

Виробнича практика з проектування баз даних та комп'ютерних мереж

Метою виробничої практики з проектування баз даних та комп'ютерних мереж є:

- формування у студентів професійних, практичних знань, вмінь та навиків, необхідних для успішної роботи в організаціях, що проектують чи експлуатують бази даних та інформаційні системи;
- отримання практичних навичок з проектування корпоративних комп'ютерних мереж в реальних умовах та віртуальній мережевій лабораторії Cisco Packet tracer, а також вивчення команд операційної системи компанії Cisco Internetwork Operating System (IOS) для конфігурації пристроїв.

У період проходження практики студенти поглиблюють і закріплюють теоретичні знання і практичні уміння, отримані в процесі вивчення навчальних дисциплін «Архітектура ЕОМ і обчислювальні системи, мережі та комп'ютерні комунікації», «Бази даних та інформаційні системи»

Основними задачами практики щодо комп'ютерних мереж є наступні:

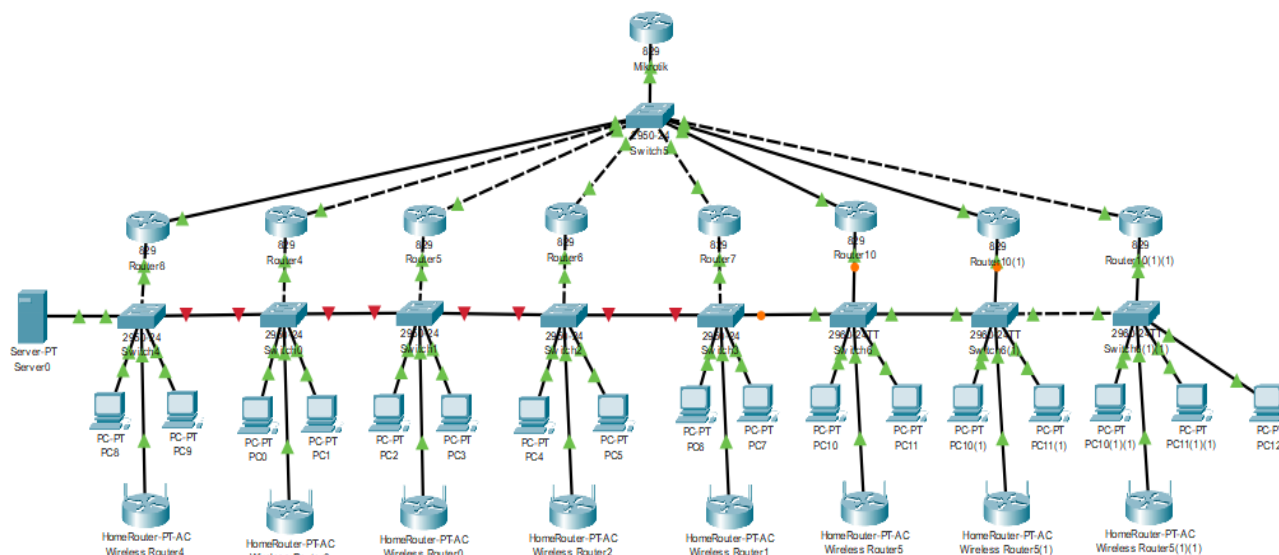
- закріпити, практичні навички, отриманих студентами під час вивчення дисципліни «Комп'ютерні мережі» щодо побудови, принципів організації внутрішньої структури сучасних комп'ютерних мереж відповідно до запропонованої топології організації;
- виконати опис апаратних пристроїв для проектування мережі, їх технічні характеристики;
- виконати розрахунки схеми IP-адресації методом VLSM з урахуванням вимог організації;
- налаштувати маршрутизатори, комутатори і комп'ютери для підтримки з'єднання IPv4;
- налаштувати технологію VLAN;
- налаштувати маршрутизацію між VLAN та між мережами;
- налаштувати DHCP та динамічний NAT;
- представити на перевірку спроектовану комп'ютерну мережу в Cisco Packet Tracer

У практичній частині при роботі з комп'ютерними мережами студенти збирають свої комп'ютерні робочі станції з окремих компонентів та створюють комп'ютерну мережу, яка може служити основою для мережі малого та середнього бізнесу. Студенти працюють у групах по 2 або 3 особи.

Групі потрібно використовувати наступне обладнання:

- 1 маршрутизатор Cisco,
- 1 комутатор Cisco,
- 1 SOHO-маршрутизатор, який використовується як точка доступу,
- 2 ПК.

Кожен мережевий пристрій налаштовується студентськими командами з їх власними



налаштуваннями. Кожна група конфігурує власну серверну операційну систему на основі налаштувань, запропонованих у практичному завданні.

В результаті, групи разом отримують доступ до інтернету та здатні надавати створені мережеві послуги іншим командам та їхнім пристроям.

Основними завданнями практики щодо баз даних та інформаційних систем є розробка та створення бази даних для віртуальної компанії/організації/установи відповідно до заданої системи умов. Завдання практики виконується проектним методом з участю 2-3 студентів. Крім того, практика також включає індивідуальне завдання з управління базою даних.

Запропоновані проекти:

1. Онлайн щоденник
2. Онлайн-магазин для меблевої компанії
3. Компанія організаторів заходів
4. Система обліку пацієнтів сімейного лікаря
5. Прокат автомобілів
6. Туристичне агентство
7. Продовольчі склади
8. Обласний конкурс з програмування
9. Ветеринарна клініка
10. Соціальна мережа

До кожного проекту належить опис, який містить вимоги користувачів. У ньому вказуються потреби та очікування клієнта щодо розроблюваної системи та бази даних. Студентам потрібно проаналізувати цей опис та спроектувати базу даних системи відповідно до висловлених у ньому вимог.

Приклад опису проекту:

Система обліку студентів і обладнання в студентському гуртожитку

Один з університетських гуртожитків хотів би вести облік і зробити більш зручним управління своїм парком обладнання та даними студентів, що проживають у гуртожитку. Завдання полягає у розробці бази даних для обліку студентів та обладнання гуртожитку.

Гуртожиток має три поверхи, на кожному з яких можуть бути кімнати, в яких проживають лише хлопці або лише дівчата. Найменша кімната розрахована на двох осіб. Кожна кімната має свій номер, який на першому поверсі має формат 1##, на другому - 2##, на третьому - 3##, де # - це цифра.

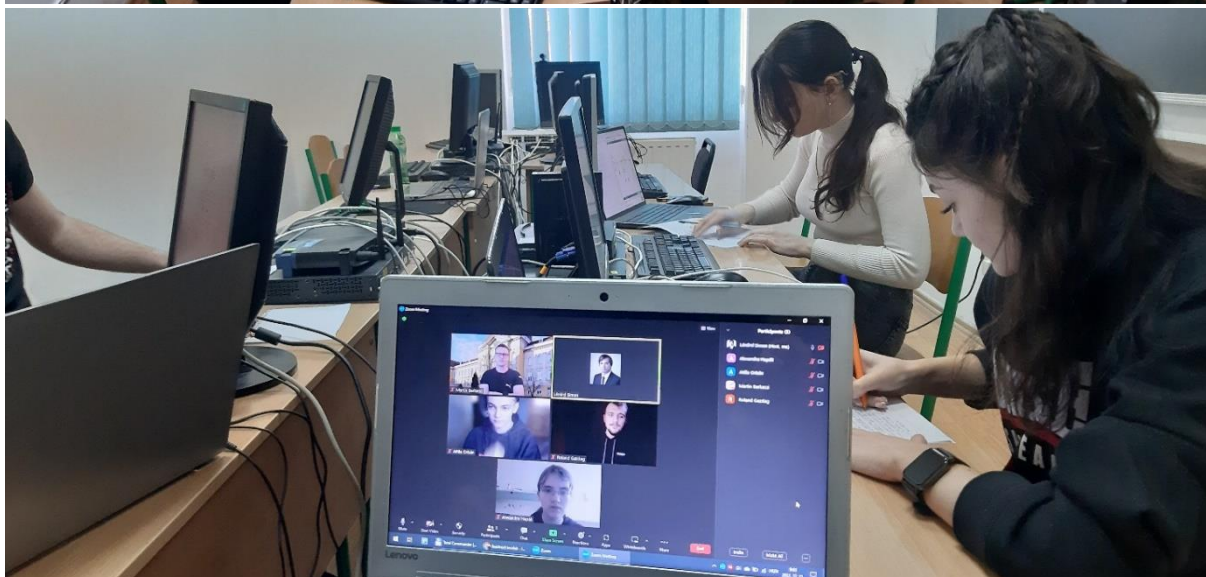
Щоб ефективно управляти обладнанням гуртожитку (ліжка, стільці, столи тощо), кожному з них присвоєно унікальний номер, який зазвичай відображається на самому обладнанні у вигляді QR-коду. Про кожен предмет обладнання хочуть знати якомога більше інформації, наприклад, до якої кімнати він належить, який у нього стан (новий, старий, потребує ремонту тощо), коли він потрапив у гуртожиток, чи був він ремонтований/оновлений і коли тощо. Крім того, про кожен предмет необхідно зберігати фотографію. Деяке обладнання потрапило у гуртожиток завдяки благодійним фондам чи іншим організаціям, про що також потрібно зберігати інформацію.

Про студентів, крім основних особистих даних, потрібно знати середній бал попереднього семестру, відстань між місцем проживання та гуртожитком, чи були вони мешканцями гуртожитку в попередньому році та скільки скарг на них надійшло і їх обґрунтування, сімейний стан (чи є велика сім'я (якщо так, скільки братів і сестер), чи один з батьків виховує дитину, сирота тощо) та ін. Тобто також інформацію, яка може вплинути на наступний набір

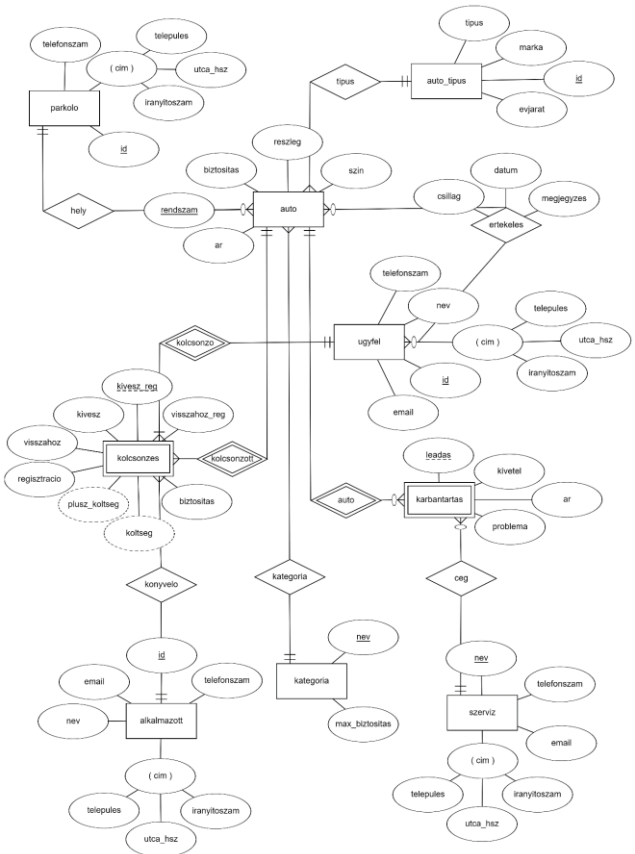
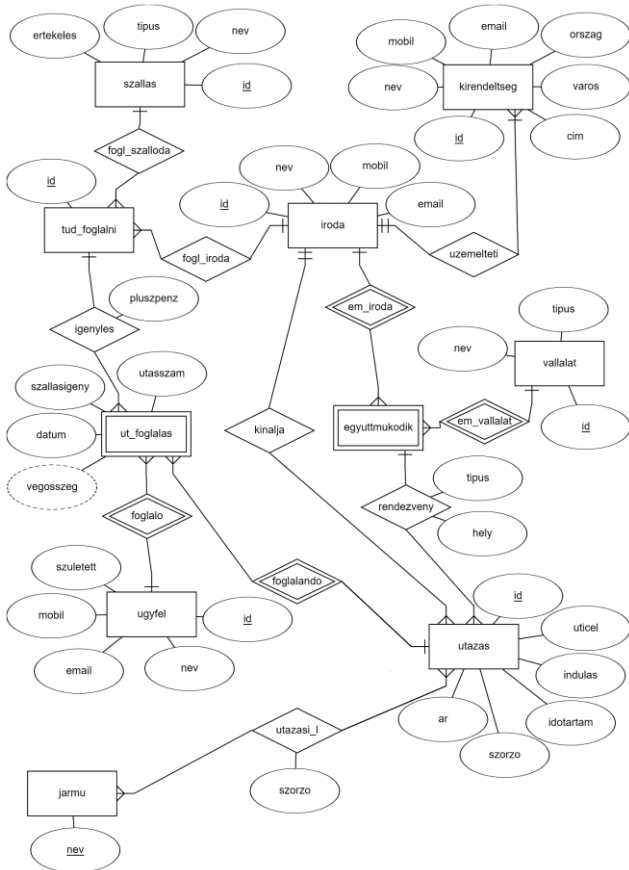
у гуртожиток. Важливо знати про кожного студента, в якій кімнаті він проживає. У кожного студента повинен бути щонайменше один родич з контактним номером телефону. Звісно, про студентів хотіли б зберігати інформацію про стан сплати щомісячного внеску та коли вони його сплачували. Якщо можливо, хотіли б, щоб можна було знайти інформацію про платежі кожного студента за минулі місяці поточного навчального року.

Щодо кімнат, потрібно знати, скільки в ній місць для проживання, скільки вікон, який її стан (недавно відремонтована, поганий стан, хороший стан тощо), хто в ній проживає, яке обладнання знаходиться в кімнаті тощо.

**Світлини з навчальної практики з проектування баз даних
та комп'ютерних мереж
2022-2023**



Из виконаних звітів робіт студентів по керуванню базами даних:



```

CREATE DATABASE autokolcsonzo CHARACTER SET 'utf8mb4' COLLATE 'utf8mb4_unicode_ci';

CREATE TABLE parkolo(
  id INT PRIMARY KEY,
  telefonszam CHAR(13) NOT NULL UNIQUE,
  iranyitoszam INT NOT NULL,
  telepules VARCHAR(30),
  utca_hsz VARCHAR(50) NOT NULL
);

CREATE TABLE auto_tipus(
  id INT PRIMARY KEY AUTO_INCREMENT,
  marka VARCHAR(30) NOT NULL,
  tipus VARCHAR(50) NOT NULL,
  evjarat INT NOT NULL
);

CREATE TABLE kategoria(
  nev VARCHAR(30) PRIMARY KEY,
  max_biztositas FLOAT NOT NULL
);

CREATE TABLE auto(
  rendszam CHAR(7) PRIMARY KEY,
  tipus INT,
  kategoria VARCHAR(30),
  hely INT,
  reszleg VARCHAR(10) NOT NULL UNIQUE,
  szin VARCHAR(30) NOT NULL,
  ar FLOAT NOT NULL,
  FOREIGN KEY(tipus) REFERENCES auto_tipus(id),
  FOREIGN KEY(kategoria) REFERENCES kategoria(nev),
  FOREIGN KEY(hely) REFERENCES parkolo(id),
  CONSTRAINT rendszamForma CHECK(((LEFT(rendszam,3)REGEXP '[A-Z]*') AND (RIGHT(rendszam,3)REGEXP '[0-9]*') AND SUBSTRING(rendszam,4,1)='-'))
);

CREATE TABLE kolcsonzes(
  ugyfel INT,
  rendszam CHAR(7),
  alkalmazott INT,
  regisztracio DATE DEFAULT CURRENT_DATE,
  kivesz_reg DATE NOT NULL,
  visszahoz_reg DATE NOT NULL,
  kivesz DATE NULL,
  visszahoz DATE NULL,
  biztositas FLOAT NOT NULL,
  koltseg FLOAT NULL,
  plusz_koltseg FLOAT NULL,
  PRIMARY KEY(rendszam, ugyfel, kivesz_reg),
  FOREIGN KEY(rendszam) REFERENCES auto(rendszam),
  FOREIGN KEY(alkalmazott) REFERENCES alkalmazott(id)
);

CREATE TABLE erkeeles(
  ugyfel INT,
  rendszam CHAR(7),
  megjegyzes VARCHAR(500),
  csillag INT(5) NOT NULL,
  datum DATE DEFAULT CURRENT_DATE,
  PRIMARY KEY(ugyfel, rendszam),
  FOREIGN KEY(ugyfel) REFERENCES ugyfel(id),
  FOREIGN KEY(rendszam) REFERENCES auto(rendszam),
  CONSTRAINT csillagok_szama CHECK(csillag BETWEEN 1 AND 5)
);

CREATE TABLE szorviz(
  nev VARCHAR(50) PRIMARY KEY,
  telefonszam CHAR(13) NOT NULL UNIQUE,
  iranyitoszam INT NOT NULL,
  telepules VARCHAR(30),
  utca_hsz VARCHAR(50) NOT NULL
);

CREATE TABLE karbantartas(
  ceg VARCHAR(50) NOT NULL,
  rendszam CHAR(7),
  problema VARCHAR(100) NOT NULL,
  ar FLOAT NULL,
  leadas DATE DEFAULT CURRENT_DATE,
  kivetel DATE NULL,
  PRIMARY KEY(ceg, rendszam, leadas),
  FOREIGN KEY(ceg) REFERENCES szorviz(nev),
  FOREIGN KEY(rendszam) REFERENCES auto(rendszam)
);

CREATE TRIGGER erkekelhet
BEFORE INSERT ON erkeeles
FOR EACH ROW
IF (new.ugyfel <>
  SELECT k1.ugyfel
  FROM kolcsonzes AS k1
  WHERE k1.ugyfel=new.ugyfel
  AND k1.rendszam=new.rendszam
  AND k1.visszahoz_reg=>ALL(
    SELECT k2.visszahoz_reg
    FROM kolcsonzes AS k2
    WHERE k1.ugyfel=k2.ugyfel
    ) AND k1.visszahoz IS NOT NULL)
  THEN SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'JOGTALAN ERTEKELES <<a megadott ugyfel es az altala erkekelendo auto kozott nincs altalunk feljegyzett kapcsolat>>;'
END IF;

CREATE TRIGGER kotelezo_ar
BEFORE UPDATE ON karbantartas
FOR EACH ROW
IF (new.kivetel IS NOT NULL AND new.ar IS NULL)
  THEN SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'PONTTALAN ADATBOVITES <<minden mezo kitoltese kotelezo>>;'
END IF;

CREATE TRIGGER idoben_korrekt_foglalas_ize
BEFORE UPDATE ON kolcsonzes
FOR EACH ROW
IF DATEDIFF(kolcsonzes.kivesz_reg, kolcsonzes.kivesz)=0 THEN
  UPDATE kolcsonzes
  SET koltseg=auto.ar*(DATEDIFF(kolcsonzes.kivesz_reg, kolcsonzes.visszahoz_reg)/86400) SECOND
END IF;
ELSE
  THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'AZ AUTO NEM ELVIHETO <<nincs ervenyes foglalas>>;'
END ELSE;
END
    
```

```

SELECT
    ugyfel.id AS "ugyfel azonositoja",
    ugyfel.nev AS "ugyfel neve",
    utazas.id AS "Utazas azonositoja",
    utazas.uticel AS "Uticel",
    ut_foglalas.datum AS "Foglalas datuma",
    utazas.indulas AS "Indulas datuma"
FROM ugyfel
RIGHT JOIN ut_foglalas ON ut_foglalas.ugyfel_id = ugyfel.id
LEFT JOIN utazas ON utazas.id = ut_foglalas.utazas_id
LEFT JOIN utazasi_l ON utazasi_l.id = utazas.id
    AND utazasi_l.szorzo = 1
    AND utazasi_l.nev LIKE "Repulogep"
ORDER BY ut_foglalas.datum;

```

```

SELECT
    utazas.id AS "Utazas azonositoja",
    utazas.uticel AS "Uticel",
    ut_foglalas.pluszpenz AS "Szallas ara",
    iroda.nev AS "Kozremukodo iroda"
FROM ut_foglalas
LEFT JOIN utazas ON utazas.id = ut_foglalas.utazas_id
LEFT JOIN iroda ON utazas.iroda_id = iroda.id
WHERE ut_foglalas.pluszpenz > 0
ORDER BY ut_foglalas.pluszpenz
LIMIT 5;

```

```

SELECT ugyfel.id AS "ugyfel azonositoja",
    ugyfel.nev AS "ugyfel neve",
    SUM(utazas.ar+(ut_foglalas.utasszam-1)*utazas.ar*utazas.szorzo+ut_foglalas.pluszpenz) AS "Utazasokra koltott osszeg"
FROM ugyfel
RIGHT JOIN ut_foglalas ON ugyfel.id = ut_foglalas.ugyfel_id
LEFT JOIN utazas ON utazas.id = ut_foglalas.utazas_id
WHERE (
    utazas.ar + (ut_foglalas.utasszam-1)*utazas.ar*utazas.szorzo + ut_foglalas.pluszpenz) IN (
    SELECT (utazas.ar + (ut_foglalas.utasszam-1)*utazas.ar*utazas.szorzo + ut_foglalas.pluszpenz)
    FROM ut_foglalas JOIN utazas ON ut_foglalas.utazas_id = utazas.id
)
GROUP BY ugyfel.id
ORDER BY `Utazasokra koltott osszeg` DESC

```

```

SELECT iroda.id AS "Iroda azonositoja",
    iroda.nev AS "Iroda neve",
    SUM(utazas.ar+(ut_foglalas.utasszam-1)*utazas.ar*utazas.szorzo+ut_foglalas.pluszpenz) AS "atfolyt penz 2022-ben"
FROM iroda
RIGHT JOIN utazas ON iroda.id = utazas.iroda_id
LEFT JOIN ut_foglalas ON ut_foglalas.utazas_id = utazas.id
WHERE utazas.indulas LIKE "2022-%"
GROUP BY iroda.id;

```

```

CREATE TABLE iroda(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nev VARCHAR(50) NOT NULL,
    mobil VARCHAR(13) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    CONSTRAINT ir_jonev CHECK(LENGTH(nev)<=50),
    CONSTRAINT ir_jomobil CHECK(mobil LIKE '+%' AND LENGTH(mobil)=13),
    CONSTRAINT ir_joemail CHECK(email LIKE '%@%.%' AND LENGTH(email)<=100)
);

```

```

CREATE TABLE kirendeltseg(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nev VARCHAR(50) NOT NULL,
    mobil VARCHAR(13) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    orszag VARCHAR(50) NOT NULL,
    varos VARCHAR(50) NOT NULL,
    cim VARCHAR(100) UNIQUE NOT NULL,
    iroda_id INT NOT NULL,
    FOREIGN KEY (iroda_id) REFERENCES iroda(id) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT kir_jonev CHECK(LENGTH(nev)<=50),
    CONSTRAINT kir_jomobil CHECK(mobil LIKE '+%' AND LENGTH(mobil)=13),
    CONSTRAINT kir_joemail CHECK(email LIKE '%@%.%' AND LENGTH(email)<100),
    CONSTRAINT kir_joorszag CHECK(LENGTH(orszag)<=50),
    CONSTRAINT kir_jovaros CHECK(LENGTH(varos)<=50),
    CONSTRAINT kir_jocim CHECK(LENGTH(cim)<=100)
);

```

```

CREATE TABLE ugyfel(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nev VARCHAR(50) NOT NULL,
    mobil VARCHAR(13) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    szuletett DATE NOT NULL,
    CONSTRAINT ugyf_jonev CHECK(LENGTH(nev)<=50),
    CONSTRAINT ugyf_jomobil CHECK(mobil LIKE '+%' AND LENGTH(mobil)=13),
    CONSTRAINT ugyf_joemail CHECK(email LIKE '%@%.%' AND LENGTH(email)<100),
    CONSTRAINT ugyf_joszuldat CHECK(szuletett>1922/01/01)
);

```

```

CREATE TABLE vallalat(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nev VARCHAR(50) NOT NULL,
    tipus VARCHAR(50) NOT NULL,
    CONSTRAINT val_jonev CHECK(LENGTH(nev)<=50),
    CONSTRAINT val_jotipus CHECK(LENGTH(tipus)<=50)
);

```

```

CREATE TABLE szallas(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nev VARCHAR(50) NOT NULL,
    tipus VARCHAR(50) NOT NULL,
    ertekeles INT NOT NULL,
    CONSTRAINT szal_jonev CHECK(LENGTH(nev)<=50),
    CONSTRAINT szal_jotipus CHECK(LENGTH(tipus)<=50),
    CONSTRAINT szal_joert CHECK(ertekeles>=1)
);

```

```

CREATE TABLE egyuttmukodik(
    iroda_id INT NOT NULL,
    vallalat_id INT NOT NULL,
    FOREIGN KEY (iroda_id) REFERENCES iroda(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (vallalat_id) REFERENCES vallalat(id) ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (iroda_id, vallalat_id)
);

```

```

CREATE TABLE tud_foglalni(
    id INT PRIMARY KEY AUTO_INCREMENT,
    iroda_id INT NOT NULL,
    szallas_id INT NOT NULL,
    FOREIGN KEY (iroda_id) REFERENCES iroda(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (szallas_id) REFERENCES szallas(id) ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE utazas(
    id INT PRIMARY KEY AUTO_INCREMENT,
    uticel VARCHAR(100) NOT NULL,
    indulas DATE NOT NULL,
    idotartam INT NOT NULL,
    ar INT NOT NULL,
    szorzo FLOAT NOT NULL,
    iroda_id INT NOT NULL,
    em_iroda_id INT NOT NULL,
    em_vallalat_id INT NOT NULL,
    tipus VARCHAR(50) NOT NULL,
    helye VARCHAR(50) NOT NULL,
    FOREIGN KEY (iroda_id) REFERENCES iroda(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (em_iroda_id, em_vallalat_id) REFERENCES egyuttmukodik(iroda_id, vallalat_id) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT utazas_jocel CHECK(LENGTH(uticel)<=100),
    CONSTRAINT utazas_johossz CHECK(idotartam BETWEEN 2 AND 14),
    CONSTRAINT utazas_joszorzo CHECK(szorzo<=1),
    CONSTRAINT utazas_nemceges CHECK(em_iroda_id IS NULL OR em_vallalat_id IS NULL)
);

```

```

CREATE TABLE jarmu(
    nev VARCHAR(50) PRIMARY KEY,
    CONSTRAINT jarmu_jonev CHECK(LENGTH(nev)<=50)
);

```

```

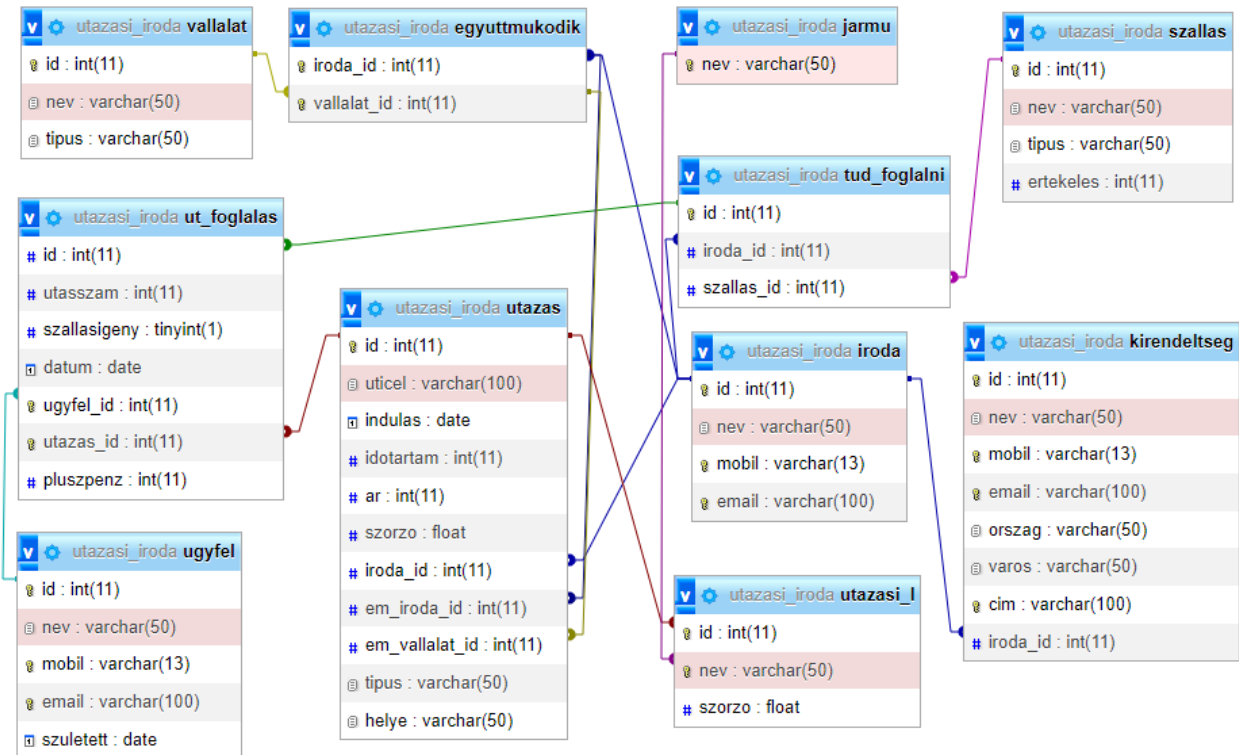
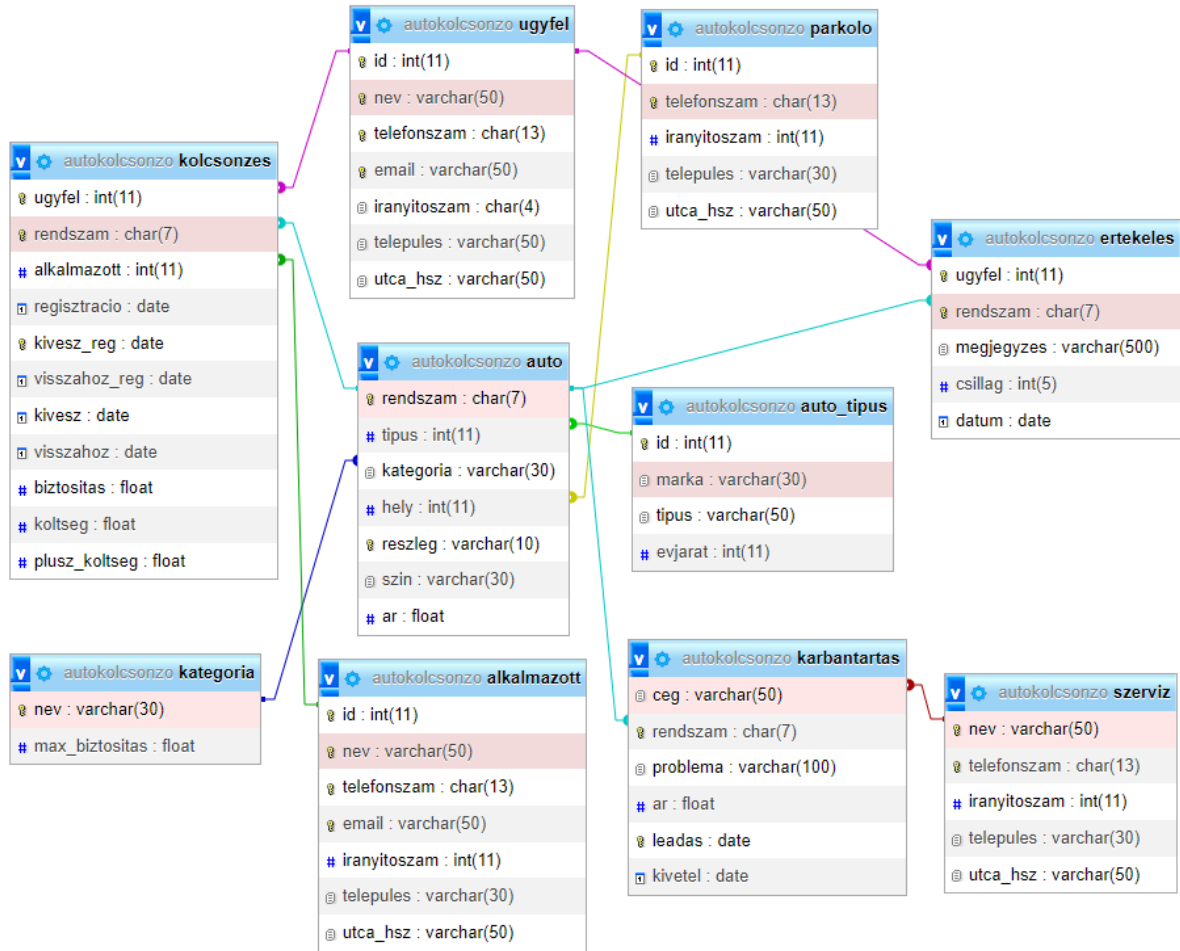
CREATE TABLE utazasi_l(
    id INT NOT NULL,
    nev VARCHAR(50) NOT NULL,
    szorzo FLOAT NOT NULL,
    FOREIGN KEY (id) REFERENCES utazas(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (nev) REFERENCES jarmu(nev) ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (id, nev),
    CONSTRAINT utleh_jonev CHECK(LENGTH(nev)<=50),
    CONSTRAINT utleh_joszorzo CHECK(szorzo>=1)
);

```

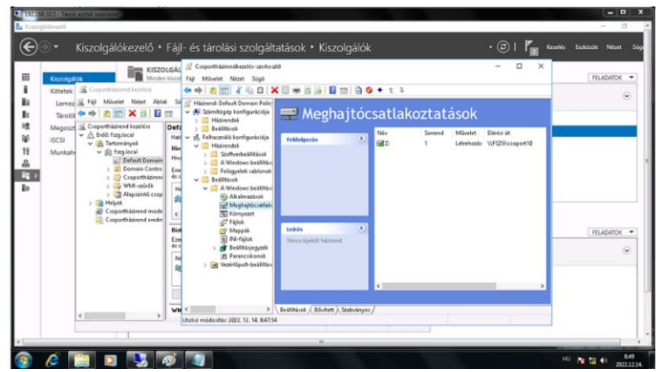
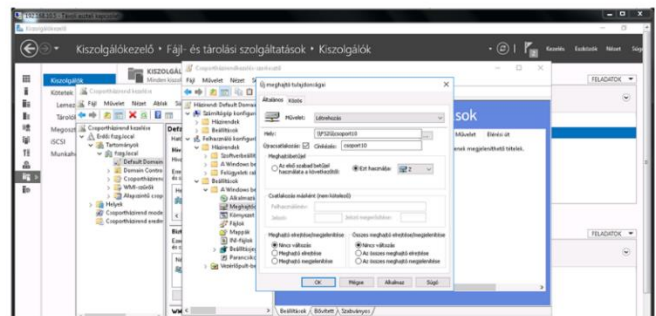
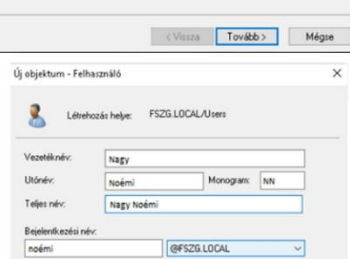
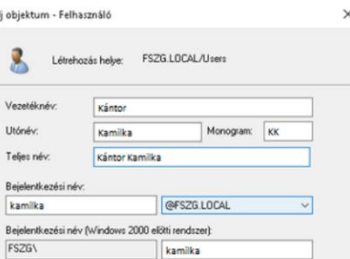
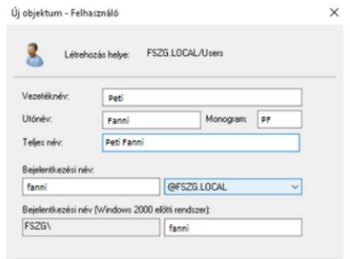
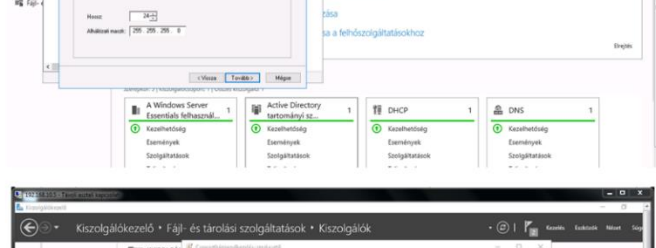
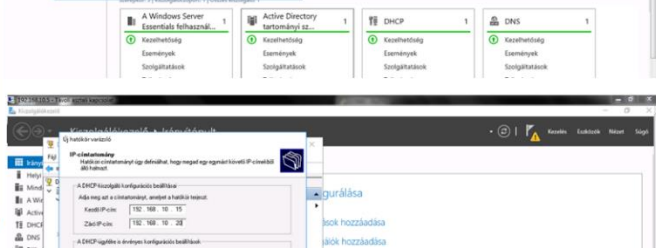
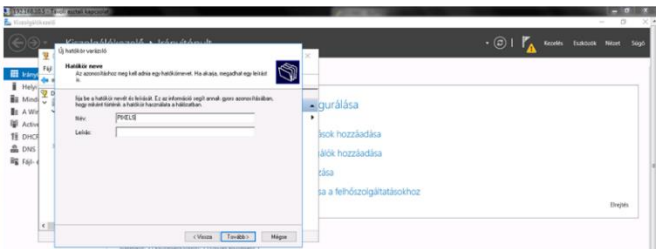
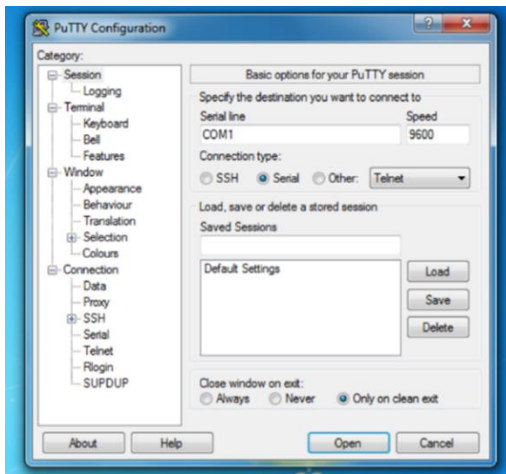
```

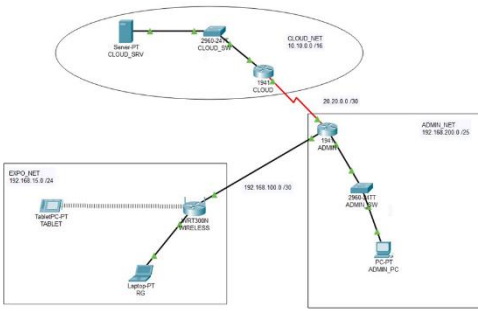
CREATE TABLE ut_foglalas(
    id INT NOT NULL,
    utasszam INT NOT NULL,
    szallasiigeny BOOLEAN NOT NULL,
    datum DATE NOT NULL,
    ugyfel_id INT NOT NULL,
    utazas_id INT NOT NULL,
    pluszpenz INT NOT NULL,
    FOREIGN KEY (ugyfel_id) REFERENCES ugyfel(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (utazas_id) REFERENCES utazas(id) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id) REFERENCES tud_foglalni(id) ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (ugyfel_id, utazas_id),
    CONSTRAINT utfogl_jousz CHECK(utasszam>=1),
    CONSTRAINT utfogl_jopp CHECK(pluszpenz>=0)
);

```



Из виконаних звітів робіт студентів по побудові та налаштуванню мережі:





CLOUD_SW

Physical Config CLI Attributes

GLOBAL Settings

Algorithm Settings

SWITCHING

VLAN Database

INTERFACE

FastEthernet0/1
FastEthernet0/2
FastEthernet0/3
FastEthernet0/4
FastEthernet0/5
FastEthernet0/6
FastEthernet0/7
FastEthernet0/8
FastEthernet0/9
FastEthernet0/10
FastEthernet0/11

NVRAM

Startup Config Load Export Merge...

Running Config Load Export Merge...

FastEthernet0/24: Line protocol on Interface Vlan1, changed state to up
FastEthernet0/24: Line protocol on Interface FastEthernet0/24, changed state to up
FastEthernet0/24: Line protocol on Interface FastEthernet0/24, changed state to up

CLOUD_SRV

Physical Config Services Desktop Programming Attributes

SERVICES

FTP On Off

HTTP On Off

DHCPv6 On Off

DNS On Off

AAA On Off

NTP On Off

EMAIL On Off

VM Management On Off

Radius EAP On Off

File

1 asa842-k8.bin
2 asa823-k8.bin
3 c:\1841-adv\services6-mz-124-15...

ADMIN_PC

Physical Config Desktop Programming Attributes

Interface FastEthernet0

IP Configuration

DHCP Static DHCP request successful.

IP Address 192.168.200.4
Subnet Mask 255.255.255.128
Default Gateway 192.168.200.126
DNS Server 30.30.30.30

IPv6 Configuration

Static Auto Config

IPv6 Address FE80:24D:BFF:FE08:85AB
IPv6 Gateway
IPv6 DNS Server 802:IX

TABLET

Physical Config Desktop Programming Attributes

Network Status

Interface IP Address 192.168.100.1
Subnet Mask 255.255.255.0
Default Gateway 192.168.100.1
DNS 1 (Optional) 0.0.0.0
DNS 2 (Optional) 0.0.0.0

Network Table

Router IP	IP Address	Subnet Mask	Next Hop	Next Hop IP
192.168.100.1	192.168.100.1	255.255.255.0	0.0.0.0	0.0.0.0

DHCP Server Settings

DHCP Enabled Disabled DHCP Reservation

IP Address 192.168.100.1
Subnet Mask 255.255.255.0
Default Gateway 192.168.100.1
DNS 1 (Optional) 0.0.0.0
DNS 2 (Optional) 0.0.0.0

TABLET

Physical Config Desktop Programming Attributes

Command Prompt

```

Volume in drive C has no label.
Volume Serial Number is 5E12-4AF3
Directory of C:\

1/1/1970  1:0 PM                5571584  asa842-k8.bin
1/1/1970  1:0 PM                   26      sampleFile.txt
           5571610 bytes    2 File(s)

C:\>ping 10.10.0.254

Pinging 10.10.0.254 with 32 bytes of data:

Reply from 10.10.0.254: byte=32 time=27ms TTL=125
Reply from 10.10.0.254: byte=32 time=12ms TTL=125
Reply from 10.10.0.254: byte=32 time=8ms TTL=125
Reply from 10.10.0.254: byte=32 time=14ms TTL=125

Ping statistics for 10.10.0.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximately round trip time in milli-seconds:
        Minimum = 8ms, Maximum = 27ms, Average = 15ms
  
```

TABLET

Physical Config Desktop Programming Attributes

Wireless0 On Off

Port Status

Bandwidth 300 Mbps

MAC Address 0008:8A0C:6353

SSID EXPO

Authentication

Disabled WPA2-PSK WEP Key

WPA WPA2 PSK Pass Phrase Exp0201*

802.1X Method

Encryption Type

Static DHCP

IP Address 192.168.0.100
Subnet Mask 255.255.255.0

IPv6 Configuration

DHCP Auto Config Static

TABLET

Physical Config Desktop Programming Attributes

Command Prompt

```

Packet Tracer PC Command Line 1.0
C:\>ping 192.168.200.4

Pinging 192.168.200.4 with 32 bytes of data:

Reply from 192.168.200.4: byte=32 time<1ms TTL=126
Reply from 192.168.200.4: byte=32 time<1ms TTL=126
Reply from 192.168.200.4: byte=32 time<1ms TTL=126
Reply from 192.168.200.4: byte=32 time<1ms TTL=126

Ping statistics for 192.168.200.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximately round trip time in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

TABLET

Physical Config Desktop Programming Attributes

Command Prompt

```

Packet Tracer PC Command Line 1.0
C:\>ftp
Packet Tracer PCftp
C:\>ftp 10.10.0.254
Trying to connect...10.10.0.254
Connected to 10.10.0.254
230 Welcome to FT FTP server
User name (username):
331 Username ok, need password
Password:
330 Logged in
ftp>dir
Listing /ftp directory from 10.10.0.254:
Error ftp://10.10.0.254/ (No such file or directory or permission denied)
350 Requested action not taken, permission denied.
ftp>get asa842-k8.bin
Reading file asa842-k8.bin from 10.10.0.254:
File transfer in progress...
  
```

Захист виконаних звітів на онлайн конференції

Roland Gazdag prezentál

2. Vállalat hálózata

8:16 | qsw-dkow-qwk

Martin Barkaszi prezentál

8:23 | qsw-dkow-qwk

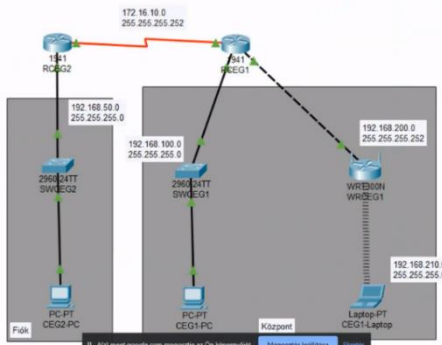
Fanni Peti prezentál

- SOHO Router, melyet Access Point-ként használunk
- Cisco Router
- Cisco Switch
- PC

8:27 | qsw-dkow-qwk

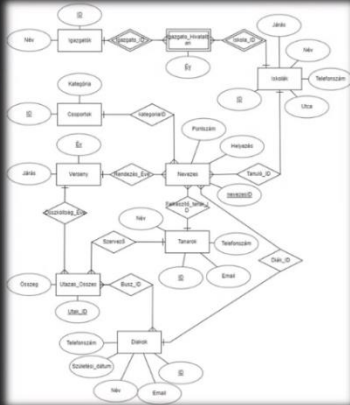
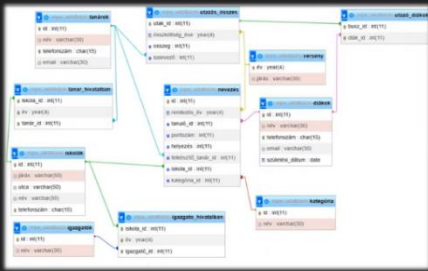
Fanni Peti prezentál

A hálózat topológiája



8:28 | qsw-dkow-qwk

Roland Gazdag prezentál



9:11 | qsw-dkow-qwk

József Kovács prezentál

EK modell és SQL séma

```
1 CREATE DATABASE onlinebolto;
2 USE onlinebolto;
3
4 CREATE TABLE auto(
5     rendszam CHAR(7) PRIMARY KEY,
6     tipus INT,
7     kategoria VARCHAR(30),
8     hely INT,
9     evkezd INT(4) NOT NULL CHECK(
10     evkezd >= YEAR(NOW()) NOT NULL,
11     evkezd <= YEAR(NOW()) + 1 NOT NULL),
12     ar FLOAT NOT NULL,
13     FOREIGN KEY(tipus) REFERENCES auto_tipus(id),
14     FOREIGN KEY(kategoriak) REFERENCES kategoria(nev),
15     FOREIGN KEY(hely) REFERENCES parhuzos(id),
16     CONSTRAINT rendszam CHECK(LEFT(rendszer, 1) <> 0);
17
18
19
20
21 CREATE TABLE auto_tipus(
22     id INT PRIMARY KEY AUTO_INCREMENT,
23     marka VARCHAR(30) NOT NULL,
24     tipus VARCHAR(50) NOT NULL,
25     evjarat INT NOT NULL,
26
27
28
29
30 CREATE TABLE kategoria(
31     nev VARCHAR(30) PRIMARY KEY,
32     max_dirojtas FLOAT NOT NULL,
33
34
35
36
37 CREATE TABLE parhuzos(
38     id INT AUTO_INCREMENT PRIMARY KEY,
39     nev VARCHAR(50) NOT NULL CHECK(
40     nev <> ''),
41     hallyal VARCHAR(50),
42     telephely VARCHAR(50),
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

9:30 | qsw-dkow-qwk

László Somi prezentál

```

DROP DATABASE IF EXISTS Utazasi_Iroda;
CREATE DATABASE Utazasi_Iroda CHARACTER SET 'utf8mb4' COLLATE 'utf8mb4_unicode_ci';
USE Utazasi_Iroda;

CREATE TABLE Iroda(
  id INT PRIMARY KEY AUTO INCREMENT,
  nev VARCHAR(50) NOT NULL,
  mobil VARCHAR(15) UNIQUE NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  CONSTRAINT ir_jonev CHECK (LENGTH(nev)<=50),
  CONSTRAINT ir_jomobil CHECK (mobil LIKE '+%' AND LENGTH(mobil)=15),
  CONSTRAINT ir_joemail CHECK (email LIKE '%@%' AND LENGTH(email)<=100)
);

CREATE TABLE Kirendeltseg(
  id INT PRIMARY KEY AUTO INCREMENT,
  nev VARCHAR(50) NOT NULL,
  mobil VARCHAR(15) UNIQUE NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  orszag VARCHAR(50) NOT NULL,
  varos VARCHAR(50) NOT NULL,
  cim VARCHAR(100) UNIQUE NOT NULL,
  Iroda_id INT NOT NULL,
  FOREIGN KEY (Iroda_id) REFERENCES Iroda(id) ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT kir_jonev CHECK (LENGTH(nev)<=50),
  CONSTRAINT kir_jomobil CHECK (mobil LIKE '+%' AND LENGTH(mobil)=15),
  CONSTRAINT kir_joemail CHECK (email LIKE '%@%' AND LENGTH(email)<=100),
  CONSTRAINT kir_joorszag CHECK (LENGTH(orszag)<=50),
  CONSTRAINT kir_jovaros CHECK (LENGTH(varos)<=50),
  CONSTRAINT kir_jocim CHECK (LENGTH(cim)<=100)
);

CREATE TABLE Ugyfel(

```

10:03 | qsw-dkow-qwk

Alexandra Hapák prezentál

• Példa több-a-többhöz kapcsolatra

```

CREATE TABLE AtvetettAru(
  atvetelID INT,
  aruID INT,
  mennyiseg FLOAT,
  PRIMARY KEY (atvetelID, aruID),
  CONSTRAINT atvetett_aru_atvetel_fk FOREIGN KEY(atvetelID)
    REFERENCES Atvetel(atvetelID) ON UPDATE CASCADE,
  CONSTRAINT atvetett_aru_aru_fk FOREIGN KEY(aruID)
    REFERENCES BeszallitottAru(aruID) ON UPDATE CASCADE
);

```

SQL megvalósítása egy AtvetettAru tábla létrehozásával történik:

10:10 | qsw-dkow-qwk