The System Catalogs

These slides are mostly taken verbatim, or with minor changes, from those prepared by
Stephen Hegner (http://www.cs.umu.se/ hegner/)
of Umeå University

Data about the Schemata

- A DBMS must store information about attributes, relations, constraints, triggers, indices, and much more.
- This information is stored in a relational database consisting of the *system catalogs*.
- Each such catalog is a relation.
- In PostgreSQL, there are many system catalogs.
- They are found in a schema called `pg_catalog`.
- A list for version 8.4 is presented on the following two slides.

Catalogs of PostgreSQL

- Here are the catalogs of PostgreSQL 8.4:

Catalog Name	Purpose
pg_aggregate	aggregate functions
pg_am	index access methods
pg_amop	access method operators
pg_amproc	access method support procedures
pg_attrdef	column default values
pg_attribute	table columns ("attributes")
pg_authid	authorization identifiers (roles)
pg_auth_members	authorization identifier membership relationships
pg_cast	casts (data type conversions)
pg_class	tables, indexes, sequences, views ("relations")
pg_constraint	check constraints, unique constraints, primary key constraints, foreign key constraints
pg_conversion	encoding conversion information
pg_database	databases within this database cluster
pg_depend	dependencies between database objects
pg_description	descriptions or comments on database objects
pg_enum	enum label and value definitions
pg_foreign_data_wrapper	foreign-data wrapper definitions
pg_foreign_server	foreign server definitions
pg_index	additional index information
pg_inherits	table inheritance hierarchy

Catalogs of PostgreSQL Continued

Catalog Name	Purpose
pg_language	languages for writing functions
pg_largeobject	large objects
pg_listener	asynchronous notification support
pg_namespace	schemas
pg_opclass	access method operator classes
pg_operator	operators
pg_opfamily	access method operator families
pg_pltemplate	template data for procedural languages
pg_proc	functions and procedures
pg_rewrite	query rewrite rules
pg_shdepend	dependencies on shared objects
pg_shdescription	comments on shared objects
pg_statistic	planner statistics
pg_tablespace	tablespaces within this database cluster
pg_trigger	triggers
pg_ts_config	text search configurations
pg_ts_config_map	text search configurations' token mappings
pg_ts_dict	text search dictionaries
pg_ts_parser	text search parsers
pg_ts_template	text search templates
pg_type	data types
pg_user_mapping	mappings of users to foreign servers

Accessing the System Catalogs

- The description of a catalog may be viewed using the \d directive, as as in the case of an ordinary user relation.

```
test=# \d pg_database
    Table "pg_catalog.pg_database"
    Column      |   Type     | Modifiers
----------------+------------+-----------
 datname        | name       | not null
 datdba         | oid        | not null
 encoding       | integer    | not null
 datcollate     | name       | not null
 datctype       | name       | not null
 datistemplate  | boolean    | not null
 datallowconn   | boolean    | not null
 datconnlimit   | integer    | not null
 datlastsysoid  | oid        | not null
 datfrozenxid   | xid        | not null
 dattablespace  | oid        | not null
 datconfig      | text[]     |
 datacl         | aclitem[]  |
Indexes:
    "pg_database_datname_index" UNIQUE, btree (datname), tablespace "pg
␣␣␣␣"pg_database_oid_index"␣UNIQUE,␣btree␣(oid),␣tablespace␣"pg_global"
Triggers:
␣␣␣␣pg_sync_pg_database␣AFTER␣INSERT␣OR␣DELETE␣OR␣UPDATE␣ON␣pg_database
STATEMENT␣EXECUTE␣PROCEDURE␣flatfile_update_trigger()
Tablespace:␣"pg_global"
␣␣
```

Viewing the List of System Catalogs

- The entire list of system catalogs may be viewed using the \dS directive.

```
test=# \dS
                    List of relations
   Schema   |             Name              | Type  |  Owner
------------+-------------------------------+-------+----------
 pg_catalog | pg_aggregate                  | table | postgres
 pg_catalog | pg_am                         | table | postgres
 pg_catalog | pg_amop                       | table | postgres
 pg_catalog | pg_amproc                     | table | postgres
 pg_catalog | pg_attrdef                    | table | postgres
 pg_catalog | pg_attribute                  | table | postgres
 pg_catalog | pg_auth_members               | table | postgres
 pg_catalog | pg_authid                     | table | postgres
 pg_catalog | pg_cast                       | table | postgres
 pg_catalog | pg_class                      | table | postgres
 pg_catalog | pg_constraint                 | table | postgres
 pg_catalog | pg_conversion                 | table | postgres
 pg_catalog | pg_cursors                    | view  | postgres
 pg_catalog | pg_database                   | table | postgres
 pg_catalog | pg_depend                     | table | postgres
 pg_catalog | pg_description                | table | postgres
 pg_catalog | pg_enum                       | table | postgres
 pg_catalog | pg_foreign_data_wrapper       | table | postgres
 pg_catalog | pg_foreign_server             | table | postgres
 pg_catalog | pg_group                      | view  | postgres
 pg_catalog | pg_index                      | table | postgres
 pg_catalog | pg_indexes                    | view  | postgres
   ...
```

Accessing the Descriptions of Individual Relations

- The system catalogs are in the catalog named pg_catalog.
- *Qualified names* may be used, although they are not necessary, since relations whose names begin with pg_ are assumed to be in the catalog pg_catalog.

```
test=# \d pg_catalog.pg_tables
    View "pg_catalog.pg_tables"
   Column    |  Type   | Modifiers
-------------+---------+-----------
 schemaname  | name    |
 tablename   | name    |
 tableowner  | name    |
 tablespace  | name    |
 hasindexes  | boolean |
 hasrules    | boolean |
 hastriggers | boolean |
View definition:
 SELECT n.nspname AS schemaname, c.relname AS tablename,
  pg_get_userbyid(c.relowner) AS tableowner, t.spcname AS tablespace,
  c.relhasindex AS hasindexes, c.relhasrules AS hasrules,
  c.relhastriggers AS hastriggers
   FROM pg_class c
   LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
   LEFT JOIN pg_tablespace t ON t.oid = c.reltablespace
  WHERE c.relkind = 'r'::"char";
```

Querying the System Catalogs

- The system catalogs may also be queried using SQL.

```
test=# select * from pg_catalog.pg_tables where tableowner='hegner';
 schemaname | tablename | tableowner | tablespace | hasindexes | hasrules | hastriggers
------------+-----------+------------+------------+------------+----------+-------------
 public     | airline   | hegner     |            | t          | f        | t
 public     | schedule  | hegner     |            | t          | f        | t
 public     | airport   | hegner     |            | t          | f        | t
 public     | ticket    | hegner     |            | t          | f        | t
 public     | flight    | hegner     |            | t          | f        | t
(5 rows)
```

Catalog Access via ODBC

- The ODBC standard provides a collection of API functions for querying the system catalog.
- They are more limited in overall scope than the directives for PostgreSQL, because they are not specific to a particular DBMS.
- The principal ones are the following:

```
SQLTables()
SQLTablePrivileges()
SQLColumns()
SQLColumnPrivileges()
SQLSpecialColumns()
SQLStatistics()
SQLPrimaryKeys()
SQLForeignKeys()
SQLProcedures()
SQLProcedureColumns()
```

- Use is relatively straightforward, but involves the detail which is typically associated with ODBC, and will not be elaborated in these slides.