# EXPERIMENT 12

## LAB INTERNAL 1 – QUESTIONS

**AIM :** Using the below tables we need to retrieve data from tables for the the given Queries.

## GIVEN TABLES :

## QUESTION NUMBER : 1

1. customer(cust_id, name, income_permonth, gender, location_pincode) cust_id is pk, location_pincode is fk on city table

2. accounts( account_number, cust_id, account_type, balance_inlakhs, ifsc_code) account_number is pk , cust_id is fk on customers , ifsc_code is fk on branch table

3. branch (ifsc_code, branch_name, location_pincode) ifsc_code is pk

4. city( location_pincode, cityname, state) location_pincode is pk

## QUERIES :

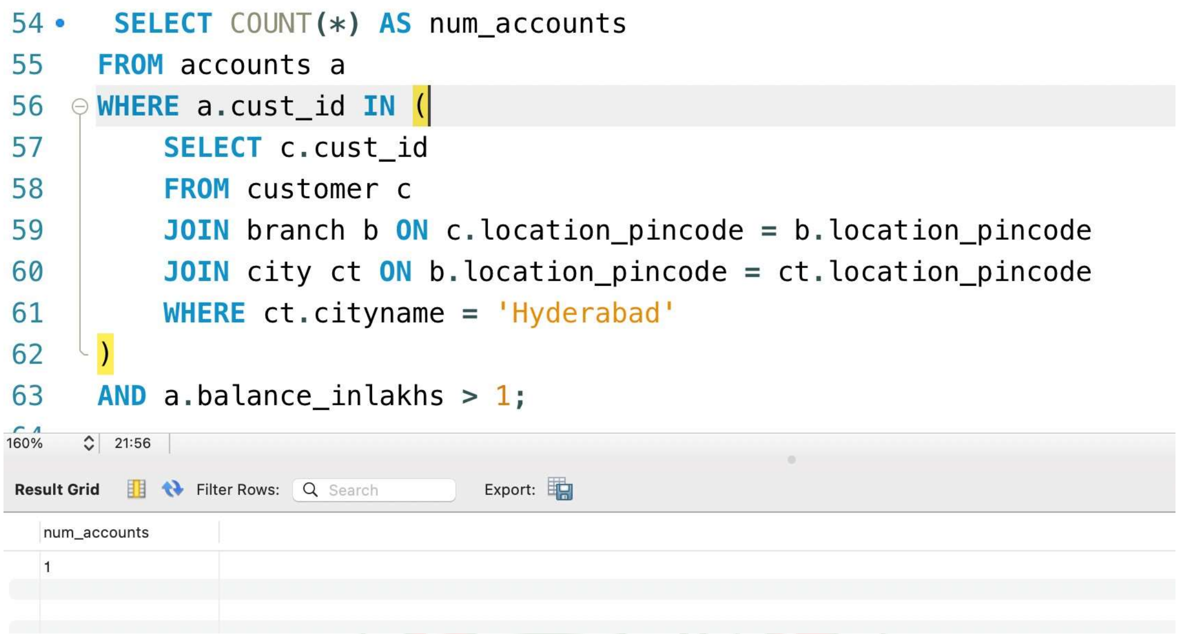
1. Find number of accounts in hyderabad city with balance amount > 1 lakh.
2. Find number of branches in vizag city with female accounts > 2.
3. Find customer names from any specified branch like Gandipet who are having more than one account. ( like savings, current , loan etc ).
4. Find city name which is having minimum average balance amount among all cities.

## CODE :

1. Find number of accounts in hyderabad city with balance amount > 1 lakh

```
SELECT COUNT(*) AS num_accounts FROM accounts a WHERE a.cust_id IN (

    SELECT c.cust_id FROM customer c

    JOIN branch b ON c.location_pincode = b.location_pincode

    JOIN city ct ON b.location_pincode = ct.location_pincode  WHERE ct.cityname = ‘Hyderabad')

    AND a.balance_inlakhs > 1;
```

## OUTPUT :



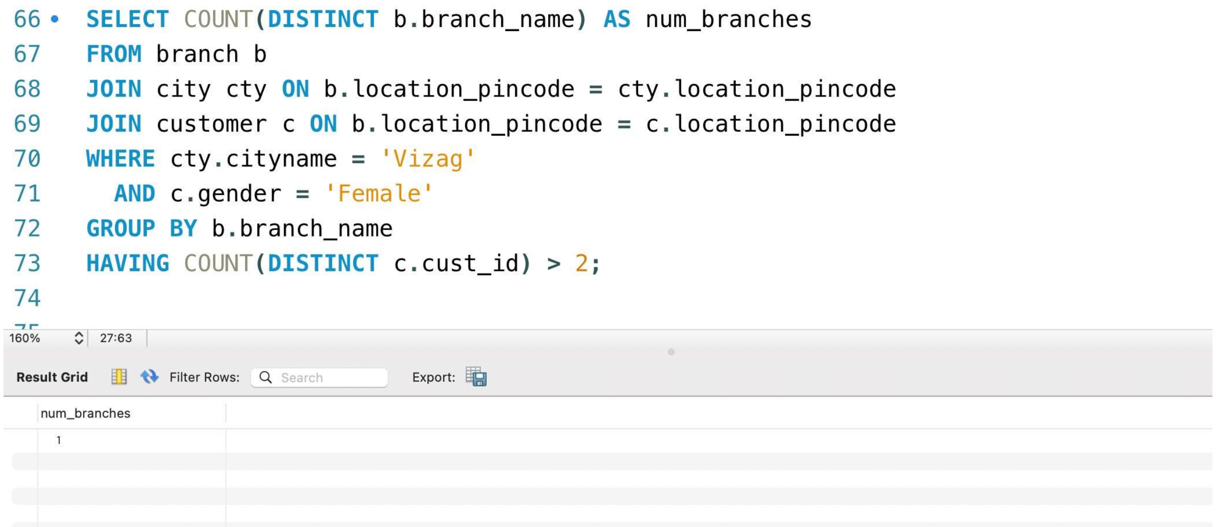
```
 SELECT COUNT(*) AS num_accounts
FROM accounts a
WHERE a.cust_id IN (
    SELECT c.cust_id
    FROM customer c
    JOIN branch b ON c.location_pincode = b.location_pincode
    JOIN city ct ON b.location_pincode = ct.location_pincode
    WHERE ct.cityname = 'Hyderabad'
)
AND a.balance_inlakhs > 1;
```

```
num_accounts
1
```

2. Find number of branches in vizag city with female accounts > 2

```
SELECT COUNT(DISTINCT b.branch_name) AS num_branches
FROM branch b
JOIN city cty ON b.location_pincode = cty.location_pincode
JOIN customer c ON b.location_pincode = c.location_pincode
WHERE cty.cityname = 'Vizag'
AND c.gender = 'Female'
GROUP BY b.branch_name
HAVING COUNT(DISTINCT c.cust_id) > 2;
```

## OUTPUT :



```
SELECT COUNT(DISTINCT b.branch_name) AS num_branches
FROM branch b
JOIN city cty ON b.location_pincode = cty.location_pincode
JOIN customer c ON b.location_pincode = c.location_pincode
WHERE cty.cityname = 'Vizag'
  AND c.gender = 'Female'
GROUP BY b.branch_name
HAVING COUNT(DISTINCT c.cust_id) > 2;
```

```
num_branches
1
```

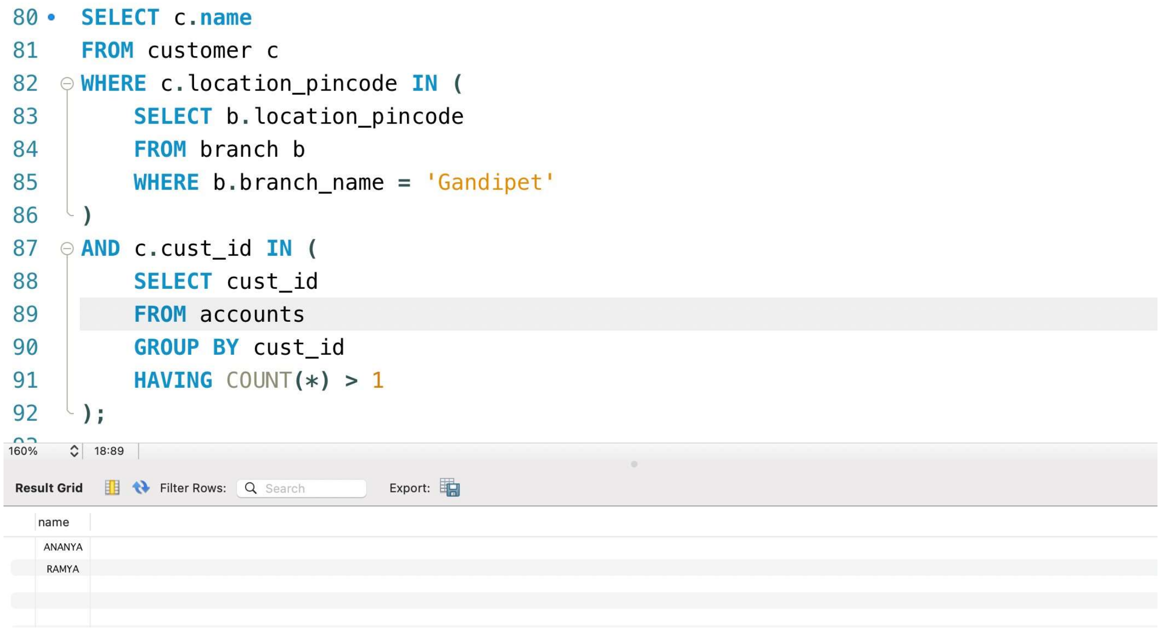
3. Find customer names from any specified branch like gandipet who are having more than one account. ( like savings, current , loan etc ).

SELECT c.name FROM customer c WHERE c.location_pincode

IN (SELECT b.location_pincode  FROM branch b WHERE b.branch_name = ‘Gandipet’) AND c.cust_id

IN (SELECT cust_id FROM accounts GROUP BY cust_id  HAVING COUNT(*) > 1);

## OUTPUT :



```
SELECT c.name
FROM customer c
WHERE c.location_pincode IN (
    SELECT b.location_pincode
    FROM branch b
    WHERE b.branch_name = 'Gandipet'
)
AND c.cust_id IN (
    SELECT cust_id
    FROM accounts
    GROUP BY cust_id
    HAVING COUNT(*) > 1
);
```

```
name
ANANYA
RAMYA
```

4. Find city name which is having minimum average balance amount among all cities.

SELECT cityname, avg_balanceFROM (SELECT ct.cityname, AVG(a.balance_inlakhs)

AS avg_balance  FROM accounts a  JOIN customer c ON a.cust_id = c.cust_id

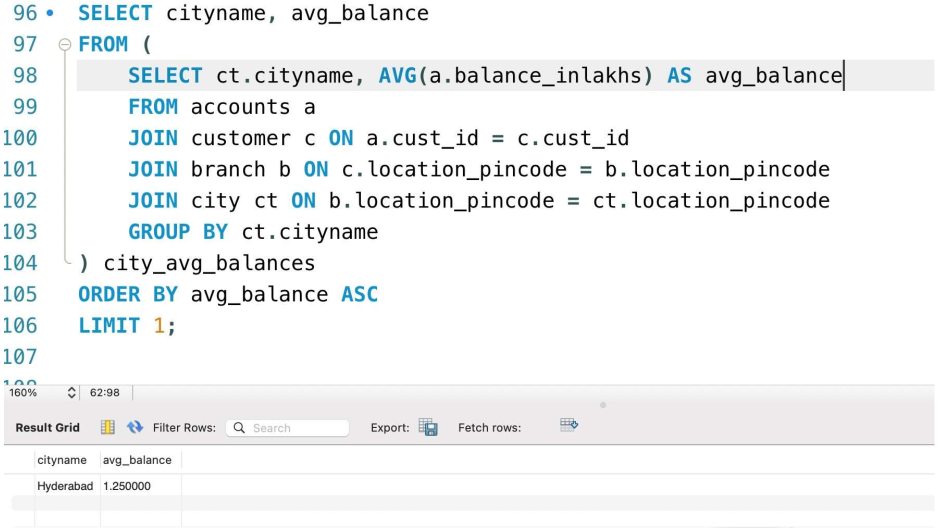
JOIN branch b ON c.location_pincode = b.location_pincode

JOIN city ct ON b.location_pincode = ct.location_pincode

GROUP BY ct.cityname) city_avg_balances

ORDER BY avg_balance ASC LIMIT 1;

## OUTPUT :



```
SELECT cityname, avg_balance
FROM (
    SELECT ct.cityname, AVG(a.balance_inlakhs) AS avg_balance
    FROM accounts a
    JOIN customer c ON a.cust_id = c.cust_id
    JOIN branch b ON c.location_pincode = b.location_pincode
    JOIN city ct ON b.location_pincode = ct.location_pincode
    GROUP BY ct.cityname
) city_avg_balances
ORDER BY avg_balance ASC
LIMIT 1;
```

```
cityname    avg_balance
Hyderabad   1.250000
```

## QUESTION NUMBER : 2

1. Emp ( emp_id, ename, salary, did, eid_of_manager) emp_id is pk and eid of manger is fk on emp_id of same table.

2. dept(did, dname) did is pk.

3. projects(pid, eid)pid is pk and eid is fk on emp.

## QUERIES :

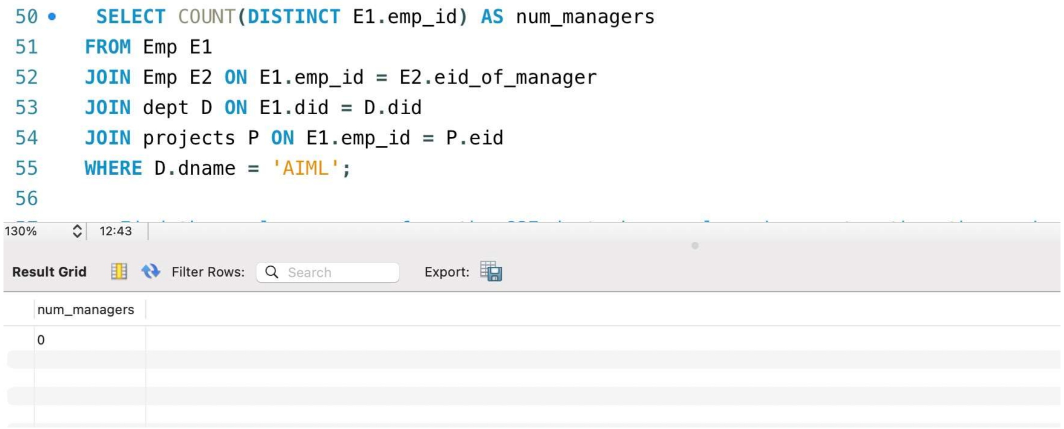
1. Find number of managers from AIML dept who are doing projects
2. Find the emp names from CSE dept whose sal > maximum avg sal of all depts.
3. Find the dept names from which none of the employees are doing atleast one projects
4. Find employee names who are managers to themselves
5. Find employee names who are not managers to any employees.

## CODE :

1.Find the number of managers from the AIML dept who are doing projects:

SELECT COUNT(DISTINCT E1.emp_id) AS num_managers
FROM Emp E1
JOIN Emp E2 ON E1.emp_id = E2.eid_of_manager
JOIN dept D ON E1.did = D.did
JOIN projects P ON E1.emp_id = P.eid
WHERE D.dname = ‘AIML';

## OUTPUT :



```
 SELECT COUNT(DISTINCT E1.emp_id) AS num_managers
FROM Emp E1
JOIN Emp E2 ON E1.emp_id = E2.eid_of_manager
JOIN dept D ON E1.did = D.did
JOIN projects P ON E1.emp_id = P.eid
WHERE D.dname = 'AIML';
```

| num_managers |
|---|
| 0 |

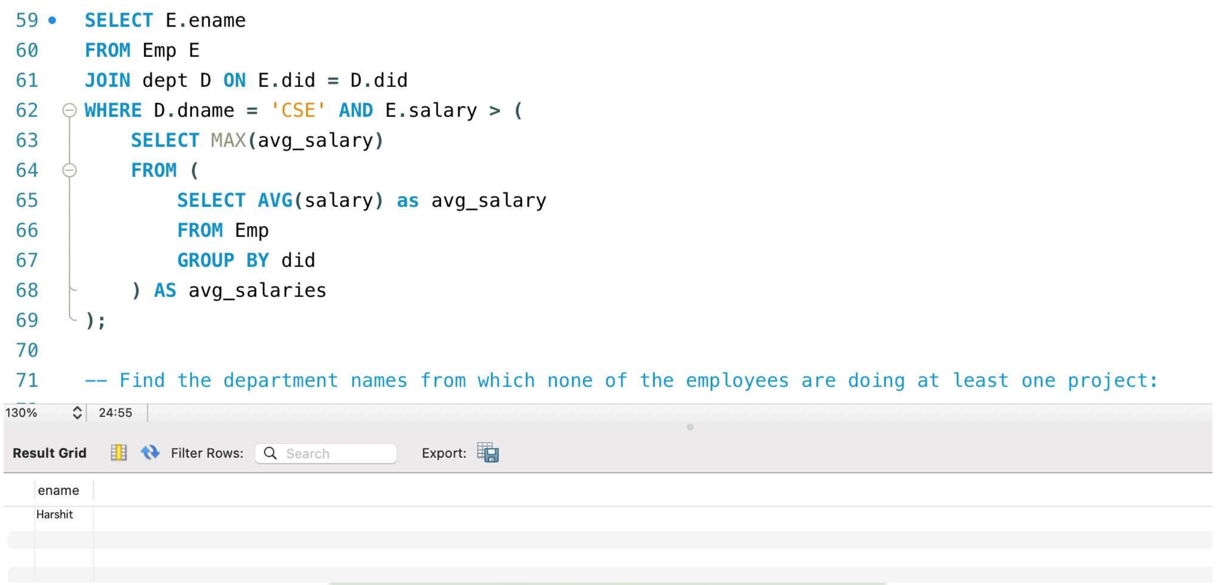
2. Find the employee names from the CSE dept whose salary is greater than the maximum average salary of all departments:

SELECT E.ename FROM Emp E JOIN dept D ON E.did = D.did

WHERE D.dname = 'CSE' AND E.salary > (SELECT MAX(avg_salary)

FROM (SELECT AVG(salary) as avg_salary FROM Emp GROUP BY did) AS avg_salaries);

## OUTPUT :



```
SELECT E.ename
FROM Emp E
JOIN dept D ON E.did = D.did
WHERE D.dname = 'CSE' AND E.salary > (
    SELECT MAX(avg_salary)
    FROM (
        SELECT AVG(salary) as avg_salary
        FROM Emp
        GROUP BY did
    ) AS avg_salaries
);

-- Find the department names from which none of the employees are doing at least one project:
```

| ename |
|---|
| Harshit |

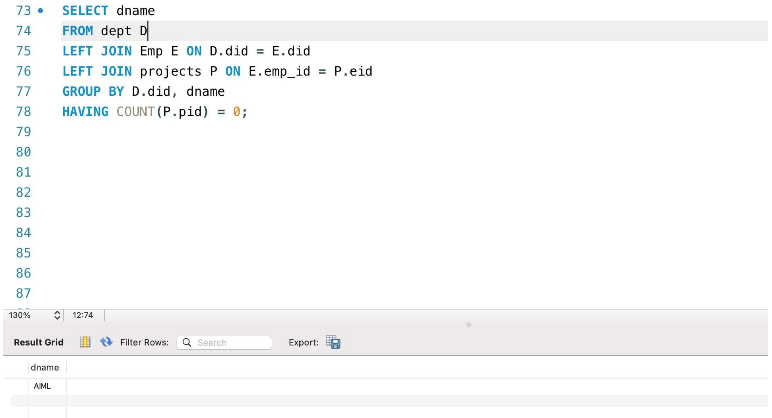
3. Find the dept names from which none of the employees are doing atleast one projects:

SELECT dname FROM dept D LEFT JOIN Emp E ON D.did = E.did

LEFT JOIN projects P ON E.emp_id = P.eid GROUP BY D.did, dname

HAVING COUNT(P.pid) = 0;

## OUTPUT :



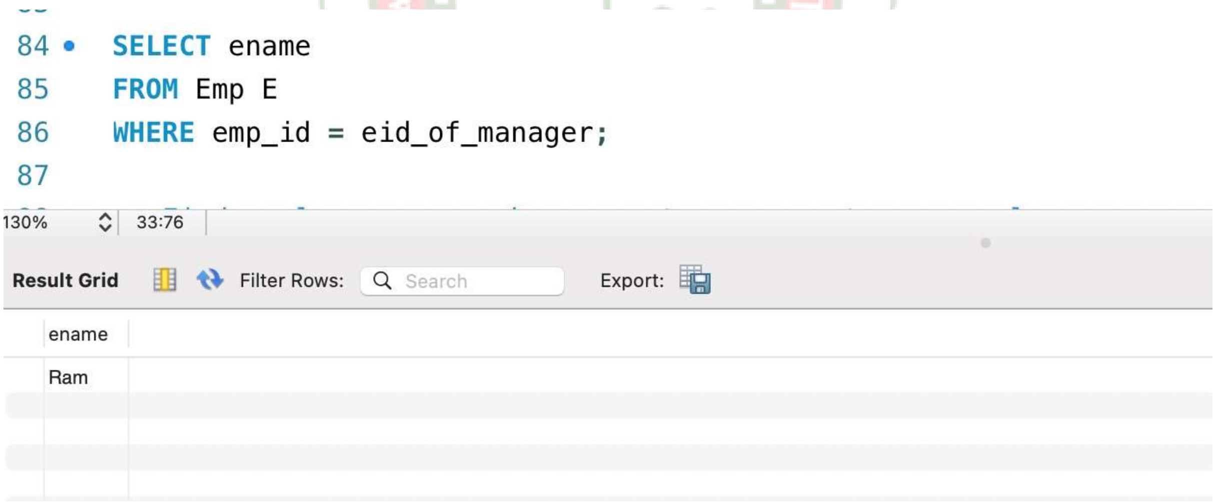
```
SELECT dname
FROM dept D
LEFT JOIN Emp E ON D.did = E.did
LEFT JOIN projects P ON E.emp_id = P.eid
GROUP BY D.did, dname
HAVING COUNT(P.pid) = 0;
```

```
dname
AIML
```

4. Find employee names who are managers to themselves.

Select Ename from Emp E where emp_id = eid_of _managers ;

## OUTPUT :



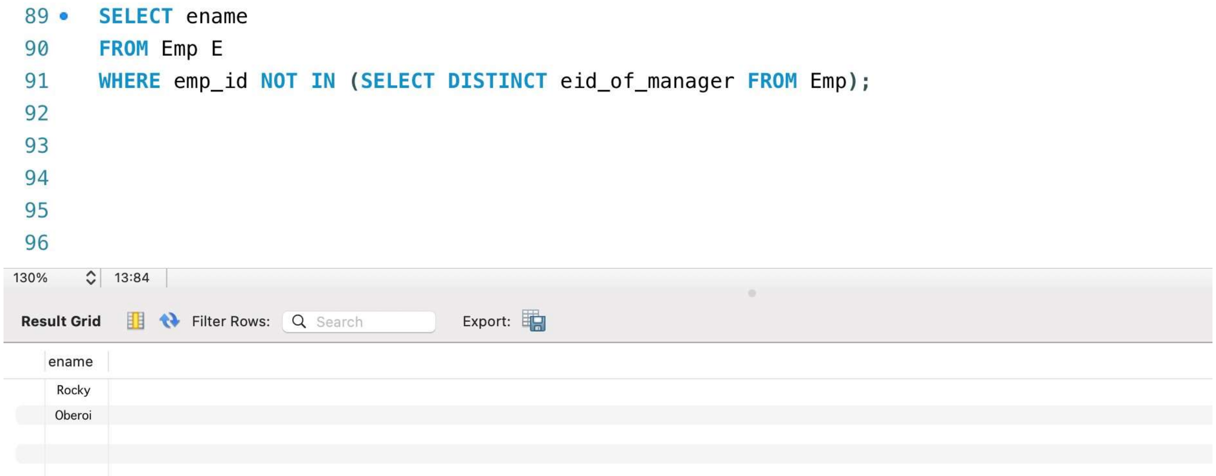
```
SELECT ename
FROM Emp E
WHERE emp_id = eid_of_manager;
```

```
ename
Ram
```

5. Find employee names who are not managers to any employees.

```
SELECT ename FROM Emp E
        WHERE emp_id NOT IN (SELECT DISTINCT eid_of_manager FROM Emp);
```

## OUTPUT :



```
SELECT ename
FROM Emp E
WHERE emp_id NOT IN (SELECT DISTINCT eid_of_manager FROM Emp);
```

```
ename
Rocky
Oberoi
```

## QUESTION NUMBER : 3

## TABLES :

1. artists ( artist_id, name, location_pincode) artist_id is pk, location pincode is fk on city table.
2. skills (skills_id, skill_name)skill id is pk.
3. artists_skills(artist_id, skill_id, exp_in_years )artist_id and skill_id combined pk ie composite pk and skill_id is fk on skills table.
4. movies(movie_name,artist_id, language, genre)movie_name and artist_id are composite pk , genres like action, horror etc.
5. city(pincode, cityname, state)pincode is pk.

## QUERIES :

1. Find the artists who are both directors and producers for action genre.
2. Find music directors from hyderabad city who have at least 5 years of experience as singers in telugu movies.
3. Find all artists names who commonly worked for kgf1, RRR and pushpa movies.

## CODE :

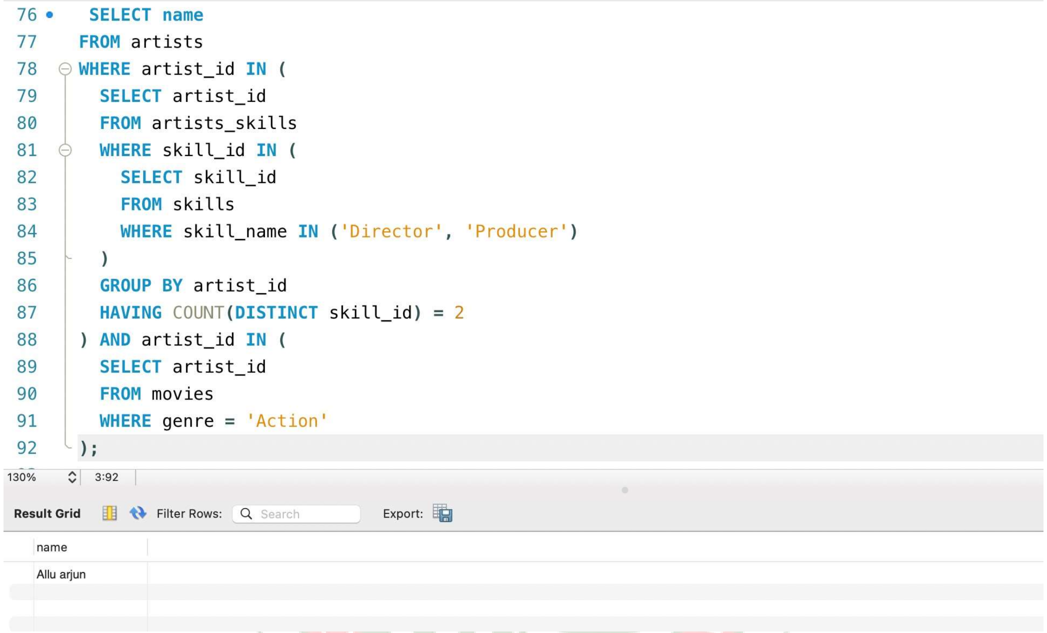
1. Find the artists who are both directors and producers for action genre.

```
SELECT name FROM artists WHERE artist_id IN (SELECT artist_id  FROM artists_skills

WHERE skill_id IN (SELECT skill_id FROM skills  WHERE skill_name IN
```

('Director', 'Producer') ) GROUP BY artist_id HAVING COUNT(DISTINCT skill_id) = 2)

AND artist_id IN ( SELECT artist_id FROM moviesWHERE genre = ‘Action');

## OUTPUT :



```
 SELECT name
FROM artists
WHERE artist_id IN (
  SELECT artist_id
  FROM artists_skills
  WHERE skill_id IN (
    SELECT skill_id
    FROM skills
    WHERE skill_name IN ('Director', 'Producer')
  )
  GROUP BY artist_id
  HAVING COUNT(DISTINCT skill_id) = 2
) AND artist_id IN (
  SELECT artist_id
  FROM movies
  WHERE genre = 'Action'
);
```

| name |
|---|
| Allu arjun |

2. Find music directors from Hyderabad city who have at least 5 years of experience as singers in telugu movies.

SELECT name FROM artists WHERE artist_id IN (SELECT artist_id

FROM artists_skills WHERE skill_id =

(SELECT skill_id FROM skills WHERE skill_name = 'Music Director'))

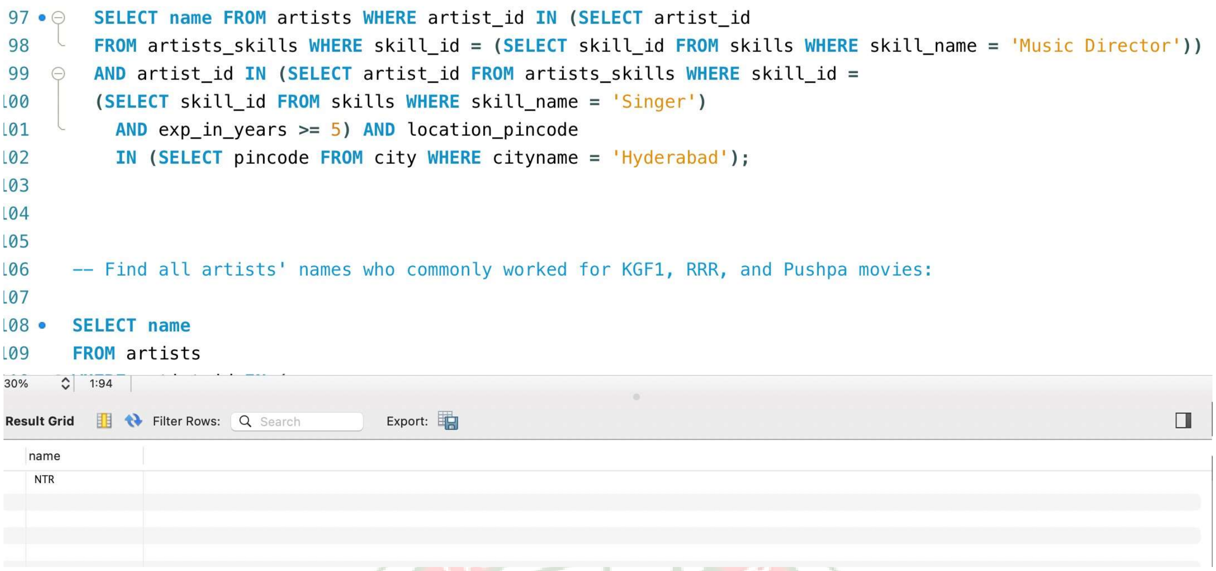
AND artist_id IN (SELECT artist_id FROM artists_skills WHERE skill_id =

(SELECT skill_id FROM skills WHERE skill_name = 'Singer')

AND exp_in_years >= 5) AND location_pincode

IN (SELECT pincode FROM city WHERE cityname = ‘Hyderabad');

## OUTPUT :



```
SELECT name FROM artists WHERE artist_id IN (SELECT artist_id
FROM artists_skills WHERE skill_id = (SELECT skill_id FROM skills WHERE skill_name = 'Music Director'))
AND artist_id IN (SELECT artist_id FROM artists_skills WHERE skill_id =
(SELECT skill_id FROM skills WHERE skill_name = 'Singer')
  AND exp_in_years >= 5) AND location_pincode
  IN (SELECT pincode FROM city WHERE cityname = 'Hyderabad');



-- Find all artists' names who commonly worked for KGF1, RRR, and Pushpa movies:

SELECT name
FROM artists
```

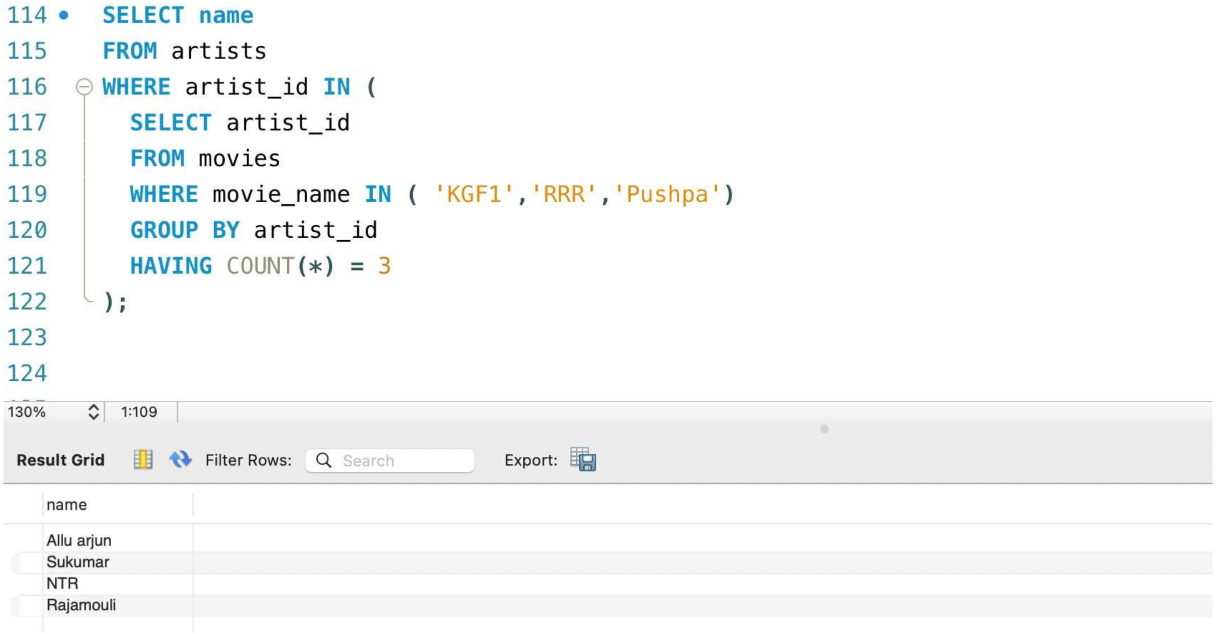
```
name
NTR
```

3. Find all artists names who commonly worked for kgf1, RRR and pushpa movies.

SELECT name FROM artists WHERE artist_id IN

(SELECT artist_id FROM movies WHERE movie_name IN ('KGF1', 'RRR', 'Pushpa') GROUP BY artist_id HAVING COUNT(DISTINCT movie_name) = 3);

## OUTPUT :



```
SELECT name
FROM artists
WHERE artist_id IN (
  SELECT artist_id
  FROM movies
  WHERE movie_name IN ( 'KGF1','RRR','Pushpa')
  GROUP BY artist_id
  HAVING COUNT(*) = 3
);
```

```
name
Allu arjun
Sukumar
NTR
Rajamouli
```

# QUESTION NUMBER : 4

## TABLES :

1. player(player_id, name, game_account_balance, location_pincode) player_id is pk , location_pincode is fk on city table.
2. matches(match_id, type_of_game , location_pincode).
3. transactions( trans_id, player_id, bet_amount, win_or_loss) Win_or_loss is Boolean column.
4. city(pincode, name) pincode is pk .

## QUERIES :

1.Find the player name who lost maximum amount in bets

2. Find city names with maximum average bet amount

3. Find the type of game which is having minimum number of bets

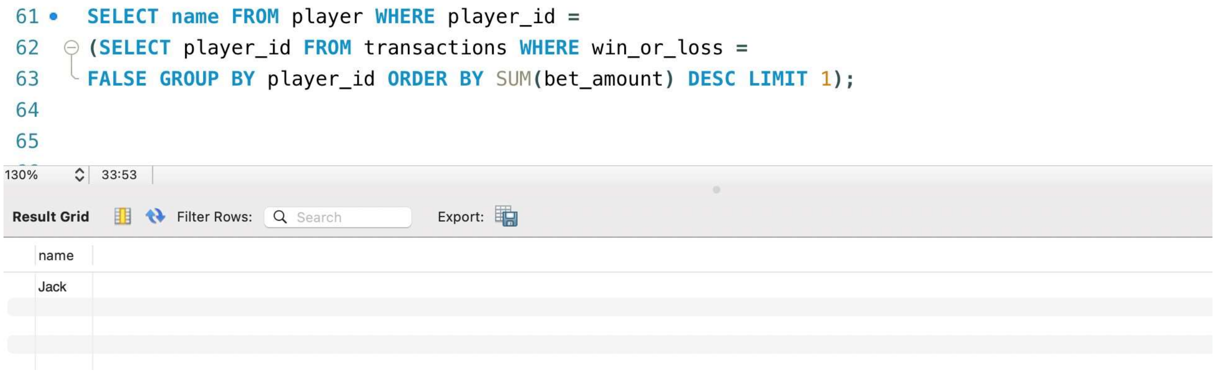
4. find city names from which no citizens bets done so far.

## CODE :

1. Find the player name who lost maximum amount in bets.

```
SELECT name FROM player WHERE player_id =

(SELECT player_id FROM transactions WHERE win_or_loss =

FALSE GROUP BY player_id ORDER BY SUM(bet_amount) DESC LIMIT 1);
```

## OUTPUT :



```
SELECT name FROM player WHERE player_id =
(SELECT player_id FROM transactions WHERE win_or_loss =
FALSE GROUP BY player_id ORDER BY SUM(bet_amount) DESC LIMIT 1);
```

```
name
Jack
```

2. Find city names with maximum average bet amount

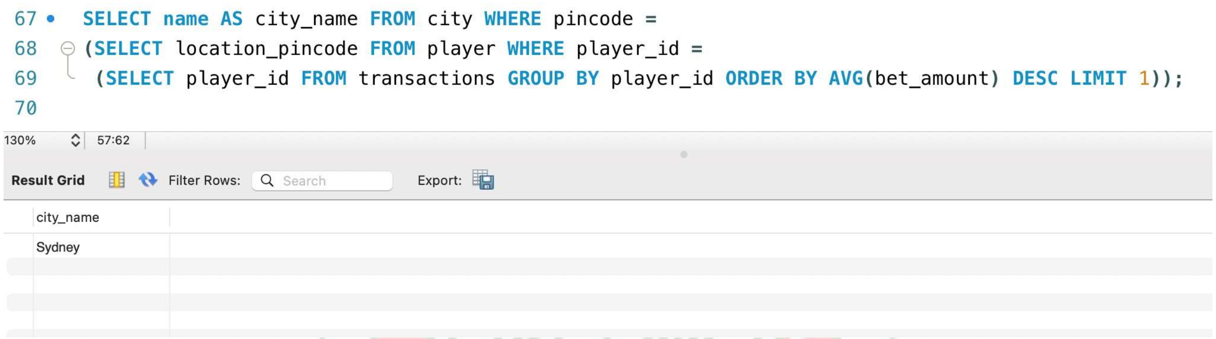
```
SELECT name AS city_name FROM city WHERE pincode =
```

(SELECT location_pincode FROM player WHERE player_id =

(SELECT player_id FROM transactions GROUP BY player_id ORDER BY

AVG(bet_amount) DESC LIMIT 1));

**OUTPUT :**



```
SELECT name AS city_name FROM city WHERE pincode =
(SELECT location_pincode FROM player WHERE player_id =
 (SELECT player_id FROM transactions GROUP BY player_id ORDER BY AVG(bet_amount) DESC LIMIT 1));
```

| city_name |
|---|
| Sydney |

3. Find the type of game which is having minimum number of bets.

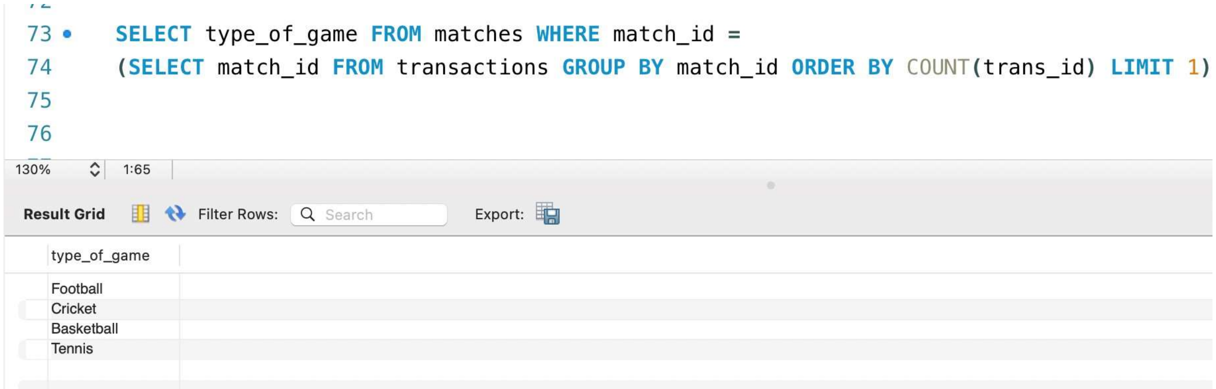
SELECT type_of_game

FROM matches

WHERE match_id = (SELECT match_id FROM transactions GROUP BY match_id

ORDER BY COUNT(trans_id) LIMIT 1);

**OUTPUT :**



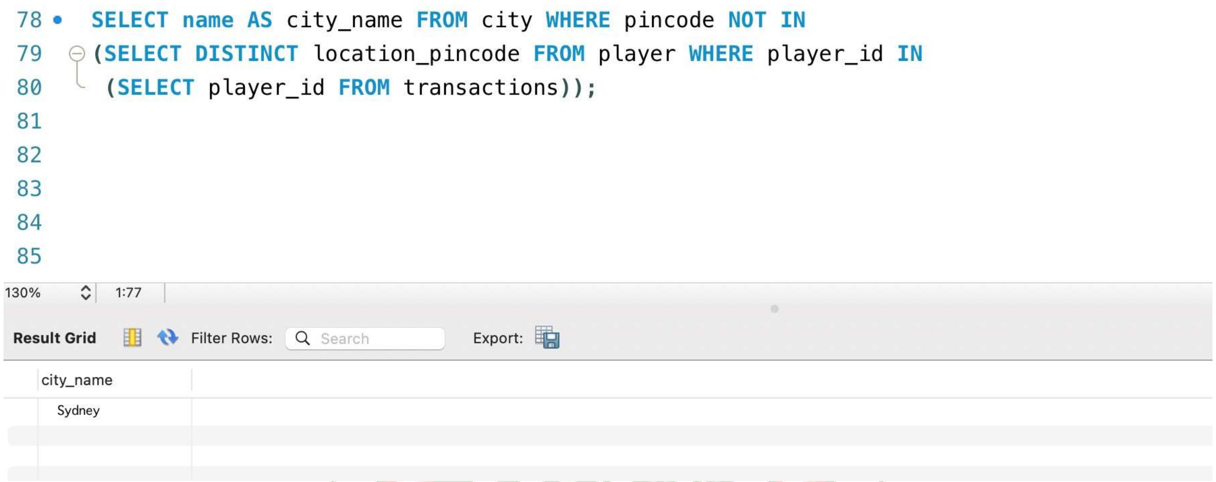
```
SELECT type_of_game FROM matches WHERE match_id =
(SELECT match_id FROM transactions GROUP BY match_id ORDER BY COUNT(trans_id) LIMIT 1)
```

| type_of_game |
|---|
| Football |
| Cricket |
| Basketball |
| Tennis |

4. Find city names from which no citizens bets done so far.

```
SELECT name AS city_name FROM city WHERE pincode NOT IN

        (SELECT DISTINCT location_pincode FROM player WHERE player_id IN


         (SELECT player_id FROM transactions));
```

## OUTPUT :



```
SELECT name AS city_name FROM city WHERE pincode NOT IN
(SELECT DISTINCT location_pincode FROM player WHERE player_id IN
 (SELECT player_id FROM transactions));
```

| city_name |
|---|
| Sydney |

**RESULT :** The above queries have been successfully executed by using various concepts such as aggregation functions, sub-queries.

**********