

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Projekt 1. část - Datový model (ERD), model případů užití  
Fitness centrum

## Databázové systémy

## **Obsah**

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Datový model (ERD)</b>	<b>2</b>
<b>3</b>	<b>Model případů užití</b>	<b>4</b>
<b>4</b>	<b>Triggery</b>	<b>7</b>
<b>5</b>	<b>Procedury</b>	<b>7</b>
<b>6</b>	<b>Explain plan</b>	<b>7</b>
<b>7</b>	<b>Materializovaný pohled</b>	<b>8</b>

## 1 Zadání

Jednoduchý IS fitness centra, které organizuje různé kurzy skupinových lekcí (zumba, TRX, kruhový trénink, atd.). Ve fitness centru pracují instruktoři, kteří vedou jednotlivé skupinové lekce, a lidé na recepci, kteří se musí kromě vítání příchozích klientů a mixování proteinových koktejlů zapojit do práce s IS fitness centra prostřednictvím vytváření členských karet pro jednotlivé klienty, kteří se rozhodli pravidelně trápit svá těla ve fitness centru a chtějí využít členské výhody. Aby karta nebyla využívána jinými klienty než jejím vlastníkem, musí být v IS uloženy základní informace o klientech, jejich rodná čísla a adresy. Zákazník si může vypsát kurzy, které navštěvuje a informace o jednotlivých lekcích. Navíc si může zobrazit rozvrh vypisovaných kurzů a zjistit počet volných míst na jednotlivých lekcích a jejich cenu. Zákazník se může registrovat buď na jednu lekci nebo na celý kurz. Kurzy mají svou délku trvání, obtížnost a popis. Skupinové lekce probíhají v různých sálech fitness centra, které mají konkrétní název, umístění a maximální kapacitu. Lekce jsou vedené jedním instruktorem, mají maximální kapacitu účastníků a odehrávají se v daném sále v určitý čas a den v týdnu. Předpokládejte, že jeden instruktor může být vyškolen pro vedení různých kurzů, toto modelujte. Kromě pravidelných skupinových lekcí nabízí fitness centrum i individuální lekce, na kterých se instruktor věnuje pouze jednomu klientovi. Tyto lekce jsou podobného charakteru jako ty skupinové, jen je konkrétnímu klientovi věnováno více pozornosti. Instruktor má možnost vložit do systému nové typy kurzů a konkrétní lekce (a to jak skupinové, tak i individuální) a měnit čas a sál, ve kterém se lekce konají. Systém musí být na požádání schopen vypsát rozvrh pro jednotlivé místnosti.

## 2 Datový model (ERD)

Datový model se skládá ze 7 entit, jimiž jsou - Klient, Členská karta, Instruktor, Kurz, Sál, Individuální lekce a skupinová lekce.

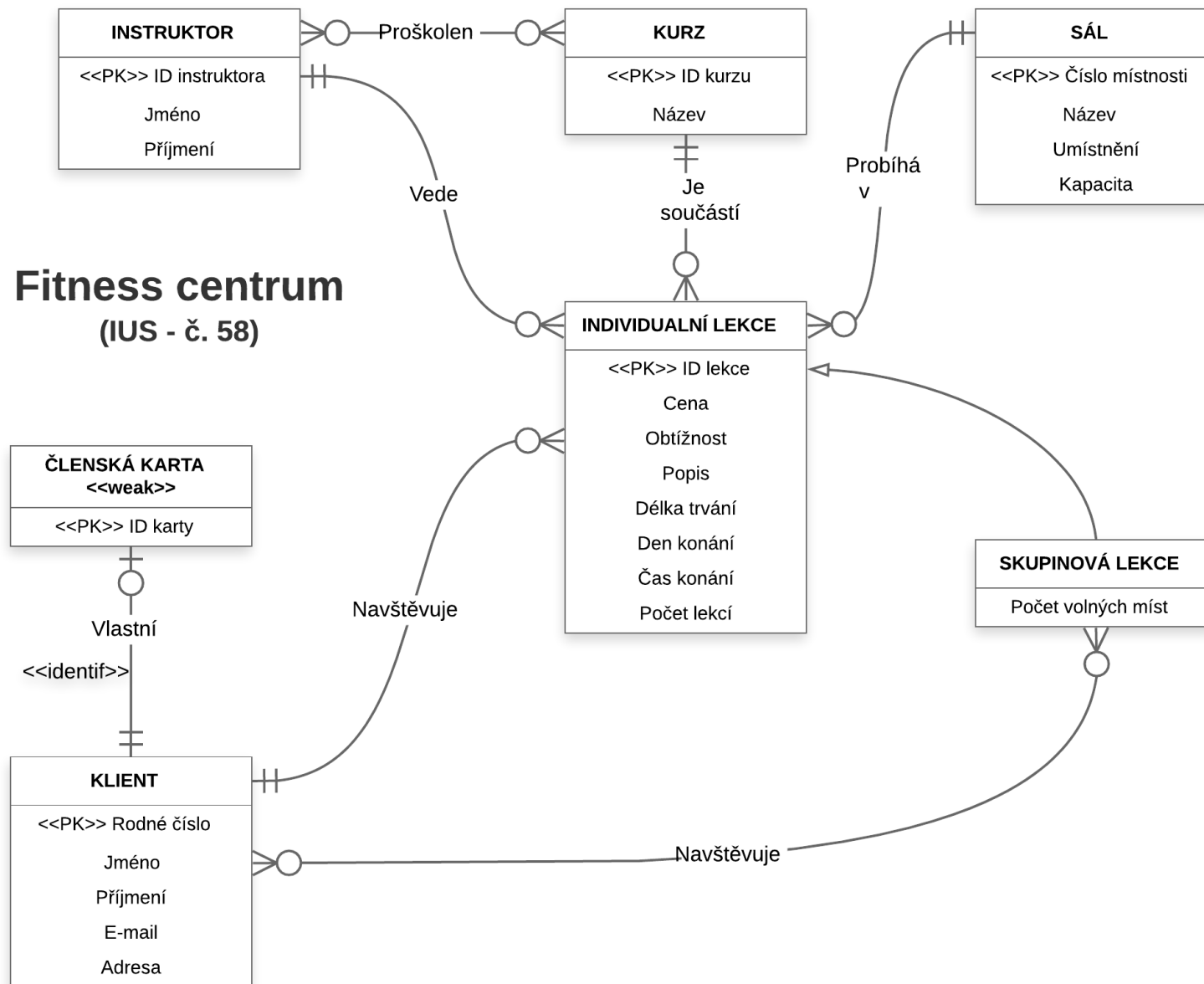
Klient obsahuje kontaktní informace jako adresa, e-mail a identifikační údaje jako je jméno a příjmení. Klient je identifikován pomocí rodného čísla. Klient může vlastnit až jednu členskou kartu, která je identifikována číslem karty.

Klient dále může navštěvovat libovolné individuální nebo skupinové lekce, kde skupinová lekce je generalizace individuální a má omezený počet míst. Každá lekce má své identifikační číslo, stupeň obtížnosti, cenu, popis, délku trvání, den konání, čas konání a počet lekcí.

Každá lekce je vedena jedním z instruktorů, který má své číslo a identifikační údaje. Instruktor musí být vyškolený pro kurz, pod který lekce spadá. Každý instruktor může být proškolen z několika kurzů a zároveň několik instruktorů může být proškolen z jednoho kurzu.

Kurz, pod který lekce spadá má vlastní název a identifikační číslo. Pod jeden kurz může patřit více lekcí.

Každá lekce se odehrává v jednom ze sálů. Každý sál je označen číslem a obsahuje informace o svém názvu, umístění a kapacitě.



Obrázek 1: Datový model (ERD)

### 3 Model případů užití

Diagram případu užití se skládá z 5 entit, jimiž jsou - Osoba, Recepce, Klient, Čas, Instruktor - a případů užití, kterých mohou nabývat. Entita **Osoba** má možnosti:

- zapsání lekce,
- zapsání do kurzu,
- vytvoření členského účtu a návazné vytvoření členské kartičky,
- zobrazení informací o kurzu, rozšířeno o možnost zobrazení rozvrhu všech kurzů,
- vypsání skupinových a individuálních lekcí,
- odhlášení z kurzu s návazným odhlášením z lekce a vypsáním skupinových a individuálních lekcí.

Entita **Recepce** tvoří rozšíření pro entitu **Osoba**. Společně s možnostmi, jež poskytuje **Osoba**, dojde k rozšíření o správu účtů a zaslání informace zákazníkům na mail.

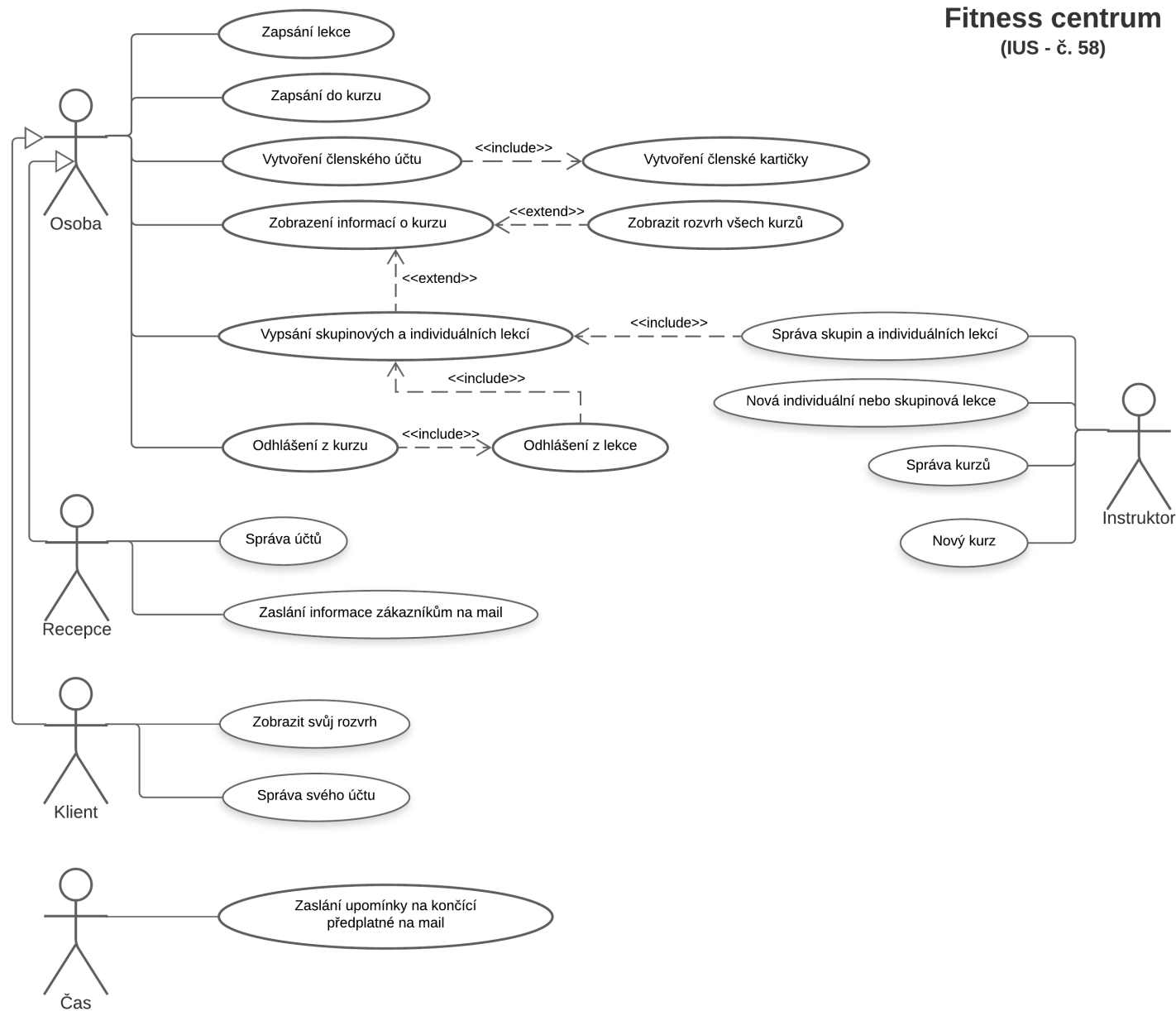
Dále entita **Klient**, rovněž rozšiřující **Osobu**. K možnostem **Osoby** přidává zobrazení svého rozvrhu a správu svého účtu.

Entita **Čas** má pouze jedinou vlastnost, jíž je zaslání upomínky na končící předplatné na mail.

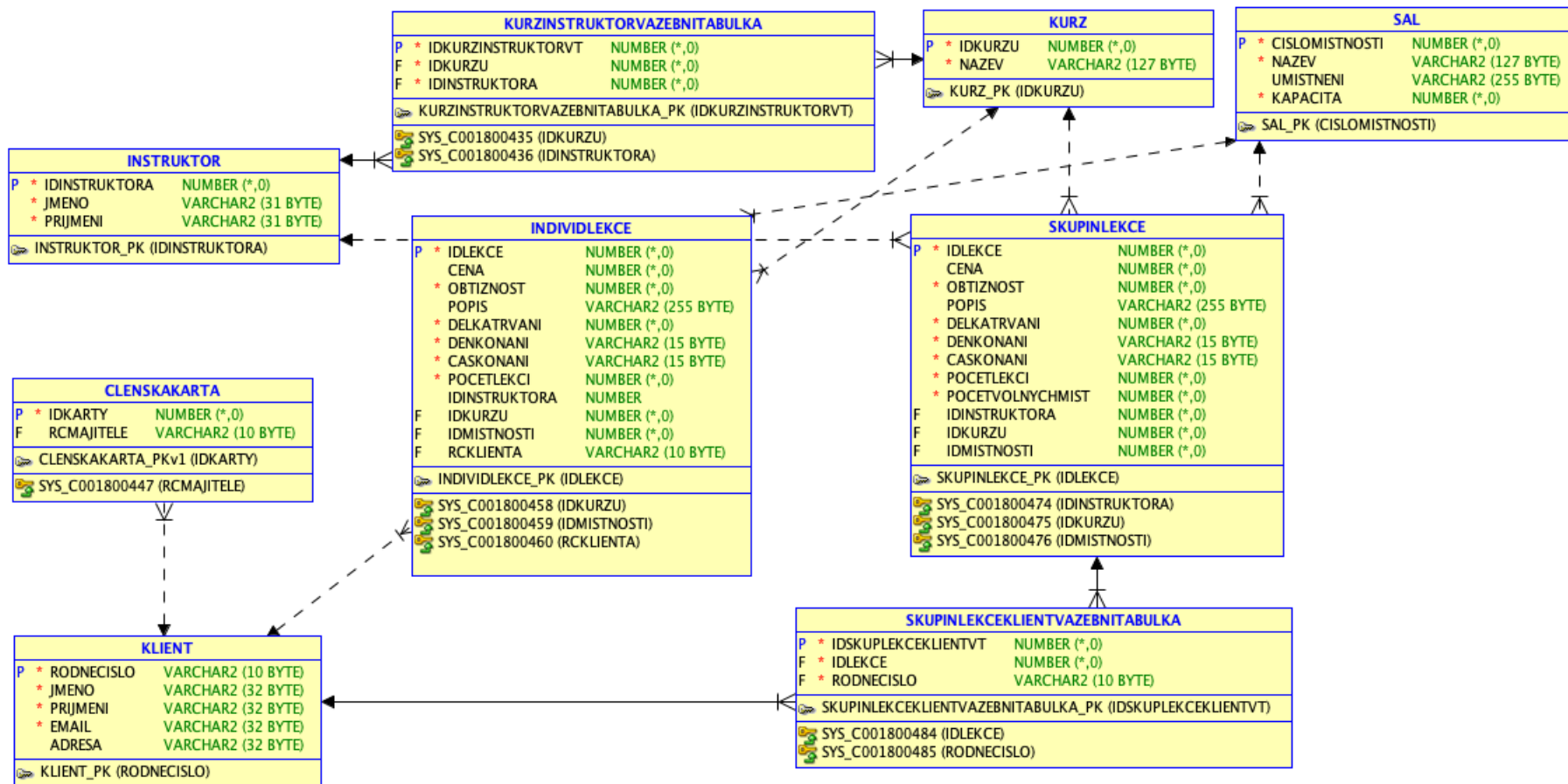
Poslední entitou je **Instruktor**, jež nabývá možností:

- nový kurz,
- správa kurzů,
- nová individuální nebo skupinová lekce,
- správa skupin a individuálních lekcí s návazným vypsáním skupinových a individuálních lekcí a možností zobrazení informací o kurzu.

## Fitness centrum (IUS - č. 58)



Obrázek 2: Model případů užití



Obrázek 3: Převod ERD na relační databázi

## 4 Triggery

První trigger slouží k automatickému generování ID členské karty, není-li explicitně uvedeno. Pomocí `create sequence` je generována posloupnost čísel. Členské karty jsou prohledány v cyklu a v případě chybějícího id karty je přidána hodnota, jenž jsme získali z vygenerované posloupnosti.

Druhy trigger vykoná smazání členské karty v případě odstranění, či při neexistenci jejího vlastníka. Opět dochází k projití tabulky členských karet pomocí cyklu. V případě, že bylo rodné číslo odstraněno, dojde k vykonání příkazu `delete` pro danou členskou kartu.

## 5 Procedury

První procedura slouží k vypsání průměrného věku klientely vypočteného z let všech klientů. Tato informace může fitness centru sloužit například k zlepšení marketingového zaměření na odpovídající věkovou skupinu. Jako první dojde k inicializování celkového počtu klientů a celkového společného věku na hodnotu 0. Poté v každém kroku cyklu dojde k inkrementaci počtu klientů a k výpočtu věku dle aktuálního roku. Věk je poté přičten k společnému věku. Na konci cyklu se zkontroluje, že se počet klientů nerovná nule, v takovém případě dojde k vyvolání chyby dělení nulou, a dojde k výpisu průměrného věku (celkový věk / počet klientů).

Druhá procedura provede výpočet celkové kapacity místností, jenž má fitness centrum k dispozici. Opět se před vstupem do cyklu inicializuje hodnota 0, nyní pro celkovou kapacitu. V každém kroku cyklu k této hodnotě přičteme kapacitu sálu. Cyklus je ukončen v momentě, kdy již nelze nalézt další data a výsledek je vypsán.

## 6 Explain plan

Explain plan byl proveden dvakrát nad stejným dotazem. První tabulka ukazuje provedení bez využití indexu, druhá s jeho použitím. Dotaz, jenž byl vykonán:

```
select instruktor.jmeno, instruktor.prijmeni, count(*) as counter
from instruktor inner join skupinLekce
  on skupinLekce.idInstruktora = instruktor.idInstruktora
  inner join sal on skupinLekce.idMistnosti = sal.cisloMistnosti
where sal.cisloMistnosti = 103
group by instruktor.jmeno, instruktor.prijmeni;
```

získá jméno a příjmení instruktora a počet lekcí, které má v uvedené místnosti.

ID	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	Select statement		2	146	6 (17)	00:00:01
1	Hash group by		2	146	6 (17)	00:00:01
2	Nested loops		2	146	5 (0)	00:00:01
3	Nested loops		2	146	5 (0)	00:00:01
4	Table access full	SkupinLekce	2	52	3 (0)	00:00:01
5	Index unique scan	SYS_C001797873	1		0 (0)	00:00:01



ID	Operation	Name	Rows	Bytes	Cost (% CPU)	Time
0	Select statement		2	146	5 (20)	00:00:01
1	Hash group by		2	146	5 (20)	00:00:01
2	Nested loops		2	146	4 (0)	00:00:01
3	Nested loops		2	146	4 (0)	00:00:01
4	Table access by index rowid batched	SkupinLekce	2	52	3 (0)	00:00:01
5	Index range scan	IndexMistnost	2		1 (0)	00:00:01

Před použitím indexu musí databázový systém zpracovat dotaz procházením tabulky řádek po řádku. Po použití indexu, který filtruje tabulku `skupinLekce` na základě sloupce `idMistnosti` došlo k optimalizaci počtu přístupů na disk, jenž se snížil i přes stejnou paměťovou náročnost.

## 7 Materializovaný pohled

Materializovaný pohled je uskutečněn pomocí jednoduchého dotazu `select` na údaje tabulky `Klient`. Často používané údaje jsou uloženy na disku a tím se sníží zatížení databáze. Pomocí klauzule `build immediate` se materializovaný pohled naplní daty hned po jeho vytvoření. Klauzule `refresh fast on commit` optimalizuje aktualizaci dat při změně údajů v tabulce `Klient`. Materializovaný pohled volá člen `xtomal02` pro databázi člena `xbobci03`.