**Žilinská univerzita v Žiline**

**Fakulta riadenia a informatiky**



**Semestrálna práca**

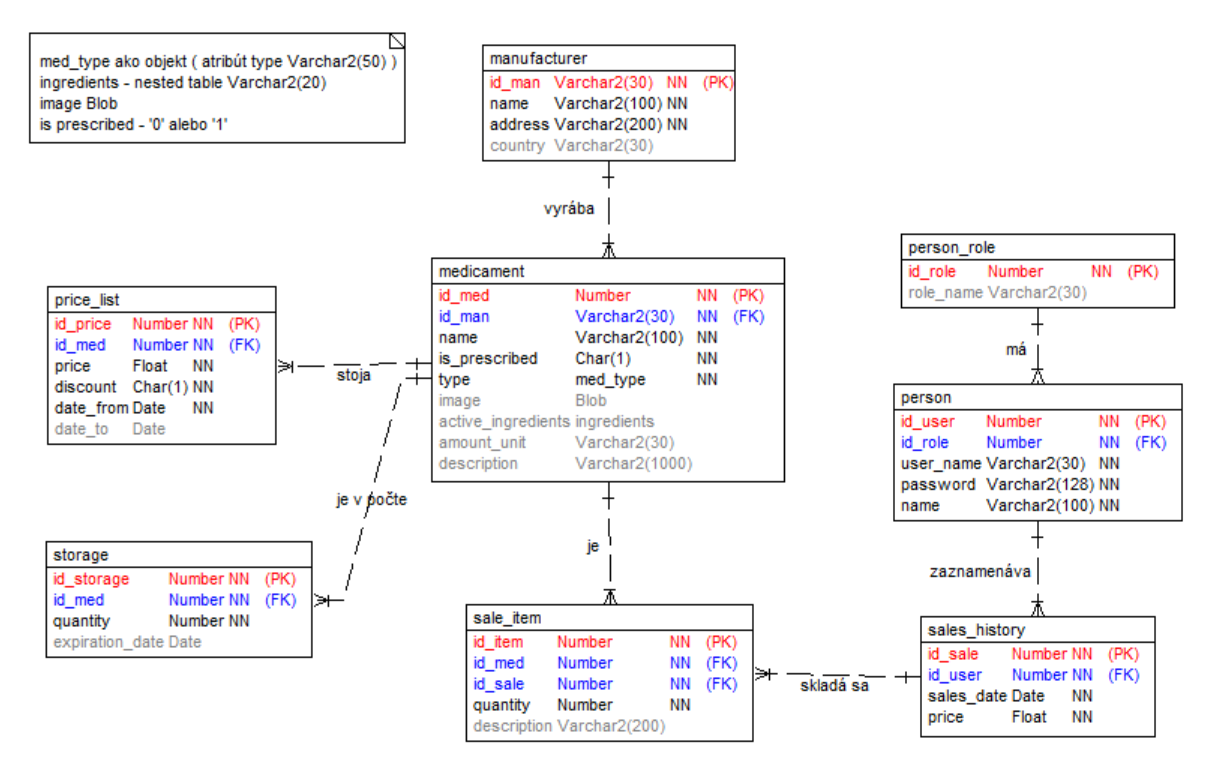
Pokročilé databázové systémy

Informačný systém pre správu zdravotníckych liekov a pomôcok

2021/2022

Matej Čajka, Dušan Novotňák, Pavol Grofčík, Veronika Nováková

# **Model systému**



Našim cieľom bolo vytvoriť aplikáciu na databázu liekov a zdravotníckych pomôcok. Na prácu sme používali C# a databázu Oracle.

Loby : image Blob atribút v tabuľke

Objekty: med\_type (nazov Varchar2(50)) – pre atribút *type* v tabuľke *medicament*

Kolekcia: nested table – ingredients ako table of Varchar2(20) – pre atribút *active\_ingredients* v tabuľke *medicament*

# **Analýza výkonnosti príkazov Select vzhľadom na indexy**

## **Počet liekov na predpis podľa manufacturera**

SELECT mf.name, is\_prescribed, COUNT(\*) TOTAL

FROM medicament m

INNER JOIN manufacturer mf ON mf.id\_man = m.id\_man

GROUP by mf.name, is\_prescribed

ORDER BY 1,2;

Bez indexu

Obrázok, na ktorom je stôl

Automaticky generovaný popis

S indexom

Obrázok, na ktorom je stôl

Automaticky generovaný popis

Vytvorenie indexu

CREATE INDEX my\_index1 ON medicament(id\_man, is\_prescribed);

**Analýza:**

Ako vidime, tak index na tabuľku medicament a jeho atribúty id\_man a is\_prescibed, optimalizoval náklady na vykonanie. Atribútu id\_man je cudzí kľúč, ktorý referencuje na tabuľku manufacturer, ktorá sa joinuje a spája s tabuľkou medicament. Z daných screenov, vidno, že po vytvorení indexu  my\_index1 sa znížili náklady na vykonanie z povôdných 144, bez použitia indexu na náklady 43, s použitím indexu. Rozdiel s použitím indexu bol finálne 144-43 = 101.

Index má efekt na vykonanie select.

## **TOP 10 bestsellerov**

WITH agg\_sale\_item AS (

SELECT si.id\_med, m.name, pl.price,  SUM(si.quantity) AS total\_sold, SUM(si.quantity)\*pl.price AS total\_price

FROM sale\_item si

INNER JOIN medicament m ON m.id\_med = si.id\_med

INNER JOIN price\_list pl ON pl.id\_med = m.id\_med

GROUP BY si.id\_med, m.name, pl.price

)

SELECT \*

FROM (

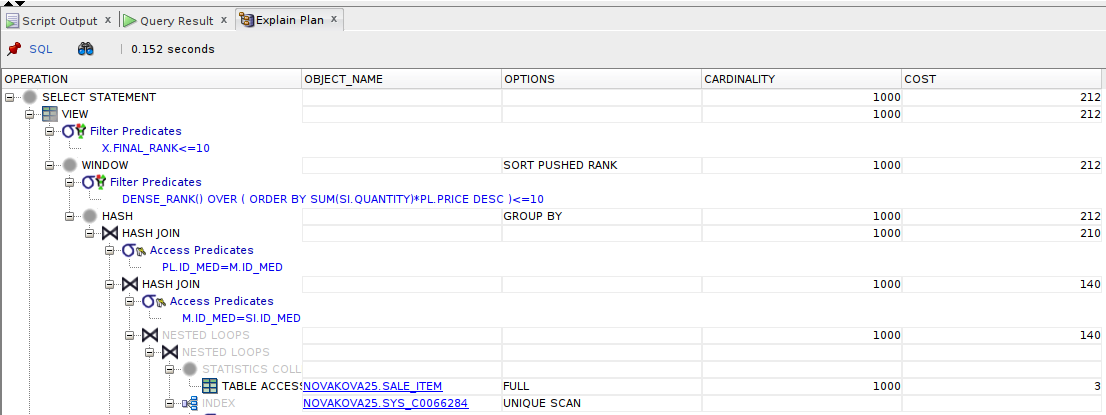
SELECT a.\*, DENSE\_RANK() OVER (ORDER BY total\_price DESC) AS FINAL\_RANK

FROM agg\_sale\_item a

) x

WHERE x.final\_rank <= 10;

Bez indexu



S indexom

Obrázok, na ktorom je stôl

Automaticky generovaný popis

Vytvorenie indexu

CREATE INDEX my\_index2 ON price\_list(id\_med, price);

CREATE INDEX my\_index3 ON sale\_item(id\_med, quantity);

**Analýza:**

Index na tabuľku price\_list  a index na tabuľku sale\_item optimalizovali náklady na vykonanie. Z daných screenov, vidno, že po vytvorení indexov  my\_index2 a my\_index3  sa znížili náklady na vykonanie. Rozdiel s použitím indexov bol finálne 212-178 = 34.

## **Výpis stavu skladu s danému dátumu**

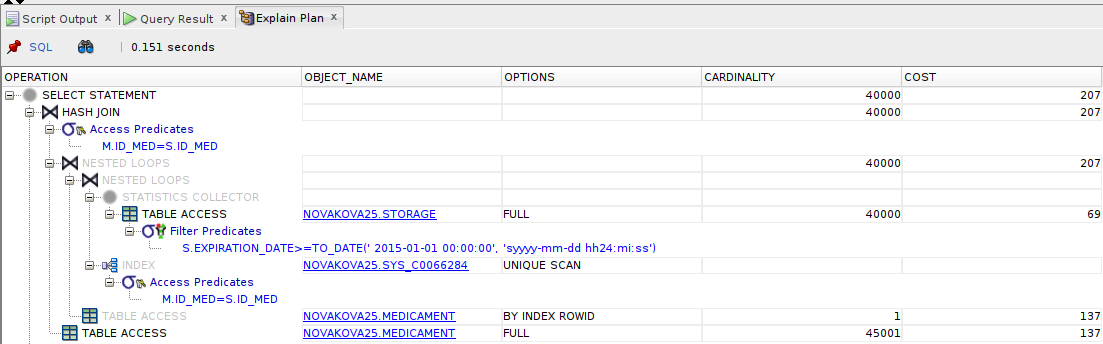
SELECT s.id\_storage, s.id\_med, m.name, s.quantity, s.expiration\_date FROM

storage s

INNER JOIN medicament m ON m.id\_med = s.id\_med

WHERE s.expiration\_date >= TO\_DATE('2015-01-01', 'yyyy-mm-dd');

Bez indexu



S indexom

Obrázok, na ktorom je text

Automaticky generovaný popis

CREATE INDEX my\_index8 ON novakova25.storage(expiration\_date, quantity, id\_med, id\_storage);

**Analýza:**

Index na tabuľku optimalizoval náklady na vykonanie. Z daných screenov, vidno, že po vytvorení indexu  my\_index8 sa znížili náklady na vykonanie. Rozdiel s použitím indexov bol finálne 207-91 = 116.

# **JSON výstupy**

Vytvorili sme 2 JSON výstupy :

* **Výpis zásob s aktuálnymi cenami**

select json\_object('medicament\_number' value id\_med,

'medicament' value med.name,

'type' value med.type.type,

'is\_prescribed' value CASE WHEN med.is\_prescribed = '0' THEN 'false' ELSE 'true' END FORMAT JSON,

'manufacturer' value man.name,

'amount\_and\_unit' value med.amount\_unit,

'description' value med.description,

'price\_euro' value p.price)

from price\_list p join medicament med using(id\_med) join manufacturer man using(id\_man) where date\_to is null;

* **Aktuálne zásoby na sklade**

select json\_object('medicament\_number' value id\_med,

'medicament' value name,

'sum\_all' value sum(storage.quantity),

'expiration\_dates' value json\_arrayagg(to\_char(storage.expiration\_date, 'DD.MM.YYYY')),

'quantities' value json\_arrayagg(storage.quantity) )

from medicament join storage using(id\_med) group by id\_med, name ;