



澳門城市大學
Universidade da Cidade de Macau
City University of Macau

設計計算工作室2

MySQL操作_視圖

主講人 | 澳門城市大學
City University of Macau

周帥
Shuai Zhou

內容

說明: 文中所有xxx都是你名字每個字拼音的首字母

- 視圖概述
- 創建視圖
- 查看視圖
- 修改視圖的定義
- 更新視圖 — update
- 更新視圖 — insert
- 更新視圖 — delete
- 刪除視圖
- with check option



視圖概述



視圖概述

視圖是從一個表或多個表中導出來的表，它是一種虛擬存在的表，並且表的結構和數據都依賴於基本表。可以對視圖進行查詢、修改、刪除等操作。

➤ 優點：

- **優化查詢語句**：簡化用戶對數據的理解，簡化對數據的操作。避免大量重複的操作。
- **安全性**：通過視圖，用戶只能查詢和修改他們所能見到的數據，其他部分的數據看不到。
- **邏輯數據獨立性**：視圖可以幫助用戶屏蔽真實表結構變化帶來的影響。



創建視圖



創建視圖

視圖中包含Select查詢的結果，因此視圖的創建基於select語句和已經存在的（1張或多張）數據表。

CREATE [OR REPLACE] [ALGORITHM={UNDEFINED|MERGE|TEMPTABLE}]

VIEW 視圖名 [(一些列)]

AS 選擇語句

[WITH [CASCADED|LOCAL] CHECK OPTION]

- CREATE, REPLACE 新建或替換
- ALGORITHM 視圖選擇算法
- 更新視圖時需遵守的約束



創建視圖——例子(基於單表創建視圖)

透過 PyMySQL 完成下列操作：

- 建立數據庫 w10_xxx
- 建立表 student 並添加數據：

| 表 student | sid | sname | chinese | math | age |
|-----------|-----|-------------|---------|-------|-----|
| 約束 | | | | | |
| 數據類型 | int | varchar(20) | float | float | int |
| 數據 | 1 | Amy | 80 | 78 | 20 |
| | 2 | Bob | 70 | 80 | 21 |
| | 3 | Cathy | 97 | 95 | 19 |



創建視圖——例子(基於單表創建視圖)

- 查詢student內的數據
- 基於student表創建視圖view_stud，結構如下：

| 列名 | sid | chinese | math | chinese+math |
|----|------|----------|-------|---------------|
| 取值 | =sid | =chinese | =math | =chinese+math |

- ```
CREATE VIEW view_stud
AS SELECT sid, chinese, math, chinese+math FROM student;
```

查看視圖view\_stud中的數據： 截圖2

- ```
SELECT * FROM view_stud;
```



創建視圖——例子(基於單表創建視圖)

- 基於student表創建視圖view2_stud（加總的列改名為total），結構如下，寫出該操作的SQL語句。

| 列名 | sid | chinese | math | total |
|----|------|----------|-------|---------------|
| 取值 | =sid | =chinese | =math | =chinese+math |

- SELECT * FROM view2_stud;

當前：

| 名 | 類型 | 內容 |
|------------|-------|-----------|
| student | table | 3 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |



創建視圖——例子(基於多表創建視圖)

- 建立表stud_info並添加數據：

| 表 stud_info | sid | class | address |
|-------------|-----|-------------|-------------|
| 約束 | | | |
| 數據類型 | int | varchar(10) | varchar(20) |
| 數據 | 1 | class01 | Macau |
| | 2 | class02 | Taiwan |
| | 3 | class03 | Beijing |

- 利用SELECT查看數據是否添加成功。

當前：

| 名 | 類型 | 內容 |
|------------|-------|-----------|
| student | table | 3 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |



創建視圖——例子(基於多表創建視圖)

創建view_class視圖，結構如下：

| 列名 | sid | name | class |
|----|-------------|--------------|-----------------|
| 列值 | student.sid | student.name | stud_info.class |

- CREATE VIEW view_class(sid,name,class)
AS SELECT student.sid, student.sname, stud_info.class FROM student, stud_info
WHERE student.sid=stud_info.sid;

當前：

- 查詢view_class的數據。

| 名 | 類型 | 內容 |
|------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |



查看視圖



查看視圖——例子

查看視圖字段信息的SQL語句：

DESC 視圖名; 或 Describe 視圖名;

例：

- DESC view_class;

| 名 | 類型 | 內容 |
|------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |



查看視圖——例子

查看視圖基本信息的SQL語句:

show table status like '視圖名'

例：

- SHOW TABLE STATUS LIKE 'view_class';
- SHOW TABLE STATUS LIKE 'view_class'\G

對比一下？

- SHOW TABLE STATUS LIKE 'student'\G

| 名 | 類型 | 內容 |
|------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |

- \G只能在命令行的MySQL中使用
- PyMySQL使用G可能會報錯



查看視圖——例子

查看創建視圖時的定義語句，和字符編碼：

show create view 視圖名;

例：

- SHOW CREATE VIEW view_class;

| 名 | 類型 | 內容 |
|------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自student |
| view_stud2 | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_view |



查看視圖——例子

對比一下以下三個查看視圖的語句，
返回的結果有何異同？

- desc view_class;
- show table status like 'view_class'\G
- show create view view_class \G

| 名 | 類型 | 內容 |
|------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |



修改視圖的定義



修改視圖的定義

如該視圖存在則替換它， 如不存在則建立它

CREATE OR REPLACE

[ALGORITHM={UNDEFINED|MERGE|TEMPTABLE}]

VIEW 視圖名 [(一些列)]

AS 選擇語句

[WITH [CASCADED|LOCAL] CHECK OPTION]



修改視圖——例子

查看視圖信息：

- DESC view_stud;

修改視圖：

- CREATE OR REPLACE
VIEW view_stud
AS SELECT * FROM student;

查看修改后的視圖和基本表信息內容一樣嗎？為什麼？

- DESC student;
- DESC view_stud;

| 名 | 類型 | 內容 |
|------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |



修改視圖——例子

修改視圖：

- CREATE OR REPLACE
VIEW view2_stud_son
AS SELECT sid, total FROM view2_stud;

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自 view2_stud |



修改視圖的定義

修改現有視圖

ALTER [ALGORITHM={UNDEFINED | MERGE | TEMPTABLE}]

VIEW 視圖名 [一些列]

AS 選擇語句

[WITH [CASCADED | LOCAL] CHECK OPTION]



修改視圖——例子

查看視圖信息

- DESC view_stud; DESC student;

修改視圖

- ALTER VIEW view_stud
AS SELECT sid, chinese FROM student WHERE chinese>=80;

查看修改后的視圖和基本表信息

- DESC view_stud; DESC student;

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自 view2_stud |



修改視圖——例子

修改視圖

- ALTER
VIEW view3_stud
AS SELECT sid, chinese FROM student WHERE
chinese<=80;

錯

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自 view2_stud |



更新視圖 — update

通過視圖來插入、刪除、更新基本表的數據



更新視圖 — update——例子

查看student和view_stud的數據內容：

- SELECT * FROM student;
- SELECT * FROM view_stud;

更新視圖view_stud：

- UPDATE view_stud SET chinese=100;

查看student和view_stud的數據內容：

- SELECT * FROM student;
- SELECT * FROM view_stud;

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自view2_stud |



更新視圖 — update——例子

更新視圖view_stud :

- UPDATE view_stud
- SET chinese=60;

查看student和view_stud的數據內容 :

- SELECT * FROM student;
- SELECT * FROM view_stud;
- 截圖8和9有何區別? 截圖9和10有何區別?

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自 student |
| view2_stud | view | 來自 student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自 view2_stud |



更新視圖 — insert

通過視圖來插入、刪除、更新基本表的數據



更新視圖 — insert——例子

查看student和view2_stud的數據內容：

- SELECT * FROM student;
- SELECT * FROM view2_stud;

更新表student：

- INSERT INTO student values (4,'David', 100, 100, 30);

查看student和view2_stud的數據內容：

- SELECT * FROM student;
- SELECT * FROM view2_stud;

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 4 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自view2_stud |



更新視圖 — insert——例子

查看student和view_stud的數據內容： 截圖14

- `SELECT * FROM student;`
- `SELECT * FROM view_stud;`

更新表student：

- `INSERT INTO view_stud VALUES (5, 99);`

查看student和view_stud的數據內容：

- `SELECT * FROM student;`
- `SELECT * FROM view_stud;`

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 5 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自view2_stud |



更新視圖 — delete

通過視圖來插入、刪除、更新基本表的數據



更新視圖 — delete——例子

查看student和view_stud2的數據內容：

- select * from student;
- select * from view2_stud;

更新表student：

- DELETE FROM view_stud2 WHERE computer=80;

查看student和view_stud2的數據內容：

- select * from student;
- select * from view2_stud;

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 4 |
| view_stud | view | 來自student |
| view2_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_info |
| view2_stud_son | view | 來自view2_stud |



刪除視圖

如該視圖存在則刪除它

DROP VIEW [IF EXISTS]

視圖1 [,視圖名2] [RESTRICT| CASCADE]



刪除視圖——例子

刪除view_stud2 :

- DROP VIEW IF EXISTS view2_stud;

查看是否刪除成功 :

- SELECT * FROM view2_stud;

| 名 | 類型 | 內容 |
|----------------|-------|-------------------|
| student | table | 3 |
| view_stud | view | 來自student |
| stud_info | table | 3 |
| view_class | view | student+stud_view |
| view_stud2_son | view | 來自view_stud2 |

刪除數據庫 :

- DROP DATABASE w10_xxx;



with check option



創建視圖

視圖中包含Select查詢的結果，因此視圖的創建基於select語句和已經存在的（1張或多張）數據表。

CREATE [OR REPLACE] [ALGORITHM={UNDEFINED|MERGE|TEMPTABLE}]

VIEW 視圖名 [(一些列)]

AS 選擇語句

[WITH [CASCADED|LOCAL] CHECK OPTION]

- CREATE, REPLACE 新建或替換
- ALGORITHM 視圖選擇算法
- 更新視圖時需遵守的約束



with check option ——例子

- 建數據庫w10a_xxx
- 建數據表如下（無任何約束）：
 - student(sid: int, age: int, chinese: int, computer: int)

- 數據：

| | sid | age | chinese | computer |
|---|-----|-----|---------|----------|
| 1 | 1 | 19 | 100 | 100 |
| 2 | 2 | 19 | 90 | 90 |
| 3 | 3 | 22 | 80 | 90 |
| 4 | 4 | 21 | 95 | 80 |



with check option ——例子

基於student表創建視圖view1, 年齡 ≥ 20 歲的人的資料 :

- CREATE VIEW view1 AS
SELECT * FROM student WHERE age ≥ 20 ;

基於view1創建視圖view2,
年齡 ≥ 20 歲且中文分數 ≥ 90 分的人的資料 :

- CREATE VIEW view2 AS
SELECT * FROM view1 WHERE chinese ≥ 90
with check option;

| 名 | 類型 | 內容 | With check option |
|---------|-------|------------|-------------------|
| student | table | 4 | |
| view1 | view | 來自 student | |
| view2 | view | 來自 view1 | Yes |



with check option ——例子

分別執行以下2行SQL語句，都成功嗎？為什麼？

- `update view1 set age=19 where sid=3`
- `update view2 set age=19 where sid=4;`

輸入以下3行SQL語句並把輸出結果截圖：

- `select * from student;`
- `select * from view1;`
- `select * from view2;`



with local check option ——例子

- 把student的內容還原，即如下表所示：

| sid | age | chinese | computer |
|-----|-----|---------|----------|
| 1 | 19 | 100 | 100 |
| 2 | 19 | 90 | 90 |
| 3 | 22 | 80 | 90 |
| 4 | 21 | 95 | 80 |



with local check option ——例子

基於student表創建視圖view3，年齡 ≥ 20 歲的人的資料：

- CREATE VIEW view3 AS
SELECT * FROM student WHERE age ≥ 20 ;

基於view3創建視圖view4，

年齡 ≥ 20 歲且中文分數 ≥ 90 分的人的資料：

- CREATE VIEW view4 AS
SELECT * FROM view3 WHERE chinese ≥ 90
with local check option;

| 名 | 類型 | 內容 | With check option |
|---------|-------|------------|-------------------|
| student | table | 4 | |
| view1 | view | 來自 student | |
| view2 | view | 來自 view1 | Yes |
| view3 | view | 來自 student | |
| view4 | view | 來自 view3 | local |



with local check option ——例子

分別執行以下2行SQL語句，都成功嗎？為什麼？

- `update view3 set age=19 where sid=3;`
- `update view4 set age=19 where sid=4;`

輸入以下3行SQL語句並把輸出結果截圖：

- `select * from student;`
- `select * from view3;`
- `select * from view4;`



沒有 with check option ——例子

- 把student的內容還原，即如下表所示：

| sid | age | chinese | computer |
|-----|-----|---------|----------|
| 1 | 19 | 100 | 100 |
| 2 | 19 | 90 | 90 |
| 3 | 22 | 80 | 90 |
| 4 | 21 | 95 | 80 |



沒有 with check option ——例子

基於student表創建視圖view5, 年齡 ≥ 20 歲的人的資料 :

- CREATE VIEW view5 AS
SELECT * FROM student WHERE age ≥ 20 ;

基於view3創建視圖view6,

年齡 ≥ 20 歲且中文分數 ≥ 90 分的人的資料 :

- CREATE VIEW view6 AS
SELECT * FROM view5 WHERE chinese ≥ 90 ;

| 名 | 類型 | 內容 | With check option |
|---------|-------|------------|-------------------|
| student | table | 4 | |
| view1 | view | 來自 student | |
| view2 | view | 來自 view1 | Yes |
| view3 | view | 來自 student | |
| view4 | view | 來自 view3 | local |
| view5 | view | 來自 student | |
| view6 | view | 來自 view5 | No |



沒有 with check option ——例子

分別執行以下2行SQL語句，都成功嗎？為什麼？

- update view5 set age=19 where sid=3;
- update view6 set age=19 where sid=4;

輸入以下3行SQL語句：

- select * from student;
 - select * from view5;
 - select * from view6;
-
- with check option、with local check option、以上兩者皆缺省，結果有何區別？由此可見，他們有甚麼作用？



練習作業



練習作業

利用Python的pymysql庫，完成以下操作：

- 創建數據庫w10_xxx
- 在數據庫w10_xxx中，創建數據表：
 - 「sc_xxx」（選課記錄表）
 - 「s_xxx」（學生表）
 - 「c_xxx」（課程表）

表的屬性名和約束如右所示，其中：

- ❑ sc_xxx的外鍵sid是源自s_xxx的sid
- ❑ sc_xxx的外鍵cid是源自c_xxx的cid

| s_xxx | sid | sname | age |
|-------|-----|-------------|-----|
| 數據類型 | int | varchar(20) | int |
| 約束 | 主鍵 | | |

| sc_xxx | sid | cid | score |
|--------|-------|-------|-------|
| 數據類型 | int | int | int |
| 約束 | 主鍵、外鍵 | 主鍵、外鍵 | |

| c_xxx | cid | cname | tid |
|-------|-----|-------------|-----|
| 數據類型 | int | varchar(20) | int |
| 約束 | 主鍵 | | |



練習作業

- 使用Python的pymysql庫，往s_xxx, c_xxx, sc_xxx數據表中增加以下数据：

Table s_xxx

| sid | sname | age |
|-----|---------|-----|
| 101 | John | 20 |
| 102 | Alice | 21 |
| 103 | Bob | 19 |
| 104 | Charlie | 22 |
| 105 | Emma | 20 |

Table c_xxx

| cid | cname | tid |
|-----|------------------|------|
| 201 | Database Basics | T001 |
| 202 | Advanced Math | T002 |
| 203 | Python Coding | T003 |
| 204 | English Speaking | T001 |
| 205 | Operating System | T004 |

Table sc_xxx

| sid | cid | score |
|-----|-----|-------|
| 101 | 201 | 95 |
| 101 | 202 | 88 |
| 102 | 201 | 98 |
| 102 | 203 | 99 |
| 103 | 202 | 100 |
| 103 | 201 | 99 |
| 104 | 204 | 90 |
| 104 | 202 | 97 |
| 105 | 205 | 99 |
| 105 | 202 | 100 |



練習作業

1. 查看 sc_xxx 表中的數據。
 2. 查看 s_xxx 表中的數據。
 3. 查看 c_xxx 表中的數據。
 4. 創建視圖view_young(內容為s_xxx表中年齡小於或等於25歲的人的所有資料)後；查看view_young的數據。
 5. 創建視圖view_xxx (內容為每個有選課的人的學號sid、姓名sname、平均分avg、最高分max、最低分min)後；查看view_xxx的數據。
 6. 創建視圖view_sc_xxx(內容為每條選課記錄的學號,人名,課程號,課程名,分數)後；查看view_sc_xxx的數據。
 7. 查看平均分最低的人的名字，及Ta的最高分那門課的分數。(提示：利用view_xxx)
 8. 查看平均分最高的人的名字，及Ta的最低分那門課的分數和課程號。
 9. 利用命令行打開mysql，查看hw07_xxx數據庫中 sc_xxx, s_xxx, c_xxx, view_xxx 的內容。
- 利用python的pymysql庫，刪除表sc_xxx、s_xxx、c_xxx。
 - 利用python的pymysql庫，刪除數據庫 w10_xxx 。

