Please complete each of the following exercises #1-18. For each exercise, include the question followed by your query, the columns you received, and the expected row count along with any pertinent screenshots of your terminal.

Using the Avengers database, complete the queries below.

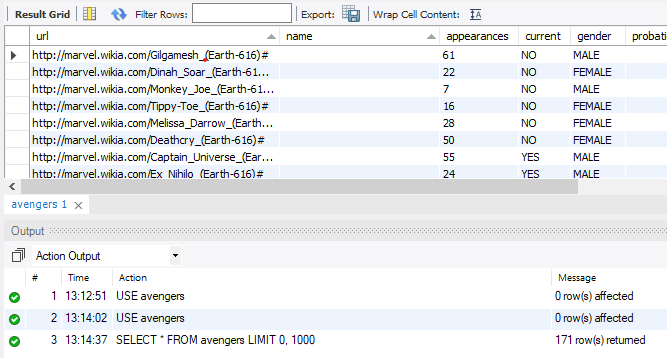
1. Show all of the information on the Avengers.
2. Query:

SELECT \* FROM avengers;

1. Columns:

|  |  |
| --- | --- |
| url | varchar(100) |
| name | varchar(50) |
| appearances | varchar(5) |
| current | varchar(3) |
| gender | varchar(6) |
| probationaryIntro | varchar(25) |
| fullAvengersIntro | varchar(25) |
| year | int |
| yearsSinceJoining | int |
| honorary | varchar(25) |
| death1 | varchar(5) |
| return1 | varchar(5) |
| death2 | varchar(5) |
| return2 | varchar(5) |
| death3 | varchar(5) |
| return3 | varchar(5) |
| death4 | varchar(5) |
| return4 | varchar(5) |
| death5 | varchar(5) |
| return5 | varchar(5) |
| notes | varchar(250) |

1. Expected Row Count: 171
2. Screenshot:



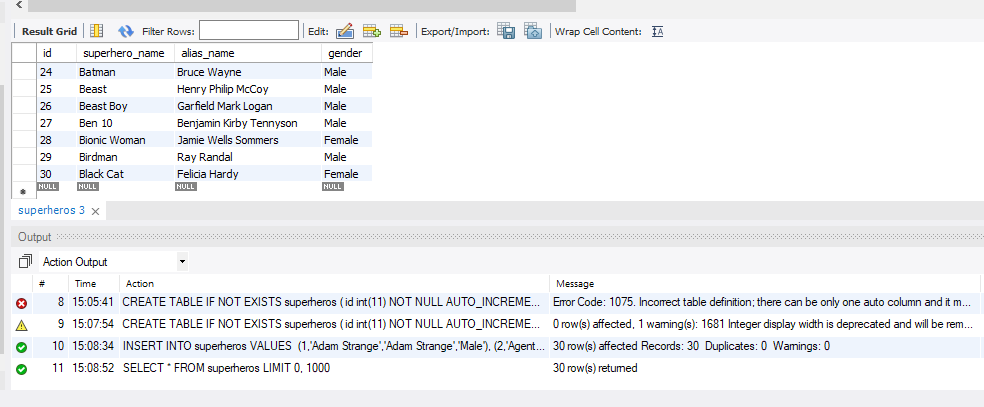
1. Create a new table with information from another super hero group [can be imported from elsewhere or fictitious]. Make sure there is an appropriate header and must be at least 30 rows.
2. Query:

CREATE TABLE IF NOT EXISTS superheros (  
 id int(11) NOT NULL AUTO\_INCREMENT,  
 superhero\_name varchar(200),  
 alias\_name varchar(200),  
 gender varchar(6),

PRIMARY KEY (id)  
);

INSERT INTO superheros VALUES   
(1,'Adam Strange','Adam Strange','Male'),  
(2,'Agent 13','Sharon Carter','Female'),  
(3,'Agent Zero','Christoph Nord','Male'),  
(4,'Alex Mercer','Alexander J. Mercer','Male'),  
(5,'Ando Masahashi','Ando Masahashi','Male'),  
(6,'Angel','Liam','Male'),  
(7,'Angel Dust','Christina','Female'),  
(8,'Angel Salvadore','Angel Salvadore Bohusk','Male'),  
(9,'Ant-Man','Henry Jonathan Pym','Male'),  
(10,'Anti-Venom','Edward Charles Allan Brock','Male'),  
(11,'Apocalypse','En Sabah Nur','Male'),  
(12,'Aquababy','Arthur Curry, Jr.','Male'),  
(13,'Aquaman','Orin','Male'),  
(14,'Arachne','Julia Carpenter','Female'),  
(15,'Archangel','Warren Kenneth Worthington III','Male'),  
(16,'Arsenal','Roy William Harper, Jr.','Male'),  
(17,'Astro Boy',NULL,'Male'),  
(18,'Atlas','Erik Stephan Josten','Male'),  
(19,'Atom','Albert Pratt','Male'),  
(20,'Atom Girl','Salu Digby','Female'),  
(21,'Banshee','Sean Cassidy','Male'),  
(22,'Bantam','Roberto Velasquez','Male'),  
(23,'Batgirl','Mary Elizabeth Kane','Female'),  
(24,'Batman','Bruce Wayne','Male'),  
(25,'Beast','Henry Philip McCoy','Male'),  
(26,'Beast Boy','Garfield Mark Logan','Male'),  
(27,'Ben 10','Benjamin Kirby Tennyson','Male'),  
(28,'Bionic Woman','Jamie Wells Sommers','Female'),  
(29,'Birdman','Ray Randal','Male'),  
(30,'Black Cat','Felicia Hardy','Female')  
;

1. Columns: id, superhero\_name, alias\_name, gender
2. Expected Row Count: 30
3. Screenshot:

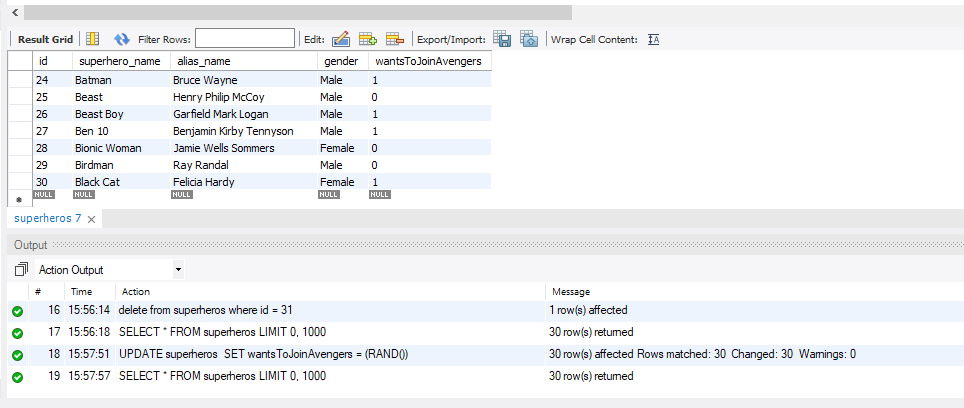


1. Add a column to your new table (the new superheros) for those who are interested in joining the Avengers. Make this be randomly assigned.
2. Query:

ALTER TABLE superheros   
ADD wantsToJoinAvengers boolean;

UPDATE superheros   
 SET wantsToJoinAvengers = (RAND());

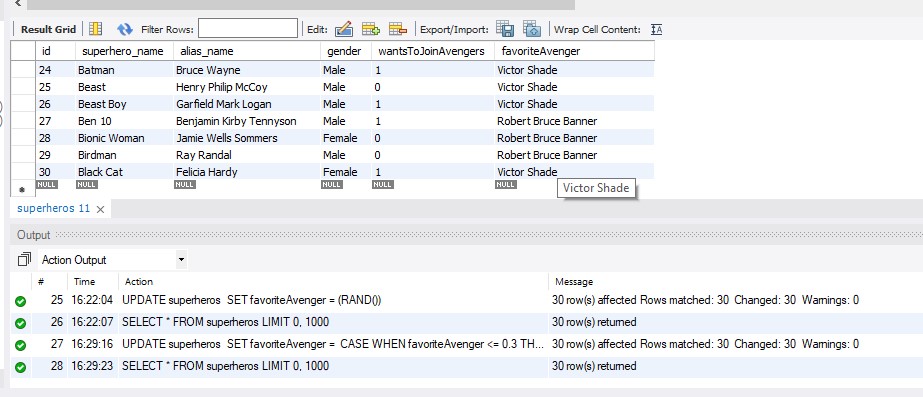
1. Columns: id, superhero\_name, alias\_name, gender, wantsToJoinAvengers
2. Expected Row Count: 30
3. Screenshot:



1. Add a column to your new table (the new superheros) for their favorite Avenger (have it randomly choose between Anthony Edward "Tony" Stark, Robert Bruce Banner, & Victor Shade).
2. Query:

ALTER TABLE superheros   
 ADD favoriteAvenger varchar(200);  
  
 UPDATE superheros   
 SET favoriteAvenger =   
 CASE WHEN favoriteAvenger <= 0.3 THEN 'Anthony Edward "Tony" Stark'  
 WHEN favoriteAvenger > 0.3 AND favoriteAvenger <= 0.6 THEN 'Robert Bruce Banner'  
 WHEN favoriteAvenger >0.6 THEN 'Victor Shade'   
 END;

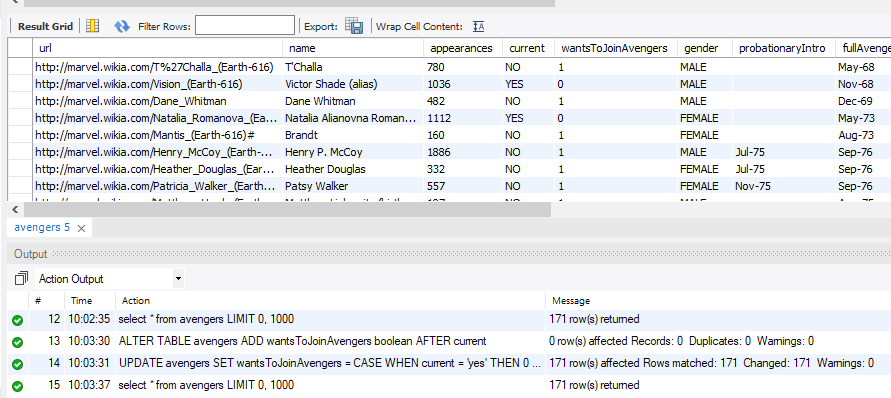
1. Columns: id, superhero\_name, alias\_name, gender, wantsToJoinAvengers, favoriteAvenger
2. Expected Row Count: 30
3. Screenshot:



1. Create an association between the two tables for who is interested in joining the Avengers.
2. Query:

ALTER TABLE avengers  
ADD wantsToJoinAvengers boolean  
AFTER current;  
  
UPDATE avengers  
SET wantsToJoinAvengers =  
CASE WHEN current = 'yes' THEN 0  
WHEN current = 'no' THEN 1  
END;

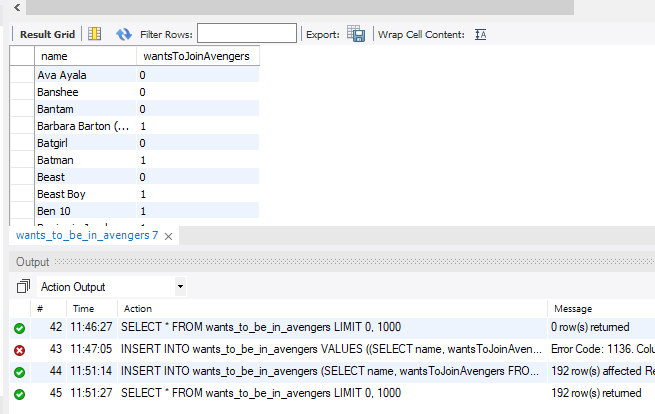
1. Columns: all
2. Expected Row Count: 171
3. Screenshot:



1. Show your association and create a new table for that association.
2. Query:

CREATE TABLE wants\_to\_be\_in\_avengers (  
name varchar(50),  
wantsToJoinAvengers BOOLEAN  
);  
  
INSERT INTO wants\_to\_be\_in\_avengers  
(SELECT name, wantsToJoinAvengers FROM avengers  
UNION   
SELECT superhero\_name, wantsToJoinAvengers FROM superheros  
ORDER BY name);

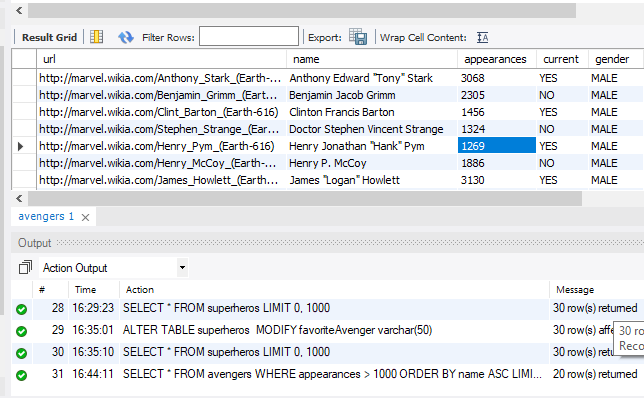
1. Columns: name, wantsToJoinAvengers
2. Expected Row Count: returned 192
3. Screenshot:

****

1. List all Avengers who have more than 1000 appearances in alphabetic order.
2. Query:

SELECT \* FROM avengers  
WHERE appearances > 1000  
ORDER BY name ASC;

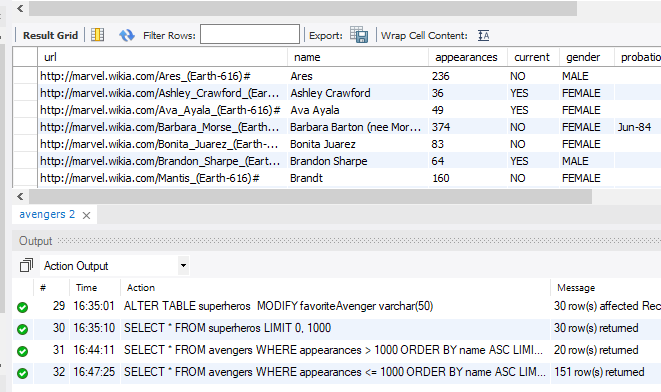
1. Columns: all
2. Expected Row Count: returned 20
3. Screenshot:



1. What Avengers do not have more than 1000 appearances?
2. Query:

SELECT \* FROM avengers  
WHERE appearances <= 1000  
ORDER BY name ASC;

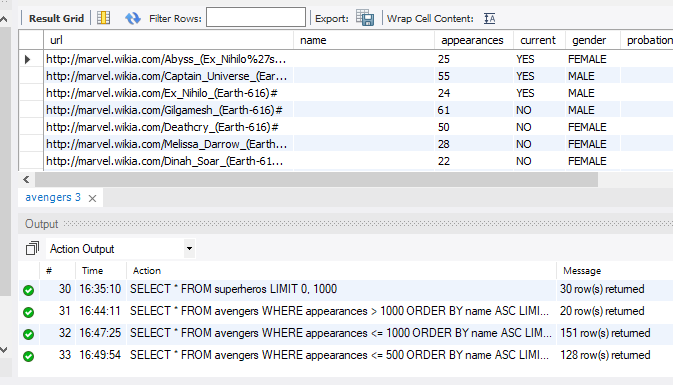
1. Columns: all
2. Expected Row Count: returned 151
3. Screenshot:



1. Not more than 500?
2. Query:

SELECT \* FROM avengers  
WHERE appearances <= 500  
ORDER BY name ASC;

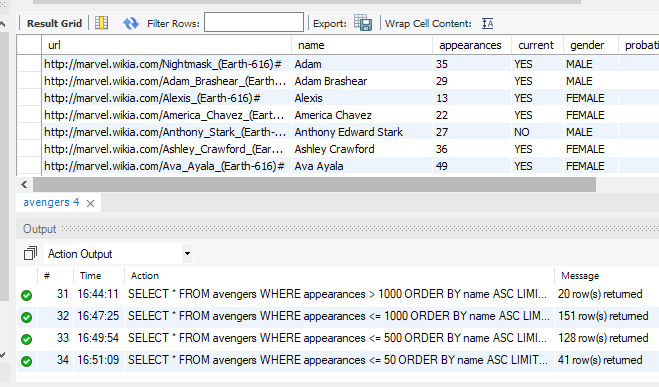
1. Columns: all
2. Expected Row Count: returned 128
3. Screenshot:



1. Not more than 50?
2. Query:

SELECT \* FROM avengers  
WHERE appearances <= 50  
ORDER BY name ASC;

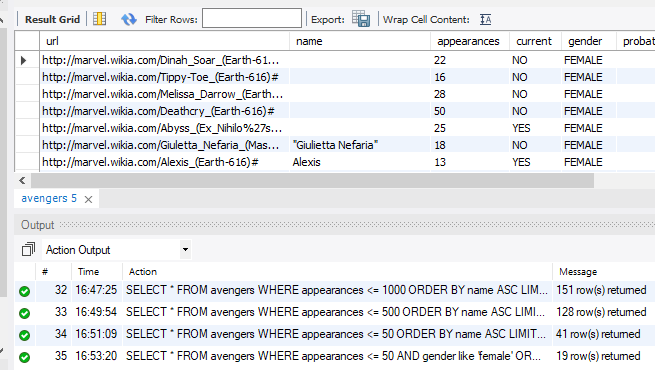
1. Columns: all
2. Expected Row Count: returned 41
3. Screenshot:



1. Not more than 50 and female?
2. Query:

SELECT \* FROM avengers  
WHERE appearances <= 50  
AND gender like 'female'  
ORDER BY name ASC;

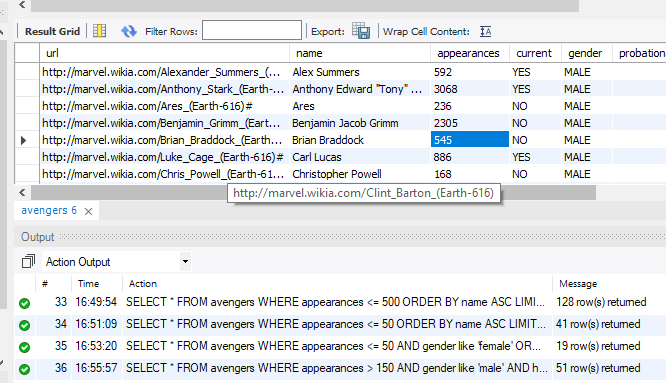
1. Columns: all
2. Expected Row Count: returned 19
3. Screenshot:



1. More than 150, male, and are full members?
2. Query:

SELECT \* FROM avengers  
WHERE appearances > 150  
AND gender like 'male'  
AND honorary like 'full'  
ORDER BY name ASC;

1. Columns: all
2. Expected Row Count: returned 51
3. Screenshot:



1. Write a function that can add a new superhero to the Avengers if they are interested in becoming an Avenger and if their favorite Avenger is Victor Shade.
2. Query:

DELIMITER //  
CREATE FUNCTION addToAvengers(wantsToJoinAvengers boolean, favoriteAvenger varchar(50))  
RETURNS varchar(50)

READS SQL DATA  
BEGIN  
 DECLARE current varchar(3);  
 SET current = 'no';  
   
 SET current =  
 CASE WHEN wantsToJoinAvengers = 1 AND favoriteAvenger = 'Victor Shade'  
 THEN 'yes'  
 ELSE 'no' END;  
 RETURN current;  
END;  
// DELIMITER ;

1. Columns: 0
2. Expected Row Count: 0
3. Screenshot:



1. Write a function that counts the total amount of records(rows/entries) in a given table.
2. Query:

DELIMITER //  
CREATE FUNCTION totalRecords(column\_name VARCHAR(50))  
RETURNS INT  
READS SQL DATA  
BEGIN  
 DECLARE totalRecordsInTable INT;  
 SET totalRecordsInTable = 0;  
   
 SET totalRecordsInTable = COUNT(column\_name);  
 RETURN totalRecordsInTable;  
END;  
// DELIMITER ;

1. Columns: 0
2. Expected Row Count: 0
3. Screenshot:



1. Call your totalRecords function on the avengers table. What is the output?
2. Query:

SELECT totalRecords(name) AS 'total records' FROM avengers;

1. Columns: Total Records
2. Expected Row Count: 1
3. Screenshot:



1. There is room for 3 new Avengers so invoke your newMember function 3 times to add 3 new members to the Avengers.
2. Query:

SELECT superhero\_name, current, addToAvengers(superheros.wantsToJoinAvengers, favoriteAvenger) FROM superheros  
JOIN avengers;

1. Columns: name, current, addToAvengers
2. Expected Row Count: returned 1000 (looked at everyone, not 1 at a time)
3. Screenshot:

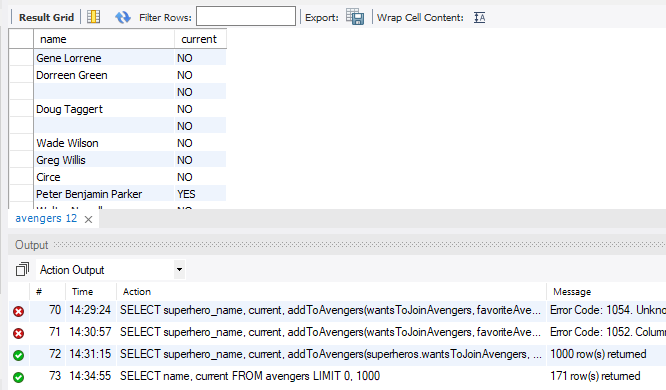


1. Show that the addition of new members was successful.

1. Query:

SELECT name, current FROM avengers;

1. Columns: name, current
2. Expected Row Count: more than 171
3. Screenshot:



1. Remove any tables that are not the original avengers table and prove that they have been removed.

1. Query:

DROP TABLE superheros, wants\_to\_be\_in\_avengers;

1. Columns: 0
2. Expected Row Count: 0
3. Screenshot:

