

VIEW (视图)

记录关系操作的函数是 view, 能存储、重用关系表达式.

```
create view viewname (col1,...coln)
as
select ...
```

↑不指定的话默认是 select 的列

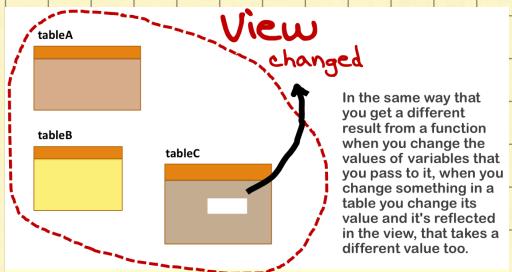
view实际上是一种查询, 若 select 正确, 就会返回有效关系.

Tables = variables

ROWS Remember that tables are relational variables, and that the rows they contain are nothing more than their "value"

values

view创建时只针对当前所涉及的表的结构、逻辑关系, 而不会拷贝内容.
若其中某表中的值改变了, view中相关的值也改变



```
create view vmovies
as select m.movieid,
         m.title,
         m.year_released,
         c.country_name
    from movies m
   inner join countries c
     on c.country_code = m.country
select *
  from vmovies
 where country_name = 'Italy'
```

当view创建后, 可将其当作表来用.

```
select *
  from (select m.movieid,
              m.title,
              m.year_released,
              c.country_name
         from movies m
        inner join countries c
          on c.country_code = m.country)
       vmovies
  where country_name = 'Italy'
```

在查询结果上两者等价.

view实际上创建了一种可选的数据模型.

DB中的view是持久的对象. 当新插入时, view不会自动更改.

Permanent object

Permanent structure

Describe

select *
 ^ col, col...
 X columns at creation!

Bad anyway

在实际生活中, view多数用在简化查询. 让用户只用写简单的SQL语句.

View在封装时会使得复杂度被隐藏, 因为不知道view是如何封装的.

Slower query to retrieve the same data



计算机的能力有限. 不要在复杂的view上再创建view.

Fewer simultaneous users served

Security

view 在 reports, user interface, security 上非常有效

DB 中如何管理 security: 访问 DB 要有权限 (可以是用户名、密码, 有的会绑定操作系统权限)

权限管理: DCL (Data Control language)

A privilege is given to a user account using this command:

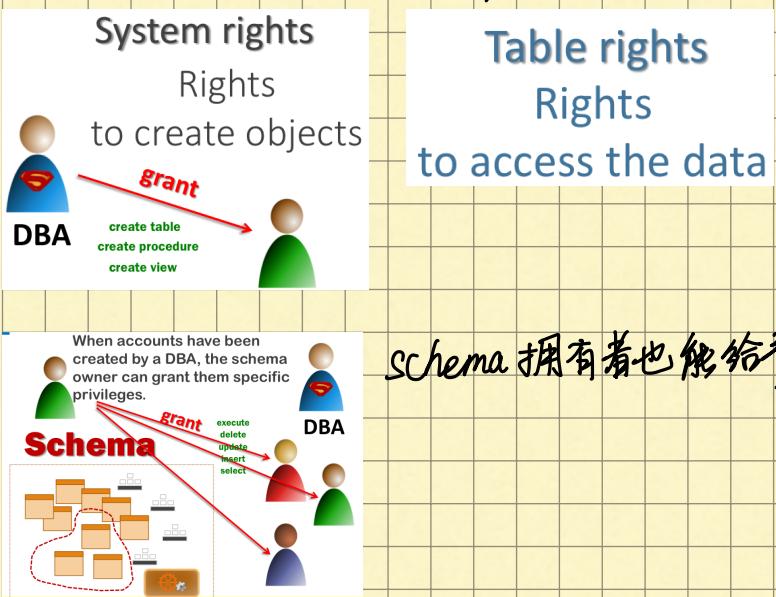
grant <right> to <account>

and can be taken back using this one:

revoke <right> to <account>

权限有两类: ① 系统权限: 让用户有权能使用 DDL, 改变 DB 结构 (少)

② 表权限: 能访问和更改部分 data (多, 每个人都有)



grant select on tablename to accountname
grant insert on tablename to accountname
grant update on tablename to accountname
grant delete on tablename to accountname
grant select, insert on tablename to accountname

可以单个赋, 也可以一次性同时赋,
update 只能给某些列的权限.

有些 DB 需要特别的其他权限才能赋 (如 PostgreSQL 要 schema 的 usage)

revoke privilege on tablename
from accountname 收回部分权限.

view 在 security 的作用.

由于 view 只有表中部分内容, grant select on view 能藏起敏感列.

The trick is to use a view that only shows what people are supposed to see, and grant SELECT on the view and not on the table.

grant select on view



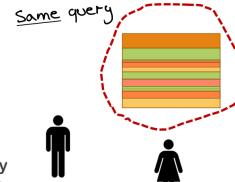
create view my_stuff
as
select * from stuff
where username = user

ORACLE
PostgreSQL
MySQL

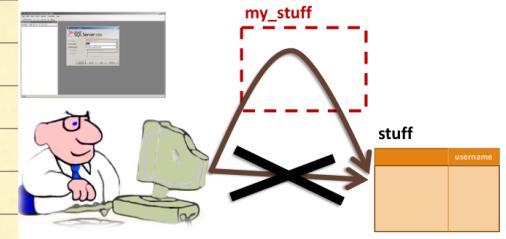
You can even hide rows by only returning rows "owned" by the user currently connected.

stuff	username

my_stuff

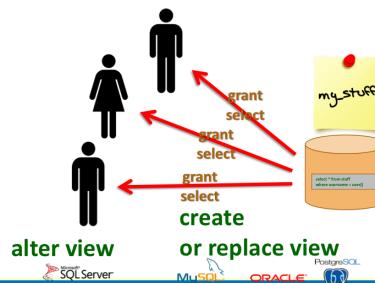


With such a view exactly the same query run by different users will return different rows.

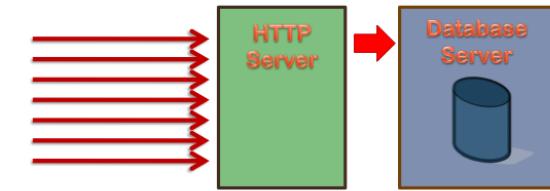


在修改view时要小心，若 drop 或 recreate view，会失去权限。

常用 create or replace 这一语句修改 view



这样操作就需要不同的个人帐户。



When accesses are run through a single connection as happens on a web server, it's not really interesting to use views for security. They can, however, be quite interesting for development, as we'll soon see.