

**1. Seznam všech dívek z tabulky KLIENT seříděný podle příjmení a jména.**

```
SELECT *  
FROM KLIENT  
WHERE pohlavi=2  
ORDER BY prijmeni, jmeno;
```

KLIENT (pohlavi=2)

**2. Seznam klientek, jejichž jméno je "Jana", seříděný podle příjmení.**

```
SELECT id_klient, id_skupiny, pohlavi, prijmeni, jmeno  
FROM KLIENT  
WHERE jmeno='Jana' ORDER BY prijmeni;
```

KLIENT (jmeno = 'Jana') [id\_klient, id\_skupiny, pohlavi prijmeni, jmeno]

**3. Seznam klientů, nikoli však dívek, seříděný podle příjmení a jména.**

```
SELECT *  
FROM KLIENT  
WHERE pohlavi <>2  
ORDER BY prijmeni, jmeno;
```

KLIENT (pohlavi != 2)

**4. Vyber všechny doprovodné pracovníky zařazené do funkce "lékař".**

```
SELECT FUNKCE.funkce, DOPROVOD.doprovod_prijmeni, DOPROVOD.doprovod_jmeno  
FROM FUNKCE INNER JOIN (  
DOPROVOD INNER JOIN (ZARAZENI) ON DOPROVOD.id_doprovod = ZARAZENI.id_doprovod) ON  
FUNKCE.id_funkce = ZARAZENI.id_funkce  
WHERE FUNKCE.funkce='lékař'  
ORDER BY DOPROVOD.doprovod_prijmeni, DOPROVOD.doprovod_jmeno;
```

```
{FUNKCE [FUNKCE.id_funkce = ZARAZENI.id_funkce] ZARAZENI [DOPROVOD.id_doprovod =  
ZARAZENI.id_doprovod] DOPROVOD}  
(FUNKCE.funkce='lékař') [FUNKCE.funkce, DOPROVOD.doprovod_prijmeni, DOPROVOD.doprovod_jmeno]
```

**5. Seznam klientů, kteří nejsou zařazeni v žádné skupině.**

```
SELECT KLIENT.prijmeni, KLIENT.jmeno, SKUPINA.cislo_skupiny  
FROM KLIENT LEFT JOIN SKUPINA ON KLIENT.id_skupiny=SKUPINA.id_skupiny  
WHERE SKUPINA.cislo_skupiny IS NULL  
ORDER BY prijmeni, jmeno;
```

```
{KLIENT [KLIENT.id_skupiny=SKUPINA.id_skupiny OR SKUPINA.cislo_skupiny IS NULL] SKUPINA}  
(SKUPINA.cislo_skupiny IS NULL) [KLIENT.prijmeni, KLIENT.jmeno, SKUPINA.cislo_skupiny]
```

**6. Seznam klientů zařazených do skupiny, kterou vede vedoucí Čermáková Kateřina.**

```
SELECT KLIENT.prijmeni, KLIENT.jmeno  
FROM (FUNKCE INNER JOIN (  
(SKUPINA INNER JOIN DOPROVOD ON SKUPINA.id_skupiny = DOPROVOD.id_skupiny)  
INNER JOIN (ZARAZENI) ON DOPROVOD.id_doprovod = ZARAZENI.id_doprovod)  
ON FUNKCE.id_funkce = ZARAZENI.id_funkce)  
INNER JOIN (KLIENT) ON SKUPINA.id_skupiny = KLIENT.id_skupiny  
WHERE ((FUNKCE.funkce='vedoucí skupiny') AND DOPROVOD.doprovod_prijmeni='Čermáková'  
AND DOPROVOD.doprovod_jmeno='Kateřina')  
ORDER BY KLIENT.prijmeni, KLIENT.jmeno;
```

```
{FUNKCE [FUNKCE.id_funkce = ZARAZENI.id_funkce] ZARAZENI [DOPROVOD.id_doprovod =  
ZARAZENI.id_doprovod] DOPROVOD [SKUPINA.id_skupiny = DOPROVOD.id_skupiny] SKUPINA  
[SKUPINA.id_skupiny = KLIENT.id_skupiny] KLIENT}  
(FUNKCE.funkce='vedoucí skupiny' AND DOPROVOD.doprovod_prijmeni='Čermáková' AND  
DOPROVOD.doprovod_jmeno='Kateřina') [KLIENT.prijmeni, KLIENT.jmeno]
```

**13. Seznam lékařů, kteří jsou zařazeni do turnusů lokality Chorvatsko.**

```
SELECT DOPROVOD.doprovod_prijmeni, DOPROVOD.doprovod_jmeno
FROM FUNKCE INNER JOIN ((
    ( LOKALITA INNER JOIN TURNUS ON LOKALITA.id_lokalita = TURNUS.id_lokalita)
    INNER JOIN DOPROVOD ON TURNUS.id_turnus = DOPROVOD.id_turnus)
    INNER JOIN ZARAZENI ON DOPROVOD.id_doprovod = ZARAZENI.id_doprovod)
    ON FUNKCE.id_funkce = ZARAZENI.id_funkce
WHERE LOKALITA.lokalita='C' AND FUNKCE.funkce='lékař'
ORDER BY DOPROVOD.doprovod_prijmeni, DOPROVOD.doprovod_jmeno;
```

**14. Klienti, kteří nemají vyplněný třetí požadavek na zařazení do turnusu.**

```
SELECT *
FROM KLIENT
WHERE NOT EXISTS (
    SELECT *
    FROM POZADAVKY
    WHERE KLIENT.id_klient = POZADAVKY.ID_KLIENT AND POZADAVKY.PRIORITA_POZADAVKU = 3)
ORDER BY KLIENT.prijmeni;
```

**15. Vybere klienty, jejichž id\_skupiny je rovno 1 a k nim přidá klienty jejichž id\_skupiny je rovno 8.**

```
SELECT KLIENT.prijmeni, KLIENT.jmeno, KLIENT.id_skupiny
FROM KLIENT
WHERE id_skupiny = 1
UNION
SELECT KLIENT.prijmeni, KLIENT.jmeno, KLIENT.id_skupiny
FROM KLIENT
WHERE id_skupiny = 8
ORDER BY id_skupiny;
```

**16. Od skupin jejichž ID je větší než 1 a menší než 50 odečte všechny skupiny, jejichž ID je větší než 30 a menší než 45.**

```
SELECT SKUPINA.id_skupiny, SKUPINA.cislo_skupiny
FROM SKUPINA
WHERE id_skupiny > 1 AND id_skupiny < 50
MINUS
SELECT SKUPINA.id_skupiny, SKUPINA.cislo_skupiny
FROM SKUPINA
WHERE id_skupiny > 30 AND id_skupiny < 45
ORDER BY id_skupiny;
```

**17. Vybere skupiny, které jsou v množině skupin, jejichž ID je menší než 45 a zároveň jsou v množině skupin, jejichž ID je větší než 35.**

```
SELECT SKUPINA.id_skupiny, SKUPINA.cislo_skupiny
FROM SKUPINA
WHERE id_skupiny < 45
INTERSECT
SELECT SKUPINA.id_skupiny, SKUPINA.cislo_skupiny
FROM SKUPINA
WHERE id_skupiny > 35;
```

**18. Přehled prioritních požadavků na zařazení do jednotlivých turnusů (v procentech).**

```
SELECT CISLO_TURNUSU, (POCET_POZADAVKU/CELKEM * 100) AS PROCENTO
FROM(
    SELECT TURNUS.cislo_turnusu, Count(POZADAVKY.id_klient) AS pocet_pozadavku, (
        SELECT Count(POZADAVKY.id_klient)
        FROM POZADAVKY
        WHERE POZADAVKY.priorita_pozadavku = 1 ) AS celkem
    FROM TURNUS INNER JOIN POZADAVKY ON TURNUS.id_turnus = POZADAVKY.id_turnus
    GROUP BY TURNUS.cislo_turnusu, POZADAVKY.priorita_pozadavku
    HAVING POZADAVKY.priorita_pozadavku=1)
ORDER BY CISLO_TURNUSU;
```

**7. Výpis všech skupin a klientů do nich zařazených.**

```
SELECT SKUPINA.cislo_skupiny, KLIENT.prijmeni, KLIENT.jmeno
FROM KLIENT NATURAL JOIN SKUPINA
ORDER BY cislo_skupiny, prijmeni, jmeno;
```

```
{SKUPINA*KLIENT}[SKUPINA.cislo_skupiny, KLIENT.prijmeni, KLIENT.jmeno]
```

**8. Kombinace všech turnusů se všemi lokalitami lokalitách.**

```
SELECT *
FROM LOKALITA CROSS JOIN TURNUS;
```

```
LOKALITA x TURNUS
```

**9. Seznam doprovodných pracovníků včetně jejich (ne)zařazení do turnusu.**

```
SELECT DOPROVOD.doprovod_prijmeni AS Prijmeni, DOPROVOD.doprovod_jmeno AS Jmeno,
       TURNUS.cislo_turnusu AS TURNUS
FROM TURNUS RIGHT JOIN DOPROVOD ON TURNUS.id_turnus = DOPROVOD.id_turnus
ORDER BY Prijmeni, Jmeno;
```

```
{TURNUS [TURNUS.id_turnus = DOPROVOD.id_turnus OR TURNUS.id_turnus = NULL] DOPROVOD}
 [DOPROVOD.doprovod_prijmeni, DOPROVOD.doprovod_jmeno, TURNUS.cislo_turnusu]
```

**10. Výpis všech doprovodných pracovníků a jejich požadavků na zařazení do funkce.**

```
SELECT FUNKCE.funkce AS FUNKCE, DOPROVOD.doprovod_jmeno AS JMENO,
       DOPROVOD.doprovod_prijmeni AS PRIJMENI
FROM DOPROVOD FULL JOIN (
    FUNKCE FULL JOIN ZARAZENI ON FUNKCE.id_funkce = ZARAZENI.id_funkce)
    ON DOPROVOD.id_doprovod = ZARAZENI.id_doprovod
ORDER BY PRIJMENI, JMENO;
```

```
{DOPROVOD [DOPROVOD.id_doprovod = ZARAZENI.id_doprovod] ZARAZENI [FUNKCE.id_funkce =
    ZARAZENI.id_funkce] FUNKCE}
```

```
[FUNKCE.funkce, DOPROVOD.doprovod_jmeno, DOPROVOD.doprovod_prijmeni]
```

**11. Seznam klientů zařazených do skupiny, jejíž vedoucí skupiny je Čermáková Kat - varianta dotazu z bodu 6., nyní s poddotazem v klauzuli WHERE.**

```
SELECT KLIENT.jmeno, KLIENT.prijmeni
FROM (SKUPINA INNER JOIN DOPROVOD ON SKUPINA.id_skupiny = DOPROVOD.id_skupiny)
    INNER JOIN KLIENT ON SKUPINA.id_skupiny = KLIENT.id_skupiny WHERE id_doprovod
    IN(
        SELECT DOPROVOD.id_doprovod
        FROM DOPROVOD
        WHERE DOPROVOD.doprovod_prijmeni='Čermáková' AND DOPROVOD.doprovod_jmeno='Kat')
ORDER BY KLIENT.prijmeni;
```

**12. Vybere skupiny s největším počtem klientů.**

```
SELECT SKUPINA_CISLO, MAX_POCET
FROM (
    SELECT SKUPINA.cislo_skupiny AS SKUPINA1, Count (KLIENT.id_klient) AS POCET_KLIENTU
    FROM SKUPINA INNER JOIN KLIENT ON SKUPINA.id_skupiny = KLIENT.id_skupiny
    GROUP BY SKUPINA.cislo_skupiny)
INNER JOIN (SELECT Max(POCET_KLIENTU) AS MAX_POCET
    FROM (
        SELECT SKUPINA.cislo_skupiny AS SKUPINA1, Count(KLIENT.id_klient) AS POCET_KLIENTU
        FROM SKUPINA INNER JOIN KLIENT ON SKUPINA.id_skupiny = KLIENT.id_skupiny
        GROUP BY SKUPINA.cislo_skupiny)
    ) ON POCET_KLIENTU = MAX_POCET;
```

**13. Seznam skupin, kde je aktuálně zařazeno méně než 5 klientů.**

```
SELECT SKUPINA.cislo_skupiny, Count(KLIENT.id_klient) AS pocet_klientu
FROM SKUPINA INNER JOIN KLIENT ON SKUPINA.id_skupiny = KLIENT.id_skupiny
GROUP BY SKUPINA.cislo_skupiny
HAVING Count(KLIENT.id_klient) < 5;
```

**19. Vytvoření "pohledu" KLIENTI\_VE\_SKUPINACH.**

```
CREATE VIEW KLIENTI_VE_SKUPINACH AS
SELECT SKUPINA.cislo_skupiny, KLIENT.jmeno, KLIENT.prijmeni
FROM SKUPINA INNER JOIN KLIENT ON SKUPINA.id_skupiny = KLIENT.id_skupiny
ORDER BY SKUPINA.cislo_skupiny, KLIENT.prijmeni;
```

**20. Z pohledu KLIENTI\_VE\_SKUPINACH vyberte klienty ze skupiny 1405.**

```
SELECT *
FROM KLIENTI_VE_SKUPINACH
WHERE KLIENTI_VE_SKUPINACH.cislo_skupiny = 1405;
```

KLIENTI\_VE\_SKUPINACH (cislo\_skupiny = 1405)

**21. Vytvoření tabulky TURNUSY\_NA\_LOKALITACH a naplnění tabulky daty.**

```
DROP TABLE TURNUS_NA_LOKALITE;
```

```
CREATE TABLE TURNUS_NA_LOKALITE( cislo_turnusu INTEGER, lokalita VarChar2(255));
```

```
INSERT INTO TURNUS_NA_LOKALITE (cislo_turnusu, lokalita)
```

```
SELECT TURNUS.cislo_turnusu, LOKALITA.lokalita
FROM LOKALITA INNER JOIN TURNUS ON LOKALITA.id_lokalita = TURNUS.id_lokalita;
```

**22. V položce TURNUS\_NA\_LOKALITE.lokalita zaměnit "Chorvatsko" za 'C' a Řecko za 'R'.**

```
UPDATE TURNUS_NA_LOKALITE SET TURNUS_NA_LOKALITE.lokalita = 'Řecko' WHERE lokalita = 'R';
UPDATE TURNUS_NA_LOKALITE SET TURNUS_NA_LOKALITE.lokalita = 'Chorvatsko' WHERE lokalita = 'C';
```

**23. Z tabulky TURNUS\_NA\_LOKALITE smazat záznamy o Chorvatsku.**

```
DELETE
FROM TURNUS_NA_LOKALITE WHERE TURNUS_NA_LOKALITE.LOKALITA = 'Chorvatsko';
```