UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

*Informační systém pro závody koloběžek*

Seminární práce z předmětu Databázové systémy I

Martin Pilný

V Chlumci nad Cidlinou dne 26.12.2022

Obsah

[Analýza 10](#_Toc70355807)

[ERD 13](#_Toc70355808)

[Relační model dat 16](#_Toc70355809)

[Fyzický model dat 17](#_Toc70355810)

[Výčet SQL dotazů 18](#_Toc70355811)

[Závěr 22](#_Toc70355812)

[Přílohy 23](#_Toc70355813)

Analýza

**1) Kompletní scénář min. 350 znaků, kterému musí odpovídat modely!!!**

Tato databáze eviduje údaje o závodech koloběžek, které se konají na různých místech. Rozlišuje se i typ závodu, tzn. že můžou být závody rozlišovány na rychlostní závod,

vytrvalostní závod nebo také štafetový závod. Pokaždé se závodu účastní různí závodníci a jednotlivé závody jsou určeny pro různé typy koloběžek.

Databáze také rozlišuje startovné pro jednotlivé závodníky. Na závěr databáze také umí vyhodnotit vítěze a poraženého závodu.

**2) Strukturální a procedurální pravidla**

STRUKTURÁLNÍ PRAVIDLA

* ZAVODNIK se může pouze 1 registrovat do závodu
* ZAVODNIK se může zúčastnit více závodů
* Každý ZAVODNIK musí mít přiřazenou adresu.
* Startovní číslo ZAVODNIKA může být pouze jedinečné pro každý závod
* Pohlaví závodníka může mít pouze tři hodnoty, ‚muž‘, ‚žena‘ nebo ‚nechci uvádět‘.

PROCEDURÁLNÍ PRAVIDLA

V tabulce ZAVODNICI může být uloženo pouze jeden záznam pro každého závodníka.

Může být definována procedura pro výpočet celkového počtu závodníků v dané kategorii nebo pohlaví.

může být definována procedura pro kontrolu, zda v tabulce ZAVODNICI jsou uvedeny všechny potřebné údaje pro každého závodníka.

**3) Jednoduchá analýza vyplývající ze scénáře, která zahrnuje označení entit (např. tučně), označení atributů (kurzívou).**

Tato databáze je mnou vymyšlená databáze vytvářející fiktivní závod koloběžek.

**Osoba** – Hlavní entita představuje osobu, Obsahuje atributy id-osoby, *jmeno*, *prijmeni*, *datum*\_*narozeni*, *telefonni\_cislo*, *typ\_osoby*, *vek*. Každá osoba musí vyplnit všechny tyto atributy pro registraci do závodu nebo jako pořadatel závodu.

**Adresa** – Tato entita se odkazuje na výše zmíněnou tabulku **Osoba**. Každý musí vyplnit údaje o vlastní adrese, proto má tyto atributy: *id\_adresy*, *cislo\_popisne*, *ulice*, *mesto*, *psc*, *stat*

**Poradatel –** tato entita představuje pořadatele závodu. Je podtypem entity **osoba.** Obsahuje atribut *pozice.*

**Zavodnik –** tato entita označuje závodníka. Je podtypem entity **osoba.** Obsahuje atributy *uroven, pohlavi.* Atribut *pohlavi* je nepovinný a pokud pohlaví závodník nechce uvádět, tak nemusí.

**Závod –** tato entita představuje samotný závod. Obsahuje atributy *id\_zavod, nazev, datum, delka, místo.*

**Registrace\_zavodu –** tato entita označuje registraci závodu. Obsahuje atributy *id\_registrace, startovni\_cislo, cas, cas\_v\_cili,*

**Vysledek –** po závodu zobrazí výsledky závodu. Obsahuje atributy *poradi, cena.*

**Typy\_zavodu –** určuje jaký to je typ závodu. Obsahuje atributy *id\_zavodu, nazev, odmena.*

**Typy\_kolobezek –** určuje jaké jsou typy koloběžek. Obsahuje atributy *id\_typ*, *nazev, max\_rychlost, brzdy.*

**Typy\_kategorii –** určuje jaké jsou typy kategorií. Obsahuje atributy *id\_kategorie, nazev.*

* **Tučně – Entity**
* *Kurzivou – atributy*

**4) CRUD analýza**

CRUD analýza se nachází v příloze

**5) Diskuze smyček -** [**diskuse možných datových nekonzistencí v důsledku smyček ve schématu**](https://dbs.fit.cvut.cz/sw/editor/loop-discussion/)**.**

* V mém databázovém modelu se nachází smyčky. Tabulka OSOBY a ZAVODY jsou spojeny tabulkami ADRESA, ZAVODNIK, PORADATEL, REGISTRACE\_ZAVODU. Přes tyto tabulky se lze dostat od osoby k závodům a opět zpět k osobě. Dále tabulka REGISTRACE\_ZAVODU a ZAVODY jsou spojeny přes VYSLEDKY anebo přes OSOBY a PORADATELE. Opět se lze dostat z tabulky REGISTRACE\_ZAVODU přes VYSLEDKY nebo přes OSOBY a PORADATELE zpět k ZAVODU. Všechny tyto tabulky považuji za důležité.

**6) Integritní omezení (IO)**

IO1: Nový závodník musí zadat všechny povinné údaje(jméno, příjmení, telefonní číslo,…)

IO2: Cena ze závodu může být pouze číslo.

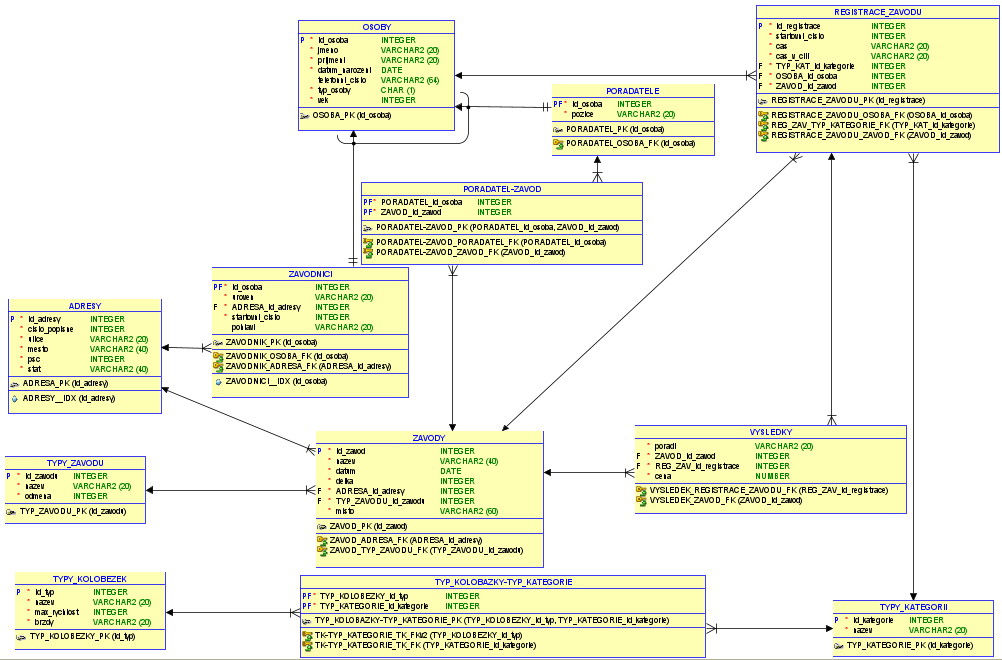
IO3: V tabulce „registrace\_zavodu“ nesmí být dva řádky se stejnou hodnotou v sloupci „startovni\_cislo“.

IO4: Závodník musí být starší než 15 let.

IO5: V tabulce „zavody“ nesmí být 2 stejné názvy.

## ERD

## Relační model dat



Fyzický model dat

Fyzický model se nachází v příloze

Výčet SQL dotazů

/1 HK: D1\*/

SELECT o.id\_osoba, o.jmeno, o.prijmeni, o.vek

FROM osoby o

JOIN poradatele p

ON o.id\_osoba = p.id\_osoba

WHERE o.vek > 30;

/\*2 HK: D2\*/

SELECT z.nazev, o.jmeno, o.prijmeni, o.vek

FROM zavody z

JOIN osoby o

ON z.id\_zavodu = o.zavod\_id

WHERE o.jmeno IS NULL AND z.datum > '2022-01-01';

/\*3 HK: D3 Tento příkaz uděluje uživateli oprávnění ke čtení dat z tabulky "adresy", vkládání nových dat do tabulky, aktualizaci existujících dat v tabulce a mazání dat z tabulky. \*/

GRANT SELECT, INSERT, UPDATE, DELETE ON adresy TO <user>;

/\*4 HK: D4\* Tento příkaz uděluje uživateli oprávnění ke čtení dat ze všech tabulek v databázi, vkládání nových dat do všech tabulek, aktualizaci existujících dat ve všech tabulkách a mazání dat ze všech tabulek. /

GRANT SELECT, INSERT, UPDATE, DELETE ON \* TO <user>;

/\*5 HK: D5\*/

SELECT \*

FROM osoby

JOIN adresy USING (id\_osoba);

/\*6 HK: D6\*/

SELECT \*

FROM osoby

JOIN adresy ON osoby.id\_osoba = adresy.id\_osoba;

/\*7 HK: D7\*/

SELECT \*

FROM osoby

NATURAL JOIN adresy;

/\*8 HK: D8\*/

SELECT \*

FROM osoby

CROSS JOIN adresy;

/\*9 HK: D9\*/

SELECT \*

FROM osoby

LEFT OUTER JOIN adresy ON osoby.id\_osoba = adresy.id\_osoba;

/\*10 HK: D10\*/

SELECT \*

FROM osoby

RIGHT OUTER JOIN adresy ON osoby.id\_osoba = adresy.id\_osoba;

/\*11 HK: D11\*/

SELECT \*

FROM osoby

FULL OUTER JOIN registrace\_závodu ON osoby.id\_osoba = registrace\_závodu.id\_osoba;

/\*12 HK: D12\*/

SELECT jmeno, přijmeni

FROM osoby

WHERE vek > (SELECT AVG(vek)

FROM osoby);

/\*13 HK: D13 Tento dotaz vrátí jméno, příjmení a startovní číslo všech osob, které se zaregistrovaly na určitý závod \*/

SELECT o.jmeno, o.prijmeni, r.startovni\_cislo

FROM osoby o,

(SELECT r.osoba\_id\_osoba, r.startovni\_cislo

FROM registrace\_zavodu r

WHERE r.zavod\_id\_zavod = 1) r

WHERE o.id\_osoba = r.osoba\_id\_osoba;

/\*14 HK: D14 Tento dotaz vybere jméno, příjmení a startovní číslo pro všechny osoby, které se zaregistrovaly na závod po 1. lednu 2022.

Vnořený dotaz v klauzuli WHERE vybírá všechny závody, které se konají po 1. lednu 2022, a používá je jako filtr pro výběr osob.\*/

SELECT o.jmeno, o.prijmeni, r.startovni\_cislo

FROM osoby o

JOIN registrace\_zavodu r

ON o.id\_osoba = r.osoba\_id\_osoba

WHERE r.zavod\_id\_zavod IN (

SELECT z.id\_zavod

FROM zavody z

WHERE z.datum > '2022-01-01'

)

/\*15 HK: D15 Tento dotaz vybere jméno a příjmení pro všechny osoby, které se zaregistrovaly na nějaký závod.

Vnořený dotaz v klauzuli WHERE testuje, zda existuje alespoň jedna registrace pro každou osobu v tabulce osoby.\*/

SELECT o.jmeno, o.prijmeni

FROM osoby o

WHERE EXISTS (

SELECT 1

FROM registrace\_zavodu r

WHERE r.osoba\_id\_osoba = o.id\_osoba

)

/\*15 Vnořený dotaz v klauzuli WHERE testuje, zda neexistuje žádný řádek pro každou osobu v tabulce poradatele.\*/

SELECT o.jmeno, o.prijmeni

FROM osoby o

WHERE NOT EXISTS (

SELECT 1

FROM poradatele p

WHERE p.id\_osoba = o.id\_osoba

)

/\*16 HK: D16 Seznam všech adres z databáze a id osob k nim přiřazených\*/

SELECT adresa, id\_osoba FROM osoby

UNION

SELECT adresa, id\_osoba FROM registrace\_zavodu

ORDER BY ADRESA;

/\*17 HK: D17 Tento dotaz vrátí jméno, příjmení a ID pro všechny osoby s ID 1, 3 nebo 5, ale ne s ID 2 nebo 4.\*/

SELECT id\_osoba, jmeno, prijmeni

FROM osoby

WHERE id\_osoba IN (1, 2, 3, 4, 5)

MINUS

SELECT id\_osoba, jmeno, prijmeni

FROM osoby

WHERE id\_osoba IN (2, 4)

/\*18 HK: D18 Tento dotaz vrátí jméno, příjmení a ID pro všechny osoby s ID 2 nebo 4.\*/

SELECT id\_osoba, jmeno, prijmeni

FROM osoby

WHERE id\_osoba IN (1, 2, 3, 4, 5)

INTERSECT

SELECT id\_osoba, jmeno, prijmeni

FROM osoby

WHERE id\_osoba IN (2, 4, 6, 8)

/\*19 HK: D19 Tento dotaz vrátí velká písmena pro jména osob, délku příjmení, první tři čísla telefonního čísla, pozici, na které se v ulici vyskytuje slovo "náměstí", celé jméno osoby (jméno a příjmení spojené mezerou) a město bez mezer na začátku a konci.\*/

SELECT UPPER(jmeno) AS "VELKE JMENO",

LENGTH(prijmeni) AS "DELKA PRIJMENI",

SUBSTR(telefonni\_cislo, 1, 3) AS "PRVNI TRI CISLA",

INSTR(ulice, 'náměstí') AS "NAMESTI?",

CONCAT(jmeno, ' ', prijmeni) AS "CELÉ JMÉNO",

TRIM(mesto) AS "MESTO BEZ MEZER"

FROM osoby

/\*20 HK: D20 Tento dotaz vrátí průměrný věk osob, součet startovních čísel, nejrychlejší a nejpomalejší čas v cíli, počet registrací, průměrný čas s 2 desetinnými místy, absolutní hodnotu -100 a zbytek po celočíselném dělení 10, 3.\*/

SELECT AVG(vek) AS "PRUMERNY VEK",

SUM(startovni\_cislo) AS "SOUCET STARTOVNICH CISEL",

MIN(cas) AS "NEJRYCHLEJSÍ CAS",

MAX(cas\_v\_cili) AS "NEJPOMALEJSÍ CAS V CILI",

COUNT(\*) AS "POCET REGISTRACI",

ROUND(AVG(cas), 2) AS "PRUMERNY CAS S 2 DESETINNYMI MIESTY",

ABS(-100) AS "ABSOLUTNI HODNOTA",

MOD(10, 3) AS "ZBYTEK PO CELOCISELNEM DELENI"

FROM registrace\_zavodu

/\*21 HK: D21 Sloupec "AKTUALNI CAS" obsahuje aktuální datum a čas pomocí funkce CURRENT\_TIMESTAMP.

Sloupec "PRED ROKEM" obsahuje datum a čas, které jsou posunuté o 1 rok zpět od aktuálního data a času pomocí funkce DATEADD.

Sloupec "POCET LET OD 2022" obsahuje počet let od data "2022-01-01" do aktuálního data pomocí funkce DATEDIFF.

Sloupec "AKTUALNI MESIC" obsahuje aktuální měsíc pomocí funkce EXTRACT a funkce CURRENT\_TIMESTAMP.

Sloupec "AKTUALNI CAS V TEXTU" obsahuje aktuální datum a čas ve formátu "den.měsíc.rok hodiny:minuty:sekundy AM/PM" pomocí funkce DATE\_FORMAT a funkce CURRENT\_TIMESTAMP.\*/

SELECT CURRENT\_DATE AS "DNESNI DATUM",

CURRENT\_TIMESTAMP AS "DNESNI CAS",

DATEADD(YEAR, -1, CURRENT\_TIMESTAMP) AS "PRED ROKEM",

DATEDIFF(YEAR, '2022-01-01', CURRENT\_TIMESTAMP) AS "POCET LET OD 2022",

EXTRACT(MONTH FROM CURRENT\_TIMESTAMP) AS "AKTUALNI MESIC",

DATE\_FORMAT(CURRENT\_TIMESTAMP, '%d. %M %Y %h:%i:%s %p') AS "AKTUALNI CAS V TEXTU"

/\*22 HK: D22 Tento dotaz vrátí dnešní datum, aktuální čas, aktuální datum a čas, datum narození osob o 30 dní později, datum narození osob o 6 měsíců dříve, počet dnů od narození osob, měsíc narození osob a rok narození osob.\*/

SELECT CURDATE() AS "DNESNI DATUM",

CURTIME() AS "AKTUALNI CAS",

NOW() AS "DATUM A CAS NYNÍ",

DATE\_ADD(datum\_narozeni, INTERVAL 30 DAY) AS "DATUM NAROZENI + 30 DNU",

DATE\_SUB(datum\_narozeni, INTERVAL 6 MONTH) AS "DATUM NAROZENI - 6 MESICU",

DATEDIFF(datum\_narozeni, NOW()) AS "POCET DNU OD NAROZENI",

MONTH(datum\_narozeni) AS "MESIC NAROZENI",

YEAR(datum\_narozeni) AS "ROK NAROZENI"

FROM osoby

/\*23 HK: D23 Tento dotaz vrátí průměrný věk pro každý typ osoby, ale zobrazí pouze typy, u kterých je průměrný věk vyšší než 30.\*/

SELECT typ\_osoby, AVG(vek) AS "PRUMERNY VEK"

FROM osoby

GROUP BY typ\_osoby

HAVING AVG(vek) > 30

/\*24 HK: D24 Všechny tři formulace produkují stejný výsledek. Liší se pouze umístěním klauzule HAVING nebo WHERE.\*/

Formulace 1:

SELECT typ\_osoby, mesto, COUNT(\*) AS "POCET OSOB"

FROM osoby

GROUP BY typ\_osoby, mesto

HAVING COUNT(\*) > 2

Formulace 2:

SELECT typ\_osoby, mesto, COUNT(\*) AS "POCET OSOB"

FROM osoby

GROUP BY typ\_osoby, mesto

WHERE COUNT(\*) > 2

Formulace 3:

SELECT typ\_osoby, mesto, COUNT(\*) AS "POCET OSOB"

FROM osoby

WHERE COUNT(\*) > 2

GROUP BY typ\_osoby, mesto

/\*25 HK: D25 Tento dotaz vybere jméno, příjmení a název kategorie pro všechny osoby, které jsou starší než 30 let, a které se zúčastnily alespoň jedné závodu.

Výsledky jsou seskupeny podle jména, příjmení a názvu kategorie a vráceny pouze pro ty kategorie, u kterých se zúčastnilo více než jedna osoba.\*/

SELECT osoby.jmeno, osoby.prijmeni, typy\_kategorii.nazev

FROM osoby

JOIN registrace\_zavodu ON osoby.id\_osoba = registrace\_zavodu.osoba\_id\_osoba

JOIN typy\_kategorii ON registrace\_zavodu.typ\_kat\_id\_kategorie = typy\_kategorii.id\_kategorie

WHERE osoby.vek > 30

GROUP BY osoby.jmeno, osoby.prijmeni, typy\_kategorii.nazev

HAVING

/\*26 HK: D26 Tento dotaz vytváří pohled s názvem "zavodnici", který obsahuje jméno, příjmení, startovní číslo a název kategorie pro všechny osoby, které se zúčastnily alespoň jedné závodu. Pohled vytvoří vnitřní spojení (JOIN) mezi tabulkami osoby, registrace\_zavodu a typy\_kategorii pomocí klauzule ON, která specifikuje, že se mají použít sloupce id\_osoby a osoba\_id\_osoba z tabulek osoby a registrace\_zavodu a sloupce typ\_kat\_id\_kategorie a id\_kategorie z tabulek registrace\_zavodu a typy\_kategorii.\*/

CREATE VIEW zavodnici AS

SELECT osoby.jmeno, osoby.prijmeni, registrace\_zavodu.startovni\_cislo, typy\_kategorii.nazev

FROM osoby

JOIN registrace\_zavodu ON osoby.id\_osoba = registrace\_zavodu.osoba\_id\_osoba

JOIN typy\_kategorii ON registrace\_zavodu.typ\_kat\_id\_kategorie =

/\*27 HK: D27 Tento dotaz vrátí všechny sloupce z pohledu "zavodnici" pro všechny ženy ve věku nad 30 let. Podmínka vek > 30 omezuje výsledky na ženy starší než 30 let a podmínka nazev = 'Ženy' omezuje výsledky pouze na ženy.\*/

SELECT \* FROM zavodnici WHERE vek > 30 AND nazev = 'Ženy'

/\*28 HK: D28 Tento dotaz vloží nové řádky do tabulky osoby z tabulky nove\_osoby. Vložené řádky obsahují hodnoty sloupců jmeno, prijmeni, datum\_narozeni, telefonni\_cislo a vek, které jsou vybrány z tabulky nove\_osoby. Sloupec typ\_osoby je vložen s hodnotou 'O' pro všechny řádky. Vek je vypočítán pomocí funkce YEAR z aktuálního data a data narození z tabulky nove\_osoby.\*/

INSERT INTO osoby (jmeno, prijmeni, datum\_narozeni, telefonni\_cislo, typ\_osoby, vek)

SELECT jmeno, prijmeni, datum\_narozeni, telefonni\_cislo, 'O', (YEAR(CURRENT\_DATE) - YEAR(datum\_narozeni))

FROM nove\_osoby

/\*29 HK: D29 Tento dotaz aktualizuje sloupec vek v tabulce osoby pro všechny řádky, kde id\_osoba je obsažena v tabulce nove\_osoby. Nová hodnota veku je vypočítána pomocí funkce YEAR z aktuálního data a data narození z tabulky osoby.\*/

UPDATE osoby

SET vek = (YEAR(CURRENT\_DATE) - YEAR(datum\_narozeni))

WHERE id\_osoba IN (SELECT id\_osoba FROM nove\_osoby)

/\*30 HK: D30 Tento dotaz odstraní všechny řádky z tabulky osoby, kde id\_osoba je obsažena v tabulce nove\_osoby.\*/

DELETE FROM osoby

WHERE id\_osoba IN (SELECT id\_osoba FROM nove\_osoby)

Závěr

Nemyslím si, že tato databáze je úplně dokonalá, přeci jenom je to moje první zkušenost, a ještě se vše učím. Určitě vím, že bych mohl vybrat lepší typy u několika atributů, ale jinak si myslím, že je databáze docela funkční a mohla by se využít pro nějaké amatérské závody koloběžek. V SQL jsem nikdy předtím nepracoval, takže jsem se toho opravdu hodně naučil. Řekl bych, že to není úplně můj šálek kávy, takže si nejsem jistý, jestli bych v budoucnu dobrovolně něco takového dělal. Doufám, že jsem ze zadaných úkolů nic nevynechal a že práce bude s drobnými chybami v pořádku.

Přílohy

1. DDL skript pro vytvoření tabulek, omezení, sekvencí, indexů, triggerů, apod.
2. DML skript pro naplnění tabulek daty.
3. Soubor dmd včetně složky (Datový model musí souhlasit s DDL skriptem)
4. Skript se všemi dotazy.
5. CRUD analýza
6. Fyzický model