16. SQL 中的安全与用户认证 与 SQL递归

16.1 SQL 中的安全与用户认证

安全 包含两方面：

- 用户只能看见他们能看见的数据；
- 对恶意用户的防卫。

SQL 安全通过授权 ID (Authorization ID) 与权限进行管理。

- **授权 ID**：SQL 环境中的元素。一个或一组用户可能会被允许拥有某对象的特定权限，由 `UserID` 与 `Role` 表示：
  - `User ID`：个人帐户名称；
  - `Role`：一个特定的权限集合，通过 `CREATE ROLE` 创建。

  `PUBLIC` 是一个特殊的内置授权 ID。
- **会话中的授权**：会话与 `User ID` 或角色名 `Role name` 相联系，提供执行 SQL 指令的授权 ID。
  - 通过连接时显式提供用户名（或者通过具体实现时定义的方法提供）。

    ```
    CONNECT TO ... USER usr;
    ```

  - 在嵌入式 SQL 中，授权 ID 可能会更改，可以通过 `SET ...` 进行设置，如

    ```
    SET SESSION AUTHORIZATION ...
    SET ROLE ...
    ```

    - `SESSION_USER` 表示 SQL 会话中的用户 ID；
    - `CURRENT_USER` 表示当前的用户 ID；
    - `CURRENT_ROLE` 表示当前的角色名称。
  - **权限**：与授权 ID 联系。共有如下 9 种类型：`SELECT`，`INSERT`，`DELETE`，`UPDATE`，`REFERENCES`，`USAGE`，`TRIGGER`，`EXECUTE`，`UNDER`。

	select	insert	update	delete	reference	trigger	usage	exec	under
Base table	√	√	√	√	√	√			
View	√	√	√	√	√	√			
Column	√	√	√	√	√				
Domain							√		
UDT							√		√
Character set							√		
Collation							√		
SQL-invoked routine								√	
Method of UDT	√								
Trigger	Creator must have the privileges needed by the condition and actions, but the user whose activity awakens the trigger does not need those privileges.								

获取权限：仅有以下两类用户：权限所有者 (owner) 与被授权者 (granted user) 拥有对一个 SQL 元素的权限。

- owner 有关于该 SQL 元素的所有权限；
- owner 可以使用 `GRANT` 向其他用户授权。

所有权的建立：在 `CREATE` 时加入 `AUTHORIZATION usr` 标识符表示该创建的元素的所有者为 `usr`。

```
CREATE SCHEMA ... AUTHORIZATION usr;    // 模式
CONNECT TO svr AS conn AUTHORIZATION usr;   // 会话
MODULE modname ... AUTHORIZATION usr;   // 模块
```

- `usr` 是 owner；
- 如果没有指定模式的 `AUTHORIZATION usr`，那么 owner 默认为模块的 owner；
- 如果没有指定模块的 `AUTHORIZATION usr`，那么 owner 默认为会话的 owner。

当前授权 ID (current authorization ID)

- 当前授权ID为所执行的模块的授权 ID，如果模块没有授权 ID，则为会话授权 ID。
- 只有当前授权 ID 有执行相关命令的权限，才会执行这些命令。

权限原则

- 如果数据所有者为 `U` 并且 `U` 是当前授权 ID，那么可以有该模块的全部权限。
- 如果数据所有者为 `O`，`U` 为当前授权 ID，且 `O` 向 `U` 授权，那么可以有该模块的特定权限（具体授权的权限）。

授权：某项权限的拥有者（包括 owner 与 granted user）可通过如下指令进行授权。

```
GRANT privileges on DB-element TO users [WITH GRANT OPTION]
```

- `priviledges` 包括 `SELECT, INSERT, SELECT(A) ...`，可以使用 `ALL PRIVILEGE` 表示所拥

有的全部权限。

- `DB-element` 通常是关系（基本表或视图）；对于其他类型的元素，使用 `type name` 来声明，如：`ASSERTION myAssertion`。
- `WITH GRANT OPTION`：带有这个标识符表示被授权者**可以继续向他人授权**，但是继续授权时，授权的权限只能小于等于当前权限。

授权图：用来追踪被授权的用户。

- 节点使用“用户/权限”表示。
  - 使用 `*` 表示 `WITH GRANT OPTION`；
  - 使用 `**` 表示 owner。
- 同一个用户，带有 `WITH GRANT OPTION` 的节点与不带有 `WITH GRANT OPTION` 的节点（如果都存在）用两个不同节点表示。
- 同一个用户，拥有两种类型的权限 p, q（可以相互包含）也需要用两个不同节点表示。
  - 尽管高级别的权限被取消，低级别的权限仍可能保留。

召回权限：授权的权限可在任意时刻召回。

```
REVOKE privileges ON DB-element FROM users [CASCADE | RESTRICT];
```

- `CASCADE`：召回当前权限，并召回任何只由于当前权限授权得到的权限。
  - 删除对应授权图的边；删除当前授权点；
  - 任何从 owner 无法到达的点也被删除。
- `RESTRICT`：如果当前召回操作会导致其他权限的召回，则不能执行当前召回操作。

召回 `GRANT OPTION`：授权的 `GRANT OPTION` 可以进行召回；仅仅召回 `GRANT OPTION`，并不召回权限本身。

```
REVOKE GRANT OPTION FOR privilege ON DB-element FROM users [CASCADE |
RESTRICT];
```

在授权图中新增无 `GRANT OPTION` 的节点与边，并删除授权者至带 `GRANT OPTION` 的被授权者的边。

`CASCADE` 与 `RESTRICT` 与召回权限时说明相同。

角色 (role)：

```
CREATE ROLE rolename [WITH ADMIN {CURRENT_USER | CURRENT_ROLE}];
DROP ROLE rolename;
```

16.2 SQL 中的递归

递归规则：一开始假设所有 IDB 关系都为空，每次通过前一轮的 IDB 与当前的 EDB 关系得到新一轮的 IDB 结果，直到 IDB 不发生变化。

`WITH` 语句：定义暂时的关系

```
WITH R AS definition_of_R
  query involving R;
```

递归

```
WITH
  [RECURSIVE] R1 AS query1,
  ...
  [RECURSIVE] Rn AS queryn
  query involving R1, R2, ..., Rn and other relations;
```

- `R1, R2, ... Rn` 可以是递归或互相调用的；包含递归的关系必须添加 `RECURSIVE` 标识符。
- 含义：首先计算 $R_1, R_2, \ldots, R_n$；然后计算询问；最后删除 $R_1, R_2, \ldots, R_n$.
- 例：

  ```
  WITH RECURSIVE Ancestor(x, y) AS
    (SELECT person AS x, parent AS y FROM Parent) union
    (SELECT a.x, p.parent AS y FROM Ancestor a, Parent p WHERE a.y=p.person)
  SELECT y FROM Ancestor WHERE x='James Bond';
  ```

合法递归：SQL 的合法递归有如下性质：

- 只支持**线性递归**，递归的左部分在右边只能出现一次！
- **单调性**：
  - 如果 P 是 Q 的函数，那么 P 对 Q 是单调的表现为：向 Q 中添加元组不会导致 P 中的元组被删除。
    - P 中元组保持不变或增多是允许的。
  - 递归左侧应该对递归右侧单调！
  - 聚合函数也可能导致非单调性：

    ```
    SELECT AVG(grade) FROM SC WHERE sno='S1';
    ```

    那么向 SC 中添加 'S1' 的成绩可能导致聚合结果平均分变化，从而原来的元组丢失，不具有单调性。