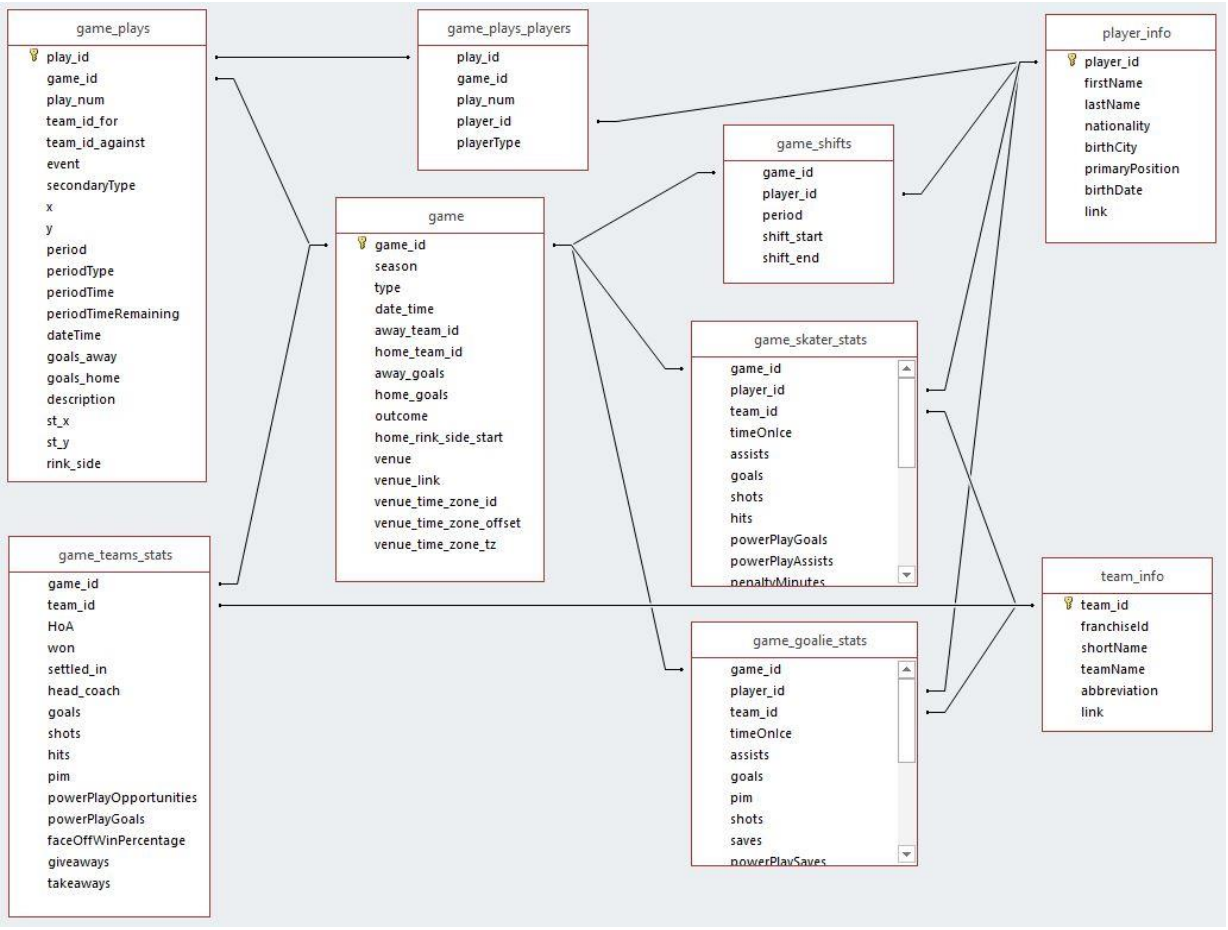
Semestrální práce – NHL data

Neo4j, MongoDB

NI-PDB 2022/2023, Tomáš Kalabis

Popis dat

Vybral jsem si jako téma práce data z hokejové ligy NHL. Data jsou dostupné na *Kaggle* pod tímto odkazem. Popisují základní údaje o hráčích, jednotlivé zápasy a statistiky tohoto zápasu, statistiky jednotlivých hráčů v jednotlivých zápasech. Data dále popisují údaje rozhodčích, údaje o jednotlivých událostech v zápase podrobněji. Ty jsem však nezahrnoval, kvůli přílišné komplikovanosti databáze. Dále jsem zahrnul data zápasů pouze ze sezóny 2019/2020 kvůli velikosti dat a dlouhému výpočetnímu času create skriptu (po zmenšení ˚ 1,312 zápasů). Vyjmul jsem také některé nezajímavé atributy tabulek jako časová pásma míst konání zápasů. Původní konceptuální model vypadá zhruba následovně. V upravené verzi nepočítám s tabulkami *game_plays*, *game_plays_players* a *game_shifts*.



MongoDB

Nejdříve bylo nutné vytvořit create script, který byl vytvořen v Pythonu. Jednotlivé datové kolekce jsou uložené v JSON formátu - konkrétně kolekce *game* a *player*. Kolekce player je přepsaná tabulka, *player_info*. Kolekce *game* je tabulka *game* s přidanými údaji z *team_info*, *game_team_stats* a statistiky jednotlivých hráčů v zápase (tabulky *game_skater_stats*, *game_goalie_stats*). Ukázky vkládaných dat viz níže.

Kolekce Game


```
{"game_id":2019020001,"season":20192020,"type":"R","date_time_GMT":"
2019-10-
02T23:00:00Z","home_team":"Canucks","away_team":"Predators","outcome
":"home win
REG","away_goals":3,"home_goals":5,"home_players":"{8477939:
{'timeOnIce': 1025.0, 'assists': 1.0, 'goals': 0.0, 'shots': 3.0,
'hits': 0.0, 'powerPlayGoals': 0.0, 'powerPlayAssists': 0.0,
'penaltyMinutes': 0.0, 'faceOffWins': 0.0, 'faceoffTaken': 4.0,
'takeaways': 1.0, 'giveaways': 1.0, 'shortHandedGoals': 0.0,
'shortHandedAssists': 0.0, 'blocked': 0.0, 'plusMinus': 0.0,
'evenTimeOnIce': 804.0, 'shortHandedTimeOnIce': 0.0,
'powerPlayTimeOnIce': 221.0},

...

 8475883: {'timeOnIce': 3591, 'assists': 0, 'goals': 0, 'pim': 0,
'shots': 26, 'saves': 23, 'powerPlaySaves': 0, 'shortHandedSaves':
2, 'evenSaves': 21, 'shortHandedShotsAgainst': 2,
'evenShotsAgainst': 24, 'powerPlayShotsAgainst': 0, 'decision': 'W',
'savePercentage': 88.4615384615385, 'powerPlaySavePercentage': nan,
'evenStrengthSavePercentage':
87.5}}","away_players":"{...}","team_stats":"{'home': {'head_coach':
'Mike Babcock', 'shots': 42.0, 'pim': 6.0, 'hits': 17.0}, 'away':
{'head_coach': 'D.J. Smith', 'shots': 26.0, 'pim': 10.0, 'hits':
44.0}}"}
```

```
db.game.insertOne({
"game_id":2019020001,
"season":20192020,
"type":"R",
"date_time_GMT":"2019-10-02T23:00:00Z",
"home_team":"Canucks",
"away_team":"Predators",
"outcome":"home win REG",
"away_goals":3,
"home_goals":5,
"home_players": {8477939: {
                    'timeOnIce': 1025.0,
                    'assists': 1.0,
                    'goals': 0.0,
                    'shots': 3.0,
                    'hits': 0.0,
                    'powerPlayGoals': 0.0,
                    'powerPlayAssists': 0.0,
                    'penaltyMinutes': 0.0,
                    'faceOffWins': 0.0,
                    'faceoffTaken': 4.0,
                    'takeaways': 1.0,
                    'giveaways': 1.0,
                    'shortHandedGoals': 0.0,
                    'shortHandedAssists': 0.0,
                    'blocked': 0.0,
                    'plusMinus': 0.0,
                    'evenTimeOnIce': 804.0,
                    'shortHandedTimeOnIce': 0.0,
                    'powerPlayTimeOnIce': 221.0},
                    ...,
                8475883: {
                    'timeOnIce': 3591,
                    'assists': 0,
                    'goals': 0,
                    'pim': 0,
                    'shots': 26,
                    'saves': 23,
```

Kolekce Player

{"player_id":8466148,"firstName":"Marian","lastName":"Hossa","nationality":"SVK","birthCity":"Star\u00e1 Lubovna","primaryPosition":"RW","birthDate":"1979-01-12 00:00:00","birthStateProvince":null,"height":"6' 1\"","height_cm":185.42,"weight":207.0,"shootsCatches":"L"}

Neo4j

Nejdříve byly vytvořeny 3 typy vrcholů - GAME, PLAYER, GAME_STATS. Vrcholy GAME popisují zápas a výsledky bez statistik hráčů a statistik týmů. Dále vrcholy PLAYER popisuje informace o hráčích (stejně jako v případě MongoDB) a konečně GAME_STATS, které popisují statistiky týmů během hry. Dále jsou přídány 2 druhy vztahů mezi vrcholy – PLAYS, HAS_STATS. Vztah PLAYS je mezi hráčem který hraje v zápase. Každý tento vztah má atributy zachycující statistiky daného hráče v daném zápase. Druhý vztah spojuje zápas s statistikami daného zápasu.

Vrchol GAME

```
CREATE (g0: GAME {game_id: 2019020001, season: 20192020, type: 'R',
date_time_GMT: '2019-10-02T23:00:00Z', home_team: 'Maple Leafs',
away_team: 'Senators', outcome: 'home win REG', away_goals: 3,
home_goals: 5} ), …
```

Vrchol PLAYER

```
CREATE (p0: PLAYER {player_id: 8466148, firstName: 'Marian',
lastName: 'Hossa', nationality: 'SVK', birthCity: 'Stará Lubovna',
primaryPosition: 'RW', birthDate: '1979-01-12 00:00:00',
birthStateProvince: nan, height: '6\' 1"', height_cm: 185.42,
weight: 207.0, shootsCatches: 'L'} ), …
```

Vrchol GAME_STATS

```
CREATE (gs0h: GAME_STATS {head_coach: 'Mike Babcock', shots: 42.0,
pim: 6.0, hits: 17.0} ), …
```

Vztah HAS_STATS

```
CREATE (g0)-[hs0h: HAS_STATS {team: 'Maple Leafs'}]->(gs0h), …
```

Vztah PLAYS

```
CREATE (p460)-[pl629a0: PLAYS {player_id: 8476923.0, timeOnIce:
1443.0, assists: 0.0, goals: 0.0, shots: 2.0, hits: 2.0,
powerPlayGoals: 0.0, powerPlayAssists: 0.0, penaltyMinutes: 2.0,
faceOffWins: 0.0, faceoffTaken: 0.0, takeaways: 1.0, giveaways: 1.0,
shortHandedGoals: 0.0, shortHandedAssists: 0.0, blocked: 2.0,
plusMinus: 0.0, evenTimeOnIce: 1224.0, shortHandedTimeOnIce: 61.0,
powerPlayTimeOnIce: 158.0, team: 'Devils'}]->( g629 ), …
```

Poznámka:

Jednotlivé vrcholy a vztahy byli nahrány do databáze přes webový prohlížeč kromě vztahu PLAYS. Vztah PLAYS kvůli své velikosti byl načten ze souboru csv. Jednotlivé příkazy a csv soubor byly vytvořeny v přiložením jupyter notebooku.

Dotazy

Dotaz 1

Získejte všechny zápasy v základní části 2019/2020 týmu "Bruins"

MongoDB

```
db.getSiblingDB("nhl").getCollection("game").find(
            {$or: [{home_team: "Bruins"}, {away_team: "Bruins"}],
             season: 20192020,
             type: "R"})
```

	_id	away_goals	away_players	away_team	da
1	63749c427cda553980e72615	2	[{"player_id": 8470638, "timeOnIce": 1111, "assists": 0, "goals": 0, "s	Bruins	2019
2	63749c427cda553980e72623	1	[{"player_id": 8470638, "timeOnIce": 1007, "assists": 1, "goals": 0, "s	Bruins	2019
3	63749c427cda553980e72636	4	[{"player_id": 8470638, "timeOnIce": 1113, "assists": 1, "goals": 0, "s	Bruins	2019
4	63749c427cda553980e72646	2	[{"player_id": 8470638, "timeOnIce": 1027, "assists": 1, "goals": 0, "s	Bruins	2019
5	63749c427cda553980e7264f	0	[{"player_id": 8476923, "timeOnIce": 1292, "assists": 0, "goals": 0, "s	Devils	2019

Neo4j

```
match (g:GAME)
where g.season = 20192020 and
(g.home_team = "Bruins" or g.away_team = "Bruins")
and g.type = "R"
return g
```


```
{"home_goals":1,"season":20192020,"home_team":"Stars","away_goals":2,"type":"R","date_time_GMT":"2019-10-04T00:30:00Z","outcome":"away win REG","game_id":2019020010,"away_team":"Bruins"}
{"home_goals":0,"season":20192020,"home_team":"Coyotes","away_goals":1,"type":"R","date_time_GMT":"2019-10-06T01:00:00Z","outcome":"away win REG","game_id":2019020027,"away_team":"Bruins"}
{"home_goals":3,"season":20192020,"home_team":"Golden Knights","away_goals":4,"type":"R","date_time_GMT":"2019-10-09T02:00:00Z","outcome":"away win REG","game_id":2019020043,"away_team":"Bruins"}
{"home_goals":4,"season":20192020,"home_team":"Avalanche","away_goals":2,"type":"R","date_time_GMT":"2019-10-11T01:00:00Z","outcome":"home win REG","game_id":2019020056,"away_team":"Bruins"}
{"home_goals":3,"season":20192020,"home_team":"Bruins","away_goals":0,"type":"R","date_time_GMT":"2019-10-12T23:00:00Z","outcome":"home win REG","game_id":2019020064,"away_team":"Devils"}
```


Dotaz 2

Získejte počet playoff zápasů jednotlivých týmů seřazených v sestupném pořadí v sezone 2019/2020.

MongoDB

```
db.game.aggregate([
            {$match: {type: "P", season: 20192020}},
            {$group: {_id: "$home_team", num_home_plays: {$sum:
1}}},
            { $merge : { into : "playoff_games" } }
]);

db.game.aggregate([
            {$match: {type: "P", season: 20192020}},
            {$group: {_id: "$away_team", num_away_plays: {$sum:
1}}},
            { $merge : { into : "playoff_games" } }
]);
```

```
db.playoff_games.aggregate([
            {$project: {_id: 1, num_plays: {$add:
["$num_home_plays", "$num_away_plays"]}}},
            {$sort: {num_plays: -1}}
]);
```

	{} _id	{} num_plays
1	Stars	24
2	Islanders	22
3	Lightning	22
4	Golden Knights	17
5	Canucks	17

Neo4j

```
match (g:GAME)
where g.season = 20192020 and g.type = "P"
with [g.home_team, g.away_team] as teams, g.game_id as game_id
unwind teams as team
return team, count(*) as appearances
order by appearances desc
```

"team"	"appearances"
"Stars"	24
"Islanders"	22
"Lightning"	22
"Canucks"	17
"Golden Knights"	17

Dotaz 3

Získejte hráče podle národnosti seřazené v sestupném pořadí.

MongoDB

```
db.getSiblingDB("nhl").getCollection("player").aggregate([
            {$group: {_id : "$nationality", num_players : {$sum:
1}}},
            { $sort : { num_players : -1} }
])
```

	{} _id	{} num_players
1	CAN	1896
2	USA	907
3	SWE	280
4	RUS	210
5	CZE	201

Neo4j

```
match (p:PLAYER)
return p.nationality, count(*) as count_of_players
ORDER BY count_of_players DESC
LIMIT 5
```

"p.nationality"	"count_of_players"
"CAN"	1896
"USA"	907
"SWE"	280
"RUS"	210
"CZE"	201

Dotaz 4

Získejte top 10 nejvyšších českých hráčů a vraťte jejich jména, výšku, datum narození, primární pozici.

MongoDB

```
db.player.find({nationality: "CZE"},
               {"firstName": 1,
                "lastName": 1,
                "height_cm": 1,
                "birthDate" : 1,
                "primaryPosition": 1, _id: 0}
    ).sort({"height_cm": -1}).limit(10)
```

	{} birthDate	{} firstName	{} height_cm	{} lastName	{} primaryPosition
1	1990-11-29 00:00:00	Andrej	200.66	Sustr	D
2	1987-02-20 00:00:00	Martin	198.12	Hanzal	C
3	1975-06-24 01:00:00	Marek	198.12	Malik	D
4	1981-05-12 01:00:00	David	198.12	Koci	LW
5	1978-11-10 00:00:00	Robert	195.58	Schnabel	D

Neo4j

```
match (p:PLAYER)

where p.nationality = "CZE"

return p.firstName, p.lastName, p.height_cm, p.birthDate,
p.primaryPosition

ORDER BY p.height_cm DESC

LIMIT 10
```

```
╒═══════════════╤══════════════╤═══════════════╤═══════════════════════╤═══════════════════╕
|"p.firstName"|"p.lastName"|"p.height_cm"|"p.birthDate"        |"p.primaryPosition"|
╞═══════════════╪══════════════╪═══════════════╪═══════════════════════╪═══════════════════╡
|"Andrej"     |"Sustr"     |200.66       |"1990-11-29 00:00:00"|"D"                |
├───────────────┼──────────────┼───────────────┼───────────────────────┼───────────────────┤
|"Marek"      |"Malik"     |198.12       |"1975-06-24 01:00:00"|"D"                |
├───────────────┼──────────────┼───────────────┼───────────────────────┼───────────────────┤
|"Martin"     |"Hanzal"    |198.12       |"1987-02-20 00:00:00"|"C"                |
├───────────────┼──────────────┼───────────────┼───────────────────────┼───────────────────┤
|"David"      |"Koci"      |198.12       |"1981-05-12 01:00:00"|"LW"               |
├───────────────┼──────────────┼───────────────┼───────────────────────┼───────────────────┤
|"Michal"     |"Sykora"    |195.58       |"1973-07-05 01:00:00"|"D"                |
└───────────────┴──────────────┴───────────────┴───────────────────────┴───────────────────┘
```

Dotaz 5

Získání počtu gólů vstřelených hráči v základní sezóně 2019/2020 seřazených v sestupném pořadí.

MongoDB

```
 db.game.aggregate([
                    {$match: {season: 20192020, type: "R"} },
                    {$project: {all_players: {$setUnion:
["$home_players", "$away_players"]}}},
                    {$unwind: "$all_players"},
                    {$group: {_id: "$all_players.player_id",
num_goals: {$sum: "$all_players.goals"}}},
                    {$lookup: {
                            from: "player",
                            localField: "_id",
                            foreignField: "player_id",
                            as: "player_info"
                    }},
                    {
                        $replaceRoot: { newRoot: { $mergeObjects: [
{ $arrayElemAt: [ "$player_info", 0 ] }, "$$ROOT" ] } }
                    },
                    {$project: {
                            firstName : 1,
                            lastName: 1,
```

```
                    nationality: 1,
                    num_goals: 1
                }},
                {$sort: {num_goals: -1}}
])
```

	_id	firstName	lastName	nationality	num_goals
1	8477956	David	Pastrnak	CZE	48
2	8471214	Alex	Ovechkin	RUS	48
3	8479318	Auston	Matthews	USA	47
4	8477934	Leon	Draisaitl	DEU	43
5	8476459	Mika	Zibanejad	SWE	41

Neo4j

```
match (g:GAME)

where g.season = 20192020 and g.type = "R"

match (p:PLAYER)-[pl:PLAYS]->(g:GAME)

return p.firstName, p.lastName, p.player_id, p.nationality,
sum(pl.goals) as goals

order by goals desc
```

"p.firstName"	"p.lastName"	"p.player_id"	"p.nationality"	"goals"
"Alex"	"Ovechkin"	8471214	"RUS"	48
"David"	"Pastrnak"	8477956	"CZE"	48
"Auston"	"Matthews"	8479318	"USA"	47
"Leon"	"Draisaitl"	8477934	"DEU"	43
"Mika"	"Zibanejad"	8476459	"SWE"	41

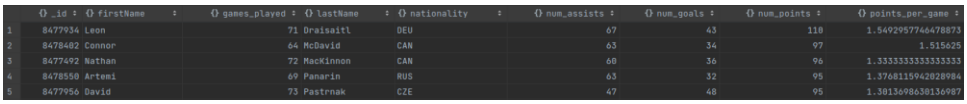
Dotaz 6

Získejte góly, asistence, body, odehrané zápasy a body na zápas hráčů v základní části sezóny 2019/2020 seřazené podle bodů v sestupném pořadí.

MongoDB

```
db.game.aggregate([
                    {$match: {season: 20192020, type: "R"} },
                    {$project: {all_players: {$setUnion:
["$home_players", "$away_players"]}}},
                    {$unwind: "$all_players"},
                    {$group: {_id: "$all_players.player_id",
num_goals: {$sum: "$all_players.goals"}, num_assists: {$sum:
"$all_players.assists"}, games_played: {$sum: 1}}},
                    {$lookup: {
                            from: "player",
                            localField: "_id",
                            foreignField: "player_id",
```

```
                        as: "player_info"
                    }},
                    {
                        $replaceRoot: { newRoot: { $mergeObjects: [
{ $arrayElemAt: [ "$player_info", 0 ] }, "$$ROOT" ] } }
                    },
                    {$project: {
                        firstName : 1,
                        lastName: 1,
                        nationality: 1,
                        games_played: 1,
                        num_goals: 1,
                        num_assists: 1,
                        num_points: {$add: ["$num_goals",
"$num_assists"]},
                        points_per_game: {$divide: [{$add:
["$num_goals", "$num_assists"]}, "$games_played"]}
                    }},
                    {$sort: {num_points: -1}}
])
```



	_id	firstName	games_played	lastName	nationality	num_assists	num_goals	num_points	points_per_game
1	8477934	Leon	71	Draisaitl	DEU	67	43	110	1.5492957746478873
2	8478402	Connor	64	McDavid	CAN	63	34	97	1.515625
3	8477492	Nathan	72	MacKinnon	CAN	60	36	96	1.3333333333333333
4	8478550	Artemi	69	Panarin	RUS	63	32	95	1.3768115942028984
5	8477956	David	73	Pastrnak	CZE	47	48	95	1.3013698630136987

Neo4j

```
match (g:GAME)
where g.season = 20192020 and g.type = "R"
match (p:PLAYER)-[pl:PLAYS]->(g:GAME)
return p.firstName,
      p.lastName,
      p.player_id,
      p.nationality,
      sum(pl.goals) as goals,
      sum(pl.assists) as assists,
      sum(pl.goals + pl.assists) as points,
      count(*) as occurences,
      toFloat(sum(pl.goals + pl.assists)) / count(*) as
points_per_game
order by points desc
```

"p.firstName"	"p.lastName"	"p.player_id"	"p.nationality"	"goals"	"assists"	"points"	"occurences"	"points_per_game"
"Leon"	"Draisaitl"	8477934	"DEU"	43	67	110	71	1.5492957746478873
"Connor"	"McDavid"	8478402	"CAN"	34	63	97	64	1.515625
"Nathan"	"MacKinnon"	8477492	"CAN"	36	60	96	72	1.3333333333333333
"Artemi"	"Panarin"	8478550	"RUS"	32	63	95	69	1.3768115942028984
"David"	"Pastrnak"	8477956	"CZE"	48	47	95	73	1.3013698630136987

Dotaz 7

Top 5 nejvyšší gólový rozdíl v jednom zápase mezi dvěma týmy v sezóně 2019/2020.

MongoDB

```
db.game.aggregate([
                    {$project: {
                        home_team: 1,
                        away_team: 1,
                        home_goals: 1,
                        away_goals: 1,
                        goal_difference: {$abs: {$subtract:
["$home_goals", "$away_goals"]}}
                    }},
                    {$sort: {goal_difference: -1}},
                    {$limit: 5}
])
```

	_id	away_goals	away_team	goal_difference	home_goals	home_team
1	63749c427cda553980e728fc	0	Stars	7	7	Wild
2	63749c427cda553980e728a1	2	Canucks	7	9	Lightning
3	63749c427cda553980e72780	8	Bruins	7	1	Canadiens
4	63749c427cda553980e7271c	2	Senators	6	8	Hurricanes
5	63749c427cda553980e72638	2	Kings	6	8	Canucks

Neo4j

```
match (g:GAME)

return g.home_team, g.away_team, abs(g.home_goals - g.away_goals) as
goal_diff, g.home_goals, g.away_goals

order by goal_diff desc
```

"g.home_team"	"g.away_team"	"goal_diff"	"g.home_goals"	"g.away_goals"
"Canadiens"	"Bruins"	7	1	8
"Lightning"	"Canucks"	7	9	2
"Wild"	"Stars"	7	7	0
"Canucks"	"Kings"	6	8	2
"Penguins"	"Flyers"	6	7	1

Dotaz 8

Týmy, proti kterým dal gól hráč "Alex Ovechkin" a počet těchto golů.

MongoDB

```
 db.game.aggregate([
        {$match: {season: 20192020, type: "R"} },
        {$unwind: "$away_players"},
        {$lookup: {
            from: "player",
            localField: "away_players.player_id",
            foreignField: "player_id",
            as: "player_info"
        }},
        {$replaceRoot: { newRoot: { $mergeObjects: [ { $arrayElemAt:
[ "$player_info", 0 ] }, "$$ROOT" ] } }},
        {$match: {lastName: "Ovechkin", firstName: "Alex"}},
        {$group: {
            _id: "$home_team",
            goals_recieved_home: {$sum: "$away_players.goals"}
        }},
        { $merge : { into : "Ovechkin_scored" } }
]);

db.game.aggregate([
        {$match: {season: 20192020, type: "R"} },
        {$unwind: "$home_players"},
        {$lookup: {
            from: "player",
            localField: "home_players.player_id",
            foreignField: "player_id",
            as: "player_info"
        }},
        {$replaceRoot: { newRoot: { $mergeObjects: [ { $arrayElemAt:
[ "$player_info", 0 ] }, "$$ROOT" ] } }},
        {$match: {lastName: "Ovechkin", firstName: "Alex"}},
        {$group: {
            _id: "$away_team",
            goals_recieved_away: {$sum: "$home_players.goals"}
        }},
        {$sort: {goals_recieved: -1}},
        { $merge : { into : "Ovechkin_scored" } }
]);

db.Ovechkin_scored.aggregate([
        {$project: {
            goals_recieved: {$add: ["$goals_recieved_home",
"$goals_recieved_away"]}
        }},
        {$sort: {goals_recieved: -1}}
]);
```

	{} _id	{} goals_recieved
1	Devils	5
2	Senators	4
3	Kings	3
4	Panthers	3
5	Hurricanes	3

Neo4j

```
match (g:GAME)
where g.type = "R" and g.season = 20192020
match (p:PLAYER)-[pl:PLAYS]->(g:GAME)
where p.lastName = "Ovechkin" and p.firstName = "Alex"
return CASE
WHEN pl.team = g.away_team THEN g.home_team
ELSE g.away_team
END as opposing_team, sum(pl.goals) as Ovechkins_goals
order by Ovechkins_goals desc
```

"opposing_team"	"Ovechkins_goals"
"Devils"	5
"Senators"	4
"Islanders"	3
"Hurricanes"	3
"Predators"	3

Dotaz 9

Získejte všechny kanadské provincie a jejich počty hráčů, kteří hráli v jakékoli sezóně.

MongoDB

```
db.getSiblingDB("nhl").getCollection("player").aggregate([
                    {$match: {nationality: "CAN"}},
                    {$project: {player_id: 1, birthStateProvince:
1}},
                    {$group: {_id: "$birthStateProvince",
num_of_players: {$sum: 1}}},
                    {$sort: {num_of_players: -1}}
])
```

	{} _id	{} num_of_players
1	ON	738
2	QC	301
3	AB	295
4	BC	200
5	SK	160

Neo4j

```
match (p:PLAYER)
where p.nationality = "CAN"
return p.birthStateProvince, count(*) as players
order by players desc
```

"p.birthStateProvince"	"players"
"ON"	738
"QC"	301
"AB"	295
"BC"	200
"SK"	160

Dotaz 10

Získejte týmy s nejvíce vítězstvími v základní hrací době v sezóně 2019/2020 (základní část + playoff) seřazené v sestupném pořadí.

MongoDB

```
db.game.aggregate([
                {$match: {season: 20192020}},
                {$group: {
                    _id: "$home_team",
                    home_wins: {$sum:
                            {$cond: [ { $gt: [ "$home_goals",
"$away_goals" ] },
                                    1,
                                     0]}},
                }},
                { $merge : { into : "team_home_wins" } }
]);

db.game.aggregate([
                {$match: {season: 20192020}},
```

```
                {$group: {
                    _id: "$away_team",
                    away_wins: {$sum:
                            {$cond: [ { $lt: [ "$home_goals",
"$away_goals" ] },
                                1,
                                0]}},
                }},
                { $merge : { into : "team_away_wins" } }
]);

db.team_home_wins.aggregate([
                {$lookup: {
                    from: "team_away_wins",
                    localField: "_id",
                    foreignField: "_id",
                    as: "wins"
                }},
                {$replaceRoot: { newRoot: { $mergeObjects: [ {
$arrayElemAt: [ "$wins", 0 ] }, "$$ROOT" ] } }},
                {$project: {
                    _id: 1,
                    wins: {$add: ["$away_wins", "$home_wins"]}
                }},
                {$sort: {wins: -1}}
]);
```

	{} _id	{} wins
1	Lightning	61
2	Stars	52
3	Golden Knights	51
4	Flyers	51
5	Avalanche	51

Neo4j

```
match (g:GAME)

where g.season = 20192020

with [{team: g.home_team, win: case when g.home_goals > g.away_goals
then 1 else 0 end},

{team: g.away_team, win: case when g.away_goals > g.home_goals then
1 else 0 end}] as teams, g.game_id as game_id

unwind teams as team

return team.team as team, sum(team.win) as wins

order by wins desc
```

"team"	"wins"
"Lightning"	61
"Stars"	52
"Golden Knights"	51
"Avalanche"	51
"Flyers"	51

Závěr

V této semestrální páci jsem si vybral databázi NHL, která byla poměrně velká, takže vytváření create scriptu pro jednotlivé databáze bylo časově náročné. Nahrávání dat do MongoDB bylo velmi ryhchlé oproti Neo4j, kde pro jisté vztahy nahrávání trvalo řádově déle. Na druhou stranu, MongoDB bylo mnohem méně přímočaré v dotazování a dotazy byly komplikovanější i více časově náročné. Neo4j v tomto ohledu výrazně předčilo MongoDB.

Při dokončování práce jsem se přesvědčil, že některé části mých databází mohli být navrženy lépe. Zejména rozdělení na tým domácích a hostů bylo problematické v dotazování v MongoDB.