

Lösungen zu DDL Befehlen

Lösungen zur Tabellenerstellung

- a)

```
CREATE TABLE mitarbeiter (  
    ma_nr          NUMBER(4),  
    m_name         VARCHAR2(10),  
    beruf          VARCHAR2 (9),  
    chef_nr        NUMBER(4),  
    einstellungsdatum DATE,  
    gehalt          NUMBER(7,2),  
    provision       NUMBER(7,2),  
    abt_nr         NUMBER(2)  
);
```
- b)

```
CREATE TABLE test AS SELECT * FROM emp;
```


oder

```
CREATE TABLE mitarbeiter (  
    ma_nr, m_name, beruf , chef_nr ,  
    einstellungsdatum, gehalt , provision,  
    abt_nr)  
  
AS SELECT empno, ename, job, mgr,  
        hiredate, sal, comm, deptno  
  
FROM emp  
WHERE 1=2;
```
- c)

```
ALTER TABLE mitarbeiter  
ADD (bemerkung VARCHAR2(30));
```
- d)

```
ALTER TABLE mitarbeiter DROP COLUMN bemerkung;
```
- e)

```
(*) CREATE TABLE test AS SELECT * FROM emp;  
INSERT INTO test SELECT * FROM test;
```

→ ca. 10 mal ' in SQL*Plus betätigen, um den Befehl wiederholt auszuführen.

oder

```
CREATE TABLE test AS  
SELECT e.* FROM emp e, emp, emp;
```
- f)

```
(**) ALTER TABLE test ADD (nr number);  
UPDATE test SET nr = ROWNUM;
```

```
ALTER TABLE test  
ADD CONSTRAINT test_nr_pk PRIMARY KEY (nr);
```

Lösungen zu Constraints

```
a) CREATE TABLE firma (
    f_nr          NUMBER(4) PRIMARY KEY,
    name          VARCHAR2(30),
    adresse       VARCHAR2(30));

CREATE TABLE person (
    f_nr          NUMBER(4) CONSTRAINT person_fk
                                REFERENCES firma(f_nr),
    p_nr          NUMBER(4),
    name          VARCHAR2(30),
    abteilungs_nr NUMBER(4),
    CONSTRAINT person_pk PRIMARY KEY (f_nr,p_nr));

CREATE TABLE notiz (
    f_nr          NUMBER(4),
    p_nr          NUMBER(4),
    k_nr          NUMBER(4),
    notiz_text    VARCHAR2(200),
    notiz_status  VARCHAR2(20) CONSTRAINT notiz_ck
                                CHECK (notiz_status IN
                                        ('OFFEN','ERLEDIGT','TERMIN')),
    datum        DATE DEFAULT SYSDATE,
    CONSTRAINT notiz_pk
                PRIMARY KEY (f_nr, p_nr, k_nr),
    CONSTRAINT notiz_fk FOREIGN KEY (f_nr, p_nr)
                REFERENCES person (f_nr, p_nr));

b) INSERT INTO firma VALUES
    (1,'Integrata Training AG','München');
INSERT INTO firma VALUES
    (2,'UITAG','Stuttgart');
INSERT INTO firma VALUES
    (3,'ALTER data GmbH','Tübingen');

INSERT INTO person VALUES (1,1,'Herr Huber',10);
INSERT INTO person VALUES (1,2,'Herr Maier',20);
INSERT INTO person VALUES (2,1,'Herr Schmidt',10);
INSERT INTO person VALUES (3,1,'Herr Bejic',10);

INSERT INTO notiz VALUES
    (1,1,1,'Hat angerufen','OFFEN','01-JAN-99');
INSERT INTO notiz VALUES
    (1,1,2,'Termin vereinbart','ERLEDIGT','01-MAI-99');
```

- c) Diese Daten verstoßen gegen die Integritätsregel und werden nicht angenommen.

Fehler:

ORA-00001: unique constraint (SCOTT.SYS_C00...) violated

- d) Nein. Fehler: ORA-02292: integrity constraint
(SCOTT.PERSON_FK) violated - child record found

Durch die Verknüpfung der Tabellen über eine Primary/Foreign Key Beziehung kümmert sich die Datenbank um die Konsistenz der Daten. Wenn Sie die Firmen löschen könnten, die noch abhängige Datensätze haben, würden diese später alleine bestehen (Mitarbeiter einer Firma ohne Ihre Firma). Deshalb ist ein Löschen nicht möglich.

- e)

```
ALTER TABLE person DROP CONSTRAINT person_fk;
INSERT INTO person VALUES (4,1, 'Herr Habicht',10);
ALTER TABLE person ADD CONSTRAINT person_fk
    FOREIGN KEY (f_nr) REFERENCES firma (f_nr);
```

oder

```
ALTER TABLE person DISABLE bzw. ENABLE CONSTRAINT
person_fk;
```