Table of Contents

1. Views 1
1.1. Reason for views 1
1.2. Example of creating and using a view 1
2. Some rules for creating views 5
3. Information about views 5

1. Views

1.1. Reason for views

Suppose you have a fairly complex query dealing with customer orders that you need to run often. The query joins several tables and uses calculated columns etc. When you run this query, you filter for different order date ranges but the rest of the query is the same. You have asked the dba for help getting the query to be efficient and you do not want to keep writing the query over each time you need to run one of these reports. What you can do is create a database object called a view which encapsulates the complex query but still lets you add a filter for the order dates. When you create your query you can use the view in the From clause and not worry about the details of the join and calculated columns (not worrying about the details is a good thing!). You could also make the view available to programmers who may not have the experience to write complex queries.

A view is essentially a named, saved Select query. The view does not store any data; instead it stores the directions (the query) for getting the data. Once the view is created, you can use it as the table expression for other queries. The definition of the view is stored on the server and is retrieved when the view is used. Each time you use the view, it goes to the actual base tables and retrieves the current data. The view metadata is stored in the data dictionary.

There are two common reasons for using views

- to create a virtual table that presents a simplified set of column for queries. The view code might include joins and filters and calculated columns that the user of the view does not have to deal with directly
- to improve security by letting a user see only certain columns and specified rows of a table.

1.2. Example of creating and using a view

We will use a fairly simple query for our view. When you work on a view, work out the details of the select statement first.

I am going to create these views in the a_oe database.

```
Use a_oe;
```

Demo 01: Creating a view

```
create or replace view  a_oe.custReportGoodCredit_01 as
select
   cust_id as CustomerID
,  concat(cust_name_first,  ' ',  cust_name_last) as CustomerName
from  a_oe.customers
where credit_limit > 3000;
```

Demo 02: Describing the view. Note that the column names are the aliases set up in the view.

```
Desc a_oe.custReportGoodCredit_01;
+--------------+------------------+------+-----+---------+-------+
| Field        | Type             | Null | Key | Default | Extra |
+--------------+------------------+------+-----+---------+-------+
| CustomerID   | int(10) unsigned | NO   |     | NULL    |       |
| CustomerName | varchar(51)      | YES  |     | NULL    |       |
+--------------+------------------+------+-----+---------+-------+
```

Demo 03: Using the view

```
Select CustomerName, CustomerID
From a_oe.custReportGoodCredit_01
Order by CustomerID;
+-------------------+------------+
| CustomerName      | CustomerID |
+-------------------+------------+
| Arnold McGold     |     400300 |
| Sally Williams    |     403000 |
| Elisha Otis       |     403010 |
| Alexis Hamilton   |     403050 |
| James Stevenson   |     403100 |
| JAMES Stevenson   |     403500 |
| Mary O'Leary      |     403750 |
| Mary O'Leary      |     403760 |
| Frederick Olmsted |     404000 |
. . . rows omitted
```

Note that the person using this view cannot see the credit-limit value.

Data displayed through a view will reflect the current data in the table at the time the query was run; not the data at the time the view was defined. A person using a view as the table expression might not be aware that this is a view and not a base table.

Demo 04: A more complex view. this includes casting a datetime attribute to a date, concatenating attributes, and creating an calculated expression

```
Create or replace view  a_oe.ordReport_01 as
Select
   ord_id      as OrderID
,  cast(ord_date as date)    as OrderDate
,  cust_id     as CustomerID
,  concat(cust_name_first,  ' ',  cust_name_last) as CustomerName
,  prod_id     as ItemPurchased
,  prod_name   as ItemDescription
,  quoted_price * quantity_ordered as LineTotal
From  a_oe.customers
Join  a_oe.order_headers  using (cust_id)
Join  a_oe.order_details using(ord_id)
Join  a_prd.products using(prod_id)
Where quoted_price > 0 and quantity_ordered > 0;
```

Demo 05: Describing the view. One thing to check is that the data types of the columns in the view are appropriate. Do not over format the columns you wish to filter; do not output a formatted number column if you might wish to filter on it.
The output for desc with a view is the same format as with a table. The point is that a table and a view should look the same to a user.

```
Desc a_oe.ordReport_01;
+-----------------+------------------------+------+-----+---------+-------+
| Field           | Type                   | Null | Key | Default | Extra |
+-----------------+------------------------+------+-----+---------+-------+
| OrderID         | int(10) unsigned       | NO   |     | NULL    |       |
| OrderDate       | date                   | YES  |     | NULL    |       |
| CustomerID      | int(10) unsigned       | NO   |     | NULL    |       |
| CustomerName    | varchar(51)            | YES  |     | NULL    |       |
| ItemPurchased   | int(10) unsigned       | NO   |     | NULL    |       |
| ItemDescription | varchar(25)            | NO   |     | NULL    |       |
```

```
| LineTotal       | decimal(16,2) unsigned | NO   |     | 0.00    |       |
+-----------------+------------------------+------+-----+---------+-------+
```

Demo 06: Using the view

```
Select orderid, orderdate, itempurchased, linetotal
From a_oe.ordreport_01;
+---------+------------+---------------+-----------+
| OrderID | OrderDate  | ItemPurchased | LineTotal |
+---------+------------+---------------+-----------+
|     378 | 2013-06-14 |          1120 |   2250.00 |
|     378 | 2013-06-14 |          1125 |   2250.00 |
|     106 | 2012-10-01 |          1060 |    255.95 |
|     113 | 2012-11-08 |          1080 |     22.50 |
|     119 | 2012-11-28 |          1070 |    225.00 |
|     301 | 2013-06-04 |          1100 |    205.00 |
|     112 | 2012-11-08 |          1110 |     99.98 |
. . .  rows omitted
```

Demo 07: Using the view with a filter

```
Select orderid, orderdate, itempurchased
From a_oe.ordreport_01
Where orderdate < '2013-01-01';
+---------+------------+---------------+
| OrderID | OrderDate  | ItemPurchased |
+---------+------------+---------------+
|     106 | 2012-10-01 |          1060 |
|     113 | 2012-11-08 |          1080 |
|     119 | 2012-11-28 |          1070 |
|     112 | 2012-11-08 |          1110 |
|     114 | 2012-11-08 |          1130 |
|     115 | 2012-11-08 |          1000 |
|     115 | 2012-11-08 |          1120 |
|     115 | 2012-11-08 |          1080 |
. . .  rows omitted
```

Demo 08: Using the view with a filter

```
Select orderid, orderdate, itempurchased
From a_oe.ordreport_01
Where month(orderdate) in (1,2,3)
And year(orderdate) in (2013)
Limit 8;
+---------+------------+---------------+
| OrderID | OrderDate  | ItemPurchased |
+---------+------------+---------------+
|     519 | 2013-03-04 |          1020 |
|     519 | 2013-03-04 |          1110 |
|     505 | 2013-01-12 |          1080 |
|     505 | 2013-01-12 |          1110 |
|     505 | 2013-01-12 |          1060 |
|     508 | 2013-02-01 |          1152 |
|     508 | 2013-02-01 |          1152 |
|     509 | 2013-02-12 |          1090 |
+---------+------------+---------------+
```

Demo 09: Using the view with a filter

```
Select orderid, orderdate, itempurchased, linetotal
From a_oe.ordreport_01
```

```
Where linetotal < 100
Limit 8;
+---------+------------+---------------+-----------+
| OrderID | OrderDate  | ItemPurchased | LineTotal |
+---------+------------+---------------+-----------+
|     113 | 2012-11-08 |          1080 |     22.50 |
|     112 | 2012-11-08 |          1110 |     99.98 |
|     519 | 2013-03-04 |          1020 |     64.75 |
|     519 | 2013-03-04 |          1110 |     49.99 |
|     115 | 2012-11-08 |          1080 |     25.00 |
|     117 | 2012-11-28 |          1150 |     19.96 |
|     117 | 2012-11-28 |          1030 |     27.00 |
|     108 | 2012-10-02 |          1080 |     22.50 |
+---------+------------+---------------+-----------+
```

Demo 10: Join the view to a base table

```
Select orderid
, itempurchased
, warehouse_id
, quantity_on_hand
, linetotal
From a_oe.ordreport_01 RPT
Join a_prd.inventory  PRD  on RPT.itempurchased = PRD.prod_id
Where linetotal >500
And quantity_on_hand < 100;
+---------+---------------+--------------+------------------+-----------+
| OrderID | ItemPurchased | warehouse_id | quantity_on_hand | LineTotal |
+---------+---------------+--------------+------------------+-----------+
|     378 |          1125 |          200 |               10 |   2250.00 |
|     118 |          1125 |          200 |               10 |   1900.00 |
|     128 |          1060 |          125 |                2 |    511.90 |
|     307 |          1125 |          200 |               10 |   2250.00 |
|     312 |          1040 |          250 |               35 |   3000.00 |
|     312 |          1050 |          100 |                0 |   2500.00 |
|     312 |          1060 |          125 |                2 |   1405.00 |
|     312 |          1060 |          125 |                2 |   2500.00 |
+---------+---------------+--------------+------------------+-----------+
```

Demo 11: Alternate syntax when creating a view -this defines the column names in the view header rather than in the select

```
create or replace view  a_oe.ordreport_02(orderid, orderdate, customerid, linetotal) as
select
   ord_id
,  ord_date
,  cust_id
,  quoted_price * quantity_ordered
From  a_oe.order_headers
Join  a_oe.order_details using(ord_id)
Join  a_prd.products using(prod_id)
Where quoted_price > 0 and quantity_ordered > 0
Order by ord_id
;
```

If you are starting to see that you can use a view in the same way you would use a table, then you have gotten the point of views.

2. Some rules for creating views

A view is an object created in a specific database.

- You can use either the Create View syntax or the Create or Replace View syntax. If the name you select for the view is already used by another view, the Create View syntax will report an error; the Create or Replace View syntax will replace the current definition of the view.
- You cannot have a view with the same name as a table.
- You cannot create a view on a table that does not exist. IF you create a view on a table and then the table is dropped the view is not dropped but you will get an error when you try to use that view.
- Since a view is a stored object, you can drop the view with a drop view command.

```
drop view my_view;
```

- Your user account has to have permissions to create views.

- You can assign new names for the columns when you define a view or you can use the names in the underlying tables.
- If you have a calculated column in the underlying select, you provide an alias for the column so that it can be referenced through the view.
- The column names in the view must be unique.
- If you define the columns in the view header, the number of columns in the view definition must match the number of columns in the select used to define the view

- A view cannot be created that uses a user-defined variable.
- A view cannot refer to a temporary table
- Do not create a view using Select *; the view will be stored with the * expanded into the column names at the time the view was created and will not be updated if the underlying tables are changed.
- MySQL allows an order by in a view definition.

We will talk about views again when we talk about updating data.

3. Information about views

The Show tables command includes the views in the result. It makes no distinctions between tables and views.

```
show tables;
```

If you use `show tables` or `show table from` you will see the names of the tables and views. There is no distinction made in the list between tables and views

```
show tables from a_oe;
+-------------------------+
| Tables_in_a_oe          |
+-------------------------+
| credit_ratings          |
| cust_orders             |
| customers               |
| custreportgoodcredit_01 |
| order_details           |
| order_headers           |
| ordreport_01            |
| ordreport_02            |
| shipping_modes          |
+-------------------------+
```

Demo 12: You can see the sql used to create a view with Show Create View; (This is not pretty)

```
show create view a_oe.ordreport_01\G
*************************** 1. row ***************************
                View: ordreport_01
         Create View: CREATE ALGORITHM=UNDEFINED DEFINER=`a_rose`@`localhost` SQL
SECURITY DEFINER VIEW `ordreport_01` AS select `a_oe`.`order_headers`.`ord_id` AS
`OrderID`,cast(`a_oe`.`order_headers`.`ord_date` as date) AS
`OrderDate`,`a_oe`.`customers`.`cust_id` AS
`CustomerID`,concat(`a_oe`.`customers`.`cust_name_first`,'
',`a_oe`.`customers`.`cust_name_last`) AS
`CustomerName`,`a_oe`.`order_details`.`prod_id` AS
`ItemPurchased`,`a_prd`.`products`.`prod_name` AS
`ItemDescription`,(`a_oe`.`order_details`.`quoted_price` *
`a_oe`.`order_details`.`quantity_ordered`) AS `LineTotal` from (((`a_oe`.`customers`
join `a_oe`.`order_header` on((`a_oe`.`customers`.`cust_id` =
`a_oe`.`order_headers`.`cust_id`))) join `a_oe`.`order_details`
on((`a_oe`.`order_headers`.`ord_id` = `a_oe`.`order_details`.`ord_id`))) join
`a_prd`.`products` on((`a_oe`.`order_details`.`prod_id` =
`a_prd`.`products`.`prod_id`))) where ((`a_oe`.`order_details`.`quoted_price` > 0) and
(`a_oe`.`order_details`.`quantity_ordered` > 0))character_set_client:
latin1collation_connection: latin1_swedish_ci
1 row in set (0.00 sec)
```

Demo 13: If you want to see the names of tables and views- distinguishing the two, use information_schema.tables

```
Select table_schema, table_name, table_type
From information_schema.tables
Where table_schema = "a_oe";
+--------------+-------------------------+------------+
| table_schema | table_name              | table_type |
+--------------+-------------------------+------------+
| a_oe         | credit_ratings          | BASE TABLE |
| a_oe         | cust_orders             | VIEW       |
| a_oe         | customers               | BASE TABLE |
| a_oe         | custreportgoodcredit_01 | VIEW       |
| a_oe         | order_details           | BASE TABLE |
| a_oe         | order_headers           | BASE TABLE |
| a_oe         | ordreport_01            | VIEW       |
| a_oe         | ordreport_02            | VIEW       |
| a_oe         | shipping_modes          | BASE TABLE |
+--------------+-------------------------+------------+
```