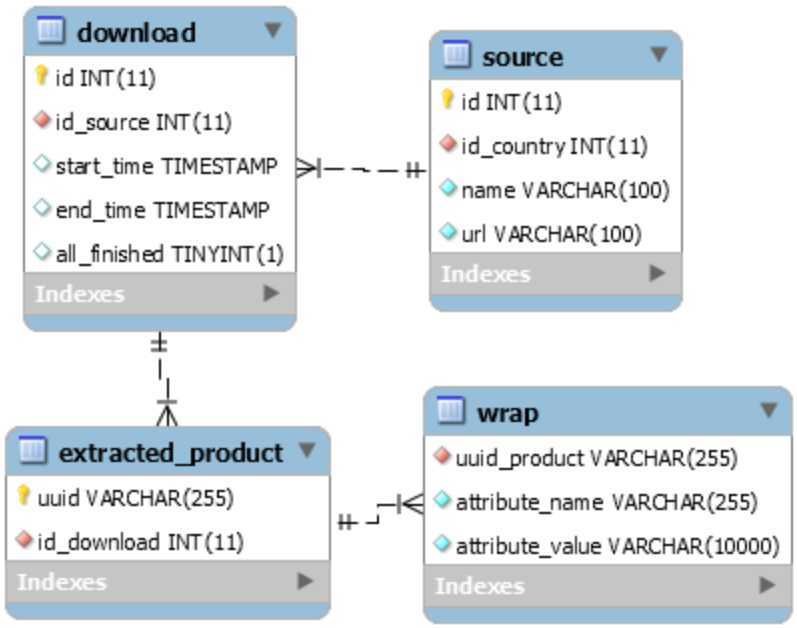
HLAVNÉ ZADANIE:

Vašou úlohou je optimalizácia databázy tak, aby nižšie spomenuté dopyty (ich zámer, teda dávajúce rovnaký výsledok, nie presné znenie) boli vykonané v čo najkratšom čase. Môžete uvažovať akúkoľvek modifikáciu databázy. Môžete napríklad zvážiť nasledovné kroky, ktoré niekedy môžu viesť k zlepšeniu rýchlosti dopytov:

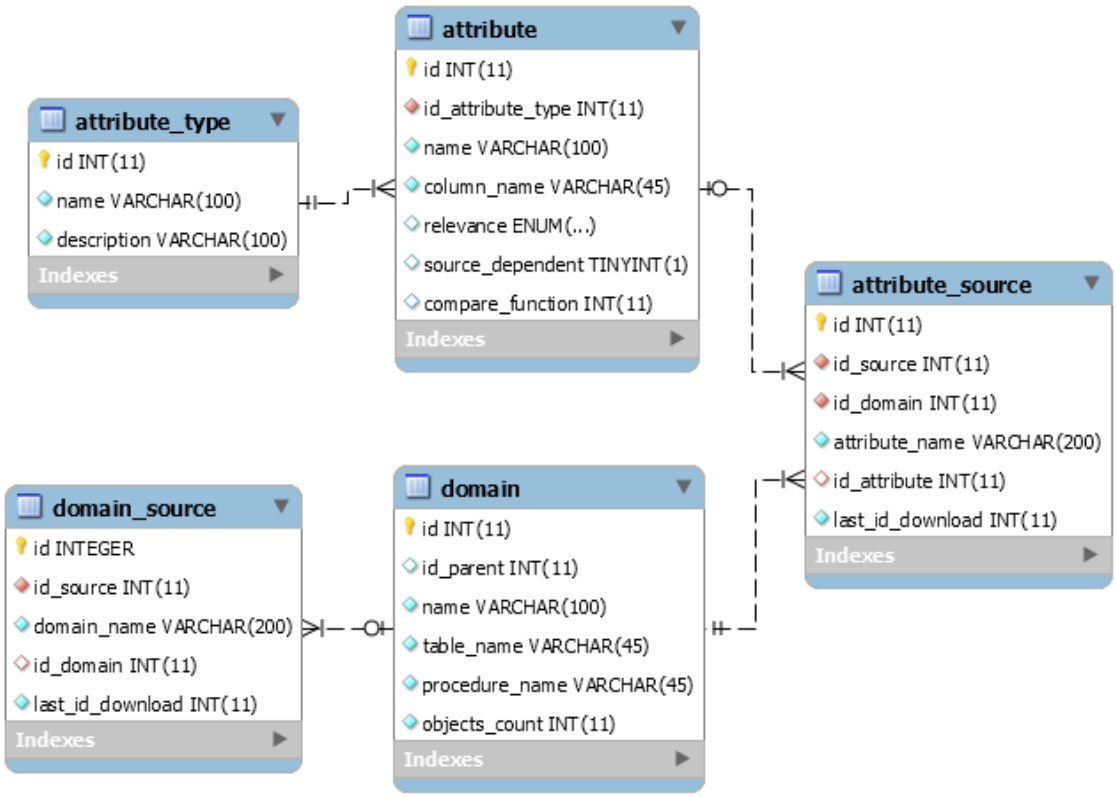
- Ak tabuľka nemá primárny kľúč a teda klastrovaný index, pridať ho
- Pridať neklastované indexy na nekľúčové stĺpce alebo množiny stĺpcov
- Zmeniť klastorvaný index t.j. zmeniť primárny kľúč, napr. zvážiť užitočnosť a potrebu klasického stĺpca id typu integer predstavujúceho umelý primárny kľúč
- Alebo naopak: nahradenie zložitého primárneho kľúča za umelý primárny kľúč id typu integer
- Rozšírenie indexu (klastrovaného alebo neklastrovaného) o ďalšie stĺpce
- Zvážiť pridanie stĺpcov do tabuľky, a tým pádom zmenšiť počet potrebných joinov aj napriek redundancii (presun alebo kopírovanie hodnôt stĺpcov medzi tabuľkami)
- Nahradenie stĺpcov s variabilnou veľkosťou hodnôt (varchar) za stĺpce s fixnou dĺžkou (char)

V projekte Kapsa sa stále dookola realizuje nasledovný proces:

1. Pre každý z množiny internetových obchodov sa každý týždeň sťahujú webové prezentácie všetkých ponúkaných produktov a z detailovej stránky každého produktu sa vyrezávajú všetky nájdené dvojice reťazcov „názov atribútu“ a „hodnota atribútu“. Jedno takéto sťahovanie všetkých produktov jedného internetového obchodu má spoločný identifikátor **id_download**. Pri takomto sťahovaní sa stiahne veľa stránok produktov. Každá stránka a teda aj produkt získa jedinečný identifikátor **uuid**. Keďže každý z produktov má viacero atribútov (názov, cena, veľkosť,...) ale s dopredu neznámym počtom, sú tieto atribúty ukladané ako množina dvojíc **attribute_name**, **attribute_value** v tabuľke **wrap**. Na nasledujúcom obrázku vidíme schému aktuálnej databázy. Tabuľka **source** uchováva v každom riadku informácie o jednom z obchodov. Tabuľka **download** predstavuje v jednom riadku jedno každotýždenné sťahovanie obchodov. Tabuľka **extracted_product** predstavuje v každom riadku jednu webstránku a zároveň jeden z mnohých produktov stiahnutých z internetového obchodu.



2. Po skončení sťahovania a naplnení týchto tabuliek nastáva proces konverzie množiny dvojíc „názov atribútu“ a „hodnota atribútu“ každého produktu, ak administrátor napísal konverzné pravidlá. Tieto konverzné pravidlá slúžia na určenie domény produktu (napr. chladnička, mobilný telefón, kosačka,...) a pre ostatné atribúty na určenie jeho typu a ďalších vlastností.
   Názov domény má každý produkt uvedený v tabuľke **wrap** ako samostatný atribút, konkrétne je to hodnota v stĺpci **attribute_value**, kde hodnota v stĺpci **attribute_name**= ‘domain_name‘. Konverzné pravidlo pre doménu priradí produktu riadok tabuľky **domain** prostredníctvom tabuľky **domain_source**. Riadok tabuľky **domain_source** hovorí „pre daný zdroj t.j. e-shop (stĺpec **id_source**) a daný názov domény produktov extrahovaných z tohto zdroja (stĺpec **domain_name**) určujeme doménu s identifikátorom **id_domain**“. Podobne pre každý extrahovaný atribút produktu určíme jeho typ a ďalšie vlastnosti – priradíme mu riadok tabuľky **attribute**. Toto priradenie sa deje cez tabuľku **attribute_source**. Pravidlo v tabuľke **attribute_source** má podobnú štruktúru ako tabuľka **domain_source** no obsahuje aj identifikátor domény produktu, teda pravidlá pre atribúty sú uvedené pre každú doménu produktu zvlášť.



Keďže množina domén aj atribútov týchto domén sa môže v čase meniť, je potrebné v tabuľkách **domain_source** aj **attribute_source** udržiavať vždy aktuálnu množinu domén atribútov v e-shope po každom download-e. V reči tabuliek to znamená, že v tabuľke **domain_source** je potrebné zapísať všetky domény vyskytujúce sa v tabuľke **wrap**. Ak pre doménu neexistovalo pravidlo mapujúce jej meno na identifikátor domény , tak sa zapíše s hodnotou **id_domain** = null. Prvým krokom k tejto aktualizácii je nasledovný dopyt:

Pridajte do **domain_source** tabuľky všetky domény, ktoré sa vyskytujú vo **wrap** tabuľke pre daný download. Pre domény zdroja, z ktorého bol daný download robený, ktoré už boli v **domain_source** tabuľke, iba aktualizujte hodnotu **last_id_download**.

Pre účely ľahkého opakovaného testovania urobte dopyt, ktorý robí skoro rovnakú prácu (až na zmenu dát):

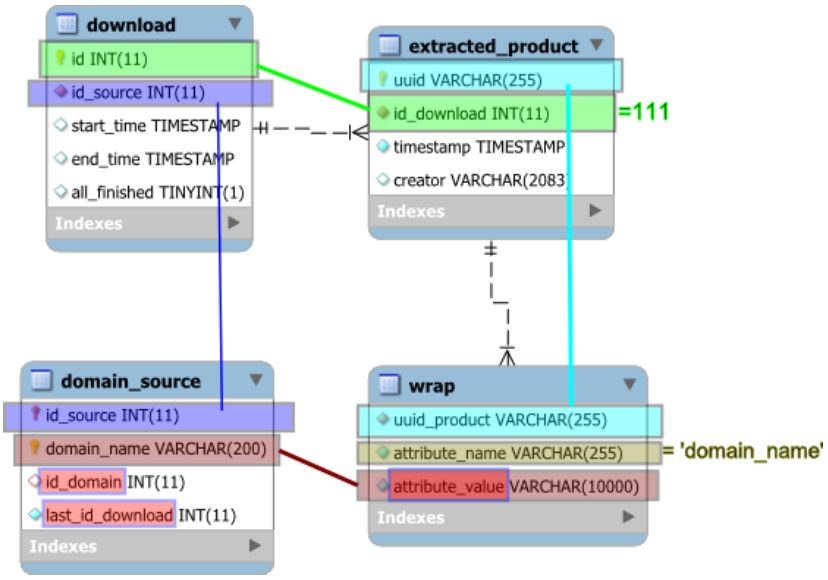
DOPYT 1:

Pre všetky domény, ktoré sa vyskytujú vo **wrap** tabuľke pre download s id = 111 vypíšte ich **domain_name**, **id_domain** a **last_id_download**, ak už boli v **domain_source** tabuľke zaznamenané v nejakom predchádzajúcom download-e, a ak neboli tak vypíšte pre **id_domain** hodnotu null a pre **last_id_download** napíšte hodnotu 111.

RIEŠENIE:

```
SELECT DISTINCT wrap.attribute_value, ds.id_domain, COALESCE(ds.last_id_download,111)
FROM download d
JOIN extracted_product ep ON d.id = ep.id_download
```

```
JOIN wrap ON ep.uuid = wrap.uuid_product
LEFT OUTER JOIN domain_source ds ON wrap.attribute_value = ds.domain_name AND
                    d.id_source = ds.id_source
WHERE ep.id_download=111 AND wrap.attribute_name = 'domain_name';
```



Aktualizácia tabuľky **attribute_source** sa realizuje nasledovným dopytom:

Pre každú doménu, ktorá má v tabuľke **domain_source** v stĺpci **id_domain** nenullovú hodnotu pridajte do **attribute_source** tabuľky všetky atribúty, ktoré sa vyskytujú vo **wrap** tabuľke pre každý objekt tejto domény daného downloadu. Pre atribúty, ktoré už boli v **attribute_source** tabuľke pre danú doménu a zdroj, iba aktualizujte hodnotu **last_id_download**

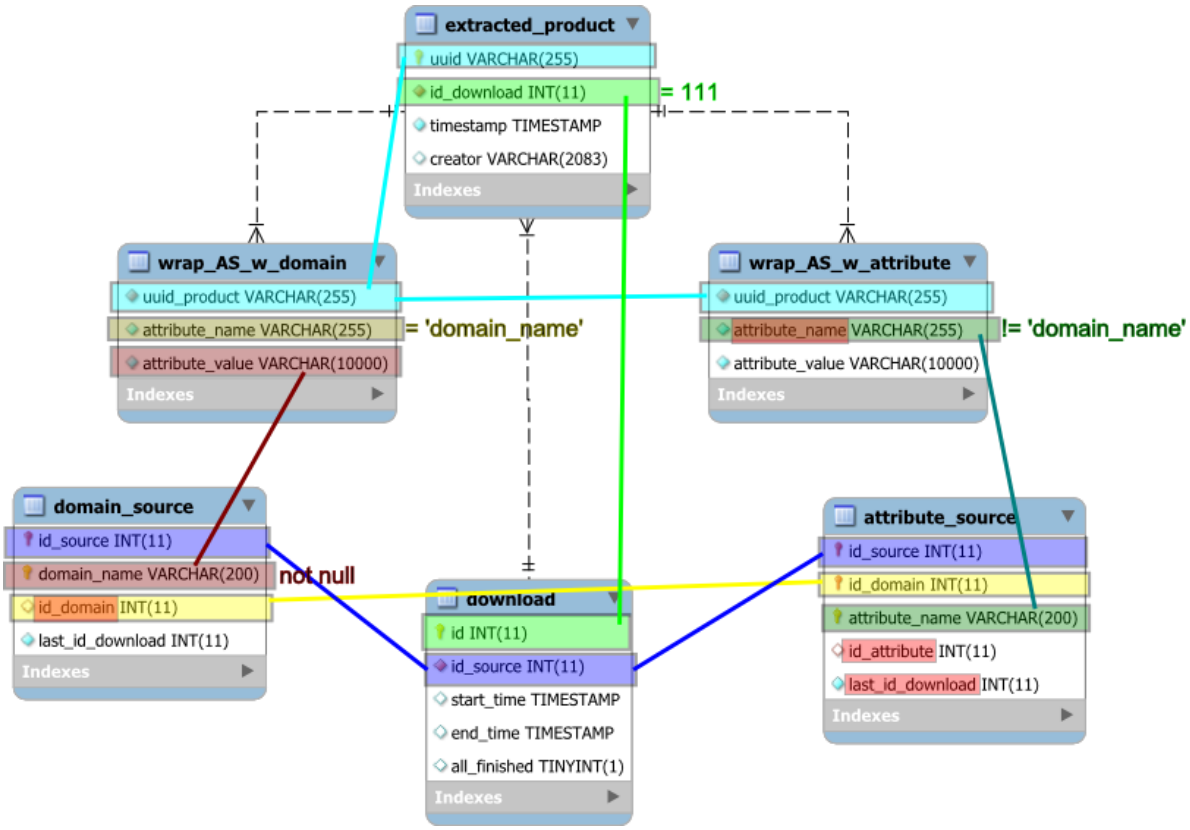
Opäť pre účely ľahkého opakovaného testovania urobte dopyt, ktorý robí skoro rovnakú prácu (až na zmenu dát):

DOPYT 2:

Pre daný download s id=111 a každú doménu, ktorá má v tabuľke **domain_source** v stĺpci **id_domain** nenullovú hodnotu z tabuľky **wrap** názvy všetkých atribútov ako **attribute_name**, ďalej **id_domain**, **id_attribute** a **last_id_download**, ak už boli v **attribute_source** tabuľke zaznamenané v nejakom predchádzajúcom download-e, a ak neboli tak vypíšte pre **id_attribute** hodnotu null a pre **last_id_download** napíšte hodnotu 110.

RIEŠENIE:

```
SELECT DISTINCT w_attribute.attribute_name, ds.id_domain, atts.id_attribute,
COALESCE(atts.last_id_download,111)
FROM download d
JOIN extracted_product ep ON d.id = ep.id_download
JOIN wrap AS w_domain ON ep.uuid = w_domain.uuid_product
JOIN domain_source ds ON w_domain.attribute_value = ds.domain_name AND d.id_source =
        ds.id_source AND ds.id_domain IS NOT NULL
JOIN wrap AS w_attribute ON w_domain.uuid_product = w_attribute.uuid_product AND
        w_attribute.attribute_name != 'domain_name'
LEFT OUTER JOIN attribute_source atts ON atts.id_source = d.id_source AND atts.id_domain =
        ds.id_domain AND atts.attribute_name = w_attribute.attribute_name
WHERE ep.id_download=111 AND w_domain.attribute_name = 'domain_name';
```



Keď administrátor vyrába spomínané konverzné pravidlá, ale aj pri ďalších automatických konverziách je potrebné poznať aj aké hodnoty nejaký atribút nadobúda. Preto je potrebný ešte nasledovný dopyt:

DOPYT 3:

Pre daný download (111), názov domény ('Eloshop.skElektronikaTelevízoryLCD/LED') a názov atribútu ('Rozlíšenie:') vypíšte z tabuľky **wrap** všetky rôzne hodnoty a ich početnosť, ktoré produkty tejto domény v tomto atribúte nadobúdajú.

RIEŠENIE:

```
SELECT w_domain.attribute_value as domain_name, w_attribute.attribute_name,
w_attribute.attribute_value, count(w_attribute.attribute_value)
FROM download d
JOIN extracted_product ep ON d.id = ep.id_download
JOIN wrap AS w_domain ON ep.uuid = w_domain.uuid_product
JOIN wrap AS w_attribute ON w_domain.uuid_product = w_attribute.uuid_product
WHERE ep.id_download=111
        AND w_domain.attribute_name = 'domain_name'
        AND w_domain.attribute_value = 'Eloshop.skElektronikaTelevízoryLCD/LED'
        AND w_attribute.attribute_name='Rozlíšenie:'
GROUP BY w_attribute.attribute_value;
```