SQL příklady

1

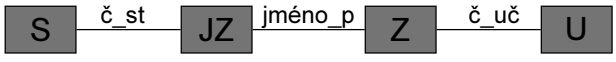
Schéma databáze:

Student(č_st: integer, jméno_st: string, obor: string, ročník: string, věk: integer)

Předmět(jméno_p: string, koná_se: time, místnost: string, č_uč: integer)

Je_zapsán(č_st: integer, jméno_p: string)

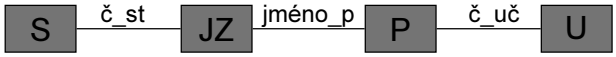
Učitel(č_uč: integer, jméno_u: string, katedra: integer)



2

Najdi jména všech prváků (ročník = ''1''), kteří jsou zapsáni na nějaký předmět, který učí ''J. Král''.

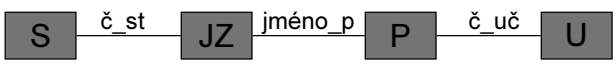
```
SELECT DISTINCT S.jméno_st
FROM Student S, Předmět P, Je_zapsán JZ, Učitel U
WHERE S.č_st = JZ.č_st AND JZ.jméno_p =
   P.jméno_p
   AND P.č_uč = U.č_uč AND
   U.jméno_u = ''J. Král'' AND S.ročník = ''1''
```



3

Najdi věk nejstarších studentů, kteří jsou buď na oboru ''Informatika' nebo jsou zapsáni v nějakém kurzu, který učí ''J. Král''.

```
SELECT MAX(S.věk)
FROM Student S
WHERE (S.obor = ''Informatika'')
   OR S.č_st IN (SELECT JZ.č_st
        FROM Předmět P, Je_zapsán JZ, Učitel U
        WHERE JZ.jméno_p = P.jméno_p AND P.č_uč =
              U.č_uč AND U.jméno_u = ''J. Král'' )
```



4

Najdi jména všech předmětů, které se buď učí v místnosti ''S4'' nebo je na ně zapsáno 5 či více studentů.

```
SELECT P.jméno_p
FROM Předmět P
WHERE P.místnost = ''S4''
   OR P.jméno_p IN
        (SELECT JZ.jméno_p
        FROM Je_zapsán JZ
        GROUP BY JZ.jméno_p
        HAVING COUNT (*) >= 5)
```

5

Najdi jména všech studentů, kteří jsou zapsáni na dva předměty konající se ve stejnou dobu.

```
SELECT DISTINCT S.jméno_st
FROM Student S
WHERE S.č_st IN
   (SELECT JZ1.č_st
   FROM Je_zapsán JZ1, Je_zapsán JZ2, Předmět P1,
                Předmět P2
   WHERE JZ1.č_st = JZ2.č_st AND JZ1.jméno_p !=
        JZ2.jméno_p AND JZ1.jméno_p = P1.jméno_p
        AND JZ2.jméno_p = P2.jméno_p AND P1.koná_se
        = P2.koná_se)
```

6

Najdi jména všech učitelů, kteří učí v každé místnosti, ve které se učí nějaký předmět.

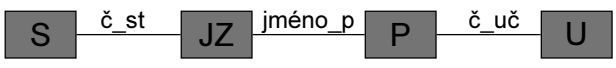
```
SELECT DISTINCT U.jméno_u
FROM Učitel U
WHERE NOT EXISTS (( SELECT P.místnost
                    FROM Předmět P )
                    EXCEPT
                    (SELECT P1.místnost
                    FROM Předmět P1
                    WHERE P1.č_uč = U.č_uč ))
```

Pz.: EXCEPT je použito za WHERE!

7

Najdi jména všech učitelů, pro něž je počet jejich kurzů, na které se někdo zapsal, menší než 5.

```
SELECT DISTINCT U.jméno_u
FROM Učitel U
WHERE 5 > (SELECT COUNT(*)
           FROM Předmět P, Je_zapsán JZ
           WHERE P.jméno_p = JZ.jméno_p
           AND P.č_uč = U.č_uč)
```



8

Vypiš ročník a průměrný věk studentů pro tento ročník (pro všechny ročníky).

```
SELECT S.ročník, AVG(S.věk)
FROM Student S
GROUP BY S.ročník
```

jednoduché

9

Vypiš ročník a průměrný věk studentů pro tento ročník (pro všechny ročníky kromě prvního).

```
SELECT S.ročník, AVG(S.věk)
FROM Student S
WHERE S.ročník != "1"
GROUP BY S.ročník
```

jednoduché

10

Najdi jména studentů, kteří jsou zapsáni v maximálním počtu předmětů.

```
SELECT DISTINCT S.jméno_st
FROM Student S
WHERE S.č_st IN (SELECT JZ.č_st
                 FROM Je_zapsán JZ
                 GROUP BY JZ.č_st
                 HAVING COUNT (*) >= ALL
                     (SELECT COUNT (*)
                     FROM Je_zapsán JZ2
                     GROUP BY JZ2.č_st ))
```

11

Najdi jména studentů, kteří nejsou zapsáni v žádném předmětu.

```
SELECT DISTINCT S.jméno_st
FROM Student S
WHERE S.č_st NOT IN (SELECT JZ.č_st
                     FROM Je_zapsán JZ )
```

jednoduché

12

Uvažujte následující schéma databáze:

Dodavatel(č_d: integer, jméno_d: string, adresa: string)
Zboží(č_zb: integer, jméno_zb: string, barva: string)
Katalog(č_d: integer, č_zb: integer, cena: real)



13

Najdi jména zboží dodávané firmou „Tomil“ a nikým jiným.

```
SELECT Z.jméno_zb
FROM Zboží Z, Katalog K, Dodavatel D
WHERE Z.č_zb = K.č_zb AND K.č_d = D.č_d
AND D.jméno_st = ‘Tomil'
AND NOT EXISTS ( SELECT *
                 FROM Katalog K1, Dodavatel D1
                 WHERE Z.č_zb = K1.č_zb AND K1.č_d =
                       D1.č_d AND D1.jméno_d != ‘Tomil')
```



14

Najdi čísla těch dodavatelů, kteří chtějí za nějaké zboží více, než je průměrná cena tohoto zboží (průměrná cena se bere přes všechny dodavatele, kteří toto zboží dodávají).

```
SELECT DISTINCT K.č_d
FROM Katalog K
WHERE K.cena > ( SELECT AVG (C1.cena)
                 FROM Katalog C1
                 WHERE C1.č_zb = K.č_zb )
```

15

Pro každé zboží najdi dodavatele, který je nabízí za nevyšší cenu.

```
SELECT Z.č_zb, D.jméno_d
FROM Zboží Z, Dodavatel D, Katalog K
WHERE K.č_zb = Z.č_zb AND K.č_d = D.č_d
   AND K.cena = (SELECT MAX (K1.cena)
                 FROM Katalog K1
                 WHERE K1.č_zb = Z.č_zb)
```

16

Najdi čísla těch dodavatelů, kteří dodávají pouze červené zboží.

```
SELECT DISTINCT K.č_d
FROM Katalog K
WHERE NOT EXISTS ( SELECT *
                   FROM Zboží Z
                   WHERE Z.č_zb = K.č_zb AND
                         Z.barva != ‘červené' )
```

17

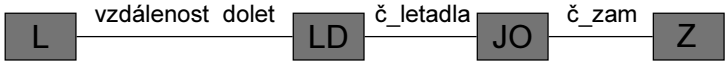
Uvažujte následující schéma letištní databáze:

Let(č_letu: integer, z: string, do: string, vzdálenost: integer, odlet: time, přílet: time, cena: integer)

Letadlo(č_letadla: integer, jméno_l: string, dolet: integer)

Je_oprávněn_pro(č_zam: integer, č_letadla: integer))

Zaměstnanec(č_zam: integer, jméno_z: string, mzda: integer)

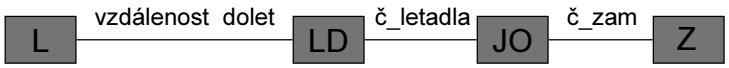


- Relace Zaměstnanec popisuje piloty a rovněž i další druhy zaměstnanců; každý pilot má oprávnění (relace Je-oprávněn) pro nějaké letadlo (další zaměstnanci nemusí být kvalifikování pro lety).

18

Najdi jména letadel takových, že všichni piloti, kteří mají oprávnění na nich létat, vydělávají více než 80,000.

```
SELECT DISTINCT LD.jméno_l
FROM Letadlo LD
WHERE LD.č_letadla IN (SELECT JO.č_letadla
   FROM Je_oprávněn_pro JO, Zaměstnanec Z
   WHERE JO.č_zam = Z.č_zam AND
   NOT EXISTS ( SELECT *
        FROM Zaměstnanec E1
        WHERE E1.č_zam = Z.č_zam AND E1.mzda <=
        80000 ))
```



19

Pro každého pilota, který má oprávnění na více než 3 letadla, najdi č_zam a maximální dolet z letadel, pro které má oprávnění.

```
SELECT JO.č_zam, MAX (LD.dolet)
FROM Je_oprávněn_pro JO, Letadlo LD
WHERE JO.č_letadla = LD.č_letadla
GROUP BY JO.č_zam
HAVING COUNT (*) > 3
```

20

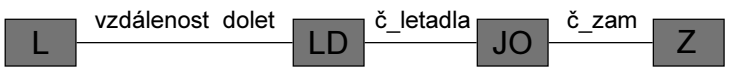
Najdi jména pilotů, jejichž mzda je menší než cena nejlevnějšího letu z Prahy do Honolulu.

```
SELECT DISTINCT Z.jméno_z
FROM Zaměstnanec Z
WHERE Z.mzda < ( SELECT MIN (L.cena)
                 FROM Let L
                 WHERE L.z = 'LA' AND L.do =
                       'Honolulu' )
```

21

Pro všechna letadla s doletem větším než 1000 najdi jejich typ a průměrnou mzdu všech pilotů, kteří mají pro toto letadlo oprávnění

```
SELECT LD.jméno_l AS typ, AVG (Z.mzda) AS Průměr
FROM Letadlo LD, Je_oprávněn_pro JO, Zaměstnanec
  Z
WHERE LD.č_letadla = JO.č_letadla AND
      JO.č_zam = Z.č_zam AND LD.dolet > 1000
GROUP BY LD.č_letadla, LD.jméno_l
```



22

Zákazník chce cestovat z Moskvy do Říma ne více než se dvěma přestupy. Najdi vyhovující čísla letů spolu s časem odletu, jestliže chce být v Římě do 18.00.

```
SELECT L0.č_letu, L0.odlet
FROM Let L0
WHERE L0.z = 'Moskva' AND L0.do = 'Řím' AND AND L0.přílet < 18.00 )
UNION
( SELECT L0.č_letu, L0.odlet
FROM Let L0, Let L1
WHERE L0.z = 'Moskva' AND L0.do != 'Řím' AND AND L0.do = L1.z AND
    L1.do = 'Řím' L1.odlet > L0.přílet AND L1.přílet < 18.00 )
UNION
( SELECT L0.č_letu, L0.odlet
FROM Let L0, Let L1, Let L2
WHERE L0.z = 'Moskva' AND L0.do = L1.z AND L1.do = L2.z
AND L2.do = 'Řím' AND L0.do != 'Řím' AND L1.do != 'Řím'
AND L1.odlet > L0.přílet AND L2.odlet > L1.přílet
AND L2.přílet < 18.00 ))
```

23

Spočti rozdíl mezi průměrnou mzdou pilotů a průměrnou mzdou všech zaměstnanců (včetně pilotů).

```
SELECT Temp1.avg - Temp2.avg
FROM ( SELECT AVG(Z.mzda) AS avg
        FROM Zaměstnanec Z
        WHERE Z.č_zam IN (SELECT DISTINCT
  JO.č_zam
                FROM Je_oprávněn_pro JO )) AS Temp1,
        ( SELECT AVG (E1.mzda) AS avg
        FROM Zaměstnanec E1 ) AS Temp2 )
```

24

Vytiskni jméno a mzdu každého ne-pilota, jehož mzda je větší než průměrná mzda pilotů.

```
SELECT Z.jméno_z, Z.mzda
FROM Zaměstnanec Z
WHERE Z.č_zam NOT IN (        SELECT DISTINCT
   JO.č_zam
                              FROM Je_oprávněn_pro JO )
   AND Z.mzda > ( SELECT AVG (Z1.mzda)
                  FROM Zaměstnanec Z1
                  WHERE Z1.č_zam IN
                        ( SELECT DISTINCT
   JO1.č_zam
                        FROM Je_oprávněn_pro JO1 ))
```

25

Uvažujte následující schéma databáze

Zam(č_zam: integer, jméno_z: string, věk: integer, mzda: real)

Pracuje(č_zam: integer, jméno_o: string, úvazek: integer)

Odd(jméno_o: string, rozpočet: real, č_ved: integer)

- Zaměstnanec může pracovat ve více než jednom oddělení; atribut úvazek v relaci Pracuje ukazuje procento, kterým zaměstnanec pracuje v daném oddělení.

26

Najdi jméno a věk každého zaměstnance, který pracuje jak v oddělení Hardware tak v oddělení Software.

```
SELECT Z.jméno_z, Z.věk
FROM Zam Z, Pracuje P1, Pracuje P2
WHERE Z.č_zam = P1.č_zam AND P1.jméno_o =
   'HARDWARE' AND
Z.č_zam = P2.č_zam AND P2.jméno_o = 'SOFTWARE'
```

27

Pro každé oddělení s více než 20 v součtu plnými úvazky (tj., součty částečných a plných úvazků dávají více než 20 úplných úvazků) vytiskni jména oddělení spolu s počtem zaměstnanců, kteří v nich pracují.

```
SELECT P.jméno_o, COUNT (P.č_zam)
FROM Pracuje P
GROUP BY P.jméno_o
HAVING 2000 < ( SELECT SUM (P1.úvazek)
                FROM Pracuje P1
                WHERE P1.jméno_o = P.jméno_o )
```

28

Vytiskni jména zaměstnanců, jejichž mzda překračuje rozpočet všech těch oddělení, ve kterých pracují.

```
SELECT Z.jméno_z
FROM Zam Z
WHERE Z.mzda > ALL (SELECT O.rozpočet
FROM Odd O, Pracuje P
WHERE Z.č_zam = P.č_zam AND O.jméno_o =
   P.jméno_o)
```

29

Najdi čísla vedoucích, kteří řídí pouze oddělení s rozpočtem větším než 1 000 000.

```
SELECT DISTINCT O.č_ved
FROM Odd O
WHERE 1000000 < ALL (SELECT O2.rozpočet
                     FROM Odd O2
                     WHERE O2.č_ved =
   O.č_ved                )
```

30

Najdi jména vedoucích, kteří řídí oddělení s největším rozpočtem.

```
SELECT Z.jméno_z
FROM Zam Z
WHERE Z.č_zam IN (SELECT O.č_ved
                  FROM Odd O
                  WHERE O.rozpočet =
                        (SELECT MAX
  (O2.rozpočet)
                        FROM Odd O2))
```

31

Jestliže vedoucí řídí více než jedno oddělení, říká se o něm, že kontrolují součet rozpočtů všech těchto oddělení. Najdi čísla vedoucích kteří kontrolují více než 5 000 000.

```
SELECT O.č_ved
FROM Odd O
WHERE 5000000 < (SELECT SUM (O2.rozpočet)
                 FROM Odd O2
                 WHERE O2.č_ved =
  O.č_ved        )
```

32

Najdi č_ved těch vedoucích, kteří kontrolují největší balík.

```
SELECT DISTINCT tempO.č_ved
FROM (SELECT DISTINCT O.č_ved, SUM (O.rozpočet)
            AS tempR
      FROM Odd O
      GROUP BY O.č_ved ) AS tempO
WHERE tempO.tempR =
      (SELECT MAX (tempO.tempR)
      FROM tempO)
```

33

Uvažujte relaci Námořník.

č_nám	j_nám	hodnost	věk
18	Janák	3	30.0
41	Jůza	6	56.0
22	Aral	7	44.0
63	Majer	null	15.0

34

Zapiš v SQL dotazy pro průměrnou hodnost použitím agregační funkce AVG, součet všech hodností použitím funkce SUM a celkový počet hodností použitím COUNT.

```
SELECT AVG (N.hodnost) AS AVERAGE
FROM Námořník N

SELECT SUM (N.hodnost)
FROM Námořník N

SELECT COUNT (N.hodnost)
FROM Námořník N
```

35

Vydělíte-li výše spočítaný součet počtem, obdržíte stejný výsledek jako vypočítaný průměr? Jak by se změnila odpověď na předchozí otázku, kdyby se místo atributu hodnost uvažoval atribut věk?

- Výsledek s použitím SUM a COUNT by byl menší než výsledek s použitím AVERAGE, jestliže existují n-tice, hodnost = NULL. To je proto, že všechny agregační funkce kromě COUNT ignorují prázdné hodnoty. Tedy v prvním přístupu by výsledek odpovídal výpočtu (3+6+7)/3, zatímco v druhém přístupu (3+6+7)/4.
- Kdyby se agregace dělaly přes atribut věk, odpovědi by v obou přístupech byly stejné, protože sloupec věk neobsahuje prázdné hodnoty.

36

Najdi jména námořníků, kteří mají vyšší hodnost než všichni námořníci s věkem < 21.

```
SELECT N.j_nám
FROM Námořník N
WHERE NOT EXISTS ( SELECT *
                   FROM Námořník N2
                   WHERE N2.věk < 21
                         AND N.hodnost <= N2.hodnost )
SELECT *
FROM Námořník N
WHERE N.hodnost > ALL ( SELECT N2.hodnost
                        FROM Námořník N2
                        WHERE N2.věk < 21 )
```


Embedding SQL in C: An Example

```
char SQLSTATE[6];
EXEC SQL BEGIN DECLARE SECTION
char c_j_nám[20]; short c_minhodnost; float c_věk;
EXEC SQL END DECLARE SECTION
c_minhodnost = random();
EXEC SQL DECLARE sinfo CURSOR FOR
   SELECT S.j_nám, S.věk FROM Námořník N
   WHERE S.hodnost > :c_minhodnost;
do {
   EXEC SQL FETCH sinfo INTO :c_j_nám, :c_věk;
   printf(``%s is %d years old\n'', c_j_nám, c_věk);
} while (SQLSTATE != `02000');
EXEC SQL CLOSE sinfo;
```