

# PostgreSQL 数据库管理

# Objectives



PolarDB



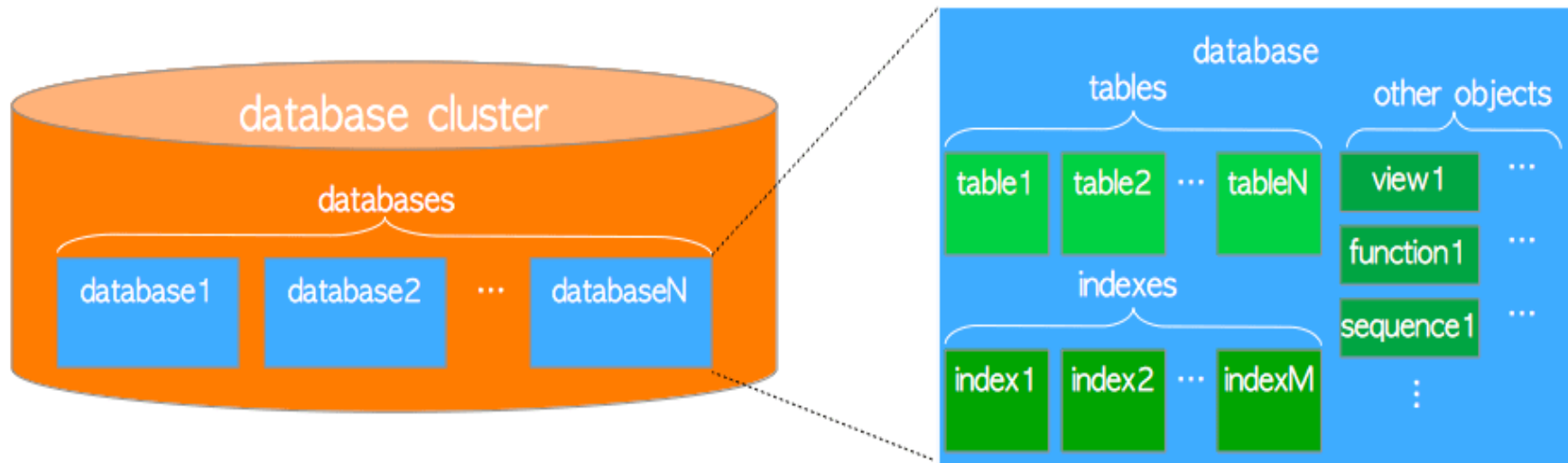
PostgreSQL

- PostgreSQL数据库结构
- PostgreSQL数据库级权限管理
- PostgreSQL数据库级环境参数设置
- PostgreSQL数据库级属性修改

# 数据库集群逻辑结构



PolarDB



每个数据库存储的对象（表、索引、视图等等）是独立的、私有的，每个数据库类似于每个房间，从房间中取东西，就需要到房间里面；同理，要访问某个数据库中的对象，就需要登录到指定的数据库中。

# PostgreSQL数据库结构



PolarDB



- 数据库集群是由PostgreSQL服务器管理的数据库的集合。PostgreSQL中的“数据库集群”一词并不意味着“一组数据库服务器”。PostgreSQL服务器在单个主机上运行，并管理单个数据库群集。
- 数据库是数据库对象的集合。在关系数据库理论中，数据库对象是用来存储或引用数据的数据结构。堆（heap）表是一个典型的例子，它有很多类似于索引、序列、视图、函数等等。在PostgreSQL中，数据库本身也是数据库对象，在逻辑上彼此分离。所有其他数据库对象（如表、索引等）都属于各自的数据库。

# PostgreSQL数据库属主

- PostgreSQL中的数据库属主属于创建者，只要有createdb的权限就可以创建数据库，数据库属主不一定拥有存放在该数据库中其它用户创建的对象访问权限。
- 数据库在创建后，允许public角色连接，即允许任何人连接。
- 数据库在创建后，不允许除了超级用户和owner之外的任何人在数据库中创建schema。
- 数据库在创建后，会自动创建名为public的schema，这个schema的all权限已经赋予了public角色，即允许任何人在里面创建对象，但对已存在的其它用户的表不具有任何权限。

# PostgreSQL数据库权限



PolarDB



- CREATE: 可以在指定数据库创建schema的权限
- CONNECT: 可以连接到指定数据库的权限
- TEMPORARY: 可以创建临时表的权限
- ALL: 指定数据库所有的权限

语法:

```
GRANT { { CREATE | CONNECT | TEMPORARY | TEMP } [, ...] | ALL [ PRIVILEGES ] }  
ON DATABASE 数据库名称 [, ...]  
TO role_specification [, ...] [ WITH GRANT OPTION ]
```

# PostgreSQL数据库权限

- 由于数据库在创建后，允许public角色连接，即允许任何人连接。所以如果要取消某个用户对指定数据库连接的权限，需要先取消public的连接权限，再取消该用户的连接权限。

--授权用户连接数据库的权限

```
grant connect on database db_name to user_name;
```

--撤销用户连接数据库的权限

```
revoke connect on database db_name from public;
```

```
revoke connect on database db_name from user_name;
```

--查看哪些用户有某个数据库的connect权限

```
select datname,datacl from pg_database where datname='db_name';
```

# PostgreSQL数据库环境设置



PolarDB



- PostgreSQL参数设置分为实例级、数据库级、用户级和会话级，而有些参数可以在所有级别中设置，优先级顺序为会话级>用户级>数据库级>实例级。

数据库参数配置语法：

ALTER DATABASE 名称 SET 配置参数 { TO | = } { 值 | DEFAULT }

ALTER DATABASE 名称 SET 配置参数 FROM CURRENT

ALTER DATABASE 名称 RESET 配置参数

ALTER DATABASE 名称 RESET ALL



# PostgreSQL数据库环境设置



PolarDB



配置示例（一）：

```
--设置数据库搜索路径:  
alter database postgres  
set search_path to "$user", public, schema_name;  
  
--配置连接某个库时可使用的工作内存  
alter database postgres set work_mem = '8MB';  
  
--配置连接某个库时可使用的维护内存  
alter database postgres set maintenance_work_mem TO '256MB';
```

# PostgreSQL数据库环境设置



PolarDB



配置示例（二）：

```
--配置连接某个库后使用的时区
alter database postgres set TimeZone to cet;

alter database postgres set DateStyle to SQL, DMY;（重新登录生效）

--配置连接某个库后执行语句最多时长（执行1秒超时）
alter database postgres set statement_timeout =1000;

--配置连接某个库后默认的客户端编码，配置客户端编码为gbk，适用于数据库编码
为utf8，应用程序编码为gbk的应用
alter database postgres set client_encoding to gbk;
```

# PostgreSQL数据库环境设置



PolarDB



配置示例（三）：

```
--配置某个库使用日志记录级别（设置后，对这个数据库的访问不记录日志）
alter database postgres set log_statement=none;

--配置连接某个库后的wal日志写盘级别（设置后，该库的更新操作只要求本地提交）
alter database postgres set synchronous_commit to local;

--配置连接某个库后禁用某个规划器（禁用indexonlyscan扫描）
alter database postgres set enable_indexonlyscan to off;
```

# PostgreSQL数据库环境设置



PolarDB



配置示例（四）：

```
--配置连接某个库后执行出错时中断连接（对新会话生效）
alter database postgres set exit_on_error to on;

--重新连接后
select pg_backend_pid();

--执行错误会导致连接中断
select * from d1;
```

# PostgreSQL数据库环境设置



PolarDB



配置示例（五）：

```
--查看所有个性化配置
\drds

--查询数据库的连接数限制只能查看数据字典表
select datname,datconnlimit from pg_database

--设置某个个性化设置为默认值
ALTER DATABASE postgres reset exit_on_error;

--设置所有个性化设置为默认值
ALTER DATABASE postgres reset ALL;
```

# PostgreSQL数据库环境设置



PolarDB



查询数据库配置:

```
postgres=# select datname, rolname, setconfig from pg_db_role_setting t1 left outer join
pg_database t2 on t1.setdatabase=t2.oid left outer join pg_authid t3 on t1.setrole=t3.oid;
-[ RECORD 1 ]-----
datname      | contrib_regression
rolname      |
setconfig    |
{lc_messages=C,lc_monetary=C,lc_numeric=C,lc_time=C,bytea_output=hex,timezone_abbreviat
s=Default}
-[ RECORD 2 ]-----
datname      |
rolname      |
setconfig    | {work_mem=8MB}
```

# PostgreSQL数据库属性修改



PolarDB



- 数据库的属性我们可以进行修改，修改范围是数据库名字、属主、表空间。

ALTER DATABASE 名称 RENAME TO 新的名称

ALTER DATABASE 名称 OWNER TO { 新的属主 | CURRENT\_USER | SESSION\_USER }

ALTER DATABASE 名称 SET TABLESPACE 新的表空间

# PostgreSQL数据库属性修改



PolarDB



示例:

--修改数据库名字

```
ALTER DATABASE newdb2 RENAME TO newdb3;
```

--修改数据库属主

```
ALTER DATABASE newdb3 OWNER TO u1;
```

--修改新表空间的名字

```
ALTER DATABASE newdb3 SET TABLESPACE new_tbl;
```



# 总结



PolarDB



PostgreSQL

- PostgreSQL数据库结构
- PostgreSQL数据库级权限管理
- PostgreSQL数据库级环境参数设置
- PostgreSQL数据库级属性修改

# 练习

- 1、创建一个数据库, 修改这个数据库的work\_mem参数.
- 2、将数据库的连接权限、创建schema的权限赋予给某个普通用户.
- 3、使用这个普通用户在这个数据库中创建schema. 在这个schema中创建表, 写入测试数据.
- 4、删除以上创建的数据库. 观察有什么报错? 如何解决?

