

1. Vypište názvy všech jídel stojících 100Kč.

```
SELECT nazev FROM JIDELNICEK WHERE cena=100;
```

2. Vypište různá jména všech hostů.

```
SELECT DISTINCT jmeno FROM host;
```

3. Zjistěte průměrnou cenu všech jídel na jídelníčku.

```
SELECT AVG(cena) from jidelnicek;
```

4. Zjistěte počet jídel, jejichž cena je mezi 100 a 150Kč.

```
SELECT COUNT(*) from jidelnicek where cena between 100 and 150;
```

5. Zjistěte počet různých jmen všech hostů.

```
* SELECT COUNT(DISTINCT jmeno) FROM host;
```

6. Zjistěte počet hostů, u kterých není vyplněna položka DATUM_NAROZENI.

```
SELECT COUNT(*) FROM host WHERE datum_narozeni is null;
```

7. Zjistěte počet hostů, jejichž jméno začíná na L nebo v jejichž příjmení se nevyskytuje písmeno A.

```
SELECT COUNT(*) FROM host WHERE prijmeni LIKE 'L%' OR prijmeni NOT LIKE '%A%'
```

8. Zjistěte názvy nejdražšího a nejlevnějšího pokrmu.

9. Pro každé číslo účtu vypište seznam objednaných pokrmů.

```
* select u.cislo_uctu, j.nazev from ucty u, polozka p, jidelnicek j where u.cislo_uctu = p.cislo_uctu and p.cislo_pokrmu = j.cislo_pokrmu;
```

10. Vypište seznam hostů s počtem jejich návštěv. Seznam seřadte sestupně podle počtu návštěv.

```
* select h.jmeno, h.prijmeni, count(u.cislo_uctu) from host h, ucty u where h.cislo_hosta = u.cislo_hosta group by h.cislo_hosta order by count(u.cislo_uctu) desc;
```

11. Zjistěte počet hostů, kteří nemají žádnou návštěvu restaurace.

```
* select count(*) from host where cislo_hosta not in (select cislo_hosta from ucty);
```

12. Vypište jména a příjmení hostů, kteří mají více jak 10 návštěv restaurace.

```
* select h.jmeno, h.prijmeni from host h, ucty u where h.cislo_hosta = u.cislo_hosta group by h.cislo_hosta having count(u.cislo_hosta) > 10;
```

13. Přidejte nového hosta.

```
insert into host values(100, 'petr', 'svetr', 1982);
```

14. Změňte novému hostu datum narození.

```
update host set datum_narozeni = 1912 where cislo_hosta = 100;
```

15. Zrušte všechny pokrmy levnější jak 30Kč.

```
delete from jidelnicek where cena < 30;
```

16. Zdražte všechny pokrmy v jídelníčku o 10%.

```
update jidelnicek set cena = cena * 1.1;
```

17. Vytvořte tabulku HOST se sloupci CISLO_HOSTA, JMENO, PRIJMENI a DATUM_NAROZENI. Číslo hosta je primární klíč, jméno a příjmení jsou povinné.

```
create table host (  
cislo_hosta number(5) primary key,  
jmeno varchar2(20) not null,  
prijmeni varchar2(20) not null,  
datum_narozeni date);
```

18. Vytvořte tabulku UCET. Primárním klíčem je CISLO_UCTU. Zajistěte referenční integritu, tj. k účtu musí existovat host.

```
create table ucet (  
cislo_uctu number(5) primary key,  
cislo_hosta number(5),  
constraints p_ch foreign key cislo_hosta  
references host (cislo_hosta)  
);
```

19. Napište, jakým způsobem lze zajistit, aby při rušení hosta byly automaticky zrušeny všechny jeho účty.

```
...  
constraints f_ch foreign key cislo_hosta  
references host (cislo_hosta)  
on delete cascade  
...
```

20. Zajistěte, aby nebylo možné zadat do jídelníčku pokrm, jehož cena není mezi 10 a 1000 Kč.

```
CREATE TABLE jidelnicek (  
....  
CENA NUMBER(4) CHECK (CENA >= 10 AND CENA <= 1000)  
);
```