JOIN Quiz 2

JOIN Quiz - part 2

SELECT basics
quiz
SELECT from world
quiz
SELECT from nobel
quiz
SELECT in SELECT
quiz
SUM and COUNT
quiz
JOIN
quiz
More JOIN
quiz
Using NULL
quiz
Self JOIN
quiz

Reference
NoSQL zoo
SELECT
Functions
SELECT .. WHERE
SELECT .. GROUP BY
SELECT .. JOIN
SELECT .. SELECT
INSERT .. VALUES
INSERT .. SELECT
UPDATE
DELETE
CREATE TABLE
CREATE VIEW
CREATE INDEX
DROP
ALTER
UNION
LEFT JOIN
NULL

Tools
What links here
Related changes
Special pages
Printable version
Permanent link
Page information

1. Select the statement which lists the unfortunate directors of the movies which have caused financial loses (gross < budget)

```
SELECT JOIN(name FROM actor, movie
       ON actor.id:director WHERE gross < budget)
 GROUP BY name
```

```
SELECT name
 FROM actor INNER JOIN movie BY actor.id = director
 HAVING gross < budget
```

```
SELECT name
  FROM actor INNER JOIN movie ON actor.id = director
 WHERE gross < budget
```

```
SELECT name
  FROM actor INNER JOIN movie ON actor.id:director
 WHERE gross < budget
```

```
SELECT name
  FROM director INNER JOIN movie ON movie.id = director.id
 WHERE gross < budget
```

2. Select the correct example of JOINing three tables

```
SELECT *
  FROM actor JOIN casting BY actor.id = actorid
 JOIN movie BY movie.id = movieid
```

```
SELECT *
  FROM actor JOIN casting ON actor.id = actorid
   AND JOIN movie ON movie.id = movieid
```

```
SELECT *
  FROM actor JOIN casting
  JOIN movie ON actor.id = actorid
   AND movie.id = movieid
```

```
SELECT *
  FROM actor JOIN casting ON actor.id = actorid
   AND movie ON movie.id = movieid
```

```
SELECT *
  FROM actor JOIN casting ON actor.id = actorid
  JOIN movie ON movie.id = movieid
```

3. Select the statement that shows the list of actors called 'John' by order of number of movies in which they acted

```
SELECT name, COUNT(movieid)
  FROM actor JOIN casting ON actorid=actor.id
 WHERE name IN 'John %'
 GROUP BY name ORDER BY 2
```

```
SELECT name, COUNT(movieid)
  FROM actor JOIN casting ON actorid=actor.id
 WHERE name LIKE 'J%'
 GROUP BY name ORDER BY 2 DESC
```

```
SELECT name, COUNT(movieid)
  FROM casting JOIN actor ON actorid=actor.id
 WHERE name LIKE 'John %'
 GROUP BY name ORDER BY 2 DESC
```

```
SELECT name, COUNT(movieid)
  FROM casting JOIN actor
 WHERE (actorid ON actor.id)
   AND name LIKE 'John %'
 GROUP BY name ORDER BY 2 DESC
```

```
SELECT name, COUNT(movieid)
  FROM casting JOIN actor
 WHERE name LIKE 'John %'
 GROUP BY name ORDER BY COUNT(movieid) DESC
```

4. Select the result that would be obtained from the following code:

```
SELECT title
  FROM movie JOIN casting ON (movieid=movie.id)
             JOIN actor   ON (actorid=actor.id)
 WHERE name='Paul Hogan' AND ord = 1
```

Table-A

"Crocodile" Dundee	1
Crocodile Dundee in Los Angeles	1
Flipper	1
Lightning Jack	1

Table-B

"Crocodile" Dundee
Crocodile Dundee in Los Angeles
Flipper
Lightning Jack

Table-C

"Crocodile" Dundee
Paul Hogan
1

Table-D

"Crocodile" Dundee	Paul Hogan	1
Crocodile Dundee in Los Angeles	Paul Hogan	1
Flipper	Paul Hogan	1
Lightning Jack	Paul Hogan	1

Table-E

"Crocodile" Dundee	Paul Hogan
Crocodile Dundee in Los Angeles	Paul Hogan
Flipper	Paul Hogan
Lightning Jack	Paul Hogan

5. Select the statement that lists all the actors that starred in movies directed by Ridley Scott who has id 351

```
SELECT name
  FROM movie JOIN casting
   AND actor ON movie.id = movieid
   AND actor.id = actorid
 WHERE ord = 1
  AND actor = 351
```

```
SELECT name
  FROM movie JOIN casting
  JOIN actor ON movie.id = movieid
    OR actor.id = actorid
 WHERE ord = 1 AND director = 351
```

```
SELECT name
  FROM movie JOIN casting ON movie.id = movieid
  JOIN actor ON actor.id = actorid
 WHERE ord = 1 AND actorid = 351
```

```
SELECT name
  FROM movie JOIN casting ON movie.id = movieid
  JOIN actor ON actor.id = actorid
WHERE ord = 1 AND director = 351
```

```
SELECT name
  FROM movie JOIN casting ON movie.id = actorid
  JOIN actor ON actor.id = movieid
 WHERE director = 351
```

6. There are two sensible ways to connect movie and actor. They are:

- link the director column in movies with the id column in actor
- join casting to itself

- link the actor column in movies with the primary key in actor
- connect the primary keys of movie and actor via the casting table

- link the director column in movies with the primary key in actor
- connect the primary keys of movie and actor via the casting table

- link the director column in movies with the primary key in actor
- connect the primary keys of movie and casting via the actor table

- link the movie column in actor with the director column in actor
- connect movie and actor via the casting table

7. Select the result that would be obtained from the following code:

```
SELECT title, yr
  FROM movie, casting, actor
 WHERE name='Robert De Niro' AND movieid=movie.id AND actorid=actor.id AND ord = 3
```

Table-A

A Bronx Tale	1993	3
Bang the Drum Slowly	1973	3
Limitless	2011	3

Table-B

A Bronx Tale	1993
Bang the Drum Slowly	1973
Limitless	2011

Table-C

A Bronx Tale	3
Bang the Drum Slowly	3
Limitless	3

Table-D

A Bronx Tale
Bang the Drum Slowly
Limitless

Table-E

A Bronx Tale	Robert De Niro	1993
Bang the Drum Slowly	Robert De Niro	1973
Limitless	Robert De Niro	2011

Score the test

Your score is: 7 out of 7

Category: Quizzes