

\*\*Priyanka Guptal, Rakesh Kumar Singh2, Rambha Agrahari3\*\*

\*\*M.Tech 학생, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P., India1,2\*\*

\*\*부교수, CSE, Kamla Nehru Institute of Technology, Sultanpur, U.P., India2\*\*

## ## 요약

학제적 데이터베이스는 관계와 속성을 포함하고 있으며, 이러한 데이터베이스는 수천 개의 관계와 속성을 포함하고 있습니다. 따라서 사용자의 요구를 충족하고 실행 시점에 쿼리 양식을 동적으로 생성할 수 있는 시스템이 필요합니다. 본 논문에서는 대규모 및 다중 데이터베이스 문제를 해결하기 위한 동적 쿼리 양식(DQE)을 제안합니다. DQE는 사용자의 관심을 포착하고 쿼리 양식 구성 요소를 제공하여 사용자가 결정을 내릴 수 있도록 합니다. DQE는 반복적인 과정이며 각 반복은 사용자의 순위 목록을 자동으로 생성하고, 사용자는 필요에 따라 추가 구성 요소를 추가할 수 있습니다. 구조 요소의 순위는 사용자의 선호를 기반으로 설정됩니다. 이제 사용자는 풍부한 쿼리 양식을 작성하고 쿼리를 실행하여 결과를 볼 수 있습니다. 이 과정은 사용자에게 편리한 환경을 제공합니다.

\*\*키워드:\*\* 쿼리 양식, 쿼리 실행, 사용자 상호작용, 쿼리 양식 생성.

## ## I. 서론

데이터베이스는 쿼리 인터페이스가 제공되지 않는 한 유용하지 않습니다. 사용자가 데이터베이스와 소통하지 못한다면 그 데이터베이스는 무용지물입니다. 구조화된 쿼리는 SQL과 XQuery로 작성할 수 있지만, SQL에 대해 아무것도 모르는 사용자에게는 적합하지 않습니다. 이 때문에 쿼리 양식은 이러한 유형의 사용자에게 가장 쉬운 방법입니다. 그들은 단순히 양식의 빙크를 채우기만 하면 됩니다. 이것이 쿼리 양식이 데이터베이스 인터페이스 제공을 위한 가장 널리 사용되는 기술 중 하나인 이유입니다. 전통적인 쿼리 양식은 개발자나 DBA에 의해 사전 정의되어, 데이터베이스가 가지고 복잡해지면서 사용자의 임의 쿼리를 충족시키지 못합니다.

현재 많은 데이터베이스 개발 도구들이 존재합니다. 예를 들어, Easy Query [1], Cold Fusion [2], Microsoft Access와 같은 도구들은 사용자가 쿼리 양식을 사용자 맞춤형으로 만들 수 있는 메커니즘을 제공합니다. 그러나 사용자가 이러한 데이터베이스 도구에 익숙하지 않다면 대규모 데이터베이스를 다루는 것이 쉽지 않습니다. DQE는 쿼리 양식을 동적으로 생성하며, DQE의 각 반복은 쿼리 양식 풍부화와 쿼리 실행을 포함합니다. 기본 쿼리 양식이 사용자에게 제공되어, 이 양식은 사용자와 시스템의 상호작용을 통해 반복적으로 풍부화되며, 사용자가 쿼리 결과에 만족할 때까지 진행됩니다.

## ## II. 쿼리 양식 기술

## ## A. Query By Example:

이는 관계형 데이터베이스를 위한 데이터베이스 쿼리 언어입니다. 사용자가 명령, 예제 요소 및 조건을 입력할 수 있는 시각적 테이블을 생성하는 데 사용되는 첫 번째 쿼리 언어입니다.

예시 양식 연락처:

- 이름: X

- 주소:

- 도시:

- 주: Y

- 우편번호:

결과 SQL:

```sql

SELECT \* FROM CONTACTS

WHERE NAME='X' AND STATE='Y';

**B. 사용자 맞춤형 쿼리 양식:**

개발자가 쿼리 양식을 생성하거나 사용자 맞춤형으로 만들 수 있는 도구들이 많이 존재합니다. 예를 들어 SAP, Microsoft Access 등이 있습니다. 그러나 이러한 도구들은 데이터베이스에 익숙한 전문 개발자를 대상으로 하며, 데이터베이스에 대해 아무것도 모르는 최종 사용자를 대상으로 하지는 않습니다.

**C. 자동 정적 쿼리 양식:**

이는 데이터 기반 기술로, 먼저 주로 쿼리되는 데이터 속성을 찾습니다. 그런 다음 이러한 속성 기반으로 쿼리 양식을 생성합니다. 이는 과거 쿼리에 클러스터링 알고리즘을 적용하여 새로운 쿼리를 찾는 과정을 포함합니다. 그런 다음 이러한 새로운 쿼리 기반으로 쿼리 양식을 생성합니다. 이 기술의 문제는 데이터베이스가 매우 크고 복잡할 경우 사용자의 쿼리가 다양해질 수 있다는 점입니다. 이러한 경우, 많은 양식을 미리 생성하더라도 사용자의 요구를 충족시키지 못할 수 있습니다. 또한 많은 쿼리 양식을 생성하면 사용자가 자신의 요구를 충족시키는 양식을 어떻게 찾을지 모를 수 있습니다.

**III. 기존 시스템**

전통적인 쿼리 양식은 개발자나 DBA에 의해 사전 정의됩니다. 데이터베이스가 가지고 복잡해지면서 정적 쿼리 양식은 사용자의 요구를 충족시키지 못합니다.

이는 주로 쿼리되는 데이터 속성을 찾습니다. 그런 다음 이러한 속성 기반으로 쿼리 양식을 생성합니다. 이는 과거 쿼리에 클러스터링 알고리즘을 적용하여 새로운 쿼리를 찾는 과정을 포함합니다. 그런 다음 이러한 새로운 쿼리 기반으로 쿼리 양식을 생성합니다. 이 기술의 문제는 데이터베이스가 매우 크고 복잡할 경우 사용자의 쿼리가 다양해질 수 있다는 점입니다. 이러한 경우, 많은 양식을 미리 생성하더라도 사용자의 요구를 충족시키지 못할 수 있습니다. 또한 많은 쿼리 양식을 생성하면 사용자가 자신의 요구를 충족시키는 양식을 어떻게 찾을지 모를 수 있습니다.

전통적인 쿼리 양식은 개발자나 DBA에 의해 사전 정의됩니다. 데이터베이스가 가지고 복잡해지면서 정적 쿼리 양식은 사용자의 요구를 충족시키지 못합니다.