6. Authorization

Distinguish

- *Identification*: attesting a person's identity;
- *Authentication*: confirm that identity.
- *Authorization*: verify permittence.

Authorization thus presupposes authentication, which presupposes identification.

We implement authorization by means of *access rights*.

- Databases may contain confidential information.
- Not every user is allowed to initiate changes of the database.
- Access rights are a means to control access to the data; access rights may be assigned for users or attached to certain roles of the users.

Roles

- `CREATE ROLE <RoleName>`
- `DROP ROLE <RoleName>`
- `GRANT <RoleName> TO <users>`
- `REVOKE <RoleName> FROM <users>`

For simplicity, we consider assigning rights to users directly.

Users and objects

If PUBLIC is assigned a right, then all users have that right.

- We may GRANT and REVOKE a right to/from a user.
- Objects, which may be assigned a right, are tables, columns, views, and others.

Rights

- SELECT, INSERT, UPDATE, DELETE, REFERENCES, USAGE, TRIGGER, ...
- Syntax:

```
GRANT <List von rights>
ON <Objekt>
TO <List of users> [WITH GRANT OPTION]

REVOKE [GRANT OPTION FOR] <List of rights>
ON <Objekt>
FROM <List of users> {RESTRICT | CASCADE}
```

- If a right is assigned with GRANT OPTION, then the right could be passed to others, as well.

Maintenance of Access Rights

The creator of a base table owns all possible rights on that table, i.e. right `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `REFERENCES` und `TRIGGER`.

Example:

Assume a user `Admin` who created all the tables in the database. Thus Admin owns all rights.
We want `Admin` to grant user `PUBLIC` read acces to table `Country`.
Moreover, user `Assistent` and `Tutor` should get the rights to read, insert, delete and update that table in a way, such that they can pass the rights to others, as well.
Finally, user `SysProg` should get right `REFERENCES` and `TRIGGER` for table `Country`.

```
GRANT SELECT ON Land TO PUBLIC

GRANT SELECT, INSERT, DELETE, UPDATE
ON Land TO Assistant, Tutor WITH GRANT OPTION

GRANT REFERENCES, TRIGGER
ON Land TO SysProg
```

Remarks

- We introduce access rights for foreign keys, integrity constraints and trigger as these concepts could be used to infer unauthorized information.
- We call an acces right *abandoned*, if the right which was required for its granting is being revoked and no other granting has been performed which is still not revoked.
- If `REVOKE` is performed with option`CASCADE`, the also implied abandoned rights are revoked;
- Option `RESTRICT` would produce an error condition when abandoned rights would result.

User Assistant grants user Tutor INSERT-right for table Country.

```
GRANT INSERT ON Country TO Tutor
```

Now Admin executes some REVOKEs.

```
REVOKE INSERT ON Country FROM Tutor
```

Tutor keeps that right as it was independently granted by Assistant.

```
REVOKE INSERT ON Country FROM Assistant CASCADE
```

Now Assistant looses the right and Tutor as well.

Assume, Admin executes instead the final operation

```
REVOKE GRANT OPTION FOR INSERT ON Country
FROM Assistant CASCADE
```

Now Assistant keeps the INSERT-right, however Tutor will lose it as the option to grant the right to him is revoked.