SELECT IME, PREZIME, DATUM\_RODENJA, RESPIRATOR, KONTAKT\_BLISKE\_OSOBE, USTANOVA, OSOBA.OIB, ADRESA\_STANOVANJA

FROM PACIJENT, OSOBA

WHERE PACIJENT.OIB = OSOBA.OIB AND DATE\_ADD(PACIJENT.DATUM\_RODENJA, INTERVAL 18 YEAR) > CURDATE()

ORDER BY DATUM\_RODENJA DESC

Neke informacija (ime, prezime, datum rođenja, podatak o stavljanju na respirator, kontakt bliske osobe, naziv ustanove gdje je osoba smještena, OIB, adresa stanovanja) maloljetnih pacijenata, uzlazno sortirano po dobi pacijenata.

Formira se pomoćna tablica T koja je Kartezijev produkt tablica PACIJENT i OSOBA.  
Potom se kreira nova pomoćna tablica U u koju dodajemo točno one retke tablice T koji ispunjavaju uvjete:  
vrijednost atributa OIB iz faktora PACIJENT jednaka je vrijednosti atributa OIB iz faktora OSOBA,  
vrijednost atributa DATUM\_RODENJA iz faktora PACIJENT uvećana pomoću funkcije DATE\_ADD() za 18 godina veća je od vrijednosti funkcije CURDATE() koja vraća trenutni datum.  
Izlaznu tablicu čini 8 stupaca (IME, PREZIME, DATUM\_RODENJA, RESPIRATOR, KONTAKT\_BLISKE\_OSOBE, USTANOVA, OSOBA.OIB, ADRESA\_STANOVANJA), svaki redak odgovara točno jednom retku tablice U, a podaci iz danog retka tablice U su ispisani u poretku koji odgovara nazivima stupaca izlazne tablice.  
Izlaznu tablicu sortiramo silazno po vrijednosti DATUM\_RODENJA.

--------------------------------------------------------------------------------------------------------------------------------------

SELECT  
CONCAT(ZARAZEN.IME, " ", ZARAZEN.PREZIME) ZARAZEN,  
GROUP\_CONCAT(UGROZEN.IME," ", UGROZEN.PREZIME SEPARATOR ', ') UGROZIO

FROM OSOBA ZARAZEN, OSOBA UGROZEN, U\_KONTAKTU

WHERE ZARAZEN.OIB = U\_KONTAKTU.OIB\_PACIJENTA AND UGROZEN.OIB = U\_KONTAKTU.OIB\_KONTAKTA

GROUP BY ZARAZEN.OIB  
ORDER BY COUNT(\*) DESC

Popis svih zaraženih osoba koje su imale bliske kontakte, zajedno sa listom tih kontakata, sortirano po broju kontakata koje je pacijent ostvario.

Formira se pomoćna tablica T koja je Kartezijev produkt dviju tablica OSOBA(prvi od ta dva faktora dobiva nadimak ZARAZEN, a drugi UGROZEN) te tablice U\_KONTAKTU.  
Zatim kreiramo novu pomoćnu tablicu U u koju dodajemo točno one retke tablice T koji ispunjavaju uvjete:  
vrijednost atributa OIB iz faktora ZARAZEN jednaka je vrijednosti atributa OIB\_PACIJENTA iz faktora U\_KONTAKTU, vrijednost atributa OIB iz faktora UGROZEN jednaka je vrijednosti OIB\_KONTAKTA iz faktora U\_KONTAKTU.  
Potom u tablici T grupiramo retke u skupine s obzirom na međusobno jednaku vrijednost atributa OIB iz faktora ZARAZEN.  
Izlazna tablica s dva stupca, ZARAZEN i UGROZIO, formira se na sljedeći način. Za svaku spomenutu skupinu u tablici T dodajemo po jedan redak u izlaznu tablicu.  
Vrijednost ZARAZEN tog retka jednaka je konkatenaciji sljedećih vrijednosti: vrijednost atributa IME iz faktora ZARAZEN, string " ", vrijednost atributa PREZIME iz faktora ZARAZEN (vrijednosti atributa IME i PREZIME iz faktora ZARAZEN potpuno funkcionalno ovise o vrijednosti atributa OIB iz faktora ZARAZEN pa u zato iste za sve retke u skupini).  
Vrijednost UGROZIO tog retka dobivamo kontkatenacijom po svim retcima dane grupe, a iz svakog pojedinog retka konkateniramo sljedeće vrijednosti: vrijednost atributa IME iz faktora UGROZEN, string " ", vrijednost atributa PREZIME iz faktora UGROZEN. Vrijednosti koje se odnose na isti redak početne tablice od ostalih su vrijednosti odvojene zarezom i dva razmaka.  
Izlaznu tablicu sortiramo silazno po broju redaka u odgovarajućoj skupini tablice T.

--------------------------------------------------------------------------------------------------------------------------------------

SELECT  
NAZIV\_USTANOVE "NAZIV USTANOVE",  
ROUND( COUNT(DISTINCT PACIJENT.OIB)/COUNT(DISTINCT LIJECNIK.OIB), 2 ) "PACIJENT/LIJECNIK",  
ROUND( COUNT(DISTINCT PACIJENT.OIB)/BR\_KREVETA\*100, 2 ) "ZAUZETOST KREVETA (%)",  
ROUND( SUM(RESPIRATOR = '1')/COUNT(DISTINCT LIJECNIK.OIB)/BR\_RESPIRATORA\*100, 2 ) "ZAUZETOST RESPIRATORA (%)"

FROM (USTANOVA INNER JOIN LIJECNIK ON USTANOVA.NAZIV\_USTANOVE = LIJECNIK.USTANOVA) INNER JOIN PACIJENT ON USTANOVA.NAZIV\_USTANOVE = PACIJENT.USTANOVA

WHERE LIJECNIK.STATUS IN (1, 2)

GROUP BY(NAZIV\_USTANOVE)  
ORDER BY BR\_KREVETA

Popis zdravstvenih ustanova u kojima se liječe kovid pozitivni pacijenti te opterećenje dotičnih ustanova izraženo u: broju pacijenata na jednog (radno aktivnog) liječnika, postotku bolničkih kreveta koje zauzimaju kovid pacijenti i postotku respiratora koje oni zauzimaju.

Formira se pomoćna tablica T koja je prirodni spoj tablica USTANOVA i LIJECNIK (tj. u tablicu T se dodaju retci nastali sljepljivanjem svih parova redaka (s tim da je pod-redak iz tablice USTANOVA uvijek prvi) iz tablica USTANOVA i LIJECNIK kojima su vrijednosti atributa NAZIV\_USTANOVE i USTANOVA jednake).  
Zatim se kreira nova pomoćna tablica U koja je prirodni spoj tablica T i PACIJENT (tj. u tablicu U se dodaju retci nastali sljepljivanjem svih parova redaka (s tim da je pod-redak iz tablice T uvijek prvi) iz tablica T i PACIJENT kojima su vrijednosti atributa NAZIV\_USTANOVE i USTANOVA jednake).  
Potom se kreira nova pomoćna tablica V u koju dodajemo točno one retke tablice U kojima je vrijednost atributa STATUS jednaka 1 ili 2.  
Na kraju u tablici V grupiramo retke u skupine s obzirom na međusobno jednaku vrijednost atributa NAZIV\_USTANOVE.  
Izlazna tablica s 4 stupca, NAZIV USTANOVE, PACIJENT/LIJECNIK, ZAUZETOST KREVETA (%), ZAUZETOST RESPIRATORA (%), formira se na sljedeći način. Za svaku spomenutu skupinu u tablici T dodajemo po jedan redak u izlaznu tablicu.  
Vrijednost 1. stupca tog retka jednaka je vrijednosti atributa NAZIV\_USTANOVE za skupinu.  
Vrijednost 2. stupca tog retka jednaka je broju različitih vrijednosti atributa PACIJENT.OIB u skupini podijeljenom sa brojem različitih vrijednosti atributa LIJECNIK.OIB u skupini te zaokruženom na 2 decimale.  
Vrijednost 3. stupca tog retka jednaka je broju različitih vrijednosti atributa PACIJENT.OIB u skupini podijeljenom sa vrijednosti atributa BR\_KREVETA u skupini (potpuno funkcionalno ovisna o NAZIV\_USTANOVE pa zato i ista za sve retke u skupini), pomnožena sa 100 kako bi dobivena vrijednost bila izražena u postocima te zaokruženom na 2 decimale.  
Vrijednost 4. stupca tog retka jednaka je sumi svih vrijednosti atributa RESPIRATOR u skupini, podijeljenoj s brojem različitih vrijednosti atributa LIJECNIK.OIB u skupini, podijeljena s vrijednošću atributa BR\_RESPIRATORA u skupini (potpuno funkcionalno ovisna o NAZIV\_USTANOVE pa zato i ista za sve retke u skupini), pomnožena sa 100 kako bi dobivena vrijednost bila izražena u postocima te zaokruženom na 2 decimale.  
Izlaznu tablicu sortiramo uzlazno po vrijednosti BR\_KREVETA.

--------------------------------------------------------------------------------------------------------------------------------------

SELECT   
IME,  
REPEAT("\*\*\*", EXISTS(SELECT \* FROM LIJECNIK WHERE OSOBA.OIB = LIJECNIK.OIB)) LIJECNIK,  
REPEAT("\*\*\*", EXISTS (SELECT \* FROM PACIJENT WHERE OSOBA.OIB = PACIJENT .OIB)) PACIJENT,  
REPEAT("\*\*\*", EXISTS (SELECT \* FROM BLISKI\_KONTAKT WHERE OSOBA.OIB = BLISKI\_KONTAKT.OIB)) BLISKI\_KONTAKT

FROM OSOBA WHERE PREZIME = "Weasley"

ORDER BY IME

Tablica koja zorno predočava koji se član obitelji “Weasley” našao u ulozi doktora, koji pacijenta, a koji bliskog kontakta zaražene osobe.

Prvo se kreira nova pomoćna tablica U u koju dodajemo točno one retke tablice OSOBA u kojima je vrijednost atributa PREZIME jednaka "Weasley".  
Izlazna tablica s 4 stupca, IME, LIJECNIK, PACIJENT, BLISKI\_KONTAKT, formira se na sljedeći način. Za svaki redak tablice U dodajemo po jedan novi redak u izlaznu tablicu.  
Vrijednost IME u novom retku je vrijednost IME u retku tablice U.  
Vrijednost LIJECNIK u novom retku R kreiramo n-strukim nadovezivanjem stringa “\*\*\*” samog na sebe, pri ćemu je n broj koji vraća logički uvjet EXISTS(SELECT \* FROM LIJECNIK WHERE OSOBA.OIB = LIJECNIK.OIB). Provjeru uvjeta radimo gradnjom pomoćne tablice P. Tablica P sastoji se od redaka tablice LIJECNIK čija je vrijednost atributa OIB jednaka vrijednosti atributa OIB retka R. Ako pomoćna tablica P sadrži barem jedan redak, n će biti jednak 1, dok će inače biti jednak 0.  
Vrijednost PACIJENT u novom retku R kreiramo n-strukim nadovezivanjem stringa “\*\*\*” samog na sebe, pri ćemu je n broj koji vraća logički uvjet EXISTS(SELECT \* FROM PACIJENT WHERE OSOBA.OIB = PACIJENT.OIB). Provjeru uvjeta radimo gradnjom pomoćne tablice P. Tablica P sastoji se od redaka tablice PACIJENT čija je vrijednost atributa OIB jednaka vrijednosti atributa OIB retka R. Ako pomoćna tablica P sadrži barem jedan redak, n će biti jednak 1, dok će inače biti jednak 0.  
Vrijednost BLISKI\_KONTAKT u novom retku R kreiramo n-strukim nadovezivanjem stringa “\*\*\*” samog na sebe, pri ćemu je n broj koji vraća logički uvjet EXISTS(SELECT \* FROM BLISKI\_KONTAKT WHERE OSOBA.OIB = BLISKI\_KONTAKT.OIB). Provjeru uvjeta radimo gradnjom pomoćne tablice P. Tablica P sastoji se od redaka tablice BLISKI\_KONTAKT čija je vrijednost atributa OIB jednaka vrijednosti atributa OIB retka R. Ako pomoćna tablica P sadrži barem jedan redak, n će biti jednak 1, dok će inače biti jendak 0.  
Izlaznu tablicu sortiramo uzlazno po vrijednosti IME.