**SELECT** c.contents, u.username **FROM** (**SELECT** \* **FROM** **comments** **WHERE** **comments**.photo\_id=3) **as** c **INNER** **JOIN** users **AS u** **ON** c.user\_id=u.id;

**SELECT** **COUNT**(c.\*) / **COUNT**(**DISTINCT** p.id) **AS** avg\_comments

**FROM** photos p **LEFT** **JOIN** **comments** c **ON** c.photo\_id = p.id;

**SELECT COUNT**(contents) **as** c\_count, photo\_id **FROM** **comments** **GROUP BY** photo\_id **ORDER BY COUNT**(contents) **DESC LIMIT** 1;

or

**SELECT** photos.id, **COUNT**(**comments**.id) **AS** number\_of\_comments

**FROM** photos **LEFT** **JOIN** **comments** **ON** photos.id = **comments**.photo\_id

**GROUP** **BY** photos.id **ORDER** **BY** number\_of\_comments **desc**

**limit** 1;

**SELECT** **COUNT**(\*) **AS** c\_count **FROM** **comments** **WHERE** photo\_id = 10

or

**SELECT** newtbl.newid, newtbl.number\_of\_comments **FROM** (**SELECT** photos.id **as** newid, **COUNT**(**comments**.id) **AS** number\_of\_comments

**FROM** photos **LEFT** **JOIN** **comments** **ON** photos.id = **comments**.photo\_id

**GROUP** **BY** photos.id) **as** newtbl **WHERE** newtbl.newid=10;

or

SELECT photos.id, COUNT(comments.id) AS comment\_count FROM photos LEFT JOIN comments ON photos.id = comments.photo\_id WHERE photos.id = 10 GROUP BY photos.id;

SELECT**comments**.user\_id,**COUNT**(**comments**.photo\_id)+**COUNT**(**comments**.id)**AS** number\_of\_activities **FROM** **comments GROUP** **BY** **comments**.user\_id **ORDER** **BY** number\_of\_activities **DESC LIMIT** 1;

**SELECT** **AVG**(**LENGTH**(contents)) **AS** avg\_comment\_chr

**FROM** **comments**;