Blatt 06 Datenbanken

Aufgabe 1]

1. Kunde( KNR, Vorname, Nachname, Telefonnummer, Adresse, Haustiername, Impfung )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Kunde |  |  |  |
| KNR | Vorname | Nachname | TelefonNr | Adresse | Haustiername | Impfung |
| 0000 | Max | Mustermann | 0664/34521 | Musterstr.3 | Bello | Tollwutimpfung |
| 0000 | Max | Mustermann | 0676/39468 | Musterstr.3 | Kratzi | RCP - Impfung |
| 0000 | Max | Mustermann | 0664/34521 | Musterstr.3 | Kratzi | Tollwutimpfung |
| 0000 | Max | Mustermann | 0676/39468 | Musterstr.3 | Bello | Tollwutimpfung |
| 0000 | Max | Mustermann | 0676/39468 | Musterstr.3 | Kratzi | Tollwutimpfung |
| 0000 | Max | Mustermann | 0664/34521 | Musterstr.3 | Kratzi | RCP - Impfung |

1. Schlüsselkandidat: (TelefonNr., Haustiername, KNR , Impfung) // 1NF
2. (Haustiername 🡪 Impfung)

(Telefonnummer 🡪 Haustiername)

1. Redundanz: Alle Daten in dieser Tabelle werden mehrmals gespeichert.

Einfügeanomalie: Ein neuer Kunde kann nicht angelegt werden ohne dass das Tier eine Impfung hat.

Entfernungsanomalie: Wenn eine Telefonnummer gelöscht wird, werden auch die anderen Daten entfernt.

Änderungsanomalie: Wenn eine Adresse oder TelNr geändert wird, muss man dies in mehreren Zeilen ändern.

1. Kunde( KNR, Vorname, Nachname, Telefonnummer, Adresse, Haustiername, Impfung )

Kundename (KNR, Vorname, Nachname)

Kundendaten(KNR, Adresse)

Kundentele (KNR, Telefonnummer)

Haustiername(KNR, Haustiername)

Impfung( Haustiername, Impfung)

Aufgabe 2]

1. SELECT p.\*, g.name

FROM person p

FULL JOIN gruppe g ON p.email = g.emailowner

SELECT p.\*, g.name

FROM person p, gruppe g

WHERE p.email = g.emailowner

UNION

SELECt p.\*, NULL AS name

FROM person p, gruppe g

WHERE p.email != g.emailowner

EXCEPT

SELECT p.\*, NULL AS name

FROM person p, gruppe g

WHERE p.email = g.emailowner

1. SELECT g.name, 'Gruppe' as typ

FROM gruppe g, person p

WHERE length(g.name) > 10

UNION

SELECT p.vorname, 'Person' as typ

FROM person p

WHERE( p.vorname LIKE 'N%'

OR p.vorname LIKE '%n%')

AND EXTRACT(YEAR FROM p.geburtsdatum ) = 1991;

1. SELECT SUM (counter) as Namen,

CASE

WHEN counter < 10 THEN (SELECT 'wenig Treffer')

WHEN counter >= 10 THEN (SELECT 'NULL')

END as Kommentar

FROM (SELECT COUNT (\*) as counter

FROM (SELECT g.name

FROM gruppe g, person p

WHERE length(g.name) > 10

UNION

SELECT p.vorname

FROM person p

WHERE( p.vorname LIKE 'N%'

OR p.vorname LIKE '%n%')

AND EXTRACT(YEAR FROM p.geburtsdatum ) = 1991 )as namen

GROUP By namen) as outerquery

GROUP By counter

Aufgabe 3]

1. CREATE TABLE magnicht (

email1 VARCHAR(255) REFERENCES person(email) NOT NULL,

email2 VARCHAR(255) REFERENCES person(email) NOT NULL,

vondatum DATE NOT NULL,

CONSTRAINT magnichtKEY PRIMARY KEY(email1,email2)

);

1. INSERT INTO magnicht(

email1,

email2,

vondatum

) values (

'Jan.Guenther@sms.at',

'P.Winkler@gmx.net',

to\_date('1999-12-30','YYYY-MM-DD')

);

Aufgabe 4]

1. ALTER TABLE hatfreund

ADD bisdatum DATE;

1. SELECT h.email, h.emailfreund

FROM hatfreund h, magnicht m

WHERE h.email = m.email1

AND m.email2 = h.emailfreund

1. Keine Inkonsistenzen.