This assignment was locked May 23 at 12:10pm.

For all problems, use the following schema:

Musicians(id, name, instrument, band\_id)

Bands(id, name, years\_together)

Shows(id, venue\_id, date)

Played\_in(band\_id, show\_id)

Venues(id, name, address)

Albums(id, name, year, band\_id, genre\_id)

Genres(id, name, description)

Songs(id, name, album\_id)

Assume that for each Entity set, the (id) field is the primary key, except in the case of Played\_in where the primary key is (band\_id, show\_id).

In addition, you can assume that Bands names and Venue names are unique (there are no duplicates) if that helps.

The relationships are as follows:

Musicians are members of bands (N:1) via (band\_id)

Bands play in shows (N:M) via (band\_id, show\_id)

Shows happen at venues (N:1) via (venue\_id)

Bands write albums (1:N) via (band\_id)

Albums fit into a genre (1:1) via (genre\_id)

Songs are on albums (N:1) via (album\_id)

--------------

QUESTIONS:

Write a single SQL query to answer each of the following questions:

1. Find the number of members, and band name, for each band that has played a show at the venue named "The Bluebird".

2. Find the names of bands who have played at venues named "The Hi Dive" and also at the venue named "Lions Lair".

3. Find the name of the earliest album (i.e. lowest year), and band name, for each band. (You can assume each band has only one earliest album, by year).

4. Find the number of members, and band name, for the band with the most members.

1.SELECT Count(m.id), b.name

FROM musician m, band b, played\_in p, show s, venues v

WHERE m.band\_id=b.id AND p.show\_id=s.id AND s.venue\_id=v.id AND v.name="Blurebird"

GROUP BY b.name;

2.SELECT b.name

FROM band b, venues v, played\_in p, show s

WHERE b.id=p.nad\_id AND p.show\_id=s.id AND s.venues\_id= v.id AND v.name="The Hi Dive" AND

b.name IN (SELECT b1.name FROM bands b1, venues v1, played p1, show s1 WHERE b1.id=p1.band\_id AND p1.show\_id=s1.id AND s1.venue\_id=v1.id AND v.name="Lions Lair");

3.SELECT a.name, b.name

FROM bands b, albums a

WHERE b.id = a.band\_id and

a.year=(SELECT MIN(a.year)

FROM Album a Bands b2

WHERE b.id=a.band\_id AND b2.id = b.id);

4.SELECT Count(m.id), b.name

FROM musician m, band b

WHERE m.band\_id=b.id

GROUP BY b.name

HAVING Count(m.id) >= ALL ( SELECT Count(m.id) FROM musician m, band b WHERE m.band\_id=b.id);

1. Find names of all bands that did not put an album out in the year 1999.

SELECT b.name

FROM bands b, albums a

WHERE NOT EXISTS (

SELECT \* FROM albums a2 WHERE a2.band\_id = b.id AND a2.year = 1999);

2. Find the names of all songs from the album with the highest ID.

SELECT s.name FROM songs s WHERE s.album\_id = (SELECT MAX(id) FROM albums);

3. Find the average number of songs per album for the band with ID 3.

SELECT AVG(c)

FROM (

SELECT COUNT(\*) as c FROM albums a, songs s WHERE a.band\_id = 3 AND s.album\_id = a.id GROUP BY a.id);

4. Report each band name, and first name of the musician in that band whose first name comes first alphabetically among members of the band (note: using the aggregate MIN() on a string field will give you the alphabetical minimum, i.e.: the first alphabetically).

SELECT b.name, MIN(m.first\_name) FROM musicians m, bands b WHERE m.band\_id = b.id GROUP BY b.name;

Answer：

1. Find the names of bands who have played together at least once.

SQL :

SELECT b.name

FROM Band b

WHERE b.id in (SELECT p.band\_id FROM Played\_in p WHERE p.show\_id in (SELECT p1.show\_id FROM Played\_in p1 GROUP BY show\_id HAVING count(band\_id)>=’1’))

2. Find the names of bands who have played together more than once.

SQL :

SELECT b.name

FROM Band b

WHERE b.id in (SELECT p.band\_id FROM Played\_in p WHERE p.show\_id in (SELECT p1.show\_id FROM Played\_in p1 GROUP BY p1.show\_id HAVING count(\*)>’1’))

3. Find the names of all songs that come from an album that fits into the genre "rock".

SQL:

SELECT s.name

FROM Song s

WHERE s.album\_id in (SELECT a.id FROM Album a WHERE a.genre\_id in (SELECT g.id FROM Genre g WHERE g.name=’rock’))

4. Find the ids of bands who have a member with the first name "John".

SQL :

SELECT m.band\_id

FROM Musician m

WHERE m.first\_name=’John’

5. Find the names of all albums by bands who have played at the venue "The Hi-Dive".

SQL:

SELECT a.name

FROM Album a

WHERE a.band\_id in (SELECT p.band\_id FROM Played\_in p WHERE p.show\_id in (SELECT s.id FROM Show s WHERE s.venue\_id in (SELECT v.id FROM Venue v WHERE v.name=’The Hi-Dive’)))

6. Find the names of all songs by bands who have played at "The Gothic" and at "The Hi-Dive".

SQL:

SELECT s.name

FROM Song s

WHERE s.album\_id in (SELECT a.id FROM Album a WHERE a.band\_id in (SELECT p.band\_id FROM Played\_in p WHERE p.show\_id in (SELECT s.id FROM Show s WHERE s.venue\_id in (SELECT v.id FROM Venue v WHERE v.name=’The Hi-Dive’ and? V.name= ‘ The Gothic’))))

7. Find the names of albums that fit in the genre "Reggae" by bands containing a member whose instrument is "Saxophone".

SQL:

SELECT a.name

FROM Album a

WHERE a.genre\_id in (SELECT g.id FROM Genre g WHERE g.name =’Reggae’) and a.band\_id in (SELECT m.band\_id FROM Musician m WHERE m.instrument=’Saxophone’ )

8. Find the name of bands who have been together 4 or more years, AND who have an album with more than 10 songs.

SQL:

SELECT b.name

FROM Band b

WHERE b.years\_together>=’4’ and b.id in (SELECT a.band\_id FROM Album a WHERE a.id in (SELECT s.album\_id FROM Song s GROUP BY album\_id HAVING count(\*)>’10’ ))

9. Find the names of the musicians in all bands who played at "The Gothic" on 01-02-2018.

SQL :

SELECT m.first name, m.last name

FROM Musician m

WHERE m.band\_id in (SELECT p.band\_id FROM Played\_in p WHERE p.show\_id in (SELECT s.id FROM Show s WHERE s.date=’01-02-2018’ and s.venue\_id in (SELECT v.id FROM Venue v WHERE v.name=’The Gothic’)))

10. Find the names of all the venues that have hosted shows with a band that put out an album that fits into the genre "Pop".

SQL:

SELECT v.name

FROM Venue v

WHERE v.id in (SELECT s.venue\_id FROM Show s WHERE s.id in (SELECT p.show\_id FROM Played\_in p WHERE p.band\_id in (SELECT a.band\_id FROM Album a WHERE a.genre\_id in (SELECT g.id FROM Genre g WHERE g.name=’Pop’))))

11. Find the descriptions of all genres for albums put out by the band "The Databases".

SQL:

SELECT g.description

FROM Genre g

WHERE g.id in (SELECT a.genre\_id FROM Album a WHERE a.band\_id in (SELECT b.id FROM Band b WHERE b.name =’The Databases’))

12. Find the names of all bands that have a member who plays "Guitar", OR have a member that plays "Keyboard".

SQL:

SELECT b.name

FROM Band b

WHERE b.id in (SELECT m.band\_id FROM Musician m WHERE m.instrument=’Guitar’ or m.instrument=’Keyboard’)

13. Find the id of all shows that includes a band who has been together more than 5 years.

SQL:

SELECT p.show\_id

FROM Played\_in p

WHERE p.band\_id in( SELECT b.id FROM Band b WHERE b.years\_together>’5’)

14. Find the instruments of all members of bands who have played at "The Bluebird".

SQL:

SELECT m.instrument

FROM Musician m

WHERE m.band\_id in (SELECT p.band\_id FROM Played\_in p WHERE p.show\_id in (SELECT s.id FROM Show s WHERE s.venue\_id in (SELECT v.id FROM Venue v WHERE v.name=’The Bluebird’)))

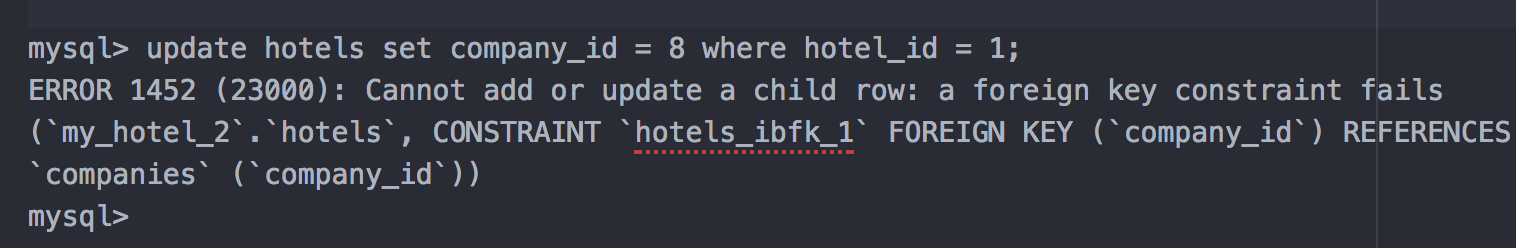
15. Find the names of all genres for albums put out by bands who played at "The Gothic".

SQL:

SELECT g.name

FROM Genre g

WHERE g.id in (SELECT a.genre\_id FROM Album a WHERE a.band\_id in (SELECT p.band\_id FROM Played\_id p WHERE p.show\_id in (SELECT s.id FROM Show s WHERE s.venue\_id in (SELECT v.id FROM Venue v WHERE v.name=’The Gothic’)) ))

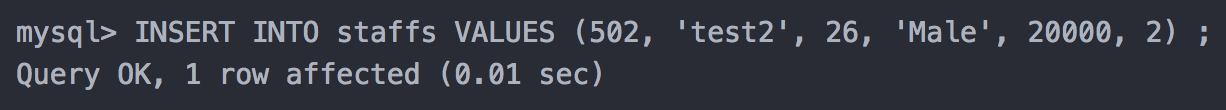


图片包含 文字

描述已自动生成

图片包含 文字

描述已自动生成



图片包含 瓶子

描述已自动生成

图片包含 物体

描述已自动生成

//I want to my staffs average salary group by hotels

mysql> select AVG(salary), hotel\_id from staffs group by hotel\_id;

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

| AVG(salary) | hotel\_id |

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

| 24798.1522 | 0 |

| 24181.1579 | 1 |

| 31120.1429 | 2 |

| 27187.6471 | 3 |

| 26711.0857 | 4 |

| 28288.7000 | 5 |

| 24366.3261 | 6 |

| 24281.0435 | 7 |

| 24202.2571 | 8 |

| 23226.4318 | 9 |

| 26771.9583 | 10 |

| 22258.6667 | 11 |

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

12 rows in set (0.00 sec)

//I want to my staffs average salary in my NO.3 hotel(id is 3)

mysql> select AVG(salary), hotel\_id from staffs group by hotel\_id having hotel\_id = 3;

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

| AVG(salary) | hotel\_id |

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

| 27187.6471 | 3 |

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

1 row in set (0.00 sec)