



**Project IDS
Gallery**

Database

Pavel Yablouski (xyadlo00)
Oleksii Korniienko (xkorni02)

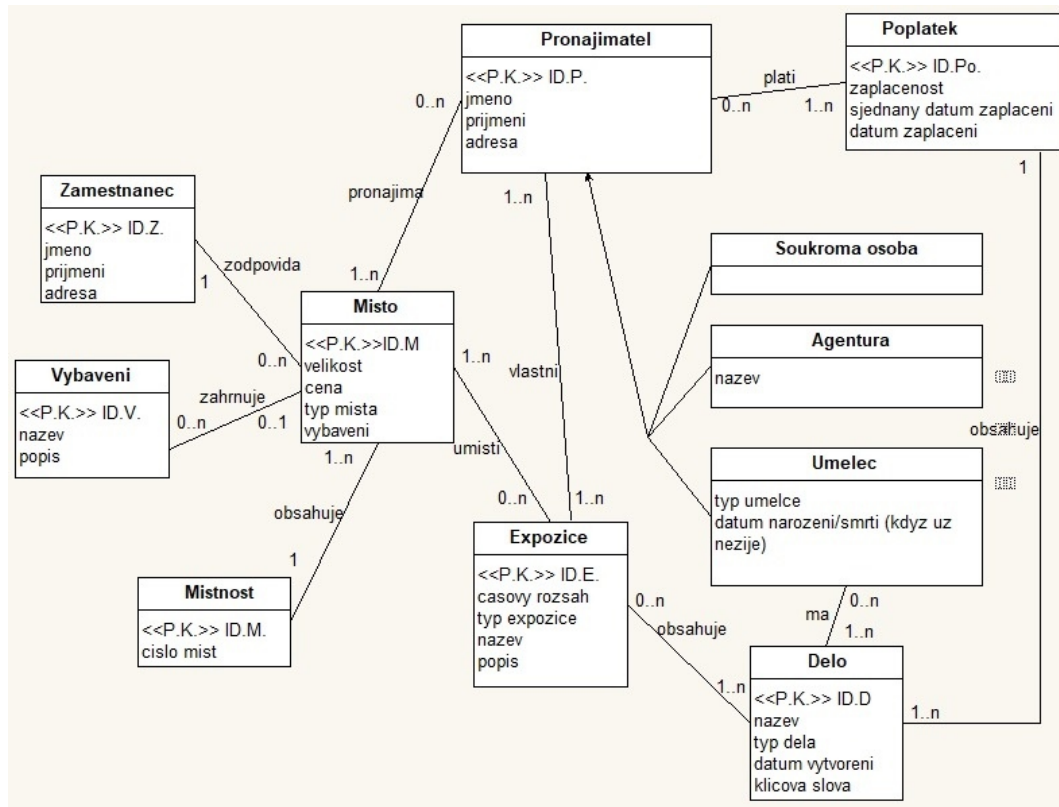
Brno University of Technologies
April, 2020

1 Task

Navrhnete IS pro galerii. Galerii tvoří několik místností (každá místnost obsahuje konkrétní expoziční místa různého typu - např. zeď, podlaha), ve kterých si umělci nebo agentury mohou pronajímat jednotlivá místa pro expozice svých uměleckých děl. Pronajímaná místa se liší co do velikosti, ceny, typu a dostupného vybavení (např. podstavec, osvětlení, bezpečnostní prvky, atd.). Systém by měl evidovat kdo si dané místo pronajal, od kdy do kdy a pro jakou expozici. Za každé místo zodpovídá příslušný zaměstnanec galerie. Systém by měl také uchovávat informace o expozicích jednotlivých umělců. Tyto informace by měly zahrnovat časový rozsah a typ expozice, konkrétní umělecká díla, jejich umístění atd. Dále systém musí umožňovat jednoduché vyhledávání umělců, jejich děl (podle typu, umístění, klíčového slova, atd.) a poskytovat souhrnné statistiky o umělcích, agenturách, expozicích, apod.). Umělci či agentury, které hodlají vystavovat, musí do sjednaného data uhradit příslušné poplatky, přičemž systém by měl poskytovat hlavnímu provozovateli (jen jemu) informace o všech poplatcích (zaplacených i nezaplacených).

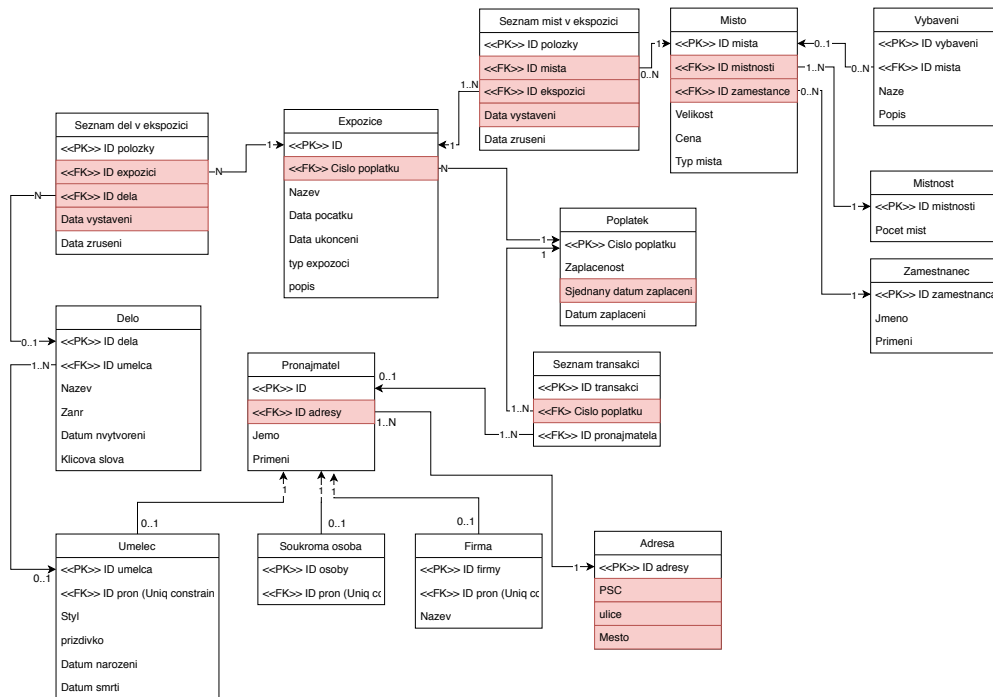
2 ER diagram

Here is entity relationship diagram



2.1 Transformed diagram

For implementation of database we transformed entity relationship diagram to database diagram



3 Implementation

3.1 Triggers

There are two triggers: *CHANGE_NULL_IN_VYBAVENI* and *SEQ_VYBAVENI*

3.1.1 CHANGE_NULL_IN_VYBAVENI

This trigger is used for changing row *popis* in tabel *VYBAVENI* if there is NULL value to sting 'neni popis'. This trigger works on insert to given tabel. In first implementation there was problem with muting tables and trigger couldn't work. This problem is solved using compound DML triggers.

3.1.2 SEQ_VYBAVENI

This trigger represents auto increment

	ID_VYBAVENI	ID_MISTO	NAZEV	POPIS
1	1	1	lampa	sveti jako slunicko
2	2	2	podstavec	(null)
3	3	1	ramecek	proste bozi
4	4	3	ramecek	(null)
5	5	(null)	podstavec	(null)
6	6	4	ramecek	(null)
7	7	(null)	stul	jako u kralovny
8	8	5	ramecek	(null)
9	9	(null)	stul	(null)

Figure 1: Before using trigger

	ID_VYBAVENI	ID_MISTO	NAZEV	POPIS
1	1	1	lampa	sveti jako slunicko
2	2	2	podstavec	neni popis
3	3	1	ramecek	proste bozi
4	4	3	ramecek	neni popis
5	5	(null)	podstavec	neni popis
6	6	4	ramecek	neni popis
7	7	(null)	stul	jako u kralovny
8	8	5	ramecek	neni popis
9	9	(null)	stul	neni popis

Figure 2: With using trigger

3.2 Procedures

Also, there are two procedures: *CHANGE_POPIS* and *COST_CHECK*.

3.2.1 CHANGE_POPIS

This procedure is designed to adjust the ABC table so that the item description is not blank. The description changes as follows: if there is at least one object that is not placed anywhere, then the description of the objects (which did not exist before) will be "used" or "not used", otherwise the description will be "description not specified".

3.2.2 COST_CHECK

This procedure is designed to calculate the cost of one square meter of each place separately. If the area is not yet calculated (0) will issue a warning, otherwise it will calculate and display a table of results.

3.3 Explain plan

Query is used for selecting column same rows from table *mista* with value in column *typ_mista* equals to *stena*

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	44	10 (20)	00:00:01
1	HASH GROUP BY		2	44	10 (20)	00:00:01
* 2	HASH JOIN		5	110	9 (12)	00:00:01
3	MERGE JOIN		5	95	6 (17)	00:00:01
* 4	TABLE ACCESS BY INDEX ROWID	MISTO	5	65	2 (0)	00:00:01
5	INDEX FULL SCAN	PK_MISTO	8		1 (0)	00:00:01

After using indexing on column *typ_mista*

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	44	9 (12)	00:00:01
1	HASH GROUP BY		2	44	9 (12)	00:00:01
* 2	HASH JOIN		5	110	8 (0)	00:00:01
* 3	HASH JOIN		5	45	6 (0)	00:00:01
4	TABLE ACCESS FULL	SEZNAM_DEL_V_EXPOZICI	3	9	3 (0)	00:00:01
5	TABLE ACCESS FULL	SEZNAM_MIST_V_EKSP0ZICI	5	30	3 (0)	00:00:01

3.4 Materialized view

The materialized view is used to collect and save the result of a table requirement without overflowing. In our case, the view (*MY_VIEW*) is created for another user's table (xkorni02), pre-granting privileges using *GRANT* for xyadlo00. The look should reflect the values of each place type in all areas.