

Baederverwaltung - Die Datenbank

Eine Baederverwaltung benoetigt eine Datenbank.

Beschreibung der Datenbank

Mitarbeiter sind eindeutig identifiziert durch die Sozialversicherungsnummer (SVNR). Es wird auch noch der Name (NAME), die Adresse (ADRESSE) und das Geburtsdatum (GEBDAT) gespeichert. Optional kann für gekündigte Mitarbeiter ein Kündigungsdatum (DATUM) gespeichert werden. Kassadienst und Bademeister sind Mitarbeiter. Für Bademeister ist bekannt, welche Ausbildung sie absolviert haben (AUSBILDUNG). Es wird auch gespeichert, welche Bademeister von welchen anderen Bademeistern vertreten werden können. Ein Schwimmbecken wird identifiziert durch eine Bezeichnung (BEZ), außerdem ist die Wassertiefe (TIEFE) gespeichert. Messwerte sind identifiziert durch das Schwimmbecken, in dem sie gemessen wurden und dem Zeitpunkt (ZEITPKT). Dann wird noch die Bezeichnung des Wertes (BEZ) z.B. pH-Wert und der Wert (WERT) selbst gespeichert. Sollte ein Werte nicht der Norm entsprechen, so wird gespeichert, welche Maßnahmen in Bezug auf diesen einen Wert eingeleitet wurden. Jede Maßnahme wird identifiziert durch einen Zeitpunkt (ZEITPKT) und ein Mittel (MITTEL) das angewendet wurde. Außerdem wird die Menge (MENGE) des Mittels gespeichert. Es wird auch gespeichert welcher Bademeister welche Maßnahmen getroffen hat. Bademeister haben auch noch eine andere Aufgabe: die Aufsicht. Daher muss vermerkt werden, welche Bademeister in welchem Zeitraum (VON, BIS) welche Becken als Rettungsschwimmer beaufsichtigen. Eine Eintrittskarte wird identifiziert durch eine laufende Tagesnummer (LFDNR) und durch ein Datum (VON) von dem an sie gültig ist. Außerdem wird das Ende der Gültigkeit (BIS) und der Preis (PREIS) gespeichert. Es wird auch gespeichert, welche Mitarbeiter vom Kassadienst welche Karte verkauft haben. Eine Karte kann dabei nur von einem Kassadienstmitarbeiter verkauft werden. Kategorien werden identifiziert durch eine Bezeichnung (BEZ) z.B. Student, halbtags... Dazu wird noch eine genaue Beschreibung (BESCHREIBUNG) gespeichert. Karten können mehrere Kategorien zugeordnet sein. Eine Jahreskarte ist eine besondere Form der Karte. Es wird hierfür gespeichert welche Kunden welche Jahreskarten besitzen. Eine Jahreskarte ist hingegen einem einzigen Kunden zugeordnet. Ein Kunde ist eindeutig identifiziert durch eine Kundennummer (KNR). Zu jedem Kunden wird auch der Name (NAME), die Adresse (ADRESSE) sowie ein Foto (FOTO) gespeichert.

1)

Geben Sie jene Kassadienst-Mitarbeiter aus, die ab dem 1. Januar 1970 geboren wurden. Verwenden Sie dazu die Funktion TO_DATE und folgendes Datumsformat: DD-MM-YYYY. Ordnen Sie die Ergebnisse absteigend nach der Sozialversicherungsnummer.

```
SELECT *
FROM Mitarbeiter NATURAL JOIN Kassadienst
WHERE gebdat >= TO_DATE('01-01-1970', 'DD-MM-YYYY')
ORDER BY svnr DESC;
```

```
SELECT *
FROM Mitarbeiter JOIN Kassadienst USING(svnr)
```

```
WHERE gebdat >= TO_DATE('01-01-1970', 'DD-MM-YYYY')
ORDER BY svnr DESC;
```

2.1)

Geben Sie das Geburtsdatum des ältesten Bademeisters aus.

```
SELECT MIN(gebdat)
FROM Bademeister NATURAL JOIN Mitarbeiter;
```

2.2)

Geben Sie die Sozialversicherungsnummer, den Namen und das Geburtsdatum des ältesten Bademeisters aus.

```
SELECT svnr, name, gebdat
FROM Bademeister NATURAL JOIN Mitarbeiter
WHERE gebdat = (SELECT MIN(gebdat)
                FROM Bademeister NATURAL JOIN Mitarbeiter);
```

```
SELECT svnr, name, gebdat
FROM Bademeister NATURAL JOIN Mitarbeiter
WHERE gebdat <= ALL (SELECT gebdat
                    FROM Bademeister NATURAL JOIN Mitarbeiter);
```

```
SELECT m.svnr, m.name, m.gebdat
FROM Mitarbeiter m JOIN Bademeister b on (m.svnr = b.svnr)
WHERE NOT EXISTS (SELECT *
                  FROM Mitarbeiter NATURAL JOIN Bademeister
                  WHERE gebdat < m.gebdat);
```

2.3)

Geben Sie die Sozialversicherungsnummer, den Namen und das Geburtsdatum des ältesten Bademeisters mit der Ausbildung 'Rettungsschwimmer' aus.

```
SELECT svnr, name, gebdat
FROM Bademeister NATURAL JOIN Mitarbeiter
WHERE ausbildung = 'Rettungsschwimmer'
AND gebdat <= ALL (SELECT gebdat
                   FROM Bademeister NATURAL JOIN Mitarbeiter
                   WHERE ausbildung = 'Rettungsschwimmer');
```

```
SELECT svnr, name, gebdat
FROM Bademeister NATURAL JOIN Mitarbeiter
WHERE ausbildung = 'Rettungsschwimmer'
AND gebdat = (SELECT MIN(gebdat)
              FROM Bademeister NATURAL JOIN Mitarbeiter
              WHERE ausbildung = 'Rettungsschwimmer');
```

```
SELECT m.svnr, m.name, m.gebdat
FROM Mitarbeiter m JOIN Bademeister b on (m.svnr = b.svnr)
WHERE b.ausbildung = 'Rettungsschwimmer'
AND NOT EXISTS (SELECT *
                FROM Mitarbeiter NATURAL JOIN Bademeister
                WHERE ausbildung = 'Rettungsschwimmer'
                AND gebdat < m.gebdat);
```

3.1)

Geben Sie eine Liste aller Jahreskartenbesitzer aus, auf der die Kundennummer und der Name des Kunden zusammen mit der laufenden Tagesnummer der Jahreskarte(n) verzeichnet sind.

```
SELECT k.knr, k.name, j.lfdnr
FROM Kunde k JOIN Jahreskarte j ON (k.knr = j.kunde);
```

3.2)

Geben Sie eine Liste ALLER Kunden (knr, name) aus zusammen mit der laufenden Tagesnummer der Jahreskarte(n), die sie besitzen. Für Kunden, die keine Jahreskarte haben, soll stattdessen 'keine Karte' ausgegeben werden. (Stichwort: COALESCE)

```
SELECT k.knr, k.name, COALESCE(j.lfdnr, 'keine Karte') AS tagesnummer
```

FROM Kunde k LEFT OUTER JOIN Jahreskarte j ON (k.knr = j.kunde);

3.3)

Geben Sie eine Liste ALLER Kunden (knr, name) aus zusammen mit der laufenden Tagesnummer der Jahreskarte(n), die sie besitzen, sowie den Namen des Mitarbeiters, der sie ihnen verkauft hat. Fuer Kunden, die keine Jahreskarte haben, soll anstelle der Tagesnummer 'keine Karte' und anstelle des Namens des Mitarbeiters 'kein Verkaeuf'er' ausgegeben werden. (Stichwort: COALESCE)

```
SELECT ku.knr, ku.name, COALESCE(j.lfdnr, 'keine Karte') AS tagesnummer, COALESCE(m.name, 'kein Verkaeuf'er') AS Verkaeuf'er
FROM Kunde ku LEFT OUTER JOIN Jahreskarte j ON (ku.knr = j.kunde) LEFT OUTER JOIN Karte ka ON (j.lfdnr = ka.lfdnr AND j.von = ka.von) LEFT OUTER JOIN Mitarbeiter m ON (ka.verkaeuf'er = m.svnr);
```

```
SELECT ku.knr, ku.name, COALESCE(j.lfdnr, 'keine Karte') AS tagesnummer, COALESCE(m.name, 'kein Verkaeuf'er') AS Verkaeuf'er
FROM Mitarbeiter m JOIN Karte ka ON (m.svnr = ka.verkaeuf'er) JOIN Jahreskarte j ON (j.lfdnr = ka.lfdnr AND j.von = ka.von) RIGHT OUTER JOIN Kunde ku ON (ku.knr = j.kunde);
```

4.1)

Mit Ausnahme von PH-Wert Bestimmungen (bez = 'pH-Wert'), soll die Bezeichnung, der Wert und der Zeitpunkt aller Messwerte ausgegeben werden, die fuer Nichtschwimmer-Becken (Tiefe maximal 135 cm) erhoben wurden.

```
SELECT m.bez, m.wert, m.zeitpkt
FROM Messwert m JOIN Becken b ON (m.becken = b.bez)
WHERE b.tiefe <= 135 AND m.bez != 'pH-Wert';
```

4.2)

Gesucht ist die Bezeichnung jener Becken, fuer die noch nie ein pH-Wert (Messwert bez = 'pH-Wert') erhoben wurde und die keine Nichtschwimmer-Becken sind. (Die Tiefe betraegt fuer Nichtschwimmerbecken maximal 135 cm.)

```
SELECT bez
FROM Becken
WHERE tiefe > 135
AND bez NOT IN (SELECT becken
                FROM Messwert
                WHERE bez = 'pH-Wert');
```

```
SELECT b.bez
FROM Becken b
WHERE tiefe > 135
AND NOT EXISTS (SELECT *
                FROM Messwert m
                WHERE m.bez = 'pH-Wert'
                AND m.becken = b.bez);
```

5.1)

Wieviele verschiedene Kategorien sind den Eintrittskarten mit der laufenden Tagesnummer '123123' zugeordnet?

```
SELECT COUNT(distinct kategorie) AS Anzahl
FROM hat
WHERE lfdnr = '123123';
```

5.2)

Geben Sie fuer ALLE Eintrittskarten die laufende Tagesnummer sowie den Preis aus und dazu die Anzahl der zugeordneten Kategorien.

```
SELECT k.lfdnr, k.preis, COUNT(h.kategorie) AS Anzahl
FROM Karte k LEFT OUTER JOIN hat h ON (k.lfdnr = h.lfdnr AND k.von = h.von)
GROUP BY k.lfdnr, k.von, k.preis;
```

```
select k.lfdnr, k.preis, (select count(kategorie) from hat where lfdnr=k.lfdnr and von
=k.von)as anzahl
from karte k;
```



```
AND svnr NOT IN (SELECT bademeister
FROM Massnahme
GROUP BY bademeister
HAVING COUNT(*) >= 3);
```

```
select name, gebdat
from mitarbeiter m
where not exists (select bez
from becken
where tiefe = 260 and bez not in
(select becken
from rdienst
where bademeister = m.svnr))
and (select count(*) from
massnahme where
bademeister = m.svnr) < 3
```

8)

Wieviele Messwerte werden pro Becken durchschnittlich erhoben?

```
SELECT AVG(a.anz) AS durchschnitt
FROM (SELECT COUNT(m.zeitpkt) AS anz
FROM Becken b LEFT OUTER JOIN Messwert m ON (b.bez = m.becken)
GROUP BY b.bez) a;
```

9)

Geben Sie die Bezeichnung und Tiefe jener Becken aus, fuer die bereits alle verschiedenen Messwerte (z.B. Clor, pH-Wert, etc.) erhoben wurden.

```
SELECT b.bez, b.tiefe
FROM Becken b JOIN Messwert m ON (b.bez = m.becken)
GROUP BY b.bez, b.tiefe
HAVING COUNT(DISTINCT m.bez) = (SELECT COUNT(DISTINCT bez)
FROM Messwert);
```

```
SELECT b.bez, b.tiefe
```

```

FROM Becken b
WHERE not exists (select *
                  from messwert
                  where bez not in (select bez
                                    from messwert m
                                    where m.becken=b.bez));

```

10)

Geben Sie eine Liste ALLER Kategorien (Bezeichnung und Beschreibung) aus und dazu die Anzahl der seit dem '01.01.2011' verkauften Eintrittskarten und wieviel diese in Summe gekostet haben. Verwenden Sie dazu die Funktion TO_DATE und folgendes Datumsformat: DD-MM-YYYY.

```

SELECT k.bez, k.beschreibung, coalesce(x.anz, 0) AS Anzahl, coalesce(x.gespr
eis, 0) AS Gesamtpreis
FROM Kategorie k LEFT OUTER JOIN (SELECT kategorie, COUNT(*) as anz, S
UM(preis) as gespreis
                                FROM hat NATURAL JOIN Karte
                                WHERE von >= TO_DATE('01-01-2011', 'DD-MM-YYYY')
                                GROUP BY kategorie) x
ON (k.bez = x.kategorie);

```

```

select bez, beschreibung, count(karte.lfdnr) as anzahl, coalesce(sum(preis), 0) as
gesamtpreis
from kategorie left join (hat join karte on (hat.lfdnr=karte.lfdnr and
hat.von=karte.von and karte.von >= TO_DATE('01-01-2011', 'DD-MM-YYYY')))) on
kategorie.bez=hat.kategorie
group by bez, beschreibung;

```