querying with data models

- a crazy idea from the middle of a session...

querying with data models

- use utypes in the SELECT and WHERE clause
- use data model in the FROM clause

querying with data models

- use utypes in the SELECT and WHERE clause
- use data model in the FROM clause

```
SELECT utype2column('foo:some.utype')
FROM model2table('http://www.ivoa.net/FooDM/v1.0') AS foo
```

- make sense conceptually, defines the namespace prefix

querying with data models

- use utypes in the SELECT and WHERE clause
- use data model in the FROM clause

```
SELECT utype2column('foo:some.utype')
FROM model2table('http://www.ivoa.net/FooDM/v1.0') AS foo
```

- make sense conceptually, defines the namespace prefix
- BUT: no table-valued function in ADQL
  - table name or subquery

querying with data models

- another try, this time syntactically legal ADQL:

```
SELECT utype2column('foo:some.utype')
FROM FooDM as foo
```

- FooDM is a logical table (re: table_type in VODataService)
  - logical: supports query by utype
  - model and version is in table metadata

querying with data models

- another try, slightly different syntax:

```
SELECT utype2column(foo, 'some.utype')
FROM FooDM as foo
```

- BUT, parser thinks foo is a column
  - value expression: column, literal, function

querying with data models

- Now what?
  - no need to change or enhance ADQL
  - agree on table_type for logical tables
  - agree on the utype2column UDF
  - 2+ groups with a common data model

  ...
  - wait for TAP dust to settle a bit

  ...
  - prototype it!