

1. Vytvořte diagram ER konceptuálního schématu pro databázi na správu účtů v malé bance: [10]

Každý účet je identifikován dvojicí obsahující vlastní číslo účtu a kód banky. Kromě toho u každého účtu uchováváme jeho aktuální zůstatek a měnu, ve které je veden (např. CZK, EUR atp.). Rozlišujeme dva konkrétní druhy účtů, a to běžné a spořicí. Běžný účet má informaci o frekvenci zasílání výpisů, naopak spořicí účet má nepovinnou roční úrokovou sazbu. Každý účet je vlastněn výhradně jedním klientem. Klient je jednoznačně určen rodným číslem, dále u něj uchováváme nepovinnou emailovou adresu a libovolný počet telefonických čísel. V systému rovněž potřebujeme ukládat informaci, že nový klient mohl být doporučen jiným stávajícím klientem (nejvýše jedním). Mezi účty mohou být realizovány převody peněz. Každý převod je navázán na zdrojový a cílový účet. Obsahuje datum, čas, částku a pořadové číslo odchozí transakce, které je unikátní v rámci daného zdrojového účtu.

Nepřidávejte uměle generované identifikátory, alespoň jednou použijte ternární vztahový typ a specifikujte kritéria pokrytí a překrytí u ISA hierarchie. Namodelujte jen to, co je požadováno.

2. Transformujte ER schéma z předcházejícího příkladu do schématu v relačním modelu. Schémata jednotlivých tabulek popište textovou notací. Uvedte všechny klíče a cizí klíče, nepřidávejte uměle generované identifikátory. Předpokládejte relační model bez NULL hodnot. [10]

Užitečný symbol: \subseteq

3. Předpokládejme následující relační schéma databáze knihovny:

Čtenář (rodné číslo, jméno, příjmení, věk, město)

Kniha (signatura, název, autor, žánr)

Výpůjčka (id, datum půjčení, kniha, rodné číslo, datum vrácení)

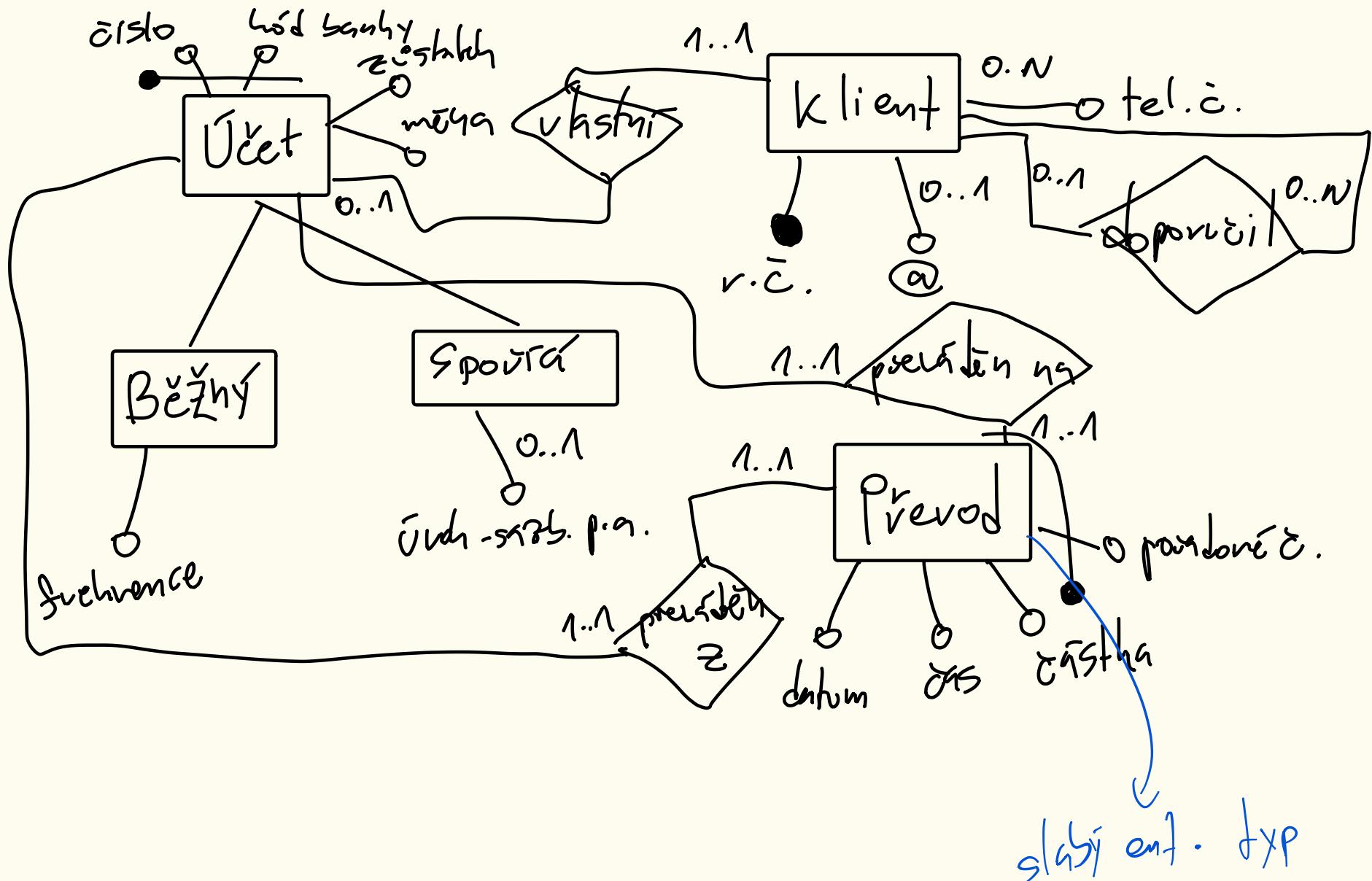
(kniha) \subseteq Kniha (signatura)

(rodné číslo) \subseteq Čtenář (ročné číslo)

Vytvořte SQL výrazy pro následující dotazy: [15]

- a) Najděte unikátní příjmení čtenářů starších 30 let, kteří si nikdy nepůjčili žádnou knihu patřící do žánru *detektivka* nebo *kuchařka*.
 - b) Najděte rodná čísla, jména a příjmení čtenářů, kteří alespoň jednou měli ve stejný okamžik vypůjčeny alespoň dvě knihy najednou.
 - c) Pro každé město zjistěte celkový počet realizovaných výpůjček libovolných knih od autora *Karel Čapek*. Omezte se však jen na výpůjčky uskutečněné čtenáři s křestním jménem jiným než *Karel*. Ve výsledku uveděte jen ta města, kde takový počet je roven alespoň 10.
4. Předpokládejme relační schéma z předcházejícího příkladu. **Vytvořte výraz pro SQL/XML dotaz**, kde pro každou realizovanou výpůjčku (na samostatném rádku výsledné tabulky) uvedete v prvním sloupci její id a ve druhém zkonstruujte XML hodnotu podle následujícího vzoru: [5]

```
<vypujcka pujceno="datum půjčení" vraceno="datum vrácení">
    <kniha>název knihy</kniha>
    <autor>autor knihy</autor>
</vypujcka>
```



Účet (číslo, hōd banky, zůstalek, nájem)

Klient (vodné číslo)

ÚčetEmail (vodné číslo)

Fk: (vodné číslo) ⊆ Klient (vodné číslo)

Tel (vodné číslo, číslo)

Fk: (vodné číslo) ⊆ Klient (vodné číslo)

Doporučil (kdo, koho)

Fk: (kdo) ⊆ Klient (vodné číslo)

Fk: (koho) ⊆ Klient (vodné číslo)

Vlastník (možné řešení, cíl, hodnota, mý)

Fh: (v·č.) ⊆ klient(možné řešení)

Fk: (objekt) ⊆ ředitel(cíl, hodnota)

Běžný (cíl, hodnota, funkce)

Fh: (objekt, hodnota) ⊆ ředitel(cíl, hodnota)

Spořit (cíl, hodnota)

Fh: (cíl, hodnota) ⊆ ředitel(cíl, hodnota)

Spořit úhrada (cíl, hodnota, garant)

Fh: (cíl, hodnota) ⊆ spořit(cíl, hodnota)

Převod (povolené řešení, datum, čas, cíl, hodnota)

Překlánění (z, f, hodnota)

Fh: (hodnota) ⊆ ředitel(cíl, hodnota)

Fh: (z) ⊆ ředitel(cíl, hodnota)

Překlánění hodnoty (hodnota, z)

Fh(hodnota) ⊆ ředitel(cíl, hodnota)

Fh(z) ⊆ ředitel(cíl, hodnota)

(3.) ^(a) SELECT DISTINCT C. prijmeni FROM Ctenar C
JOIN Vypisna V ON C.vohre-cislo = V.vohre-cislo
= C. vohre-cislo

JOIN Kniha K ON C.kniha = K - signatura
WHERE K.zahr NOT IN ('lektifikus', 'hukavik')
AND C.reh > 30.

(b) (DISTINCT)
SELECT C.vohre-cislo, C.jmeno, C.prijmeni
FROM Ctenar C
JOIN vypisna V1 ON C.vohre-cislo = V1.vohre-cislo
JOIN vypisna V2 ON C.vohre-cislo = V2.vohre-cislo
WHERE V1.datum-prijeni ≤ V2.datum-prijeni
AND V1.datum-prijeni ≥ V2.datum-prijeni
AND V1.id <> V2.id

5. Opět předpokládejme stejné relační schéma. Formulujte výrazy v relační algebře pro tyto dotazy: [10]

Čtenář (rodné číslo, jméno, příjmení, věk, město)
Kniha (signatura, název, autor, žánr)
Výpůjčka (id, datum půjčení, kniha, rodné číslo, datum vrácení)
(kniha) \subseteq Kniha (signatura)
(rodné číslo) \subseteq Čtenář (ročné číslo)

- a) Signatury a názvy knih od *Karla Čapka*, které nikdy nebyly vypůjčeny ani jedním čtenářem z *Liberce* starším než 30 let.
- b) Jména a příjmení čtenářů z jiného města než *Liberce*, kteří si alespoň jednou půjčili každou knihu od *Karla Čapka*.
6. Najděte všechny klíče v relačním schématu s množinou atributů $\{K, L, M, N, O\}$ a množinou funkčních závislostí $\{K \rightarrow MO, L \rightarrow O, KL \rightarrow N, N \rightarrow L, MNO \rightarrow K\}$. Postup komentujte, použijte algoritmus Lucchesi-Osborn. Následně určete normální formu každé jednotlivé závislosti a také celého uvedeného schématu. [10]

7. Cizí klíče a referenční integrity [5]

- a) Cizí klíč musí být v referující a stejně tak i v referenční tabulce klíčem (PRIMARY KEY nebo UNIQUE)
- b) U daného cizího klíče může být definována nejvýše jedna referenční akce obsluhující událost DELETE, UPDATE nebo INSERT
- c) Referenční akce popisuje dopad na řádky v referující tabulce, pokud by příslušná aktualizační operace v referenční tabulce způsobila narušení referenční integrity
- d) Pokud není definována žádná akce, chování je identické jako u varianty NO ACTION
- e) Výsledek referenčních akcí NO ACTION a RESTRICT může být v případě použití vhodně implementovaných triggerů rozdílný

- a) true false
b) true false
c) true false
d) true false
e) true false

2.

8. Databázové pohledy [5]

- a) Pohled je definovaný SELECT dotazem a obsahuje data, která odpovídají výsledku tohoto dotazu v okamžiku vytvoření daného pohledu pomocí CREATE VIEW
- b) Existující pohledy je možné používat ve FROM klauzulích stejným způsobem jako obyčejné tabulky
- c) Pohled může být závislý na jiném pohledu, dokonce i tranzitivně, ale nikdy nemůže vzniknout jako spojení (join) více tabulek a/nebo pohledů
- d) Je-li uvedena klauzule WITH LOCAL CHECK OPTION, provedení aktualizační operace bude umožněno jen tehdy, pokud změna bude v pohledu viditelná (např. u operace INSERT bude vkládaný řádek splňovat případné vyhledávací podmínky v definici pohledu)
- e) Klauzule WITH CHECK OPTION je nepovinná; je-li však uvedena a nemí-li vybrána její varianta, za výchozí je považována varianta CASCDED

- a) true false
b) true false
c) true false
d) true false
e) true false

9. B⁺-stromový index [5]

- a) B⁺-strom je výškově vyvážený strom, časová složitost základních operací je logaritmická
- b) Vnitřní uzly (včetně kořenového) mají jinou strukturu než uzly listové, ty totiž obsahují říd ukazatele na záznamy do primárního souboru
- c) Vnitřní uzly obsahují řádově i stovky dělících hodnot, každý uzel je fyzicky uložen jako jeden blok (stránka) indexového souboru
- d) B⁺-stromový index může být konstruován jen pro vyhledávací klíče, které jsou unikátní
- e) Uzly v listové úrovni B⁺-stromu jsou navzájem provázány pomocí ukazatelů

- a) true false
b) true false
c) true false
d) true false
e) true false

10. Zotavitelnost (recoverability) rozvrhu (historie) v transakčním zpracování

[5]

- a) Rozvrh (historie) je nezotavitelný (unrecoverable) právě tehdy, když pro každou úspěšně dokončenou transakci T platí, že operaci COMMIT provedla nejdříve tehdy, když skončily všechny další transakce ovlivňující T a) true false
- b) Pokud každá transakce může číst jen úspěšně potvrzená (commitovaná) data, garantujeme kaskádové rušení transakcí (cascade aborts) b) true false
- c) Pokud je precedenční graf (precedence graph) acyklický, daný rozvrh (historie) už nezbytně nutně musí být zotavitelný (recoverable) c) true false
- d) Dvoufázový zamýkací protokol není schopen garantovat zotavitelnost (recoverability) rozvrhu (historie) ani předcházet kaskádovému rušení transakcí d) true false
- e) Striktní dvoufázový zamýkací protokol je schopen garantovat zotavitelnost (recoverability) rozvrhu (historie) stejně jako je schopen předcházet kaskádovému rušení transakcí e) true false

11. Striktní dvoufázový zamýkací protokol

[5]

- a) Pokud chce transakce provést operaci `write(A)`, musí nejprve úspěšně získat (nebo už vlastnit) exkluzivní nebo alespoň sdílený zámek pro entitu A a) true false
- b) Všechny požadované zámky musí být u striktního dvoufázového zamýkacího protokolu získány najednou hned na úplném začátku transakce b) true false
- c) Všechny získané zámky jsou uvolněny až na úplném konci transakce, tedy jako inherentní součást provádění operací COMMIT nebo ABORT c) true false
- d) Striktní dvoufázový zamýkací protokol garantuje, že precedenční graf je acyklický, a tedy že rozvrh (historie) je konfliktově uspořádatelný (serializovatelný) d) true false
- e) Striktní dvoufázový zamýkací protokol je schopen garantovat zotavitelnost rozvrhu (historie), ale už není schopen předcházet kaskádovému rušení transakcí e) true false

12. Pojmenujte a vysvětlete 4V charakteristiky pojmu Big Data.

[5]

13. Vyjmenujte a vysvětlete jednotlivé Coffmanovy podmínky. S jakou problematikou souvisí a jakým způsobem je můžeme použít?

[5]

14. Vysvětlete použití prioritní fronty v rámci algoritmu vnějšího třídění (N-way External Merge Sort), využití stránek systémové paměti a dopad na velikost běhů.

[5]

ER model

Sensor
 id-sensor SERIAL PK
 mesto VARCHAR(64) NOT NULL
 ulice VARCHAR(64) NOT NULL
 UNIQUE (mesto, ulice)

Kamera

id-sensor	INT	FK		
max-velikost	DECIMAL(2,10)	NOT NULL		
presnost	INT	NOT NULL		
			vaha	
			id-sensor	INT FK
			min-vazivost	INT NOTNULL
			presnost	INT NOTNULL

Mereni

id-mereni	SERIAL	PK	
sprz	VARCHAR(10)	FK	
cas	DATETIME	NOT NULL	
velikost	DECIMAL(5,2)		
vaha	DECIMAL(10,2)		

id-sensor

INT

Hasi
 id-presneph
 id-mereni

FK

Fk
 Fk

Presneph

id-presneph

SERIAL

PK

drvod	TEXT	NOT NULL	
zaplavens	BOOLEAN	NOT NULL	DEFAULT FALSE
cispl	DECIMAL(10,2)	NOT NULL	
vozidlo			

id-vozidlo

spz	VARCHAR(10)	PK	
max-w	VARCHAR(64)	UNIQUE	NOT NULL
magitek	DECIMAL(10,2)		
	CHAR(64)		

spz
 max-w
 magitek

	not NULL		
			specifikem id-vozidlo id-mestneph

$$F = \{ A \rightarrow BEF, BC \rightarrow DE, BDE \rightarrow F, ADF \rightarrow CE, \\ E \rightarrow CBD \}$$

$$G = \{ A \rightarrow B, AB \rightarrow E, AD \rightarrow C, BC \rightarrow E, BCE \rightarrow FD, \\ E \rightarrow C, CE \rightarrow B \}$$

$$G_1: A \rightarrow B$$

$$A \rightarrow BEF$$

$$A \rightarrow B \checkmark$$

$$G_2: AB \rightarrow E$$

$$A \rightarrow BEF$$

$$A \rightarrow E$$

$$AB \rightarrow E \checkmark$$

$$G_3: AD \rightarrow C$$

$$A \rightarrow BEF$$

$$A \rightarrow B$$

$$A \rightarrow E$$

$$A \rightarrow F$$

$$ADF \rightarrow CE$$

$$ADF \rightarrow C$$

$$AD \rightarrow C \checkmark$$

$$G_4: BC \rightarrow E$$

$$BC \rightarrow DE$$

$$DE \rightarrow E$$

$$BC \rightarrow E \checkmark$$

$$G_5: BCE \rightarrow FD$$

$$E \rightarrow CBD$$

$$CBD \rightarrow D$$

$$G_6: E \rightarrow C$$

$$E \rightarrow CBD$$

$$E \rightarrow C \checkmark$$

$$G_7: CE \rightarrow B$$

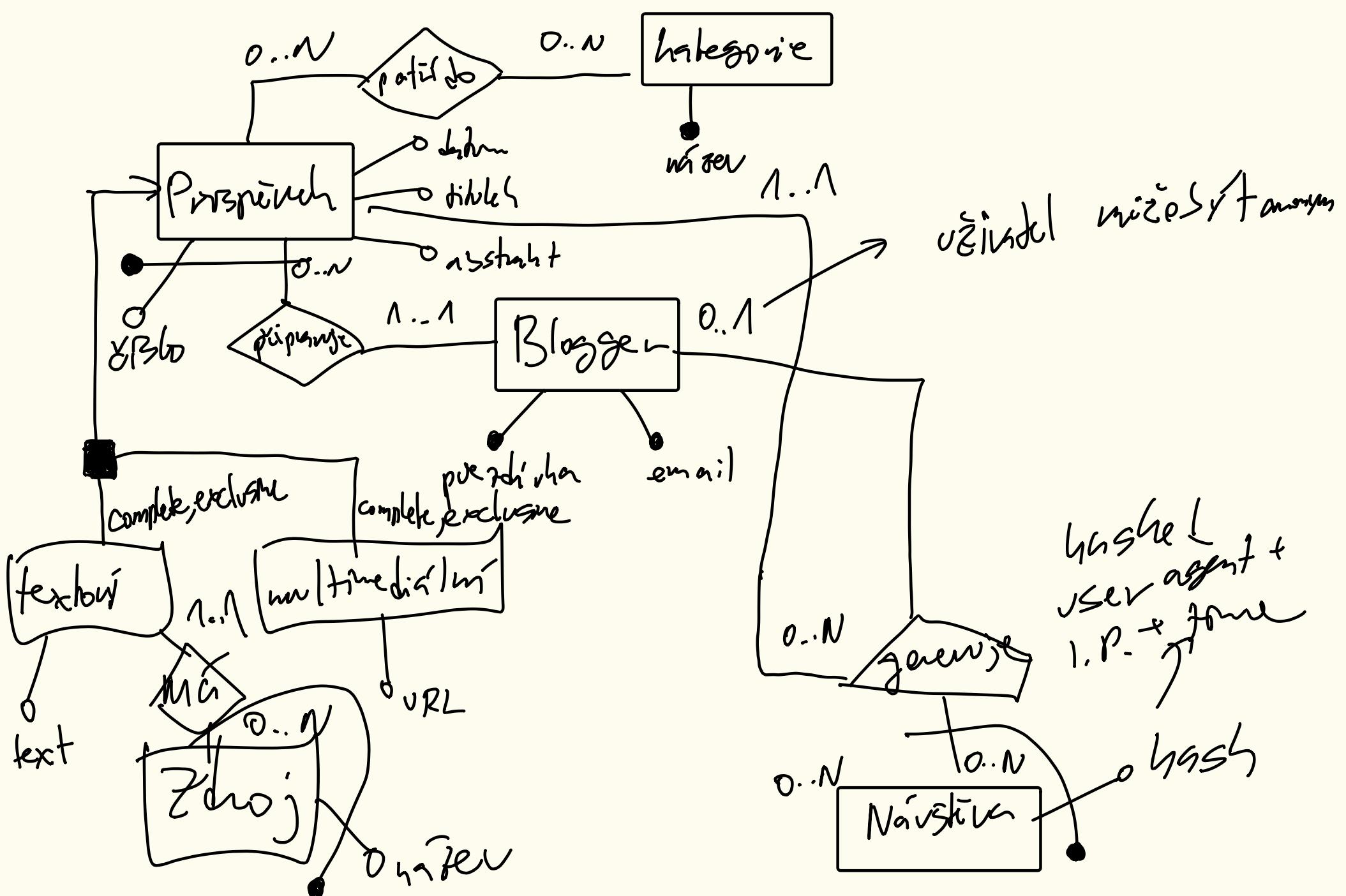
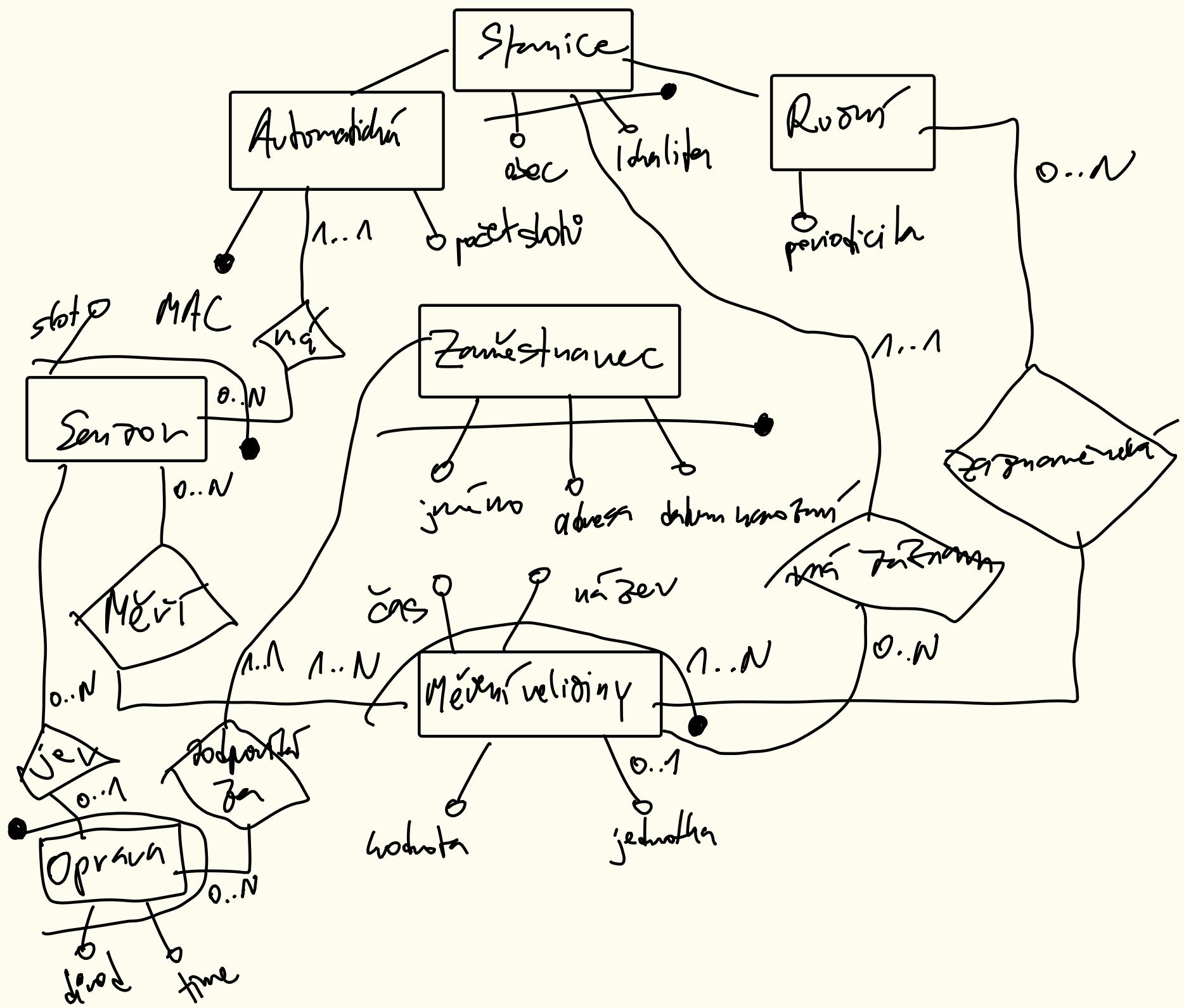
$$E \rightarrow D$$

$$BDF \rightarrow F$$

$$BCE \rightarrow FD \checkmark$$

$$E \rightarrow B$$

$$CE \rightarrow B \checkmark$$



Blogger (pesstika, email)

Příspěvek (cílo, pesstika, datum, titulek, soubor)

Fh: (pesstika) ⊆ Blogger (pesstika)

Kategorie (název)

Párov do (název, číslo, pesstika)

Fh: (číslo) ⊆ kategorie (název)

Fh: (číslo, pesstika) ⊆ Příspěvek (cílo, pesstika)

Textový (cílo, názv, text)

Fh: (číslo, pesstika) ⊆ Příspěvek (cílo, názv)

2NF Textový (číslo, pesstika, text zdroj)

Multimedialní (číslo, pesstika, URL)

Fh: (číslo, pesstika) ⊆ Příspěvek (cílo, pesstika)

Zdroj (číslo, pesstika, název)

Fh: (číslo, pesstika) ⊆ Textový ()

Návštěva (čas)

Galerie (čas, číslo, pesstika, Yagger)

* Fh: (číslo, pesstika) ⊆ Příspěvek

Fk: (číslo, pesstika) ⊆ Blogger (pesstika)

Generische Anonym (aus, sys, prg, dsl)

X

ER Blogger

id-blogger	S	PK
password	V(32)	NN
email	V(64)	NN

NN = NOT NULL
 U = UNIQUE $V(x) = \text{VARCHAR}(x)$ S = SERIAL
 $\text{INT} = \text{INTEGER}$ $O(x,y) = \text{DECIMAL}(x,y)$

Prispech

id-prispech	S	
id-blogger	INT	
datum	DATETIME	NN
titel	VARCHAR(64)	NN
abstrakt	TEXT	NN

Multimedialität

PK	—	id-prispech	INT	FK
FK	—	URL	TEXT	NN
	Textory			
	id-prispech	INT	FK	
	text	TEXT	NN	

Wursten

id-wursten	S	PK
CAS	DATETIME	NN
id-prispech	INT	FK NN
zettel	INT	FK NN
hash	BYTTEA	NN

Zettel

id-zettel	S	PK
text	TEXT	NN

Prispech

id-prispech	INT	PK
id-wursten	INT	FK
id-prispech	INT	FK
hash	BYTTEA	NN
wurst	V(64)	NN

Wursten

SELECT b.prezdivka, AVG(LEN(l.text))

FROM Blogger b

JOIN Prispevok p ON p.id_blogger = b.id_blogger

JOIN Textbox l ON l.id_prispevok = p.id_prispevok

WHERE p.datum BETWEEN '01-01-2022' AND '12-31-2022'

GROUP BY (b.prezdivka)

HAVING COUNT(p.id_prispevok) > 15

ORDER BY LEN(l.text), prezdivka DESC

SELECT n.cas, n.vzivatel, n.hash

FROM Prispevok p → next NULL

WHERE p.titulek = 'Super prispevok'

JOIN Naslask n ON n.id_prispevok = p.id_prispevok

SELECT b.prezdivka

FROM Blogger b

JOIN Prispevok p ON p.id_blogger = b.id_blogger

JOIN Jezercaski_js ON js.id_prispevok = p.id_prispevok

JOIN kategoriek ON k.id_kategorie = js.id_kategorie

GROUP BY (k.jmeno)

HAVING COUNT(SELECT * FROM k WHERE k.jmeno = 'Jirka')

= 0

SELECT b.name FROM Blogger b
 WHERE NOT EXISTS (
 SELECT 1 FROM Prispevok p WHERE b.id-b=p.id-b
 JOIN JeSacasti js ON js.id-p=p.id-p
 JOIN Kategorie k ON k.id-k=js.id-k
 WHERE k.jmeno = 'doplava')
 AND

(7.) vytvorení indexu na tabuľke kategorii
(poté v log. čase)

(8.) 3NF → neexistujúca závislosť nesplňuje
vyprávad premeníť entity zo dvoj jazero
atribut Textový príspevok

(9) Armstrong

identita $I \rightarrow I$

$$\begin{array}{c}
 \text{dekompozícia} \quad A \rightarrow BC \\
 \hline
 \Rightarrow A \rightarrow B \qquad \qquad \text{kompozícia} \\
 \qquad \qquad A \rightarrow C \qquad \qquad \qquad A \rightarrow C \\
 \hline
 \Rightarrow A \rightarrow BC
 \end{array}$$

reflexivita:

$$A \subset X \Rightarrow X \rightarrow A$$

transitivita

$$A \rightarrow B \quad B \rightarrow C \Rightarrow A \rightarrow C$$

rozštrenie:

$$A \rightarrow B \Rightarrow A_1 \rightarrow B_1 \wedge A_2 \rightarrow B_2 \quad (\text{potreba, odstrániť redundanciu -...})$$

využíta pri dedukcii
veľká závislosť v
relačných dátobázach

(10) Transacce slouzí v uživateli v TCP protocolu jeho celku, pro když je důležitý vysoký počet uživatelů, edikci lze zavést dle.

Pohled by za příklad dojde k selhání jehoho inspekce, osobní identita neexistuje (COMMIT, ROLLBACK)
(BEGIN?)

splituse

A - atomicity → všechny
C - consistency → každý má pouze jednu stanovu
I - isolate → samostatná transakce reaguje s ostatními
D - durability → závěry jsou trvalé

$$F = \{B \rightarrow ACF, CD \rightarrow EF, CEF \rightarrow A, ABE \rightarrow DF, F \rightarrow CDE\}$$

$$G = \{B \rightarrow C, BC \rightarrow F, BE \rightarrow D, CD \rightarrow F, CDF \rightarrow AE, F \rightarrow D, DF \rightarrow C\}$$

$$G1: B \rightarrow C \quad G2: BC \rightarrow F \quad G3: BE \rightarrow D$$

$$B \rightarrow ACF$$

$$BC \rightarrow B$$

$$BF \rightarrow B$$

$$B \rightarrow C \checkmark$$

$$B \rightarrow ACF$$

$$B \rightarrow ACF$$

$$B \rightarrow F$$

$$B \rightarrow F$$

$$BC \rightarrow F \checkmark$$

$$F \rightarrow CDE$$

$$F \rightarrow D$$

$$G4: CD \rightarrow E$$

$$CD \rightarrow EF$$

$$CD \rightarrow E \checkmark$$

$$CDF \rightarrow CD$$

$$G6: F \rightarrow D$$

$$CD \rightarrow EF$$

$$F \rightarrow CDE$$

$$CD \rightarrow F$$

$$F \rightarrow D \checkmark$$

$$CD \rightarrow E$$

$$G7: DF \rightarrow C$$

$$CE \rightarrow A$$

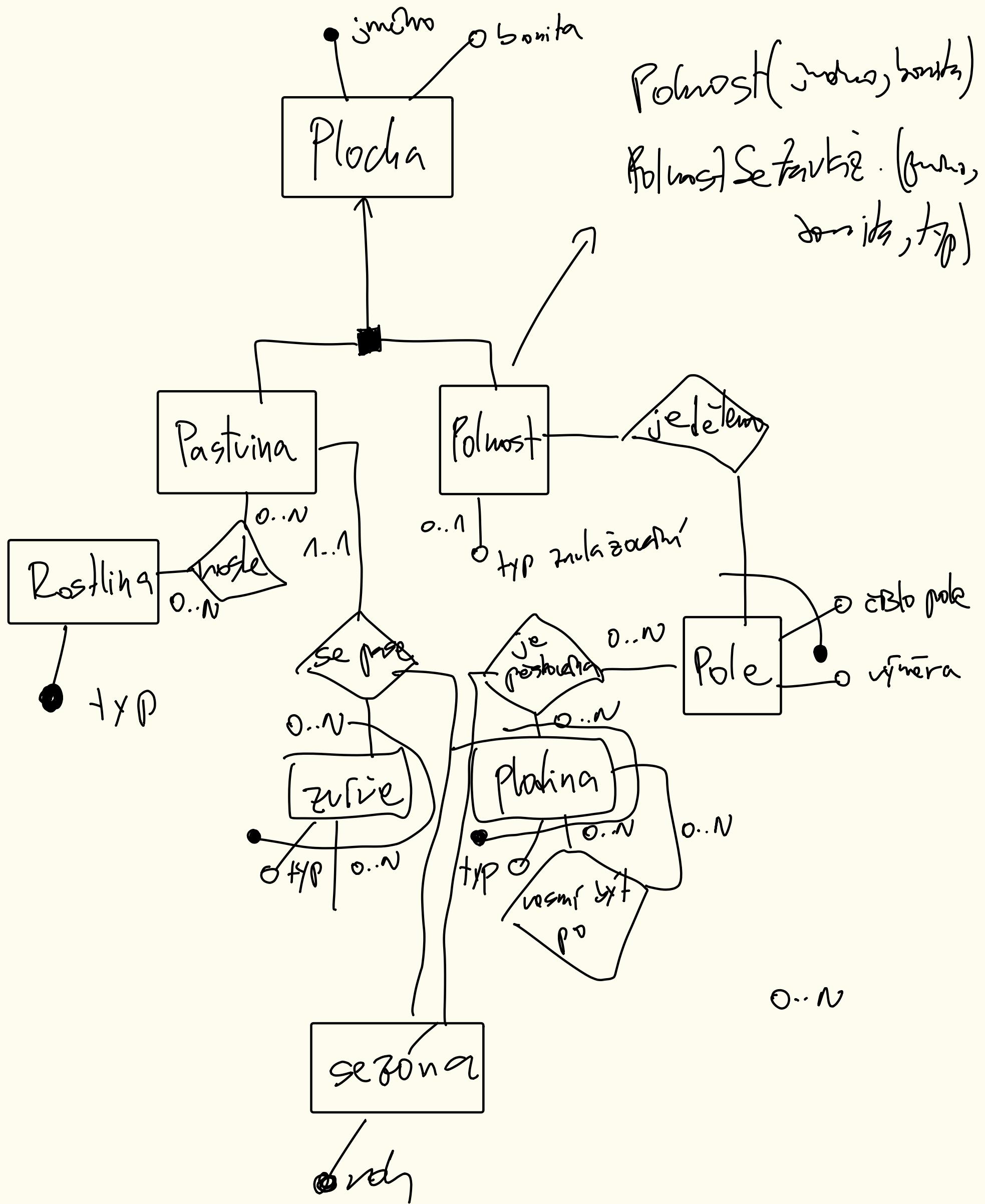
$$DF \rightarrow F$$

$$CDF \rightarrow AE \checkmark$$

$$F \rightarrow CDE$$

$$DF \rightarrow C$$

$$F \rightarrow C$$



Plachae

id-plachae \subseteq PK
jmonos V(bh) \cup NN
omita D(10,2) NN

Pischiae

id-plachae INT Fh
id-pischiae \subseteq Ph

Rosthiae

id-rosthiae \subseteq Ph
jmonos V(bh) NN

Phosphae

id-pischiae Fh
id-rosthiae Fh
Ph(id-pischiae, id-rosthiae)
Ph(id-pischiae, id-rosthiae)

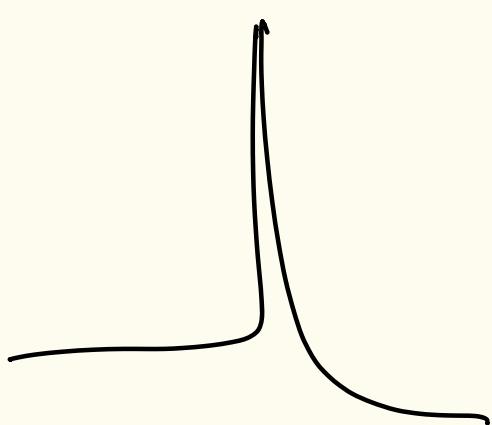
Zvire

id-zvire

Hf

Sezona Zvire

id-sezona



Polmost

id-plachae INT Fh
typ-polmost V(bh)
id-polmost Sph
Pole (jmonos)
Fh: (jmonos) \subseteq Polmost(jmonos)

Pole

id-polmost INT Fh
omita D(10,2) NN
cisto pole INT NN
id-pole \subseteq Ph
vng (id-polmost, cisto pole)

Sezonal Plachae

SELECT P · mzev
FROM Pole P
WHERE NOT EXISTS(
SELECT 1 FROM JePestovana JP
JOIN Plodina pl ON pl.id-pole = jp.id-pole
WHERE jp.id-sezona > 2015)
AND pl.jmeno = 'řepe obecné' AND jp.id-pole = pl.id-pole
ORDER BY (P.mzev) ASC

SELECT pl.jmeno, SUM(s.vymena)
FROM Pole P
JOIN JePestovana JP ON jp.id-pole = P.id-pole
JOIN Plodina pl ON jp.id-plodina = pl.id-plodina
JOIN Sezona s ON jp.id-sezona = s.id-sezona
AND s.id-plodina = pl.id-plodina
GROUP BY (pl.jmeno)
WHERE s.id-sezona > 2010
HAVING SUM(s.vymena) > 100

ORDER BY (pl.jmew) ASC, SUM(s.vmen) DESC

SELECT pst.jmew, z.jmew

FROM Pushing PST

JOIN SePse sp ON sp.id-pashan =
pst.id-pashina

JOIN Sazon s ON s.id-pastura = s.id-pash

JOIN Zine z ON z.id-zine = sp.id-zane

WHERE s.id = 2023

UNION

SELECT p.jmew, pl.jmew

FROM Pole P

JOIN JeRestraun JP ON jp.id-pole =
p.id-pole

JOIN Sazon s ON s.id-pole = s.id-pole

JOIN PlotraN pl ON pl.id-plotra = jp.id-plotra

WHERE s.id = 2023

(7) implicit constraints trigger BEFORE INSERT
proceed by & check, then se množ gložen
nechovat → plodná & min. efektivní
poloh dano, insertuje se založí
(poprvé do transakce?)

(8) 3NF → neexistují žádné
transf. funkce závislosti na hřešich

zavedení redundantních informací (spojení
dnu tabulek dohromady)

(A) pro spolužití všech může dojít k destrukci
cyklického úkolu → destrukce může být v celém
výkonu významně → proces s se může dočasně
ztrátit výkon a může sami

hold & wait → proces může zůstat o
druhé možnosti i když
vždy jeden drží
zájem o vložení → jeden může zůstat o
místo jednoho procesu může