

# Fitness Center - Die Datenbank

Eine Fitnesscenterkette benötigt eine Datenbank.

## Beschreibung der Datenbank

Die Leiterin einer Fitnesscenterkette möchte, dass die Verwaltung in Zukunft mithilfe einer Datenbank erfolgt. Sie möchte ausserdem, eine Filialsuche auf der Webseite der Kette integrieren. Deshalb wird der Standort eines Fitnesscenters gespeichert. Dafür ist es notwendig, die Strasse (Strasse), die Hausnummer (Nr), die Postleitzahl (Plz) und den Ort (Ort) des Standortes zu kennen. Ein Standort wird eindeutig durch die Strasse und die Hausnummer identifiziert. Ein Fitnesscenter wird eindeutig durch den Standort identifiziert. Es wird auch der Name (Name), die Telefonnummer (TelNr) und die Mailadresse (Email) gespeichert. Von den Kunden werden folgende Daten gespeichert: Eine Kundennummer (KNR) dient der eindeutigen Identifikation eines Kunden. Ausserdem werden der Vorname (VName), der Nachname (NName), das Geburtsdatum (GebDat), die Adresse (Adresse), die Telefonnummer (Telefon) und falls der Kunde ein Abonnement hat, wird das Ablaufdatum des Abos gespeichert. Die Mitarbeiter werden eindeutig durch die Sozialversicherungsnummer (SVNR) identifiziert. Des Weiteren werden der Name (Name), die Telefonnummer (Telefon), die Adresse (Adresse) und das Eintrittsdatum (EintrittsDat) festgehalten. Zusätzlich ist bekannt welche Mitarbeiter in welchen Fitnesscentern arbeiten. Es werden zwei Arten von Mitarbeitern unterschieden: die administrativen Mitarbeiter und die Trainer. Von den Mitarbeitern der Administration ist die Funktion (Funktion) bekannt. Von den Trainern die Ausbildung (Ausbildung) und der Stundensatz (Stundensatz). Ein Fitnesscenter wird von genau einem Mitarbeiter geleitet umgekehrt kann ein Mitarbeiter nur ein Fitnesscenter leiten. Die Räume werden über eine Nummer (RNr) eindeutig pro Fitnesscenter identifiziert. Zusätzlich haben die Räume einen Namen (Name) und eine Kapazität (Kapazitaet). Ein Fitnesscenter stellt seinen Kunden verschiedene Geräte zur Verfügung. Jedes dieser Geräte soll in der Datenbank erfasst werden. Ein Gerät hat eine eindeutige Seriennummer (SNr), einen Namen (Name), eine Beschreibung (Beschreibung), ein Anschaffungsdatum (AnschaffDat) und eine Garantiezeit in Monaten (Garantie). Ausserdem wird noch gespeichert, in welchem Raum sich das Gerät befindet. Es werden verschiedene Kurse angeboten. Diese werden durch eine eindeutige Nummer (KNr) identifiziert. Zusätzlich werden die Beschreibung (Beschreibung) wie z.B.: 'High Intensity Cardio Programm für Fortgeschrittene', ein Datum (Datum) und die Beginn- und Endzeit (Beginn und Ende) eines Kurses festgehalten. Ein Kurs wird in einem Raum unter der Leitung von ein bis mehreren Trainern abgehalten. Die Kurse können eine hierarchische Organisation haben. Ein Kurs kann mehrere vorausgehende und nachfolgende Kurse haben. Es gibt aber auch Kurse, die keine vorausgehenden und nachfolgenden Kurse haben. Die Kunden können an mehreren Kursen teilnehmen. Ausserdem werden noch Einzeltrainings angeboten. Dazu ist bekannt, welche Trainer, welche Kunden mit welchem Schwerpunkt (Schwerpunkt) betreuen.

---

1)

Geben Sie jene Mitarbeiter aus, die nach 01-01-2013 eingestellt wurden. Verwenden Sie hierzu die Funktion TO\_DATE mit dem Datumsformat 'DD-MM-YYYY'. Sortieren Sie das Ergebnis aufsteigend nach Eintrittsdatum.

```
SELECT * FROM mitarbeiter
WHERE eintrittsdat > to_date('01-01-2013', 'DD-MM-YYYY')
ORDER BY eintrittsdat;
```

---

2)

Geben Sie jene Mitarbeiter (Name) aus, die in mehr als 2 Fitnesscenter angestellt sind.

```
SELECT m.name
FROM mitarbeiter m
JOIN arbeitet a ON m.svnr=a.svnr
HAVING count(*)>2
GROUP BY m.name, m.svnr;
```

```
SELECT name
FROM mitarbeiter
NATURAL JOIN arbeitet
HAVING count(*)>2
GROUP BY name, svnr;
```

```
SELECT m.name
FROM mitarbeiter m
WHERE m.svnr in (SELECT svnr FROM arbeitet HAVING count(svnr)>2 GROUP
BY svnr);
```

---

3.1)

Geben Sie den hoechsten Stundensatz der Trainer aus.

```
SELECT MAX(stundensatz) as stundensatz
FROM trainer;
```

```
SELECT stundensatz
FROM trainer m
WHERE stundensatz >= ALL(SELECT stundensatz
FROM trainer);
```

```
SELECT stundensatz
FROM trainer m
WHERE NOT EXISTS ( SELECT * FROM trainer WHERE stundensatz>m.stunde
nsatz);
```

---

3.2)

Geben Sie jene Trainer (Name, Stundensatz) aus, die den hoechsten Stundensatz haben.

```
SELECT m.name, t.stundensatz
FROM mitarbeiter m NATURAL JOIN trainer t
WHERE t.stundensatz = (SELECT MAX(stundensatz) FROM trainer);
```

```
SELECT m.name, t.stundensatz
FROM mitarbeiter m NATURAL JOIN trainer t
WHERE t.stundensatz >= ALL (SELECT stundensatz FROM trainer);
```

```
SELECT m.name, t.stundensatz
FROM mitarbeiter m NATURAL JOIN trainer t
WHERE NOT EXISTS (SELECT * FROM trainer
WHERE stundensatz>t.stundensatz);
```

---

3.3)

Geben Sie jene Trainer (Name) aus, die nach 01-01-2005 eingestellt und den hoechsten Stundensatz haben. Verwenden Sie hierzu die Funktion TO\_DATE mit dem Datumsformat 'DD-MM-YYYY'.

```
SELECT m.name, t.stundensatz
FROM mitarbeiter m NATURAL JOIN trainer t
WHERE t.stundensatz = (SELECT MAX(t.stundensatz)
FROM trainer t JOIN mitarbeiter m on t.svnr=m.svnr where m.eintritts
dat>TO_DATE('01-01-2005', 'DD-MM-YYYY'))
and m.eintrittsdat>TO_DATE('01-01-2005', 'DD-MM-YYYY');
```

```
SELECT m.name, t.stundensatz
```

```
FROM mitarbeiter m NATURAL JOIN trainer t
WHERE t.stundensatz >= ALL (SELECT t.stundensatz
                           FROM trainer t JOIN mitarbeiter m on t.svnr=m.svnr where m.eintr
ittsdat>TO_DATE('01-01-2005', 'DD-MM-YYYY'))
and m.eintrittsdat>TO_DATE('01-01-2005', 'DD-MM-YYYY');
```

```
SELECT m.name, t.stundensatz
FROM mitarbeiter m NATURAL JOIN trainer t
WHERE NOT EXISTS (SELECT * FROM trainer t2 JOIN mitarbeiter m on t2.svnr
=m.svnr where m.eintrittsdat>TO_DATE('01-01-2005', 'DD-MM-YYYY')
and stundensatz>t.stundensatz)
and m.eintrittsdat>TO_DATE('01-01-2005', 'DD-MM-YYYY');
```

---

4.1)

Geben Sie aus wieviele Geraete die Fitnesscenterkette in Summe besitzt. (Anzahl).

```
SELECT count(*) as Anzahl from Geraet;
```

---

4.2)

Geben Sie fuer jedes Fitnesscenter (Name) die Anzahl der Fitnessgeraete aus, die eine Garantie von unter 6 Monaten haben.

```
SELECT f.name, count(*) as anzahl
FROM fitnesscenter f JOIN ist_im im ON(f.strasse=im.fcstrasse and f.nr=im.fcnr)
JOIN Geraet g on(g.snr=im.snr)
where garantie < 6
GROUP BY f.name;
```

---

5)

Geben Sie Trainer (Name) aus, die die meisten Kunden mit dem Schwerpunkt 'Salsa' betreuen.

```
SELECT t.name
FROM mitarbeiter t join einzeltraining e on t.svnr=e.trainer
where e.schwerpunkt= 'Salsa'
```

```
having count(e.kundennr) <= ALL(SELECT count(Kundennr)
FROM einzeltraining where Schwerpunkt like 'Salsa' group by
Trainer)
group by t.name, t.svnr;
```

```
SELECT t.name
FROM mitarbeiter t join einzeltraining e on t.svnr=e.trainer
where e.schwerpunkt= 'Salsa'
having count(e.kundennr) <= (Select max(anzahl) from (SELECT count(Kundennr)
as anzahl
FROM einzeltraining where Schwerpunkt like 'Salsa' group by Tr
ainer))
group by t.name, t.svnr;
```

---

6.1)

Geben Sie jene Kunden (Vorname, Nachname, Ablaufdatum) aus, die ein Abo besitzen, das vor 2015 abläuft. Verwenden Sie hierzu die Funktion TO\_DATE mit dem Datumsformat 'DD-MM-YYYY'. Sortieren Sie das Ergebnis aufsteigend nach Kundennamen (Vorname, Nachname).

```
SELECT k.vname, k.nname, a.ablaufdat
FROM kunde k NATURAL JOIN abonnement a
WHERE a.ablaufdat<TO_DATE('01-01-2015', 'DD-MM-YYYY')
ORDER BY k.vname, k.nname;
```

---

6.2)

Geben Sie eine Liste ALLER Kunden (Vorname, Nachname, Ablaufdatum) aus, die ein Abo besitzen. Für Kunden, die kein Abo besitzen, soll 'kein Abo' ausgegeben werden. Verwenden Sie dazu die Funktion COALESCE und TO\_CHAR.

```
SELECT k.vname, k.nname, COALESCE(TO_CHAR(a.ablaufdat), 'kein Abo') as
ablaufdatum
FROM kunde k LEFT OUTER JOIN abonnement a ON (k.knr=a.knr);
```

---

6.3)

Geben Sie eine Liste ALLER Kunden (Vorname, Nachname, Ablaufdatum) aus, die ein Abo besitzen, das

vor 2015 abläuft. Für alle anderen Kunden, soll '-' ausgegeben werden. Verwenden Sie dazu die Funktion COALESCE. Sortieren Sie das Ergebnis aufsteigend nach Kundennamen (Vorname, Nachname).

```
SELECT k.vname, k.nname, COALESCE(TO_CHAR(a.ablaufdat), '-') as Ablaufd
atum
FROM kunde k LEFT OUTER JOIN abonnement a ON (k.knr=a.knr and a.ablaufd
at<TO_DATE('01-01-2015', 'DD-MM-YYYY'))
ORDER BY k.vname, k.nname;
```

---

7)

Geben Sie ALLE Trainer (Name, Kurse, Einzeltrainings) aus mit der Anzahl der Kurse die sie abgehalten haben, sowie der Anzahl der Einzeltrainings.

```
SELECT m.name,
(SELECT count(*) FROM abhaltung WHERE trainer= t.svnr) as kurse,
(SELECT count(*) FROM einzeltraining WHERE trainer= t.svnr) as einzeltrainings
FROM trainer t LEFT OUTER JOIN mitarbeiter m ON (t.svnr=m.svnr)
```

```
select NAME, KURSE, EINZELTRAININGS from
(select NAME, SVNR, count(KUNDENNR) as EINZELTRAININGS from TRAINER
natural join MITARBEITER
left outer join EINZELTRAINING on TRAINER=SVNR
group by NAME, SVNR)
natural join
(select NAME, SVNR, count(KURSNR) as KURSE from TRAINER natural join MI
TARBEITER
left outer join ABHALTUNG on TRAINER=SVNR
group by NAME, SVNR)
```

---

8)

Geben Sie die Namen jener Mitarbeiter aus, die vor 2005 in das Unternehmen eingetreten sind und in allen Fitnesscentern arbeiten. Verwenden Sie hierzu die Funktion TO\_DATE mit dem Datumsformat 'DD-MM-YYYY'.

```
SELECT m.name
FROM mitarbeiter m
WHERE NOT EXISTS(
SELECT *
```

```

FROM fitnesscenter f
WHERE NOT EXISTS(
  SELECT *
  FROM arbeitet a
  WHERE a.fcnr = f.nr AND a.fcstrasse = f.strasse and a.svnr=m.svnr
)
)
and m.eintrittsdat<TO_DATE('01-01-2005', 'DD-MM-YYYY')

```

```

SELECT name
FROM mitarbeiter NATURAL JOIN arbeitet
WHERE eintrittsdat<TO_DATE('01-01-2005', 'DD-MM-YYYY')
HAVING count(*) >= ALL(SELECT count(*)
  FROM fitnesscenter)
GROUP BY svnr, name;

```

---

9.1)

Geben Sie jene Kunden (Kundennummer) aus, die mindestens einen Kurs besucht haben, der einen Vorgaenger hat und die noch nie ein Einzeltraining gehabt haben.

```

SELECT distinct k.knr
FROM kunde k JOIN teilnehmen t ON (k.knr= t.kundenr) JOIN ist_nachfolger_von
nf ON (nf.nachfolger=t.kursnr)
WHERE k.knr not in (SELECT kundenr FROM einzeltraining);

```

```

SELECT distinct k.knr
FROM kunde k JOIN teilnehmen t ON (k.knr= t.kundenr) JOIN ist_nachfolger_von
nf ON (nf.nachfolger=t.kursnr)
WHERE not exists (SELECT * FROM einzeltraining where kundenr=k.knr);

```

---

9.2)

Geben Sie alle Kunden aus, die weder ein Abo haben, noch ein Einzeltraining und auch keinen Kurs besucht haben.

```

Select * from kunde k
where k.knr not in (select kundenr from einzeltraining )

```

```
and k.knr not in (select kundenr from teilnehmen)
and k.knr not in (select knr from abonnement);
```

```
Select * from kunde k
where not exists (select * from einzeltraining where kundenr=k.knr)
and not exists (select * from teilnehmen where kundenr=k.knr)
and not exists (select * from abonnement where knr=k.knr);
```

---

10)

Geben Sie jene Vormittagskurse (Beginnzeit bis 12 Uhr) (Bezeichnung, Datum, Beginnzeit) aus, die ab 01-06-2014 stattfinden werden. Verwenden Sie hierzu die Funktion TO\_DATE mit dem Datumsformat 'DD-MM-YYYY'. Sortieren Sie das Ergebnis nach Bezeichnung. (Bezeichnung, Datum, Beginnzeit im Format hh24mi). Tipp: Verwenden Sie die Funktion TO\_CHAR um die Beginnzeit mittels dem Format 'hh24mi' in eine Zeichenkette umzuwandeln.

```
SELECT bezeichnung, datum, TO_CHAR(beginn,'hh24mi') as beginn
FROM kurs
WHERE TO_CHAR(beginn,'hh24mi') <='1200' and datum >= to_date('01-06-201
4', 'DD-MM-YYYY')
order by bezeichnung;
```