DataViadotto

ENTITY/RELATIONSHIP PROFILING WITH THE DATAVIADOTTO PROFILER

Whitepaper

SUMMARY

Entity/Relationship profiling is the task of computing all keys and foreign keys that hold on given data sets. In previous whitepapers we had argued that the DataViadotto Profiler is the only tool available to accomplish this task, and how the results fundamentally help any data professional deliver value. Indeed, it becomes possible to explore all insight stories hidden in an organization's data assets by identifying all their business entities and their relationships.

In this whitepaper we will illustrate the distinguished features of the DataViadotto Profiler on a publicly curated Hockey data set. Step by step, we will go through the process of connecting to the data source, sampling all available tables, selecting parameters for the discovery algorithms, finding all keys and foreign keys based on the parameters, and exploring the meaningfulness of the results based on sample data. Without re-organizing the given tables, we will then substantially improve the underlying logical model of data based on our findings, and also point out several opportunities to improve data quality.

1. The Hockey Data Set

The Hockey data set is publicly accessible at https://relational.fit.cvut.cz/dataset/Hockey and was originally sourced from http://www.opensourcesports.com/hockey/ . In addition to the NHL, the Hockey data set covers the following early and alternative leagues: NHA, PCHA, WCHL and WHA. It contains individual and team statistics from the 1909/10 through to the 2011/12 season. Together, it contains 22 tables, 96,403 rows and 300 columns, and has a size of 15.6 MB.

The original conceptual data model is illustrated in Fig. 1 on the next page. Out of the 22 tables, nine tables have neither a primary key nor any unique constraints specified on them, while the remaining 13 tables have only a primary key specified on them without any other unique constraint. When a field name is part of the primary key of a table, the name of the field is underlined and the letters PK for Primary Key appear next to it. Some of the referential integrity constraints are not foreign keys since they do not reference a unique constraint (this is a minimal requirement on any foreign key, and it means, in particular, that those constraints do also not reference the primary key of the table if it exists). Not having a canidate key specified on the table and having referential constraints that are not foreign keys violates basic design principles. There are other database design issues, such as providing a single table (the table called Master) for different people such as players, coaches, managers etc., which is one of the reasons why no candidate key exists for this table. However, the purpose of this white paper is not to discuss database design but to focus on the value that Entity/Relationship profiling can bring.

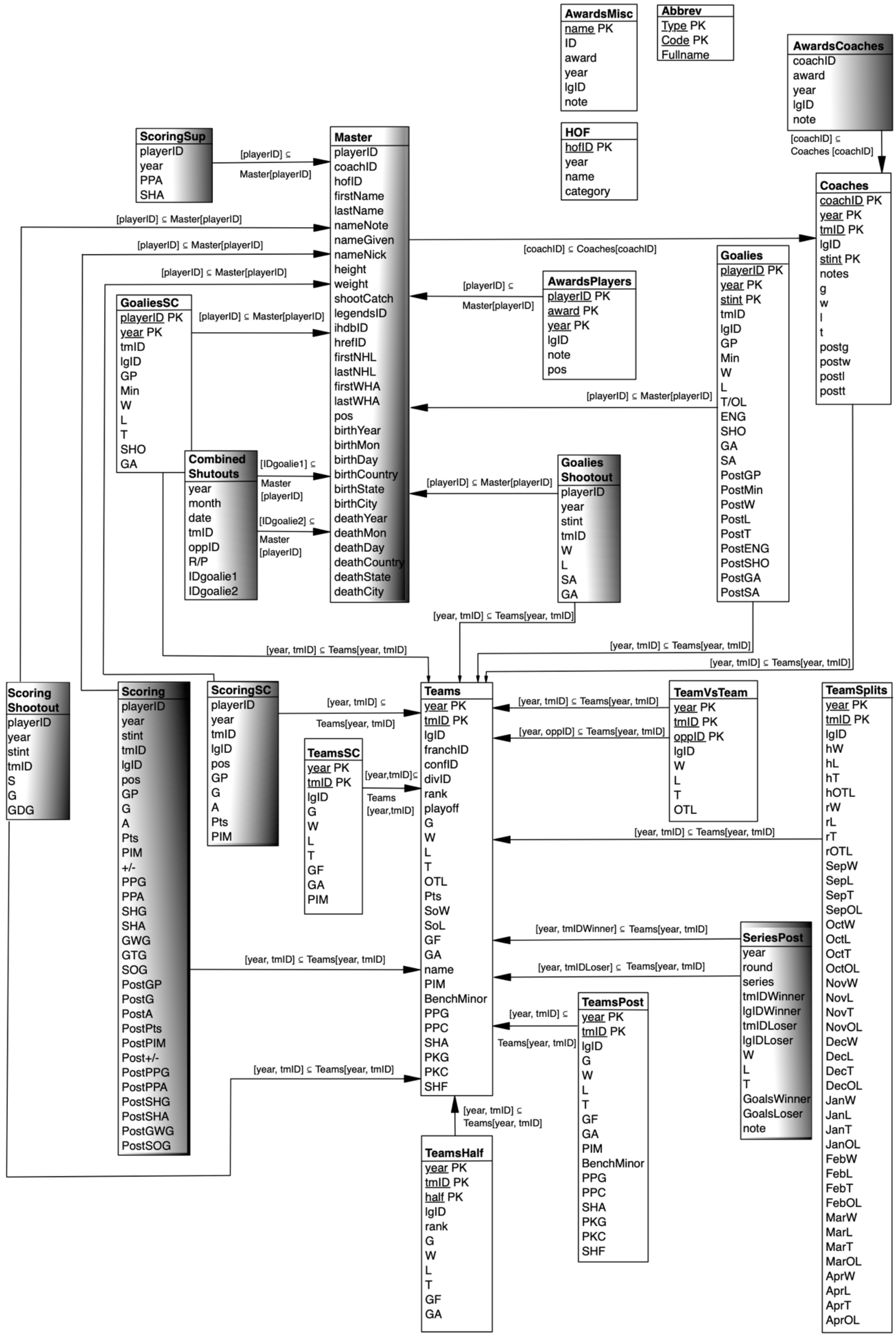


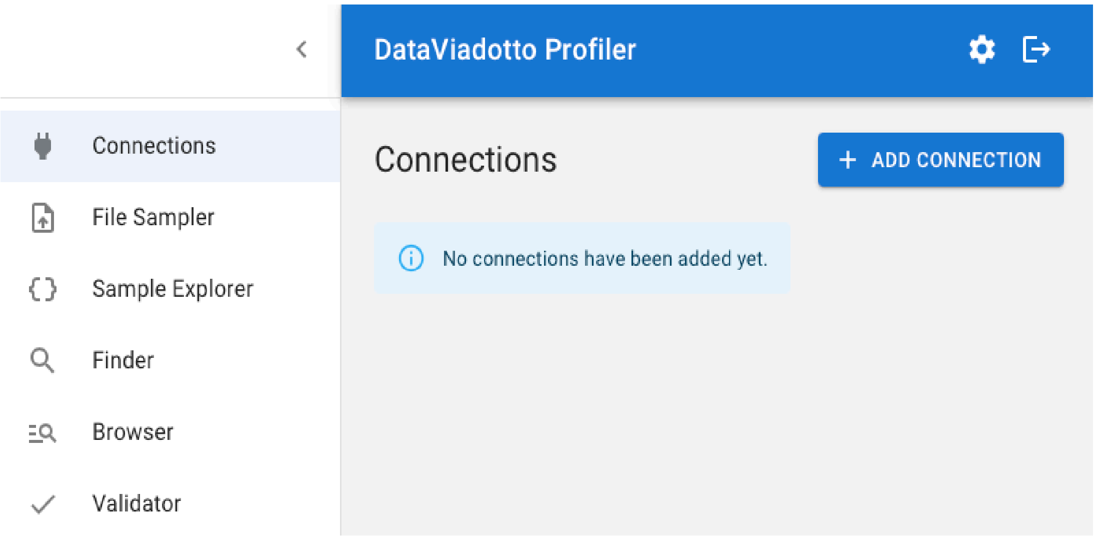
Fig. 1: Details of the Original Conceptual Diagram for the Hockey Data Set

2. Applying Entity/Relationship Profiling to the Hockey Data Set

We will now go through a typical process of Entity/Relationship Profiling by applying the DataViadotto Profiler to the Hockey Data Set. The simple graphical user interface is shown in Fig. 2, outlining the six main items in the Profiler menu:

1. Connections
2. File Sampler
3. Sample Explorer
4. Finder
5. Browser
6. Validator

Let's illustrate these steps on our showcase example.



Connections

No connections have been added yet.

Fig. 2: Main Menu Items and Connections Panel of DataViadotto Profiler

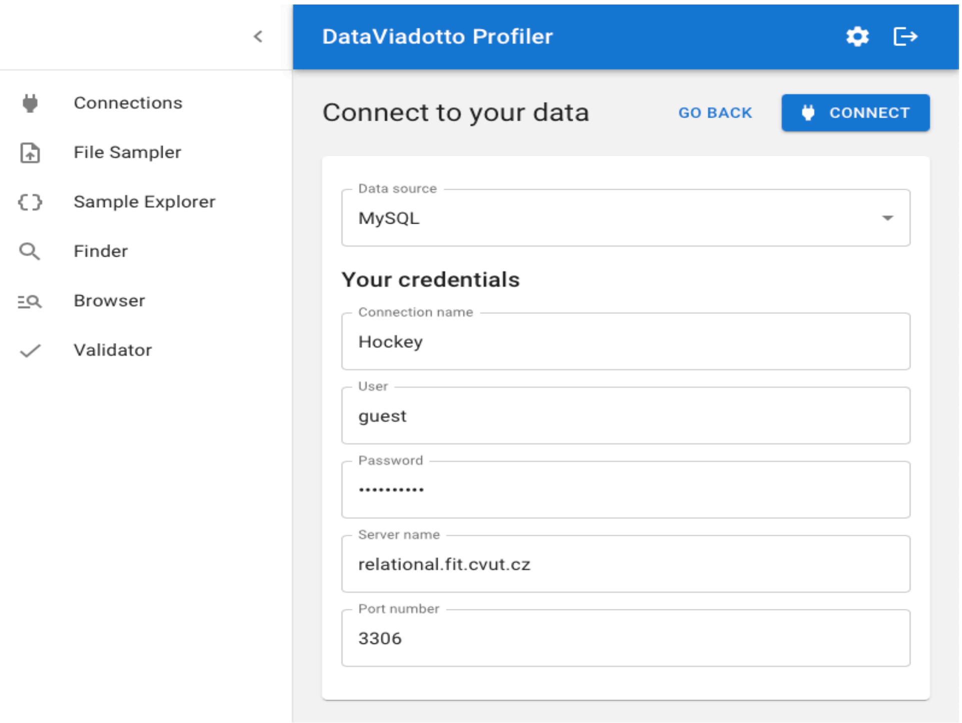
2.1 Connect

First we connect to the Hockey data source through the Connections panel. This is easily achieved using the login details from the public repository, see Fig. 3:

Data source:	MySQL
Connection:	Hockey
User:	guest
Password:	hockey
Server name:	relational.fit.cvut.cz
Port number:	3306

After all available data sets appear, simply scroll down the list and select *Hockey*.

Instead of connection to a data repository as just described, it is also possible to upload some csv files that can be mined. In this case, you may use the file sampler item from the main menu.



Connect to your data

Data source: MySQL

Your credentials

Connection name: Hockey

User: guest

Password: ••••••••••

Server name: relational.fit.cvut.cz

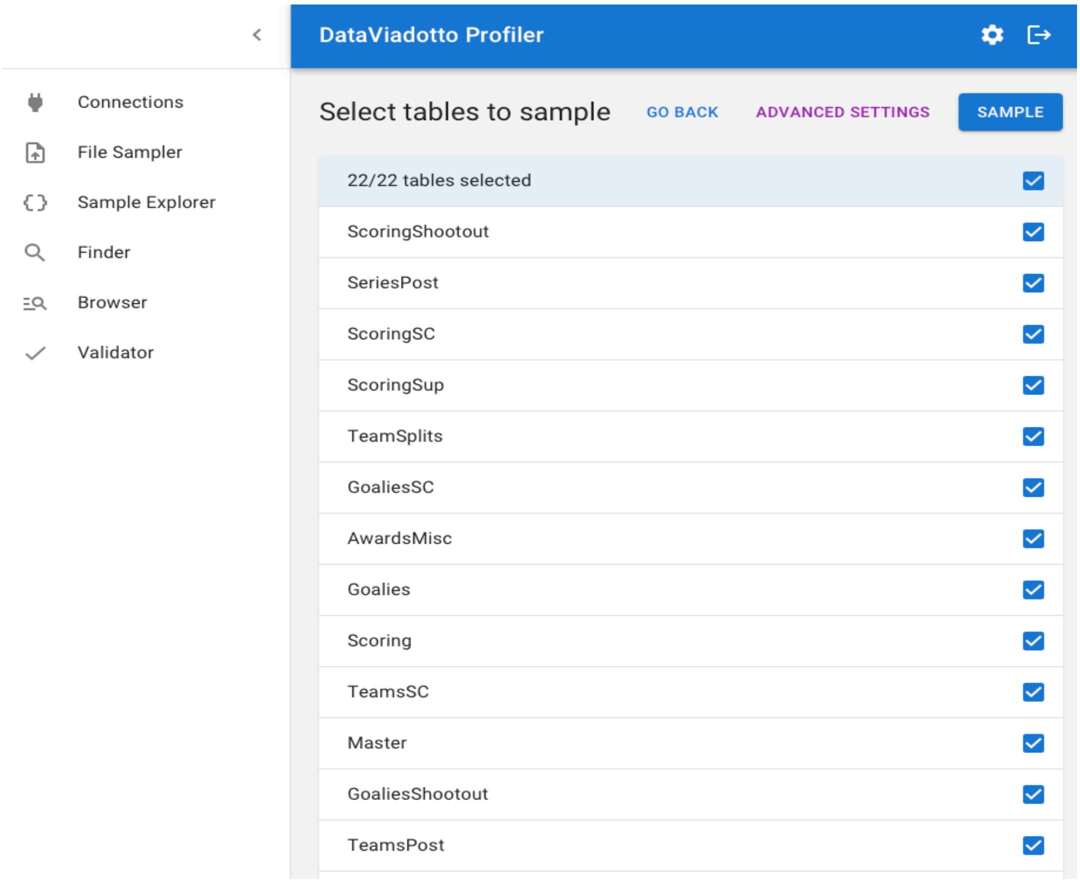
Port number: 3306

Fig. 3: Connections Panel with Details of Credentials

2.2 Sampling

Now that you are connected to the data sources, you have access to all tables, in this case you should see 22 different tables, with the details shown in Fig. 1.

The next step of the process is sampling where we apply unique methods to retain those records for profiling that will provide different results. Among other things, the sampling process allows us to scale Entity/Relationship profiling for data sets with huge numbers of records. For a modestly-sized data set such as Hockey, we will simply retain all records by default. On our local desktop machine, for example, sampling took only 17 seconds. It is important to point out that any tables, intended for use of profiling, need to undergo sampling first. Fig. 4 shows a screenshot of the Sample Explorer panel where users can select tables for sampling.

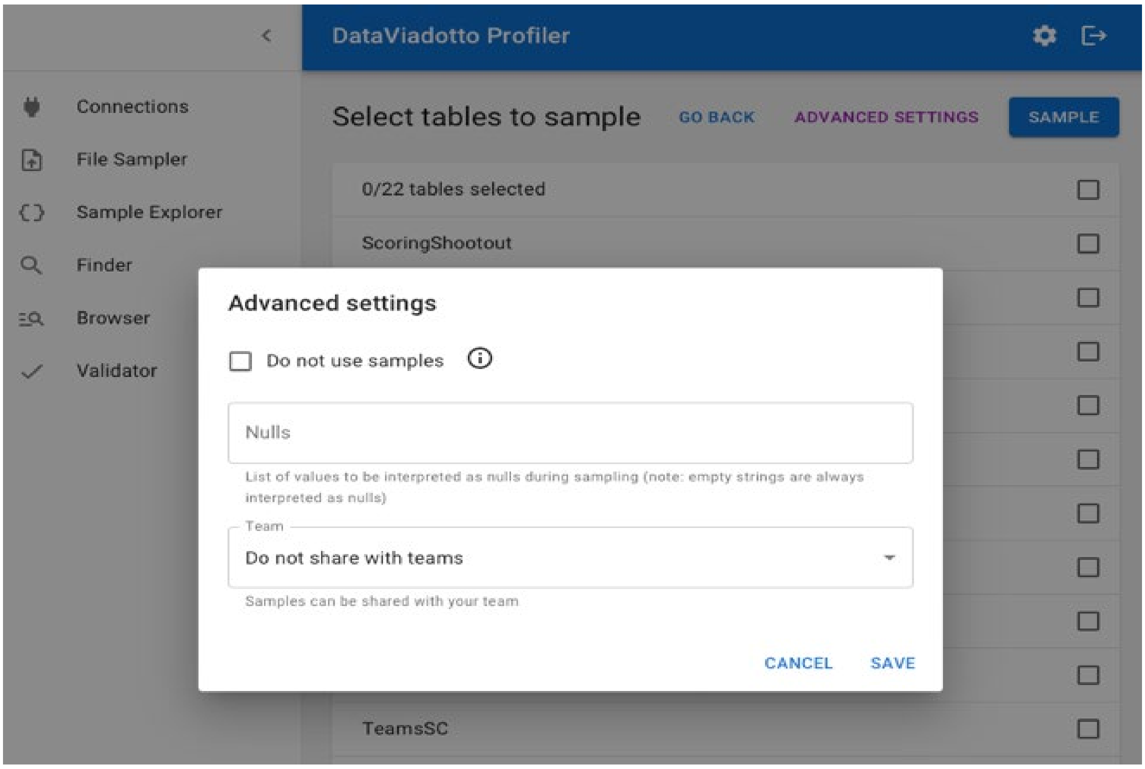


Select tables to sample

- 22/22 tables selected ☑
- ScoringShootout ☑
- SeriesPost ☑
- ScoringSC ☑
- ScoringSup ☑
- TeamSplits ☑
- GoaliesSC ☑
- AwardsMisc ☑
- Goalies ☑
- Scoring ☑
- TeamsSC ☑
- Master ☑
- GoaliesShootout ☑
- TeamsPost ☑

Fig. 4: Selecting Tables for Sampling

Another important feature of sampling is the ability to dedicate special attention to missing data values. The empty string, for example, is always interpreted as an occurrence of the null marker, but the user is able to enter additional values that the sampler should interpret as null marker occurrences. Fig. 5 shows the pop-up menu under Advanced Settings, where users can i) override the default sampling process and retain all records for profiling, ii) provide input of further values that the sampler should interpret as missing, and iii) select with which members of a team the samples should be shared.



Advanced settings

□ Do not use samples ⓘ

Nulls

List of values to be interpreted as nulls during sampling (note: empty strings are always interpreted as nulls)

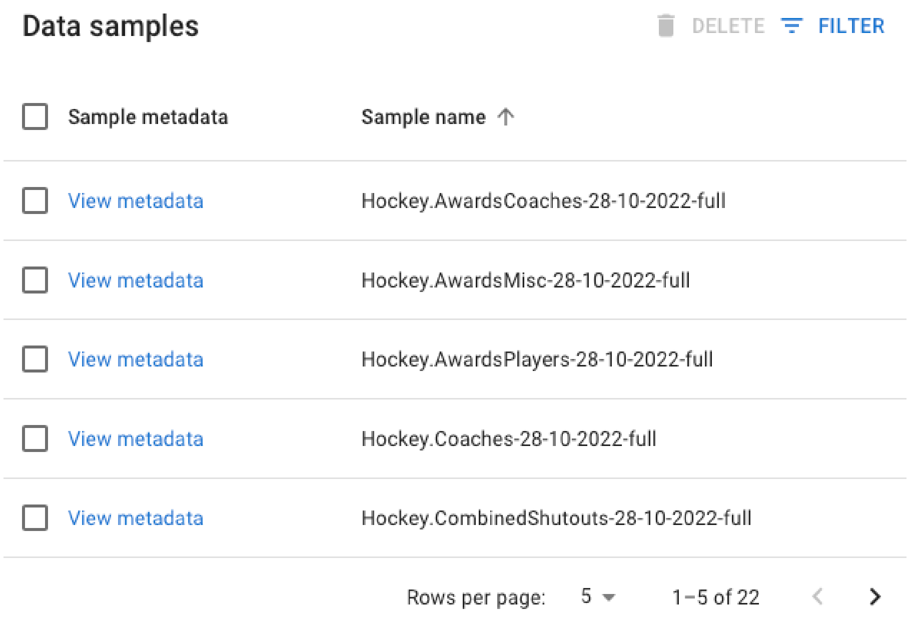
Team: Do not share with teams

Samples can be shared with your team

Fig. 5: Pop-up to Select Parameters for Sampling

After the sampling process has completed, the data samples are available for profiling, and listed as shown in Fig. 6 using timestamps. This is useful, for example, when tracking Entity/Relationship Profiling results over time to detect data drifts.

Fig. 6 also shows that metadata is available for each sample. This contains elementary profiling information for each field of every sample, including counts of *nulls*, *distinct values*, and *rows*.



□ Sample metadata	Sample name ↑
□ View metadata	Hockey.AwardsCoaches-28-10-2022-full
□ View metadata	Hockey.AwardsMisc-28-10-2022-full
□ View metadata	Hockey.AwardsPlayers-28-10-2022-full
□ View metadata	Hockey.Coaches-28-10-2022-full
□ View metadata	Hockey.CombinedShutouts-28-10-2022-full

Fig. 6: Listing with Time-stamped Samples of Data

Fig. 7 shows such metadata for the Table called *Master*.

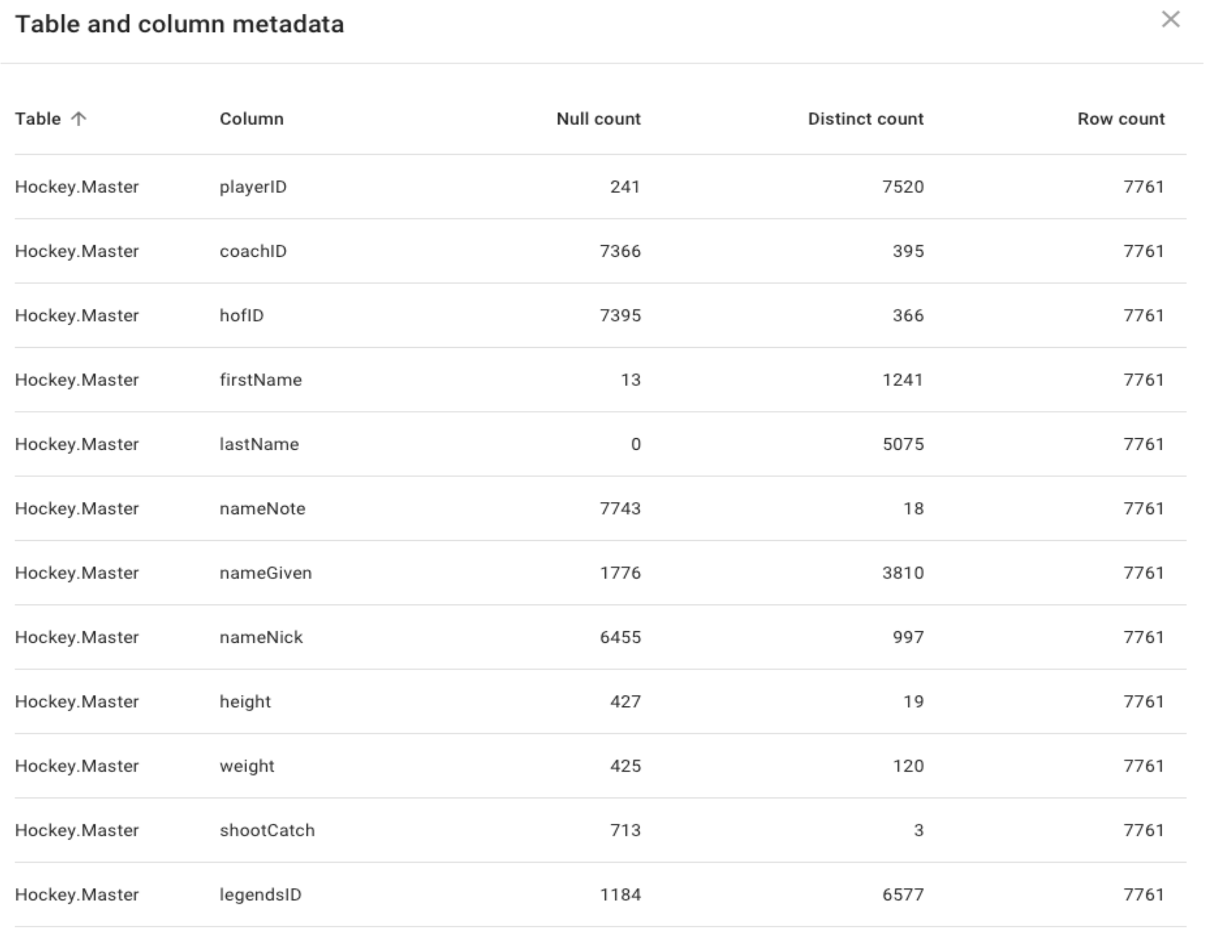


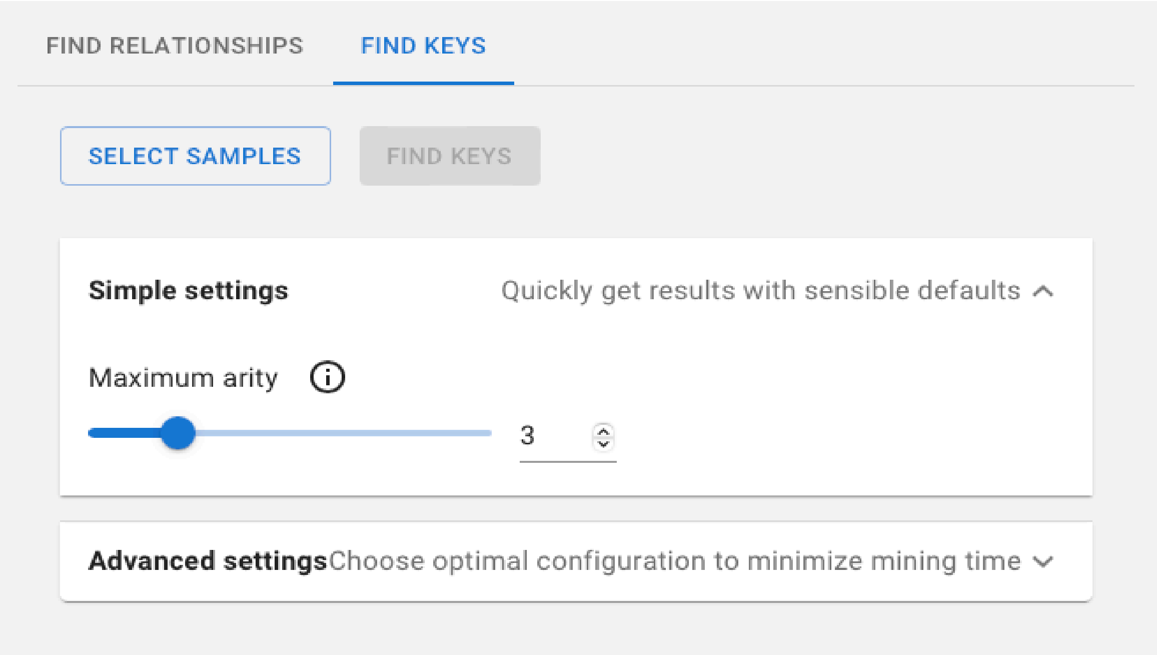
Table and column metadata

Table ↑	Column	Null count	Distinct count	Row count
Hockey.Master	playerID	241	7520	7761
Hockey.Master	coachID	7366	395	7761
Hockey.Master	hofID	7395	366	7761
Hockey.Master	firstName	13	1241	7761
Hockey.Master	lastName	0	5075	7761
Hockey.Master	nameNote	7743	18	7761
Hockey.Master	nameGiven	1776	3810	7761
Hockey.Master	nameNick	6455	997	7761
Hockey.Master	height	427	19	7761
Hockey.Master	weight	425	120	7761
Hockey.Master	shootCatch	713	3	7761
Hockey.Master	legendsID	1184	6577	7761

Fig. 7: Metadata for the Table called Master with Counts of Null, Distinct Values, and Rows for each Field

2.3 Finding and browsing keys

Things are now starting to get exciting as the *Finder* menu item is next in our list. Indeed, we first focus on finding keys. For that purpose, we can see the panel in Fig. 8 that lets users decide which keys they would like to find. More precisely, users select the maximum arity, which denotes up to how many field names any minimal key we aim to find can have. Most composite keys have not more than three field names, so the default value is sensible and also ensures that results are quickly returned. For instance, it will take about 4 seconds to mine all 22 tables together and return a total of 127 valid minimal keys. Here, minimal means that removing any field name from a key does not result in a valid key, that is, multiple records with matching values on all the field names exist.



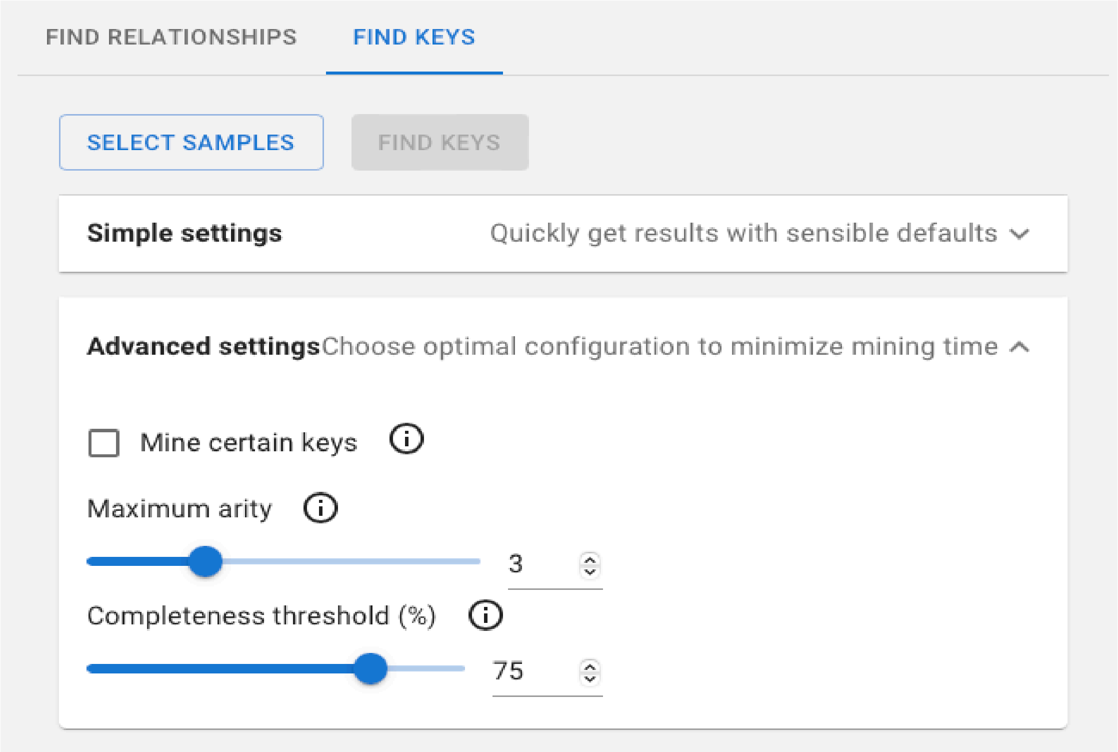
Simple settings Quickly get results with sensible defaults

Maximum arity 3

Advanced settings Choose optimal configuration to minimize mining time

Fig. 8: Panel for Selecting Parameters to Find Keys

You may have noticed the item Advanced Settings, where users can provide more input to the key finder. In particular, users get access to change the default value of 75% for the required completeness threshold by which any mined key should hold. Strictly speaking, constraints that have null marker occurrences in any of their fields cannot be called candidate keys as they do not uniquely identify every record in the table. For that purpose, these constraints are called unique constraints (UCs) by the SQL standard. Hence, candidate keys are special unique constraints with a completeness ratio of 100%, and the primary key is a distinguished candidate key. Again, note that all the constraints, including primary keys, candidate keys, and unique constraints, returned by our algorithm are minimal. Uniqueness constraints can thus be ranked by their completeness ratio as it indicates how many records they are able to identify uniquely. Users may also mine certain keys, which are special composite candidate keys that can uniquely identify every record even though some fields may have missing values.



Simple settings Quickly get results with sensible defaults

Advanced settings Choose optimal configuration to minimize mining time

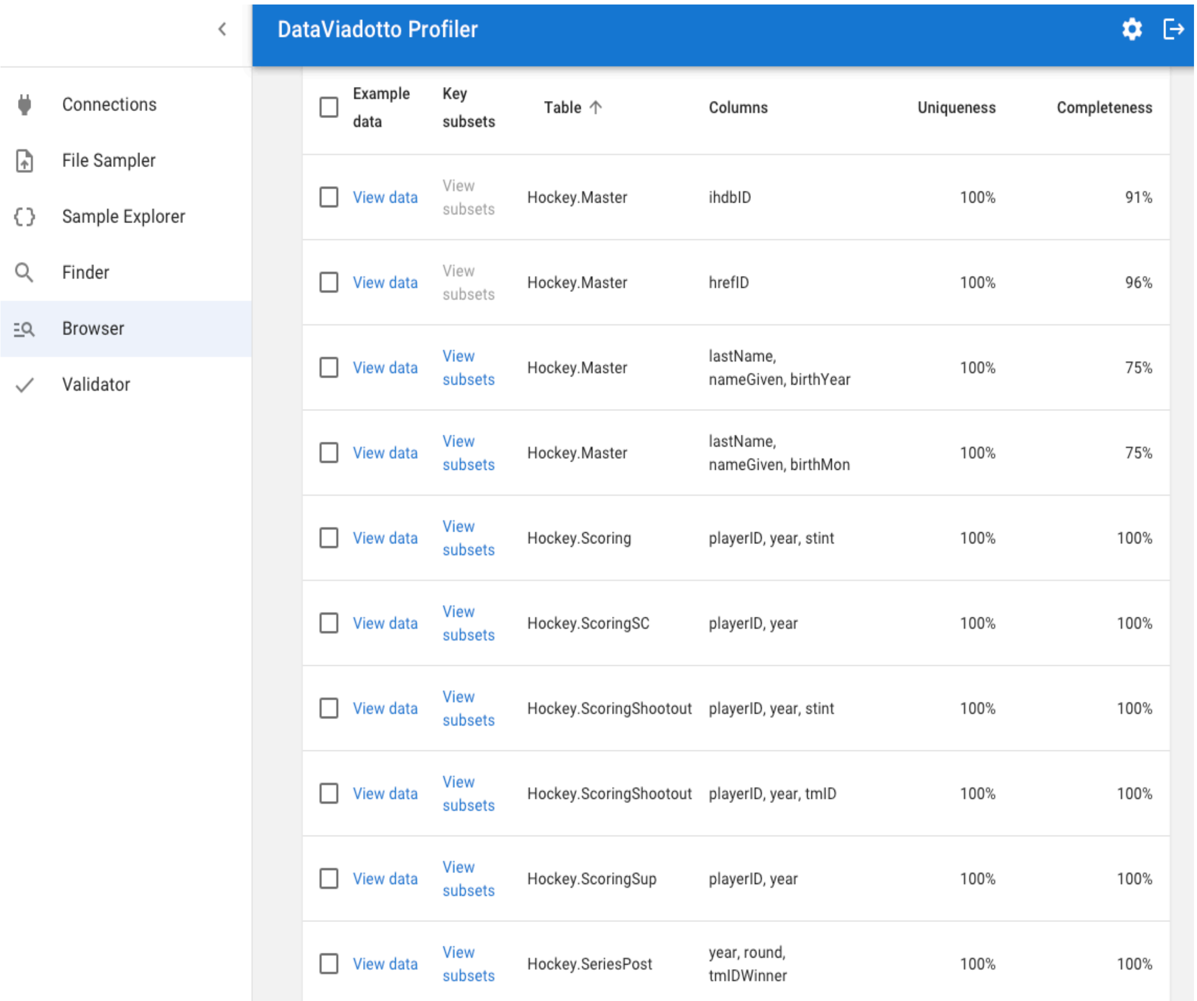
□ Mine certain keys

Maximum arity: 3

Completeness threshold (%): 75

Fig. 9: Advanced Settings Panel to Find Keys

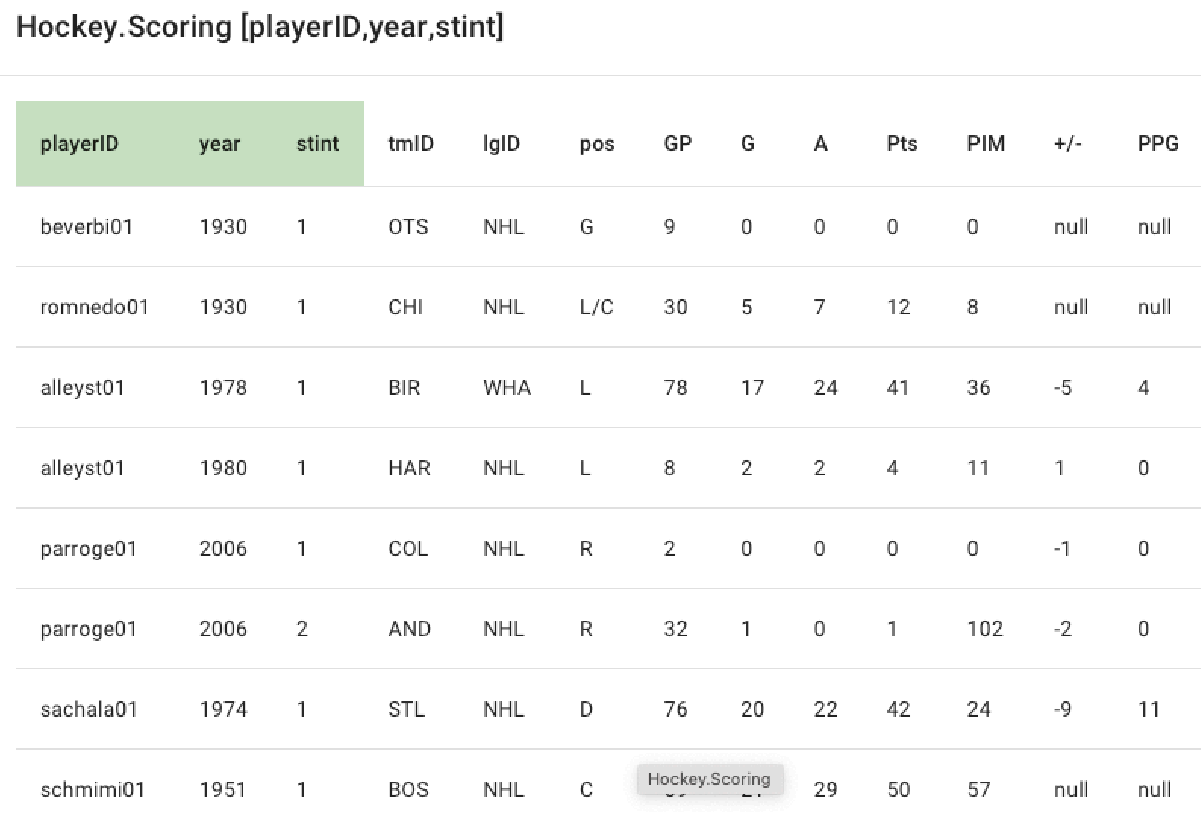
Fig. 10 shows a screenshot of the results panel for the Key Finder Algorithm. Users can rank the results by Table, Column, and Completeness. As the mined unique constraints all hold with 100% uniqueness and are unique, there cannot be any two different records in the table with matching non-null values on all the fields of any constraint. However, we may have meaningful unique constraints that are violated due to entity duplication as the constraints are not enforced or have gone unnoticed. Such constraints then appear as subsets of the mined unique constraints, and our browsing panel gives full access to inspecting all of the subsets. For instance, we may be interesting in learning more about the composite key *{playerID, year, stint}* on the table called *Scoring*.



□	Example data	Key subsets	Table ↑	Columns	Uniqueness	Completeness
□	View data	View subsets	Hockey.Master	ihdbID	100%	91%
□	View data	View subsets	Hockey.Master	hrefID	100%	96%
□	View data	View subsets	Hockey.Master	lastName, nameGiven, birthYear	100%	75%
□	View data	View subsets	Hockey.Master	lastName, nameGiven, birthMon	100%	75%
□	View data	View subsets	Hockey.Scoring	playerID, year, stint	100%	100%
□	View data	View subsets	Hockey.ScoringSC	playerID, year	100%	100%
□	View data	View subsets	Hockey.ScoringShootout	playerID, year, stint	100%	100%
□	View data	View subsets	Hockey.ScoringShootout	playerID, year, tmID	100%	100%
□	View data	View subsets	Hockey.ScoringSup	playerID, year	100%	100%
□	View data	View subsets	Hockey.SeriesPost	year, round, tmIDWinner	100%	100%

Fig. 10: Panel for Browsing Mined Keys

Clicking on *View data* leads to the screen shown in Fig. 11, which enables users to inspect carefully chosen records that illustrate why this key is minimal. Indeed, the first two records have matching values on year and stint, records three and four have matching values on playerID and stint, and records five and six have matching values on playerID and year. As each of these records appears to be reasonable, the composite key *{playerID, year, stint}* seems to represent a good choice of a meaningful business key.

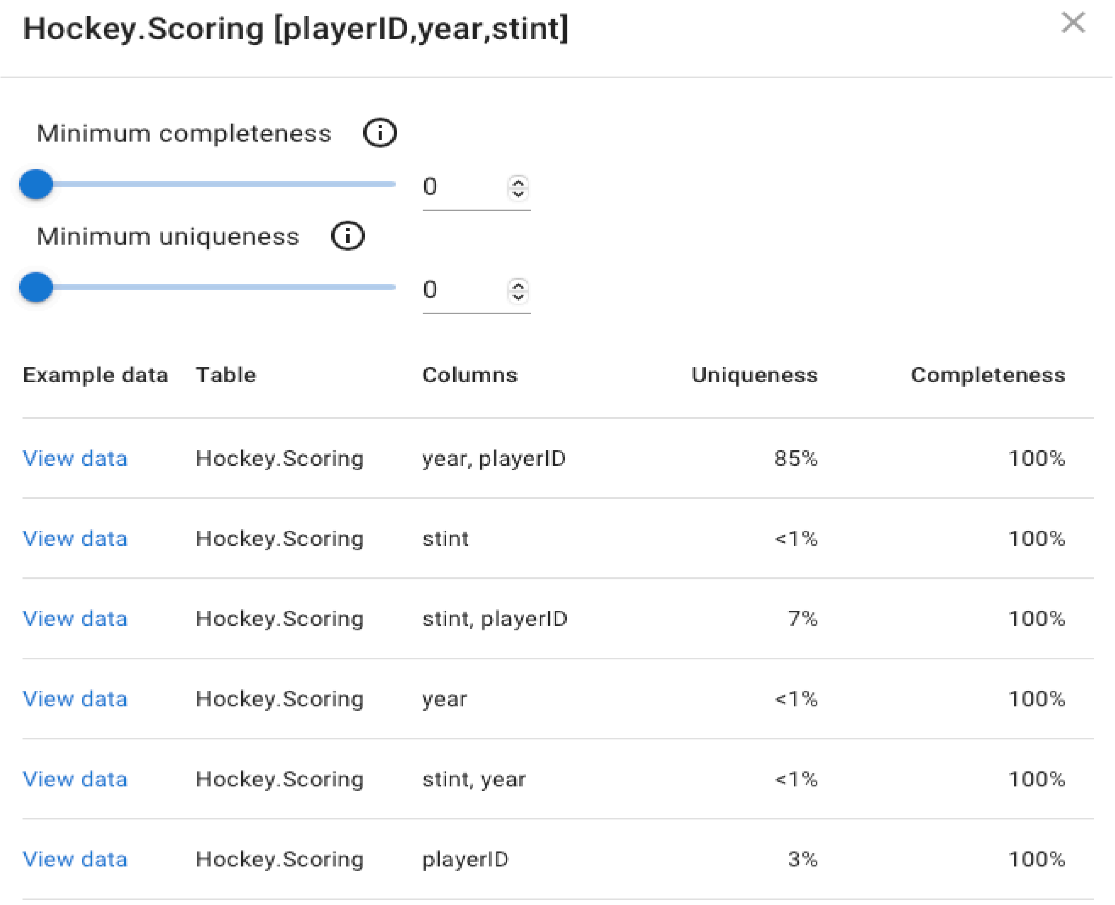


Hockey.Scoring [playerID,year,stint]

playerID	year	stint	tmID	lgID	pos	GP	G	A	Pts	PIM	+/-	PPG
beverbi01	1930	1	OTS	NHL	G	9	0	0	0	0	null	null
romnedo01	1930	1	CHI	NHL	L/C	30	5	7	12	8	null	null
alleyst01	1978	1	BIR	WHA	L	78	17	24	41	36	-5	4
alleyst01	1980	1	HAR	NHL	L	8	2	2	4	11	1	0
parroge01	2006	1	COL	NHL	R	2	0	0	0	0	-1	0
parroge01	2006	2	AND	NHL	R	32	1	0	1	102	-2	0
sachala01	1974	1	STL	NHL	D	76	20	22	42	24	-9	11
schmimi01	1951	1	BOS	NHL	C	[illegible]	[illegible]	29	50	57	null	null

Fig. 11: Inspecting Data Samples to Identify Meaningful Unique Constraints

In addition, we may want to know the uniqueness ratio for all the subsets of some keys. Clicking on View subsets in the Browser Panel as illustrated in Fig. 10, will lead us to to another panel with measures for subsets of keys, as shown in Fig. 12. For example, the fact that {year, playerID} holds with 83% uniqueness means that 17% of the records have players that played for different teams within the same year, namely at different stints. Hence, the ability to inspect data samples and measures for subsets of mined unique constraints enables users to understand the underlying data and business keys that govern entity integrity. In the Browser Panel it is further possible to select the unique constraints of interest to the user and download them. The file itself may be used as additional input to the relationship discovery algorithm for more targeted and efficient results.



Hockey.Scoring [playerID,year,stint]

Minimum completeness 0

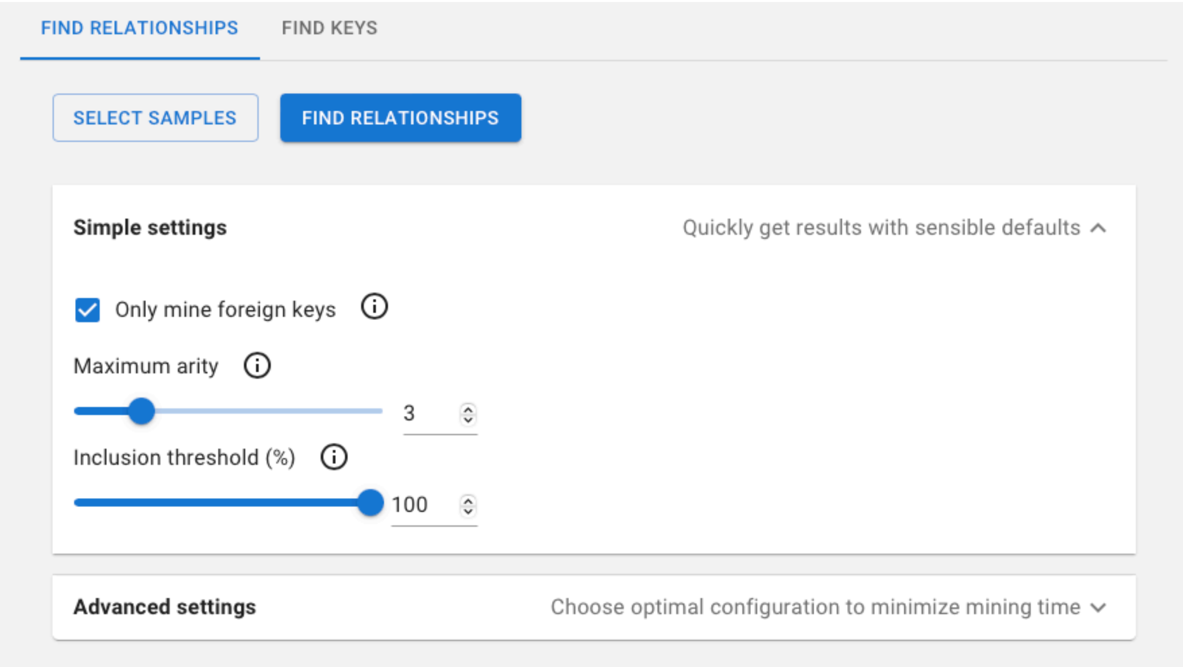
Minimum uniqueness 0

Example data	Table	Columns	Uniqueness	Completeness
View data	Hockey.Scoring	year, playerID	85%	100%
View data	Hockey.Scoring	stint	<1%	100%
View data	Hockey.Scoring	stint, playerID	7%	100%
View data	Hockey.Scoring	year	<1%	100%
View data	Hockey.Scoring	stint, year	<1%	100%
View data	Hockey.Scoring	playerID	3%	100%

Fig. 12: Uniqueness and Completeness for all Key Subsets

2.4 Finding and browsing relationships

Similar to finding different kinds of keys, the DataViadotto Profilers enables users to find relationships as well. Computationally, this is an even harder problems since we need to search through sequences of fields across different tables. The process of finding relationships is similar to that of keys. First, users select some parameters that let them decide which kinds of relationships they are interested in. The left of bottom left of Fig. 13 shows the default values in the simple settings panel. By default, the profiler finds only foreign keys of maximum arity 3 and with a (partial) inclusion threshold of 100%. Here, partial refers to one of the three semantics the SQL standard offers for the interpretation of missing values. Partial means that every record of the referencing table must have a partial match in some record of the referenced table. Hence, records with null marker occurrences in their foreign key fields still need to have partial matches.



Simple settings Quickly get results with sensible defaults

☑ Only mine foreign keys ⓘ

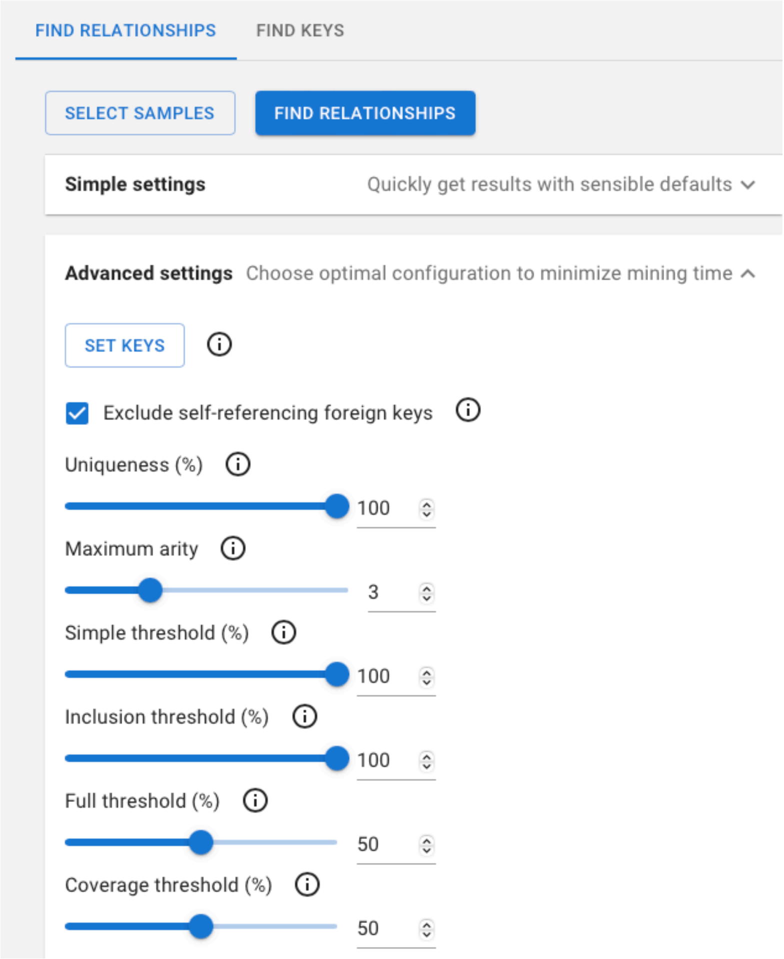
Maximum arity ⓘ 3

Inclusion threshold (%) ⓘ 100

Advanced settings Choose optimal configuration to minimize mining time

Fig. 13: Simple Settings to Select Parameters for Finding Relationships

Under Advanced Settings, illustrated in Fig. 14, users can adjust various metrics, such as the uniqueness ratio required for the referenced sequence of fields, the maximum number of fields (arity) a relationship may have that the algorithm is looking for, a threshold for each of the three semantics associated with null marker occurrences (simple, partial, full) – that is, which ratio of records must have a corresponding match, as well as the coverage threshold that is the ratio of records from the referenced table that are actually referenced. Each of the parameters has a strong impact on the number of results and the time it takes to generate them. The thresholds, in particular, determine which percentage of records can offend referential integrity while still being considered as a candidate relationship in the output of our algorithm. As a consequence, users can still find meaningful relationships even if they are violated by the given data sets. After the samples and parameters have been selected for mining, the algorithms run. In the case of our running example, it takes about 25 seconds with all default values and across all of the 22 tables to return 47 foreign keys. If we use some curated list of mined keys from before, then it will take less than 3 seconds to mine all foreign keys that reference any of these keys, and 17 foreign keys will be returned.



☑ Exclude self-referencing foreign keys

Uniqueness (%) 100

Maximum arity 3

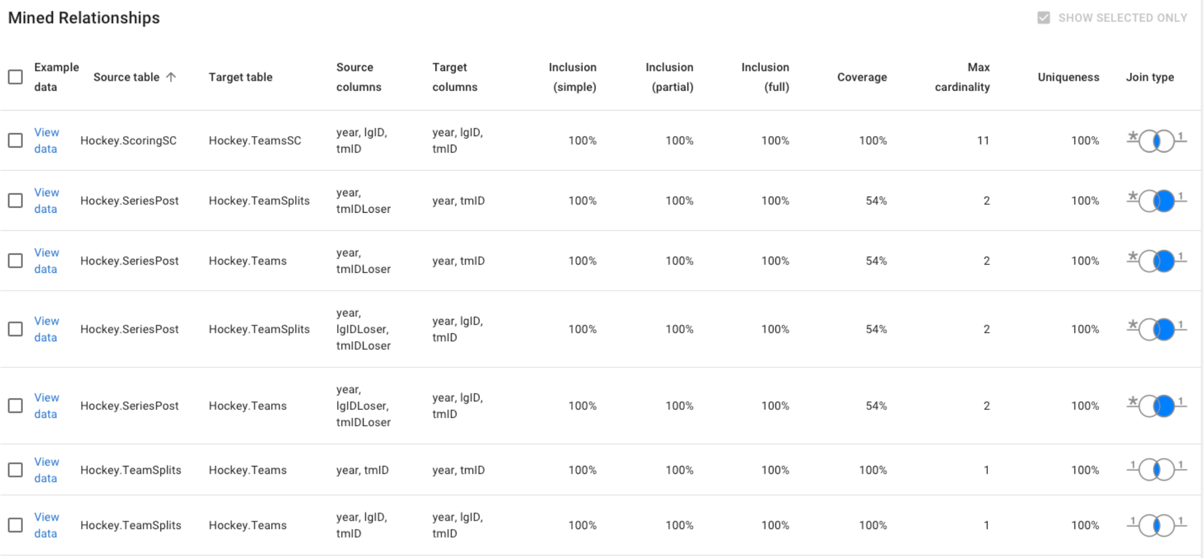
Simple threshold (%) 100

Inclusion threshold (%) 100

Full threshold (%) 50

Coverage threshold (%) 50

Fig. 14: Advanced Settings to Select Parameters for Finding Relationships



Mined Relationships

☑ SHOW SELECTED ONLY

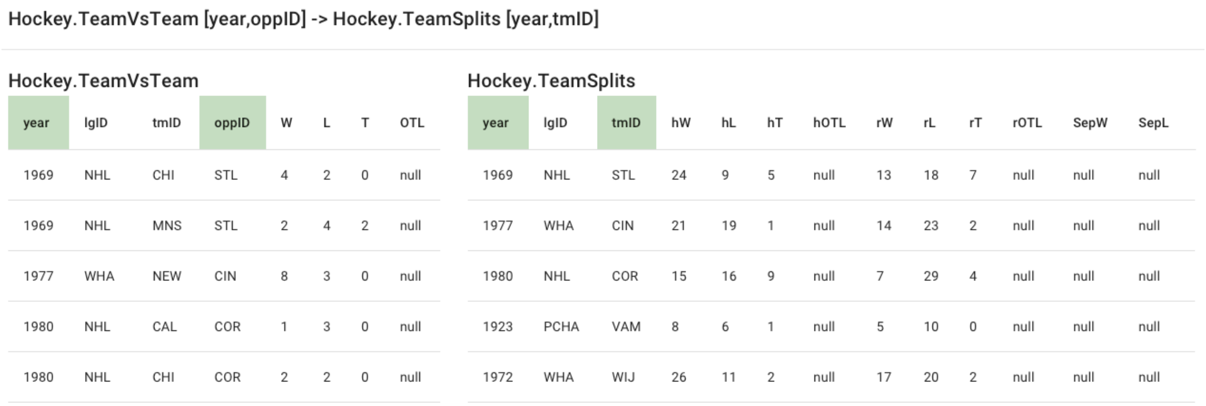
☐ Example data	Source table ↑	Target table	Source columns	Target columns	Inclusion (simple)	Inclusion (partial)	Inclusion (full)	Coverage	Max cardinality	Uniqueness	Join type
☐ View data	Hockey.ScoringSC	Hockey.TeamsSC	year, lgID, tmID	year, lgID, tmID	100%	100%	100%	100%	11	100%	* ∩ 1
☐ View data	Hockey.SeriesPost	Hockey.TeamSplits	year, tmIDLoser	year, tmID	100%	100%	100%	54%	2	100%	* ○● 1
☐ View data	Hockey.SeriesPost	Hockey.Teams	year, tmIDLoser	year, tmID	100%	100%	100%	54%	2	100%	* ○● 1
☐ View data	Hockey.SeriesPost	Hockey.TeamSplits	year, lgIDLoser, tmIDLoser	year, lgID, tmID	100%	100%	100%	54%	2	100%	* ○● 1
☐ View data	Hockey.SeriesPost	Hockey.Teams	year, lgIDLoser, tmIDLoser	year, lgID, tmID	100%	100%	100%	54%	2	100%	* ○● 1
☐ View data	Hockey.TeamSplits	Hockey.Teams	year, tmID	year, tmID	100%	100%	100%	100%	1	100%	1 ∩ 1
☐ View data	Hockey.TeamSplits	Hockey.Teams	year, lgID, tmID	year, lgID, tmID	100%	100%	100%	100%	1	100%	1 ∩ 1

Fig. 15: Browser Panel for Inspecting Mined Relationships and Their Metrics

Fig. 15 shows the Browser Panel for Inspecting some of the Relationships that have been mined from the given data sets, here the 22 Hockey tables. Apart from the Source and Target tables, and the corresponding sequences of source and target fields, each relationship comes with the value for the various metrics of simple, partial, full inclusion, and coverage by which it holds on the data sets. In addition, we list the maximum number of records in the source table that reference any given record from the target table, as well as the uniqueness ratio of the target columns in the target table. Finally, we list the join type of the relationship, such as many-to-many, many-to-one, one-to-many, or one-to-one relationships for inner, left-outer, right-outer of full-outer joins.

Similar to the data samples for mined keys, users can also inspect data samples for mined relationships, the difference being that we have linked samples over the source and target table. Unless the relationships hold with 100%, the samples do include records that do not have a match. Such records are marked with a red background color and may appear in both the source (no match in target table) and target table (no match in source table). Fig. 16 shows a screenshot for such a data sample, illustrating that each year a team plays as an opposition in the TeamVsTeam table, a corresponding team exists for the same year in the TeamSplits table (that is, every opponent is a team every year). Note that not all records are shown, but the fact they are not marked in red means they each have matches in the fields marked green. Again, the inspection of such sample data clearly facilitates the understanding of a user for the domain, which rules may represent a meaningful rule, and where violations of entity or referential integrity occur.

It is important for users to experiment with different parameters to understand their impact. While choosing high values for the parameters ensures a higher validity of the relationships that are returned and a quicker search, conducting discovery with lower values may return meaningful relationships with larger numbers of records that violate referential integrity. This also provides users with a better understanding of the quality the given data sets exhibit.



Hockey.TeamVsTeam [year,oppID] -> Hockey.TeamSplits [year,tmID]

Hockey.TeamVsTeam

year	lgID	tmID	oppID	W	L	T	OTL
1969	NHL	CHI	STL	4	2	0	null
1969	NHL	MNS	STL	2	4	2	null
1977	WHA	NEW	CIN	8	3	0	null
1980	NHL	CAL	COR	1	3	0	null
1980	NHL	CHI	COR	2	2	0	null

Hockey.TeamSplits

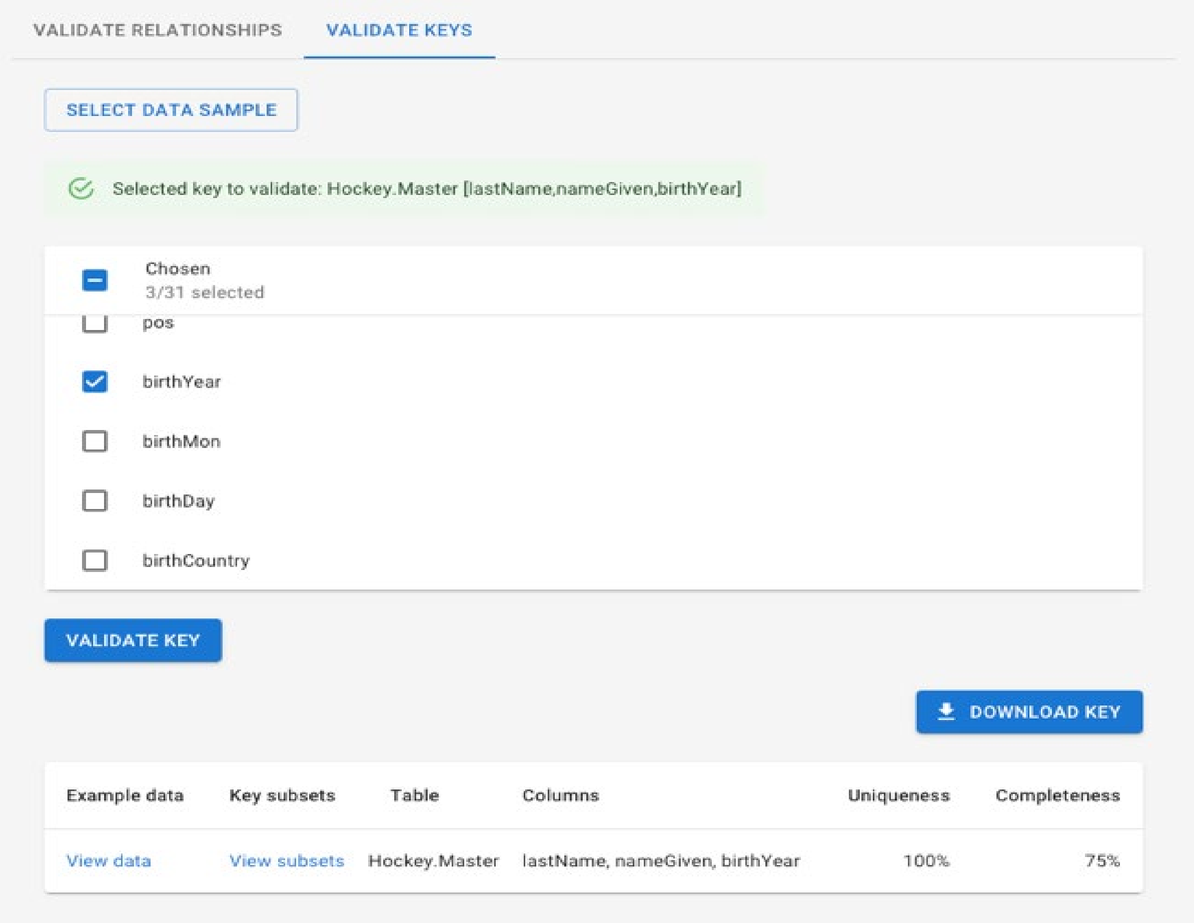
year	lgID	tmID	hW	hL	hT	hOTL	rW	rL	rT	rOTL	SepW	SepL
1969	NHL	STL	24	9	5	null	13	18	7	null	null	null
1977	WHA	CIN	21	19	1	null	14	23	2	null	null	null
1980	NHL	COR	15	16	9	null	7	29	4	null	null	null
1923	PCHA	VAM	8	6	1	null	5	10	0	null	null	null
1972	WHA	WIJ	26	11	2	null	17	20	2	null	null	null

Fig. 16: Sample Data for Inspecting Candidate Relationships

2.5 Validating keys and foreign keys

As the final feature of DataViadotto Profiler, we show how users can validate a particular key or relationship of their own choosing. This feature is also called key or foreign key analysis by other tools and is useful whenever a user want to analyse a specific constraint, such as the validity of a known key or foreign key after it has been turned off while processing transactions.

Starting with keys, the validation panel is shown in Fig. 17, where a user needs to select a sample for a given target table, and then the fields of the key for validation. The bottom of Fig. 17 shows that {lastName, nameGiven, birthYear} form a unique constraint that can identify every record of the Master table with non-null marker occurrences in any of these fields, which make up 75% of all records in this table.



Selected key to validate: Hockey.Master [lastName,nameGiven,birthYear]

Chosen
3/31 selected

☐ pos

☑ birthYear

☐ birthMon

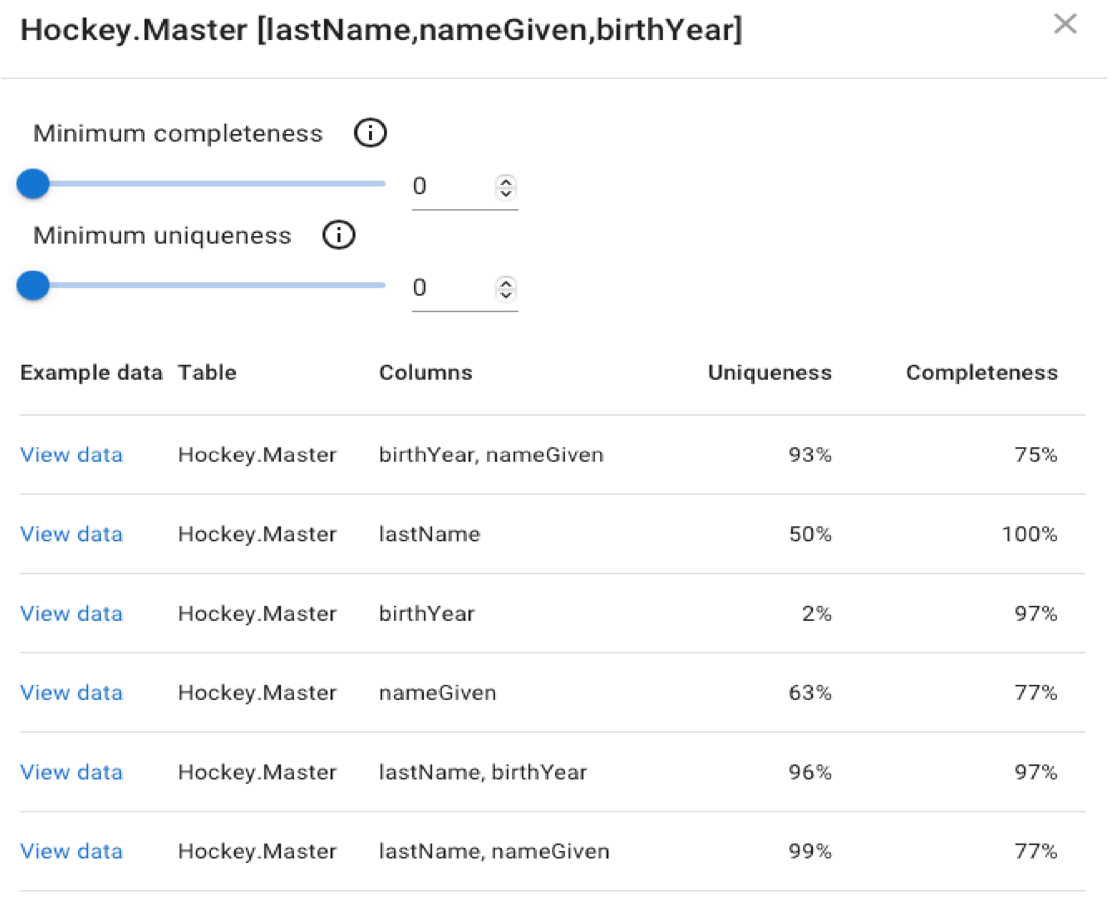
☐ birthDay

☐ birthCountry

Example data	Key subsets	Table	Columns	Uniqueness	Completeness
View data	View subsets	Hockey.Master	lastName, nameGiven, birthYear	100%	75%

Fig. 17: Selecting Target Table and Key for Validation

Fig. 18 shows the panel that opens up with we click View subsets, and we can see that some of the subsets form constraints with high ratios of uniqueness, such as {lastName, nameGiven} or {lastName, birthYear}. For the former, we may then click on View data to bring up Fig. 19 which shows more sample data.



Hockey.Master [lastName,nameGiven,birthYear]

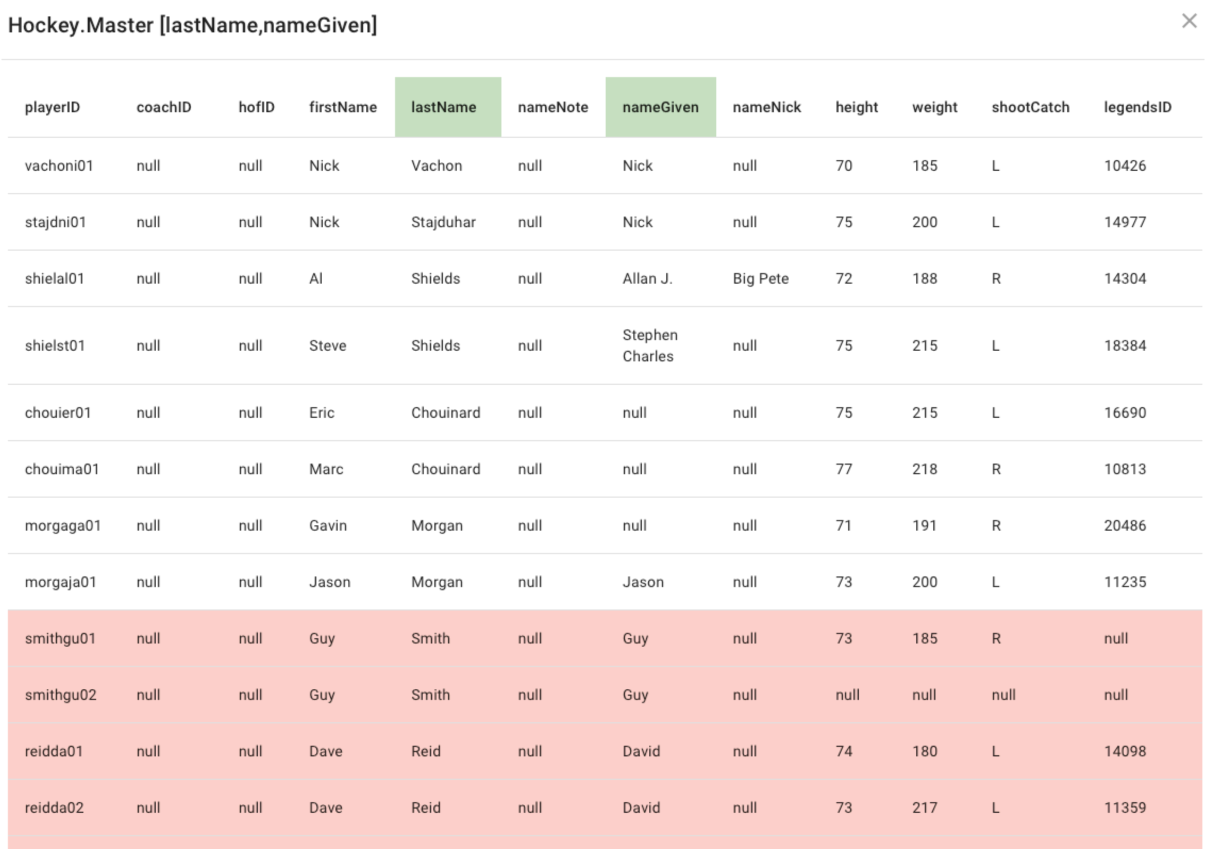
Minimum completeness 0

Minimum uniqueness 0

Example data	Table	Columns	Uniqueness	Completeness
View data	Hockey.Master	birthYear, nameGiven	93%	75%
View data	Hockey.Master	lastName	50%	100%
View data	Hockey.Master	birthYear	2%	97%
View data	Hockey.Master	nameGiven	63%	77%
View data	Hockey.Master	lastName, birthYear	96%	97%
View data	Hockey.Master	lastName, nameGiven	99%	77%

Fig. 18: Viewing Subsets of Target Key

Indeed, while the last and second to last record in red appear to represent different people with the same name of David Reid, as evidenced by different heights, weights and legendsID, the third record from the bottom is subsumed by the fourth record from the bottom, in the sense that no non-matching non-null values appear in any of the fields apart from playerID. In fact, the record with smithgu02 under playerID appears to be a duplicate record of the that with smithgu01 under playerID. Note that the data sets have been curated for decades, but we are still able with our analysis to find evidence for duplicate records.



Hockey.Master [lastName,nameGiven]

playerID	coachID	hofID	firstName	lastName	nameNote	nameGiven	nameNick	height	weight	shootCatch	legendsID
vachoni01	null	null	Nick	Vachon	null	Nick	null	70	185	L	10426
stajdni01	null	null	Nick	Stajduhar	null	Nick	null	75	200	L	14977
shielal01	null	null	Al	Shields	null	Allan J.	Big Pete	72	188	R	14304
shielst01	null	null	Steve	Shields	null	Stephen Charles	null	75	215	L	18384
chouier01	null	null	Eric	Chouinard	null	null	null	75	215	L	16690
chouima01	null	null	Marc	Chouinard	null	null	null	77	218	R	10813
morgaga01	null	null	Gavin	Morgan	null	null	null	71	191	R	20486
morgaja01	null	null	Jason	Morgan	null	Jason	null	73	200	L	11235
smithgu01	null	null	Guy	Smith	null	Guy	null	73	185	R	null
smithgu02	null	null	Guy	Smith	null	Guy	null	null	null	null	null
reidda01	null	null	Dave	Reid	null	David	null	74	180	L	14098
reidda02	null	null	Dave	Reid	null	David	null	73	217	L	11359

Fig. 19: Sample Data with Unique Records (white background) and Duplicate Records (red background)

Likewise, user may attempt to impute missing values by employing crowd-sourcing for entities that can be uniquely identified by business keys. As an example, consider the following full sequence of field names for the table called Master, with our example business key {lastName, nameGiven, birthYear} highlighted in bold font:

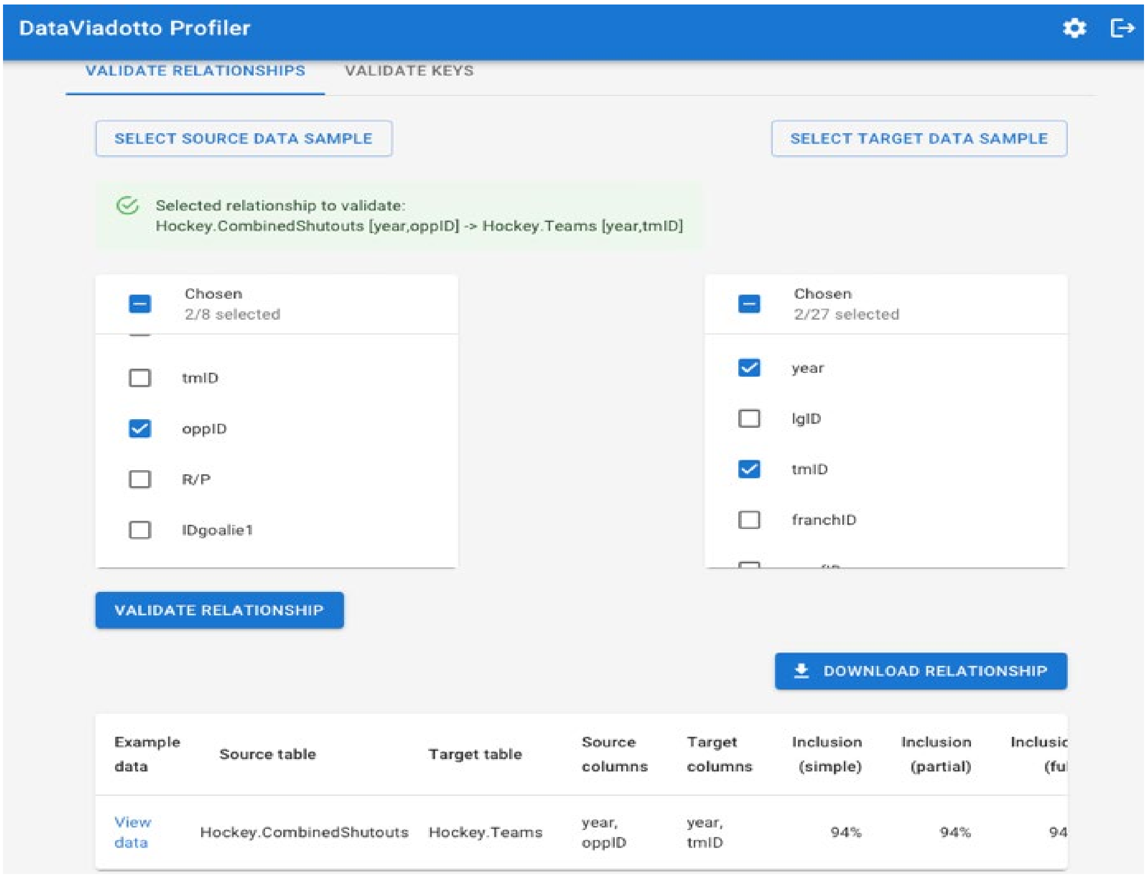
playerID,coachID,hofID,firstName,**lastName**,nameNote,**nameGiven**,nameNick,height,weight,shootCatch,legendsID,ihdbID,hrefID,firstNHL,lastNHL,firstWHA,lastWHA,pos,**birthYear**,birthMon,birthDay,birthCountry,birthState,birthCity,deathYear,deathMon,deathDay,deathCountry,deathState,deathCity

and the following record with unique key values highlighted in bold font, too.

anderjo03,NULL,NULL,Jocko,**Anderson**,NULL,**"John Wilberforce"**, NULL, 67, 150, L,NULL,60142, NULL, NULL, NULL, NULL, NULL, L, **1892**, 10, 4, Canada, MB, "Dynevor St. Peters", 1960 (NULL), 7(NULL), 22(NULL), NULL, NULL, NULL.

The field names marked in red color are fields with null marker occurrences in the original record that have been imputed by the values in red using the corresponding Wikipedia page, which is possible due to the business key {lastName, nameGiven, birthYear}.

We will now turn to the validation of referential constraints, such as foreign keys. The Validate Relationships panel is illustrated on our running example in Fig. 20. For relationships, users need to select a source and target data sample, and a corresponding sequence of field names from each of the source and target tables. The bottom of the panel then shows the candidate relationship together with the metrics associated with it, as we have already seen from the Browser panel.



Selected relationship to validate:
Hockey.CombinedShutouts [year,oppID] -> Hockey.Teams [year,tmID]

Chosen
2/8 selected

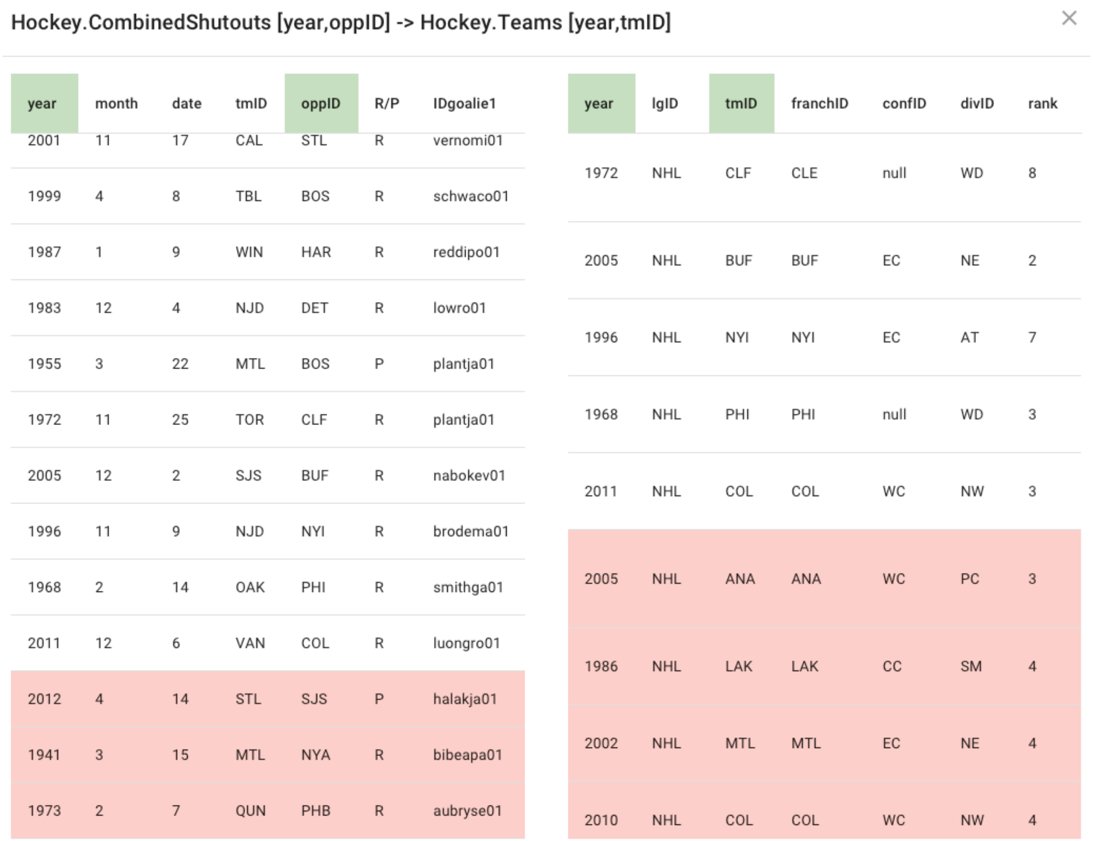
□ tmID
☑ oppID
□ R/P
□ IDgoalie1

Chosen
2/27 selected

☑ year
□ lgID
☑ tmID
□ franchID

Example data	Source table	Target table	Source columns	Target columns	Inclusion (simple)	Inclusion (partial)	Inclusic (fu
View data	Hockey.CombinedShutouts	Hockey.Teams	year, oppID	year, tmID	94%	94%	94

Fig. 20: Selecting Tables and Fields for Validating Relationships



Hockey.CombinedShutouts [year,oppID] -> Hockey.Teams [year,tmID]

year	month	date	tmID	oppID	R/P	IDgoalie1
2001	11	17	CAL	STL	R	vernomi01
1999	4	8	TBL	BOS	R	schwaco01
1987	1	9	WIN	HAR	R	reddipo01
1983	12	4	NJD	DET	R	lowro01
1955	3	22	MTL	BOS	P	plantja01
1972	11	25	TOR	CLF	R	plantja01
2005	12	2	SJS	BUF	R	nabokev01
1996	11	9	NJD	NYI	R	brodema01
1968	2	14	OAK	PHI	R	smithga01
2011	12	6	VAN	COL	R	luongro01
2012	4	14	STL	SJS	P	halakja01
1941	3	15	MTL	NYA	R	bibeapa01
1973	2	7	QUN	PHB	R	aubryse01

year	lgID	tmID	franchID	confID	divID	rank
1972	NHL	CLF	CLE	null	WD	8
2005	NHL	BUF	BUF	EC	NE	2
1996	NHL	NYI	NYI	EC	AT	7
1968	NHL	PHI	PHI	null	WD	3
2011	NHL	COL	COL	WC	NW	3
2005	NHL	ANA	ANA	WC	PC	3
1986	NHL	LAK	LAK	CC	SM	4
2002	NHL	MTL	MTL	EC	NE	4
2010	NHL	COL	COL	WC	NW	4

Fig. 21: Data Sample with Matching and Non-matching Records

Users may then select View data to inspect data samples and understand whether the candidate relationship is meaningful or not. In our example, the inclusion metrics measure 94% of all records that satisfy referential integrity for the candidate relationship. An inspection of the data samples, as illustrated in Fig. 21, shows indeed records from the CombinedShutouts table with no matches in Teams table. This is a clear violation of referential integrity, since every opponent team in any given year should be a team in that year listed in the Teams table. Clearly, this is not the case and this should undergo some curation efforts. We remark that this relationship was incorrectly not specified on the original database, therefore resulting in 6% violation of referential integrity over the years.

3 RESULTS AND IMPACT

We conclude our demonstration of the general Entity/Relationship Profiling process on our running example with a detailed discussion of results that have come out of our analysis after inspecting the results of the Profiling process. As will be seen below, the analysis will result in conceptual and logical models, as well as information for a data catalogue that has clearly not been possible for the previous decades in which the database has been used in public.

Based on the profiling results and smart data samples, the DataViadotto Profiler provides an engaging platform for its users to comprehend which business entities are represented in which data assets, how they are represented, and what their relationships are across these assets. We will now illustrate the impact of using the Profiler on the Hockey Data Set.

Firstly, each of the 13 primary keys that have been specified on Hockey tables exhibit 100% uniqueness and completeness. For this reason, we do not need to discuss them further, and simply refer to Fig. 1.

The known referential constraints from the Hockey schema are listed in Tab. 1 below, together with our now familiar metrics, including the best available join type for each of them.

Source Table	Target Table	Source columns	Target columns	Inclusion (Simple)	Inclusion (Partial)	Inclusion (Full)	Coverage	Max cardinality	Uniqueness	Join type
AwardsPlayers	Master	playerID	playerID	100%	100%	100%	8%	51	100%	* 1
CombinedShutout	Master	IDgoalie1	playerID	100%	100%	100%	<1%	5	100%	* 1
CombinedShutout	Master	IDgoalie2	playerID	100%	100%	100%	<1%	3	100%	* 1
Goalies	Master	playerID	playerID	100%	100%	100%	10%	22	100%	* 1
GoaliesSC	Master	playerID	playerID	100%	100%	100%	<1%	7	100%	* 1
GoaliesShootout	Master	playerID	playerID	100%	100%	100%	1%	9	100%	* 1
Scoring	Master	playerID	playerID	100%	100%	100%	96%	25	100%	* 1
ScoringSC	Master	playerID	playerID	100%	100%	100%	1%	7	100%	* 1
ScoringShootout	Master	playerID	playerID	100%	100%	100%	8%	10	100%	* 1
ScoringSup	Master	playerID	playerID	100%	100%	100%	1%	3	100%	* 1
Coaches	Teams	year,tmID	year,tmID	100%	100%	100%	99%	3	100%	* 1
Goalies	Teams	year,tmID	year,tmID	100%	100%	100%	100%	7	100%	* 1
GoaliesSC	Teams	year,tmID	year,tmID	100%	100%	100%	1%	2	100%	* 1
GoaliesShootout	Teams	year,tmID	year,tmID	100%	100%	100%	13%	6	100%	* 1
Scoring	Teams	year,tmID	year,tmID	100%	100%	100%	99%	22	100%	* 1
ScoringSC	Teams	year,tmID	year,tmID	100%	100%	100%	1%	11	100%	* 1
ScoringShootout	Teams	year,tmID	year,tmID	100%	100%	100%	13%	17	100%	* 1
SeriesPost	Teams	year,tmIDWinner	year,tmID	100%	100%	100%	30%	4	100%	* 1
SeriesPost	Teams	year,tmIDLoser	year,tmID	100%	100%	100%	54%	2	100%	* 1
TeamSC	Teams	year,tmID	year,tmID	100%	100%	100%	1%	1	100%	1 1
TeamsHalf	Teams	year,tmID	year,tmID	100%	100%	100%	1%	2	100%	* 1
TeamSplits	Teams	year,tmID	year,tmID	100%	100%	100%	100%	1	100%	1 1
TeamsPost	Teams	year,tmID	year,tmID	100%	100%	100%	61%	1	100%	1 1
TeamVsTeam	Teams	year,tmID	year,tmID	100%	100%	100%	100%	29	100%	* 1
TeamVsTeam	Teams	year,oppID	year,tmID	100%	100%	100%	100%	29	100%	* 1
AwardsCoaches	Coaches	coachID	coachID	100%	100%	100%	30%	9	5%	* *
Master	Coaches	coachID	coachID	100%	100%	5%	100%	1	5%	1 *

Tab 1: Existing Referential Constraints on the Hockey Schema together with their Metrics

All three inclusion metrics (simple, partial, full) apply to all records from the source tables, except for the last constraint and full semantics. Indeed, the direction of the contraint is incorrect, since it should say that the coachID of every coach is listed as the coachID in the Master table, similar to all other types of people. This is further confirmed by the join type, in particular the cardinality and uniqueness metric, indicating there is a many-to-one foreign key from the Coaches to the Master table, as coachID is unique on Master but not on Coaches.

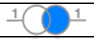
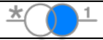
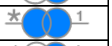
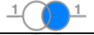
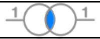
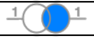
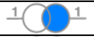
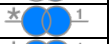
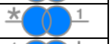
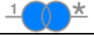
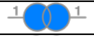
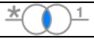
Hence, based on the constraints that have already been specified on the underlying schema, most things appear to be conformant (The incorrect foreign key is, of course, not.) This could have indeed been confirmed by using key and foreign key analysis available in other tools.

However, the real value of Entity/Relationship Profiling comes from its ability to point at other keys and referential constraints that have been overlooked, in this case for decades. As examples, we have listed various uniqueness constraints in Tab. 2 we think would constitute valuable additions to Hockey tables. For tables in bold font, no uniqueness constraints had been specified originally. We propose 35 new uniqueness constraints, with 24 of them being candidate keys. These constitute 35 new way of identifying business entities uniquely.

Table	**Columns**	**Uniqueness**	**Completeness**	**Type**
Master	playerID	100%	96%	Uniqueness constraint
	hrefID	100%	96%	Uniqueness constraint
	ihdbID	100%	91%	Uniqueness constraint
	legendsID	100%	84%	Uniqueness constraint
	lastName, nameGiven, birthYear	100%	75%	Uniqueness constraint
	coachID	100%	5%	Uniqueness constraint
	hofID	100%	4%	Uniqueness constraint
Teams	year, franchID	100%	100%	Candidate key
	year, name	100%	100%	Candidate key
	year, divID, rank	100%	77%	Uniqueness constraint
Abbrev	Fullname	100%	100%	Candidate key
AwardsMisc	ID	100%	68%	Uniqueness constraint
HOF	year, name	100%	100%	Candidate key
AwardsCoaches	coachID, year	100%	100%	Candidate key
	award, year	100%	100%	Candidate key
Coaches	coachID, year, stint	100%	100%	Candidate key
ScoringSup	playerID, year	100%	100%	Candidate key
GoaliesShootout	playerID, year, tmID	100%	100%	Candidate key
	playerID, year, stint	100%	100%	Candidate key
Combined Shutouts	year, month, date, tmID	100%	100%	Candidate key
	year, date	100%	100%	Candidate key
	date, tmID	100%	100%	Candidate key
	year, month, tmID	100%	100%	Candidate key
	month, tmID, oppID	100%	100%	Candidate key
ScoringShootout	playerID, year, tmID	100%	100%	Candidate key
	playerID, year, stint	100%	100%	Candidate key
Scoring	playerID, year, stint, tmID	100%	100%	Candidate key
	playerID, year, stint, pos	100%	98%	Uniqueness constraint
ScoringSC	playerID, year	100%	100%	Candidate key
TeamsSC	year, lgID	100%	100%	Candidate key
TeamsHalf	year, half, rank	100%	100%	Candidate key
SeriesPost	year, tmIDWinner, tmIDLoser	100%	100%	Candidate key
	year, round, tmIDWinner	100%	100%	Candidate key
	year, round, tmIDLoser	100%	100%	Candidate key
	year, series	100%	88%	Uniqueness constraint

Tab 2: Proposed Unique Constraints and Candidate Keys to Add to Hockey Tables following Entity Profiling

Similarly, Tab. 3 lists 12 different referential integrity constraints that have emerged as result of our Relationship Profiling exercise on the Hockey tables. Note that each of these does not follow from any constraints specified on the Hockey tables, but constitutes a meaningful constraint that needs to be enforced on the Hockey data to guarantee referential integrity. It appears that some of these constraints do not hold with 100% according to any of the three metrics (simple, partial, full). However, each of them represents meaningful rules and this is further supported by high metrics. Indeed, every record from a source table that has no matching record in the target table violates referential integrity, and represents an opportunity to increase referential integrity, the available join type, which would result in more accurate reporting or predictive analytics. Fig. 20, for example, showed examples of such records based on the fourth referential constraint from the top listed in Tab. 3.

Source Table	Target Table	Source columns	Target columns	Inclusion (Simple)	Inclusion (Partial)	Inclusion (Full)	Coverage	Max cardinality	Uniqueness	Join type
HOF	Master	hofID	hofID	100%	100%	100%	4%	1	100%	
Coaches	Master	coachID	coachID	100%	100%	100%	5%	31	100%	
CombinedShutouts	Teams	year,tmID	year,tmID	98%	98%	98%	3%	2	100%	
CombinedShutouts	Teams	year,oppID	year,tmID	94%	94%	94%	3%	1	100%	
Teams	TeamSplits	year,tmID	year,tmID	100%	100%	100%	100%	1	100%	
GoaliesSC	ScoringSC	playerID,year,tmID	playerID,year,tmID	100%	100%	100%	10%	1	100%	
GoaliesShootout	Goalies	playerID,year,stint,tmID	playerID,year,stint,tmID	100%	100%	100%	11%	1	100%	
SeriesPost	TeamsPost	year,tmIDWinner	year,tmID	99%	99%	99%	50%	4	100%	
SeriesPost	TeamsPost	year,tmIDLoser	year,tmID	99%	99%	99%	88%	2	100%	
AwardsCoaches	Coaches	coachID,year	coachID,year	98%	98%	98%	4%	1	98%	
ScoringShootout	Scoring	playerID,year,stint,tmID	playerID,year,stint,tmID	99%	99%	99%	4%	1	100%	
ScoringSC	TeamsSC	year,tmID	year,tmID	100%	100%	100%	100%	11	100%	

Tab 3: Proposed Referential Constraints To Add Across Hockey Tables following Relationship Profiling

We may now return to the original conceptual diagram of the Hockey data set, shown in Fig. 1, and apply our insight from Entity/Relationship Profiling to it. The revised conceptual diagram is shown in Fig. 21 using different color codings:

- Everything in black has not undergone any changes compared to the original diagram
- Everything in orange is new information, in the form of either unique constraints and candidate keys, or referential constraints.
- Referential constraints in blue are from the original diagram, but become redundant after introducing some of the orange referential constraints.
- The red referential constraint needs to be removed as it does not constitute a meaningful rule.

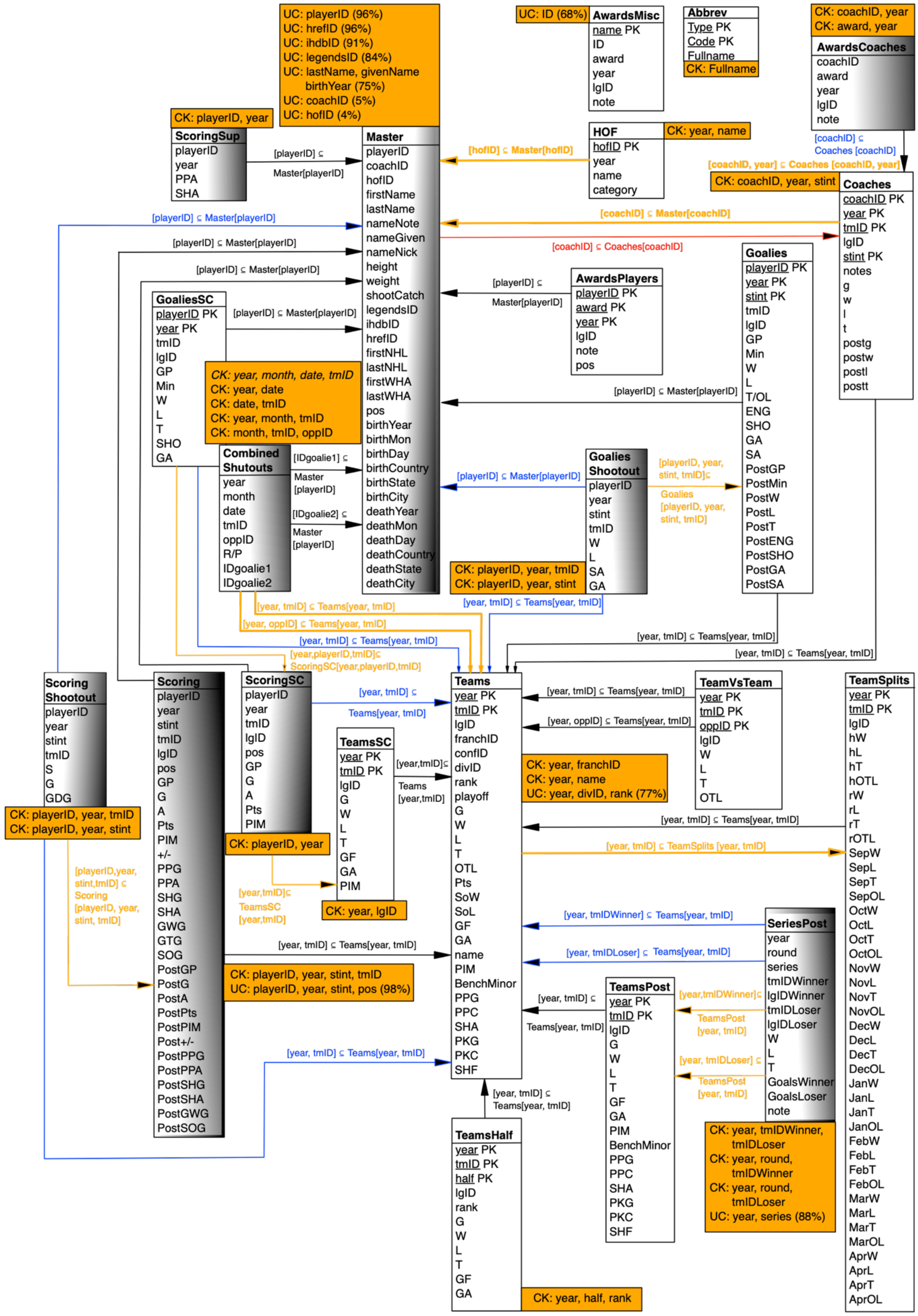


Fig. 22: Revised Conceptual Diagram for Hockey Data Set following Entity/Relationship Profiling

For the data steward, Fig. 22 constitutes an invaluable high-level view of all the data sets available in this project, and how they are connected. Indeed, the constraints form important part of the data catalogue.

For the data architect, Fig. 22 constitutes a blueprint for deriving a logical model of data from the diagram, or deriving further data models for more specific projects.

For the data engineer, Fig. 22 summarizes all the constraints that need to be added to the tables and monitored. The unique constraints will result automatically in unique indexes, speeding up database operations for other users, such as the analyst or scientist. In addition, any violations of these constraints constitute opportunities for identifying and cleaning up data inconsistencies or imputing missing values, as illustrated beforehand.

For the data analyst and scientists, the changes in models mean better and faster access, reporting and prediction can be done with higher accuracy, more transparency by the use of business keys rather than identifiers, and higher speed.

Note that all these benefits emerge even without making any changes to the underlying design for any of the individual tables.

4 CLOSING

In summary, we have showcased the process and benefits of Entity/Relationship Profiling with the DataViadotto Profiler. Its unique features across all data profiling tools make data profitable and lift any data-related role to new levels of insight, effectiveness and efficiency. In choosing the DataViadotto Profiler for your organization, you will enable staff to understand data better and faster, make the most of your data assets, and bring data-driven decision making to life.

ABOUT DATAVIADOTTO

DataViadotto is the industry pioneer for Entity/Relationship profiling technology. The company draws on decades of academic research in the subject to make the process of discovering models from data more effective, efficient and intuitive. Ultimately, data becomes profitable.

FOR ADDITIONAL QUESTIONS, CONTACT DATAVIADOTTO

www.viadotto.tech