DATENBANK-ARCHITEKTUR FÜR FORTGESCHRITTENE

Sicherheit und Zugriffssteuerung

Dani Schnider

Unterscheidung von SQL-Befehlen

DML = Data Manipulation Language

- SELECT (1)
- INSERT
- UPDATE
- DELETE
- MERGE
- LOCK TABLE

DDL = Data Definition Language

- CREATE
- ALTER
- DROP
- TRUNCATE

Implizites COMMIT

DCL = Data Control Language (2)

- GRANT
- REVOKE

Implizites COMMIT

TCL = Transaction Control Language (2)

- COMMIT
- ROLLBACK
- SAVEPOINT

(1) Teilweise als DQL (Data Query Language) bezeichnet

(2) Wird bei Oracle als DDL bezeichnet

USERS, PRIVILEGES & ROLES

Sicherheit und Zugriffssteuerung

User Accounts

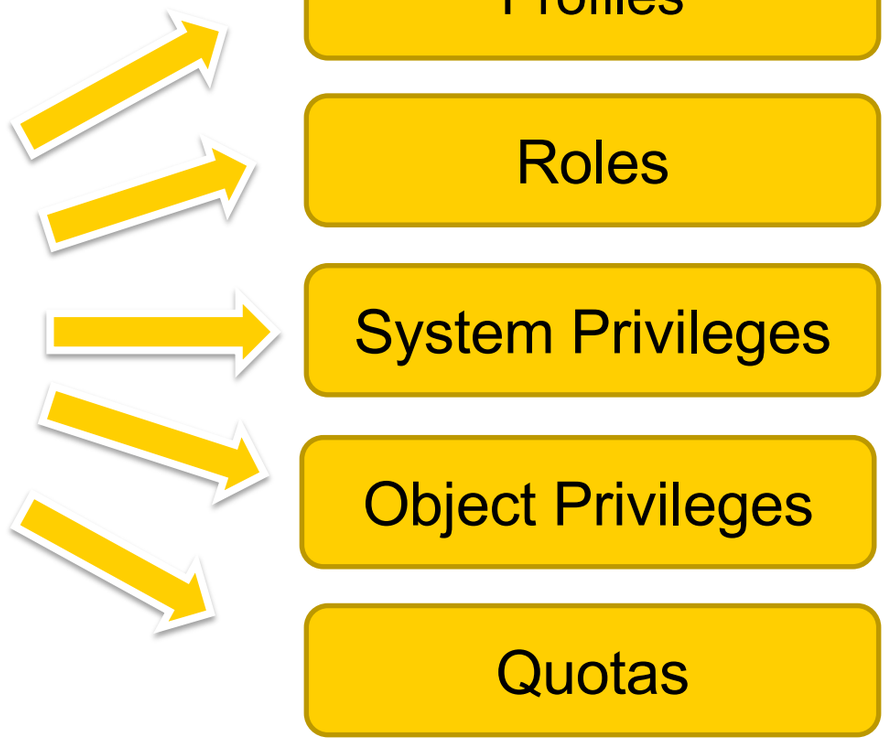
Erstellen eines User Accounts:

```
CREATE USER scott IDENTIFIED BY password$123
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON users
QUOTA 1M ON examples
PROFILE default;
```

Schema:

- Logische Zusammenfassung von Datenbankobjekten (Tabellen, Views, etc.)
- In Oracle entspricht ein Schema einem User Account, welches eigene Datenbankobjekte besitzt
- Können als Schema-only Accounts erstellt werden:

```
CREATE USER myapp NO AUTHENTICATION;
```



Quotas und Profiles

Quotas

- Speicherbeschränkungen pro Tablespace
- Kann übersteuert werden mit Systemprivileg UNLIMITED TABLESPACE

Profiles

- Systemlimiten pro User (CPU, Memory, Connection Time, etc.)
- Passwort-Regeln
- Pro User kann ein Profile zugewiesen werden

PROFILE	RESOURCE_NAME	RESOURCE_TYPE	LIMIT	COMMON	INHERITED
DEFAULT	COMPOSITE_LIMIT	KERNEL	UNLIMITED	NO	NO
DEFAULT	PRIVATE_SGA	KERNEL	UNLIMITED	NO	NO
DEFAULT	CONNECT_TIME	KERNEL	UNLIMITED	NO	NO
DEFAULT	IDLE_TIME	KERNEL	UNLIMITED	NO	NO
DEFAULT	LOGICAL_READS_PER_CALL	KERNEL	UNLIMITED	NO	NO
DEFAULT	LOGICAL_READS_PER_SESSION	KERNEL	UNLIMITED	NO	NO
DEFAULT	CPU_PER_CALL	KERNEL	UNLIMITED	NO	NO
DEFAULT	CPU_PER_SESSION	KERNEL	UNLIMITED	NO	NO
DEFAULT	SESSIONS_PER_USER	KERNEL	UNLIMITED	NO	NO
DEFAULT	PASSWORD_GRACE_TIME	PASSWORD	7	NO	NO
DEFAULT	PASSWORD_LOCK_TIME	PASSWORD	1	NO	NO
DEFAULT	PASSWORD_VERIFY_FUNCTION	PASSWORD	NULL	NO	NO
DEFAULT	PASSWORD_REUSE_MAX	PASSWORD	UNLIMITED	NO	NO
DEFAULT	FAILED_LOGIN_ATTEMPTS	PASSWORD	10	NO	NO
DEFAULT	PASSWORD_LIFE_TIME	PASSWORD	180	NO	NO
DEFAULT	INACTIVE_ACCOUNT_TIME	PASSWORD	UNLIMITED	NO	NO
DEFAULT	PASSWORD_ROLLOVER_TIME	PASSWORD	0	NO	NO
DEFAULT	PASSWORD_REUSE_TIME	PASSWORD	UNLIMITED	NO	NO

Privileges

System Privileges

- **Was** darf ein User tun?
- Beispiele: CREATE SESSION, CREATE TABLE, ALTER USER, SELECT ANY TABLE, …
- Data Dictionary Views: DBA_SYS_PRIVS

```
GRANT CREATE SESSION, CREATE TABLE TO dani;
```

Object Privileges

- Auf **welche Daten** darf ein User zugreifen?
- Beispiele: SELECT, INSERT, UPDATE, DELETE, EXECUTE, ALTER, …
- Für INSERT und UPDATE auch auf Attributeben definierbar
- Bezieht sich immer auf ein Datenbankobjekt (Tabelle, View, Prozedur)

```
GRANT SELECT, INSERT, UPDATE ON emp TO dani;
GRANT UPDATE(sal,comm) ON emp TO hr_user;
GRANT EXECUTE ON load_procedure TO dwh_run;
```

Roles

Rolle = Gruppierung von Privilegien und Rollen

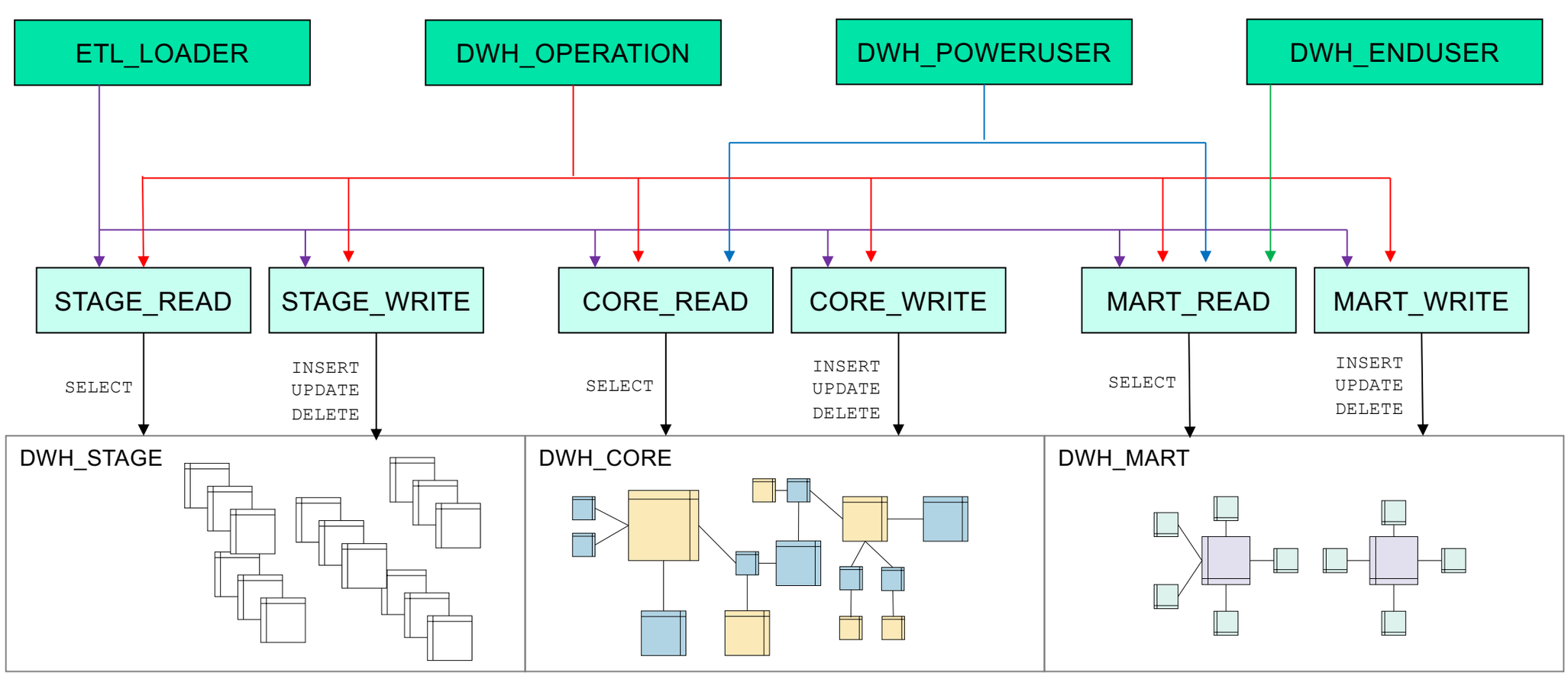
- Vereinfachung von Zugriffsberechtigungen
- Einer Rolle können System und/oder Object Privileges zugewiesen werden
- Einer Rolle können weitere Rollen zugewiesen werden
- Einem User können Privilegien direkt oder via Rollen zugewiesen werden
- Eine Rolle kann optional mittels Passwort geschützt werden

```
CREATE ROLE power_users;
GRANT SELECT, INSERT, UPDATE, DELETE ON emp TO power_users;
GRANT SELECT, INSERT, UPDATE, DELETE ON dept TO power_users;

CREATE ROLE student_role;
CREATE ROLE teacher_role;
GRANT power_users TO student_role, teacher_role;

GRANT teacher_role TO dani;
```

Rollenkonzept (Beispiel)



COLUMN-/ROW-LEVEL SECURITY

Sicherheit und Zugriffssteuerung

Column-Level Security

EMPNO	ENAME	JOB	MGR	HIREDATE	COMM	DEPTNO
7782	CLARK	MANAGER	7839	09.06.1981		10
7839	KING	PRESIDENT		17.11.1981		10
7934	MILLER	CLERK	7782	23.01.1982		10
7566	JONES	MANAGER	7839	02.04.1981		20
7902	FORD	ANALYST	7566	03.12.1981		20
7876	ADAMS	CLERK	7788	12.01.1983		20
7369	SMITH	CLERK	7902	17.12.1980		20
7788	SCOTT	ANALYST	7566	09.12.1982		20
7521	WARD	SALESMAN	7698	22.02.1981	500	30
7844	TURNER	SALESMAN	7698	08.09.1981	0	30
7499	ALLEN	SALESMAN	7698	20.02.1981	300	30
7900	JAMES	CLERK	7698	03.12.1981		30
7698	BLAKE	MANAGER	7839	01.05.1981		30
7654	MARTIN	SALESMAN	7698	28.09.1981	1400	30

Row-Level Security

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	02.04.1981	2975		20
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7876	ADAMS	CLERK	7788	12.01.1983	1100		20
7369	SMITH	CLERK	7902	17.12.1980	800		20
7788	SCOTT	ANALYST	7566	09.12.1982	3000		20

Row-Level Security

```
CREATE OR REPLACE VIEW EMP_20 AS
SELECT * FROM EMP
WHERE DEPTNO = 20;

GRANT SELECT ON EMP_20 TO JONES, FORD, ADAMS, SMITH, SCOTT;
```

```
CREATE OR REPLACE VIEW EMP_IN_MY_DEPT AS
SELECT * FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO FROM EMP WHERE ENAME = USER);

GRANT SELECT ON EMP_IN_MY_DEPT TO PUBLIC;
```