안녕하세요. 텀프로젝트 발표를 맡은 발표자 송은서입니다.

저희 팀은 유치원 유아 모집 등록 관리 시스템을 개발하고자 하였습니다.

(목차)

목차는 다음과 같습니다.

(1. 시스템 정의 및 개발 목적)

저희는 유치원 유아 모집과 등록을 데이터베이스를 통해 관리하는 시스템을 구축하고자 하였습니다. 이를 통해 기대할 수 있는 바는 다음과 같습니다. 첫째, 아이의 유치원 지원을 위해 학부모들에게 편리한 정보를 제공할 수 있다는 점, 둘째, 유치원 원아 선발 과정에 있어 효율성을 증대시킬 수 있다는 점, 마지막으로 유치원 정원 미달, 과열 등의 정보 제공을 통해 지원 과정 중 원아 몰림 현상을 방지할 수 있는 점입니다.

(1. 시스템 정의 및 개발 목적, 시스템 가정)

시스템 운영을 위한 가정은 다음과 같습니다. 화면을 참고해주시기 바랍니다.

(2. 사용자 요구분석)

저희는 시스템의 사용자가 학부모, 유치원, 시스템관리자 총 세개라고 판단하였습니다. 각 사용자의 요구사항에 따른 쿼리입니다. 학부모는 원하는 조건에 따른 유치원 목록, 지원하고자 하는 유치원의 경쟁률, 선발후 시기부터 등록기간 종료전까지 나의 현재 대기 순위와 관심 유치원 내에서의 결원 발생 사항, 또한 유치원의 입장에서는 현재 대기중인 학부모의 수, 해당 모집의 미달여부, 시스템 관리자의 입장에서는 유치원 모집 일정에 대해 요구할 것이라 생각하여, 그에 대응하는 쿼리를 작성하였습니다.

(3. ERD)

ERD는 다음과 같습니다. 학부모, 유치원, 시스템 관리자 총 세개의 entity가 존재하며 시스템관리자와 학부모는 1:N의 관계, 시스템관리자와 유치원은 1:N의 관계, 학부모와 유치원은 M:N의 관계를 띠고 있습니다. 학부모와 유치원은 M:N관계이므로, relationship이었던 지원은 associative entity가 되어 각각의 entity와 1:N의 관계를 이루게 됩니다.

(3 . ERD in MySQL)

MySQL 워크벤치에서의 ERD입니다.

(4. Relation schema)

각Entity의 attribute입니다. 학부모 엔티티에는 학부모 아이디 ~가 있습니다. 유치원 엔티티에는 ~가 있습니다. 시스템관리자는 ~~가 있습니다. 지원 엔티티에는 학부모 아이디 ~이 있습니다. 실선은 primary key, 점선은 entity간의 관계에 의한 외래키입니다.

(5. 로직 모델링)

다음은 앞서 살펴보았던 각 쿼리에 대한 수도코드입니다. 실제 데이터 베이스 구현에 대해서는 각 쿼리와 SQL상에서의 코드에서 자세히 다루도록 하겠습니다.

(6. database-학부모유치원)

시스템을 운영하기 위한 데이터 베이스에 대해 설명드리겠습니다. 좌측은 학부모, 우측은 유치원 테이블입니다. 각 table에 필요한 속성들을 column으로 넣었습니다.

(6. database-학부모유치원)

좌측은 지원 테이블입니다. 여기서 인롤 체크는 등록 여부로, 0과 1로만 표현되도록 데이터를 설정했습니다. 0일 경우 해당 학부모는 ‘등록 하지 않은상태이고,’, 1일 경우 해당학부모는 등록상태입니다. 우측은 시스템관리자 테이블로, 각 모집 기간에 대해 데이트 형식으로 날짜를 기입하였습니다.

(7. 시스템 운영 프로시저)

다음은 시스템을 운영하는 기본 프로시저입니다.

먼저 지원 프로시저가 있습니다. 학부모가 유치원에 지원할 경우, 해당 지원 정보가 지원 table에 기입되는 프로시저로, 실행시 우측의 결과와 같이 대기 순위가 변경됩니다. 만약 지원할 수 있는 3번의 기회를 다 사용한 학부모 ID를 넣었다면, 이미 3번을 지원했다는 string이 뜨도록 하였습니다. 이 경우 값이 입력되지 않습니다.

(대기순위 업데이트1)

다음은 등록 및 대기 순위 업데이트 프로시저입니다. 등록이 발생하면 등록 가능 여부를 확인한 후, 등록 시에는 대기 순위를 업데이트하는 프로시저입니다.

(대기순위 업데이트2)

가장 먼저 이미 다른 유치원에 등록했는지를 지원 테이블을 통해 확인하고, 등록한 경우 등록이 되었다는 string을 출력합니다. 만약 wait\_rank 변수가 0이하인 경우 등록 가능한 상태로 판단하여 이 때에만 등록 절차가 진행 가능합니다. 입력된 학부모 id와 유치원 id에 해당하는 지원 정보의 등록 여부를 1로 바꾸어 등록처리합니다.

(대기순위 업데이트3)

등록 절차를 진행할 때, 지원했으나 등록하지 않은 유치원들은 자동적으로 등록포기가 되도록 하였습니다. 이때에도 포기된 유치원의 등록 여부의 업데이트가 실행됩니다.

(대기순위 업데이트4)

현재 보시는 결과 창처럼, 등록 신청을 하면 각 유치원의 등록 가능여부와 대기순위를 알게 되고, 등록을 확정 지으면 등록 및 취소를 포함한 등록 여부의 업데이트가 되도록 하였습니다.

(등록인원 업데이트)

마지막으로 등록인원 업데이트 프로시저입니다. 실행시, 유치원 테이블의 등록학부모수를 업데이트하는 프로시저입니다. 유치원 테이블 내에 해당 유치원의 등록학부모수 cell의 숫자가 업데이트 됩니다.

앞서 언급한 프로시저들에서는 학부모들의 유치원에서 최초대기 순위와 현재 순위를 임의적으로 저희가 배치했다는 것을 한계로 볼 수 있습니다. 이를 보완하여 이후 시스템 개발 과정에서는 최초순위와 현재 순위를 sql 시스템 내에서 자동으로 배정하는 것 또한 생각해 볼 수 있습니다.

(8. 쿼리 수행 및 결과)

마지막으로 쿼리 수행을 위한 코드 작성 및 결과를 다루겠습니다.

첫째 쿼리는 학부모 A의 선호조건에 적합하는 유치원 목록은 무엇인가?입니다. 이를 수행하기 위해 설립유형, 돌봄유형, 통학차량유무등의 기준을 통해 유치원을 16가지로 분류하였고, 학부모가 원하는 조건에 따른 TYPE의 유치원 목록을 유치원 테이블에서 가져올 수 있도록 하였습니다.

다음 쿼리는 유치원 B의 정원대비 관심수의 비율은 얼마인가입니다. 유치원 테이블에서 관심수와 정원값을 사용해 경쟁률과 선호율을 계산하여 보여주도록 하였습니다.

다음 쿼리는 내가 ‘대기상태’인 유치원에서의 내 대기순위는 얼마인가?입니다. 앞서 언급한 프로시저를 사용해, 학부모 ID를 입력하면 등록을 하지 않은 학부모를 대상으로 지원한 유치원들의 최초, 현재 순위를 알 수 있습니다. 등록을 마친 학부모의 경우 이미 등록된 유치원이 존재한다는 결과 메시지가 출력 됩니다.

다음 쿼리는 유치원 B에서 결원이 발생했는가?입니다. 입력받은 유치원 이름의 공석을 확인하는 프로시저로, 실행시 정원과 현재 등록인원의 차이를 출력함으로써 공석의 개수를 알 수 있습니다.

다음 쿼리는 유치원 B에 대기 중인 학부모의 수는 무엇인가? 입니다. 최초 선발 이후 특정 유치원의 현재 대기 인원을 알려주며, 특정 유치원 ID를 입력 시, 유치원 테이블에서 해당 유치원의 현재 대기 순위 중 가장 높은 수를 대기중인 인원 결과 칼럼으로 조회할 수 있습니다.

다음 쿼리는 유치원 B의 정원보다 등록인원이 적은가? 입니다. 지원 테이블에서 등록여부의 합과 유치원 테이블에서의 정원값을 비교하여 사용자가 해당 유치원의 미달, 충족사항을 조회할 수 있도록 하였습니다 .

마지막 쿼리는 현재는 유치원 모집 일정의 어느 단게인가?입니다.

현재시간과 비교해 유치원 모집 일정의 단계를 알려줍니다. 모집일정은 2023년 5월을 가정으로 하였기에, 현재 실행하면 ‘유치원 지원기간이 아닙니다.’라는 결과가 뜨는 것을 알 수 있습니다.

이상으로 저희 팀의 발표를 마치겠습니다.