**로고학습일지**

**kt ds University 자바 기반의 데이터 사이언티스트 양성과정**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 08. 03 (금) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | 데이터베이스 | | | | |
| 작 성 자 | 유재헌 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | **01. 조인** |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| **01. 조인**  - 조인이란, 여러 테이블에 흩어져 있는 정보 중 사용자가 필요한 정보만 가져와서 테이블처럼 만들어서 결과를 보여주는 것으로 2개의 테이블을 조합하여 하나의 열로 표현하는 것이다.  - 조인의 종류  1) INNER JOIN(내부 조인)  - INNOR JOIN은 키 값이 있는 테이블의 컬럼 값을 비교 후 조건에 맞는 값을 가져오는 것이다.  - 간단히 말하면 서로 연관된 내용만 검색하는 조인 방법이다.    (1) 명시적 조인 표현  🡪 명시적 조인 표현에서는 테이블에 조인을 하라는 것을 지정하기 위해 ‘JOIN’ 키워드를 사용하고 ON의 키워드를 조인에 대한 구문을 지정하는데 사용한다.  (2) 암시적 조인 표현  🡪 암시적 조인 표현은 SELECT 구문의 FROM 절에서 콤마(,)를 사용하여 단순이 조인을 위한 여러 테이블을 나열하기만 하면 된다.  1-1) EQUI JOIN(동등 조인)  - EQUAL 연산자(=)를 사용하여 EQUI JOIN이라고 한다.  - WHERE 절에 기술되는 JOIN 조건을 검사해서 양쪽 테이블에 같은 조건의 값이 존재할 경우 해당 데이터를 가져오는 조인방법이다.  C:\Users\Main\AppData\Local\Microsoft\Windows\INetCache\Content.Word\암시적 조인.png    - 즉, INNER JOIN, EQUI JOIN, NATURAL JOIN 다 같은 말이다.  2) OUTER JOIN  - OUTER JOIN은 조인하는 여러 테이블에서 한 쪽에는 데이터가 있고, 다른 한쪽에는 데이터가 없는 경우, 데이터가 있는 쪽 테이블의 내용을 모두 출력하는 방법이다.  - 즉, 조인 조건에 만족하지 않아도 해당 행을 출력하고 싶을 때 사용할 수 있다.  - OUTER JOIN에는 LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN이 있다.  2-1) LEFT OUTER JOIN  🡪 LEFT OUTER JOIN은 조인문의 왼쪽에 있는 테이블의 모든 결과를 가져온 후 오른쪽 테이블의 데이터를 매칭하고, 매칭되는 데이터가 없는 경우 NULL을 표시한다.          2-2) RIGHT OUTER JOIN  🡪 RIGHT OUTER JOIN은 조인문의 오른쪽에 있는 테이블의 모든 결과를 가져온 후 왼쪽의 테이블의 데이터를 매칭하고, 매칭되는 데이터가 없는 경우 NULL을 표시한다.          2-3) FULL OUTER JOIN  🡪 FULL OUTER JOIN은 LEFT OUTER JOIN과 RIGHT OUTER JOIN을 합친 것이다.  🡪 양쪽 모두 조건이 일치 하지 않는 것들까지 모두 결합하여 출력한다.  3) SELF JOIN  - SELF JOIN은 테이블에서 자기자신을 조인시키는 것이다.  **02. 서브쿼리**  - 서브쿼리란, SELECT 문 내의 또 다른 SELECT문이라고 생각하면 된다.  - 서브쿼리 사용하는 이유 : 기본적인 SELECT문의 데이터 외에 데이터를 가공하여 사용할 필요가 있을 때 사용한다.  - 서브쿼리는 SELECT, INSERT, UPDATE, DELETE 어디에서나 사용이 가능하고 SELECT 리스트, FROM 절 WHERE절에서도 사용이 가능하다.  - 기존의 테이블의 컬럼과 비교 대상의 컬럼을 비교할 때 사용한다.  - 서브쿼리의 유형  1) 단일 행(Single-Row) 서브쿼리 : SELECT 문장으로부터 오직 하나의 행만을 검색하는 질의어다.  🡪 복수 행 연산자(IN, NOT IN, ANY, ALL, EXISTS)를 사용할 수 있다.  🡪 IN은 =와 OR가 결합한 형태이다.  🡪 NOT IN은 !=와 OR가 결합한 형태이다.  🡪 ANY는 비교 연산자와 OR  🡪 ALL은 비교 연산자와 AND  🡪 EXISTS는 데이터의 존재 유무를 파악하는데 사용한다.      🡪 서브 쿼리에 있는 EMPLOYEE\_ID가 108인 JOB\_ID를 검색한 FI\_MGR 결과를 바깥 쿼리문의 WHERE 절의 JOB\_ID를 비교해서 같은게 있으면 FIRST\_NAME, LAST\_NAME, JOB\_ID, EMPLOYEE\_ID를 출력하는 쿼리이다.  2) 다중 행(Multiple-Row) 서브쿼리 : 하나 이상의 행을 반환하는 서브쿼리를 다중 행 서브쿼리라고 한다.  🡪 FIRST\_NAME이 Steven인 사원과 같은 부서에서 일을 하는 모든 사원들의 정보를 조회  3) FROM 절상의 서브쿼리(INLINE-VIEW) : FROM 절 상에 오는 서브쿼리로 VIEW처럼 작용한다.  🡪 즉, SELECT의 FROM절에 테이블명이 오는 것이 아니라 SELECT 절이 온다.  🡪 사용하는 이유 : 서브쿼리를 포함해 테이블 조인이 이루어질 경우 검색하는 필드명은 테이블.필드명처럼 하려면 길어지게 된다.  🡪 서브쿼리는 메인 쿼리로 올라갈 수록 점점 더 길어져 불편할 수 있는데 이때, FROM절에서 SELECT문을 통해 추출한 필드명에 ALIAS를 줘서 간단하게 만들 수 있다. 이러한 목적 이외에도 예를들어, 전체 데이터에서 비교하는 것보다 ROWNUM을 이용해서 10번째 튜플 정보만 가져와 그 중에서 조건을 따져 데이터를 가져온다면 비교하는 횟수가 줄어들게 된다. 이때, ROWNUM을 통해 가져오는 객수를 제한 하는 방식을 N-TOP 질의라고 한다.  C:\Users\Main\AppData\Local\Microsoft\Windows\INetCache\Content.Word\INLINE.PNG  🡪 150번 사원보다 늦게 입사한 사원 중 150번 사원보다 더 많은 연봉을 받는 사원을 조회  🡪 평균 급여가 가장 많은 부서를 조회.  🡪 먼저, 부서별 평균 연봉을 구한 다음, 가장 높은 평균이 높은 연봉 순으로 정렬한다.  🡪 그리고 그 중 가장 높은 평균 연봉의 부서\_ID를 뽑는데 이때, ROWNUM을 사용한다.  🡪 ROWNUM = 1은 첫번째 ROW를 뽑아내는 역할을 한다.  **03. 계층형 쿼리**  - 계층형 쿼리는 오라클에서만 지원하고 있는 막강한 기능 중 하나다.  - 관계형이라는 의미가 서로 평등하고 수평적인 관계를 의미하는 반면, 계층형 구조는 이와는 다르게 평등한 관계가 아닌 계급적이고 수직적인 관계를 가진다.  - 계층구조 쿼리 문법  1) START WITH  🡪 계층 질의의 루트(부모행)로 사용될 행을 지정한다.  🡪 즉, 계층 구조가 어떤 행에서 시작하는지 지정하는 기능을 한다.  🡪 서브쿼리를 사용할 수도 있다.  2) CONNECT BY  🡪 이 절을 이용하여 계층 질의에서 상위계층(부모행)과 하위계층(자식행)의 관계를 규정할 수 있다.  🡪 PRIOR 연산자와 함께 사용하여 계층 구조로 표현할 수 있다.  🡪 CONNECT BY PRIOR 자식컬럼 = 부모컬럼 : 부모에서 자식으로 트리구성 (Top Down)  🡪 CONNECT BY PRIOR 부모컬럼 = 자식컬럼 : 자식에서 부모로 트리 구성 (Bottom Up)  🡪 CONNECT BY NOCYCLE PRIOR : NOCYCLE 파라미터를 이용하여 무한루프 방지  🡪 서브쿼리를 사용할 수 없다.  3) LEVEL  🡪 LEVEL은 계층구조 쿼리에서 수행결과의 DEPTH를 표현하는 의사컬럼이다.  - CONNECT BY의 실행 순서  1) START WITH 절  2) CONNECT BY 절  3) WHERE 절 순서로 풀리게 되어 있다.  - 계층적 쿼리 예제  🡪 100번의 사원을 상사로 두고 있는 모든 사원들의 모든 정보를 조회    - 계층적 쿼리의 결과    - 이런 식으로 해당 상사와 그 밑에 있는 사원들을 계층적 구조로 표현할 수 있다. |
|  |

|  |  |
| --- | --- |
| 특이사항 |  |