



## IDS - Projekt č. 27 (Hotel)

Andrzej Szczepaniak (xszcze01), Roland Schulz (xschul06)

1.5.2022

### **Abstract**

Návrh IS recepce hotelu, který by poskytoval přehled o dostupných pokojích, základní údaje o zákaznících a jejich pobytech v hotelu, jejich požadavcích na služby, placení apod. Systém musí umožnit také rezervaci pokojů.

# Optimalizace

Query k optimalizaci:

Počet všech rezervací vytvořených hostem v posledních 30 dnech:

```
select "name", COUNT("reservationId") as reservation_count_last_30
from reservations natural join guests natural join users
where "start" between trunc(sysdate) - 30 and trunc(sysdate)
group by "name";
```

Před optimalizací:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	79	4 (25)	00:00:01
1	HASH GROUP BY		1	79	4 (25)	00:00:01
* 2	FILTER					
3	NESTED LOOPS		1	79	3 (0)	00:00:01
4	NESTED LOOPS		1	79	3 (0)	00:00:01
5	NESTED LOOPS		1	48	3 (0)	00:00:01
* 6	TABLE ACCESS FULL	RESERVATIONS	1	22	3 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID	GUESTS	1	26	0 (0)	00:00:01
* 8	INDEX UNIQUE SCAN	GUESTS_PK	1		0 (0)	00:00:01
* 9	INDEX UNIQUE SCAN	USERS_PK	1		0 (0)	00:00:01
10	TABLE ACCESS BY INDEX ROWID	USERS	1	31	0 (0)	00:00:01

Optimalizace vytvořením indexu na jméno uživatele:

```
/* index odstrani jeden NESTED LOOP */
create index name_id on users ("name");
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	79	3 (0)	00:00:01
1	SORT GROUP BY NOSORT		1	79	3 (0)	00:00:01
* 2	FILTER					
3	NESTED LOOPS		1	79	3 (0)	00:00:01
4	NESTED LOOPS		1	79	3 (0)	00:00:01
5	MERGE JOIN CARTESIAN		1	53	3 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	USERS	1	31	0 (0)	00:00:01
7	INDEX FULL SCAN	NAME_ID	1		0 (0)	00:00:01
8	BUFFER SORT		1	22	3 (0)	00:00:01
* 9	TABLE ACCESS FULL	RESERVATIONS	1	22	3 (0)	00:00:01
* 10	INDEX UNIQUE SCAN	GUESTS_PK	1		0 (0)	00:00:01
* 11	TABLE ACCESS BY INDEX ROWID	GUESTS	1	26	0 (0)	00:00:01

Výsledkem optimalizace pomocí indexu je ušetření jednoho NESTED LOOPS.

## Materializovaný pohled

Materializovaný pohled pro hosty hotelu pocházejících z Brna.

```
create materialized view HosteBrnaci  
refresh on commit as  
    select * from XSZCZE01.guests where "city" = 'Brno';
```

Pohled předpokládá dopředné přiřazení oprávnění uživatelem XSZCZE01 pro select z tabulky guests, např.:

```
grant all privileges on guests to XSCHUL06;
```

## Procedury

### Procedura pro vytvoření nového uživatele jako zaměstnance

```
create or replace procedure CreateEmployee(username in varchar(32),
                                           password in varchar(32),
                                           jmeno in VARCHAR(32),
                                           email in varchar(32),
                                           phone in varchar(16),
                                           badge in varchar(32)) as

declare uzivID int;
begin
    insert into users ("username", "password", "name", "email", "phone", "permissionId")
    values (username, password, jmeno, email, phone, 2) returning "userId" into uzivID;

    insert into employees ("badgeCode", "userId") values (badge, uzivID);
end;
```

Provolání procedury:

```
begin
    CreateEmployee ('pepa.vrana123',
                   'fjs18565$2!#',
                   'Pepa Vrána',
                   'pepa.vrana@company.com',
                   '711-252-252',
                   'E-223');
end;
```

### Procedura pro získání neobsazených hotelových pokojů

```
create or replace procedure VacantRooms as
begin
    select * from rooms where "roomId" not in (
        select "roomId" from reservations
        where "end" < sysdate
    );
end;
```

Provolání procedury:

```
begin VacantRooms (); end;
```

## Triggery

### Trigger pro přípravu nezaplacené platby při tvorbě rezervace

K Rezervaci vytvoří novou platbu a inicializuje ji na “ještě nezaplaceno” (`externalPaymentId = 0`).

```
create or replace trigger PaymentProRezervaci after insert on room_services
referencing new as nove for each row
begin
    insert into payments ("amountHellers", "date", "externalPaymentId", "guestId", "employeeId")
    values (nove."priceHellers",
           current_date,
           0,
           (SELECT "guestId" from reservations where nove."reservationId" = reservations."reservationId")
           nove."employeeId");
end;
```

### Trigger pro smazání uživatele při smazání zaměstnance (vyhození zaměstnance)

Nutno dodat že v naší současné implementaci tento trigger není využitelný kvůli deklarovaným omezením, avšak jiný trigger nás již nenapadl. Trigger by ale bylo možné využít v případě že omezení definovaná nejsou.

```
create or replace trigger SmazaniZamestnanceUzivatele before delete on users
referencing old as vyhozeny for each row
begin
    delete from employees where employees."userId" = vyhozeny."userId";
end;
```