

Marskolonie

Beispiel-Testdaten

1. Koordinate

```
INSERT INTO Koordinate (id, breitengrad, laengengrad) VALUES  
(1, 18.65, 226.20), -- Olympus Mons  
(2, -4.59, 137.44), -- Gale Crater  
(3, -14.56, 175.47); -- Valles Marineris
```

2. Stadt

```
INSERT INTO Stadt (stadt_id, stadt_name, koord_id) VALUES  
(1, 'Olympus Station', 1),  
(2, 'Gale Habitat', 2),  
(3, 'Valles Outpost', 3);
```

3. Adresse

```
INSERT INTO Adresse (adresse_id, stadt_id, strasse, hausnummer) VALUES  
(1, 1, 'Kuppelring', 'A1'),  
(2, 2, 'Versorgungsachse', 'B3');
```

4. Bewohner

```
INSERT INTO Bewohner (bewohner_id, adresse_id, vorname, nachname, geb) VALUES  
(1, 1, 'Lena', 'Kowalski', 20850312),  
(2, 2, 'Noah', 'Hernandez', 20791204);
```

5. Berufung

```
INSERT INTO Berufung (berufung_id, berufung_name, arbeitszeit, arbeitszeit_model)  
VALUES  
(1, 'Mars-Ingenieur', 42, 'Schichtbetrieb'),  
(2, 'Transportpilot', 40, 'Mission-basiert');
```

6. Mitarbeiter

```
INSERT INTO Mitarbeiter  
(mitarbeiter_id, berufung_id, bewohner_id, login, password, gehalt) VALUES  
(1, 2, 1, 'pilot01', 'mars123', 5200),  
(2, 1, 2, 'engineer01', 'redplanet', 6100),  
(3, 1, NULL, 'engineer02', 'habitat', 5900);
```

7. Lager

```
INSERT INTO Lager (lager_id, lager_typ, groesse) VALUES  
(1, 'Zentraallager Sauerstoff', 8000),  
(2, 'Nahrungsdepot', 5000);
```

8. Energiequelle

```
INSERT INTO Energiequelle  
(eg_id, lager_id, aktuelle_leistung, kapazitaet_max, eq_typ) VALUES  
(1, 1, 450, 600, 'Solarfeld'),  
(2, 2, 300, 500, 'Kernreaktor');
```

9. Fahrzeuge

```
INSERT INTO Fahrzeuge (fahrzeug_id, f_name, f_status) VALUES  
(1, 'Rover-T1', 'bereit'),  
(2, 'Cargo-T2', 'im Einsatz');
```

10. Route

```
INSERT INTO Route (route_id, start_id, end_id) VALUES  
(1, 1, 2),  
(2, 2, 3);
```

11. Transportwege

```
INSERT INTO Transportwege  
(tpw_id, mitarbeiter_id, fahrzeug_id, route_id, stadt_id, dauer, streckentyp, tpw_status)  
VALUES  
(1, 1, 1, 1, 6, 'Oberfläche', 'abgeschlossen'),  
(2, 1, 2, 2, 2, 10, 'Untertunnel', 'aktiv');
```

12. Ressource

```
INSERT INTO Ressource  
(ressource_id, t_typ, menge, gewicht, volumen, ablaufdatum, prioritaet) VALUES  
(1, 'Sauerstoff', 1200, 300, 50, '2095-12-31', 1),  
(2, 'Wasser', 900, 900, 80, '2090-01-01', 1),  
(3, 'Nahrung', 600, 500, 70, '2087-06-30', 2);
```

13. ist_gelagert_in

```
INSERT INTO ist_gelagert_in (lager_id, ressource_id) VALUES  
(1, 1),  
(1, 2),  
(2, 3);
```

14. transportiert

```
INSERT INTO transportiert (fahrzeug_id, ressource_id) VALUES  
(1, 2),  
(2, 3);
```

SQL-Abfragen

1. Alle Mars-Städte mit Koordinaten

```
SELECT s.stadt_name, k.breitengrad, k.laengengrad  
FROM Stadt s  
JOIN Koordinate k ON s.koord_id = k.id;
```

2. Mitarbeiter und ihre Rolle in der Kolonie

```
SELECT m.login, b.berufung_name, m.gehalt  
FROM Mitarbeiter m  
JOIN Berufung b ON m.berufung_id = b.berufung_id;
```

3. Mitarbeiter ohne zugewiesenen Wohnort

```
SELECT mitarbeiter_id, login  
FROM Mitarbeiter  
WHERE bewohner_id IS NULL;
```

4. Bewohner mit Adresse und Wohnort

```
SELECT bw.vorname, bw.nachname, a.strasse, s.stadt_name  
FROM Bewohner bw  
JOIN Adresse a ON bw.adresse_id = a.adresse_id  
JOIN Stadt s ON a.stadt_id = s.stadt_id;
```

5. Aktive Transportmissionen

```
SELECT tpw_id, tpw_status, dauer  
FROM Transportwege  
WHERE tpw_status = 'aktiv';
```

6. Transportmissionen mit Pilot und Fahrzeug

```
SELECT t(tpw_id, m.login AS pilot, f.f_name, t.dauer  
FROM Transportwege t  
JOIN Mitarbeiter m ON t.mitarbeiter_id = m.mitarbeiter_id  
JOIN Fahrzeuge f ON t.fahrzeug_id = f.fahrzeug_id;
```

7. Routen zwischen Mars-Sektoren

```
SELECT r.route_id, s1.stadt_name AS start, s2.stadt_name AS ziel  
FROM Route r  
JOIN Stadt s1 ON r.start_id = s1.stadt_id  
JOIN Stadt s2 ON r.end_id = s2.stadt_id;
```

8. Durchschnittliche Transportdauer

```
SELECT AVG(dauer) AS avg_transportdauer_stunden  
FROM Transportwege;
```

9. Lager und gespeicherte Ressourcen

```
SELECT l.lager_typ, r.t_typ, r.menge  
FROM ist_gelagert_in ig  
JOIN Lager l ON ig.lager_id = l.lager_id  
JOIN Ressource r ON ig.ressource_id = r.ressource_id;
```

10. Ressourcen mit höchster Priorität (Überlebenswichtig)

```
SELECT t_typ, menge  
FROM Ressource  
WHERE prioritaet = 1;
```

11. Ressourcen, die bald ablaufen

```
SELECT t_typ, ablaufdatum  
FROM Ressource  
WHERE ablaufdatum < CURRENT_DATE + INTERVAL '180 days';
```

12. Gesamtmenge aller Ressourcen

```
SELECT SUM(menge) AS gesamt_menge  
FROM Ressource;
```

13. Energieauslastung der Lager

```
SELECT lager_id,  
       eq_typ,  
       ROUND((aktuelle_leistung / kapazitaet_max) * 100, 2) AS auslastung_prozent  
FROM Energiequelle;
```

14. Lager ohne Energiequelle (kritisch)

```
SELECT l.lager_id, l.lager_typ  
FROM Lager l  
LEFT JOIN Energiequelle e ON l.lager_id = e.lager_id  
WHERE e.eg_id IS NULL;
```

15. Fahrzeuge im Einsatz

```
SELECT f_name  
FROM Fahrzeuge  
WHERE f_status = 'im Einsatz';
```

16. Welche Ressourcen werden aktuell transportiert?

```
SELECT f.f_name, r.t_typ  
FROM transportiert t  
JOIN Fahrzeuge f ON t.fahrzeug_id = f.fahrzeug_id  
JOIN Ressource r ON t.ressource_id = r.ressource_id;
```

17. Anzahl Transportmissionen pro Pilot

```
SELECT m.login, COUNT(t.tpw_id) AS missionen  
FROM Mitarbeiter m  
LEFT JOIN Transportwege t ON m.mitarbeiter_id = t.mitarbeiter_id  
GROUP BY m.login;
```

18. Städte ohne aktive Transportverbindung

```
SELECT s.stadt_name  
FROM Stadt s  
LEFT JOIN Transportwege t ON s.stadt_id = t.stadt_id  
WHERE t.tpw_id IS NULL;
```

19. Kritische Lager (Sauerstoff < 50%)

```
SELECT lager_id, aktuelle_leistung, kapazitaet_max  
FROM Energiequelle  
WHERE (aktuelle_leistung / kapazitaet_max) < 0.5;
```

20. Vollständiger Missionsbericht

```
SELECT  
    t.tpw_id,  
    m.login AS pilot,  
    f.f_name AS fahrzeug,  
    s.stadt_name AS sektor,  
    t.dauer,  
    t.tpw_status  
FROM Transportwege t  
JOIN Mitarbeiter m ON t.mitarbeiter_id = m.mitarbeiter_id  
JOIN Fahrzeuge f ON t.fahrzeug_id = f.fahrzeug_id  
JOIN Stadt s ON t.stadt_id = s.stadt_id;
```