL |

| 85 | 'Columbia' Close to the Wind | 1899 | NULL |

| 86 | 'Columbia' vs. 'Defender' | 1899 | NULL |

| 87 | 'Columbia' Winning the Cup | 1899 | 3.6 |

| 88 | 'Columbia' Winning the Cup (1901/I) | 1901 | NULL |

| 89 | 'Columbia' Winning the Cup (1901/II) | 1901 | NULL |

| 90 | 'Corsair', The | 1899 | NULL |

| 91 | 'Courageous', O | 1932 | NULL |

| 93 | 'Courier' | 1900 | NULL |

| 94 | 'Crocodile' Dundee II | 1988 | 4.9 |

| 95 | 'Cross the Mexican Line | 1914 | NULL |

| 96 | 'Den vide verden's billedbog nr. 1 | 1938 | NULL |

| 97 | 'Deutschland' Leaving New York at Full Speed (with Prince Henry of Prussia) | 1902 | NULL |

| 98 | 'Diamond S' Ranch, The | 1912 | NULL |

| 99 | 'Dikke Bertha' op doortocht in Brussel | 1919 | NULL |

| 100 | 'E' | 1981 | 7.9 |

| 101 | 'El Chicko' - der Verdacht | 1995 | 4.1 |

| 102 | 'Elf' Jukebox | 2004 | NULL |

| 103 | 'Erin', The | 1899 | NULL |

+-----+-----------------------------------------------------------------------------+------+-----------+

#result-set: a set of rows that form the result of a query along with column-names and meta-data.

SELECT name,year FROM movies;

+-----------------------------------------------------------------------------+------+

| name | year |

+-----------------------------------------------------------------------------+------+

| #28 | 2002 |

| #7 Train: An Immigrant Journey, The | 2000 |

| $ | 1971 |

| $1,000 Reward | 1913 |

| $1,000 Reward | 1915 |

| $1,000 Reward | 1923 |

| $1,000,000 Duck | 1971 |

| $1,000,000 Reward, The | 1920 |

| $10,000 Under a Pillow | 1921 |

| $100,000 | 1915 |

| $100,000 Pyramid, The | 2001 |

| $1000 a Touchdown | 1939 |

| $20,000 Carat, The | 1913 |

| $21 a Day Once a Month | 1941 |

| $2500 Bride, The | 1912 |

| $30 | 1999 |

| $30,000 | 1920 |

| $300 y tickets | 2002 |

| $40,000 | 1996 |

| $5,000 Reward | 1918 |

| $5,000,000 Counterfeiting Plot, The | 1914 |

| $5.15/Hr. | 2004 |

| $5.20 an Hour Dream, The | 1980 |

| $50,000 Challenge, The | 1989 |

| $50,000 Climax Show, The | 1975 |

| $50,000 Jewel Theft, The | 1915 |

| $50,000 Reward | 1924 |

| $500 Reward, The | 1911 |

| $500,000 Reward | 1911 |

| $pent | 2000 |

| $ucces Part One | 1986 |

| $windle | 2002 |

| & frres | 2000 |

| ''Bear'' Facts, The | 1914 |

| '15' | 2002 |

| '24-25' ne vozvrashchayetsya | 1968 |

| '38 | 1987 |

| '42 | 1951 |

| '49, un souffle de colre | 1996 |

| '49-'17 | 1917 |

| '60s Pop Rock Reunion | 2004 |

| '60s, The | 1999 |

| '68 | 1988 |

| '70s Celebration: The Beat Is Back, A | 1993 |

| '70s: The Decade That Changed Television, The | 2000 |

| '88 Dodge Aries | 2002 |

| '93 jie tou ba wang | 1993 |

| '94 du bi dao zhi qing | 1994 |

| '?' Motorist, The | 1906 |

| 'A' | 1965 |

| 'A' gai waak | 1983 |

| 'A' gai waak juk jaap | 1987 |

| 'Abbot' and 'Cresceus' Race, The | 1901 |

| 'Akasaka no shimai' yori: yoru no wada | 1960 |

| 'Amphitrite', The | 1897 |

| 'Ang Galing galing mo, Babes' | 2002 |

| 'Arriet's Baby | 1913 |

| 'Arry and 'Arriet's Evening Out | 1909 |

| 'Arry on the Steamboat | 1901 |

| 'As se hizo' - Torremolinos 73 | 2003 |

| 'Babicky dobjejte presne!' | 1983 |

| 'Bad Buck' of Santa Ynez | 1915 |

| 'Bats' Abound | 1999 |

| 'Belle de Varsovie, La' | 1993 |

| 'Betty Bee' (sopravvivere d'arte) | 1999 |

| 'Billy Elliot' Boy, The | 2001 |

| 'Borovo' Institut | 1984 |

| 'Breaker' Morant | 1980 |

| 'Brennus', Le | 1897 |

| 'Broadway Melody of 1940' | 1940 |

| 'Bullitt': Steve McQueen's Commitment to Reality | 1968 |

| 'burbs, The | 1989 |

| 'C' Word, The | 1993 |

| 'Cat and Mouse' at the Home | 1983 |

| 'Catch Me If You Can': Behind the Camera | 2003 |

| 'Columbia' | 1899 |

| 'Columbia' and 'Defender' Rounding Stake-Boat | 1899 |

| 'Columbia' and 'Shamrock II' Finishing Second Race | 1901 |

| 'Columbia' and 'Shamrock II': Jockeying and Starting | 1901 |

| 'Columbia' and 'Shamrock II': Start of Second Race | 1901 |

| 'Columbia' and 'Shamrock II': Starting in the Third Race | 1901 |

| 'Columbia' and 'Shamrock II': Turning the Outer Stake Boat | 1901 |

| 'Columbia' Close to the Wind | 1899 |

| 'Columbia' vs. 'Defender' | 1899 |

| 'Columbia' Winning the Cup | 1899 |

| 'Columbia' Winning the Cup (1901/I) | 1901 |

| 'Columbia' Winning the Cup (1901/II) | 1901 |

| 'Corsair', The | 1899 |

| 'Courageous', O | 1932 |

| 'Courier' | 1900 |

| 'Crocodile' Dundee II | 1988 |

| 'Cross the Mexican Line | 1914 |

| 'Den vide verden's billedbog nr. 1 | 1938 |

| 'Deutschland' Leaving New York at Full Speed (with Prince Henry of Prussia) | 1902 |

| 'Diamond S' Ranch, The | 1912 |

| 'Dikke Bertha' op doortocht in Brussel | 1919 |

| 'E' | 1981 |

| 'El Chicko' - der Verdacht | 1995 |

| 'Elf' Jukebox | 2004 |

| 'Erin', The | 1899 |

+-----------------------------------------------------------------------------+------+

# will show name and year as column and show all rows from table ‘MOVIES’

SELECT rankscore,name FROM movies;

#row order same as the one in the table

+-----------+-----------------------------------------------------------------------------+

| rankscore | name |

+-----------+-----------------------------------------------------------------------------+

| NULL | #28 |

| NULL | #7 Train: An Immigrant Journey, The |

| 6.4 | $ |

| NULL | $1,000 Reward |

| NULL | $1,000 Reward |

| NULL | $1,000 Reward |

| 5 | $1,000,000 Duck |

| NULL | $1,000,000 Reward, The |

| NULL | $10,000 Under a Pillow |

| NULL | $100,000 |

| NULL | $100,000 Pyramid, The |

| 6.7 | $1000 a Touchdown |

| NULL | $20,000 Carat, The |

| NULL | $21 a Day Once a Month |

| NULL | $2500 Bride, The |

| 7.5 | $30 |

| NULL | $30,000 |

| NULL | $300 y tickets |

| 9.6 | $40,000 |

| NULL | $5,000 Reward |

| NULL | $5,000,000 Counterfeiting Plot, The |

| NULL | $5.15/Hr. |

| NULL | $5.20 an Hour Dream, The |

| NULL | $50,000 Challenge, The |

| 2.6 | $50,000 Climax Show, The |

| NULL | $50,000 Jewel Theft, The |

| NULL | $50,000 Reward |

| NULL | $500 Reward, The |

| NULL | $500,000 Reward |

| 4.3 | $pent |

| NULL | $ucces Part One |

| 5.4 | $windle |

| NULL | & frres |

| NULL | ''Bear'' Facts, The |

| 6.8 | '15' |

| NULL | '24-25' ne vozvrashchayetsya |

| 6.7 | '38 |

| NULL | '42 |

| NULL | '49, un souffle de colre |

| 5.8 | '49-'17 |

| NULL | '60s Pop Rock Reunion |

| NULL | '60s, The |

| 5.5 | '68 |

| NULL | '70s Celebration: The Beat Is Back, A |

| NULL | '70s: The Decade That Changed Television, The |

| NULL | '88 Dodge Aries |

| NULL | '93 jie tou ba wang |

| 5.7 | '94 du bi dao zhi qing |

| 6.8 | '?' Motorist, The |

| 7.1 | 'A' |

| 7.2 | 'A' gai waak |

| 7.2 | 'A' gai waak juk jaap |

| NULL | 'Abbot' and 'Cresceus' Race, The |

| NULL | 'Akasaka no shimai' yori: yoru no wada |

| NULL | 'Amphitrite', The |

| NULL | 'Ang Galing galing mo, Babes' |

| NULL | 'Arriet's Baby |

| NULL | 'Arry and 'Arriet's Evening Out |

| NULL | 'Arry on the Steamboat |

| NULL | 'As se hizo' - Torremolinos 73 |

| 5.6 | 'Babicky dobjejte presne!' |

| NULL | 'Bad Buck' of Santa Ynez |

| NULL | 'Bats' Abound |

| NULL | 'Belle de Varsovie, La' |

| NULL | 'Betty Bee' (sopravvivere d'arte) |

| NULL | 'Billy Elliot' Boy, The |

| NULL | 'Borovo' Institut |

| 7.9 | 'Breaker' Morant |

| NULL | 'Brennus', Le |

| 7.1 | 'Broadway Melody of 1940' |

| 6.6 | 'Bullitt': Steve McQueen's Commitment to Reality |

| 5.9 | 'burbs, The |

| NULL | 'C' Word, The |

| NULL | 'Cat and Mouse' at the Home |

| NULL | 'Catch Me If You Can': Behind the Camera |

| NULL | 'Columbia' |

| NULL | 'Columbia' and 'Defender' Rounding Stake-Boat |

| NULL | 'Columbia' and 'Shamrock II' Finishing Second Race |

| NULL | 'Columbia' and 'Shamrock II': Jockeying and Starting |

| NULL | 'Columbia' and 'Shamrock II': Start of Second Race |

| NULL | 'Columbia' and 'Shamrock II': Starting in the Third Race |

| NULL | 'Columbia' and 'Shamrock II': Turning the Outer Stake Boat |

| NULL | 'Columbia' Close to the Wind |

| NULL | 'Columbia' vs. 'Defender' |

| 3.6 | 'Columbia' Winning the Cup |

| NULL | 'Columbia' Winning the Cup (1901/I) |

| NULL | 'Columbia' Winning the Cup (1901/II) |

| NULL | 'Corsair', The |

| NULL | 'Courageous', O |

| NULL | 'Courier' |

| 4.9 | 'Crocodile' Dundee II |

| NULL | 'Cross the Mexican Line |

| NULL | 'Den vide verden's billedbog nr. 1 |

| NULL | 'Deutschland' Leaving New York at Full Speed (with Prince Henry of Prussia) |

| NULL | 'Diamond S' Ranch, The |

| NULL | 'Dikke Bertha' op doortocht in Brussel |

| 7.9 | 'E' |

| 4.1 | 'El Chicko' - der Verdacht |

| NULL | 'Elf' Jukebox |

| NULL | 'Erin', The |

+-----------+-----------------------------------------------------------------------------+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LIMIT:

SELECT name,rankscore FROM movies LIMIT 20;

+-------------------------------------+-----------+

| name | rankscore |

+-------------------------------------+-----------+

| #28 | NULL |

| #7 Train: An Immigrant Journey, The | NULL |

| $ | 6.4 |

| $1,000 Reward | NULL |

| $1,000 Reward | NULL |

| $1,000 Reward | NULL |

| $1,000,000 Duck | 5 |

| $1,000,000 Reward, The | NULL |

| $10,000 Under a Pillow | NULL |

| $100,000 | NULL |

| $100,000 Pyramid, The | NULL |

| $1000 a Touchdown | 6.7 |

| $20,000 Carat, The | NULL |

| $21 a Day Once a Month | NULL |

| $2500 Bride, The | NULL |

| $30 | 7.5 |

| $30,000 | NULL |

| $300 y tickets | NULL |

| $40,000 | 9.6 |

| $5,000 Reward | NULL |

+-------------------------------------+-----------+

SELECT name,rankscore FROM movies LIMIT 20 OFFSET 40;

+-----------------------------------------------+-----------+

| name | rankscore |

+-----------------------------------------------+-----------+

| '60s Pop Rock Reunion | NULL |

| '60s, The | NULL |

| '68 | 5.5 |

| '70s Celebration: The Beat Is Back, A | NULL |

| '70s: The Decade That Changed Television, The | NULL |

| '88 Dodge Aries | NULL |

| '93 jie tou ba wang | NULL |

| '94 du bi dao zhi qing | 5.7 |

| '?' Motorist, The | 6.8 |

| 'A' | 7.1 |

| 'A' gai waak | 7.2 |

| 'A' gai waak juk jaap | 7.2 |

| 'Abbot' and 'Cresceus' Race, The | NULL |

| 'Akasaka no shimai' yori: yoru no wada | NULL |

| 'Amphitrite', The | NULL |

| 'Ang Galing galing mo, Babes' | NULL |

| 'Arriet's Baby | NULL |

| 'Arry and 'Arriet's Evening Out | NULL |

| 'Arry on the Steamboat | NULL |

| 'As se hizo' - Torremolinos 73 | NULL |

+-----------------------------------------------+-----------+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ORDER BY:

# list recent movies first

SELECT name,rankscore,year FROM movies ORDER BY year DESC LIMIT 10;

+------------------------------------------+-----------+------+

| name | rankscore | year |

+------------------------------------------+-----------+------+

| $5.15/Hr. | NULL | 2004 |

| '60s Pop Rock Reunion | NULL | 2004 |

| 'Elf' Jukebox | NULL | 2004 |

| 'As se hizo' - Torremolinos 73 | NULL | 2003 |

| 'Catch Me If You Can': Behind the Camera | NULL | 2003 |

| 'Ang Galing galing mo, Babes' | NULL | 2002 |

| $windle | 5.4 | 2002 |

| '88 Dodge Aries | NULL | 2002 |

| $300 y tickets | NULL | 2002 |

| '15' | 6.8 | 2002 |

+------------------------------------------+-----------+------+

# will show name, rankscore and year as column and all rows from ‘MOVIES’ in descending order wrt year

# default:ASC

SELECT name,rankscore,year FROM movies ORDER BY year LIMIT 10;

+-----------------------------------------------+-----------+------+

| name | rankscore | year |

+-----------------------------------------------+-----------+------+

| 'Amphitrite', The | NULL | 1897 |

| 'Brennus', Le | NULL | 1897 |

| 'Columbia' Winning the Cup | 3.6 | 1899 |

| 'Columbia' Close to the Wind | NULL | 1899 |

| 'Columbia' vs. 'Defender' | NULL | 1899 |

| 'Columbia' | NULL | 1899 |

| 'Corsair', The | NULL | 1899 |

| 'Erin', The | NULL | 1899 |

| 'Columbia' and 'Defender' Rounding Stake-Boat | NULL | 1899 |

| 'Courier' | NULL | 1900 |

+-----------------------------------------------+-----------+------+

# the output row order maynot be same as the one in the table due to query optimzier and internal data-structres/indices.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DISTINCT:

# list all genres of

SELECT DISTINCT genre FROM movies\_genres;

+-------------+

| genre |

+-------------+

| Documentary |

| Short |

| Comedy |

| Crime |

| Western |

| Family |

| Animation |

| Drama |

| Romance |

| Mystery |

| Thriller |

| Adult |

| Music |

| Action |

| Fantasy |

| Sci-Fi |

| Horror |

| War |

| Musical |

+-------------+

# will show all genre as column and rows having distinct genre in movies\_genres

# multiple-column DISTINCT

SELECT DISTINCT first\_name, last\_name FROM directors;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WHERE:

# list all movies with rankscore>9

SELECT name,year,rankscore FROM movies WHERE rankscore>4;

+--------------------------------------------------+------+-----------+

| name | year | rankscore |

+--------------------------------------------------+------+-----------+

| $ | 1971 | 6.4 |

| $1,000,000 Duck | 1971 | 5 |

| $1000 a Touchdown | 1939 | 6.7 |

| $30 | 1999 | 7.5 |

| $40,000 | 1996 | 9.6 |

| $pent | 2000 | 4.3 |

| $windle | 2002 | 5.4 |

| '15' | 2002 | 6.8 |

| '38 | 1987 | 6.7 |

| '49-'17 | 1917 | 5.8 |

| '68 | 1988 | 5.5 |

| '94 du bi dao zhi qing | 1994 | 5.7 |

| '?' Motorist, The | 1906 | 6.8 |

| 'A' | 1965 | 7.1 |

| 'A' gai waak | 1983 | 7.2 |

| 'A' gai waak juk jaap | 1987 | 7.2 |

| 'Babicky dobjejte presne!' | 1983 | 5.6 |

| 'Breaker' Morant | 1980 | 7.9 |

| 'Broadway Melody of 1940' | 1940 | 7.1 |

| 'Bullitt': Steve McQueen's Commitment to Reality | 1968 | 6.6 |

| 'burbs, The | 1989 | 5.9 |

| 'Crocodile' Dundee II | 1988 | 4.9 |

| 'E' | 1981 | 7.9 |

| 'El Chicko' - der Verdacht | 1995 | 4.1 |

+--------------------------------------------------+------+-----------+

SELECT name,year,rankscore FROM movies WHERE rankscore>4 ORDER BY rankscore DESC LIMIT 20;

# list all movies with rankscore>9 with in descending order wrt rankscore

+--------------------------------------------------+------+-----------+

| name | year | rankscore |

+--------------------------------------------------+------+-----------+

| $40,000 | 1996 | 9.6 |

| 'E' | 1981 | 7.9 |

| 'Breaker' Morant | 1980 | 7.9 |

| $30 | 1999 | 7.5 |

| 'A' gai waak | 1983 | 7.2 |

| 'A' gai waak juk jaap | 1987 | 7.2 |

| 'A' | 1965 | 7.1 |

| 'Broadway Melody of 1940' | 1940 | 7.1 |

| '15' | 2002 | 6.8 |

| '?' Motorist, The | 1906 | 6.8 |

| $1000 a Touchdown | 1939 | 6.7 |

| '38 | 1987 | 6.7 |

| 'Bullitt': Steve McQueen's Commitment to Reality | 1968 | 6.6 |

| $ | 1971 | 6.4 |

| 'burbs, The | 1989 | 5.9 |

| '49-'17 | 1917 | 5.8 |

| '94 du bi dao zhi qing | 1994 | 5.7 |

| 'Babicky dobjejte presne!' | 1983 | 5.6 |

| '68 | 1988 | 5.5 |

| $windle | 2002 | 5.4 |

+--------------------------------------------------+------+-----------+

# Condition's outputs: TRUE, FALSE, NULL

# Comparison Operators: = , <> or != , < , <= , >, >=

SELECT \* FROM movies\_genres WHERE genre = 'Comedy';

# list all movies\_genres where genre is equal to ‘COMEDY’

+----------+--------+

| movie\_id | genre |

+----------+--------+

| 2 | Comedy |

| 6 | Comedy |

| 8 | Comedy |

| 11 | Comedy |

| 15 | Comedy |

| 18 | Comedy |

| 21 | Comedy |

| 28 | Comedy |

| 29 | Comedy |

| 35 | Comedy |

| 47 | Comedy |

| 50 | Comedy |

| 52 | Comedy |

| 53 | Comedy |

| 59 | Comedy |

| 73 | Comedy |

| 74 | Comedy |

| 76 | Comedy |

+----------+--------+

SELECT \* FROM movies\_genres WHERE genre <> 'Horror';

+----------+-------------+

| movie\_id | genre |

+----------+-------------+

| 1 | Documentary |

| 1 | Short |

| 2 | Comedy |

| 2 | Crime |

| 5 | Western |

| 6 | Comedy |

| 6 | Family |

| 8 | Animation |

| 8 | Comedy |

| 8 | Short |

| 9 | Drama |

| 10 | Family |

| 11 | Comedy |

| 12 | Crime |

| 12 | Drama |

| 12 | Short |

| 13 | Animation |

| 13 | Short |

| 14 | Drama |

| 14 | Romance |

| 14 | Short |

| 15 | Comedy |

| 15 | Short |

| 17 | Short |

| 18 | Comedy |

| 19 | Mystery |

| 20 | Crime |

| 20 | Thriller |

| 21 | Comedy |

| 21 | Drama |

| 22 | Drama |

| 24 | Adult |

| 25 | Short |

| 26 | Western |

| 28 | Comedy |

| 28 | Short |

| 29 | Comedy |

| 29 | Short |

| 31 | Drama |

| 32 | Documentary |

| 32 | Music |

| 33 | Action |

| 33 | Crime |

| 33 | Thriller |

| 34 | Short |

| 35 | Comedy |

| 35 | Short |

| 36 | Documentary |

| 36 | Short |

| 37 | Crime |

| 38 | Drama |

| 40 | Documentary |

| 41 | Western |

| 42 | Music |

| 43 | Drama |

| 44 | Drama |

| 45 | Music |

| 46 | Documentary |

| 47 | Comedy |

| 47 | Short |

| 50 | Comedy |

| 50 | Fantasy |

| 50 | Sci-Fi |

| 50 | Short |

| 51 | Short |

| 52 | Action |

| 52 | Comedy |

| 53 | Action |

| 53 | Comedy |

| 53 | Crime |

| 56 | Documentary |

| 57 | Drama |

| 58 | Short |

| 59 | Comedy |

| 60 | Short |

| 61 | Documentary |

| 61 | Short |

| 62 | Sci-Fi |

| 64 | Documentary |

| 66 | Documentary |

| 67 | Documentary |

| 68 | Documentary |

| 68 | Short |

| 69 | Drama |

| 69 | War |

| 70 | Documentary |

| 70 | Short |

| 71 | Musical |

| 72 | Documentary |

| 72 | Short |

| 73 | Comedy |

| 73 | Family |

| 74 | Comedy |

| 74 | Short |

| 76 | Animation |

| 76 | Comedy |

| 76 | Short |

| 77 | Documentary |

+----------+-------------+

NULL => doesnot-exist/unknown/missing

# "=" doesnot work with NULL, will give you an empty result-set.

SELECT name,year,rankscore FROM movies WHERE rankscore = NULL;

O/P:

Empty set

SELECT name,year,rankscore FROM movies WHERE rankscore IS NULL LIMIT 20;

O/P:

Empty set

SELECT name,year,rankscore FROM movies WHERE rankscore IS NOT NULL LIMIT 20;

O/P:

Empty set

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# LOGICAL OPERATORS: AND, OR, NOT, ALL, ANY, BETWEEN, EXISTS, IN, LIKE, SOME

# website search filters

SELECT name,year,rankscore FROM movies WHERE rankscore>4 AND year>2000;

+---------+------+-----------+

| name | year | rankscore |

+---------+------+-----------+

| $windle | 2002 | 5.4 |

| '15' | 2002 | 6.8 |

+---------+------+-----------+

SELECT name,year,rankscore FROM movies WHERE NOT year<=2000 LIMIT 20;

| name | year | rankscore |

+------------------------------------------+------+-----------+

| #28 | 2002 | NULL |

| $100,000 Pyramid, The | 2001 | NULL |

| $300 y tickets | 2002 | NULL |

| $5.15/Hr. | 2004 | NULL |

| $windle | 2002 | 5.4 |

| '15' | 2002 | 6.8 |

| '60s Pop Rock Reunion | 2004 | NULL |

| '88 Dodge Aries | 2002 | NULL |

| 'Ang Galing galing mo, Babes' | 2002 | NULL |

| 'As se hizo' - Torremolinos 73 | 2003 | NULL |

| 'Billy Elliot' Boy, The | 2001 | NULL |

| 'Catch Me If You Can': Behind the Camera | 2003 | NULL |

| 'Elf' Jukebox | 2004 | NULL |

+------------------------------------------+------+-----------+

SELECT name,year,rankscore FROM movies WHERE rankscore>4 OR year>2007;

+--------------------------------------------------+------+-----------+

| name | year | rankscore |

+--------------------------------------------------+------+-----------+

| $ | 1971 | 6.4 |

| $1,000,000 Duck | 1971 | 5 |

| $1000 a Touchdown | 1939 | 6.7 |

| $30 | 1999 | 7.5 |

| $40,000 | 1996 | 9.6 |

| $pent | 2000 | 4.3 |

| $windle | 2002 | 5.4 |

| '15' | 2002 | 6.8 |

| '38 | 1987 | 6.7 |

| '49-'17 | 1917 | 5.8 |

| '68 | 1988 | 5.5 |

| '94 du bi dao zhi qing | 1994 | 5.7 |

| '?' Motorist, The | 1906 | 6.8 |

| 'A' | 1965 | 7.1 |

| 'A' gai waak | 1983 | 7.2 |

| 'A' gai waak juk jaap | 1987 | 7.2 |

| 'Babicky dobjejte presne!' | 1983 | 5.6 |

| 'Breaker' Morant | 1980 | 7.9 |

| 'Broadway Melody of 1940' | 1940 | 7.1 |

| 'Bullitt': Steve McQueen's Commitment to Reality | 1968 | 6.6 |

| 'burbs, The | 1989 | 5.9 |

| 'Crocodile' Dundee II | 1988 | 4.9 |

| 'E' | 1981 | 7.9 |

| 'El Chicko' - der Verdacht | 1995 | 4.1 |

+--------------------------------------------------+------+-----------+

# will discsuss about ANY and ALL when we discuss sub-queries

SELECT name,year,rankscore FROM movies WHERE year BETWEEN 1990 AND 2000;

+-----------------------------------------------+------+-----------+

| name | year | rankscore |

+-----------------------------------------------+------+-----------+

| #7 Train: An Immigrant Journey, The | 2000 | NULL |

| $30 | 1999 | 7.5 |

| $40,000 | 1996 | 9.6 |

| $pent | 2000 | 4.3 |

| & frres | 2000 | NULL |

| '49, un souffle de colre | 1996 | NULL |

| '60s, The | 1999 | NULL |

| '70s Celebration: The Beat Is Back, A | 1993 | NULL |

| '70s: The Decade That Changed Television, The | 2000 | NULL |

| '93 jie tou ba wang | 1993 | NULL |

| '94 du bi dao zhi qing | 1994 | 5.7 |

| 'Bats' Abound | 1999 | NULL |

| 'Belle de Varsovie, La' | 1993 | NULL |

| 'Betty Bee' (sopravvivere d'arte) | 1999 | NULL |

| 'C' Word, The | 1993 | NULL |

| 'El Chicko' - der Verdacht | 1995 | 4.1 |

+-----------------------------------------------+------+-----------+

#inclusive: year>=1999 and year<=2000

SELECT name,year,rankscore FROM movies WHERE year BETWEEN 2000 AND 1999;

#lowvalue <= highvalue else you will get an empty result set

SELECT director\_id, genre FROM directors\_genres WHERE genre IN ('Comedy','Horror');

# same as genre='Comedy' OR genre='Horror'

SELECT name,year,rankscore FROM movies WHERE name LIKE 'Tis%';

# % => wildcard character to imply zero or more characters

SELECT first\_name, last\_name FROM actors WHERE first\_name LIKE '%es';

# first name ending in 'es'

+------------+-----------+

| first\_name | last\_name |

+------------+-----------+

| James | 52X |

| T??rres | Aadland |

+------------+-----------+

SELECT first\_name, last\_name FROM actors WHERE first\_name LIKE '%es%';

#first name contains 'es'

+------------+-----------+

| first\_name | last\_name |

+------------+-----------+

| James | 52X |

| T??rres | Aadland |

| Jesper | Aagaard |

+------------+-----------+

SELECT first\_name, last\_name FROM actors WHERE first\_name LIKE 'Agn\_s';

# '\_' implies exactly one character.

# If we want to macth % or \_, we should use the backslash as the escape character: \% and \\_

SELECT first\_name, last\_name FROM actors WHERE first\_name LIKE 'L%' AND first\_name NOT LIKE 'Li%';

+--------------+------------+

| first\_name | last\_name |

+--------------+------------+

| Luis | 'El Plojo' |

| Luis Roberto | 'Formiga' |

| Lucky | 7 |

| Lejaren | a'Hiller |

| Lasse | Aagaard |

| Lee | Aaker |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Aggregate functions: Computes a single value on a set of rows and returns the aggreagate

COUNT, MIN, MAX, SUM, AVG

SELECT MIN(year) FROM movies;

+-----------+

| MIN(year) |

+-----------+

| 1897 |

+-----------+

SELECT MAX(year) FROM movies;

+-----------+

| MAX(year) |

+-----------+

| 2004 |

+-----------+

SELECT COUNT(\*) FROM movies;

+----------+

| COUNT(\*) |

+----------+

| 100 |

+----------+

SELECT COUNT(\*) FROM movies where year>2000;

+----------+

| COUNT(\*) |

+----------+

| 13 |

+----------+

SELECT COUNT(year) FROM movies;

+-------------+

| COUNT(year) |

+-------------+

| 100 |

+-------------+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GROUP-BY

# find number of movies released per year

SELECT year, COUNT(year) FROM movies GROUP BY year;

+------+-------------+

| year | COUNT(year) |

+------+-------------+

| 2002 | 6 |

| 2000 | 4 |

| 1971 | 2 |

| 1913 | 3 |

| 1915 | 4 |

| 1923 | 1 |

| 1920 | 2 |

| 1921 | 1 |

| 2001 | 2 |

| 1939 | 1 |

| 1941 | 1 |

| 1912 | 2 |

| 1999 | 4 |

| 1996 | 2 |

| 1918 | 1 |

| 1914 | 3 |

| 2004 | 3 |

| 1980 | 2 |

| 1989 | 2 |

| 1975 | 1 |

| 1924 | 1 |

| 1911 | 2 |

| 1986 | 1 |

| 1968 | 2 |

| 1987 | 2 |

| 1951 | 1 |

| 1917 | 1 |

| 1988 | 2 |

| 1993 | 4 |

| 1994 | 1 |

| 1906 | 1 |

| 1965 | 1 |

| 1983 | 3 |

| 1901 | 9 |

| 1960 | 1 |

| 1897 | 2 |

| 1909 | 1 |

| 2003 | 2 |

| 1984 | 1 |

| 1940 | 1 |

| 1899 | 7 |

| 1932 | 1 |

| 1900 | 1 |

| 1938 | 1 |

| 1902 | 1 |

| 1919 | 1 |

| 1981 | 1 |

| 1995 | 1 |

+------+-------------+

SELECT year, COUNT(year) FROM movies GROUP BY year ORDER BY year;

+------+-------------+

| year | COUNT(year) |

+------+-------------+

| 1897 | 2 |

| 1899 | 7 |

| 1900 | 1 |

| 1901 | 9 |

| 1902 | 1 |

| 1906 | 1 |

| 1909 | 1 |

| 1911 | 2 |

| 1912 | 2 |

| 1913 | 3 |

| 1914 | 3 |

| 1915 | 4 |

| 1917 | 1 |

| 1918 | 1 |

| 1919 | 1 |

| 1920 | 2 |

| 1921 | 1 |

| 1923 | 1 |

| 1924 | 1 |

| 1932 | 1 |

| 1938 | 1 |

| 1939 | 1 |

| 1940 | 1 |

| 1941 | 1 |

| 1951 | 1 |

| 1960 | 1 |

| 1965 | 1 |

| 1968 | 2 |

| 1971 | 2 |

| 1975 | 1 |

| 1980 | 2 |

| 1981 | 1 |

| 1983 | 3 |

| 1984 | 1 |

| 1986 | 1 |

| 1987 | 2 |

| 1988 | 2 |

| 1989 | 2 |

| 1993 | 4 |

| 1994 | 1 |

| 1995 | 1 |

| 1996 | 2 |

| 1999 | 4 |

| 2000 | 4 |

| 2001 | 2 |

| 2002 | 6 |

| 2003 | 2 |

| 2004 | 3 |

+------+-------------+

SELECT year, COUNT(year) year\_count FROM movies GROUP BY year ORDER BY year\_count;

# year\_count is an alias.

+------+------------+

| year | year\_count |

+------+------------+

| 1939 | 1 |

| 1951 | 1 |

| 1932 | 1 |

| 1941 | 1 |

| 1917 | 1 |

| 1960 | 1 |

| 1900 | 1 |

| 1975 | 1 |

| 1938 | 1 |

| 1924 | 1 |

| 1909 | 1 |

| 1902 | 1 |

| 1923 | 1 |

| 1994 | 1 |

| 1919 | 1 |

| 1918 | 1 |

| 1986 | 1 |

| 1906 | 1 |

| 1984 | 1 |

| 1981 | 1 |

| 1921 | 1 |

| 1965 | 1 |

| 1940 | 1 |

| 1995 | 1 |

| 1980 | 2 |

| 1971 | 2 |

| 1989 | 2 |

| 1912 | 2 |

| 1988 | 2 |

| 1897 | 2 |

| 1996 | 2 |

| 1911 | 2 |

| 2003 | 2 |

| 1920 | 2 |

| 1968 | 2 |

| 2001 | 2 |

| 1987 | 2 |

| 1913 | 3 |

| 1914 | 3 |

| 2004 | 3 |

| 1983 | 3 |

| 2000 | 4 |

| 1915 | 4 |

| 1999 | 4 |

| 1993 | 4 |

| 2002 | 6 |

| 1899 | 7 |

| 1901 | 9 |

+------+------------+

# often used with COUNT, MIN, MAX or SUM.

# if grouping columns contain NULL values, all null values are grouped together.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HAVING:

# Print years which have >1000 movies in our DB [Data Scientist for Analysis]

SELECT year, COUNT(year) year\_count FROM movies GROUP BY year HAVING year\_count>1000;

# specify a condition on groups using HAVING.

Order of execution:

1. GROUP BY to create groups

2. apply the AGGREGATE FUNCTION

3. Apply HAVING condition.

# often used along with GROUP BY. Not Mandatory.

SELECT name, year FROM movies HAVING year>2000;

# HAVING without GROUP BY is same as WHERE

| name | year |

+------------------------------------------+------+

| #28 | 2002 |

| $100,000 Pyramid, The | 2001 |

| $300 y tickets | 2002 |

| $5.15/Hr. | 2004 |

| $windle | 2002 |

| '15' | 2002 |

| '60s Pop Rock Reunion | 2004 |

| '88 Dodge Aries | 2002 |

| 'Ang Galing galing mo, Babes' | 2002 |

| 'As se hizo' - Torremolinos 73 | 2003 |

| 'Billy Elliot' Boy, The | 2001 |

| 'Catch Me If You Can': Behind the Camera | 2003 |

| 'Elf' Jukebox | 2004 |

+------------------------------------------+------+

SELECT year, COUNT(year) year\_count FROM movies WHERE rankscore>9 GROUP BY year HAVING year\_count>20;

# HAVING vs WHERE

## WHERE is applied on individual rows while HAVING is applied on groups.

## HAVING is applied after grouping while WHERE is used before grouping.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

JOINs:

#combine data in multiple tables

# For each movie, print name and the genres

SELECT m.name, g.genre from movies m JOIN movies\_genres g ON m.id=g.movie\_id LIMIT 20;

| name | genre |

+-------------------------------------+-------------+

| #7 Train: An Immigrant Journey, The | Documentary |

| #7 Train: An Immigrant Journey, The | Short |

| $ | Comedy |

| $ | Crime |

| $1,000 Reward | Western |

| $1,000,000 Duck | Comedy |

| $1,000,000 Duck | Family |

| $10,000 Under a Pillow | Animation |

| $10,000 Under a Pillow | Comedy |

| $10,000 Under a Pillow | Short |

| $100,000 | Drama |

| $100,000 Pyramid, The | Family |

| $1000 a Touchdown | Comedy |

| $20,000 Carat, The | Crime |

| $20,000 Carat, The | Drama |

| $20,000 Carat, The | Short |

| $21 a Day Once a Month | Animation |

| $21 a Day Once a Month | Short |

| $2500 Bride, The | Drama |

| $2500 Bride, The | Romance |

+-------------------------------------+-------------+

# table aliases: m and g

# natural join: a join where we have the same column-names across two tables.

#T1: C1, C2

#T2: C1, C3, C4

SELECT \* FROM T1 JOIN T2;

SELECT \* FROM T1 JOIN T2 USING (C1);

# returns C1,C2,C3,C4

# no need to use the keyword "ON"

# Inner join (default) vs left outer vs right outer vs full-outer join.

T1: C1, C2, C3

SELECT m.name, g.genre from movies m LEFT JOIN movies\_genres g ON m.id=g.movie\_id LIMIT 20;

| name | genre |

+-------------------------------------+-------------+

| #7 Train: An Immigrant Journey, The | Documentary |

| #7 Train: An Immigrant Journey, The | Short |

| $ | Comedy |

| $ | Crime |

| $1,000 Reward | Western |

| $1,000,000 Duck | Comedy |

| $1,000,000 Duck | Family |

| $10,000 Under a Pillow | Animation |

| $10,000 Under a Pillow | Comedy |

| $10,000 Under a Pillow | Short |

| $100,000 | Drama |

| $100,000 Pyramid, The | Family |

| $1000 a Touchdown | Comedy |

| $20,000 Carat, The | Crime |

| $20,000 Carat, The | Drama |

| $20,000 Carat, The | Short |

| $21 a Day Once a Month | Animation |

| $21 a Day Once a Month | Short |

| $2500 Bride, The | Drama |

| $2500 Bride, The | Romance |

+-------------------------------------+-------------+

#LEFT JOIN or LEFT OUTER JOIN

#RIGHT JOIN or RIGHT OUTER JOIN

#FULL JOIN or FULL OUTER JOIN

#JOIN or INNER JOIN

# NULL for missing counterpart rows.

# 3-way joins and k-way joins

SELECT a.first\_name, a.last\_name FROM actors a JOIN roles r ON a.id=r.actor\_id JOIN movies m on m.id=r.movie\_id AND m.name='Officer 444';

#Practical note about joins: Joins can be expensive computationally when we have large tables.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub-Queries or Nested Queries or Inner Queries

# List all actors in the movie Schindler's List

#https://www.imdb.com/title/tt0108052/fullcredits/?ref\_=tt\_ov\_st\_sm

SELECT first\_name, last\_name from actors WHERE id IN

( SELECT actor\_id from roles WHERE movie\_id IN

(SELECT id FROM movies where name='Schindler's List)

);

# Syntax:

SELECT column\_name [, column\_name ]

FROM table1 [, table2 ]

WHERE column\_name OPERATOR

(SELECT column\_name [, column\_name ]

FROM table1 [, table2 ]

[WHERE])

# first the innner query is executed and then the outer query is executed using the output values in the inner query

# IN, NOT IN, EXISTS, NOT EXISTS, ANY, ALL, Comparison operators

#EXISTS returns true if the subquery returns one or more records or NULL

# ANY operator returns TRUE if any of the subquery values meet the condition.

# ALL operator returns TRUE if all of the subquery values meet the condition.

SELECT \* FROM movies where rankscore >= ALL (SELECT MAX(rankscore) from movies);

# get all movies whose rankscore is same as the maximum rankscore.

# e.g: rankscore <> ALL(...)

# https://en.wikipedia.org/wiki/Correlated\_subquery

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Data Manupulation Language: SELECT, INSERT, UPDATE, DELETE

INSERT INTO movies(id, name, year, rankscore) VALUES (412321, 'Thor', 2011, 7);

INSERT INTO movies(id, name, year, rankscore) VALUES (412321, 'Thor', 2011, 7), (412322, 'Iron Man', 2008, 7.9), (412323, 'Iron Man 2', 2010, 7);

# INSERT FROM one table to annother using nnested sub query: https://en.wikipedia.org/wiki/Insert\_(SQL)#Copying\_rows\_from\_other\_tables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# UPDATE Command

UPDATE <TableName> SET col1=val1, col2=val2 WHERE condition

UPDATE movies SET rankscore=9 where id=412321;

# Update multiple rows also.

# Can be used along with Sub-queries.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#DELETE

DELETE FROM movies WHERE id=412321;

# Remove all rows: TRUNCATE TABLE TableName;

# Same as selete without a WHERE Clause.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Data Definition Language

CREATE TABLE language ( id INT PRIMARY, lang VARCHAR(50) NOT NULL);

# Datatypes: https://www.journaldev.com/16774/sql-data-types

# Constraints: https://www.w3schools.com/sql/sql\_constraints.asp

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

FOREIGN KEY - Uniquely identifies a row/record in another table

CHECK - Ensures that all values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column when no value is specified

INDEX - Used to create and retrieve data from the database very quickly

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ALTER: ADD, MODIFY, DROP

ALTER TABLE language ADD country VARCHAR(50);

ALTER TABLE language MODIFY country VARCHAR(60);

ALTER TABLE langauge DROP country;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Removes both the table and all of the data permanently.

DROP TABLE Tablename;

DROP TABLE TableName IF EXISTS;

#https://dev.mysql.com/doc/refman/8.0/en/drop-table.html

TRUNCATE TABLE TableName;

# as discussed earlier same as DELETE FROM TableName;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*