**OE NIK 2013**

**Táborhely Üzemeltető segédszoftver**

**Teljes dokumentáció**

**Lovas István K971UM**

# Tartalom Jegyzék:

Táborhely Üzemeltető segédszoftver dokumentáció.

# A Szoftver céljai és megszorításai

A web-es adatbázis-alkalmazás célja az, hogy a tábor személyzetének megkönnyítse a tábor lakóival kapcsolatos adminisztrációt. A nyilvántartásban tárolja a lakók, a tábori csoportok, illetve a tábori férőhelyekkel kapcsolatos legfontosabb adatokat. A rendszer adminisztrátorának, az SQL futtatását megkönnyítő szakasszal is rendelkezik, illetve egy olyan részegységgel, mely információt biztosít a rendszer környezetének OpenCl képességeiről.

# A Szoftverrel kapcsolatos technikai elvárások és megszorítások

Az elkészítés során a következő nyelveket és technológiákat alkalmaztam:

* Az alkalmazás az Oracle11g APEX környezetének felhasználásával készült.
  + Oracle11g:
    - Egy nagyvállalati környezetbe készült, és ilyen környezetben közkedvelt adatbázis kezelő eszköz.
  + APEX (Application Express):
    - Az Oracle RAD eszköze, mely adatbázis intenzív webes alkalmazások egyszerű elkészítésére hivatott.
* SQL - Structured Query Language (strukturált lekérdezőnyelv):
  + Mely egy olyan deklaratív nyelv mely az adatbázis kezelés megkönnyítésére hivatott.
* PL/SQL - Procedural Language/Structured Query Language:
  + Az Oracle által kifejlesztett programozási nyelv, mely az SQL procedurális eszközökkel való kiegészítése.
* Java:
  + Egy olyan általános célú, objektumorientált programozási nyelv, aminek tervezése és megalkotása során nagy figyelmet fordítottak a robosztus, hibatűrő, biztonságos működésre, és hogy lehetőleg kevés ráfordítással lehessen, könnyen hordozható, platform független, vállalati szintű programokat előállítani a segítségével.
  + A fenti előnyei, és a tény, hogy jól együttműködik az Oracle adatbázis kezelő rendszerrel, kiváló választássá tette az adott feladatra.
* Aparapi:
  + Egy olyan Java nyelvi programozási felület (API), mely segíti a külső gyorsító eszközök programozását.
  + Tervezésénél ügyeltek arra, hogy használata ne legyen idegen egy Java programozó számára. Ezt olyannyira sikeresen teljesítették, hogy egy egyszerű kliens megírása nem sokkal bonyolultabb egy Java szál indításánál.
  + Az Oracle bejelentette, hogy a Java 9 OpenCl képességeit az Aparapi mintájára, annak eredményeit felhasználva kívánják implementálni.
  + Az Aparapi ezért egy nagyon előremutató és jó eszköz a kívánt cél elérésére.
* NetBeans:
  + Egy nyílt forráskódú, de az Oracle által is koordinált fejlesztésű IDE, a Java kódok előállítását segítette.

# A Szoftver beüzemelése és tesztadatokkal való feltöltése

A program beüzemelésének lépései:

* A program vázának elkészítése az f102.sql APEX-be való importálásával kezdődik.
  + f102.sql
* A táblák elkészítése:
  + Csak scriptek segítségével:
    1. 1a\_create\_tables.sql
    2. 1b\_alter\_constraints.sql
    3. 2a\_init\_test\_Házak.sql
    4. 2b\_init\_test\_Csoportok.sql
    5. 2c\_init\_test\_Lakók.sql
  + Az APEX SQL Workshop > Utilities > Data Workshop eszközével
    1. Data Load > Text Data eszközzel importálandó fájlok:
       - csoportok.txt
       - házak.txt
       - lakók.txt
    2. 1b\_alter\_constraints.sql script futtatása, a megszorítások létrehozása végett
* A Java nyelvű külső rutin beüzemelése:

Az összes fent említett script, és importálandó fájl megtalálható a   
CD mellékélet gyökér\Scriptek\_Programok\_Fájlok\ mappában.

# Jogosultsági szintek a rendszerben

# Betétlap

BETÉTLAP

Lovas István Féléves Feladatához

1. Autentikációs adatok:

- Adminisztrátor / adminisztrátor

- dolgozó / dolgozó

2. A Féléves Feladat kiemelkedő képességei:

- Gyorsító eszközök Java nyelvből történő elérése

# Táblaszerkezet tervezése:

## A feladat rövid leírása

* Egy ifjúsági tábor lakóival kapcsolatos fontosabb adatok tárolása.

## A tárolandó tulajdonságok leírása

|  |  |
| --- | --- |
| **A tárolandó adat** | **Rövid leírása** |
| név | a lakó neve |
| SzületésiDátum | a lakó születési dátuma |
| nem | a lakó neme |
| egészségbiztosítási\_szám | a lakó egészségbiztosítási száma |
| személyigazolvány\_szám | a lakó személyigazolvány száma |
| tartózkodas\_tól | az a dátum amikortól az adott lakó a táborban tartózkodik |
| tartózkodas\_ig | az a dátum ameddig az adott lakó a táborban tartózkodik |
| ország | a lakó származási országa |
| ház\_szám | a ház száma,a táboron belül a házak egytől számozódnak |
| ház\_megnevezés | a ház neve, a táboron belül lehet több azonos nevű ház is. |
| férőhely\_szám | egy szám, mely megmondja, hogy az adott ház hány személyt képes befogadni |
| csoport\_név | a tábor lakói csoportokat képeznek, táboron belül több csoportnak is lehet azonos a neve |
| csoportvezető | minden csoportnak csak egy vezetője van, de egy vezetőnek lehet több csoportja |
| csoport\_pontszám | egy-egy csoport által, a csapatversenyeken elért, összpontszámát tárolja |

## Az ősmodell:

* Lakók< név, SzületésiDátum, nem, egészségbiztosítási\_szám, személyigazolvány\_szám, tartózkodas\_tól, tartózkodas\_ig, ország, ház\_szám, ház\_megnevezés, férőhely\_szám, csoport\_név, csoportvezető, csoport\_pontszám >

## Függőségek:

1. Egy lakó pontos meghatározásához szükséges a neve és az egészségbiztosítási száma. **flakók** ={név, egészségbiztosítási\_szám } -> { SzületésiDátum, nem, személyigazolvány\_szám, tartózkodas\_tól, tartózkodas\_ig, ország, ház\_szám, ház\_megnevezés, férőhely\_szám, csoport\_név, csoportvezető, csoport\_pontszám }
2. Egy Csoport egyértelmű meghatározásához szükséges, a csoport neve, illetve a vezetője.  
   **fcsoportok** = { csoport\_név, csoportvezető } -> { csoport\_pontszám }
3. A házszámok különbözőek, ezért a házszámból egyértelműen képes megállapítani a rendszer, hogy melyik házról van szó. Megengedjük viszont, hogy több ház viselje ugyanazt a nevet.  
   **fházak** = { ház\_szám } -> { ház\_megnevezés, férőhely\_szám }

## A függőségek normalizálása

1. **flakók** normalizálása:
   1. 1NF-ben van, mert nincs összetett attribútuma
   2. 2NF-ben van, nincs olyan másodlagos attribútum, aminek a meghatározásához elegendő lenne az elsődleges attribútumok egy része.
   3. Nincs 3NF-ben, hisz **fcsoportok** és **fházak** belső függések, ezeket kiemelve:  
      **flakók1** = {név, egészségbiztosítási\_szám } -> { SzületésiDátum, nem, személyigazolvány\_szám, tartózkodas\_tól, tartózkodas\_ig, ország, ház\_szám, csoport\_név, csoportvezető }
   4. BCNF-ben van, hisz 3NF-ben van, és a baloldal egyetlen valódi részhalmaza sem függ másik elsődleges, vagy másodlagos attribútumtól.
2. **fcsoportok** normalizálása:
   1. 1NF-ben van, hisz nincs összetett attribútuma.
   2. 2NF-ben van, hisz a jobb oldal egy attribútumból áll, és az függ mindkét elsődleges attribútumtól.
   3. 3NF-ben van, hisz a jobb oldal csak egy elemből áll.
   4. BCNF-ben van, hisz az elsődleges attribútumok nem függnek egymástól, vagy a másodlagos attribútumtól.
3. **fházak** normalizálása:
   1. 1NF-ben van, hisz nincs összetett attribútuma.
   2. 2NF-ben van, hisz a baloldal csak egyetlen attribútumból áll.
   3. 3NF-ben van, hisz a másodlagos attribútumok között nincs összefüggés.
   4. BCNF-ben van, hisz 3NF-ben van, és a bal oldalán csak egyetlen attribútum áll.

## Az eredményfüggőségek összefoglalása

* **flakók1** = { név, egészségbiztosítási\_szám } -> { SzületésiDátum, nem, személyigazolvány\_szám, tartózkodas\_tól, tartózkodas\_ig, ország, ház\_szám, csoport\_név, csoportvezető }
* **fcsoportok** = { csoport\_név, csoportvezető } -> { csoport\_pontszám }
* **fházak** = { ház\_szám } -> { ház\_megnevezés, férőhely\_szám }

## Az eredmény adatmodell felírása

* Lakók < név, egészségbiztosítási\_szám, SzületésiDátum, nem, személyigazolvány\_szám, tartózkodas\_tól, tartózkodas\_ig, ország, *ház\_szám*, *csoport\_név, csoportvezető* >
  + Ahol a név és egészségbiztosítási\_szám attribútumok,   
    tehát a **Klakók** = { név, egészségbiztosítási\_szám } attribútum halmaz a Lakók relációra vonatkozó elsődleges kulcs.
  + A *ház\_szám attribútum a házak relációra vonatkozó idegen kulcs.*
  + *A* *csoport\_név és csoportvezető attribútumok pedig a csoportok relációra vonatkozó idegen kulcsok.*
* Csoportok *<*csoport\_név, csoportvezető, csoport\_pontszám>
  + A **Kcsoportok** = { csoport\_név, csoportvezető } attribútum halmaz a Csoportok relációra vonatkozó elsődleges kulcs.
* Házak < ház\_szám, ház\_megnevezés, férőhely\_szám >

A **Kházak** = { ház\_szám } attribútum halmaz a Házak relációra vonatkozó elsődleges kulcs.

# Táblák Implementációja

## A táblák szerkezete

A táblák tervben foglaltak szerint készültek.

A kívánt szöveges és szám mezők meghatározásakor, ahol tudtam, a lehető legnagyobb, még ésszerű foglalási méretet választani. Ezt néhol kutatás is megelőzte, például az ország név mező hosszának meghatározásakor számításba vettem ilyeneket, mint: The United Kingdom of Great Britain and Northern Ireland, mely 48 karakter hosszú. Az ország mezőt 50 hosszú VARCHAR2-ként vettem fel. A VARCHAR2 típusról tudni kell, hogy nem foglalódik le alapból a teljes meghatározott karakternyi méret (előző példában 50), ezért úgy láttam, érdemesebb túltervezni, ahol csak lehet. Az Egészségbiztosítási\_szám, illetve a Személyigazolvány\_szám fix hosszúságú szövegek, ennek megfelelő CHAR típusnak határoztam meg mind a kettőt. A szám típusoknál, mint a ház\_szám, illetve a férőhely\_szám három helyértéken tárol, a csoport pontszám egy tizedes jegy pontossággal, és öt egész helyértékkel tárol, ami a táborban minden bizonnyal elég lesz.

### A megvalósító script: 1a\_create\_tables.sql

ALTER SESSION SET NLS\_LANGUAGE= 'HUNGARIAN';

ALTER SESSION SET NLS\_TERRITORY= 'HUNGARY';

ALTER SESSION SET NLS\_DATE\_FORMAT= 'YYYY-MM-DD';

ALTER SESSION SET NLS\_DATE\_LANGUAGE = 'HUNGARIAN';

ALTER SESSION SET NLS\_ISO\_CURRENCY = 'HUNGARY';

ALTER SESSION SET NLS\_NUMERIC\_CHARACTERS = ',.';

ALTER SESSION SET NLS\_SORT = 'HUNGARIAN';

--SELECT \* FROM nls\_session\_parameters;

DROP TABLE Lakók;

CREATE TABLE Lakók (

Név VARCHAR2(30),

Egészségbiztosítási\_szám CHAR(9) UNIQUE,

SzületésiDátum DATE,

Nem CHAR(1) CHECK( Nem IN ('f','n') ),

Személyigazolvány\_szám CHAR(11) UNIQUE,

Tartózkodas\_tól DATE,

Tartózkodas\_ig DATE,

Ország VARCHAR2(50),

ház\_szám NUMBER(4),

Csoport\_név VARCHAR2(50),

Csoportvezető VARCHAR2(30)

);

DROP TABLE Csoportok;

CREATE TABLE Csoportok(

csoport\_név VARCHAR2(50),

csoportvezető VARCHAR2(30),

csoport\_pontszám NUMBER(5,1)

);

DROP TABLE Házak;

CREATE TABLE Házak (

ház\_szám NUMBER(3),

ház\_megnevezés VARCHAR2(25) NOT NULL,

férőhely\_szám NUMBER(3) NOT NULL

);

A fentiekben látható, hogy a Lakók táblában az Egészségbiztosítási\_szám, amellett, hogy az elsődleges kulcs része, még egyedinek is jelöltem. Egyedi továbbá a Személyigazolvány\_szám mező is. Az egészségbiztosítási szám, mivelhogy az elsődleges kulcs része, kötelező a kitöltése is. A Személyigazolvány\_szám kitöltése ellenben nem kötelező, hisz lehetséges, hogy kiskorú személy is tartózkodik a tábor területén.

## A táblák megszorításai

A terv alapján megalkottam az elsődleges, illetve az idegen kulcsokat.

### A megvalósító script: 1b\_alter\_constraints.sql

ALTER TABLE Lakók ADD CONSTRAINTS p\_Lakók PRIMARY KEY (Név,Egészségbiztosítási\_szám);

ALTER TABLE Csoportok ADD CONSTRAINTS p\_Csoportok PRIMARY KEY (csoport\_név,csoportvezető);

ALTER TABLE Házak ADD CONSTRAINTS p\_Házak PRIMARY KEY (ház\_szám);

ALTER TABLE Lakók ADD CONSTRAINTS f\_Lakók\_Csoportok FOREIGN KEY (Csoport\_név,Csoportvezető)

REFERENCES Csoportok (csoport\_név,csoportvezető);

ALTER TABLE Lakók ADD CONSTRAINTS f\_Lakók\_Házak FOREIGN KEY (ház\_szám)

REFERENCES Házak (ház\_szám);

# Implementáció részleteinek leírása

Általános megjegyzések

## Kezdőlap

### A komponens célja

A kezdőlap általános leírást nyújtani a webalkalmazás céljairól, és használatáról.

A kezdőoldal bárki számára látható.

### A megvalósításának fontosabb részletei

Az információt HTML egyszerű régiókban tárolja.  
Azt, hogy az oldalt bárki láthassa a Kezdőlap > Page Attributes > Security > Authentication választómezőjének Page Is Public értékre állításával értem el.

### Futási eredménye

## Táblák Megtekintése

### A komponens célja

Könnyen kezelhető eszköz a rendszerben szereplő táblák megjelenítésére.

Táblaválasztó súgószövege:  
Válasszon táblát, majd jelölje be a megjelenítendő oszlopokat, és ha szükséges, megadhat egy oszlopot, mely szerint az adatokat rendezi a rendszer.

Csak dolgozók láthatják.

### A megvalósításának fontosabb részletei

* A táblát egy legördülő listából lehet választani
  + értéklista lekérdezése:  
    select distinct table\_name, table\_name as "a" from user\_tables  
    where table\_name not like 'APEX%' AND table\_name not like 'DEMO%';
* A tábla kiválasztása után az adott tábla oszlopait checkbox segítségével lehet kiválasztani:
  + értéklista lekérdezése:  
    select column\_name as val, column\_name from all\_tab\_columns where table\_name = :P2\_TV
* A rendezést rádiógomb segítségével lehet kiválasztani:
  + értéklistája:  
    STATIC:Növekvő;ASC,Csökkenő;DESC,Nincs;NO
* A rendező oszlopot egy automatikus szövegkiegészítést nyújtó szövegdobozban lehet megadni
  + értéklista lekérdezése: megegyezik a checkbox lekérdezésével.
* Az oldal elküldése után két validáció futhat le, az egyik a kiválasztott tábla változtatásánál nulláz minden attól függő mezőt, míg a másik lekérdez gomb hatására ellenőrzi, hogy minden szükséges mezőt kitöltött-e a felhasználó.
* A lekérdezés szövegét egy folyamat állítja össze egy rejtett mezőbe.

DECLARE

tabla VARCHAR2(50);

oszlopok VARCHAR2(200);

sql\_str VARCHAR2(300);

rendez VARCHAR2(50);

BEGIN

:P2\_HID\_SQL\_CONTAINER := '';

tabla := :P2\_TV;

oszlopok:= apex\_util.table\_to\_string(apex\_util.string\_to\_table(:P2\_CHK\_OSZLOPOK,':'),',');

if oszlopok is null then

oszlopok := '\*';

end if;

if :P2\_RDO\_SORT != 'NO' then

rendez:='ORDER BY' || ' ' || :P2\_TB\_SORT\_FIELD || ' ' || :P2\_RDO\_SORT;

end if;

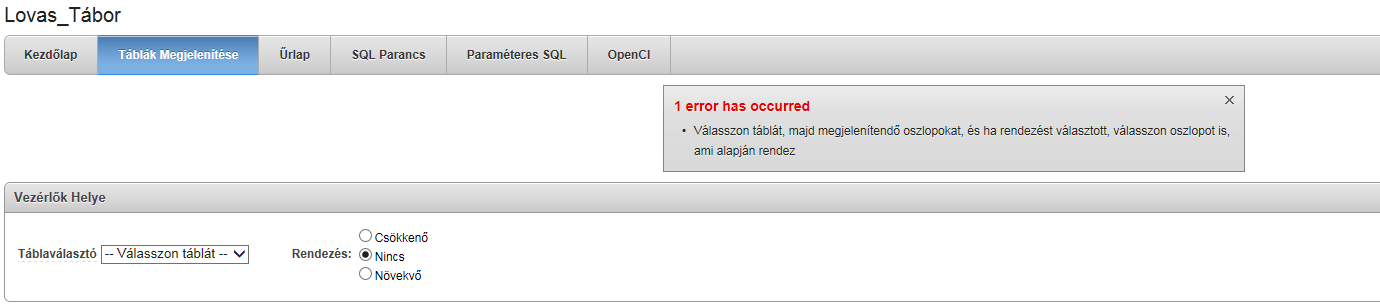
sql\_str := 'SELECT '|| oszlopok ||' FROM '|| tabla || ' ' || rendez;

:P2\_HID\_SQL\_CONTAINER := sql\_str;

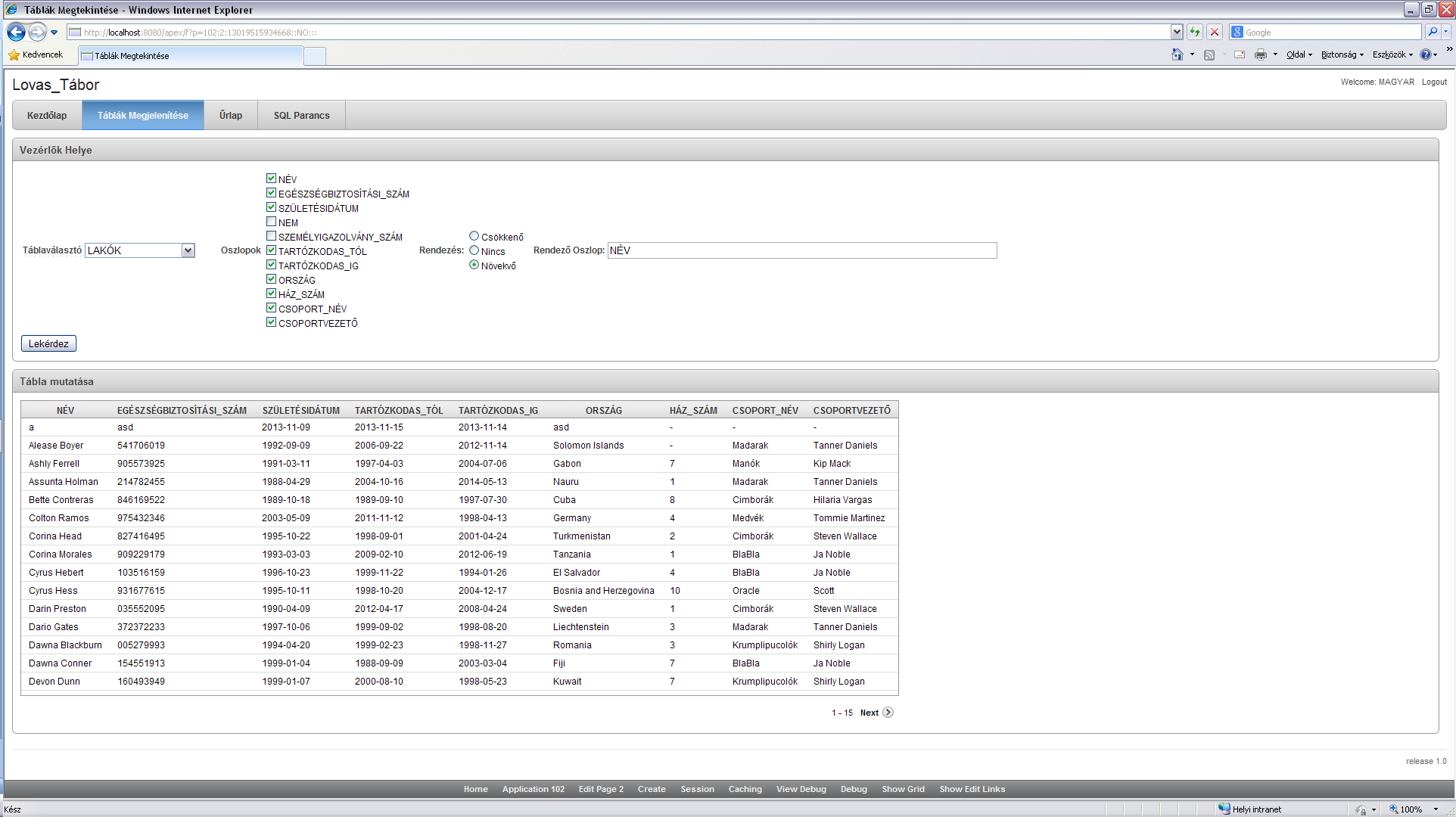
END;

### Futási eredménye

Az alábbi kép bemutatja azt az esetet, amikor a felhasználó nem tölti ki a megfelelő mezőket, látható a magyar nyelvű figyelmeztető szöveg.



Az alábbi kép a helyes működést ábrázolja.



## Űrlap

### A komponens célja

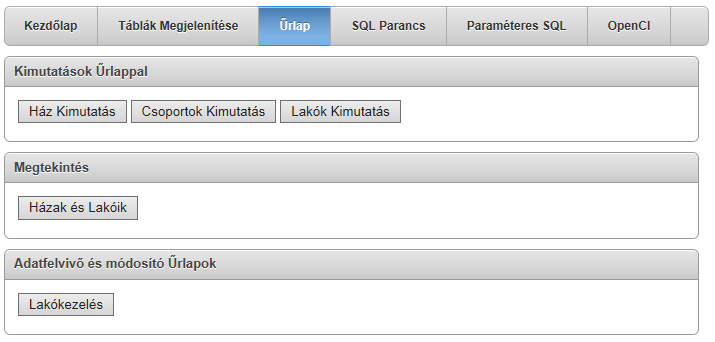
Innen érhetik el a dolgozók a rendszer által nyújtott űrlapokat (form) és kimutatásokat (report)

### A megvalósításának fontosabb részletei

A különböző alegységeket régiók különítik el, és a régiókban gombok vannak, melyek segítségével el lehet jutni a konkrét űrlaphoz vagy kimutatáshoz.

Az alegységeket külön-külön részletezem.

### Futási eredménye



## Kimutatások Űrlappal

### A komponens célja

### A megvalósításának fontosabb részletei

### Futási eredménye

## Megtekintés

### A komponens célja

### A megvalósításának fontosabb részletei

### Futási eredménye

## Adatfelvivő és módosító Űrlapok

### A komponens célja

### A megvalósításának fontosabb részletei

### Futási eredménye

## SQL Parancs

### A komponens célja

### A megvalósításának fontosabb részletei

### Futási eredménye

## Paraméteres SQL

### A komponens célja

### A megvalósításának fontosabb részletei

### Futási eredménye

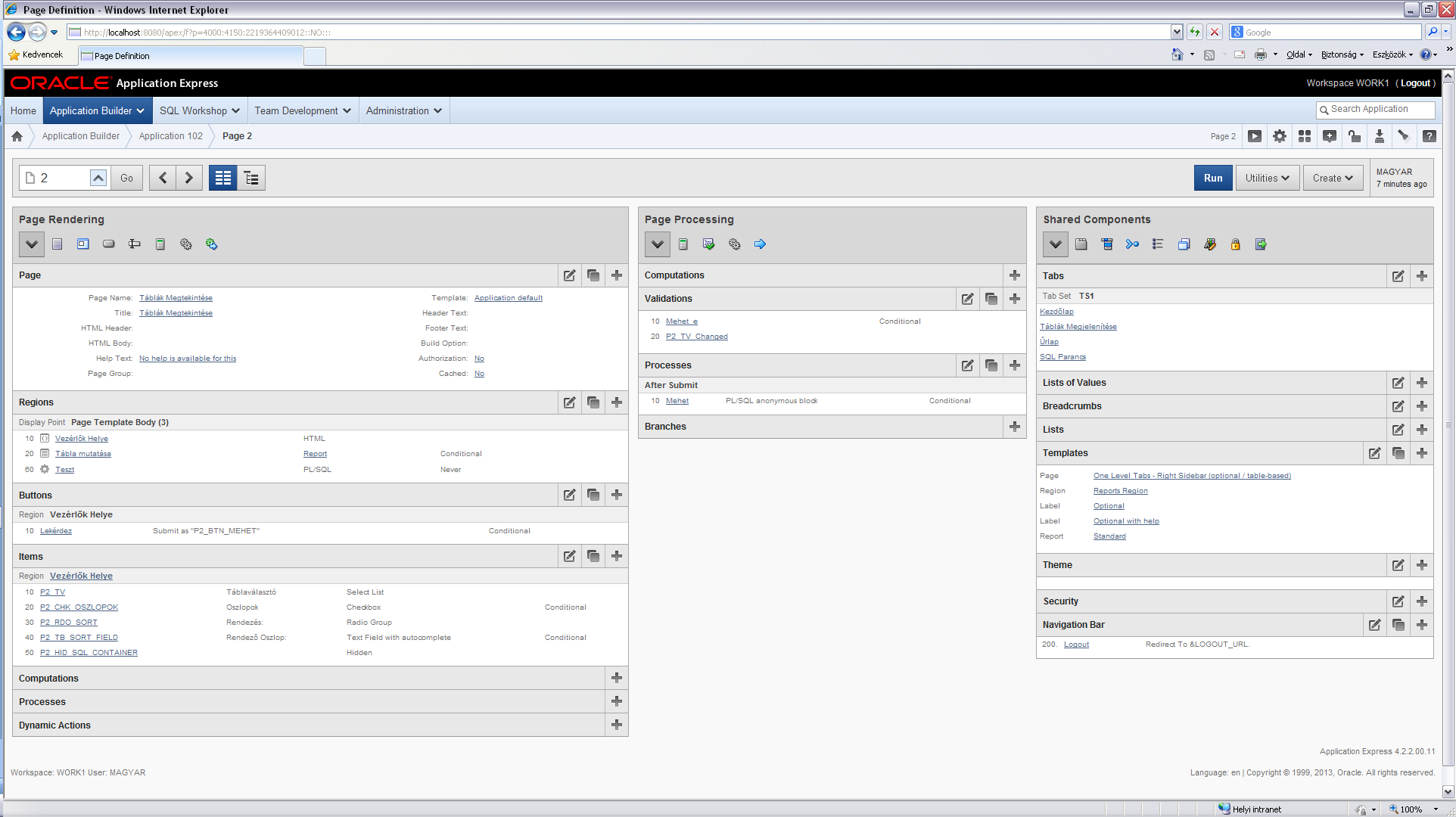
## OpenCL

### A komponens célja

### A megvalósításának fontosabb részletei

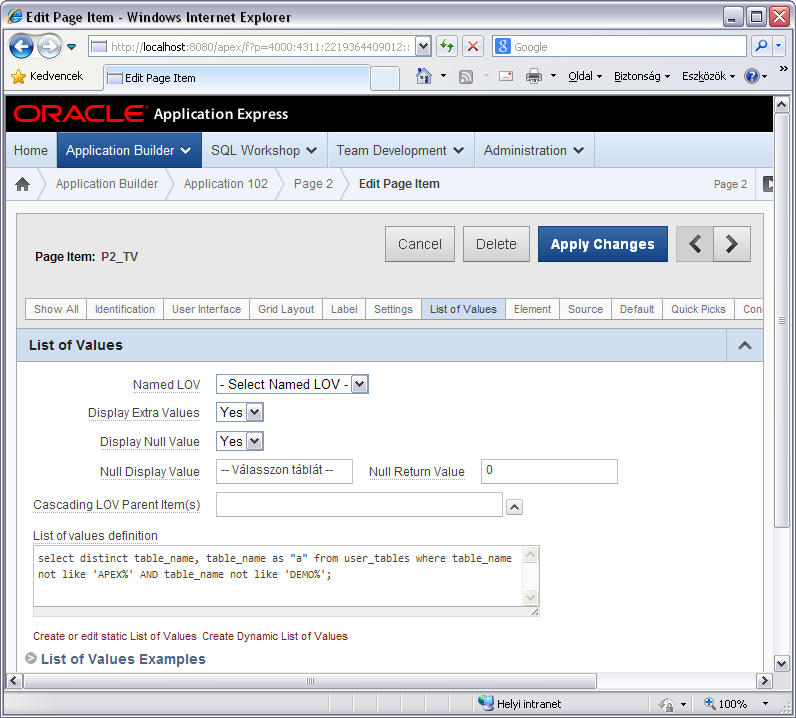
### Futási eredménye

### Megvalósítás:

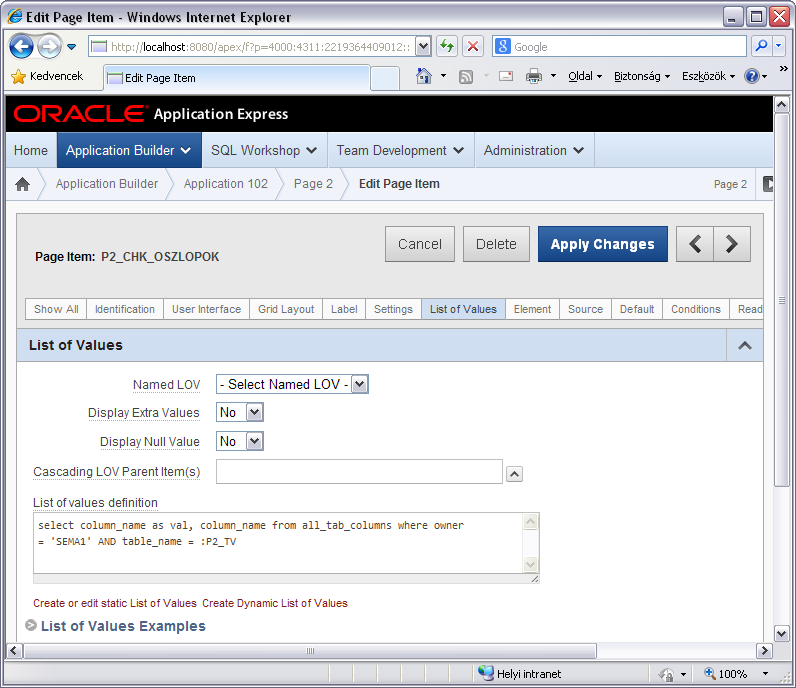


#### Itemek:

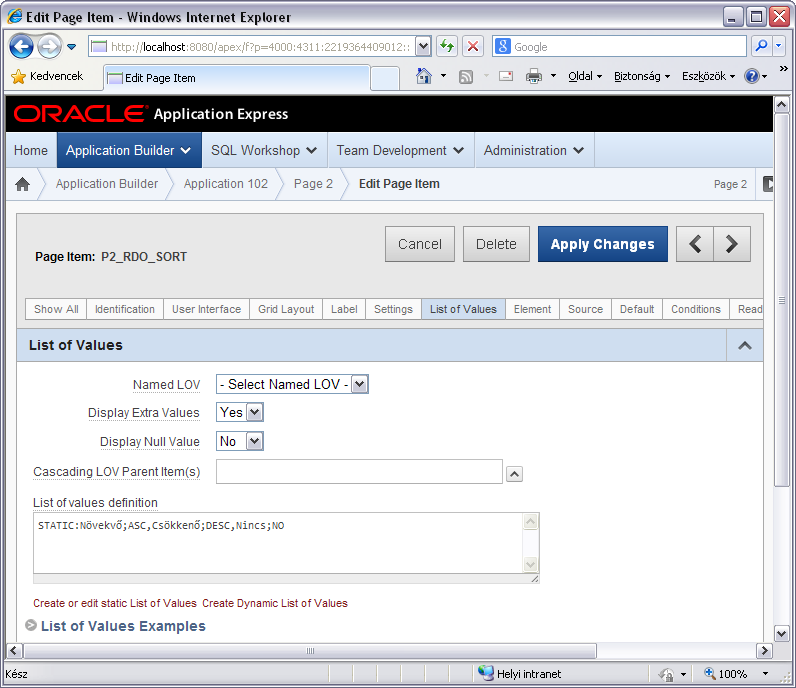
P2\_TV:



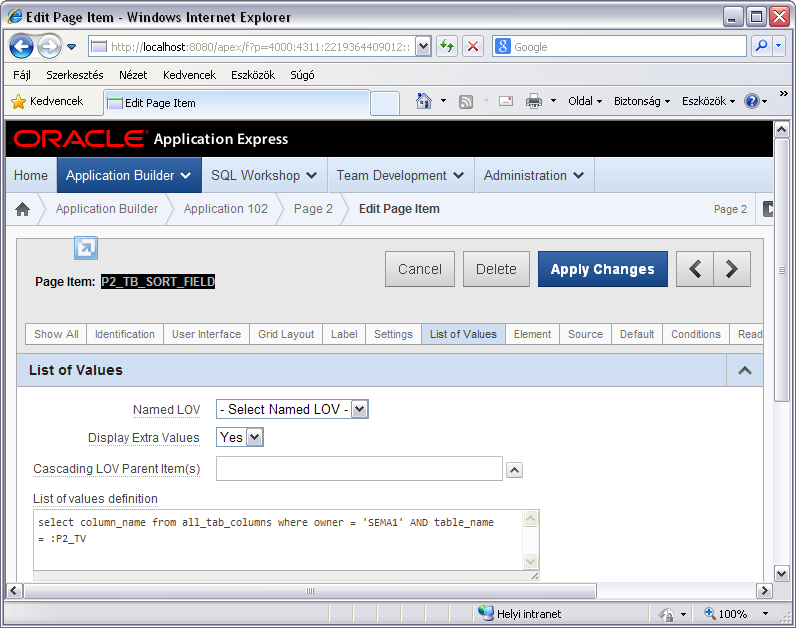
P2\_CHK\_OSZLOPOK:



P2\_RDO\_SORT:

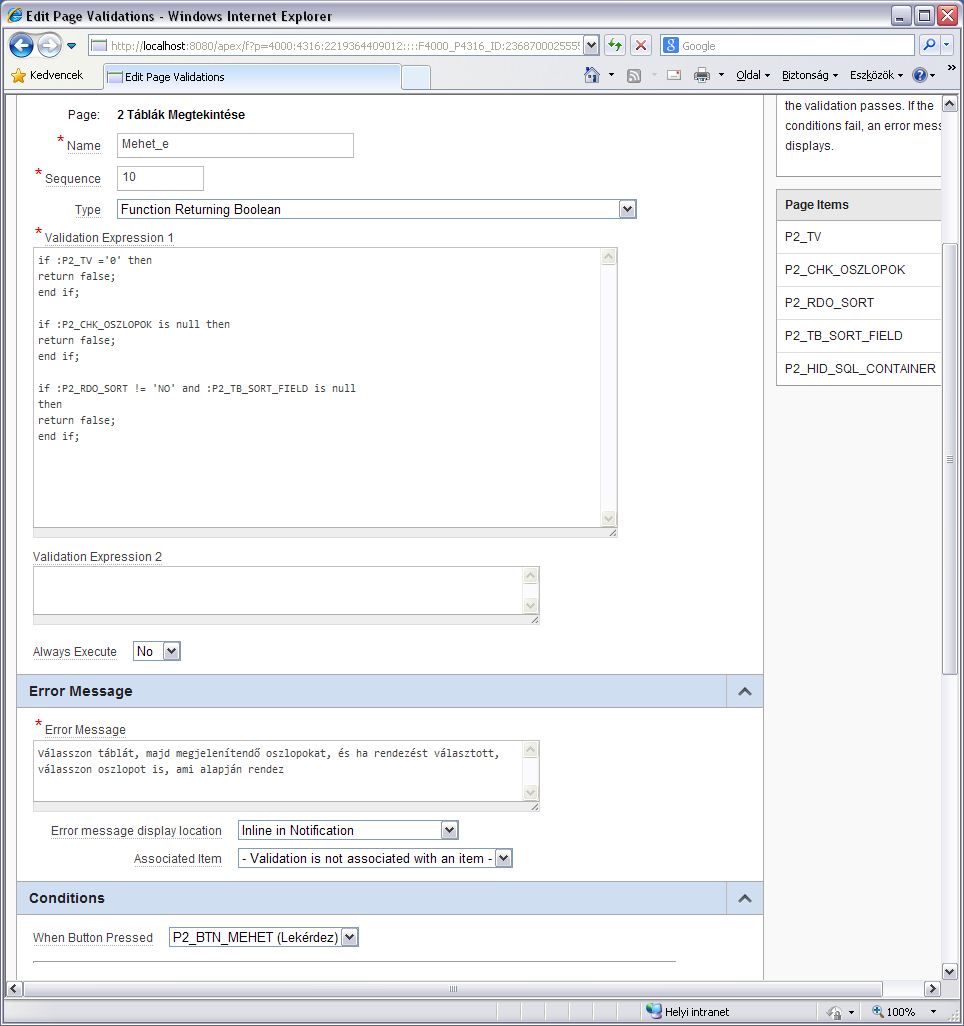


P2\_TB\_SORT\_FIELD:

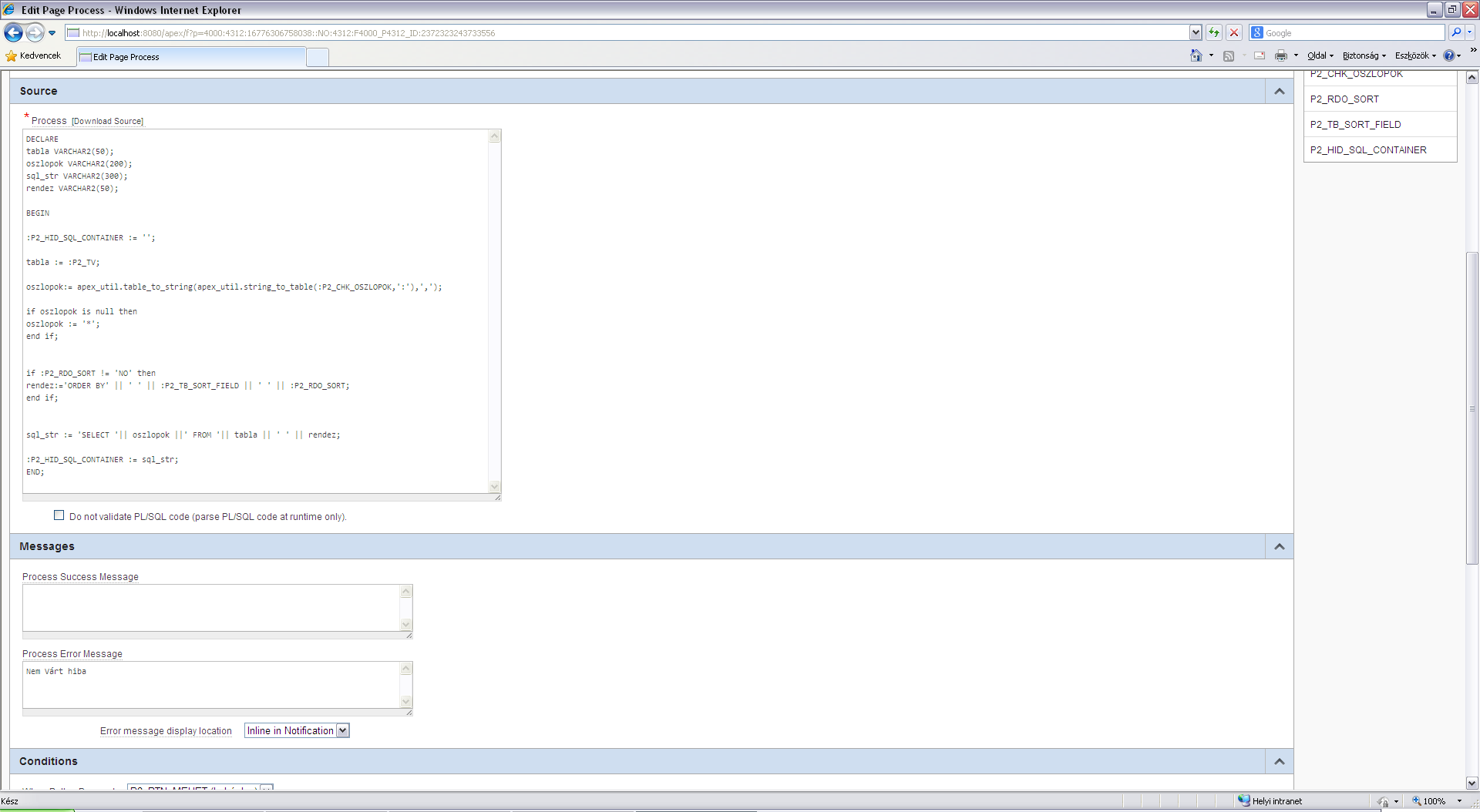


#### Page Process-ek:

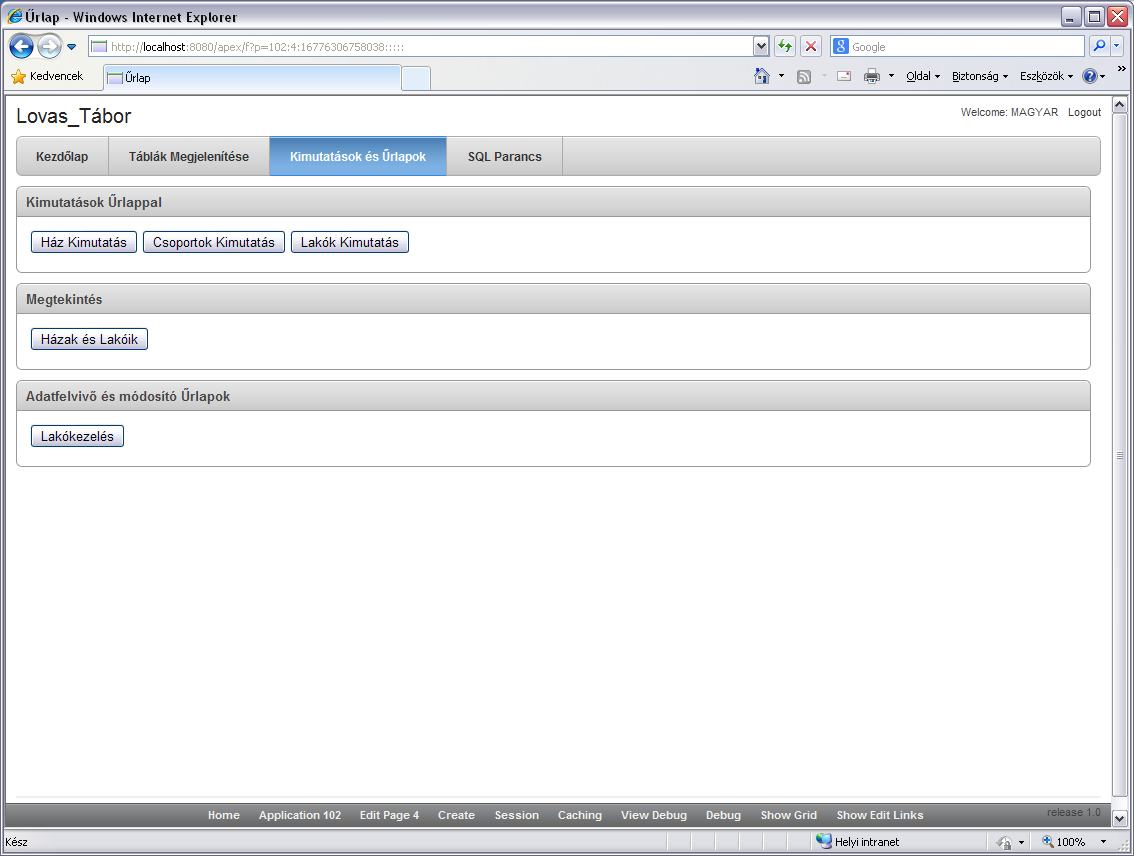
##### Validáció (Mehet\_e):

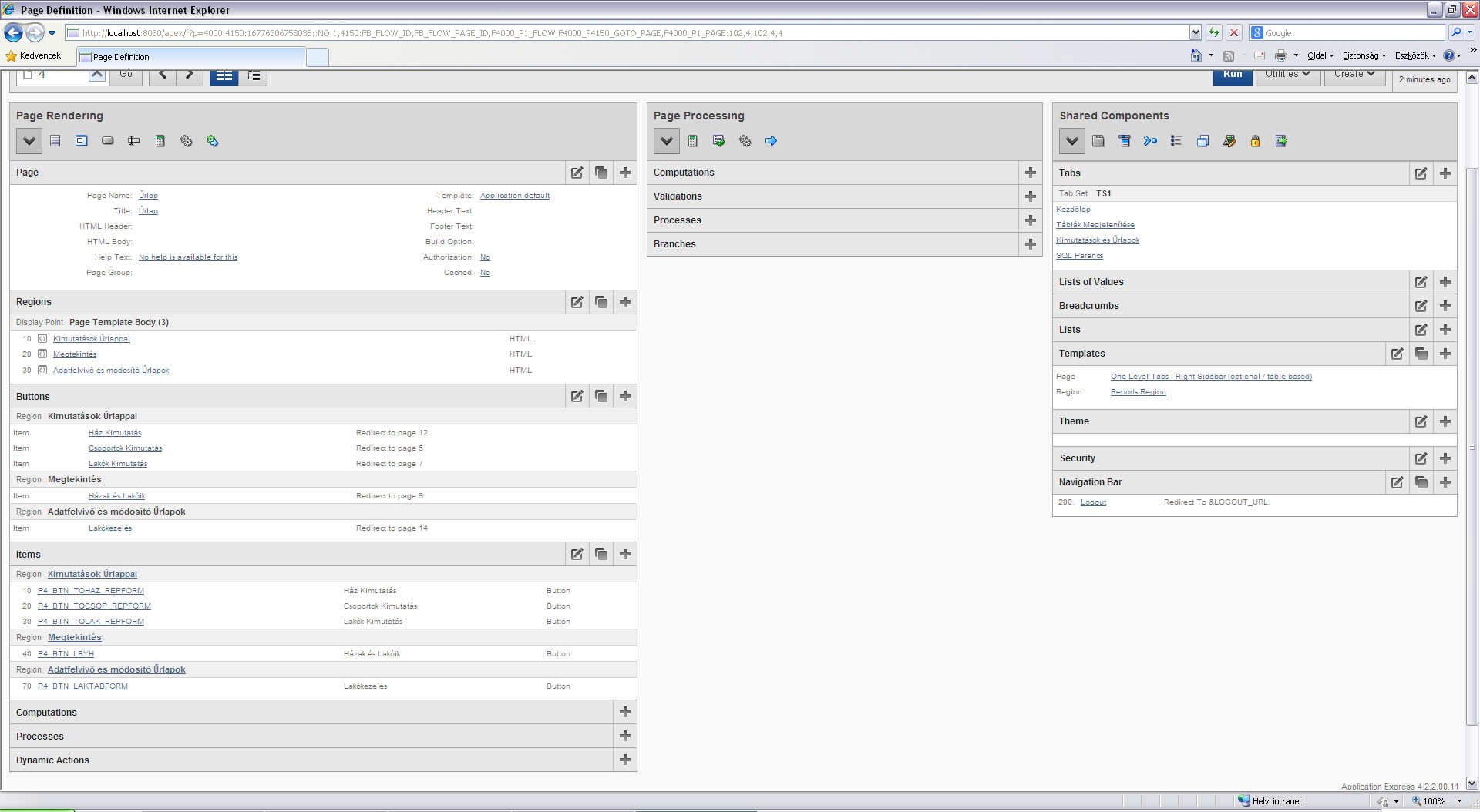


##### Process (Mehet)



## Kimutatások és Űrlapok





### Kimutatások Űrlappal

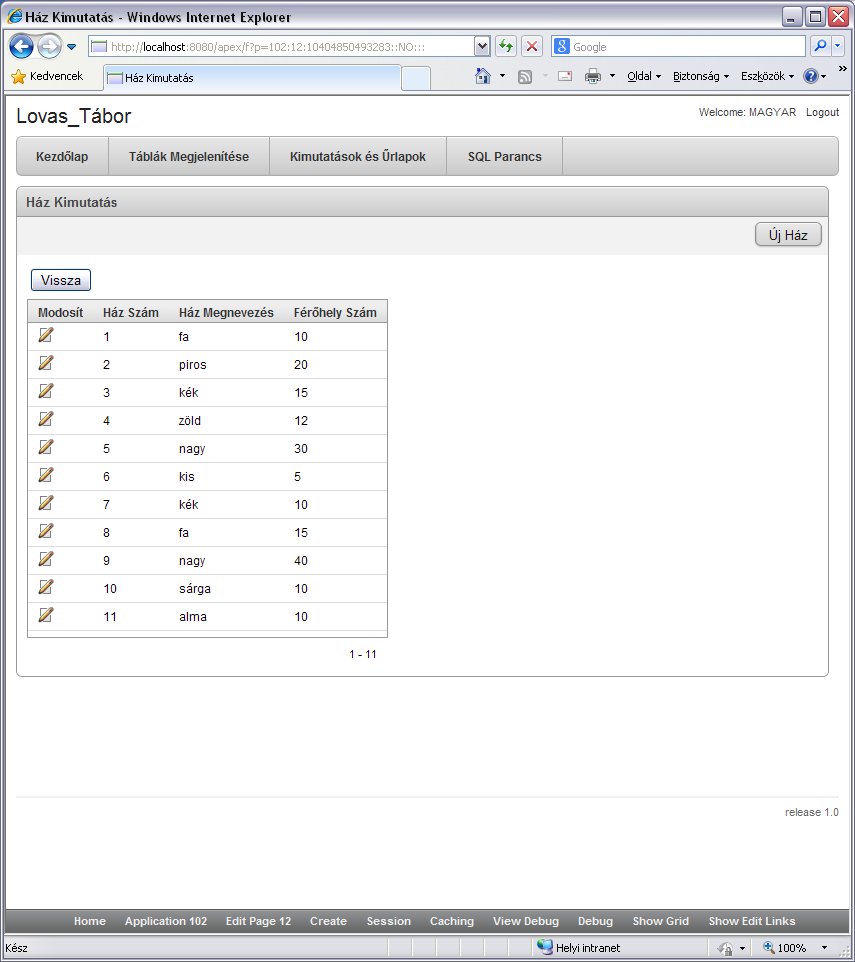
Itt található három gomb

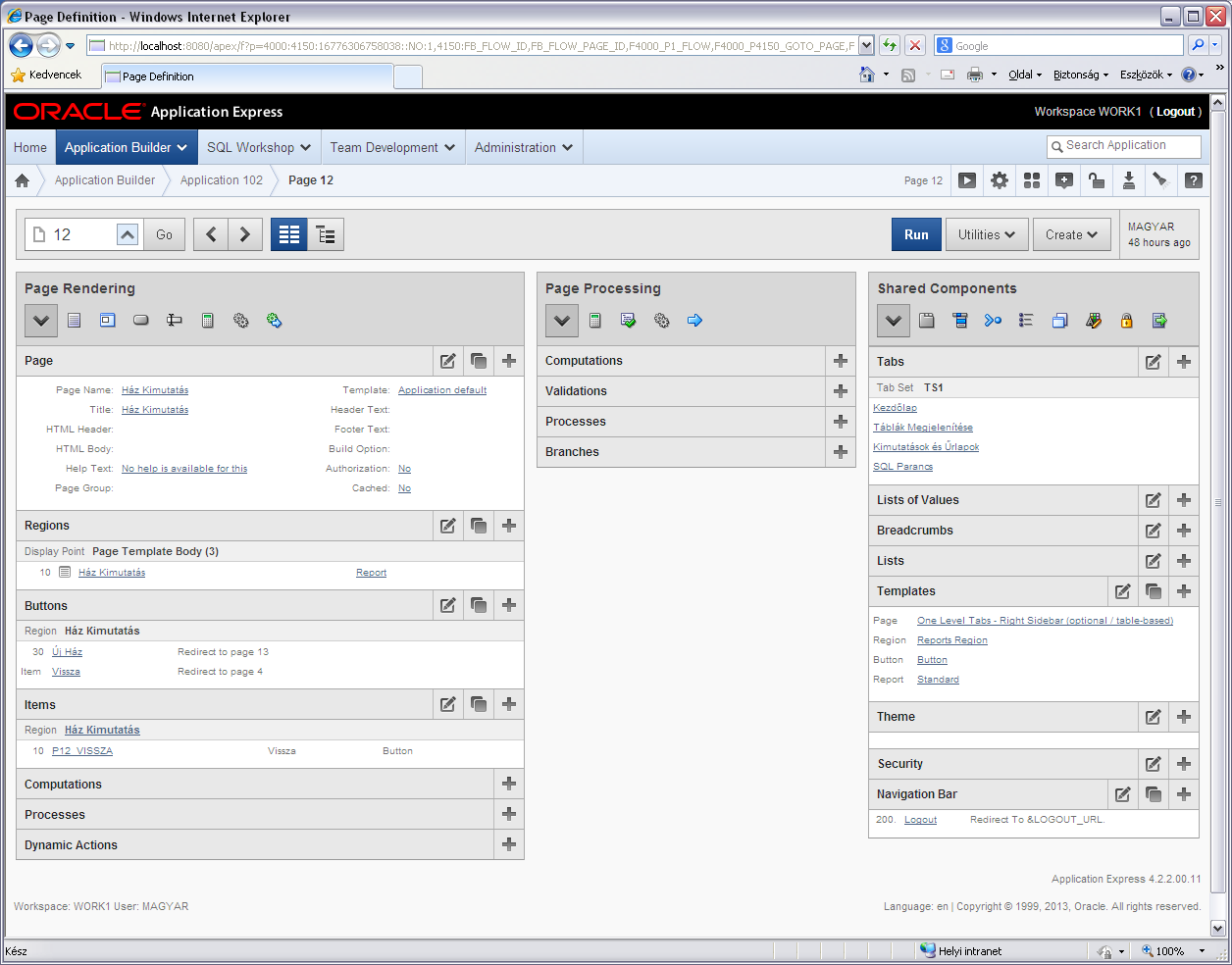
* Ház Kimutatás
* Csoportok Kimutatás
* Lakók Kimutatás

Mind egy ReportForm technikával megvalósított adatmegjelenító/felvivő/módosító részhez vezet.

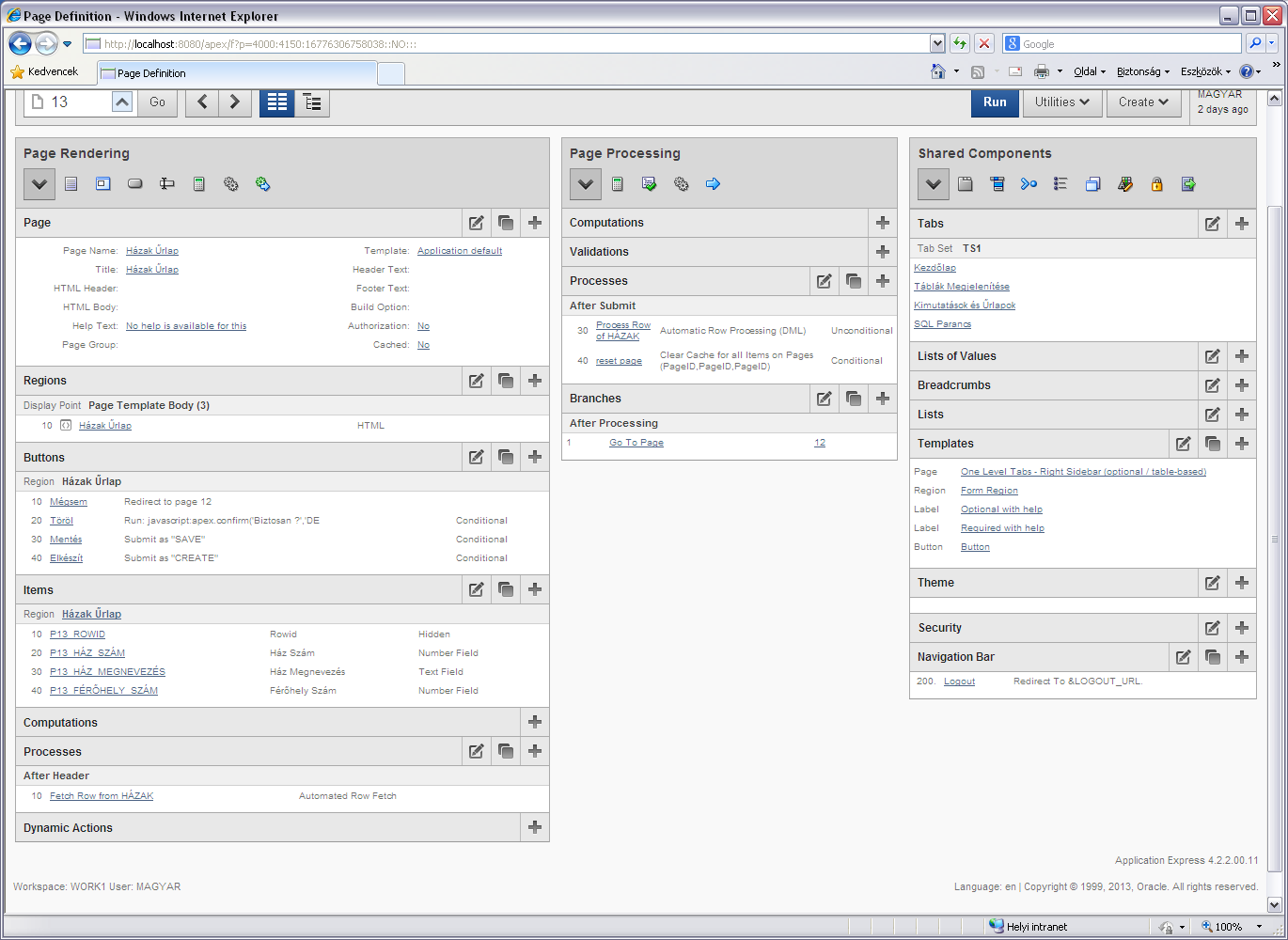
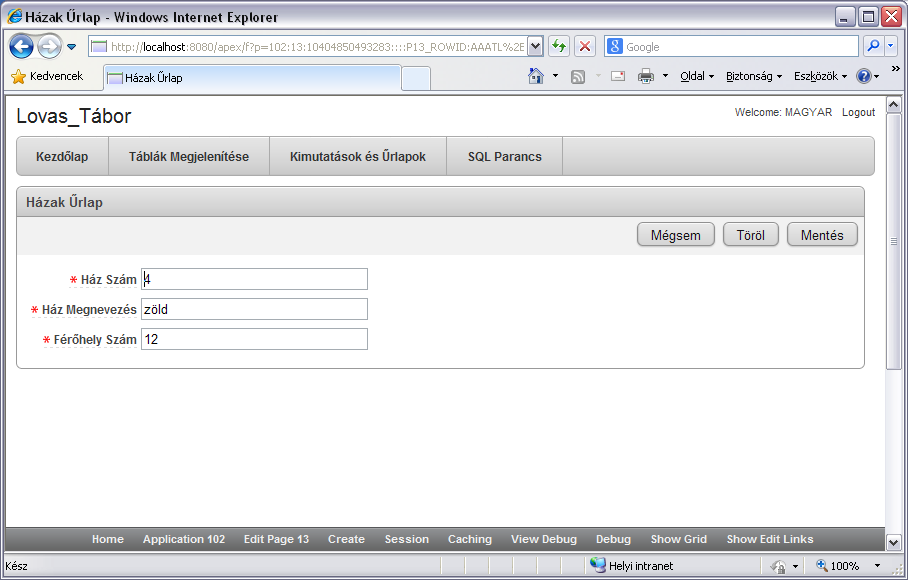
### Ház Kimutatás

#### Kimutatás (Riport)





#### Űrlap (Form)



### Csoportok Kimutatás

### Lakók Kimutatás