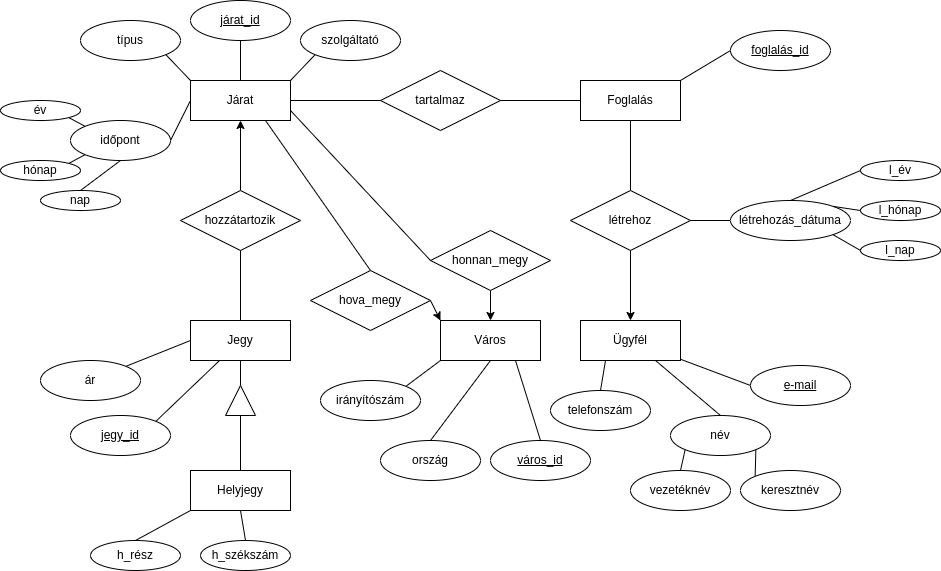
Busz, vasút, repülő helyfoglalás

Adatbázisok kötelező feladat

Napjainkban általános igény, hogy problémamentesen hozzáférhessünk az utazási igényeinket kielégítő foglalási felületekhez. A Travall adatbázis-alkalmazás ezt a vágyat hivatott teljesíteni, oly módon, hogy ezt mind egy helyen tehessük meg. Legyen szó busz-, vasút- vagy repülőutazásról, a platformon akármelyikhez van lehetőség hozzáférni. Nem létezik korlátozás egy szolgáltatóra, egy sokkal általánosabb megoldást kísérel adni a problémára: a különböző szolgáltatók járataira egy helyen lehessen jegyet foglalni. Egyaránt lehetséges egyszerű jegyet, valamint helyjegyet foglalni, megfelelően listázni a például a célállomások alapján, hogy a legoptimálisabb járathoz férhessünk hozzá.

# Egyed-kapcsolat modell



Az Ügyfélnek néhány adata lett modellezve, mely alapján egyértelműen beazonosítható. Feltétel, hogy az adatbázisban egy e-mailhez egy ügyfél tartozhat. Az ügyfelek létre tudnak hozni tetszőleges számú foglalást, melyhez tetszőleges számú járatok tartoznak, egy foglalás több járatot is tartalmazhat. A járatok, foglalások, városok, jegyek mind egyedi azonosítóval lettek

ellátva a könnyebb kezelhetőség érdekében. Egy városba több járat is érkezhet és mehet. A járat típusa határozza meg, hogy busz-, repülő- vagy vasúti utazásról van szó. Minden járathoz jegyek tartoznak, melyek lehetnek egyszerű jegyek is, helyfoglalás nélkül (ez talán a busz és vonatjegyekre lehet jellemző), viszont helyjegyek is rendelve vannak a járatokhoz (melyek például repülőjárat esetében kötelezőnek mondhatóak).

# Relációs adatbázisséma

ÜGYFÉL (e-mail, vezetéknév, keresztnév, telefonszám)

FOGLALÁS (foglalás\_id, *e-mail*, l\_év, l\_hónap, l\_nap)

VÁROS (város\_id, irányítószám, ország)

JÁRAT (járat\_id, típus, szolgáltató, év, hónap, nap, *honnan\_város\_id, hova\_város\_id*)

JEGY (jegy\_id, ár, h\_rész, h\_székszám, *járat\_id*)

TARTALMAZ (*foglalás\_id, járat\_id*)

Megjegyzés(ek): A helyjegy nem kapott külön sémát, ugyanis előreláthatóan az adatbázis által kezelt legtöbb jegy helyjegy, ezért relatív kevés NULL mező keletkezik.

Az attribútumok doméniumai (absztrakt adattípusokban – típus, hossz):

**ÜGYFÉL**: e-mail (string, 80); vezetéknév (string, 50); keresztnév (string,50); telefonszám (string, 20)

**FOGLALÁS**: foglalás\_id (integer, 8); e-mail (string, 80); l\_év (integer, 4); l\_hónap (integer, 2); l\_nap (integer, 2)

**VÁROS**: város\_id (integer, 8); irányítószám (string, 10); ország (string, 56)

**JÁRAT**: járat\_id (integer, 8); típus (string, 6); szolgáltató (string, 50); év (integer, 4); hónap (integer, 2); nap (integer, 2); honnan\_város\_id (integer, 8); hova\_város\_id (integer, 8)

**JEGY**: jegy\_id (integer, 12); ár (integer, 8); h\_rész (character, 1); h\_székszám (integer, 4); járat\_id (integer, 8)

**TARTALMAZ**: foglalás\_id (integer, 8); járat\_id (integer, 8)

Megjegyzés(ek): irányítószámnál a leghosszabb postal code 10 jegy hosszú (Iráni Iszlám Köztársaság), míg az amerikai ZIP+4 rendszerben 9 jegy és egy kötőjel; a leghosszabb ország hivatalos neve The United Kingdom of Great Britain and Northern Ireland (56 hosszú).

# Funkcionális függőség, kulcs

ÜGYFÉL (e-mail, vezetéknév, keresztnév, telefonszám)

{e-mail} → {vezetéknév, keresztnév, telefonszám}

Az {e-mail, ...} halmaztól az ÜGYFÉL - {e-mail, ...} mindig függeni fog. Más függőség nincs.

Az {e-mail} egyelemű szuperkulcs, tehát ez lesz a kulcs is és más kulcs nincs.

FOGLALÁS (foglalás\_id, *e-mail*, l\_év, l\_hónap, l\_nap)

{foglalás\_id} → {e-mail, l\_év, l\_hónap, l\_nap}

Hasonlóan az előzőhöz, {foglalás\_id, ...} halmaztól függ minden más.

A kulcs – az előzőhöz hasonlóan – itt is a {foglalás\_id}.

VÁROS (város\_id, irányítószám, ország)

{város\_id} → {irányítószám, ország}

{ország} → {irányítószám}, ha feltételezzük, hogy az országok egyértelműen meghatározzák az irányítószámokat, tehát nincs két egyforma irányítószám két különböző országban. Viszont ez [nem feltétlenül teljesül](https://www.quora.com/Do-the-same-ZIP-codes-exist-in-different-countries). Ezért {ország} -/-> {irányítószám}. Hasonlóan lehet kijelenteni ezt {város\_id} -/-> {irányítószám} esetében is.

Ahogyan eddig is, a {város\_id, ...} halmaztól függ minden más és {város\_id} a kulcs.

JÁRAT (járat\_id, típus, szolgáltató, év, hónap, nap, *honnan\_város\_id, hova\_város\_id*); JEGY (jegy\_id, ár, h\_rész, h\_székszám, *járat\_id*)

Ezekben a sémákban az előzőekhez hasonlóan az aláhúzott kulcsattribútum egyértelműen a kulcs, és attól – vagy annak a kiegészítésétől (uniója más attribútum(okkal)) – függ az összes többi.

TARTALMAZ (*foglalás\_id, járat\_id*)

Nem tartalmaz funkcionális függést. A két attribútum alkotja a kulcsot.

|  |  |
| --- | --- |
| Séma | Kulcs |
| ÜGYFÉL | e-mail |
| FOGLALÁS | foglalás\_id |
| VÁROS | város\_id |
| JÁRAT | járat\_id |
| JEGY | jegy\_id |
| TARTALMAZ | foglalás\_id, járat\_id |

# Normalizálás

**1NF**

Az összes séma megfelel az első normálforma előírásainak, hiszen leképezés után minden attribútom doméniuma atomi elemekből áll.

**2NF**

Az ÜGYFÉL séma 2NF-ben van, mert minden – az egyetlen – kulcs egy attribútumból áll.

Hasonló jelenthető ki a FOGLALÁS, VÁROS, JÁRAT és JEGY sémákról is.

A TARTALMAZ szintén megfelel a második normálformának, hiszen nem tartalmaz másodlagos attribútumot.

**3NF**

A TARTALMAZ sémában nincs másodlagos attribútum, ezért megfelel a harmadik normálformának.

Az ÜGYFÉL, FOGLALÁS, VÁROS, JÁRAT és JEGY sémák esetén kijelenthető, hogy nincs bennük tranzitív függés, ezért lesznek 3NF-ben.

# Táblatervek

TODO

# Összetet lekérdezések

TODO

# Megvalósítás, funkciók

TODO

Megjegyzés: Ha valami nem megfelelően jelenne meg, a [projekthez tartozó github repository](https://github.com/a-bakk/db) docs mappájában minden – a dokumentációhoz tartozó – rész megtalálható.