

Semestrální projekt

Vězeňský informační systém

Databázové a informační systémy, 2019/2020

Cvičící: **Ing. Lukáš Zátopek**

Cvičení: **ČT 16:00 – 18:30**

Adam Šárek

SAR0083

Obsah

1	Specifikace zadání	3
2	Datový model	5
2.1	E-R diagram	5
2.2	Lineární zápis	6
2.3	Datový slovník	7
2.4	Seznam integritních omezení	9
3	Stavová analýza	10
4	Funkční analýza	12
4.1	Seznam funkcí	12
4.2	Detailní popis netriviálních funkcí	15
5	Návrh uživatelského rozhraní	23
5.1	Struktura menu	23
5.2	Návrh formulářů	25

1 Specifikace zadání

Proč?

Motivací pro vytvoření systému Věznice je snaha o komplexní, přehledné a podrobné řešení pro uchování dat o chodu věznic v České republice. Jelikož je celá síť věznic již velmi rozsáhlá, nepřehledná a vedení záznamů o jejím fungování v papírové podobě již příliš nákladné, je potřeba vytvořit systém, který veškerou správu zpřehlední a zrychlí.

Kdo?

V systému je nejvýznamnějším uživatelem **ředitel věznice**. **Ředitel** spravuje mimo jiné seznam zaměstnanců a má tedy více pravomocí než **zaměstnanec věznice**. Dále je schopen provádět stejné úkony jako běžný zaměstnanec. **Zaměstnanec** má na starosti správu vězňů, a také správu návštěv. **Návštěvník** může požádat o povolení navštívit některého z vězňů.

Historie změn

Sledujeme přesuny vězňů v rámci jednotlivých cel a také období, kdy byl daný vězeň v určité cele.

Vstupy

Hlavní funkcí systému je správa vězňů, zaměstnanců a návštěvníků věznic.

U **vězňů** sledujeme především jméno, příjmení, pohlaví, datum narození, a také datum začátku a konce trestu.

U **zaměstnanců** (včetně ředitelů) evidujeme jméno, příjmení, pohlaví a datum narození.

U **návštěvníků** evidujeme jméno, příjmení, pohlaví a datum narození. Dále také evidujeme, zda se jedná o aktivního návštěvníka či zda má tento návštěvník zakázáno navštěvovat.

Zaznamenáváme údaje také o **věznicích**, a to jejich typ či také adresu. U **cel** pak navíc zaznamenáváme i jejich kapacitu. Vedeme také seznam jednotlivých **typů věznic**. U **návštěv** pak zaznamenáváme, zda je daná návštěva povolena či nikoliv, a také její datum konání.

Výstupy

Mezi nejvýznamnější výpisy patří výpis seznamu vězňů. Zaměstnanci tedy mohou zobrazit seznam vězňů v rámci dané cely, případně celé věznice. Dále má také zaměstnanec přístup k výpisu jednotlivých cel v rámci dané věznice. Zaměstnanec může rovněž zobrazit seznam návštěv a tuto návštěvu povolit či zakázat. K seznamu zaměstnanců dané věznice má přístup pouze ředitel věznice z důvodu bezpečnosti.

Příklady výstupů

Seznam zaměstnanců

Jméno	Příjmení	Pohlaví	Datum narození	Ředitel	Věznice
Albert	Smith	Muž	7. 9. 1990	ANO	Praha
Robert	Malý	Muž	2. 3. 1991	NE	Brno
Anna	Smetana	Žena	15. 1. 1992	NE	Ostrava
...

Seznam návštěvníků

Jméno	Příjmení	Pohlaví	Datum narození	Aktivní	Zakázaný
Albert	Smith	Muž	7.9.1976	NE	ANO
Robert	Malý	Muž	2.3.1985	ANO	NE
Anna	Smetana	Žena	15.1.1965	NE	NE
...

Jedná se pouze o vzorové příklady, jejichž data se nemusí shodovat reálnými daty v systému.

Funkce

Pro zajištění bezpečnosti je většina funkcí v systému omezena pouze pro zaměstnance a ředitele věznic. Ředitel má poté o něco více pravomocí než zaměstnanec. Má zde však přístup také návštěvník, který přes systém žádá o povolení k návštěvě.

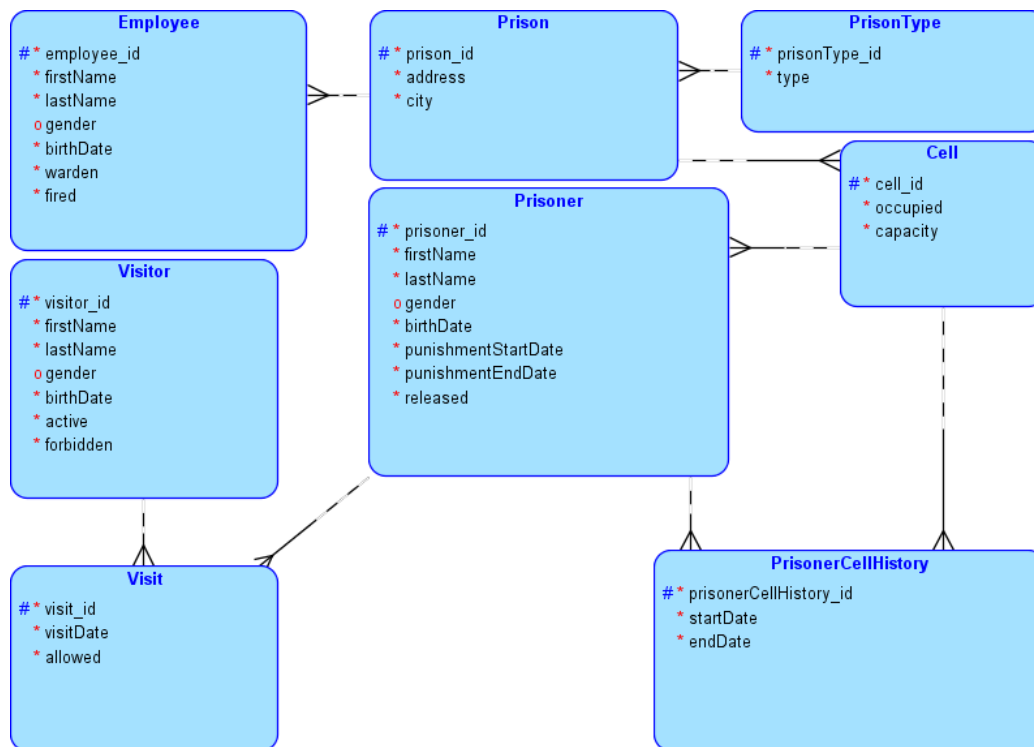
Ředitel věznice má výhradní pravomoc spravovat zaměstnance a rovněž vykonávat činnosti typické pro zaměstnance věznice.

Zaměstnanec má méně pravomocí než ředitel, nicméně má možnost spravovat seznam vězňů a návštěv. Dále také spravuje jednotlivé cely v rámci věznice, ve které působí.

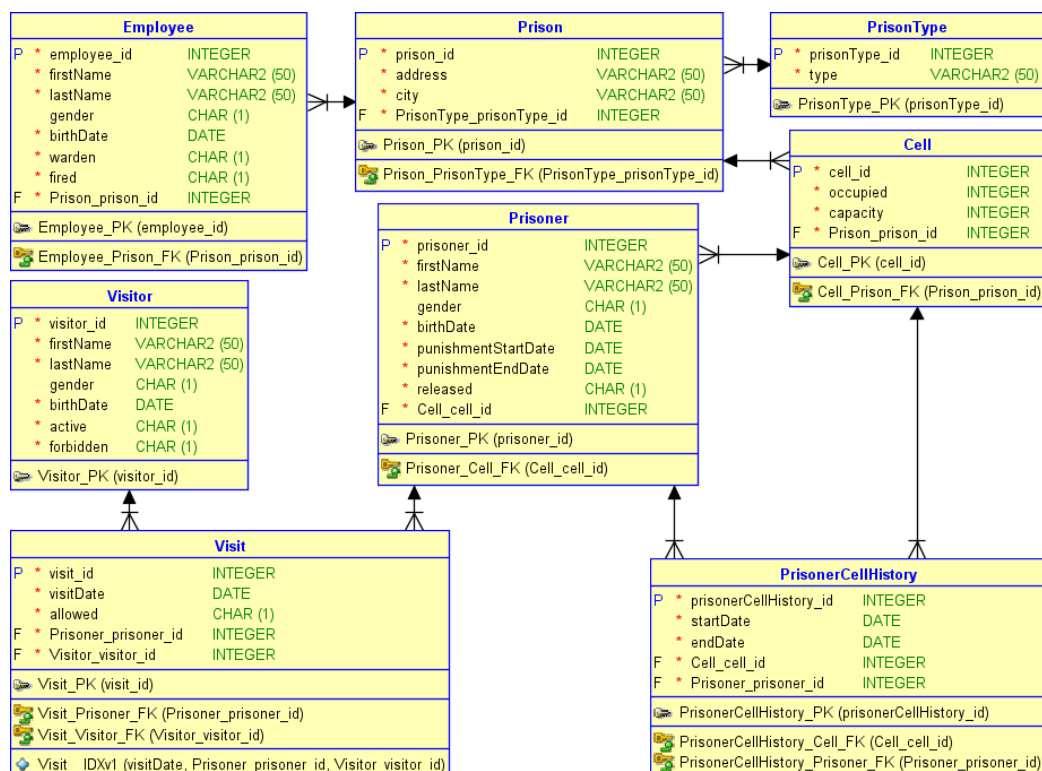
Návštěvník má velmi omezený přístup a může tedy pouze požádat o povolení k návštěvě.

2 Datový model

2.1 Logický model



2.2 E-R diagram



2.3 Lineární zápis

Legenda: **Tabulka**, primární klíč, *cizí klíč*, atribut

Cell(cell_id, occupied, capacity, *Prison_prison_id*)

Employee(employee_id, firstName, lastName, gender, birthDate, warden, fired, *Prison_prison_id*)

Prison(prison_id, address, city, *PrisonType_prisonType_id*)

Prisoner(prisoner_id, firstName, lastName, gender, birthDate, punishmentStartDate, punishmentEndDate, released, *Cell_cell_id*)

PrisonerCellHistory(prisonerCellHistory_id, startDate, endDate, *Cell_cell_id*, *Prisoner_prisoner_id*)

PrisonType(prisonType_id, type)

Visit(visit_id, visitDate, allowed, *Prisoner_prisoner_id*, *Visitor_visitor_id*)

Visitor(visitor_id, firstName, lastName, gender, birthDate, active, forbidden)

Lineární zápis typů vztahů

Cell_Prison(Cell, Prison)	N:1
Cell_Prisoner(Cell, Prisoner)	1:N
Cell_PrisonerCellHistory (Cell, PrisonerCellHistory)	1:N
Employee_Prison(Employee, Prison)	N:1
Prison_PrisonType(Prison, PrisonType)	N:1
Prisoner_PrisonerCellHistory(Prisoner, PrisonerCellHistory)	1:N
Prisoner_Visit(Prisoner, Visit)	1:N
Visit_Visitor(Visit, Visitor)	N:1

2.4 Datový slovník

Tabulka **Cell**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
cell_id	Integer		Primární	Ne	
occupied	Integer			Ne	Počet obsazených míst v cele
capacity	Integer			Ne	Kapacita cely
Prison_prison_id	Integer		Cizí	Ne	ID věznice, ve které se cela nachází

Tabulka **Employee**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
employee_id	Integer		Primární	Ne	
firstName	Varchar	50		Ne	Jméno zaměstnance
lastName	Varchar	50		Ne	Příjmení zaměstnance
gender	Char	1		Ano	Pohlaví zaměstnance
birthDate	Date			Ne	Datum narození zaměstnance
warden	Char	1		Ne	Je zaměstnanec ředitel?
fired	Char	1		Ne	Je zaměstnanec vyhozen?
Prison_prison_id	Integer		Cizí	Ne	ID věznice, kde zaměstnanec pracuje

Tabulka **Prison**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
prison_id	Integer		Primární	Ne	
address	Varchar	50		Ne	Ulice, kde se nachází
city	Varchar	50		Ne	Město, kde se nachází
PrisonType_prisonType_id	Integer		Cizí	Ne	

Tabulka **Prisoner**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
prisoner_id	Integer		Primární	Ne	
firstName	Varchar	50		Ne	Jméno vězně
lastName	Varchar	50		Ne	Příjmení vězně
gender	Char	1		Ano	Pohlaví vězně
birthDate	Date			Ne	Datum narození vězně
punishmentStartDate	Date			Ne	Datum začátku trestu
punishmentEndDate	Date			Ne	Datum konce trestu
released	Char	1		Ne	Je již vězeň propuštěn?
Cell_cell_id	Integer		Cizí	Ne	ID cely, kde se vězeň nachází

Tabulka **PrisonerCellHistory**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
prisonerCellHistory_id	Integer		Primární	Ne	
startDate	Date			Ne	Počáteční den vězně v cele
endDate	Date			Ne	Konečný den vězně v cele
Cell_cell_id	Integer		Cizí	Ne	ID cely
Prisoner_prisoner_id	Integer		Cizí	Ne	ID vězně

Tabulka **PrisonType**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
prisonType_id	Integer		Primární	Ne	
type	Varchar	50		Ne	Typ věznice

Tabulka **Visit**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
visit_id	Integer		Primární	Ne	
visitDate	Date			Ne	Datum návštěvy
allowed	Char	1		Ne	Je návštěva povolena?
Prisoner_prisoner_id	Integer		Cizí	Ne	ID vězně
Visitor_visitor_id	Integer		Cizí	Ne	ID návštěvníka

Tabulka **Visitor**

Atribut	Dat. typ	Délka	Klíč	Null	Význam
visitor_id	Integer		Primární	Ne	
firstName	Varchar	50		Ne	Jméno návštěvníka
lastName	Varchar	50		Ne	Příjmení návštěvníka
gender	Char	1		Ano	Pohlaví návštěvníka
birthDate	Date			Ne	Datum narození návštěvníka
active	Char	1		Ne	Je návštěvník aktivní?
forbidden	Char	1		Ne	Je návštěvník zakázaný?

2.5 Seznam integritních omezení

- Atribut `occupied` v tabulce `Cell` musí být větší nebo rovno 0 a zároveň menší nebo rovno atributu `capacity`.
- Atribut `capacity` v tabulce `Cell` musí být větší nebo rovno 1.
- Atribut `warden` v tabulce `Employee` musí obsahovat znak 0 nebo 1.
- Atribut `fired` v tabulce `Employee` musí obsahovat znak 0 nebo 1.
- Atribut `punishmentStartDate` v tabulce `Prisoner` musí být menší než atribut `punishmentEndDate`.
- Atribut `released` v tabulce `Prisoner` musí obsahovat znak 0 nebo 1.
- Atribut `startDate` v tabulce `PrisonerCellHistory` musí být menší než atribut `endDate`.
- Atribut `allowed` v tabulce `Visit` musí obsahovat znak 0, 1 nebo 2.
- Atribut `active` v tabulce `Visitor` musí obsahovat znak 0 nebo 1.
- Atribut `forbidden` v tabulce `Visitor` musí obsahovat znak 0 nebo 1.

3 Stavová analýza

Definujeme tyto stavy cely:

- **Je plná** – Cella, jejíž počet obsazených míst je stejný jako kapacita.
`Cell.occupied = Cell.capacity`
- **Není plná** – Cella, jejíž počet obsazených míst je menší nebo rovný kapacitě.
`Cell.occupied <= Cell.capacity`

Definujeme tyto stavy zaměstnance:

- **Je ředitelem** – Zaměstnanec, který je zároveň ředitelem.
`Employee.warden = 1`
- **Není ředitelem** – Zaměstnanec, který není ředitelem.
`Employee.warden = 0`
- **Je vyhozen** – Zaměstnanec, který již ve věznici nepracuje.
`Employee.fired = 1`
- **Není vyhozen** – Zaměstnanec, který stále ještě ve věznici pracuje.
`Employee.fired = 0`

Definujeme tyto stavy vězně:

- **Je propuštěn** – Vězeň, který již byl propuštěn.
`Prisoner.released = 1`
- **Není propuštěn** – Vězeň, který ještě nebyl propuštěn.
`Prisoner.released = 0`

Definujeme tyto stavy návštěvy:

- **Nerozhodnutá** – Návštěva, která zatím nemá určeno, zda bude povolena či nikoliv.
`Visit.allowed = 2`
- **Povolená** – Návštěva, kterou zaměstnanec povolil.
`Visit.allowed = 1`
- **Nepovolená** – Návštěva, kterou zaměstnanec nepovolil.
`Visit.allowed = 0`

Definujeme tyto stavy návštěvníka

- **Aktivní** – Návštěvník, který má naplánovanou alespoň jednu návštěvu v rozmezí 30 dnů před a po aktuálním datu.
Visitor.active = 1
- **Neaktivní** – Návštěvník, který nemá naplánovanou ani jednu návštěvu v rozmezí 30 dnů před a po aktuálním datu.
Visitor.active = 0

\$visitor_id – reprezentuje aktuálního návštěvníka

\$currentDate – reprezentuje aktuální datum

\$Visit_count – reprezentuje počet naplánovaných návštěv

```
SELECT COUNT(*) INTO $Visit_count FROM Visit WHERE Visitor_visitor_id =  
$visitor_id AND visitDate BETWEEN $currentDate - 30 AND $currentDate + 30;
```

```
IF $Visit_count > 0 THEN  
    UPDATE Visitor SET active = 1 WHERE visitor_id = $visitor_id;  
ELSE  
    UPDATE Visitor SET active = 0 WHERE visitor_id = $visitor_id;  
END IF;
```

- **Zakázaný (na černé listině)** – Návštěvník, který byl umístěn na černou listinu a nemůže již navštěvovat žádného z vězňů.
Visitor.forbidden = 1
- **Povolený** – Návštěvník, který není na černé listině.
Visitor.forbidden = 0

4 Funkční analýza

4.1 Seznam funkcí

1. Evidence cel

Tabulka: Cell

1.1 Přidání cely

Zodpovědnost: Admin, Ředitel (své věznice)

1.2 Seznam cel

Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

1.3 Aktualizace cely

Zodpovědnost: Admin, Ředitel (své věznice)

1.4 Detail cely

Zodpovědnost: Admin, Ředitel (své věznice)

2. Evidence zaměstnanců

Tabulka: Employee

2.1 Přidání zaměstnance

Zodpovědnost: Admin, Ředitel (své věznice)

2.2 Seznam zaměstnanců

Zodpovědnost: Admin, Ředitel (své věznice)

2.3 Aktualizace zaměstnance

Zodpovědnost: Admin, Ředitel (své věznice)

2.4 Propustit zaměstnance

Zodpovědnost: Admin, Ředitel (své věznice)

2.5 Detail zaměstnance

Zodpovědnost: Admin, Ředitel (své věznice)

3. Evidence vězňů

Tabulka: Prisoner, Cell, PrisonerCellHistory, Visit

- 3.1 **Přidání vězně** – zjistí, zda je možné daného vězně přidat do dané cely, a pokud ano, tak jej přidá a zvýší počet obsazených míst cely o jedno
Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)
- 3.2 Seznam vězňů
Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice), Návštěvník
- 3.3 **Aktualizace vězně** – zjistí, zda je možné daného vězně (pokud ještě nebyl propuštěn) přesunout do dané cely, a pokud ano, tak jej přesune a zvýší počet obsazených míst nové cely o jedno a zároveň sníží počet míst staré cely o jedno a přidá záznam do historie obsazení cel
Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)
- 3.4 **Propustit vězně** – vězeň bude mít nově aktivní vlastnost propuštěn a zároveň se sníží počet obsazených míst cely ve které byl naposledy o jedno a přidá záznam do historie obsazení cel a všechny naplánované návštěvy tohoto vězně se změní na nepovolené
Zodpovědnost: Admin, Ředitel (své věznice)
- 3.5 Detail vězně
Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

4. Evidence historie obsazení cel

Tabulka: PrisonerCellHistory, Prisoner

- 4.1 **Přidání záznamu** – přidá záznam do historie obsazení cel
Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)
- 4.2 Seznam záznamů
Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

5. Evidence návštěv

Tabulka: Visit, Prisoner, Visitor

- 5.1 **Přidání návštěvy** – zjistí, zda je možné danou návštěvu přidat na základě toho, zda je její návštěvník povolen a navštívený vězeň ještě nebyl propuštěn, a pokud tyto podmínky platí, tak se daná návštěva přidá a aktualizuje se aktivita daného návštěvníka

Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice), Návštěvník

- 5.2 Seznam návštěv

Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

- 5.3 Aktualizace návštěvy

Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

- 5.4 Detail návštěvy

Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

6. Evidence návštěvníků

Tabulka: Visitor, Visit

- 6.1 Přidání návštěvníka

Zodpovědnost: Admin, Ředitel, Zaměstnanec

- 6.2 Seznam návštěvníků

Zodpovědnost: Admin, Ředitel, Zaměstnanec

- 6.3 Aktualizace návštěvníka

Zodpovědnost: Admin, Ředitel, Zaměstnanec

- 6.4 **Aktualizace aktivity návštěvníka** – na základě datumu poslední naplánované návštěvy (v rámci 30 dnů od aktuálního data do minulosti a budoucnosti) dojde k zaznamenání aktivity/neaktivity návštěvníka

Zodpovědnost: Admin, Ředitel, Zaměstnanec

- 6.5 **Zakázání návštěvníka** – návštěvník bude mít nově aktivní vlastnost zakázán a zároveň se všechny naplánované návštěvy tohoto návštěvníka změny na nepovolené

Zodpovědnost: Admin, Ředitel, Zaměstnanec

- 6.6 Detail návštěvníka

Zodpovědnost: Admin, Ředitel (své věznice), Zaměstnanec (své věznice)

4.2 Detailní popis netriviálních funkcí

Funkce 3.1 Přidání vězně

Vstupy:

\$firstName – reprezentuje jméno přidávaného vězně

\$lastName – reprezentuje příjmení přidávaného vězně

\$gender – reprezentuje pohlaví přidávaného vězně

\$birthDate – reprezentuje datum narození přidávaného vězně

\$punishmentStartDate – reprezentuje datum začátku trestu přidávaného vězně

\$punishmentEndDate – reprezentuje datum konce trestu přidávaného vězně

\$Cell_cell_id – reprezentuje celu přidávaného vězně

Popis funkce:

Funkce umožňuje přidání vězně. Funkce zjistí, zda je možné daného vězně přidat do dané cely, a pokud ano, tak jej přidá a zvýší počet obsazených míst cely o jedno. Pokud byla cela před přidáním již plná, tak systém vyvolá chybu a daného vězně neumožní přidat. Tato funkce bude realizována jako transakce.

- 1) Do *\$Cell_occupied* načteme počet obsazených míst a do *\$Cell_capacity* kapacitu dané cely.

```
SELECT occupied, capacity FROM Cell WHERE cell_id = $Cell_cell_id
```

- 2) Zkontrolujeme, jestli počet obsazených míst (+1 za přidávaného vězně) přesahuje kapacitu cely.

```
IF (($Cell_occupied + 1) > $Cell_capacity) THEN
```

- a) Pokud je podmínka splněna, tak kapacita byla překročena a dojde k vyvolání výjimky s chybovou zprávou: „Cell *\$Cell_cell_id* is already full. You cannot add more prisoners there!“

- 3) Pokud podmínka není splněna, tak kapacita nebyla překročena. Do *\$prisoner_id* načteme nejvyšší id vězně + 1.

```
SELECT (MAX(prisoner_id) + 1) FROM Prisoner
```

- 4) Přidáme vězně.

```
INSERT INTO Prisoner VALUES($prisoner_id, $firstName, $lastName,  
$gender, $birthDate, $punishmentStartDate, $punishmentEndDate, 0,  
$Cell_cell_id)
```

5) Navýšíme počet obsazených míst.

```
UPDATE Cell SET occupied = occupied + 1 WHERE cell_id = $Cell_cell_id
```

Funkce 3.3 Aktualizace vězně

Vstupy:

\$prisoner_id – reprezentuje aktualizovaného vězně
\$firstName – reprezentuje jméno aktualizovaného vězně
\$lastName – reprezentuje příjmení aktualizovaného vězně
\$gender – reprezentuje pohlaví aktualizovaného vězně
\$birthDate – reprezentuje datum narození aktualizovaného vězně
\$punishmentStartDate – reprezentuje datum začátku trestu aktualizovaného vězně
\$punishmentEndDate – reprezentuje datum konce trestu aktualizovaného vězně
\$NewCell_cell_id – reprezentuje celu aktualizovaného vězně

Popis funkce:

Funkce umožňuje aktualizaci vězně. Funkce zjistí, zda je možné daného vězně (pokud ještě nebyl propuštěn) přesunout do dané cely, a pokud ano, tak jej přesune a zvýší počet obsazených míst nové cely o jedno a zároveň sníží počet míst staré cely o jedno a přidá záznam do historie obsazení cel. Pokud byla cela před přesunutím již plná, tak systém vyvolá chybu a daného vězně neumožní přesunout. Pokud nedošlo ke změně cely nebo byl vězeň již propuštěn, tak dojde pouze k aktualizaci údajů bez změny cely. Tato funkce bude realizována jako transakce.

1) Do *\$released* načteme informaci, zda byl vězeň propuštěn a do *\$OldCell_cell_id* aktuální celu vězně před aktualizací.

```
SELECT released, Cell_cell_id FROM Prisoner WHERE prisoner_id =  
$prisoner_id
```

2) Zkontrolujeme, jestli není nová cela stejná jako stará cela nebo zda vězeň nebyl již propuštěn.

```
IF ($OldCell_cell_id = $NewCell_cell_id OR $released = 1) THEN
```

a) Pokud je podmínka splněna, tak je nová cela stejná jako stará nebo byl vězeň již propuštěn. Dojde k aktualizaci zbylých údajů vězně.


```

UPDATE Prisoner
SET firstName = $firstName, lastName = $lastName, gender = $gender,
  birthDate = $birthDate, punishmentStartDate = $punishmentStartDate,
  punishmentEndDate = $punishmentEndDate
WHERE prisoner_id = $prisoner_id

```

- 3) Pokud podmínka není splněna, tak je nová cela odlišná od staré cely a zároveň vězeň nebyl propuštěn. Do *\$Cell_occupied* načteme počet obsazených míst a do *\$Cell_capacity* kapacitu nové cely.

```

SELECT occupied, capacity FROM Cell WHERE cell_id = $NewCell_cell_id

```

- 4) Zkontrolujeme, jestli počet obsazených míst (+1 za aktualizovaného vězně) přesahuje kapacitu nové cely.

```

IF (($Cell_occupied + 1) > $Cell_capacity) THEN

```

- a) Pokud je podmínka splněna, tak kapacita byla překročena a dojde k vyvolání výjimky s chybovou zprávou: „Cell *\$NewCell_cell_id* is already full. You cannot add more prisoners there!“

- 5) Pokud podmínka není splněna, tak kapacita nebyla překročena. Přidáme záznam do historie obsazení cel (viz. Funkce 5.1 Přidání záznamu)

```

insertPrisonerCellHistory($prisoner_id, $OldCell_cell_id)

```

- 6) Snížíme počet obsazených míst staré cely a zvýšíme počet obsazených míst nové cely.

```

UPDATE Cell SET occupied =
CASE
  WHEN cell_id = $OldCell_cell_id THEN occupied - 1
  WHEN cell_id = $NewCell_cell_id THEN occupied + 1
  ELSE occupied
END;

```

- 7) Aktualizujeme vězně

```

UPDATE Prisoner
SET firstName = $firstName, lastName = $lastName, gender = $gender,
  birthDate = $birthDate, punishmentStartDate = $punishmentStartDate,
  punishmentEndDate = $punishmentEndDate, Cell_cell_id = $NewCell_cell_id
WHERE prisoner_id = $prisoner_id

```

Funkce 3.4 Propustit vězně

Vstupy:

\$prisoner_id – reprezentuje propouštěného vězně

Popis funkce:

Funkce umožňuje propuštění vězně. Funkce zajistí, že vězeň bude mít aktivní vlastnost propuštěn a zároveň se sníží počet obsazených míst cely ve které byl naposledy o jedno a přidá záznam do historie obsazení cel a všechny naplánované návštěvy tohoto vězně se změň na nepovolené. Tato funkce bude realizována jako transakce.

- 1) Do *\$Cell_cell_id* načteme aktuální celu vězně před propuštěním.

```
SELECT Cell_cell_id FROM Prisoner WHERE prisoner_id = $prisoner_id
```

- 2) Přidáme záznam do historie obsazení cel (viz. Funkce 5.1 Přidání záznamu)

```
insertPrisonerCellHistory($prisoner_id, $Cell_cell_id)
```

- 3) Snížíme počet obsazených míst aktuální cely.

```
UPDATE Cell SET occupied = occupied - 1 WHERE cell_id = $Cell_cell_id
```

- 4) Nepovolíme všechny naplánované návštěvy tohoto vězně.

\$currentDate – reprezentuje aktuální datum

```
UPDATE Visit SET allowed = 0 WHERE Prisoner_prisoner_id = $prisoner_id  
AND visitDate > $currentDate
```

- 5) Propustíme vězně.

```
UPDATE Prisoner SET released = 1 WHERE prisoner_id = $prisoner_id
```

Funkce 4.1 Přidání záznamu

Vstupy:

\$prisoner_id – reprezentuje zaznamenávaného vězně

\$Cell_cell_id – reprezentuje celu zaznamenávaného vězně

Popis funkce:

Funkce umožňuje přidání záznamu do historie obsazení cel.

- 1) Do *\$prisonerCellHistory_id* načteme nejvyšší id záznamu + 1.

```
SELECT (MAX(prisonerCellHistory_id) + 1) FROM PrisonerCellHistory
```

- 2) Do *\$startDate* načteme datum konce pobytu v minulé cele (pokud takový záznam existuje, jinak načteme datum začátku trestu daného vězně).

```
SELECT  
CASE  
  WHEN COUNT(endDate) > 0 THEN MAX(endDate)  
  ELSE punishmentStartDate END  
FROM Prisoner  
  LEFT JOIN PrisonerCellHistory ON Prisoner_prisoner_id = prisoner_id  
WHERE prisoner_id = $prisoner_id  
GROUP BY punishmentStartDate
```

- 3) Přidáme záznam do historie obsazení cel

\$prisonerCellHistory_id – reprezentuje vkládaný záznam

\$startDate – reprezentuje počáteční datum pobytu vězně v dané cele

\$currentDate – reprezentuje aktuální datum

```
INSERT INTO PrisonerCellHistory VALUES($prisonerCellHistory_id,  
$startDate, $currentDate, $Cell_cell_id, $prisoner_id)
```

Funkce 5.1 Přidání návštěvy

Vstupy:

\$visitDate – reprezentuje datum návštěvy

\$Prisoner_prisoner_id – reprezentuje navštíveného vězně

\$Visitor_visitor_id – reprezentuje návštěvníka

Popis funkce:

Funkce umožňuje přidání návštěvy. Funkce zjistí, zda je možné danou návštěvu přidat na základě toho, zda je její návštěvník povolen a navštívený vězeň ještě nebyl propuštěn, a pokud tyto podmínky platí, tak se daná návštěva přidá a aktualizuje se aktivita daného návštěvníka. Tato funkce bude realizována jako transakce.

- 1) Do *\$Prisoner_released* načteme informaci, zda byl vězeň propuštěn.

```
SELECT released FROM Prisoner WHERE prisoner_id = $prisoner_id
```

- 2) Do *\$visit_id* načteme nejvyšší id návštěvy + 1.

```
SELECT (MAX(visit_id) + 1) FROM Visit
```

- 3) Do *\$Visitor_forbidden* načteme informaci, zda je návštěvník zakázán.

```
SELECT forbidden FROM Visitor WHERE visitor_id = $Visitor_visitor_id
```

- 4) Zkontrolujeme, jestli vězeň nebyl propuštěn a zároveň návštěvník není zakázán.

```
IF ($Prisoner_released = 0 AND $Visitor_forbidden_id = 0) THEN
```

- a) Pokud podmínka není splněna, tak funkce končí.

- 5) Pokud je podmínka splněna, tak je vězeň stále ve výkonu trestu a návštěvník není zakázán, takže přidáme návštěvu.

```
INSERT INTO Visit VALUES($visit_id, $visitDate, $Prisoner_prisoner_id,  
$Visitor_visitor_id)
```

- 6) Aktualizujeme stav aktivity návštěvníka (viz. Funkce 8.5 Aktualizace aktivity návštěvníka)

```
updateVisitorActivity($Visitor_visitor_id)
```

Funkce 6.4 Aktualizace aktivity návštěvníka

Vstupy:

\$visitor_id – reprezentuje aktualizovaného návštěvníka

Popis funkce:

Funkce umožňuje aktualizaci aktivity návštěvníka. Na základě datumu poslední naplánované návštěvy (v rámci 30 dnů od aktuálního data do minulosti a budoucnosti) dojde k zaznamenání aktivity/neaktivity návštěvníka.

- 1) Do *\$Visit_count* načteme počet naplánovaných návštěv v rozmezí 30 dnů do minulosti a 30 dnů do budoucnosti.

\$currentDate – reprezentuje aktuální datum

```
SELECT COUNT(*)  
FROM Visit  
WHERE Visitor_visitor_id = $visitor_id AND visitDate BETWEEN  
$currentDate - 30 AND $currentDate + 30
```

- 2) Zkontrolujeme, jestli platí *\$Visit_count* > 0 a pokud ano, tak nastavíme návštěvníka jako aktivního. V opačném případě jej nastavíme jako neaktivního.

```
UPDATE Visitor SET active =  
CASE  
    WHEN $Visit_count > 0 THEN 1  
    ELSE 0 END  
WHERE visitor_id = $visitor_id
```

Funkce 6.5 Zakázání návštěvníka

Vstupy:

\$visitor_id – reprezentuje zakázaného návštěvníka

Popis funkce:

Funkce umožňuje zakázání návštěvníka. Návštěvník bude mít nově aktivní vlastnost zakázán a zároveň se všechny naplánované návštěvy tohoto návštěvníka změni na nepovolené. Tato funkce bude realizována jako transakce.

1) Všechny naplánované návštěvy se změni na nepovolené.

\$currentDate – reprezentuje aktuální datum

```
UPDATE Visit
SET allowed = 0
WHERE Visitor_visitor_id = $visitor_id AND visitDate > $currentDate
```

2) Zakázání návštěvníka.

```
UPDATE Visitor
SET forbidden = 1
WHERE visitor_id = $visitor_id
```

5 Návrh uživatelského rozhraní

5.1 Struktura menu

1. Správa cel

Zodpovědnost: Ředitel (své věznice)

- Přidat celu – akce: 1.1 Přidání cely
- Seznam cel – akce 1.2 Seznam cel
- Upravit celu – akce 1.3 Aktualizace cely
- Přidat vězně – akce: 3.1 Přidání vězně
- Seznam vězňů – akce 3.2 Seznam vězňů
- Upravit vězně – akce 3.3 Aktualizace vězně
- Propustit vězně – akce 3.4 Propustit vězně

2. Správa zaměstnanců

Zodpovědnost: Ředitel (své věznice)

- Přidat zaměstnance – akce: 2.1 Přidání zaměstnance
- Seznam zaměstnanců – akce 2.2 Seznam zaměstnanců
- Upravit zaměstnance – akce 2.3 Aktualizace zaměstnance
- Propustit zaměstnance – akce 2.4 Propustit zaměstnance

3. Správa vězňů

Zodpovědnost: Ředitel (své věznice), Zaměstnanec (své věznice)

- Přidat vězně – akce: 3.1 Přidání vězně
- Seznam vězňů – akce 3.2 Seznam vězňů
- Upravit vězně – akce 3.3 Aktualizace vězně
- Propustit vězně – akce 3.4 Propustit vězně
- Přidat návštěvu – akce: 5.1 Přidání návštěvy
- Seznam návštěv – akce 5.2 Seznam návštěv
- Upravit návštěvu – akce 5.3 Aktualizace návštěvy

4. Správa návštěvníků

Zodpovědnost: Ředitel (své věznice), Zaměstnanec (své věznice)

- Přidat návštěvníka – akce: 6.1 Přidání návštěvníka
- Seznam návštěvníků – akce 6.2 Seznam návštěvníků
- Upravit návštěvníka – akce 6.3 Aktualizace návštěvníka
- Zakázat návštěvníka – akce 6.5 Zakázání návštěvníka
- Přidat návštěvu – akce: 5.1 Přidání návštěvy
- Seznam návštěv – akce 5.2 Seznam návštěv
- Upravit návštěvu – akce 5.3 Aktualizace návštěvy

5.2 Návrh formulářů

Návrh funkce 4.3 Aktualizace vězně

Upravení vězně

Upravení vězně

Jméno

Příjmení

Pohlaví

☐ Muž ☐ Žena

Datum narození

Začátek trestu

Konec trestu

Upravit vězně

Propustit vězně

Detail vězně

Jméno

Mirek

Příjmení

Crhan

Pohlaví

Muž

Datum narození

5.10.1998

Začátek trestu

1.1.2020

Konec trestu

3.4.2034

Funkce 3.3 Aktualizace vězně

Funkce 3.4 Propustit vězně

Funkce 3.5 Detail vězně

Seznam návštěv

	Datum návštěvy	Povolena	Jméno návštěvníka	Pohlaví	Věk
1	2.2.2020	ANO	Milan Onderka	Muž	50
2	4.2.2020	ANO	Ludmila Smutna	Žena	38

Přidat návštěvu

Upravit návštěvu

Funkce 5.1 Přidání návštěvy

Funkce 5.3 Aktualizace návštěvy (pouze pro vybranou návštěvu)

Funkce 5.2 Seznam návštěv (vypíše pouze návštěvy vybraného vězně)

Návrh menu 1. Správa cel

