Lesson 18: Using Views

SQL in 10 Min

11/08/2022

What are views?

Virtual table

- Why use views?
  - Reuse SQL statements
  - Can be used without having to know the underlying query
  - Expose parts of a table instead of complete tables
- Things to remember
  - Execute every time the view is used
  - Data source change, return will change

Rules and restrictions

- Uniquely named
- Security access
- Views can be nested
- Views cannot be indexed

How to use Views?

- Create views
- Filter views

```
SELECT cust_name, cust_contact
FROM ProductCustomers
WHERE prod_id = 'RGAN01';
```

- Reformat data

```
CREATE VIEW VendorLocations AS
SELECT RTRIM(vend_name) + ' (' + RTRIM(vend_country) + ')'
       AS vend_title
FROM Vendors;
```

- Calculate fields

Using Views to Simplify Complex Joins

One of the most common uses of views is to hide complex SQL, and this often involves joins. Look at the following statement:

Input ▾

Click here to view code image

```
CREATE VIEW ProductCustomers AS
SELECT cust_name, cust_contact, prod_id
FROM Customers, Orders, OrderItems
WHERE Customers.cust_id = Orders.cust_id
 AND OrderItems.order_num = Orders.order_num;
```

```
CREATE VIEW OrderItemsExpanded AS
SELECT order_num,
       prod_id,
       quantity,
       item_price,
       quantity*item_price AS expanded_price
FROM OrderItems
```

Lesson 19: working with stored proc

SQL in 10 Min

11/08/2022

What is stored proc?

Collections of one or more SQL statements

- Why to use stored proc?
  - Simplify complex operations by encapsulating processes into one single unit
  - Reuse code
  - Performance

How to use stored proc?

- Executing stored proc

```
EXECUTE AddNewProduct('JTS01',
                      'Stuffed Eiffel Tower',
                      6.49,
                      'Plush stuffed toy with
-the text La Tour Eiffel in red white and blue');
```

- Creating stored proc

```
CREATE PROCEDURE MailingListCount  (
  ListCount OUT INTEGER
)
IS
v_rows INTEGER;
BEGIN
    SELECT COUNT(*) INTO v_rows
    FROM Customers
    WHERE NOT cust_email IS NULL;
    ListCount := v_rows;
END;
```

```
var ReturnValue NUMBER
EXEC MailingListCount(:ReturnValue);
SELECT ReturnValue;
```