

A **subquery** is a SQL query nested inside a larger query.

- A subquery may occur in :
  - A SELECT clause
  - A FROM clause
  - A WHERE clause
- The subquery can be nested inside a SELECT, INSERT, UPDATE, or DELETE statement or inside another subquery.
- A subquery is usually added within the WHERE Clause of another SQL SELECT statement.
- You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, or ALL.

--SQL query to find the actors who played a role in the movie 'Annie Hall'. Return all the fields of actor table.

```
select * from Actor where act_id in (select act_id from Movie_Cast  
where mov_id in (select mov_id from Movie where mov_title='Annie Hall'));
```

O/P:

|     |       |       |   |
|-----|-------|-------|---|
| 111 | Woody | Allen | M |
|-----|-------|-------|---|

--SQL query to find the director of a film that cast a role in 'Eyes Wide Shut'. Return director first name, last name

```
select dir_fname,dir_lname from director  
where dir_id in (select dir_id from movie_direction where mov_id in (select mov_id from Movie where  
mov_title= 'Eyes Wide Shut'));
```

O/P:

|         |         |
|---------|---------|
| Stanley | Kubrick |
|---------|---------|

--find for movies whose reviewer is unknown.Return movie title,year,release date, director first name, last name, actor first name, last name.

```
SELECT mov_title, mov_year, format(mov_dt_rel,'yyyy-MM-dd') as MovieReleaseDate, dir_fname,  
dir_lname,act_fname, act_lname  
FROM movie a, movie_direction b, director c, rating d, reviewer e, actor f, movie_cast g  
WHERE a.mov_id=b.mov_id and b.dir_id=c.dir_id  
and a.mov_id=d.mov_id and d.rev_id=e.rev_id  
and a.mov_id=g.mov_id and g.act_id=f.act_id and e.rev_name is null;
```

O/P:

|                   |      |            |        |          |          |       |
|-------------------|------|------------|--------|----------|----------|-------|
| Blade Runner      | 1982 | 1982-09-09 | Ridley | Scott    | Harrison | Ford  |
| Princess Mononoke | 1997 | 2001-10-19 | Hayao  | Miyazaki | Claire   | Danes |

--SQL query to find those movies directed by the director whose first name is Woody and last name is Allen. Return movie title.

```
SELECT mov_title from Movie  
where mov_id in (select mov_id from movie_direction where dir_id in  
(select dir_id from director where dir_fname='Woody' and dir_lname='Allen'))
```

O/P:

|            |
|------------|
| Annie Hall |
|------------|

--SQL query to determine those years in which there was at least one movie that received a rating of at least three stars. Sort the result-set in ascending order by movie year. Return movie year

```
select distinct mov_year from Movie
where mov_id in (select mov_id from rating where rev_stars > 3)
order by mov_year;
```

O/P:

|      |
|------|
| 1958 |
| 1961 |
| 1962 |
| 1977 |
| 1982 |
| 1986 |
| 1995 |
| 1997 |
| 1999 |
| 2001 |
| 2004 |
| 2008 |
| 2009 |

--SQL query to search for movies that do not have any ratings. Return movie title.

```
select distinct mov_title from movie
where mov_id IN ( select mov_id FROM movie where mov_id NOT IN (select mov_id FROM Rating));
```

O/P:

|                             |
|-----------------------------|
| Amadeus                     |
| Backto the Future           |
| Deliverance                 |
| Eyes Wide Shut              |
| Seven Samurai               |
| Spirited Away               |
| The Deer Hunter             |
| The Prestige                |
| The Shawshank<br>Redemption |

--SQL query to find those reviewers who have not given a rating to certain films. Return reviewer name (or Find the names of all reviewers who have ratings with a NULL value)

```
select rev_name from Reviewer
where rev_id in (Select rev_id from Rating where rev_stars is null )
```

O/P:

|              |
|--------------|
| Neal Wruck   |
| Scott LeBrun |

--SQL query to find movies that have been reviewed by a reviewer and received a rating. Sort the result-set in ascending order by

--reviewer name, movie title, review Stars. Return reviewer name, movie title, review Stars.

```
select mov_title, rev_name, rev_stars from Rating r ,Reviewer v,Movie m
where v.rev_id =r.rev_id and m.mov_id=r.mov_id
and v.rev_name is not null
```

And r.rev\_stars is not null

order by v.rev\_name,m.mov\_title,r.rev\_stars ;

O/P:

|                     |                    |     |
|---------------------|--------------------|-----|
| Aliens              | Brandt Sponseller  | 8.4 |
| Lawrence of Arabia  | Flagrant Baronessa | 8.3 |
| Donnie Darko        | Hannah Steele      | 8.1 |
| The Innocents       | Jack Malvern       | 7.9 |
| Good Will Hunting   | Josh Cates         | 4   |
| Braveheart          | Krug Stillo        | 7.7 |
| Annie Hall          | Mike Salvati       | 8.1 |
| Boogie Nights       | Paul Monks         | 3   |
| Beyond the Sea      | Richard Adams      | 6.7 |
| Titanic             | Righty Sock        | 7.7 |
| Vertigo             | Righty Sock        | 8.4 |
| American Beauty     | Sasha Goldshtein   | 7   |
| The Usual Suspects  | Simon Wright       | 8.6 |
| Avatar              | Victor Woeltjen    | 7.3 |
| Slumdog Millionaire | Vincent Cadena     | 8   |

--SQL query to find those movies, which have received highest number of stars. Group the result set on movie title and sorts the result-set in

--ascending order by movie title. Return movie title and maximum number of review stars.

select mov\_title , max(rev\_stars) as Max\_Rating from Movie m ,Rating r

where m.mov\_id=r.mov\_id and r.rev\_stars is not null

group by mov\_title

order by mov\_title;

O/P:

|                     |     |
|---------------------|-----|
| Aliens              | 8.4 |
| American Beauty     | 7   |
| Annie Hall          | 8.1 |
| Avatar              | 7.3 |
| Beyond the Sea      | 6.7 |
| Blade Runner        | 8.2 |
| Boogie Nights       | 3   |
| Braveheart          | 7.7 |
| Donnie Darko        | 8.1 |
| Good Will Hunting   | 4   |
| Lawrence of Arabia  | 8.3 |
| Princess Mononoke   | 8.4 |
| Slumdog Millionaire | 8   |
| The Innocents       | 7.9 |
| The Usual Suspects  | 8.6 |
| Titanic             | 7.7 |
| Vertigo             | 8.4 |

--SQL query to find all reviewers who rated the movie 'American Beauty'. Return reviewer name.

select rev\_name from Reviewer

where rev\_id in(select rev\_id from Rating where mov\_id in(select mov\_id from Movie where mov\_title ='American Beauty'));

--or

```
select rev_name from Reviewer v, Rating r, Movie m
where m.mov_id=r.mov_id and v.rev_id= r.rev_id
and m.mov_title ='American Beauty';
```

O/P:

|                  |
|------------------|
| Sasha Goldshtein |
|------------------|

--SQL query to find the movies that have not been reviewed by any reviewer body other than 'Paul Monks'. Return movie title.

```
select mov_title from Movie
where mov_id in (select mov_id from Rating where rev_id not in (select rev_id from Reviewer where rev_name
='Paul Monks'));
```

O/P:

|                     |
|---------------------|
| Vertigo             |
| The Innocents       |
| Lawrence of Arabia  |
| Blade Runner        |
| The Usual Suspects  |
| Chinatown           |
| Boogie Nights       |
| Annie Hall          |
| Princess Mononoke   |
| American Beauty     |
| Titanic             |
| Good Will Hunting   |
| Trainspotting       |
| Donnie Darko        |
| Slumdog Millionaire |
| Aliens              |
| Beyond the Sea      |
| Avatar              |
| Braveheart          |

--SQL query to find the movies with the lowest ratings. Return reviewer name, movie title, and number of stars for those movies.

```
select mov_title, rev_name, [rev_stars] from Rating r, Reviewer v, Movie m
where r.rev_stars in (select min(rev_stars) from Rating)
and m.mov_id=r.mov_id and r.rev_id=v.rev_id ;
```

O/P:

|               |            |   |
|---------------|------------|---|
| Boogie Nights | Paul Monks | 3 |
|---------------|------------|---|

--SQL query to find the movies directed by 'James Cameron'. Return movie title.

```
select mov_title from Movie where mov_id in (select mov_id from Movie_Direction
where dir_id in (select dir_id from Director where dir_fname='James' and dir_lname = 'Cameron'));
```

O/P:

|         |
|---------|
| Titanic |
| Aliens  |

--SQL to find the movies in which one or more actors appeared in more than one film.Return movie title and mov\_id.

```
select mov_id,mov_title from Movie
where mov_id in( select mov_id from Movie_Cast where act_id in (select act_id from Actor
                        where act_id in ( select act_id from Movie_Cast group by act_id having count(act_id)>1)));
```

O/P:

|     |                 |
|-----|-----------------|
| 914 | American Beauty |
| 923 | Beyond the Sea  |