



Dokumentace k IDS projektu

**Varianta 36: Evidence řidičů včetně dopravních
přestupků, evidence motorových vozidel**

Autor:

Dominik Pop <xpopdo00>

Adam Hos <xhosad00>

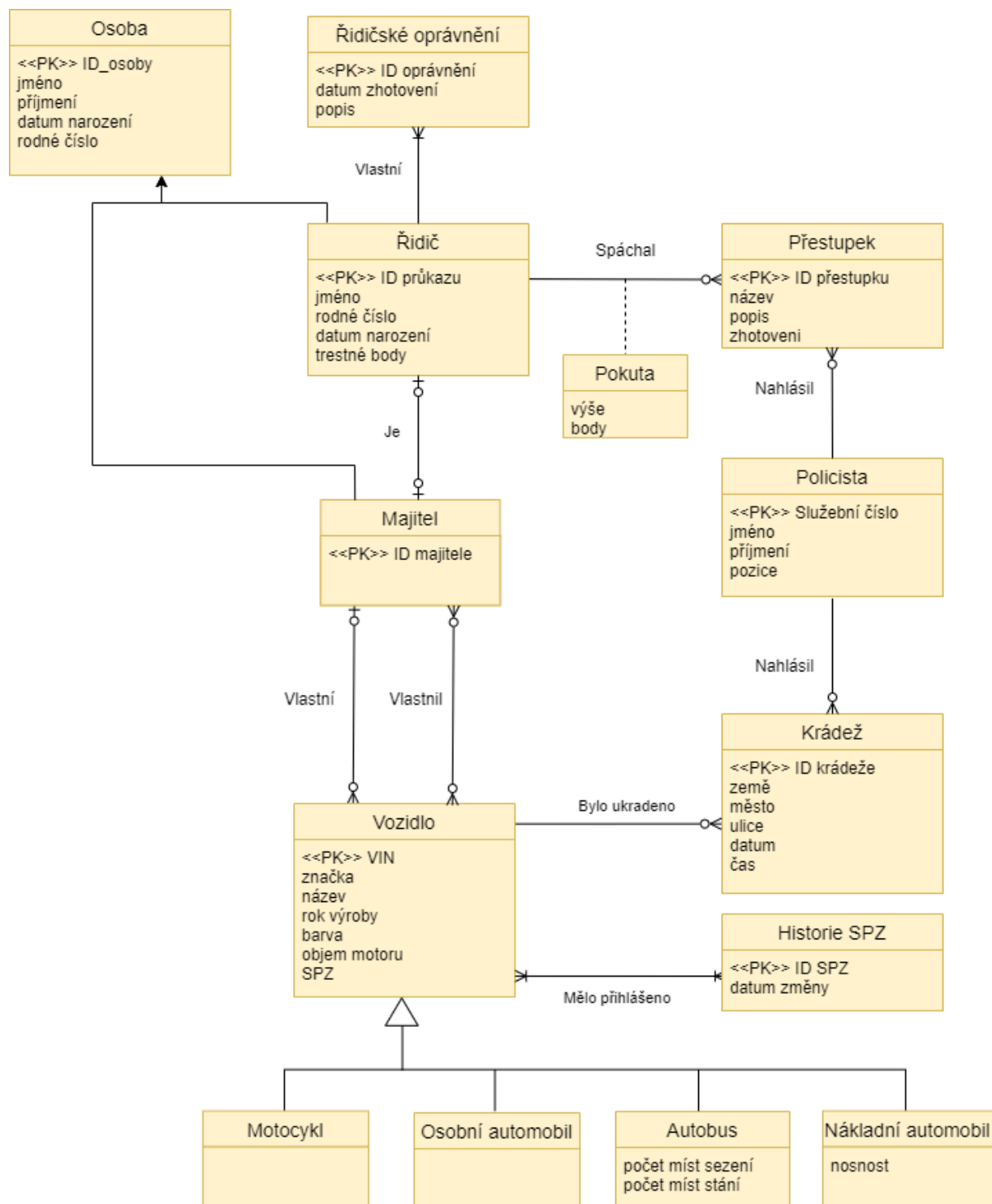
Obsah

Zadání	3
Návrh	3
Implementace.....	4
2. část – Inicializace	4
3. část – Selecty	4
4. část – Pokročilé objekty.....	5

Zadání

Navrhněte IS, který eviduje standardní údaje o řidičích a jejich přestupcích. Dále eviduje standardní údaje o motorových vozidlech a umožní ověřování údajů např. na dotaz dopravní policie. Systém obsahuje i evidenci kradených vozidel.

Návrh



Implementace

2. část – Inicializace

Zpočátku se v bloku několika DROP příkazů smažou všechny tabulky a sekvence. Nastaví se formát NLS_TIMESTAMP_FORMAT a NLS_DATE_FORMAT. Vytvoří se sekvence a tabulky. Ještě před vložením dat se vytvoří procedury a trigger. (viz 4. část) Následně se pomocí INSERT vloží data.

- Tabulky
 - Osoba
 - Ridic
 - Majitel
 - Vozidlo
 - NakladniAutomobil
 - Autobus
 - OsobniAutomobil
 - Motocykl
 - HistorieSPZ
 - MelSPZ (pro vazbu m-n mezi Vozidlem a HistorieSPZ)
 - RidickeOpraveni
 - Policista
 - Prestupek
 - Kradez. Ještě před vložením dat se vytvoří procedury a trigger. (viz 4. část). Následně se pomocí INSERT vloží data.

3. část – Selecty

Vytvořili jsme selecty s následující funkcionalitou:

- Vybere rodiče, kteří mají zákaz řízení a vlastní víc jak 1 řidičské oprávnění
- Vybere Přestupky spáchané řidičem Pepa Novák ze dne 10.4.2020
- Vybere Policisty, kteří zapsali přestupek dne 10.4.2020 ve výši 4500 Kč
- Zobrazí vozidla a jména jejich majitelů
- Zobrazí sumu, kterou museli řidiči zaplatit za všechny své přestupky
- Zobrazí rodiče, u kterých neexistují přestupky
- Zobrazí majitele, kteří mají vozidlo od společnosti Honda nebo Yamaha
- Zobrazí informace o vozidlech, které byly ukradené a informace o jejich krádeži zapsal Ludovic Cruchot
- Zobrazí policisty, kteří nezapsali žádnou krádež
- Zobrazí počet přestupku pro všechny řidiče, seřazené sestupně
- Zobrazí všechny přestupky a policisty, kteří je zapsali
- Zobrazí informace o všech vozidlech, které byly ukradeny

4. část – Pokročilé objekty

Byly vytvořeny následující objekty

- Triggery
 - policista_id – přiřazuje služební čísla od určité hodnoty (420000 – vymyšlené hodnota pro první číslo odkazu policejního pluku)
 - update_historie_SPZ – při přiřazení nové SPZ vozidlu se stará SPZ automaticky přidá do historie SPZ tohoto vozidla
 - update_trestne_body po přidání přestupku aktualizuje trestné body rodiče a případně přidá zákaz řízení
- Procedurey
 - vekPod18 – vypíše na výstup jméno, příjmení a věk všech řidičů mladších 18 let
 - zastoupeni_vozidel – vypočítá procentuální zastoupení jednotlivých typu vozidel (automobil, motocykl, autobus, nákladní automobil)
- Přidání práv uživateli ohledně všech tabulek a procedur
- Materializovaný pohled
 - ridici_view – vypíše čísla průkazů řidičů, zdali má tento řidič zákaz řízení, jeho počet trestných bodů a ID osoby. Je vytvořen s parametry BUILD IMMEDIATE a REFRESH ON COMMIT aby byl ihned vytvořen a zaručil aktualizaci při změně v tabulce Řidič

ID_PRUKAZU	ZAKAZ_RIZENI	TRESTNEBODY	ID_OSOBY
1 EG123456	1	12	1
2 FF112346	0	0	3
3 EA843115	0	0	4
4 EB693227	0	0	6
5 EB693248	0	0	5

Pomocí EXPLAIN PLAN jsme získali postup realizace operací, cenu operace potřebný čas

EXPLAIN PLAN FOR

```
SELECT O.jmeno, O.prijmeni, SUM(P.vyse) celkem_presupky
FROM Prestupek P, Ridic R, Osoba O
WHERE P.ID_prukazu = R.ID_prukazu AND R.ID_osoby = O.ID_osoby
GROUP BY O.jmeno, O.prijmeni;
SELECT * FROM TABLE(DBMS_XPLAN.display);
```

A výsledek získaný bez použití indexů je:

Plan hash value: 643572492							

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	

0	SELECT STATEMENT		1	56	7 (29)	00:00:01	
1	HASH GROUP BY		1	56	7 (29)	00:00:01	
2	NESTED LOOPS		1	56	6 (17)	00:00:01	
3	NESTED LOOPS		1	56	6 (17)	00:00:01	
4	NESTED LOOPS		1	38	5 (20)	00:00:01	
5	VIEW	VW GBC 10	1	24	4 (25)	00:00:01	
6	HASH GROUP BY		1	14	4 (25)	00:00:01	
7	TABLE ACCESS FULL	PRESTUPEK	3	42	3 (0)	00:00:01	
8	TABLE ACCESS BY INDEX ROWID	RIDIC	1	14	1 (0)	00:00:01	
* 9	INDEX UNIQUE SCAN	PK ID PRUKAZU	1		0 (0)	00:00:01	
* 10	INDEX UNIQUE SCAN	PK ID OSOBY	1		0 (0)	00:00:01	
11	TABLE ACCESS BY INDEX ROWID	OSOBA	1	18	1 (0)	00:00:01	

Predicate Information (identified by operation id):							

9 - access("ITEM 1"="R"."ID PRUKAZU")							
10 - access("R"."ID OSOBY"="O"."ID OSOBY")							
Note							

- this is an adaptive plan							

Pro zlepšení jsem vytvořily následující indexy pro všechny použité tabulky:

```
CREATE INDEX index_ridic ON Ridic (ID_pukazu, ID_osoby);
CREATE INDEX index_prestupek ON Prestupek (ID_pukazu, Vyse);
CREATE INDEX index_osoba ON Osoba (ID_osoby, jmeno, prijmeni);
```

Nyní při použití EXPLAIN PLAIN lze vidět, že se použily indexy a zlepšil se průběh operace

Plan hash value: 3107870334							

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	

0	SELECT STATEMENT		3	138	4 (25)	00:00:01	
1	HASH GROUP BY		3	138	4 (25)	00:00:01	
* 2	HASH JOIN		3	138	3 (0)	00:00:01	
* 3	HASH JOIN		3	84	2 (0)	00:00:01	
4	INDEX FULL SCAN	INDEX PRESTUPEK	3	42	1 (0)	00:00:01	
5	INDEX FULL SCAN	INDEX RIDIC	5	70	1 (0)	00:00:01	
6	INDEX FULL SCAN	INDEX OSOBA	6	108	1 (0)	00:00:01	

Predicate Information (identified by operation id):							

2 - access("R"."ID OSOBY"="O"."ID OSOBY")							
3 - access("P"."ID PRUKAZU"="R"."ID PRUKAZU")							
Note							

- this is an adaptive plan							

Závěr

Skript v jazyce SQL byl testován na školním Oracle serveru v SQLDeveloperu. Komunikace probíhala přes dedikovaný Discord server nebo osobně. Jako zdroj informací jsme používali IDS přednášky.