Walkthrough Sample Midterm

CS1106/CS6503: Intro to Relational Databases

Dr Kieran T. Herley

Semester One, 2023-24

School of Computer Science & Information Technology
University College Cork

Preliminaries

Reminder

When In-class Tuesday, 7 November 2023 at 1pm

Where Brookfield BHSC G.01

What

- Closed book/device/neighbour pen-and-paper test
- Duration 45 minutes ($13:05--13:50$)
- “Lab-like” SQL queries (12)

DB schema for sample test

```
persons(person_id, first_name, last_name, gender,
    birth_date, street, town, county)
likes(person_id, food)
knows(person_id, friend_id)
```

Problems and Solutions

Problem 1

Problem: List the names of all those whose first or last name contains the letter ’r’ .

Problem 1

Problem: List the names of all those whose first or last name contains the letter ’r’ .

```
SELECT first_name, last_name
FROM persons
WHERE first_name LIKE ”%r%” OR last_name LIKE ”%r%”;
```

Problem 2

Problem: List for each county the number of individuals who hail from that county. List the results alphabetically by county.

Problem 2

Problem: List for each county the number of individuals who hail from that county. List the results alphabetically by county.

```
SELECT county, COUNT(*)
FROM persons
GROUP BY county
ORDER BY county;
```

Problem 3

Problem: List the names of all individuals who like a food that contains the letter 'e' . Each name should appear at most once.

Problem 3

Problem: List the names of all individuals who like a food that contains the letter 'e' . Each name should appear at most once.

```
SELECT DISTINCT first_name, last_name
FROM persons join likes
    ON persons.person_id = likes . person_id
    WHERE food LIKE "%e%";
```

Problem 4

Problem: List the names of all those individuals who know Aoife Ahern.

Problem 4

Problem: List the names of all those individuals who know Aoife Ahern.

```
SELECT p2.first_name, p2.last_name
FROM persons AS p1 JOIN knows as k JOIN persons AS p2
ON p1.person_id = k.person_id AND k.friend_id = p2.person_id
WHERE p1.first_name = ”Aoife” AND p1.last_name = ”Ahern”;
```

Problem 5

Problem: List the names of all those individuals who do not know Aoife Ahern.

Problem 5 solution

```
SELECT first_name, last_name
FROM persons
EXCEPT
SELECT p2.first_name, p2.last_name
FROM persons AS p1 JOIN knows as k JOIN persons AS p2
ON p1.person_id = k.person_id AND k.friend_id = p2.person_id
WHERE p1.first_name = ”Aoife” AND p1.last_name = ”Ahern”;
```

Problem 6

Problem: List the names of all those individuals who know both Aoife Ahern and Declan Duffy.

Problem 6 solution

```
SELECT p2.first_name, p2.last_name
FROM persons AS p1 JOIN knows as k JOIN persons AS p2
ON p1.person_id = k.person_id AND k.friend_id = p2.person_id
WHERE p1.first_name = ”Aoife” AND p1.last_name = ”Ahern”
INTERSECT
SELECT p2.first_name, p2.last_name
FROM persons AS p1 JOIN knows as k JOIN persons AS p2
ON p1.person_id = k.person_id AND k.friend_id = p2.person_id
WHERE p1.first_name = ”Declan” AND p1.last_name = ”Duffy”;
```

Problem 7

Problem: List the names of the youngest and oldest individuals. In the event of a tie, list at least one of the youngest/oldest.

Problem 7 solution

```
SELECT first_name, last_name, birth_date
FROM persons
WHERE birth_date = (SELECT MIN(birth_date) FROM persons)
OR birth_date = (SELECT MAX(birth_date) FROM persons)
ORDER BY birth_date DESC;
```

Problem 8

Problem: List in descending order of popularity all the foods liked by people from Cork.

Problem 8 solution

```sql
SELECT food, COUNT(*)
FROM persons JOIN likes
ON persons.person_id = likes . person_id
WHERE county = ”Cork”
GROUP BY food
ORDER BY COUNT(*) DESC;
```

Problem 9

Problem: List all counties that have at least three people from there who all like the same food e.g. three people who like beer, or three who like pizza or whatever.

Problem 9 solution

```
SELECT DISTINCT persons.county
FROM persons JOIN likes
ON persons.person_id = likes . person_id
GROUP BY persons.county, likes.food
HAVING COUNT(*) >= 3;
```

Problem 10

Problem: List all pairs of individuals whose first names have the same first letter e.g. Anne and Amy. Note the substr() function may prove useful. List distinct pairs only.

Problem 10 solution

```
SELECT p1.first_name, p1.last_name, p2.first_name, p2.last_name
FROM persons AS p1 JOIN persons AS p2
ON substr(p1.first_name, 1, 1) = substr(p2.first_name, 1, 1)
WHERE p1.person_id < p2.person_id;
```