

[Skrýt menu](#)[Semestrální práce](#)[Editor semestrální práce](#)[Kontrola a odevzdání](#)[Podmínky klasifikace](#)**Navigace dotazů**

- D1 - Typ vleku na sjezdovkách
- D2 - Kolik instruktorů pracuje
- D3 - ID masérů, kteří dosud n
- D4 - Seznam všech zaměstn
- D5 - Jméno, příjmení a věk v
- D6 - Jména vlekařů, kteří obs
- D7 - Jména zaměstnanců, kte
- D8 - Objednávky ze dne 20.0
- D9 - Objednávky, které obsa
- D10 - Telefony sester Cockne
- D11 - Id_objednávky, datum,
- D12 - Kontrola D1 - masážov
- D13 - ID objednávky, která ob
- D14 - Zaměstnanec, který př
- D15 - Všechny možné kombin
- D16 - Vyber zákazníky a všecl
- D17 - Telefony zákazníků - m
- D18 - Vyberte objednávky, kt
- D19 - Do objednávky ID 3 pří
- D20 - Vytvořit pohled instruk
- D21 - Protože práce v půjčov
- D22 - Stát je po koronaviru n
- D23 - Typy masáží a jejich po

Databáze pro skiareál

Matěj Gorgol

Semestrální práce > Semestrální práce

[Rozbalit vše](#)[Sbalit vše](#)[Download](#)[Upravit](#)

Popis

Nově otevřené moderní zimní středisko v Rakouských Alpách Ski Easy potřebuje nový systém, který by jim pomohl se správou areálu. Toto nové středisko nabízí mnoho služeb pro zákazníky, kteří zde přijíždějí z celé Evropy. Středisko nabízí dlouhé sjezdovky všech typů a obtížností. Krom sjezdovek nabízí toto středisko i jiné služby, mezi nejoblíbenější služby střediska patří například masáže od profesionálních masérů, výcházky s průvodcem a extrémní zážitky (například cesta helikoptérou až na vrcholky hory, nebo seskok padákom).

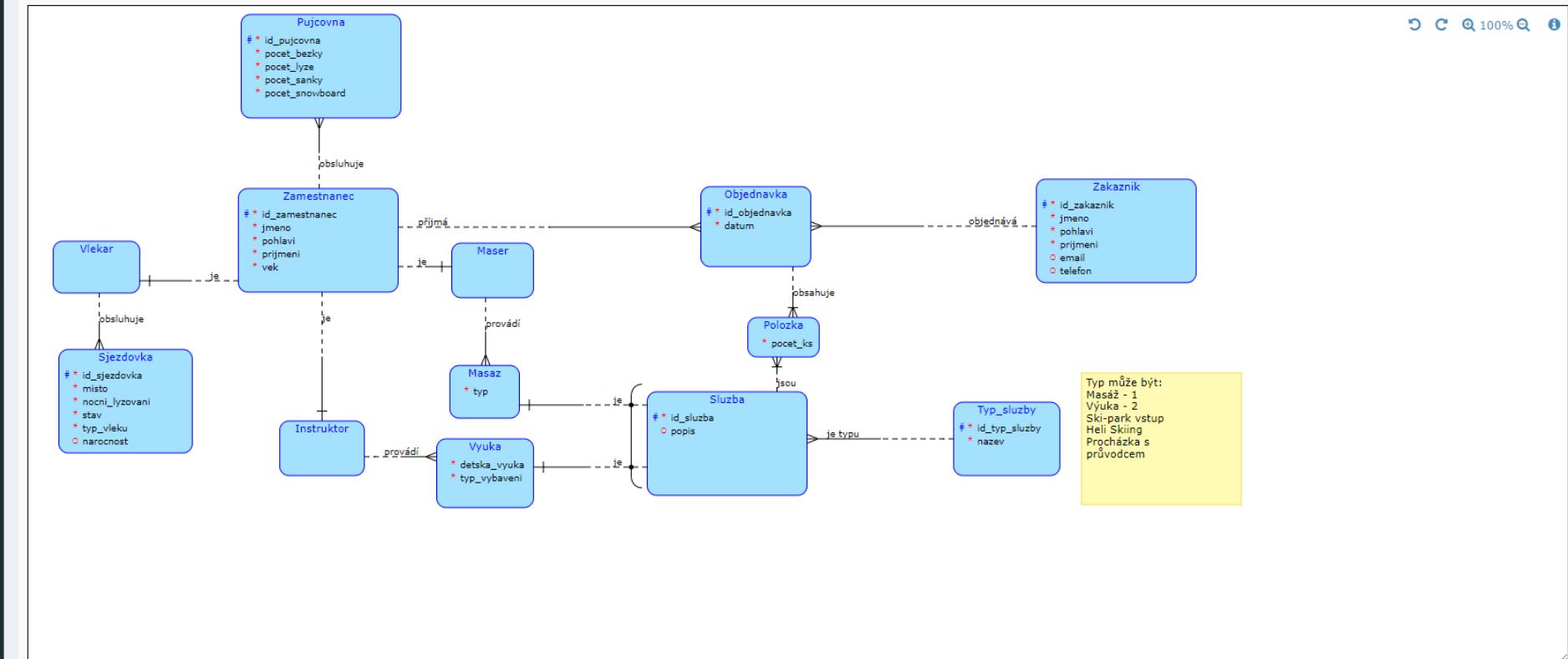
V systému si bude moci zákazník vybrat a objednat službu která ho zajímá, půjčit si lyže, snowboard, běžky a sáňky. Dále bude v systému možné si najít a rezervovat instruktora, nebo maséra podle svých představ. Pro areál pak systém nabízí možnost monitorování stavu sjezdovek a využitost půjčovny.

Systém bude monitorovat **sjezdovky** (místo, náročnost, typ vleku, možnost večerního lyžování, stav), **zaměstnance**(ID, jméno, příjmení, věk, pohlaví a co je jejich výkonem práce) **základníky**(ID, jméno, příjmení, pohlaví, telefon, email), **služby**(typ, datum a o jakou službu se jedná). Mezi služby patří výuka lyžování a snowboardování, masáže více druhů a ostatní extrémní i neextrémní zážitky (bobovou drahou počínaje a heli skiingem konče). Dále pak monitoruje **půjčovny lyží snowboardů a ostatních potřeb pro zimní rádovánky**(ID půjčovny a počet kusů vybavení na skladě) a jednotlivě **objednávky**(ID, typ objednané služby a datum zadání služby), které zákazník zadá.

Vlekař může být pouze u jedné sjezdovky, ale více vlekařů může být u stejné sjezdovky. Instruktor může mít lekci pouze s jedním zákazníkem v jeden čas, to samé platí i pro maséra. Masáž i výuka může být prováděna pouze jedním zaměstnancem.

Počet slov: 261

Konceptuální schema

[Upravit](#)

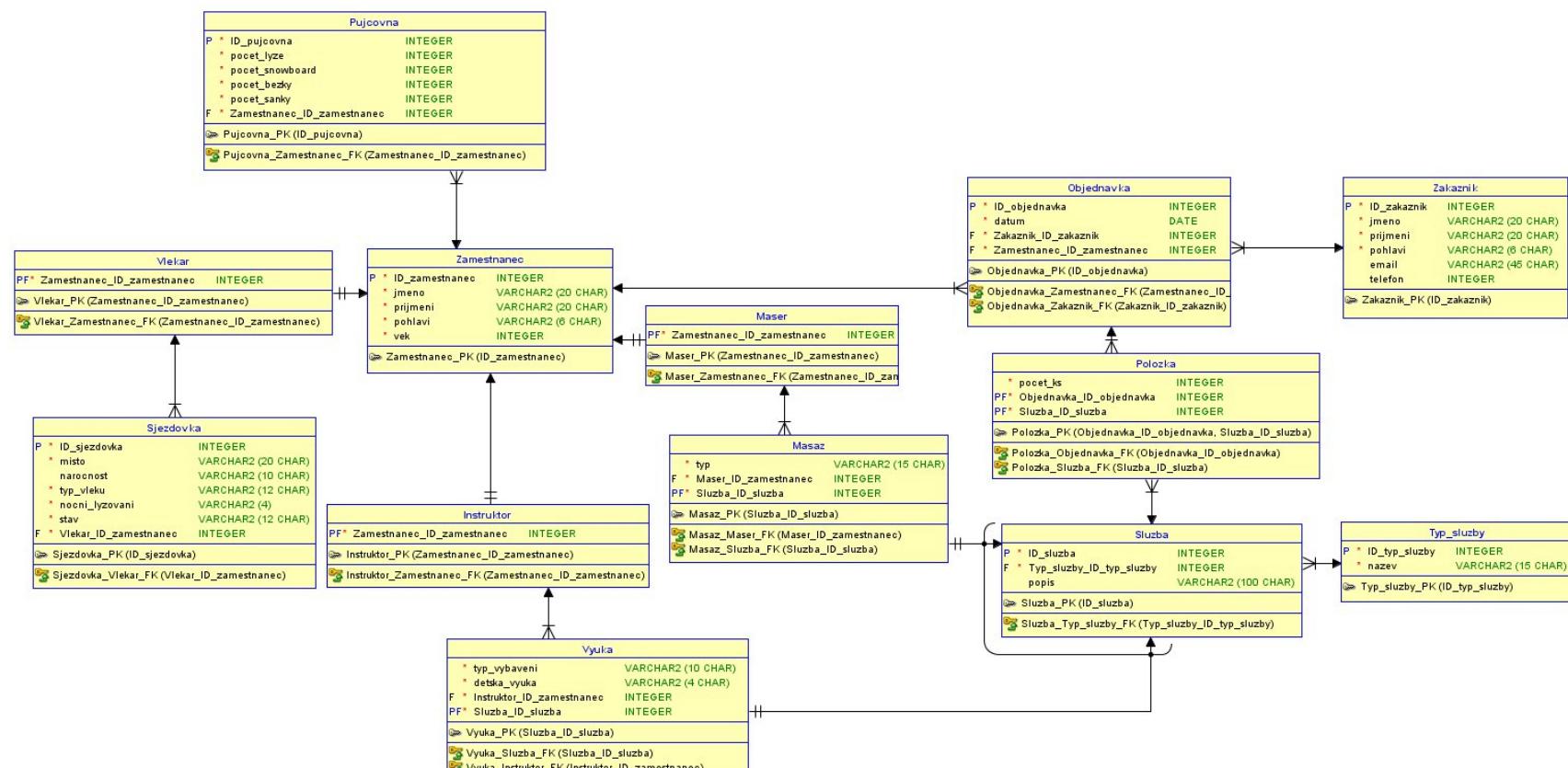
Diskuse smyček

[Upravit](#)

Schéma obsahuje 3 smyčky a to **Služba-Položka-Objednávka-Zaměstnanec(Masér)-Masáz**, **Služba-Položka-Objednávka-Zaměstnanec(Instruktor)-Výuka a Zaměstnanec-Instruktor-Výuka-Služba-Masáz-Masér**. První 2 smyčky nevadí, protože zaměstnanec, který příjma objednávky není závislý na zaměstnanci, který objednanou službu bude vykonávat. Může i nemusí to být ten stejný zaměstnanec, obě možnosti jsou v pořadku. 3. smyčka je také v pořadku, protože ten stejný zaměstnanec může být Masér i Instruktor zároveň a navíc, jedna služba nemůže být Masáz a Výuka zároveň.

Relační schema

Upravit



Create script

Vyhledat ve scriptu Upravit

```

    VTPN1_CIZI_KLICE;
    for v_rec in (select distinct TABLE_NAME from USER_TABLES)
    loop
        execute immediate 'truncate table ||v_rec.TABLE_NAME|| drop storage';
    end loop;
    ZAPNI_CIZI_KLICE;

    -- Nastaví všechny sekvence od 1
    for v_rec in (select distinct SEQUENCE_NAME from USER_SEQUENCES)
    loop
        execute immediate 'alter sequence ||v_rec.SEQUENCE_NAME|| restart start with 1';
    end loop;

```

```
end VYMAZ_DATA_VSECH_TABULEK;
/
```

```
prompt #-----#
```

Insert script

Vyhledat ve scriptu Upravit

```
create or replace procedure SMAZ_VSECHNY_TABULKY AS
-- pokud v logu bude uvedeno, ze nektery objekt nebyl zrusen, protoze na nej jiny jeste existujici objekt stavi, spust proceduru opakovane, dokud se nezrusi vse
begin
    for iRec in
        (select distinct OBJECT_TYPE, OBJECT_NAME,
         'drop'||OBJECT_TYPE||' "'||OBJECT_NAME||'"'
        case OBJECT_TYPE when 'TABLE' then ' cascade constraints purge' else '' end as PRIKAZ
        from USER_OBJECTS where OBJECT_NAME not in ('SMAZ_VSECHNY_TABULKY', 'VYPNI_CIZI_KLICE', 'ZAPNI_CIZI_KLICE', 'VYMAZ_DATA_VSECH_TABULEK')
    ) loop
        begin
        | dbms_output.put_line('Priaz: '||iRec.priaz);
        execute immediate iRec.priaz;
        exception
        | when others then dbms_output.put_line('NEPOVEDLO SE!');
        end;
    end;
```

Dotazy

Rozbalit vše Sbalit vše Upravit

D1 Typ vleků na sjezdovkách střední obtížnosti, které mají večerní lyžování.

RA

```
sjezdovka(NOCNI_LYZOVANI = 'Ano'  $\wedge$  NAROCNOST = 'Střední')[TYP_VLEKU]
```

SQL

```
select distinct s.TYP_VLEKU from sjezdovka s
where NOCNI_LYZOVANI = 'Ano' and NAROCNOST = 'Střední';
```

D2 Kolik instruktorů pracuje v areálu.

A F1 G3 H1

SQL

```
select count(*) as Pocet from(
    select z.jmeno from zamestnanec z
    join instruktor i on i.zamestnanec_id_zamestnanec = z.ID_zamestnanec
);
```

D3 ID masérů, kteří dosud nikoho nemasírovali.

A F1 G1 G4 H2 J

```
select ZAMESTNANEC_ID_ZAMESTNANEC as ID from MASER
  where not exists (
    select * from masaz where ZAMESTNANEC_ID_ZAMESTNANEC = MASER_ID_ZAMESTNANEC
  )
order by ID;

-----
```



```
select ZAMESTNANEC_ID_ZAMESTNANEC as ID from maser
  where ZAMESTNANEC_ID_ZAMESTNANEC not in
  (
    select MASER_ID_ZAMESTNANEC from masaz
  )
order by ID;
```

D4 Seznam všech zaměstnanců, kterým je nad 40 let.

RA

```
zamestnanec (vek > 40)
```

SQL

```
Select * from zamestnanec
  where vek > 40;
```

D5 Jméno, příjmení a věk vlekařů, kteří obsluhovali kotvy nebo kabinky, ale neobsluhovali oba tyto typy vleků.

A B F1 G1 H1 H2 H3

SQL

```
select jmeno, prijmeni, vek from zamestnanec where ID_zamestnanec in(
  select zamestnanec_id_zamestnanec as ID_zamestnanec from vlekar v inner join sjezdovka s on v.zamestnanec_id_zamestnanec = s.vlekar_id_zamestnanec
  where TYP_VLEKU = 'Kotva'
union
  select zamestnanec_id_zamestnanec as ID_zamestnanec from vlekar v inner join sjezdovka s on v.zamestnanec_id_zamestnanec = s.vlekar_id_zamestnanec
  where TYP_VLEKU = 'Kabinka'
)
minus
(
  select zamestnanec_id_zamestnanec as ID_zamestnanec from vlekar v inner join sjezdovka s on v.zamestnanec_id_zamestnanec = s.vlekar_id_zamestnanec
  where TYP_VLEKU = 'Kotva'
intersect
  select zamestnanec_id_zamestnanec as ID_zamestnanec from vlekar v inner join sjezdovka s on v.zamestnanec_id_zamestnanec = s.vlekar_id_zamestnanec
  where TYP_VLEKU = 'Kabinka'
```

D6 Jména vlekařů, kteří obsluhovali vleky v Chacharově.

A F1 G1

RA

```
{}{vlekar [zamestnanec_id_zamestnanec = vlekar_id_zamestnanec] sjezdovka (misto = 'Chacharov')} [ZAMESTNANEC_ID_ZAMESTNANEC = ID_ZAMESTNANEC] zamestnanec {[jmeno]
```

SQL

```
select jmeno from zamestnanec where ID_zamestnanec in(
    select distinct zamestnanec_id_zamestnanec as ID_zamestnanec from vlekar v
        inner join sjezdovka s on v.zamestnanec_id_zamestnanec = s.vlekar_id_zamestnanec
    where misto = 'Chacharov');
```

D7 Jména zaměstnanců, kteří mohou obsluhovat vlek i provádět masáže.

A F2 G1

RA

```
 {{vlekar * maser} [ZAMESTNANEC_ID_ZAMESTNANEC = ID_ZAMESTNANEC] zamestnanec }[jmeno]
```

SQL

```
select jmeno from zamestnanec where ID_zamestnanec IN(
    select * from vlekar join maser using (zamestnanec_id_zamestnanec)
)
```

D8 Objednávky ze dne 20.04.2020, které nevyřízoval zaměstnanec Snoop Dogg.

A B F1 G2 H2

RA

```
 {{zamestnanec[ID_ZAMESTNANEC = ZAMESTNANEC_ID_ZAMESTNANEC] objednavka(DATUM = '20.04.2020')}}

 \ {{zamestnanec(JMENO = 'Snoop' AND PRIJMENI = 'Dogg')[ID_ZAMESTNANEC = ZAMESTNANEC_ID_ZAMESTNANEC] objednavka(DATUM = '20.04.2020')}}

 }[ID_OBJEDNAVKA, DATUM, JMENO, PRIJMENI, ID_ZAMESTNANEC]
```

SQL

```
select ID_OBJEDNAVKA , DATUM, JMENO, PRIJMENI, ID_ZAMESTNANEC from
(
    (select o.Zamestnanec_ID_zamestnanec, o.ID_objednavka, o.datum from objednavka o
     | where o.datum = to_date('20200420', 'YYYYMMDD')
    minus
    (select o2.Zamestnanec_ID_zamestnanec, o2.ID_objednavka, o2.datum from objednavka o2
     | join zamestnanec z on o2.Zamestnanec_ID_zamestnanec = z.ID_zamestnanec
     | and z.jmeno = 'Snoop'
     | and z.prijmeni = 'Dogg'
     | where o2.datum = to_date('20200420', 'YYYYMMDD'))
    )xd
    join zamestnanec z on z.ID_zamestnanec = xd.Zamestnanec_ID_zamestnanec
    order by ID_OBJEDNAVKA asc
)
```

D9 Objednávky, které obsahují pouze Heli-Skiing.

C F1 G2 H2

RA

```
 {{objednavka [objednavka.id_objednavka = polozka.OBJEDNAVKA_ID_OBJEDNAVKA] polozka(sluzba_ID_sluzba = 4)}[ID_OBJEDNAVKA,DATUM]}
 !<
 {{objednavka [objednavka.id_objednavka = polozka.OBJEDNAVKA_ID_OBJEDNAVKA] polozka(sluzba_ID_sluzba != 4)}[ID_OBJEDNAVKA,DATUM]}
```

SQL

```
select ID_OBJEDNAVKA, DATUM from (
    select TD.OBJEDNAVKA.DATUM from objednavka o
```

```

join polozka p on p.OBJEDNAVKA_ID_OBJEDNAVKA = o.ID_OBJEDNAVKA
join sluzba s on s.ID_SLUZBA = p.SLUZBA_ID_SLUZBA
join typ_sluzby ts on ts.ID_TYP_SLUZBY = s.TYP_SLUZBY_ID_TYP_SLUZBY
where ts.ID_TYP_SLUZBY = 4)
minus
(select ID_OBJEDNAVKA,DATUM from objednavka o
join polozka p on p.OBJEDNAVKA_ID_OBJEDNAVKA = o.ID_OBJEDNAVKA
join sluzba s on s.ID_SLUZBA = p.SLUZBA_ID_SLUZBA
join typ_sluzby ts on ts.ID_TYP_SLUZBY = s.TYP_SLUZBY_ID_TYP_SLUZBY
where ts.ID_TYP_SLUZBY != 4)
);

```

D10 Telefony sester Cockney. (pohlavi = žena, příjmení = Cockney)

RA

```
zakaznik(pohlavi = 'žena' ∧ prijmeni = 'Cockney') [JMENO, PRIJMENI, TELEFON]
```

SQL

```
SELECT JMENO, PRIJMENI, TELEFON FROM ZAKAZNIK
WHERE pohlavi = 'Žena' AND prijmeni = 'Cockney';
```

D11 Id_objednávky, datum, id_zákazníka, jméno, příjmení, email a telefon zákazníka a objednávky, která obsahuje všechny služby typu masáž.

A D1 F1 H1

SQL

```
select ID_OBJEDNAVKA,DATUM,ID_ZAKAZNIK,JMENO,PRIJMENI,EMAIL,TELEFON from objednavka o
join zakaznik on o.ZAKAZNIK_ID_ZAKAZNIK = ID_ZAKAZNIK
where(
    (select count(DISTINCT ID_SLUZBA) from polozka p
     join sluzba ss on ss.ID_SLUZBA = p.SLUZBA_ID_SLUZBA
     where ss.TYP_SLUZBY_ID_TYP_SLUZBY = 1 and p.objednavka_ID_objednavka = o.ID_objednavka)
    =
    --počet služeb typu masaz
    (select count(ID_SLUZBA) from sluzba s
     join typ_sluzby t on s.TYP_SLUZBY_ID_TYP_SLUZBY = t.ID_TYP_SLUZBY
     where t.ID_TYP_SLUZBY = 1)
)
```

D12 Kontrola D1 - masážové služby, které si objednal zákazník z D11 ménus všechny masážové služby.

A D2 F1 H2 H3

SQL

```

    (select count(DISTINCT ID_SLUZBA) from polozka p
     join objednavka o on o.ID_objednavka = p.objednavka_ID_objednavka
     join sluzba ss on ss.ID_SLUZBA = p.SLUZBA_ID_SLUZBA
     where ss.TYP_SLUZBY_ID_TYP_SLUZBY = 1)
    =
    --počet služeb typu masaz
    (select count(ID_SLUZBA) from sluzba s
     join typ_sluzby t on s.TYP_SLUZBY_ID_TYP_SLUZBY = t.ID_TYP_SLUZBY
     where t.ID_TYP_SLUZBY = 1)
)
minus
(select DISTINCT ID_SLUZBA from sluzba )

```

D13 ID objednávky, která obsahuje všechny služby.

D1

RA

```
{polozka[SLUZBA_ID_SLUZBA = ID_SLUZBA]sluzba}[OBJEDNAVKA_ID_OBJEDNAVKA, ID_SLUZBA]+sluzba[ID_SLUZBA]
```

D14 Zaměstnanec, který příjmal objednávky pouze 20.4.2018.

C F1 H2

RA

```
 {{zamestnanec[ID_ZAMESTNANEC = ZAMESTNANEC_ID_ZAMESTNANEC] objednavka(DATUM = '20.04.2020'))[ID_ZAMESTNANEC, JMENO, PRIJMENI, POHLAVI, VEK]}  
 \ {{zamestnanec[ID_ZAMESTNANEC = ZAMESTNANEC_ID_ZAMESTNANEC] objednavka(DATUM != '20.04.2020'))[ID_ZAMESTNANEC, JMENO, PRIJMENI, POHLAVI, VEK]}
```

SQL

```
select id_zamestnanec, jmeno, prijmeni, pohlavi, vek from ZAMESTNANEC  
join objednavka o2 on id_zamestnanec = zamestnanec_id_zamestnanec  
where o2.datum = '20.04.2020'  
  
MINUS  
  
select id_zamestnanec, jmeno, prijmeni, pohlavi, vek from ZAMESTNANEC  
join objednavka o2 on id_zamestnanec = zamestnanec_id_zamestnanec  
where o2.datum != '20.04.2020';
```

D15 Všechny možné kombinace vlekaře a sjezdovek.

F3

RA

```
vlekar x sjezdovka
```

SQL

```
SELECT * FROM VLEKAR  
CROSS JOIN SJEZDOVKA;
```

D16 Vyber zákazníky a všechny zaměstnance, kteří je obsloužili, včetně těch zákazníků, kteří nebyli nikdy obsluženi.

A F4 F5

SQL

```
select z.jmeno as jmeno_zakaznika,  
 z.prijmeni as prijmeni_zakaznika,  
 zz.jmeno as jmeno_zamestnance,  
 zz.prijmeni as prijmeni_zamestnance  
from ZAKAZNIK z  
 full outer join ODETMENIVA o on z.ID_ZAKAZNIK = o.ID_ZAKAZNIK  
 full outer join ZAKAZNIK z2 on z.ID_ZAKAZNIK = z2.ID_ZAKAZNIK  
 where z.jmeno is null or z.prijmeni is null  
 or z2.jmeno is null or z2.prijmeni is null
```

```
TUJÍ VOLEJ JUDÍT VOLÁVATKA A VÍT ZÍD_ZAMĚSTNANEC_ID_ZAMĚSTNANEC  
left join ZAMESTNANEC zz on ID_ZAMESTNANEC=o.ZAMESTNANEC_ID_ZAMESTNANEC;
```

D17 Telefony zákazníků - mužů, kteří nemají uveden email.

SQL

```
select JMENO, PRIJMENI, TELEFON from zakaznik  
where pohlavi = 'Muž' and email is null and telefon is not null
```

D18 Vyberte objednávky, které obsahují alespoň 3 různé služby o 2 a více kusech a seřaďte je podle ID objednávky. (musí obsahovat min. 3 položky, kde počet kusů je min. 2)

A F1 L 2 K

SQL

```
select ID_OBJEDNAVKA as ID_OBJEDNAVKY, DATUM as DATUM_OBJEDNANI from objednavka o  
join polozka p on ID_OBJEDNAVKA = OBJEDNAVKA_ID_OBJEDNAVKA  
join sluzba s on ID_SLUZBA = SLUZBA_ID_SLUZBA  
where pocet_ks >= 2  
group by ID_OBJEDNAVKA, DATUM  
having count (*) >= 3  
order by ID_OBJEDNAVKA
```

D19 Do objednávky ID 3 přidej všechny položky z objednávky ID 1.

N

SQL

```
--select * from polozka where OBJEDNAVKA_ID_OBJEDNAVKA = 3; --kontrolovaci  
--select * from polozka where OBJEDNAVKA_ID_OBJEDNAVKA = 1; --kontrolovaci  
  
insert into polozka (POCET_KS, OBJEDNAVKA_ID_OBJEDNAVKA, SLUZBA_ID_SLUZBA)  
    select POCET_KS,3,SLUZBA_ID_SLUZBA from polozka where OBJEDNAVKA_ID_OBJEDNAVKA = 1;  
  
--select * from polozka where OBJEDNAVKA_ID_OBJEDNAVKA = 3; --kontrolovaci  
rollback;  
--transakce
```

D20 Vytvořit pohled instruktorů a jejich hodin, a vypsat jaké hodiny měli instruktoři a instruktorky s příjmením Clown.

A F1 L M

SQL

```
create view v_Instruktori as  
select id_zamestnanec as ID, jmeno, prijmeni, typ_vybaveni, detska_vyuka, popis  
from zamestnanec z  
join instruktor on ID_ZAMESTNANEC = ZAMESTNANEC_ID_ZAMESTNANEC  
join vyuka v on INSTRUKTOR_ID_ZAMESTNANEC = ZAMESTNANEC_ID_ZAMESTNANEC  
join sluzba s on ID_SLUZBA = SLUZBA_ID_SLUZBA  
order by z.id_zamestnanec asc;  
  
select * from v_Instruktori where prijmeni = 'Clown';  
  
drop view v_Instruktori;
```

D21 Protože práce v půjčovně je pro mladé to nejnudnější na světě, tak všichni zaměstnanci pod 30 let, kteří pracovali v půjčovně zestáli nudou o 10 let.

A F1 G1 0

SQL

```
update zamestnaneck
set vek = vek + 10
where ID_ZAMESTNANEC in
  (select z.ID_ZAMESTNANEC from pujcovna p
   join zamestnaneck on p.ZAMESTNANEC_ID_ZAMESTNANEC = z.ID_ZAMESTNANEC
   and z.vek < 30
  );
rollback; --Ať jim můžeme přidávat do nekonečna :
```

D22 Stát je po koronaviru na mizině a tak ruší příspěvky na zaměstnávání lidí, kteří už jsou v pokročilém důchodovém věku. Bohužel už si je tedy nemůžeme dovolit zaměstnávat a musíme je propustit.

G1 P

SQL

```
delete zamestnaneck
  where ID_ZAMESTNANEC in
    (select ID_ZAMESTNANEC from zamestnaneck where vek >= 75);
rollback; --necháme si je jestě chvilku
```

D23 Typy masáží a jejich popis.

A F1

SQL

```
select typ, popis from sluzba s
  join masaz m on sluzba_id_sluzba = id_sluzba
  where s.typ_sluzby_id_typ_sluzby = 1
```

D24 Vyberte nejstaršího zaměstnance.

A F2 G2 H1

SQL

```
select * from zamestnaneck
  join (select max(vek) as vek from zamestnaneck) using (vek)
```

D25 Zaokrouhlený průměrný věk zaměstnanců - mužů.

H1

SQL

```
select round(avg(vek)) from zamestnaneck
  where pohlavi = 'Muž'
```

| | |
|---|--|
| D26 | Zakazníci, u kterých nevidujeme žádný kontakt. |
| SQL | |
| <pre>select ID_ZAKAZNIK, JMENO, PRIJMENI, POHLAVI from zakaznik where email is null and telefon is null</pre> | |

| Kategorie pokryté dotazy | | Pokryvá |
|--------------------------|---|--|
| A | A - Pozitivní dotaz nad spojením alespoň dvou tabulek | D2 D3 D5 D6 D7 D8 D11 D12 D16 D18 D20 D21 D23 D24 |
| A _R | A (RA) - Pozitivní dotaz nad spojením alespoň dvou tabulek | D6 D7 D8 |
| B | B - Negativní dotaz nad spojením alespoň dvou tabulek | D5 D8 |
| B _R | B (RA) - Negativní dotaz nad spojením alespoň dvou tabulek | D8 |
| C | C - Vyber ty, kteří mají vztah POUZE k ... | D9 D14 |
| D ₁ | D1 - Vyber ty, kteří/které jsou ve vztahu se všemi - dotaz s univerzální kvantifikací | D11 |
| D _{1N} | D1 (NATURAL) - Vyber ty, kteří/které jsou ve vztahu se všemi - dotaz s univerzální kvantifikací | D11 D13 |
| D _{1R} | D1 (RA) - Vyber ty, kteří/které jsou ve vztahu se všemi - dotaz s univerzální kvantifikací | D13 |
| D ₂ | D2 - Kontrola výsledku dotazu z kategorie D1 | D12 |
| F ₁ | F1 - JOIN ON | D2 D3 D5 D6 D8 D9 D11 D12 D14 D18 D20 D21 D23 |
| F _{1R} | F1 (RA) - JOIN ON | D6 D8 D9 D14 |
| F ₂ | F2 - NATURAL JOIN JOIN USING | D7 D24 |
| F _{2R} | F2 (RA) - NATURAL JOIN JOIN USING | D7 |
| F ₃ | F3 - CROSS JOIN | D15 |
| F ₄ | F4 - LEFT RIGHT OUTER JOIN | D16 |
| F ₅ | F5 - FULL (OUTER) JOIN | D16 |
| G ₁ | G1 - Vnořený dotaz v klauzuli WHERE | D3 D5 D6 D7 D21 D22 |
| G _{1R} | G1 (RA) - Vnořený dotaz v klauzuli WHERE | D6 D7 |
| G ₂ | G2 - Vnořený dotaz v klauzuli FROM | D8 D9 D24 |
| G _{2R} | G2 (RA) - Vnořený dotaz v klauzuli FROM | D8 D9 |
| G ₃ | G3 - Vnořený dotaz v klauzuli SELECT | D2 |
| G ₄ | G4 - Vzařazený vnořený dotaz (EXISTS, NOT EXISTS) | D3 |
| H ₁ | H1 - Množinové sjednocení - UNION | D5 |

| | | |
|------|---|------------------------|
| H2 | H2 - Množinový rozdíl - MINUS nebo EXCEPT | D3 D5 D8 D9 D12 D14 |
| H2_R | H2 (RA) - Množinový rozdíl - MINUS nebo EXCEPT | D8 D9 D14 |
| H3 | H3 - Množinový průnik - INTERSECT | D5 |
| I1 | I1 - Agregační funkce (count sum min max avg) | D2 D11 D12 D18 D24 D25 |
| I2 | I2 - Agregační funkce nad seskupenými řádky - GROUP BY (HAVING) | D18 |
| J | J - Stejný dotaz ve třech různých formulacích SQL | D3 |
| K | K - Všechny klauzule v 1 dotazu - SELECT FROM WHERE GROUP BY HAVING ORDER BY | D18 |
| L | L - VIEW | D20 |
| M | M - Dotaz nad pohledem | D20 |
| N | N - INSERT, který vloží do některé tabulky množinu řádků, které jsou vybrány dotazem z vybraných tabulek (příkaz INSERT, ve kterém je klauzule VALUES nahrazena vnořeným poddotazem). | D19 |
| O | O - UPDATE s vnořeným SELECT příkazem | D21 |
| P | P - DELETE s vnořeným SELECT příkazem | D22 |

Závěr

Upravit ▾

Předmět DBS mě docela baví a semestrální práci jsem si užil. Naučil jsem se nějaké základní operace a prací s databází. Vypracování semestrální práce bylo, až na menší obtíže, bezproblémové. Portál DBS je velice přehledný, pěkně udělaný a dobře se v něm pracuje, což se ale bohužel nedá říci o SQL developeru. Ale i to se dalo vydržet a ke konci jsem si na to už zvykl.

Zdroje

Upravit ▾

Pomocné procedury z nové ukázkové semestrální práce: [zde](#)

Prosemináře Jiřího Hunky: [zde](#)