A cartoon of a coat of arms

AI-generated content may be incorrect.

TASK 1

Sophie Fidan 21068639

UFCFU3-15-3 Advanced Databases

|  |
| --- |
| FIRST NORMAL FORM |

|  |
| --- |
| SECOND NORMAL FORM |

|  |
| --- |
| THIRD NORMAL FORM |

|  |
| --- |
| ENTITY RELATIONSHIP DIAGRAM |

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| sql queries |

-- Display person's name and their age in years:

SELECT name,

TIMESTAMPDIFF(YEAR, dob, CURDATE()) AS age

FROM Person;

A screenshot of a black screen

AI-generated content may be incorrect.

-- Group Persons by their favourite drink and return average age of each group

SELECT f.value AS favorite\_drink,

ROUND(AVG(TIMESTAMPDIFF(YEAR, p.dob, CURDATE())), 2) AS average\_age

FROM Person p

JOIN Favorite f ON p.id = f.person\_id

WHERE f.type = 'Drink'

GROUP BY f.value;

A screenshot of a menu

AI-generated content may be incorrect.

-- Display average age of people who likes Hiking

SELECT f.value as activity, ROUND(AVG(TIMESTAMPDIFF(YEAR, dob, CURDATE())), 2) AS average\_age

FROM Person p

JOIN Favorite f ON p.id = f.person\_id

WHERE f.type = 'Activity' AND f.value = 'Hiking';

A screenshot of a computer

AI-generated content may be incorrect.

-- Display the total number of people from each City and sort it in ascending order by total number of people

SELECT a.city, COUNT(p.id) AS total\_number\_of\_people

FROM Person p

JOIN Address a ON p.id = a.person\_id

GROUP BY a.city

ORDER BY total\_number\_of\_people ASC;

A screenshot of a black screen

AI-generated content may be incorrect.

-- Display name of person(s) whose neighbour is neighbour C

SELECT p.name as person\_name, n.name as neighbour\_name

FROM Person p

JOIN PersonNeighbour pn ON p.id = pn.person\_id

JOIN Neighbour n ON pn.neighbour\_id = n.id

WHERE n.name = 'Neighbor C';

A screenshot of a computer

AI-generated content may be incorrect.