

4. INTERMEDIATE SQL

Index, Large Object Data Type, Authorization

BUILT-IN DATA TYPE (IN SQL)

SQL에서 추가적으로 지원하는 내장형 데이터 타입

- **DATE** : 연, 월, 일 순으로 구성된 날짜 저장

예) **DATE** '2020-02-02'

- **TIME** : 그 날의 시간을 시, 분, 초로 저장 (소수점 저장 ○)

예) **TIME** '09:00:30' **TIME** '09:00:30.75'

- **TIMESTAMP** : DATE + TIME

예) **TIMESTAMP** '2020-02-02 09:00:30.75'

- **INTERVAL** : 시간 간격을 저장하고 조작할 때 사용

예) **INTERVAL** '1' day // 하루의 간격을 나타냄

: 날짜에 대한 **interval** = 날짜 - 날짜 / 시간에 대한 **interval** = 시간 - 시간

: **date**, **time**, **timestamp** 값에 더해질 수 있음

- 1, 2, 3의 데이터 타입들은 비교, 산술 연산 가능

INDEX

- 인덱스로 검색 속도 높이기

예) *student* 릴레이션을 생성할 때, 검색 속도를 높이기 위해 인덱스 하나 만들기

```
CREATE TABLE student (  
    ID          VARCHAR (5),  
    name        VARCHAR (20) NOT NULL,  
    dept_name    VARCHAR (20),  
    tot_cred     NUMERIC (3, 0) DEFAULT 0,  
    PRIMARY KEY (ID) ) ;
```

```
CREATE INDEX studentID_index ON student (ID)  
// CREATE INDEX 인덱스 이름 ON 릴레이션 (속성)
```

- 인덱스 : 모든 튜플 살펴봄 x, 속성의 특정 값으로 레코드에 접근 → 속도높이는 데이터구조

INDEX - 예시

예) *student* 릴레이션에서 ID가 12345인 학생 찾기

```
SELECT *  
FROM student  
WHERE ID='12345'
```

앞 예시에서 *student* 릴레이션의 ID 속성값에 대한 INDEX가 만들어졌으므로, 쿼리 수행 시 모든 튜플을 살펴보지 않고 원하는 튜플만 인덱스를 사용해 찾아볼 수 있음

INDEX 장단점

- 장점 : 쿼리의 수행속도 빨라짐
- 단점 : 인덱스를 위한 추가적인 메모리 공간 필요
 - : 유지보수 비용
 - : 데이터 삽입, 갱신, 삭제 작업이 느려짐

LARGE-OBJECT TYPES

- **Large Objects** (대형 객체) : 대용량의 데이터를 저장하기 위한 데이터 타입
예) 사진, 비디오, CAD 파일 등
- **Large-Object Types** (**-lob**, large object는 대형객체를 의미함)
 1. **blob** (binary large object)
: 대용량의 해석되지 않은 이진 데이터 모음, 대용량의 이진 데이터 저장
(해석되지 않은 데이터들은 DB 내부에서 0, 1 형태로 있다가 외부 응용 프로그램에서 해석)
 2. **clob** (character large object)
: 대용량의 문자 데이터 모음, 대용량의 문서 데이터 저장
: 쿼리 결과가 large object를 반환할 때는 객체 자체가 아닌 포인터를 돌려줌

AUTHORIZATION (1)

- 사용자는 DB 데이터에 접근하여 SQL 문을 실행(삽입, 갱신 삭제)할 때 SQL을 실행할 수 있는 권한을 가지고 있어야 함

DB 일부에 대한 여러가지 권한

- Read : 데이터 읽기 O, 수정 X
- Insert : 데이터 삽입 O, 수정/삭제 X
- Update : 데이터 수정 O, 삭제 X
- Delete : 데이터 삭제 O

AUTHORIZATION (2)

- 권한이 없는 사용자가 동작을 수행하려고 하면 DB 시스템에 의해 거부됨
- 권한을 가지고 있는 사용자는 권한을 다른 사용자에게 주거나 자신이 주었던 권한을 취소할 수 있음 (최고권한 : DB 관리자)

DB 스키마에 대한 권한

- **Index** : 새로운 인덱스 생성, 삭제 ○
- **Resources** : 새로운 릴레이션 생성 ○
- **Alteration** : 릴레이션에 속성 추가, 삭제 ○
- **Drop** : 릴레이션 삭제 ○

GRANT – 권한 부여

- **GRANT** <부여할 권한> **ON** <권한을 사용할 릴레이션 or 뷰 이름> **TO** <사용자 ID>
- <사용자 ID>
 - : 사용자의 ID
 - : **public** = 현재 사용하고 있는 사용자와 앞으로 사용할 모든 사용자를 의미
(public에 부여된 권한은 모든 사용자에게 자동으로 부여됨)
- 뷰에 대한 권한 ≠ 뷰를 만들 때 사용한 기본 릴레이션에 대한 권한
- 권한을 부여하는 사용자 : 이미 특정 항목에 대해 권한을 갖고 있거나, DB 관리자

REVOKE – 권한 취소 (1)

- REVOKE <취소할 권한> ON <릴레이션 or 뷰 이름> FROM <사용자 ID>

예) 사용자 U1, U2, U3로부터 *branch* 릴레이션의 데이터를 읽고 검색할 권한을 취소하기

```
REVOKE SELECT ON branch FROM U1, U2, U3
```

- <취소할 권한>

: **ALL**을 넣으면 모든 권한 취소

예) **REVOKE ALL ON *branch* FROM U1, U2, U3** // *branch* 릴레이션에 대한 모든 권한 취소

: **PUBLIC** 넣으면 명시적으로 부여된 권한 제외, 모든 사용자가 권한 잃음

예) **GRANT SELECT ON *branch* TO U1** // 사용자 U1에게 명시적으로 SELECT 권한 부여

예) **REVOKE SELECT ON *branch* FROM PUBLIC** // 사용자 U1 제외, 모든 사용자 권한 취소

REVOKE – 권한 취소 (2)

- 동일한 사용자가 여러 사용자로부터 권한을 여러 번 부여 받은 경우, 어떤 한 권한을 뺏는다고 해도, 그 권한은 유지될 수 있음
- 어떤 권한이 취소되면, 그 취소된 권한에 의존해서 만들어진 모든 권한들 취소
- 권한을 넘겨준 사용자가 권한을 잃으면, 권한을 넘겨받은 사용자도 해당 권한 잃음

SQL 표준에서의 권한

- **SELECT** : 릴레이션을 읽거나 뷰를 사용하여 질의할 수 있는 권한

예) 사용자 U1, U2, U3에게 *instructor* 릴레이션에 대한 **SELECT** 권한 주기

```
GRANT SELECT ON instructor TO U1, U2, U3
```

- **INSERT** : 릴레이션에 튜플을 삽입할 수 있는 권한 (릴레이션의 모든 속성 or 일부 가능)
- **UPDATE** : 릴레이션의 튜플을 수정할 수 있는 권한 (모든 속성 or 일부 가능)

예) *department* 릴레이션의 *budget* 속성을 수정할 권한 주기

```
GRANT UPDATE (budget) ON department TO 사용자ID
```

- **DELETE** : 릴레이션의 튜플을 삭제할 수 있는 권한
- **ALL PRIVILEGES** : 사용되는 모든 권한의 축약 형태

(해당 릴레이션을 만든 사람이 자동적으로 갖는 권한)

ROLE

- **ROLE** : 여러 사용자에게 대한 권한 관리를 단순화 하기 위해 사용
- 어떤 사용자가 해당 롤이라고 인식되면 개별적인 권한을 사용자에게 명시할 필요 없이 롤에 부여된 모든 권한이 해당 사용자에게 부여됨

- 롤 만들기 : **CREATE ROLE** *롤의 이름*
- 롤을 사용자에게 부여 : **GRANT ROLE** *롤의 이름* **TO** *사용자ID*
- 롤에 권한 부여 : **GRANT** 권한 **ON** *릴레이션* **TO** 롤

ROLE 체인

- 롤의 체인

```
CREATE ROLE 롤1 ;  
CREATE ROLE 롤2 ;  
CREATE ROLE 롤3 ;  
GRANT 롤1 TO 롤2 ;  
GRANT 롤2 TO 롤3 ;  
GRANT 롤3 TO 사용자 ;
```

// 사용자는 롤3의 권한, 롤2의 권한, 롤2가 상속받은 롤1의 권한 모두를 가짐