

# 1 Validátory

Vzhledem k některým guardům, které nemapují entity, pokud je model v nekonzistentním stavu jsme se rozhodli pro vytvoření validačního query, které zkontroluje stav modelu. Toto query se dá použít na vstupní i výstupní model.

## 2 Validátor aplikačního modelu

Nekonzistentní stavy modelu se dělí do těchto kategorií:

- duplicitní jména(Class, property, ...)
- hierarchie dědičnosti
- primitivní typy
- vazby
- embeddedClass

### 2.1 Duplicitní jména

Neexistuje Class v generaci se jménem shodným s jménem jiné třídy v téže generaci

Neexistuje property ve třídě se shodným jménem s property téže třídy nebo jejího předka - zamezujeme překrývání vyobrazeném na obrázku 1, ačkoli toto je v Jave možné.

### 2.2 hierarchie dědičnosti

Neexistuje třída, která by sobě samé byla předkem ( v hierarchii nevzniká cyklus)

Třída je embedded nebo je primitive nebo má parenta nebo má právě jednu property s atributem isID nastaveným na true.

### 2.3 Primitivní třídy

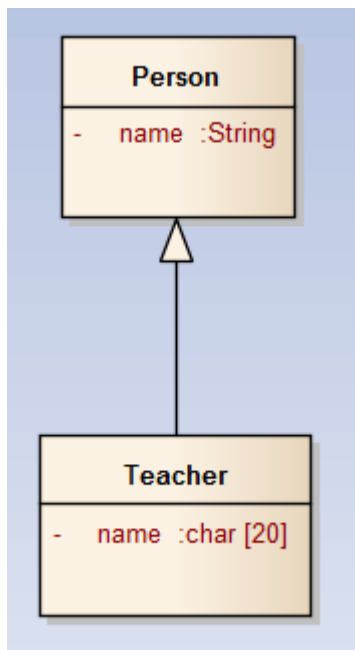
Neexistuje primitivní třída, která by měla id

### 2.4 Property + vazby

Neexistuje třída, která by měla property neprimitivního typu, jejíž opposite property by nebyla nastavená původní property. Všechny primitivní Property mají nastavenou oppositeProperty na null

Neexistuje property nelistové třídy (nelistová třída = třída mající potomka) mapovanými přes InheritanceType TablePerClass a upperBound vyšší než 1.

Neexistuje primitivní property s arititou 1..N , kde  $N \geq 1$ , která by měla stejné jméno jako nějaká třída v generaci. Pozn. jde o mapování kolekcí, které potřebují vlastní tabulku.



Obrázek 2: Překrývání atributů tříd

## 2.5 EmbeddedClass

Neexistuje embeddedClass v generaci, která by měla property jiné arity než 0 ... 1 x 1 nebo 1 x 1

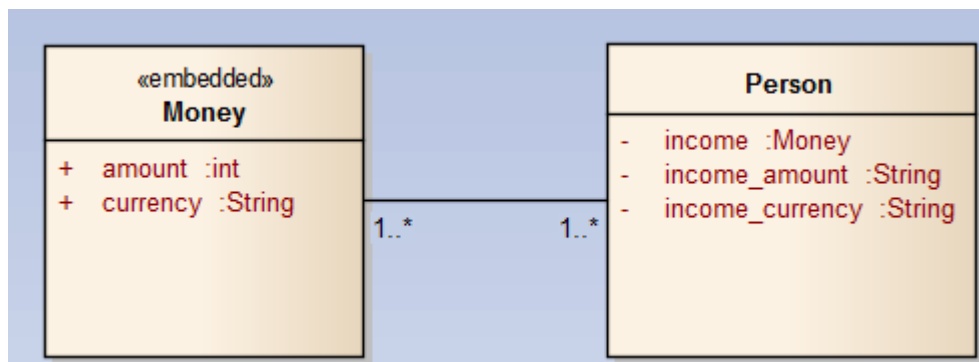
V rámci třídy neexistuje Property v hierarchii, která by měla name shodné s Stringem reprezentujícím property typu EmbeddedTřída. Poznámka: reprezentativní Stringy property se rovnají:  
`property.name + "_" + property.type.property[i].name`

Na obr. 3 vidíme příklad takové kolize jmen. Reprezentativními Stringy jsou "income\_amount" a "income\_currency", které jsou v kolizi s jménem property třídy Person. Druhou chybou je multiplicita M x N.

## 3 Validator DB modelu

### 3.1 Case Sentivity

SQL není case sensitive jazyk, takže kolize jmen v DB modelu nastává i v případě neshodných identifikátorů např. "lambda" a "LaMBDA". Nynější implementace povoluje vytvoření modelu s velkými i malými písmeny.



Obrázek 4: EmbeddedClass kolize

### 3.2 Identifikátory

Podle dokumentace PostgreSQL viz [1] a [2] každý SQL identifikátor v PostgreSQL je může obsahovat písmena (a-z), pomlčku -, diakritické znaky, písmena jiné než latinské abecedy, číslice, symbol dolar(#) nebo podtržítka. Dále pak každý identifikátor musí začínat jedním z písmen (a-z) a nesmí být rezervované klíčové slovo či nerezervované slovo v kontextu.

Pro zjednodušení jsme si určili, že každý identifikátor musí začínat písmeny "(a-z)" a kromě těchto písmen může dále pak obsahovat pomlčku (-) a latinské číslice.

Model není validní, pokud není některý vygenerovaný identifikátor validní.

Model není validní, pokud identifikátor koliduje s klíčovým SQL slovem z následujícího seznamu

ABORT, ABSOLUTE, ACCESS, ACTION, ADD, ADMIN, AFTER, AGGREGATE, ALL, ALSO, ALTER, ALWAYS, ANALYSE, ANALYZE, AND, ANY, ARRAY, AS, ASC, ASSERTION, ASSIGNMENT, ASYMMETRIC, AT, AUTHORIZATION, BACKWARD, BEFORE, BEGIN, BETWEEN, BIGINT, BINARY, BIT, BOOLEAN, BOTH, BY, CACHE, CALLED, CASCADE, CASCADED, CASE, CAST, CATALOG, CHAIN, CHAR, CHARACTER, CHARACTERISTICS, CHECK, CHECKPOINT, CLASS, CLOSE, CLUSTER, COALESCE, COLLATE, COLUMN, COMMENT, COMMENTS, COMMIT, COMMITTED, CONCURRENTLY, CONFIGURATION, CONNECTION, CONSTRAINT, CONSTRAINTS, CONTENT, CONTINUE, CONVERSION, COPY, COST, CREATE, CREATEDB, CREATEROLE, CREATEUSER, CROSS, CSV, CURRENT, CURRENT\_CATALOG, CURRENT\_DATE, CURRENT\_ROLE, CURRENT\_SCHEMA, CURRENT\_TIME, CURRENT\_TIMESTAMP, CURRENT\_USER, CURSOR, CYCLE, DATA, DATABASE, DAY, DEALLOCATE, DEC, DECIMAL, DECLARE, DEFAULT, DEFAULTS, DEFERRABLE, DE-

FERRED, DEFINER, DELETE, DELIMITER, DELIMITERS, DESC, DICTIONARY, DISABLE, DISCARD, DISTINCT, DO, DOCUMENT, DOMAIN, DOUBLE, DROP, EACH, ELSE, ENABLE, ENCODING, ENCRYPTED, END, ENUM, ESCAPE, EXCEPT, EXCLUDE, EXCLUDING, EXCLUSIVE, EXECUTE, EXISTS, EXPLAIN, EXTERNAL, EXTRACT, FALSE, FAMILY, FETCH, FIRST, FLOAT, FOLLOWING, FOR, FORCE, FOREIGN, FORWARD, FREEZE, FROM, FULL, FUNCTION, FUNCTIONS, GLOBAL, GRANT, GRANTED, GREATEST, GROUP, HANDLER, HAVING, HEADER, HOLD, HOUR, IDENTITY, IF, ILIKE, IMMEDIATE, IMMUTABLE, IMPLICIT, IN, INCLUDING, INCREMENT, INDEX, INDEXES, INHERIT, INHERITS, INITIALLY, INLINE, INNER, INOUT, INPUT, INSENSITIVE, INSERT, INSTEAD, INT, INTEGER, INTERSECT, INTERVAL, INTO, INVOKER, IS, ISNULL, ISOLATION, JOIN, KEY, LANGUAGE, LARGE, LAST, LC\_COLLATE, LC\_CTYPE, LEADING, LEAST, LEFT, LEVEL, LIKE, LIMIT, LISTEN, LOAD, LOCAL, LOCALTIME, LOCALTIMESTAMP, LOCATION, LOCK, LOGIN, MAPPING, MATCH, MAXVALUE, MINUTE, MINVALUE, MODE, MONTH, MOVE, NAME, NAMES, NATIONAL, NATURAL, NCHAR, NEXT, NO, NOCREATEDB, NOCREATEROLE, NOCREATEUSER, NOINHERIT, NOLOGIN, NONE, NOSUPERUSER, NOT, NOTHING, NOTIFY, NOTNULL, NOWAIT, NULL, NULLIF, NULLS, NUMERIC, OBJECT, OF, OFF, OFFSET, OIDS, ON, ONLY, OPERATOR, OPTION, OPTIONS, OR, ORDER, OUT, OUTER, OVER, OVERLAPS, OVERLAY, OWNED, OWNER, PARSER, PARTIAL, PARTITION, PASSWORD, PLACING, PLANS, POSITION, PRECEDING, PRECISION, PREPARE, PREPARED, PRESERVE, PRIMARY, PRIOR, PRIVILEGES, PROCEDURAL, PROCEDURE, QUOTE, RANGE, READ, REAL, REASSIGN, RECHECK, RECURSIVE, REFERENCES, REINDEX, RELATIVE, RELEASE, RENAME, REPEATABLE, REPLACE, REPLICA, RESET, RESTART, RESTRICT, RETURNING, RETURNS, REVOKE, RIGHT, ROLE, ROLLBACK, ROW, ROWS, RULE, SAVEPOINT, SCHEMA, SCROLL, SEARCH, SECOND, SECURITY, SELECT, SEQUENCE, SEQUENCES, SERIALIZABLE, SERVER, SESSION, SESSION\_USER, SET, SETOF, SHARE, SHOW, SIMILAR, SIMPLE, SMALLINT, SOME, STABLE, STANDALONE, START, STATEMENT, STATISTICS, STDIN, STDOUT, STORAGE, STRICT, STRIP, SUBSTRING, SUPERUSER, SYMMETRIC, SYSID, SYSTEM, TABLE, TABLES, TABLESPACE, TEMP, TEMPLATE, TEMPORARY, TEXT, THEN, TIME, TIMESTAMP, TO, TRAILING, TRANSACTION, TREAT, TRIGGER, TRIM, TRUE, TRUNCATE, TRUSTED, TYPE, UNBOUNDED, UNCOMMITTED, UNENCRYPTED, UNION, UNIQUE, UNKNOWN, UNLISTEN, UNTIL, UPDATE, USER, USING, VACUUM, VALID, VALIDATOR, VALUE, VALUES, VARCHAR, VARIADIC, VARYING, VERBOSE, VERSION, VIEW, VOLATILE, WHEN, WHERE, WHITESPACE, WINDOW, WITH, WITHOUT, WORK, WRAPPER, WRITE, XML, XMLATTRIBUTES, XMLCONCAT, XMLELEMENT, XMLFOREST, XMLPARSE, XMLPI, XMLROOT, XMLSERIALIZE, YEAR, YES, ZONE

## Reference

- [1] PostgreSQL community. Identifikátory postgresql, 2011. URL  
[http://www.postgresql.org/docs/9.1/interactive/sql-syntax-lexical.html%  
-SYNTAX-IDENTIFIERS](http://www.postgresql.org/docs/9.1/interactive/sql-syntax-lexical.html%5F-SYNTAX-IDENTIFIERS).
- [2] PostgreSQL community. Klíčová slova dialektu postgresql, 2011. URL  
<http://www.postgresql.org/docs/9.0/interactive/sql-keywords-appendix.html>.